

noeud LoRa alimenté par énergie solaire

une solution IdO modulaire, compacte et polyvalente

Dr. Claus Kühnel (Suisse)

Les nœuds de capteurs fonctionnant de manière autonome requièrent une communication sans fil, et une source d'énergie autonome. Dans cet article nous vous présentons un nœud IdO alimenté par une batterie, tamponné par une cellule solaire, et doté d'une connectivité LoRaWAN. Étant donné la disponibilité de nombreux modules de capteurs et l'existence d'un support logiciel robuste, il est possible de modifier ce projet selon vos besoins.

Je m'appuie sur l'écosystème WisBlock développé par RAKwireless pour développer un nœud LoRa alimenté par énergie solaire et adapté à la connexion de capteurs. La carte de base RAK19007 WisBlock 2^{ème} génération constitue une excellente base pour intégrer la carte contrôleur WisBlock Core ainsi que différents modules de capteurs. Le RAK4631-R est un WisBlock Core basse consommation basé sur un microcontrôleur nRF52840 de Nordic Semiconductor et une puce LoRa SX1262 de Semtech, offrant une connectivité Bluetooth



Figure 1. Carte RAK19007 WisBlock Base Board 2nd Gen. (Source : RAKwireless)

LoRa et LoRaWAN

LoRa, pour « Long Range », est une technologie de communication sans fil conçue pour transmettre des données sur de longues distances avec une faible consommation d'énergie, ce qui la rend idéale pour les applications IdO. LoRaWAN, un protocole basé sur LoRa, offre un cadre standardisé et évolutif de gérer la connectivité des appareils LoRa sur de longues distances. Ensemble, LoRa et LoRaWAN permettent de faciliter le déploiement de réseaux économies en énergie à grande échelle adaptés à une multitude d'applications. Cet article explore en détail cette technologie prometteuse.

Low Energy (BLE) et LoRaWAN. (Voir l'encadré « LoRa et LoRaWAN »).

Le RAK4631-R intègre l'interface unifiée RAKwireless V3 (RUI3), ce qui simplifie considérablement le développement de logiciels pour les nœuds IdO. J'utilise l'API RUI3 pour programmer les applications. En outre, je démontre la configuration du dispositif IdO avec l'application WisToolBox. Pour obtenir des informations plus détaillées sur l'écosystème WisBlock, je vous invite à consulter le site web de RAKwireless [1] et mon livre électronique, *IoT-Projects for Makers* [2], qui offre une introduction et des exemples pratiques d'application.

Alimentation électrique

Le RAK19007 est une carte de base WisBlock 2^{ème} génération qui permet de connecter différents modules WisBlock. Il est doté d'un connecteur USB Type-C, de connecteurs pour l'alimentation électrique, et d'une interconnexion avec les modules associés. La **figure 1** montre une vue de dessus d'un RAK19007. La section à l'extrême gauche montre les options d'alimentation disponibles avec les connecteurs JST pour l'alimentation solaire et la batterie ainsi que l'USB-C pour l'alimentation et le chargement de programme. Le module prend en charge l'alimentation par batterie de faible puissance, la charge de la

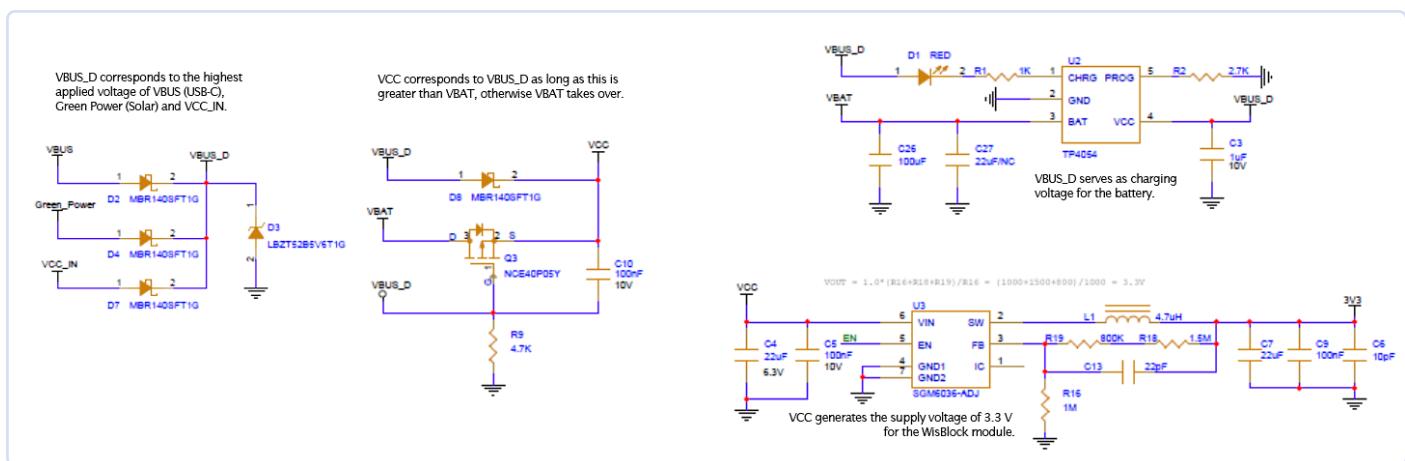


Figure 2. Alimentation du RAK19007. (Extrait du schéma de circuit de la carte de base. (Source: RAKwireless)

batterie au lithium (LiPo) et la charge par panneau solaire d'une batterie LiPo connectée. La figure 2 illustre les principaux éléments de l'alimentation d'un RAK19007. Le RAK19007 peut être alimenté de trois manières : par VBUS fournie via l'USB-C, par l'énergie fournie par une cellule solaire ou par une tension externe VCC_IN. La source ayant la tension la plus élevée est commutée sur VBUS_D. Tant que VBUS_D est supérieure à la tension de la batterie VBAT, VBUS_D détermine la tension VCC. Si VBUS_D est inférieure ou inexiste, alors VBAT détermine la tension d'entrée pour générer la tension système de 3,3 V pour les modules WisBlock, y compris le WisBlock Core, par un convertisseur abaissant de tension à très faible consommation SGM6036.

Gestion de l'énergie

Le RAK19007 ne dispose pas d'un système de gestion de l'énergie comme l'AXP192 ; il faut

donc se baser sur les tensions mesurables pour évaluer le niveau de charge de la batterie LiPo. La tension de la batterie VBAT est mesurée par un diviseur de tension relié à l'entrée analogique AIN0. L'entrée analogique AIN1 est accessible via le connecteur J11 pour mesurer la tension de l'alimentation solaire. Un diviseur de tension est également nécessaire. Dans un test pratique, je mesurerai les tensions `Vsol` et `Vbat` et utiliserai ces mesures pour calculer le rapport `State = Vsol/Vbat`. Si `State > 1`, cela indique que la batterie est chargée et que le panneau solaire fournit de l'énergie à l'appareil. Si `State < 1`, c'est la batterie qui alimente l'appareil, signifiant qu'elle se décharge. Utiliser l'interface RAKwireless Unified Interface V3 (RUI3) rend la création d'une application IoT avec un micrologiciel personnalisé particulièrement aisée. Voici le code nécessaire pour recueillir les valeurs des tensions `Vsol` et `Vbat` :

```
float Vsol = readADC(AIN1);
float Vbat = api.system.bat.get();
float State = Vsol/Vbat;
```

Configuration

Avec le programme *RAK4631-R_Sensor-Test.ino*, j'ai d'abord testé l'ADC interne ainsi que les modules RAK1903 (lumière ambiante) et RAK14001 (LED RGB) dont le schéma fonctionnel est représenté dans la figure 3. Ce programme est disponible sur mon dépôt GitHub [3]. Le module de capteur de lumière ambiante RAK1903 WisBlock surveille les variations de lumière ambiante tout au long de la journée. Avec la LED RGB, le RAK14001 offre une option de signalisation en plus des LED bleues et vertes de la carte de base. Ces deux modules sont facultatifs et non essentiels pour atteindre les objectifs décrits dans l'article.

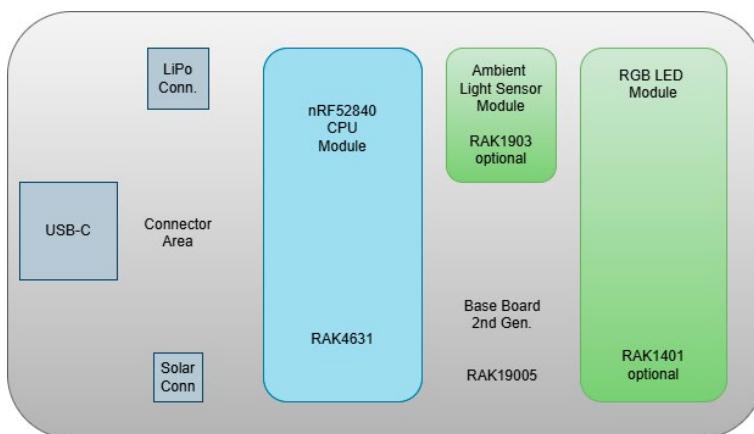


Figure 3. Appareil testé.

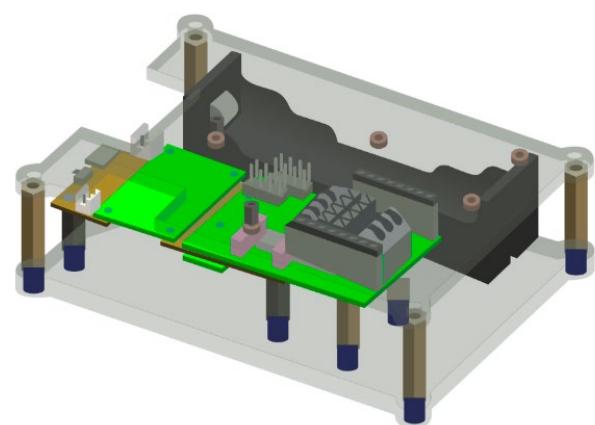


Figure 4. Boîtier acrylique transparent RAKBox-B5. (Source : RAKwireless)



Figure 5. Variantes de la batterie LiPo NCR18650 (Source : RAKwireless)

Pour assurer un fonctionnement sécurisé du dispositif de test, j'utilise un boîtier acrylique transparent RAKBox-B5, bien que celui-ci ne soit pas étanche (**figure 4**). Ce boîtier est avantageux car il intègre un support pour une batterie LiPo de type 18650. Je préfère utiliser ce format de batterie LiPo car il est plus pratique à manipuler en raison de la conception du boîtier. Les batteries LiPo 18650 sont disponibles en diverses capacités, et un aperçu des différents types et fabricants peut être trouvé sur le site [4]. Le support de batterie dans ce boîtier est spécifiquement conçu pour des batteries LiPo de taille « courte ». Référez-vous à la figure 5 pour voir les différentes variantes des batteries LiPo 18650.

Panneau solaire

Selon les tests de batterie publiés sur [4], une batterie LiPo de 3500 mAh et un courant de décharge de 1000 mA à une tension de 2,5 V offre une autonomie de 200 h. Grossièrement estimé, un courant de décharge de 1 mA pourrait permettre une durée d'utilisation estimée à 138 jours. La réduction de la consommation électrique implique que le nœud IoT demeure principalement en mode veille, ne s'activant que pour les mesures et la transmission de données. Pour assurer le fonctionnement autonome du nœud IdO sur une période donnée, il est crucial d'équilibrer la décharge de la batterie. La carte de base RAK19007 est capable de recharger la batterie lorsqu'elle est connectée à un panneau solaire sous une exposition solaire adéquate. Un panneau solaire MicroUSB revolt constitue une solution efficace pour le stockage d'énergie solaire, illustré dans la **figure 6**.

Le panneau solaire MicroUSB revolt illustré à la **figure 6** est composé de cellules solaires

monocristallines, réputées pour leur efficacité et leur longévité. Il fournit une tension de sortie de 5 V à un panneau solaire avec une puissance de charge de 5 W. Avec un câble d'une longueur de 3 mètres, son support de montage flexible et sa protection IP65, il est parfaitement adapté à une utilisation en extérieur. En raison de la tension système de 3,3 V du module de base RAK4631 utilisé, nous avons besoin d'un diviseur de tension pour mesurer la tension du panneau solaire. À cette fin, j'utilise une carte breakout microUSB, comme le montre la **figure 7**, pour ajouter le diviseur de tension. Vsol et GND sont connectés au connecteur JST de la carte de base RAK19007. La tension VAIN1 est acheminée à l'entrée analogique AIN1 sur l'en-tête J11. La **figure 8** montre l'équipement de test complet pour surveiller les tensions du panneau solaire et de la batterie LiPo.

Programmes d'application

Le programme d'application est conçu pour interroger les tensions de la cellule solaire et de la batterie LiPo et transférer les résultats sans fil. Pour réduire la consommation de courant de l'appareil, le noyau RAK4631 doit être mis en veille la plupart du temps. Grâce à l'API RUI3, l'implémentation de ce mode de veille est simplifiée. Les programmes d'application présentés dans les chapitres suivants diffèrent des programmes Arduino standard. La fonction `loop()` exécute en continu les opérations, ici elle sert uniquement à mettre le RAK4631 en veille. Un gestionnaire contrôlé par un timer exécute toutes les fonctionnalités pendant les phases actives. J'ai développé deux versions de l'application : une transmet les mesures via BLE UART et l'autre via LoRaWAN.



Figure 6. Panneau solaire MicroUSB de revolt.

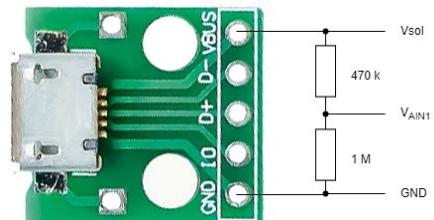


Figure 7. La carte Breakout microUSB.

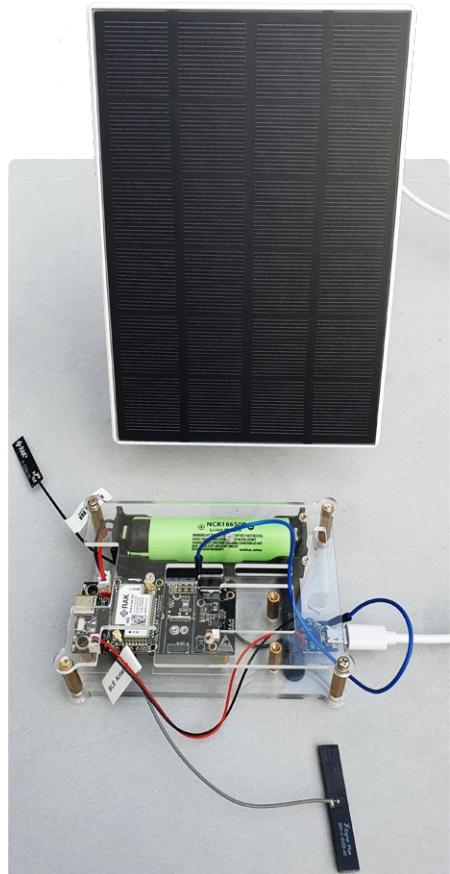


Figure 8. Équipement de test complet.

Application BLE

Après l'initialisation par `setup()`, un gestionnaire contrôlé par un timer interroge les tensions et stocke les données dans un paquet envoyé sans fil via BLE. Comme mentionné, la fonction `loop()` remet le RAK4631 en mode veille. Le programme *RAK4631-R_Sleep_Test_BLE.ino* est disponible sur GitHub [5]. Pendant la période de veille, j'ai mesuré la consommation de courant sur une tension source de 3,8 V de 270 μ A. Pour visualiser les données reçues par BLE, j'utilise l'application Android *Serial Bluetooth Terminal 1.46* [6] de Kai Morich, disponible sur Google Play Store. La **figure 9** montre l'état de la tension après la mise sous tension le soir. State < 1 signifie que la batterie alimente l'appareil et sera déchargée. Le jour suivant était ensoleillé en octobre, State > 1, et le panneau solaire a fourni suffisamment de tension pour alimenter l'appareil et recharger la batterie (**figure 10**).

Application LoRaWAN

Avant de programmer l'application LoRaWAN, j'ai utilisé l'application de bureau WisToolBox pour configurer le cœur RAK4631. L'application de bureau WisToolBox communique avec les appareils WisBlock via USB, et après avoir sélectionné le bon port USB et le module central RAK4631, nous pouvons voir les informations sur l'appareil sur le tableau de bord (**figure 11**). Vous pouvez régler tous les paramètres requis pour la configuration LoRaWAN via les paramètres du menu, comme le montrent les **figure 12** et **figure 13**. Les paramètres définis dans WisToolBox sont envoyés au noyau RAK4631 et stockés dans la mémoire Flash. Dans le programme d'application, leur réglage n'est plus nécessaire. Dans mon programme, disponible sur GitHub [7], *RAK4631-R_Sleep_Test_LoRaWAN.ino*, j'ai utilisé la méthode traditionnelle de déclaration explicite. Pendant la période de veille, j'ai mesuré la consommation de courant sur une tension source de 3,8 V de 235 μ A.

The Things Stack

J'utiliserai ici le cluster The Things Stack Sandbox (TTSS) pour l'Europe (eu1), autrefois The Things Stack (Community Edition, TTSC), comme serveur de réseau LoRaWAN (LNS). Ce cluster, destiné aux tests et expérimentations à petite échelle, est non commercial et open-source, supporté par The Things Network [10].

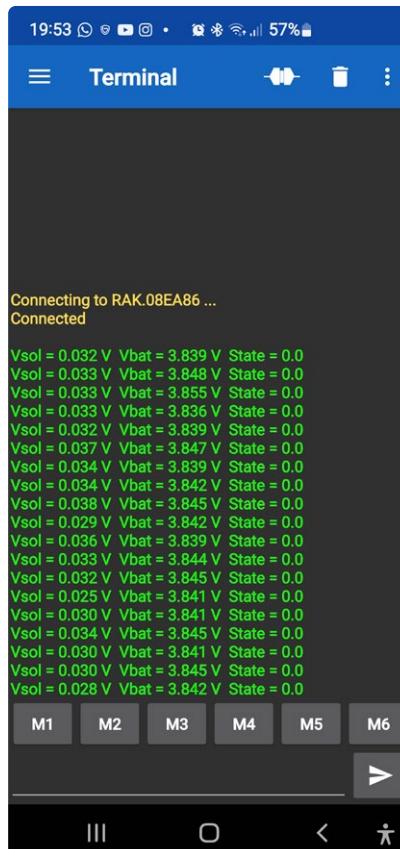


Figure 9. État de la tension après la mise sous tension pendant la soirée.

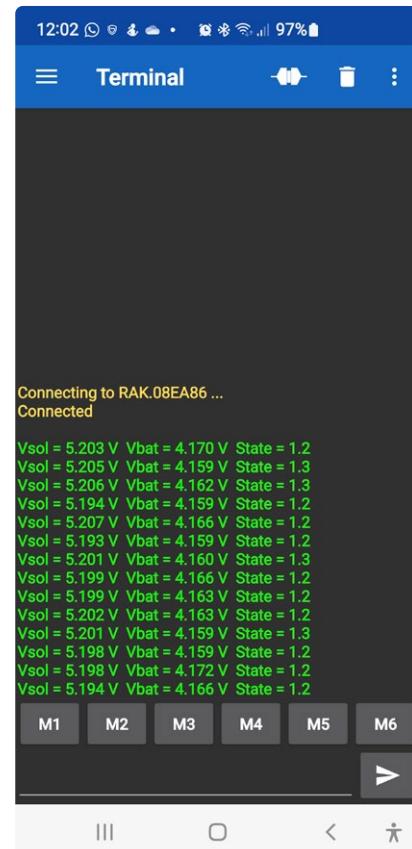


Figure 10. État de la tension lors d'une journée ensoleillée en octobre.

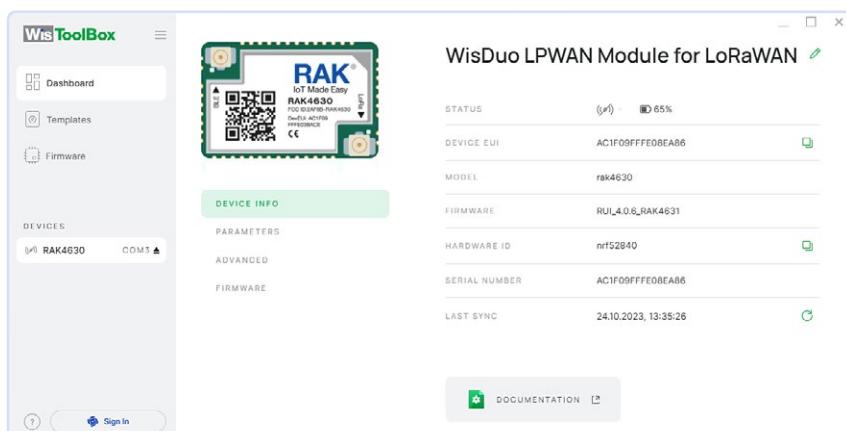


Figure 11. Tableau de bord WisToolBox - Informations sur l'appareil.

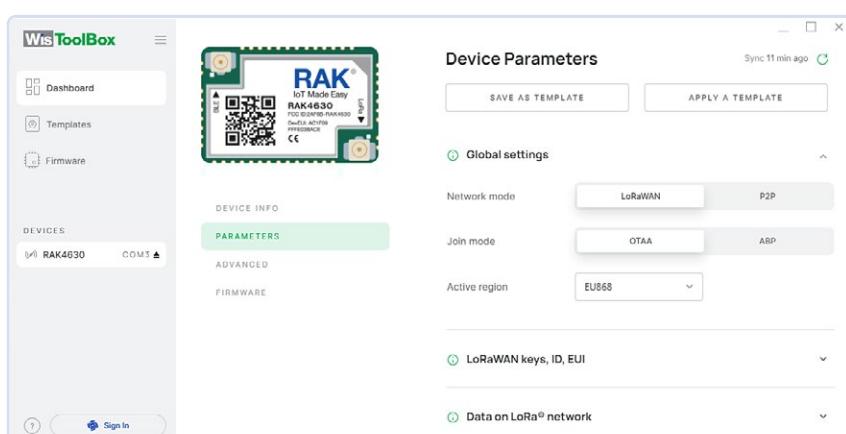


Figure 12. Tableau de bord WisToolBox - Paramètres généraux.

TTSS nécessite l'enregistrement de mon appareil ; les clés LoRaWAN, etc. montrées dans la **figure 12** sont nécessaires pour l'enregistrement. Vous pouvez également trouver des détails sur TTSS en tant que serveur de réseau LoRaWAN, et Datacake [8] en tant qu'outil de visualisation, dans mon livre intitulé *Develop and Operate Your LoRaWAN IoT Nodes* [9]. Sur TTSS, une *politique d'utilisation équitable* s'applique, qui limite le temps d'antenne en liaison montante à 30 secondes par jour (24 heures) par nœud et les messages en liaison descendante à 10 messages par jour par nœud. La charge utile et la fréquence du téléchargement déterminent le temps d'antenne sur la liaison montante. Vous pouvez obtenir des informations détaillées sur la fréquence de téléchargement possible à l'aide d'un calculateur de temps d'antenne [11]. Avec une charge utile de cinq octets et un débit de données DR5 (SF7BW125), le temps d'antenne résultant est de 51,5 ms. Entre deux téléchargements, il doit y avoir un temps de 148,2 secondes minimum pour respecter la politique d'utilisation équitable. Le temps de téléchargement dans le programme d'application est de 5 min. Après avoir enregistré et démarré l'application avec succès, la console TTSS affiche les messages de liaison montante reçus (**figure 14**). Le décodeur de charge utile utilisé est également disponible sur GitHub.

Résultats

Les détails de l'événement de cette sortie de console sont listés dans le **listing 1**, qui confirme que les spécifications sont respectées. La bande passante, le facteur d'étalement et le temps d'antenne résultant sont conformes aux attentes et marqués en gras. Pour visualiser les données reçues par le serveur réseau LoRaWAN, j'utilise Datacake, qui peut facilement s'intégrer à TTSS. Étudiez la documentation en ligne [12] ou les exemples dans le livre mentionné ci-dessus. Après cette intégration, vous pouvez créer un tableau de bord Datacake similaire à la **figure 15**. Vous pouvez visualiser les tensions réelles du panneau solaire et de la batterie LiPo et l'évolution de ces tensions sur une journée. Au

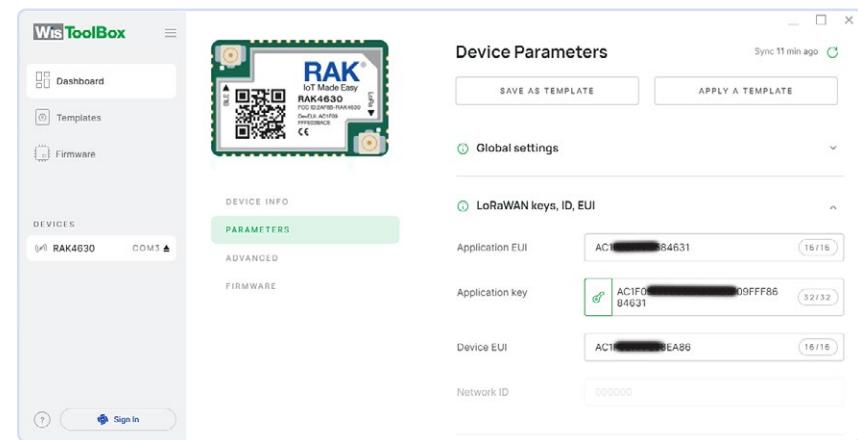


Figure 13. Tableau de bord WisToolBox - Clés LoRaWAN, ID, EUI.



Listing 1. Sortie de la console.

```

"settings": {
  "data_rate": {
    "lora": {
      "bandwidth": 125000,
      "spreading_factor": 7,
      "coding_rate": "4/5"
    }
  },
  "frequency": "867100000",
  "timestamp": 4014400019,
  "time": "2023-10-25T13:28:28.098315Z"
},
"received_at": "2023-10-25T13:28:28.116904692Z",
"consumed_airtime": "0.051456s",
"version_ids": {
  "brand_id": "rakwireless",
  "model_id": "wisblock-4631",
  "hardware_version": "1.0",
  "firmware_version": "1.2.0",
  "band_id": "EU_863_870"
},

```

cours de la journée, le panneau solaire fournit du courant pour charger la batterie LiPo. En début de soirée, la tension du panneau tombe en dessous de celle de la batterie LiPo. La batterie alimente alors le nœud et se décharge

naturellement. Le test s'est déroulé sur plusieurs semaines et a montré que le nœud IdO fonctionnait de manière fiable, même dans des conditions météorologiques défavorables en automne et en hiver. ↗

230668-04



Figure 14. Sortie de la console TTSS (CE).



Figure 15. Tableau de bord Datacake pour le nœud de capteur RAK.



À propos de l'auteur

Claus Kühnel a étudié les technologies de l'information à l'université technique de Düsseldorf. Depuis plusieurs années, il se consacre au développement de systèmes embarqués, notamment pour les équipements de diagnostic en laboratoire. C'est à travers ce secteur interdisciplinaire qu'il a découvert la communauté des « makers ». Au fil de sa carrière, il a rédigé de nombreux articles et publié plusieurs ouvrages consacrés aux matériels et logiciels de microcontrôleurs, tant en Allemagne qu'à l'international. Claus est particulièrement passionné par les innovations en matière de microcontrôleurs.



Produits

- C. Kühnel, *Develop and Operate Your LoRaWAN IoT Nodes* (Elektor, 2022) www.elektor.fr/20147
- LR1302 LoRaWAN HAT pour Raspberry Pi (EU868) www.elektor.fr/20744
- Carte de développement RA-08H LoRaWAN avec RP2040 intégré et écran LCD 1,8" (EU868) www.elektor.fr/20541

Questions ou commentaires ?

Envoyez un courriel à l'auteur (claus@ckuehnel.ch) ou contactez Elektor (redaction@elektor.fr).



SUJET À LA UNE

Visitez notre page **sans-fil & communication** pour des articles, des projets, des actualités et des vidéos. www.elektormagazine.fr/sans-fil-communication



LIENS

- [1] RAKwireless WisBlock system: <https://www.rakwireless.com/en-us/products/wisblock>
- [2] C. Kühnel, IoT-Projects for Makers (Elektor, 2023): <https://www.amazon.com/dp/B0C8VCF4DF>
- [3] RAK4631-R_Sensor-Test.ino sur GitHub : https://github.com/ckuehnel/WisBlock/tree/main/RAK4631-R_Sensor-Test
- [4] 18650 batteries at Akkuline: <https://www.akkuline.de/test/18650-lithium-ion-zelle-vergleich?projektLinId=11>
- [5] RAK4631-R_Sleep_Test_BLE.ino sur GitHub : https://github.com/ckuehnel/WisBlock/tree/main/RAK4631-R_Sleep_Test_BLE
- [6] Serial Bluetooth Terminal app: <https://www.kai-morich.de/android>
- [7] RAK4631-R_Sleep_Test_LoRaWAN.ino on GitHub: https://github.com/ckuehnel/WisBlock/tree/main/RAK4631-R_Sleep_Test_LoRaWAN
- [8] Datacake: <https://datacake.co/>
- [9] C. Kühnel, Develop and Operate Your LoRaWAN IoT Nodes (Elektor, 2022) : <http://www.elektor.fr/20147>
- [10] The Things Network: <https://www.thethingsnetwork.org/>
- [11] Airtime calculator: <https://avbentem.github.io/airtime-calculator/ttn/eu868>
- [12] Datacake online documentation: <https://docs.datacake.de/device/configuration>