

# optimisation et contrôle des appareils Thread à faible consommation d'énergie

faible consommation... peu d'effort ?



**Koen Vervloesem (Belgique)**

Le nouveau livre d'Elektor, intitulé *Building Wireless Sensor Networks with OpenThread*, ne se contente pas de présenter un ensemble riche d'informations sur la construction et la programmation de vos réseaux personnels pour lire différents capteurs. Il explore également des applications pratiques en utilisant le dongle nRF52840 de Nordic Semiconductor, qui est fourni avec le livre. Examinons un exemple pratique pour découvrir si naviguer dans le domaine des technologies à faible consommation, telles que Zephyr, CoAP, Wireshark, et Nordic est aussi simple.

**Note de la rédaction :** cet article est un extrait du livre d'Elektor *Building Wireless Sensor Networks with OpenThread* formaté et légèrement modifié pour correspondre aux normes éditoriales et à la mise en page du magazine Elektor. L'auteur et l'éditeur seront heureux de répondre aux questions – pour les contacter, voir l'encadré « **Questions ou commentaires ?** ».

Thread est souvent considéré comme un « réseau maillé sans fil à faible consommation », il est donc essentiel de disposer des données précises sur la consommation d'énergie des applications Thread ! Pour combler cette lacune, cet article vous propose de :

- mesurer la consommation énergétique du micrologiciel Thread de base sur un dongle nRF52840 de Nordic Semiconductor
- réduire la consommation d'énergie en désactivant le matériel non utilisé
- réduire la consommation énergétique en transformant l'application en un appareil en veille (*Sleepy End Device, SED*).

Pour commencer, chargez l'application de Thread (vous trouverez plus de détails dans le chapitre 5 du livre) sur un dongle nRF52840 (le code est disponible sur [4]). Assurez-vous ensuite que le dongle est bien connecté à votre réseau Thread, par exemple en surveillant les paquets réseau transmis dans Wireshark. Vous êtes maintenant prêt à étudier et à optimiser la consommation d'énergie de l'application sur le dongle.

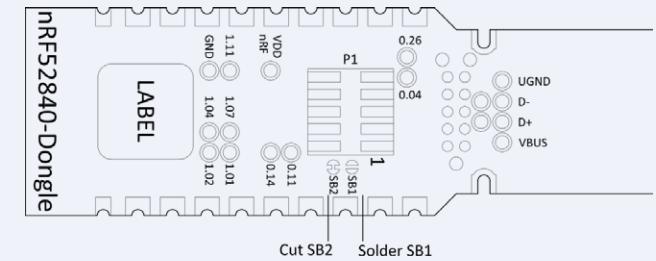


Figure 1. Couper SB2 et souder SB1 sur le dongle nRF52840 pour recevoir l'alimentation d'une source externe régulée de 1,8 à 3,6 V à travers VDD OUT. (Source de l'image : Nordic Semiconductor)

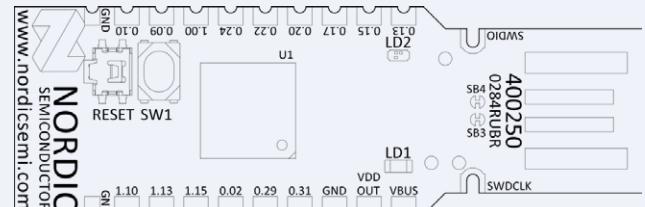


Figure 2. Brochage du Dongle nRF52840. (Source de l'image : Nordic Semiconductor)

## Mesurer la consommation d'énergie avec le kit Power Profiler

Afin d'optimiser votre application Thread pour réduire la consommation d'énergie, il est essentiel de pouvoir mesurer cette consommation. Le Power Profiler Kit II (PPK2) de Nordic Semiconductor est un kit matériel intéressant idéal pour cet objectif. Ce dispositif permet de mesurer et, potentiellement de fournir des courants allant de quelques  $\mu$ A à 1 A, avec des tensions inférieures à 5 V. L'appareil dont vous mesurez la consommation de courant (dans ce cas, le dongle nRF52840) est appelé l'appareil à tester (DUT). Côté logiciel, vous utiliserez l'application Power Profiler incluse dans nRF Connect for Desktop [1], le logiciel de développement multiplateforme de Nordic Semiconductor. Téléchargez le dernier package compatible avec votre système (Windows/Linux/macOS) et exécutez-le. Note : Sous Linux, assurez-vous de rendre d'abord le fichier AppImage téléchargé exécutables, par exemple, en utilisant la commande :

```
chmod +x nrfconnect-4.2.1-x86_64.appimage
```

Ouvrez ensuite nRF Connect for Desktop et cliquez sur *Install* à côté de Power Profiler.

Pour le matériel, configurez le Dongle nRF52840 pour pouvoir l'alimenter par une source externe via sa broche VDD OUT. Pour ce faire, vous devez couper la piste du circuit imprimé qui court-circuite le pont de soudure SB2, et vous devez souder SB1, comme expliqué dans [2]. SB1 et SB2 se trouvent sur la face inférieure de la carte, comme illustré dans la **figure 1**. Attention : après cette modification, votre dongle nRF52840 nécessitera une alimentation externe en permanence, même pour flasher une nouvelle image du micrologiciel via USB.

Le PPK2 offre deux modes de mesure de la consommation d'énergie :

- **Mode Ampere Meter** : l'appareil testé est alimenté par une source externe (par exemple, USB ou une batterie). Vous pouvez utiliser ce mode pour tester la consommation d'énergie du dongle lorsqu'il fonctionne sur une batterie de 3 V.
- **Mode Source Meter** : l'appareil testé est alimenté par le PPK2.

Dans les deux modes, vous devez connaître l'emplacement des broches VDD OUT et GND du dongle, voir la **figure 2**.

En mode Ampere Meter, connectez :

- PPK2 VIN à BAT+ de la batterie ;
- PPK2 VOUT à la sortie VDD OUT du dongle ;
- PPK2 GND à GND du dongle ;
- BAT de la batterie à GND du dongle.

Dans cette configuration, la batterie alimente le dongle nRF52840, mais avec le courant circulant à travers le Power Profiler Kit II, permettant ainsi sa mesure.

En mode Source Meter, connectez :

- PPK2 VOUT à la sortie VDD du dongle ;
- PPK2 GND à GND du dongle.

Les connexions sont illustrées dans la **figure 3**. Dans cette configuration, le Power Profiler Kit II alimente le dongle nRF52840 et mesure simultanément sa consommation en courant (**figure 4**).

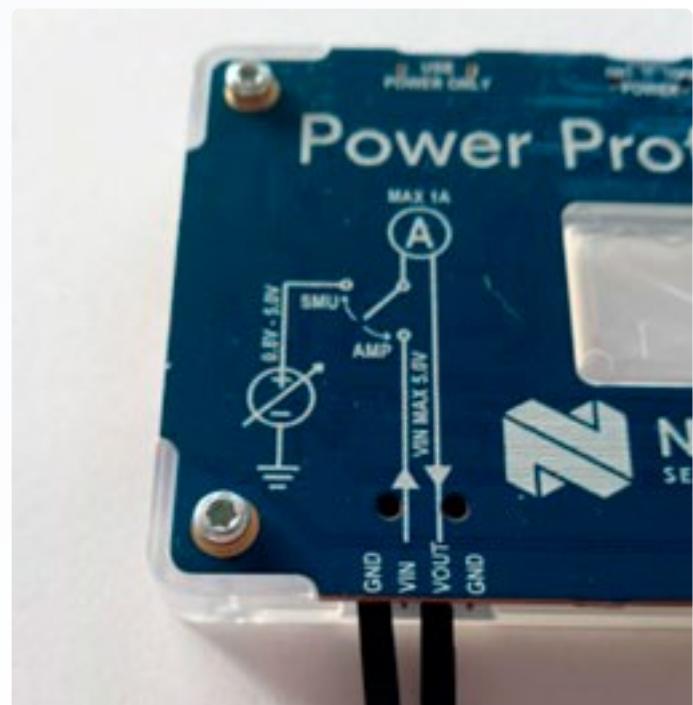


Figure 3. Connectez le Power Profiler Kit II au dongle nRF52840 en mode Source Meter.

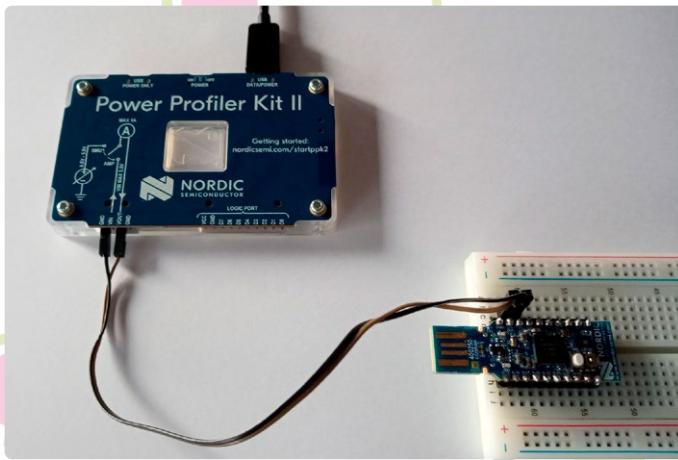


Figure 4. Le Power Profiler Kit II est prêt à mesurer la consommation de courant du Dongle nRF52840 en mode Source Meter.



Figure 5. Connectez le Power Profiler Kit II à votre PC.

## Consommation d'énergie de l'application Thread

Connectez le PPK2 à votre PC via un câble USB branché sur le port USB DATA/POWER, et mettez l'interrupteur POWER situé à côté du port USB sur ON, comme illustré sur la **figure 5**. Lancez l'application Power Profiler dans nRF Connect for Desktop et cliquez sur Select Device (sélectionner un appareil) en haut à gauche. Choisissez PPK2. L'application pourrait vous demander de programmer le dispositif. Confirmez en cliquant sur Program .

Choisissez la mode Source meter ou Amper meter, en fonction de la façon dont le PPK2 est connecté à votre dongle. Réglez la tension d'alimentation à 3300 mV et activez l'interrupteur Enable power output (même en mode Amper meter). Dès que cet interrupteur est activé, le dongle reçoit la tension et vous pouvez voir ses paquets sur le réseau Thread dans Wireshark.

Cliquez ensuite sur START dans Power Profiler. Comme illustré dans la **figure 6**, l'onglet Data Logger affichera en continu la consommation de courant de l'appareil.

D'après les données statistiques affichées sous la courbe, la consommation moyenne de courant est de 6,29 mA. Cette valeur élevée est attendue, étant donné que vous n'avez pas encore appliqué une

optimisation de la consommation énergétique. Si vous laissez le Power Profiler fonctionner pendant un moment avec Wireshark, vous observerez un pic plus élevé à chaque fois que l'appareil Thread reçoit une annonce d'établissement de lien Mesh à l'adresse **ff02::1** Link-Local All-Nodes. Si vous zoomez sur ce pic, vous verrez quelque chose qui ressemble à la **figure 7**.

## Réduire la consommation d'énergie

De nombreux éléments du dongle que vous n'utilisez pas sont activés par défaut. Mais chaque composant activé consomme de l'énergie ! Par conséquent, examinons attentivement toutes les définitions de l'arborescence des périphériques dans :

`~/zephyrproject/zephyr/boards/arm/nrf52840dongle_nrf52840/nrf52840dongle_nrf52840.dts`

Désactivez ceux dont vous n'avez pas besoin dans le fichier de superposition de l'arborescence du projet :

`boards/nrf52840dongle_nrf52840.overlay:`

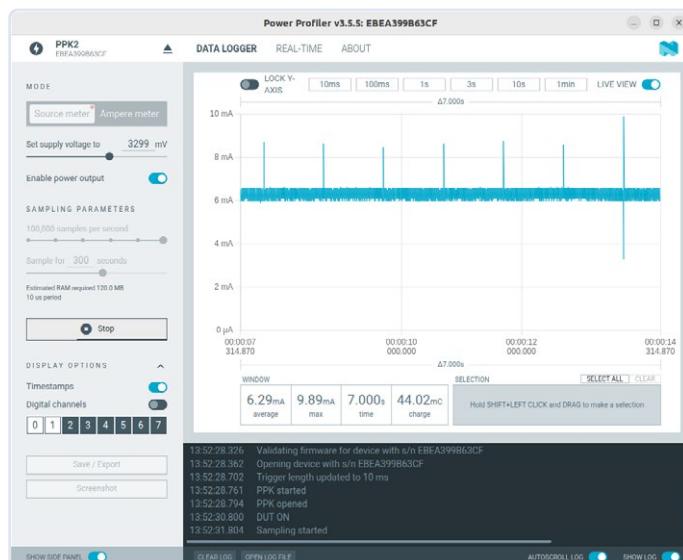


Figure 6. L'application Thread de base consomme en moyenne 6,29 mA.

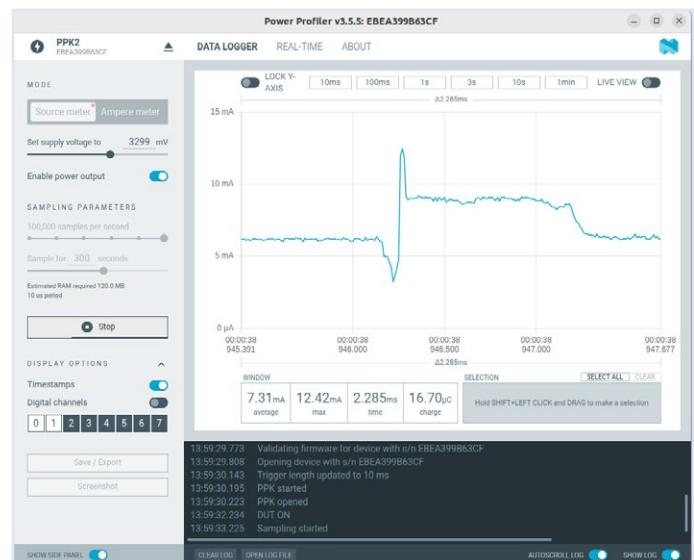


Figure 7. La réception d'une annonce MLE entraîne un pic élevé de consommation d'énergie.



## Listage 1. Device tree definitions.

```
/*
 * Copyright (c) 2024 Koen Vervloesem
<koen@vervloesem.eu>
 *
 * SPDX-License-Identifier: MIT
 */

/*
 * Disabled unused hardware
 */

&adc {
    status = "disabled";
};

&uart0 {
    status = "disabled";
};

&i2c0 {
    status = "disabled";
};

&i2c1 {
    status = "disabled";
};

&pwm0 {
    status = "disabled";
};

&spi0 {
    status = "disabled";
};

&spi1 {
    status = "disabled";
};

&usbd {
    status = "disabled";
};
```



## Listage 2. Kconfig configuration file (prj.conf).

```
#
# Copyright (c) 2024 Koen Vervloesem
#
# SPDX-License-Identifier: Apache-2.0
#
# Enable networking and OpenThread
CONFIG_NETWORKING=y
CONFIG_NET_IPV6_NBR_CACHE=n
CONFIG_NET_IPV6_MLD=n
CONFIG_NET_L2_OPENTHREAD=y
CONFIG_OPENTHREAD_THREAD_VERSION_1_3=y
CONFIG_OPENTHREAD_SLAAC=y
CONFIG_OPENTHREAD_PING_SENDER=y
CONFIG_OPENTHREAD_DNS_CLIENT=y
CONFIG_OPENTHREAD_MLR=y

# Kernel options
CONFIG_MAIN_STACK_SIZE=2560

# Enable power management
CONFIG_PM_DEVICE=y

# Create Sleepy End Device
CONFIG_OPENTHREAD_MTD=y
CONFIG_OPENTHREAD_MTD_SED=y
CONFIG_OPENTHREAD_POLL_PERIOD=1000

# Disable USB
CONFIG_USB_DEVICE_STACK=n
CONFIG_BOARD_SERIAL_BACKEND_CDC_ACM=n
```

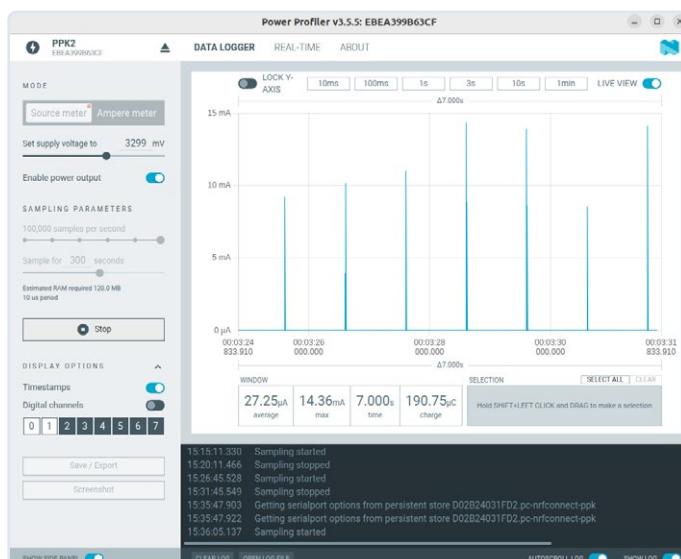


Figure 8. L'application Thread de base qui envoie des requêtes à son routeur en tant que Sleepy End Device toutes les secondes consomme en moyenne 27 µA.

Le programme permettant de le faire est présenté dans le **listage 1**. Note : L'interpréteur de commandes Zephyr ne sera plus accessible via USB si l'on désactive `uart0`.

De plus, le fichier de configuration Kconfig `prj.conf` ressemble au **listage 2**. Il permet essentiellement d'activer la gestion de l'alimentation et de désactiver l'USB. Il manque également les options pour activer la journalisation (logging) et l'interpréteur de commandes Zephyr. En outre, il configure l'appareil en tant que SED, de sorte que la radio peut rester en veille la plupart du temps, et il définit la période de polling (l'intervalle entre les requêtes du SED à son routeur) à 1000 ms.

Reconstruisez maintenant le micrologiciel :

```
$ west build -p auto -b nrf52840dongle_nrf52840
```

Créez le paquet et flashez le micrologiciel sur le dongle. Si vous mesurez la consommation de courant maintenant, la moyenne est considérablement inférieure à environ 27 µA (**figure 8**). Vous observerez clairement un pic de plus de 10 mA dans la consom-

mation de courant chaque seconde, et en même temps Wireshark montre une requête de données du SED à son routeur. Si vous envoyez une requête ping au SED, vous observerez que la réponse peut prendre jusqu'à une seconde, ce qui correspond à la période de polling de l'appareil (**figure 9**).

Plus la période de polling est longue, plus la consommation de courant est faible, mais moins l'appareil est réactif. Vous pouvez tester cela en réglant la variable de configuration `CONFIG_OPENTHREAD_POLL_PERIOD` à 5 000 millisecondes. Reconstruisez le micrologiciel et chargez-le sur l'appareil Thread. Si vous mesurez ensuite sa consommation de courant, vous trouverez une moyenne de 7,5  $\mu$ A, avec des pics de courant espacés de cinq secondes, comme illustré dans la **figure 10**. L'inconvénient de cette période prolongée est que la requête ping au SED peut prendre jusqu'à cinq secondes, ce qui le rend beaucoup moins réactif.

## Résumé et autres

Dans cet article, nous avons exploré la méthode de mesure de la consommation énergétique d'une application Thread fonctionnant sur un dongle nRF52840. Grâce au Power Profiler Kit II de Nordic Semiconductor et à l'application Power Profiler accompagné dans nRF Connect for Desktop, vous pouvez obtenir des informations détaillées sur la consommation d'énergie de votre application. Comme exemple, nous avons analysé la consommation d'énergie de l'application Thread de base présentée dans ce livre. Vous avez appris comment réduire la consommation d'énergie en désactivant le matériel non nécessaire et en configurant le périphérique comme un SED. Vous avez également appris l'impact de la période de polling sur la consommation d'énergie du périphérique : plus la période de polling est longue, plus la consommation de courant est faible, mais moins le périphérique est réactif.

Cet article n'est qu'un point de départ dans cette démarche d'optimisation. Si vous souhaitez réduire au maximum la consommation d'énergie de votre application Thread, vous devez tirer parti de l'API de gestion de l'énergie de Zephyr [3]. Cela vous permet d'utiliser les fonctions d'économie d'énergie du SoC et d'autres appareils, tels que les capteurs connectés. Vous pouvez ainsi programmer votre application pour qu'elle lise les mesures d'un capteur, envoie les données sur le réseau Thread, désactive le capteur, se met en veille, se réactive, réactive le capteur et recommence tout le cycle. 

240225-04

## Questions ou commentaires ?

Envoyez un courriel à l'auteur (koen@vervloesem.eu) ou contactez Elektor (redaction@elektor.fr).



## À propos de l'auteur

Koen Vervloesem écrit depuis plus de 20 ans sur Linux, les logiciels libres, la sécurité, la domotique, l'intelligence artificielle (IA), la programmation et l'Internet des objets (IoT). Il est titulaire d'un Master en ingénierie informatique, d'un Master en philosophie et d'un certificat de sécurité LPIC-3 303. Il donne des cours de Linux et de Python aux étudiants qui aspirent à un diplôme d'associé en IdO.

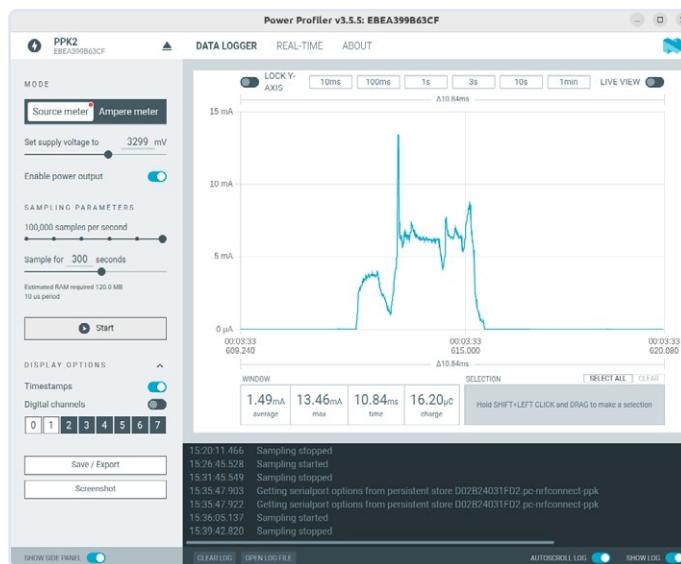


Figure 9. À chaque requête de données adressée à son routeur, l'appareil en veille subit un pic de consommation de courant.

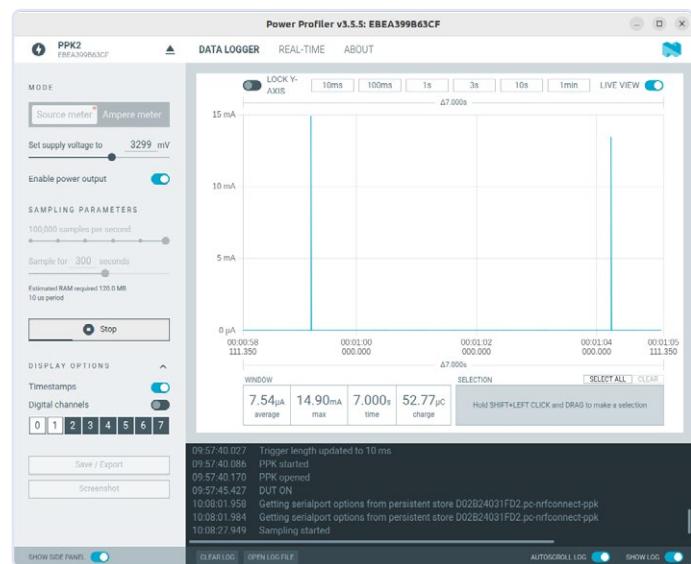
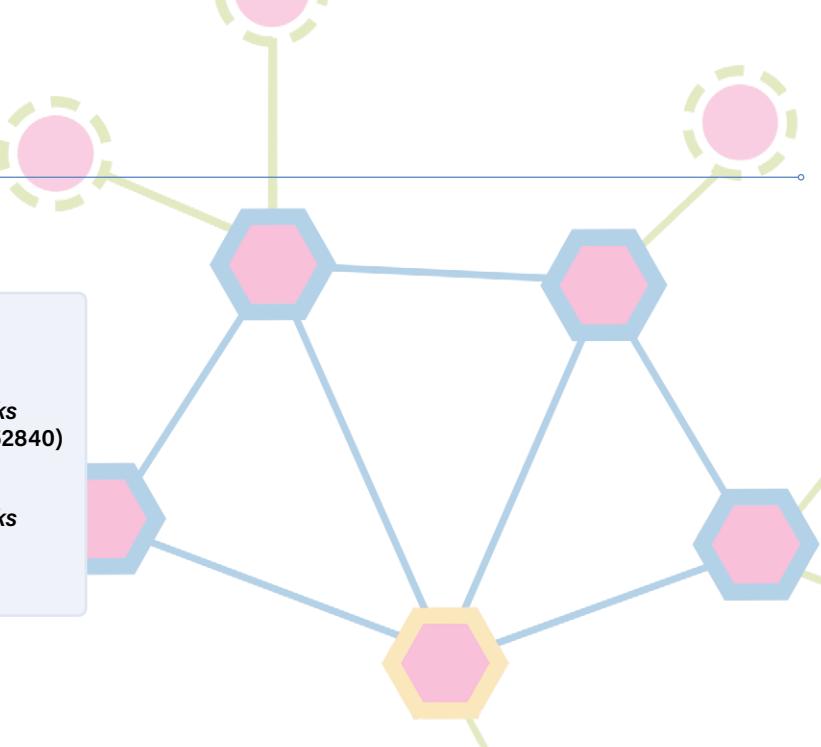


Figure 10. Avec une période de polling de 5 s, le SED ne consomme que 7,5  $\mu$ A en moyenne.



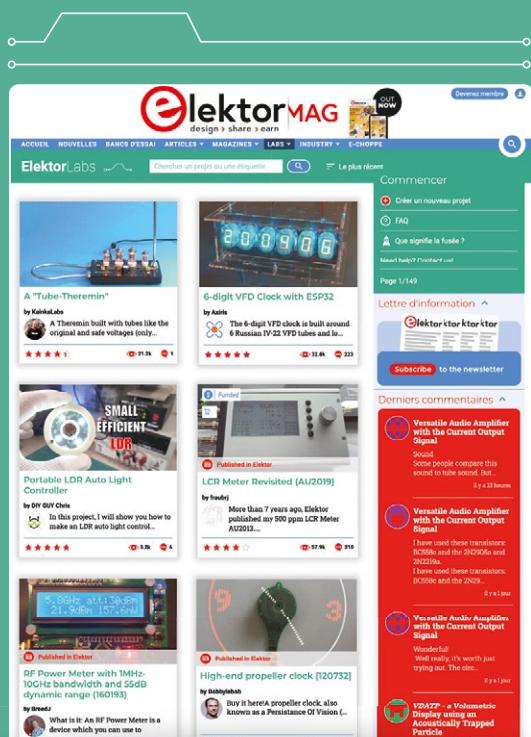
## Produits

- **Koen Vervloesem, Building Wireless Sensor Networks with OpenThread, Elektor 2024 (livre et dongle nRF52840)**  
[www.elektor.fr/20860](http://www.elektor.fr/20860)
- **Koen Vervloesem, Building Wireless Sensor Networks with OpenThread, Elektor 2024 (livre numérique)**  
[www.elektor.fr/20861](http://www.elektor.fr/20861)



## LIENS

- [1] nRF Connect for Desktop : <https://www.nordicsemi.com/Products/Development-tools/nRF-Connect-for-Desktop>
- [2] nRF52840 Dongle hardware mod : <https://tinyurl.com/rvj5mp47>
- [3] Zephyr's power-management API : <https://docs.zephyrproject.org/latest/services/pm/index.html>
- [4] Main.c, CMakeLists.txt, and configuration overlay for nRF52840 dongle :  
<https://github.com/koenvervloesem/openthread-applications>



**Partagez vos projets dès maintenant !**  
[www.elektormagazine.fr/e-labs](http://www.elektormagazine.fr/e-labs)

Stimulez vos innovations en  
électroniques avec  
**ElektorLabs**

- Partage gratuit de projets
- Soutien d'experts
- Opportunités de collaboration
- Accès à des ressources exclusives
- Publication dans la magazine Elektor



**elektor**  
design > share > earn