sourcemètre **BattLab-One**

Mesurer et optimiser la durée de vie des batteries des appareils IdO

Tam Hanna (Slovaquie)

Pour choisir une batterie appropriée incorporée dans un appareil autonome relié à l'Internet des objets (IdO), vous pouvez utiliser un multimètre numérique pour mesurer le courant actif et le courant de veille. Vous pourrez ainsi estimer l'autonomie de l'appareil avec une charge, optimiser la conception en modifiant le rapport des modes veille/activité, voire même opter pour une capacité ou un type de batteries différents. Raccordé via un port USB, le BattLab One peut faire tout cela d'un simple clic de souris. Il alimente l'appareil testé et affiche toutes les données relatives à l'autonomie prévue de la batterie…

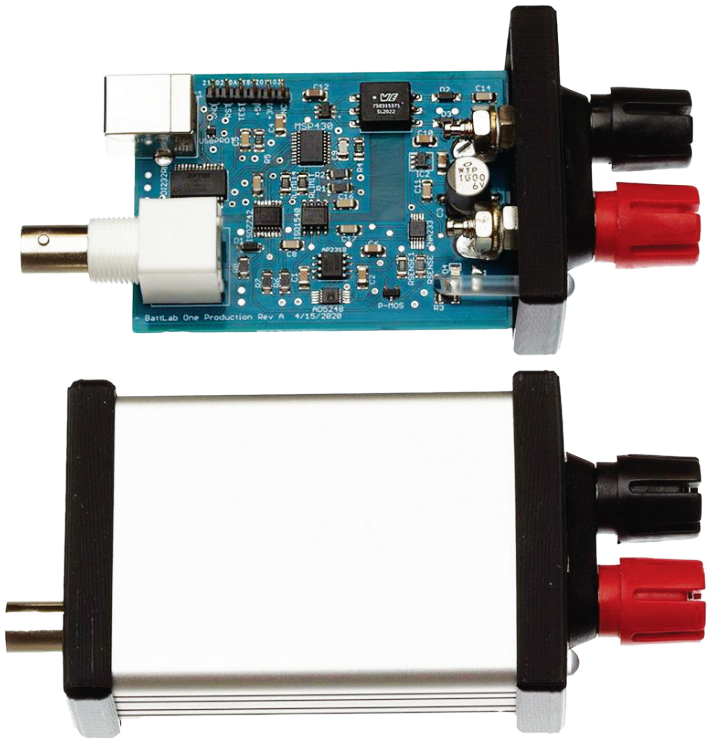

Figure 1. Le BattLab-One et son boîtier en aluminium.

Obtenir une mesure fiable de l'énergie consommée par un appareil doté d'un microcontrôleur n'est pas anodin. Comme pour toute puce informatique, la consommation d'énergie dépend surtout de la charge de traitement instantanée et de l'état opérationnel. Il ne suffit pas de prendre en compte le fonctionnement normal et les états de veille ou de sommeil profond, dans lesquels la consommation de courant des microcontrôleurs les plus diffusés peut varier de moins de 1 µA à plusieurs centaines de milliampères (mA). Certains périphériques du dispositif peuvent également n'être actifs qu'à certains moments, de sorte que la consommation de courant ne sera pas constante. Un outil spécialisé comme BattLab-One est très utile à cet égard et peut tenir compte de ce comportement dynamique pour faire une estimation raisonnablement réaliste de la durée de fonctionnement de l'appareil lorsqu'il est alimenté par différents types de batteries.

Qu'est-ce que BattLab-One ?

Dans le domaine de la mesure de la consommation électrique, un sourcemètre (SMU) est un outil très utile dans votre labo. Ce type d'alimentation apporte du courant au dispositif testé et effectue des mesures précises de la puissance consommée sur des périodes définies. Parmi des exemples de sourcemètres professionnels figurent les appareils fabriqués par Keithley. Le BattLab-One de Bluebird Labs remplit une fonction similaire, mais pour un prix inférieur à 100 €. L'instrument (y compris le boîtier métallique) est disponible dans la boutique Elektor [1].

Reportez-vous à la **figure 1** pour voir la prise USB de type B sur la gauche ainsi qu'une prise BNC acceptant un signal d'entrée de déclenchement. Sur la droite se trouvent les deux prises de 4 mm servant à alimenter l'appareil testé. La **figure 2** montre les garnitures en plastique à dévisser pour fixer les pinces crocodiles aux bornes. Un examen plus attentif des coiffes d'extrémité du boîtier montre une texture de surface adaptée à l'impression 3D.

BattLab-One est alimenté par le port USB d'un ordinateur et peut délivrer une gamme de tensions d'alimentation comprises entre 1,2 à 4,5 V et des courants atteignant 450 mA maximum. Les profils de décharge des types de batteries fréquemment utilisés (comme les technologies Li-Ion, $LiFePO_4$ alcaline, NiMh, NiCd) sont inclus dans le logiciel et utilisés pour calculer les prévisions de durée de vie. BattLab-One ne tient pas compte de certaines subtilités comme

les chutes de tension liées à la charge et l'âge de la batterie. Selon le fabricant, l'appareil de mesure offre une simulation de base de la batterie et une plage de mesure de courant comprise entre 10 µA et 500 mA. Un CA/N de 16 bits avec échantillonnage à 1 kHz est incorporé pour mesurer le courant lié au dispositif testé.

BattLab-One et Linux

Le logiciel de l'instrument a été développé par Bluebird Labs en Python. Pour ma première tentative, j'ai voulu faire fonctionner le système directement sous Ubuntu 20.04 LTS. Le code source du logiciel est disponible sur GitHub [2] et peut être compilé à l'aide des méthodes habituelles. Cependant, Bluebird Labs a adopté une approche différente en utilisant *pipenv*. Tout utilisateur qui, comme moi, ne dispose que de la variante habituelle de *venv* doit installer *pipenv*. Pour moi, le processus s'est déroulé comme suit :

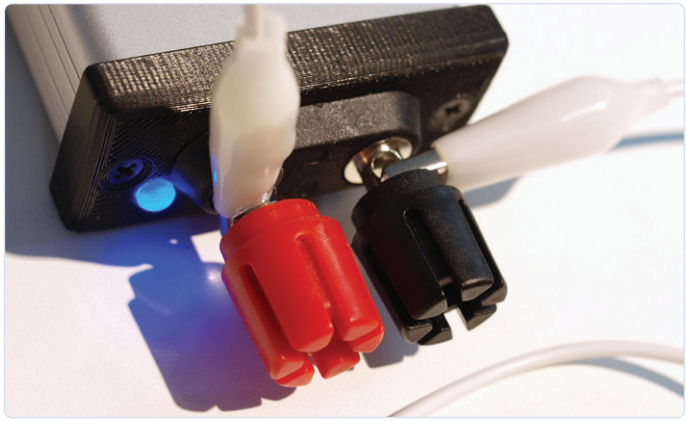

Figure 2. Branchement électrique des pinces crocodiles.

```
tamhan@TAMHAN18:~/BattLabonespace/
BattLab-One$ sudo apt install pipenv
```

Les premiers problèmes sont apparus lors de la configuration de l'environnement virtuel, car elle suppose la version 3.7. de Python. Si, comme moi, vous voulez utiliser une version plus récente (c'est-à-dire la version 3.8 préinstallée sous Ubuntu 20.04 LTS), il faudra ajouter la déclaration suivante :

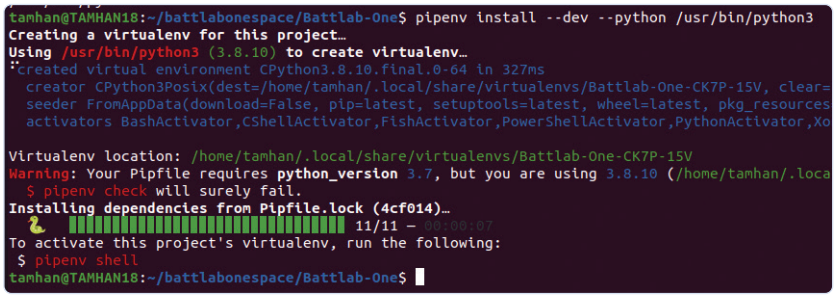

Figure 3. Message d'erreur PipEnv.

```
tamhan@TAMHAN18:~/BattLabonespace/
BattLab-One$ which python3
/usr/bin/python3
tamhan@TAMHAN18:~/BattLabonespace/BattLab-One$ pipenv
install --dev --python /usr/bin/python3
```

Considérons maintenant le processus de création d'un *virtualenv* pour ce projet. Le logiciel émet des avertissements (**fig. 3**) lors de la saisie des commandes. L'environnement virtuel est donc activé selon le schéma suivant. Il restera actif comme un environnement *venv* normal jusqu'à ce que la fenêtre du terminal soit fermée :

```
tamhan@TAMHAN18:~/BattLabonespace/BattLab-One$ pipenv
shell
. . .
```

L'exécution des trois fichiers *.py* avec le logiciel est maintenant possible en utilisant la méthode suivante :

```
(BattLab-One-CK7P-15V) tamhan@TAMHAN18:~/
BattLabonespace/BattLab-One$ python3
BattLab-Release_V1.2.1.py
Traceback (most recent call last):
  File "BattLab-Release_V1.2.1.py", line 59, in
<module>
    import pkg_resources.py2_warn
ModuleNotFoundError: No module named 'pkg_resources.
py2_warn'
```

Malheureusement, Bluebird Labs ne se conforme pas à la spécification des outils d'installation, mais inclut explicitement une bibliothèque qui n'est disponible que dans certaines versions de ces outils. Le fichier *.py* doit être modifié pour supprimer l'instruction suivante :

```
import pkg_resources.py2_warn
```

Le logiciel a aussi un problème concernant les chemins d'accès locaux :

```
(BattLab-One-CK7P-15V) tamhan@TAMHAN18:~/
BattLabonespace/BattLab-One$ python
BattLab-Release_V1.2.1.py
. . .
_tkinter.TclError: error reading bitmap file "icons\
bbirdlogo.xbm"
```

La solution consiste à ouvrir l'interface graphique de TK-Inter en supprimant la ligne :

```
root.iconbitmap(bitmap=GetIconPath())
```

pour supprimer l'icône du programme. Par la même occasion, vous pouvez adapter le chemin d'accès au fichier de la manière suivante :

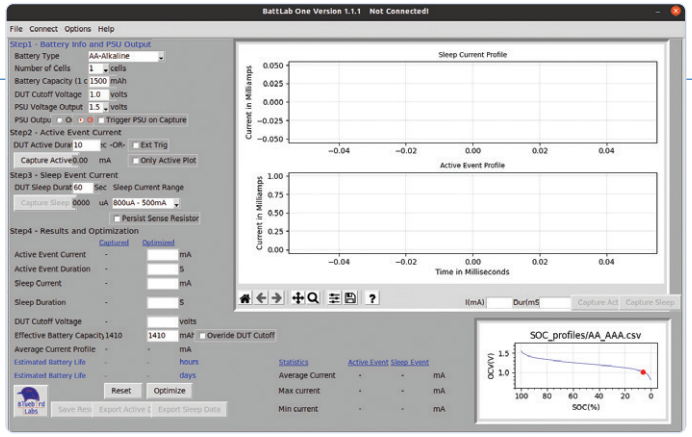

Figure 4. L'interface graphique Linux BattLab.

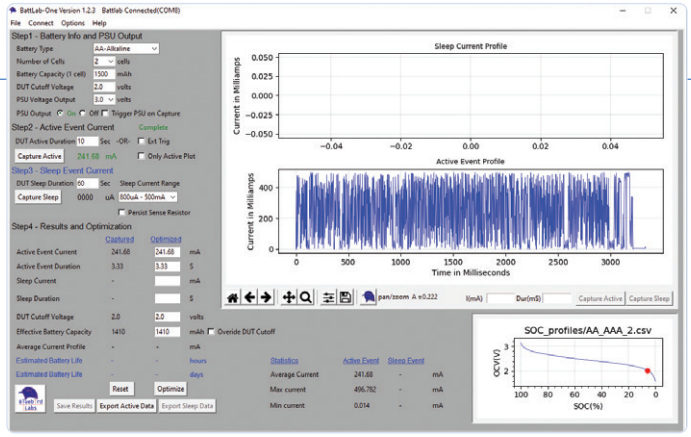

Figure 5. BattLab-One enregistre et affiche la consommation de courant du dispositif testé.

```
img = PhotoImage(file='/home/tamhan/BattLabonespace/
BattLab-One/icons/bbirdlogo_png1.png')
```

La tentative suivante d'exécution montre que l'environnement TK-Inter recherche le fichier *bblogo.gif* dans le mauvais dossier.

```
_tkinter.TclError: couldn't open "/home/tamhan/.
local/share/virtualenvs/BattLab-One-CK7P-15V/lib/
python3.8/site-packages/matplotlib/mpl-data/images/
bblogo.gif": no such file or directory
```

Vous pouvez donc copier n'importe quel fichier GIF dans ce répertoire et lui donner le nom approprié. Si vous mettez la ligne suivante en commentaire, vous pouvez démarrer le programme avec une gamme de fonctions légèrement limitée (**fig. 4**) :

```
#toolbar.children['!button8'].
config(command=select_range)
```

La désactivation du bouton de la plage de mesure s'avère peu critique dans la pratique, car le diagramme MatPlotLib est de toute façon problématique en termes de convivialité. De nombreuses étiquettes apparaissent tronquées et la taille de la fenêtre ne peut être ajustée. Comme nous allons le voir, le programme a moins de problèmes à fonctionner sous Windows 10. Avant de changer de système d'exploitation, nous pouvons faire le ménage pour supprimer l'environnement virtuel créé et ainsi libérer de l'espace de stockage de masse :

```
(BattLab-One-CK7P-15V) tamhan@TAMHAN18:~/
BattLabonespace/BattLab-One$ exit
exit
tamhan@TAMHAN18:~/BattLabonespace/BattLab-
One$ cd /home/tamhan/.local/share/virtualenvs/
BattLab-One-CK7P-15V
tamhan@TAMHAN18:~/.local/share/virtualenvs/BattLab-
One-CK7P-15V$ rm * -rf
tamhan@TAMHAN18:~/.local/share/virtualenvs/BattLab-
One-CK7P-15V$ ls
```

BattLab-One fonctionne sous Windows 10

La situation est plus simple dans la mesure où des fichiers *.exe* prêts à l'emploi sont fournis sous [3]. L'archive ZIP comprend, entre autres, le sous-dossier *SOC_profiles*, qui contient les différents profils de batterie sous forme de tableau. Vous devez dézipper l'archive.

Lorsque le fichier *BattLab-Release_V1.2.3.exe* est exécuté, l'affichage de l'interface utilisateur se présente sous la forme de la version Linux de la **figure 4**. La version Windows se comporte en grande partie de la même manière, mais présente moins d'erreurs de rendu. Tout d'abord, nous pouvons sélectionner le type de batterie dans la zone supérieure gauche. Les données de batterie associées servent à créer la courbe de décharge caractéristique présentée en bas à droite. À droite du champ *PSU Output*, les boutons *On* et *Off* permettent de commuter le convertisseur de tension intégré pour alimenter l'appareil testé connecté aux deux prises de sortie.

Je travaille depuis quelque temps sur mon propre projet de capteur IdO appelé *HygroSage*. Avec son écran couleur et son processeur puissant, il serait un candidat idéal pour tester le BattLab-One. J'ai connecté le capteur. En cliquant sur l'option *Capture active*, le logiciel affiche une barre de progression verte clignotante intitulée *Active Event Current*, puis ne fait rien de visible pendant 10 s. Pendant ce temps, le système mesure les informations et finit par les afficher comme le montre la **figure 5**.

Le curseur permet d'afficher la valeur de n'importe quel point de la forme d'onde. En pratique, le fonctionnement n'est pas aussi convaincant car la valeur indiquée dans la zone de texte ci-dessous ne montre pas la valeur mesurée, mais la position du pointeur de la souris sur le graphique. Vous devez donc déplacer le pointeur de la souris vers un point de la courbe. En outre, il n'y a pas de moyen vraiment pratique de zoomer dans le diagramme (produit par MatPlotLib). Malgré la résolution « full HD » de mon écran, je n'ai pas pu agrandir la fenêtre de l'interface utilisateur. Il est cependant possible de sélectionner une zone de la forme d'onde en utilisant l'option *Zoom to Rectangle*.

Pour effectuer un test, j'ai connecté un multimètre étalonné de haute qualité en série avec l'appareil testé. J'ai ainsi pu mesurer un courant d'environ 32 mA. Mais BattLab-One ne partageait pas cet avis. Il semble que le régulateur à découpage de type MCP1640 du BattLab-One produit un bruit qui interfère réellement avec son fonctionnement. Comme deuxième test, j'ai utilisé une résistance de 1 kΩ comme charge et spécifié une tension de sortie de 3 V. Le résultat est la trace représentant le bruit de quantification des mesures dans la

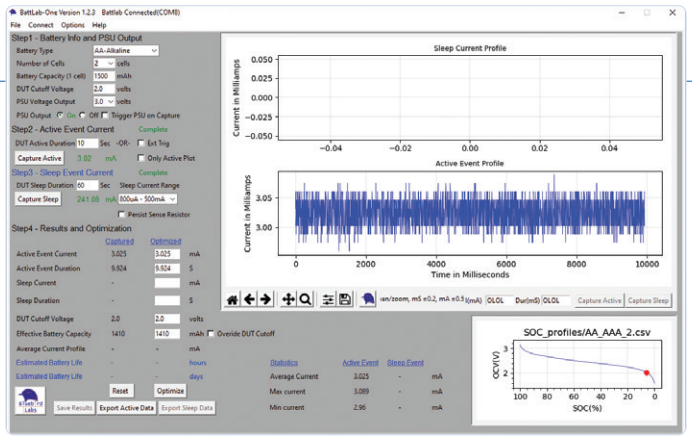

Figure 6. Trace montrant le courant circulant dans une résistance de 1 kΩ sous 3 V avec un certain niveau de bruit apparent.

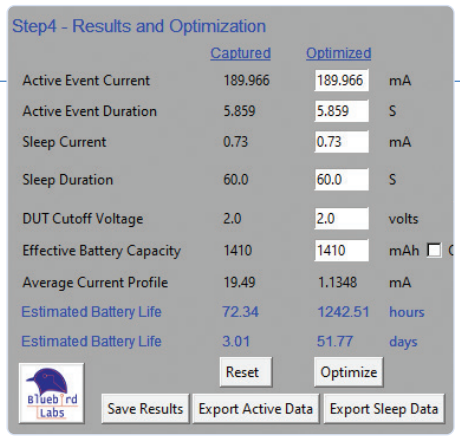

Figure 7. L'autonomie de la batterie peut être optimisée en utilisant les valeurs de courant en mode veille et en mode actif et la capacité de la batterie.

figure 6. La connexion d'un condensateur électrolytique de 100 µF en parallèle n'a pas entraîné d'amélioration significative de la forme d'onde de consommation de courant quantifiée.

Courant de veille et optimisation

Une fois la consommation de courant actif obtenue à l'étape 2, nous pouvons passer à l'étape 3 pour mesurer le courant de veille. Ici, nous devons nous assurer que le dispositif testé sera en mode veille au moment où la mesure sera effectuée. Nous allons entrer les paramètres comme nous l'avons fait à l'étape 2 pour la capture du mode actif. Sauf si le projet conçu ne fonctionne pas correctement, l'appareil consommera toujours beaucoup moins d'énergie en mode veille. BattLab-One vous donne la possibilité de commuter la plage de courant en mode veille entre 10 uA et 800 µA ou entre 800 µA et 500 mA pour une résolution supérieure de la mesure. Pour information, l'appareil ne doit pas consommer trop de courant dans la plage de mesure inférieure. Selon la documentation, un courant de charge dépassant 250 mA peut endommager le matériel. Lorsque le dispositif testé est en mode veille, nous pouvons appuyer sur *Capture* pour enregistrer le courant.

Les résultats obtenus après un test avec *Hygrosage* sont présentés dans la **figure 7**. À l'étape 4, le système affiche les valeurs calculées. À l'aide de ces données, nous pouvons effectuer des expérimentations avec les caractéristiques et la capacité de la batterie, ainsi qu'avec les temps d'activité et de veille. Il suffit ensuite de cliquer sur *Optimize* pour voir facilement à quel degré d'autonomie nous pouvons nous attendre pour le dispositif lorsqu'il sera déployé sur le terrain.

La prédiction de l'autonomie de la batterie sera raisonnablement précise pour les appareils qui fonctionnent avec une période d'activité et de sommeil définie. Certains appareils dépendent d'une interruption d'événement externe pour sortir du mode veille ; dans ce cas, nous ne pouvons calculer l'autonomie de la batterie qu'en utilisant une estimation de la fréquence à laquelle les interruptions peuvent se produire.

En résumé

Le matériel du BattLab-One est tout à fait utilisable. Il est cependant possible d'améliorer le logiciel. Puisqu'il est basé sur des éléments *open source*, on peut s'attendre à ce que son évolution bénéficie du soutien de la communauté. Cela vous donne également la possibilité d'explorer les rouages de BattLab, et même d'adapter et d'améliorer le logiciel si nécessaire pour qu'il réponde à vos propres besoins. En ce qui concerne le matériel, BattLab-One fonctionne bien pour l'essentiel, malgré quelques inconvénients mineurs. Son faible prix et ses performances élevées en font un bon complément au banc d'essais pour ceux qui développent des dispositifs IdO.

210473-04

Des questions, des commentaires ?

Envoyez un courriel à l'auteur (tamhan@tamoggemon.com) ou contactez Elektor (redaction@elektor.fr).

Contributeurs

Texte et illustrations : **Tam Hanna**
Rédaction : **Thomas Scherer, C. J. Abate**
Mise en page : **Giel Dols**
Traduction : **Pascal Godart**

LIENS

[1] BattLab-One : http://www.elektor.fr/battlab-one-battery-life-optimizer-with-enclosure
[2] Logiciel sur Github : http://github.com/petersdw1/BattLab-One.git
[3] Logiciel pour Windows 10 : https://github.com/petersdw1/BattLab-One/releases/tag/V1.2.3