

ESP32 avec OpenDTU pour les centrales solaires sur balcon

Relever les données des petits onduleurs avec des microcontrôleurs

Source : Midjourney (edited)

Thomas Scherer (Elektor)

Les micro-onduleurs fiables de Hoymiles ne disposent pas d'une connectivité Wifi intégrée. Pour lire la puissance actuelle, la production d'énergie quotidienne et cumulée, ainsi que d'autres données, une unité de transfert de données (DTU) supplémentaire est nécessaire. Au lieu de dépenser trop d'argent, vous pouvez fabriquer un DTU abordable et l'utiliser grâce à un logiciel open-source.

Il y a plus de deux ans, j'avais construit une petite centrale sur balcon d'une puissance de 600 W [1]. J'ai maintenant un nouveau système (avec des panneaux plus puissants et un nouvel onduleur). Je lis les données avec un petit circuit, qui me fournit une page web informative sur mon propre réseau local (voir **figure 1**). Malheureusement, les bons micro-onduleurs de Hoymiles, très répandus, ne sont accessibles que par une liaison radio Nordic spéciale avec un protocole propriétaire. Mais l'achat d'un DTU prêt à l'emploi auprès du fabricant ne va pas avec mon sens de création, car il est possible de construire soi-même un tel appareil, ce qui permet d'économiser de l'argent et d'empêcher le transfert de données vers des nuages peu fiables.

Open Source

Grâce à l'utilisation répandue des onduleurs Hoymiles, quelques fabricants inventifs se sont réunis et ont fait un peu de réingénierie pour découvrir quelles données sont transmises par la radio Nordic et comment. Vous pouvez lire la discussion originale (en allemand) sur le forum Mikrocontroller.net [2]. Le logiciel a également été développé pour les célèbres cartes à microcontrôleurs ESP32 d'Espressif (plus un module radio Nordic). Pourquoi ce microcontrôleur ? Le choix est évident : un ESP32 est équipé d'un processeur à double cœur avec une fréquence d'horloge de 240 MHz et est livré avec le Wifi intégré ainsi que le Bluetooth. Tout cela nécessite moins d'un demi-watt en moyenne.

Le logiciel développé à cet effet lit non seulement toutes les données pertinentes des onduleurs Hoymiles et des clones utilisant la même conception, mais il peut également modifier la puissance injectée sur le réseau. Cette mode est appelée "zero feed-in". L'onduleur est configuré pour fournir autant d'énergie qu'il en consomme. Il est possible d'utiliser d'autres types d'onduleurs que ceux de la série HM, éprouvée, mais ils nécessitent un module radio différent, que nous n'aborderons pas ici. Rapidement, deux solutions logicielles très pratiques ont vu le jour, OpenDTU [3] et AhoyDTU [4], qui diffèrent sur certains points. Sur les sites web de ces projets, vous pouvez déterminer à quels onduleurs spécifiques ils sont adaptés et leurs fonctionnalités. J'ai essayé les deux solutions. Il est difficile de déterminer laquelle est la "meilleure". Tout d'abord, les onduleurs HM-300 à

HM-1500 peuvent utiliser les deux versions dans tous les cas.

Si vous souhaitez utiliser des cartes ESP8266 plus au moins chères et moins puissantes, vous devez opter pour AhoyDTU. Les deux solutions fonctionnent sur la plateforme ESP32, plus puissante. Dans ce qui suit, nous décrivons le fonctionnement avec OpenDTU.

Matériel

Comme nous l'avons déjà mentionné, OpenDTU nécessite une carte d'interface ESP32 (BoB) comme base. La **figure 2** montre trois cartes ESP32 couramment utilisées. En plus, un module radio Nordic est nécessaire. Il existe des modèles avec une antenne imprimée sur la carte ou avec un connecteur SMA pour une antenne externe (**figure 3**). Vous pouvez certainement deviner lequel des deux a la plus grande portée. Mais pour moi, même la version avec l'antenne imprimée (à seulement 25 % de la puissance d'émission maximale) était suffisante pour établir une liaison radio acceptable, même sur une distance de 11 m à travers un mur de briques d'argile de 34 cm d'épaisseur. En fait, c'est tout. Pour OpenDTU, vous pouvez combiner presque toutes les versions d'ESP32 avec l'un ou l'autre des modules radio Nordic. Vous pouvez même contrôler de petits écrans, mais nous y reviendrons plus tard. J'ai conçu des circuits imprimés pour les trois variantes d'ESP32 illustrées dans la figure 2. Avec la version mini ci-dessous, vous pouvez créer un DTU relativement petit. Bien entendu, vous pouvez également connecter manuellement le BoB ESP32 et le module Nordic sans circuit imprimé.

Voilà pour la théorie. Mais le problème réside dans les détails. Outre les modules radio "nRF24L01+", il en existe également sans le "+". Attention : la version " + " est obligatoire ! Alors que les cartes plus simples avec une antenne imprimée sont disponibles à des prix compris entre 1 € (Chine) et 4 € (UE), la version plus luxueuse du fabricant eByte (**figure 3**, à droite) est livrée avec une unité radio et coûte entre 4,50 € (Chine) et 7 € (UE), y compris l'antenne. Ainsi, un DTU D/Y avec un module Nordic disposant d'une antenne externe n'est pas beaucoup plus cher qu'un autre sans. J'ai essayé les versions avec et sans antenne externe et n'ai pas trouvé de différences significatives dans mon cas, mais avoir une portée étendue peut s'avérer utile. D'ailleurs, une antenne Wifi externe normale convient, puisque les modules Nordic émettent également à 2,4 GHz. La seule chose importante est qu'elle doit avoir un connecteur à vis SMA mâle.

Selon le modèle, il faut débourser entre 4 € (Chine) et 10 à 15 € (UE) pour une carte ESP32. Avec une ESP32 et une carte BoB nRF24, vous pouvez construire un DTU entièrement fonctionnel comprenant un simple boîtier en plastique et un câble Micro-USB pour moins de 12 € (Chine) ou 20 € (UE). C'est une véritable aubaine par rapport aux 100 à 250 € que coûte un DTU original de Hoymiles !

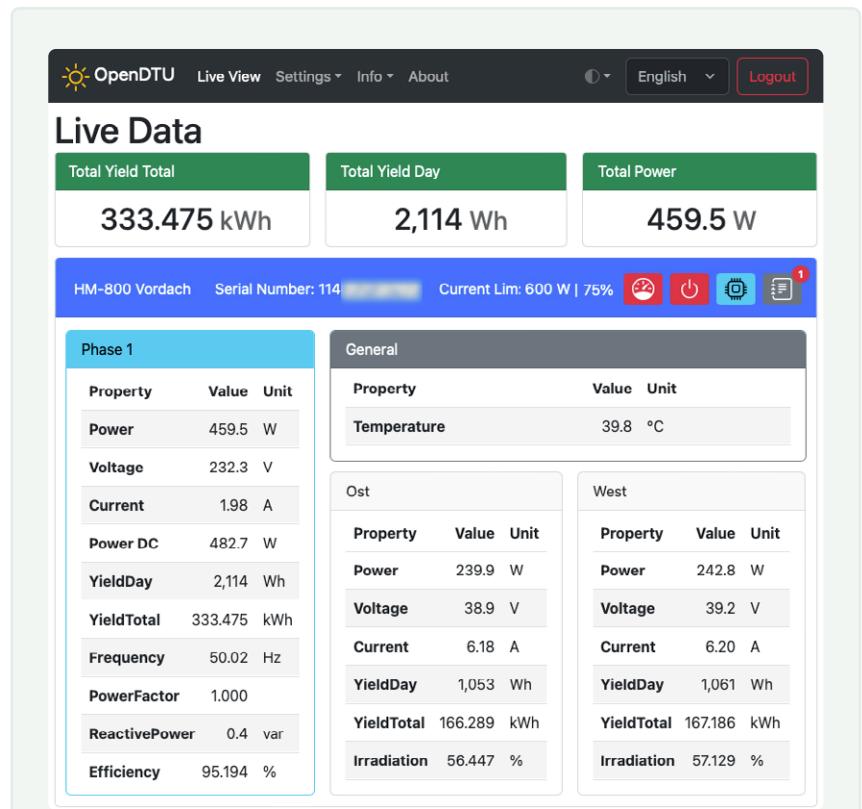


Figure 1. Un court test de puissance le 31 juillet 2023, sous une légère couverture nuageuse. L'onduleur de 800 W était limité à 600 W et, néanmoins, 4,7 kWh ont été accumulés à la fin de la journée.

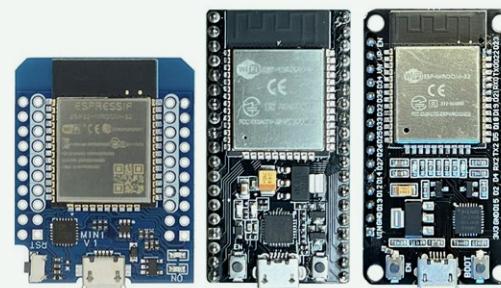


Figure 2. Comparaison de trois cartes de différentes dimensions : à droite, un BoB ESP32 à 30 broches, au centre un avec 38 broches, à gauche, une version ESP32 plus petite (D1 mini), où deux des 40 broches ne sont pas utilisées.



Figure 3. Deux versions de cartes avec la puce radio Nordic nRF24L01+, à gauche avec une antenne sur la carte et à droite, avec un connecteur SMA pour une antenne externe de 2,4 GHz.

Figure 4. Deux BoB ESP32 à 38 broches. En haut avec l'antenne imprimée qui déborde et en bas avec un connecteur pour antenne Wifi externe. Les deux fonctionnent bien.

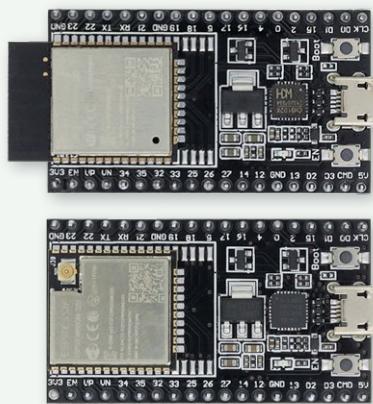


Figure 5. Schéma de circuit/câblage servant de guide pour la connexion des BoB assemblés sur une plaque d'essai avec des fils toronnés colorés. Ici, on présente le câblage de la version à 30 broches.

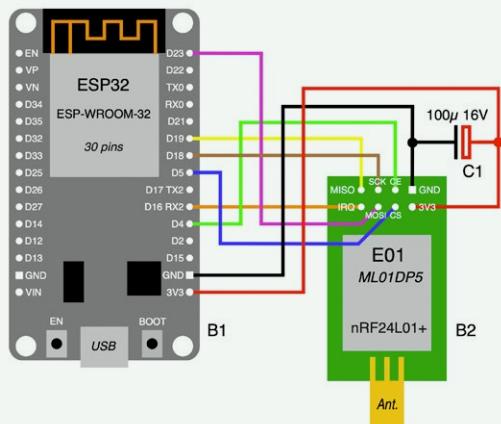


Figure 6. Schéma de câblage pour les BoB ESP32 à 38 broches.

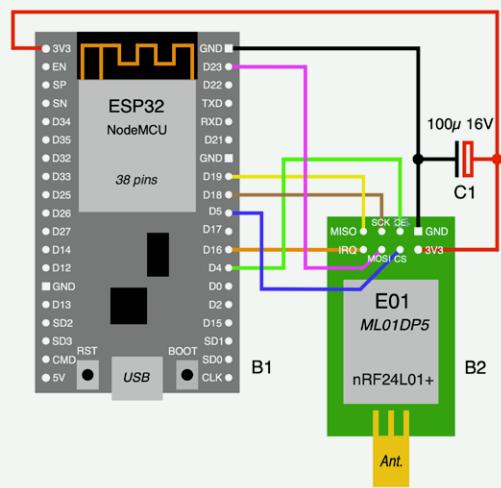
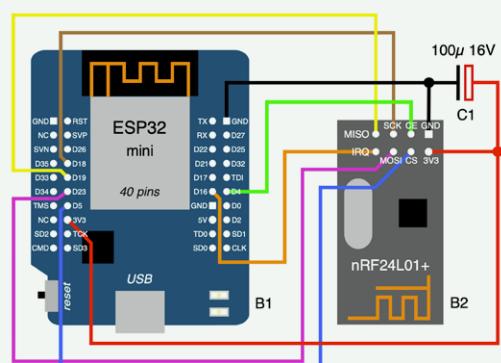


Figure 7. Schéma de câblage pour le petit BoB ESP32 mini avec un petit module radio (antenne imprimée). Vous pouvez également équiper un module d'une antenne externe.



La principale difficulté est qu'il existe de nombreuses versions des BoB ESP. Même sur le site d'Espressif [5], on a l'embarras du choix. Et puis il y a plusieurs fabricants d'Extrême Est qui intègrent différents Soc ESP32 sur différentes cartes avec différents composants, différentes antennes et, surtout, différents brochages. Tout d'abord, la plupart de ces modèles fonctionneront toujours sur le DTU décrit ici. Cependant, je ne recommanderai pas les versions allégées avec moins de mémoire flash, ou les versions à simple cœur.

La **figure 2** montre trois variantes de BoB ESP32 avec 30, 38 et 40 broches, qui correspondent probablement à plus de 90 % du marché. Visuellement et technique-ment, il existe d'autres différences non pertinentes, qui concernent par exemple l'antenne ou la puce de conver-tisseur USB. La plupart des modèles bon marché sont dotés d'une antenne Wifi imprimée et d'une prise Micro-USB. Certains disposent de cette antenne qui déborde un peu de la carte et d'autres ont une petite prise pour connecter une antenne Wifi externe à la carte (**figure 4**). Il existe également des versions dotées d'une prise USB-C plus robuste au lieu d'une prise Micro-USB. L'USB-C est plus adapté aux branchements fréquents.

Les cartes ESP32 sont suffisamment puissantes pour piloter un petit écran et lui transmettre des données. Vous pouvez choisir parmi plusieurs afficheurs. L'idéal est d'avoir des écrans OLED de 128x64 pixels, contrô-lés via I²C. Deux versions très courantes utilisent les contrôleurs SH1106 (1.3 pouce de diagonal) ou SSD1306 (0.96 pouce de diagonal). En plus, il y a des plus grands avec le contrôleur SSD1309 pour les diagonales de 1.54 et 2.42 pouces qui fonctionneront aussi. Grâce à l'I²C, nous n'avons besoin que de deux ports d'entrée/sortie supplémentaires pour l'ESP32, en plus des broches +5 V et de la masse. Les écrans sont également abordables, avec 2 € à 10 € (Chine) et 4 € à 20 € (UE). Je les ai tous testés et j'ai constaté ce qui suit : sur la version 2.42 pouces avec un connecteur I²C à quatre broches, D2 doit être démonté de l'arrière et remplacé par un cavalier pour qu'il génère le signal ACK comme prévu. La consommation additionnelle d'un écran OLED est facile à contourner avec < 0.1 W même pour la plus grande version.

Au début, je me suis passé d'un tel écran car il me suffisait de récupérer les données via un navigateur web. Cela fonctionne également très bien avec un smartphone si vous activez un VPN sur votre propre routeur (à condi-tion qu'il soit possible de le faire) et que vous l'utilisez pour accéder à votre DTU à distance. Mais finalement, j'ai été tenté par l'idée d'ajouter un écran et j'ai conçu un nouveau circuit imprimé avec écran : d'abord avec un écran OLED de 1.3 pouce et ensuite une grande version avec une diagonale de 2.42 pouces. Mon DTU avec écran est maintenant placé sur le haut-parleur du PC, de sorte que je peux toujours voir le changement des rayonne-ments solaires. Dans ce qui suit, nous montrons comment connecter un tel écran avec quatre fils simples.

Circuit(s) et construction

Les trois BoB ESP32 de la figure 2 nécessitent un câblage différent en raison de l'affectation différente des broches. Heureusement, les deux modules radio Nordic sont disponibles avec la même affectation des broches. Nous n'utilisons pas ici d'autres versions plus petites avec des pastilles CMS au lieu des connecteurs à 2x4 broches. Cela permet de réduire le nombre de possibilités à trois schémas de circuit ou cartes. Pour faciliter la réalisation, nous n'avons pas dessiné un schéma de circuit classique, mais plutôt un schéma de câblage réalisable. Ainsi, il est possible de souder un module ESP et un module radio sur une plaque perforée et d'effectuer les huit connexions nécessaires très simplement de broche à broche avec fils souples fins et isolés. Les **figures 5, 6 et 7** sont donc similaires. Il n'est pas nécessaire d'établir des nomenclatures détaillées.

Le type de module radio utilisé n'affecte pas les schémas de câblage. Dans les trois circuits, en plus des deux BoB, un petit condensateur électrolytique tampon de 100 µF est inclus, qui, selon des rapports sur Internet, est censé fournir un comportement plus stable du module radio s'il est soudé aussi près que possible des broches 3V3 et GND du module. Certains disent que c'est le cas, d'autres disent le contraire - en tout cas, je l'ai inclus et je l'ai également monté moi-même. À mon avis, ce n'est pas absolument nécessaire.

La **figure 8** montre comment connecter un petit écran OLED I^C à quatre fils ou un morceau de câble ruban à quatre broches à un BoB ESP et afficher les données actuelles de votre onduleur sans PC ni smartphone. Veuillez noter que sur certains modules d'affichage, les broches GND et VCC sont inversées. Avant de les connecter, vérifiez attentivement l'étiquetage des broches ; sinon, vous risquez de sauter en un rien de temps ! Un écran connecté et actif nécessite entre 10 et 25 mA sous 5 V - en moyenne plutôt à la limite inférieure. Sur certaines cartes ESP32, la broche de 5 V est étiquetée VIN. Elle est directement connectée à la ligne 5 V de la prise USB.

Au lieu d'une carte perforée, j'ai utilisé une plaque d'essai pour mes premiers tests. D'un côté, les deux BoB sont branchées, de l'autre, quelques broches sont soudées, puis toutes les broches nécessaires au câblage sont soudées. A ce stade, vous pouvez utiliser les huit morceaux de fil toronné nécessaires et faire les connexions broche par broche. La **figure 9** montre les étapes de réalisation de mon premier prototype. Lorsque l'on utilise une plaque d'essai comme celle-ci, un problème se pose : cinq broches (a à e et f à j) sont connectées. Cela convient bien aux connecteurs des cartes ESP, mais pas à la disposition du connecteur 2x4 du module radio. Ici, vous devez couper les deux tracés associés de la carte en trois endroits avant de souder sur la face supérieure. Le bas de la figure 9 montre comment cela a été fait avec un Dremel et un minidisque

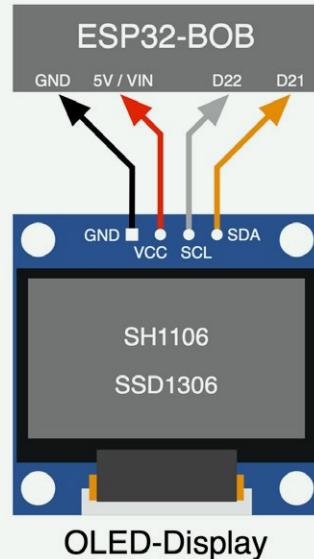


Figure 8. Voici comment connecter un écran OLED via une connexion I^C. Attention : +5 V et GND peuvent être inversées. Une inversion de polarité détruira le circuit !

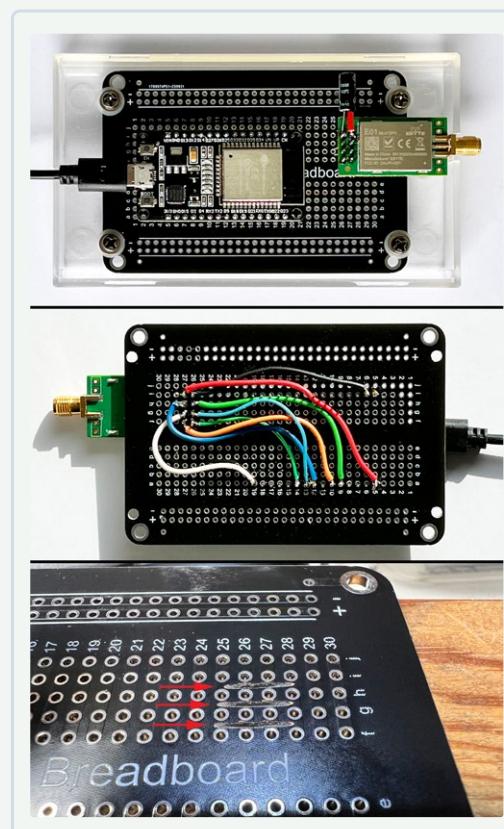


Figure 9. Mon premier prototype sur une plaque d'essai montée avec des fils torsadés. Attention : au niveau du module radio, il faut couper les tracés (en bas, voir le texte).



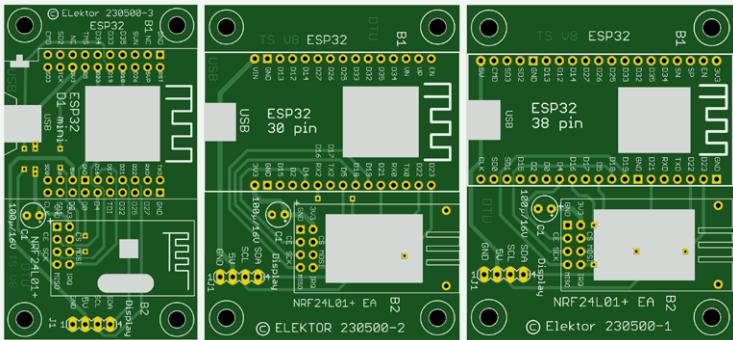


Figure 11. Les trois BoB ESP32 choisi nécessitent des circuits imprimés différents. Les fichiers de dispositions sont disponibles sur [6].

Figure 12. Cartes réalisées. Celle pour le BoB ESP32 à 38 broches est encore une version beta.



Figure 13. Écran de démarrage du programmeur en ligne ESP Tool d'Espressif avec un débit en bauds correctement réglé.

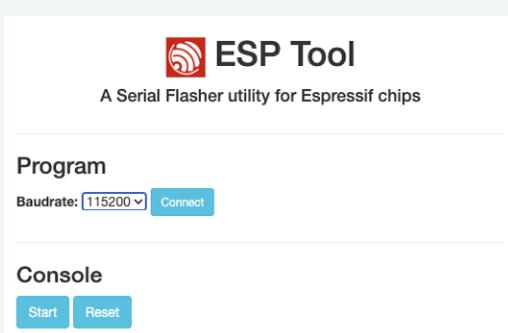
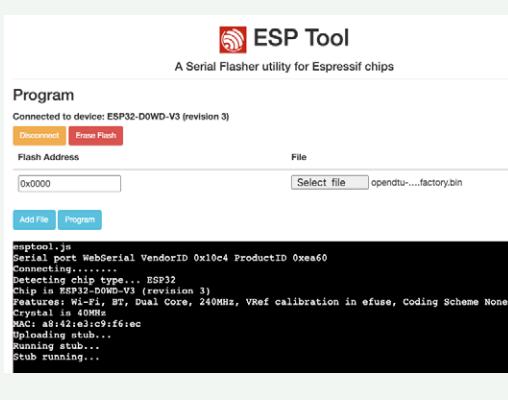


Figure 14. Cette fenêtre pop-up devrait afficher le port série virtuel de l'ESP32 connecté si le pilote correct est installé.



Figure 15. Le premier flashage doit être effectué avec le fichier du micrologiciel *opendtu-generic.factory.bin* à l'adresse 0x0000. Les boutons doivent être pressés pour se connecter (voir texte).



à découper. Si vous oubliez cette étape, vous aurez plusieurs courts-circuits et la carte ne fonctionnera pas. La **figure 10** montre le prototype terminé avec l'antenne externe connectée dans un boîtier semi-transparent. Une solution plus facile, qui permet d'éviter les erreurs d'un montage réalisé manuellement, consiste à utiliser l'un des circuits imprimés que j'ai créés (**figure 11**). Leurs fichiers de disposition sont téléchargeables gratuitement au format Eagle sur la page de cet article sur le site d'Elektor [6]. La version gratuite du logiciel Eagle CAD est suffisante vous permet de concevoir des cartes double face jusqu'à la moitié de la taille d'une carte Euro. Lors du peuplement des BoB ESP32, il suffit de souder les broches avec des pastilles carrées et rondes - cela suffit pour la stabilité et les connexions électriques nécessaires. Dès que j'ai reçu les 3 (premières versions des) cartes du fabricant après avoir envoyé les fichiers Gerber, je les ai assemblées. Elles sont munies de pastilles destinées aux fils torsadés pour la connexion d'un écran. Il est également possible de monter des connecteurs quadrupolaires à cet emplacement pour connecter l'écran de manière amovible. Le haut de la **figure 12** présente la vue de dessus des cartes assemblées, mais pas encore finalisées.

Logiciel

Le premier chargement du micrologiciel OpenDTU sur l'ESP32 vierge n'est pas simple. Tout d'abord, vous devez visiter le site web correspondant [3] et vous documenter. Plusieurs méthodes sont décrites. Je préfère l'outil en ligne d'Espressif [7], qui ne fonctionne pas avec Safari sur Mac - mais d'autres navigateurs sont supportés. Après avoir connecté le DTU au PC ou au Mac via USB, il faut d'abord régler un débit en bauds de 115 200 sous Program, puis cliquer sur le bouton bleu Connect (figure 13). Dans la fenêtre contextuelle qui apparaît (figure 14), sélectionnez l'interface série appropriée (bleu clair). Vous pouvez déjà voir le type de la puce convertisseur USB/série - ici un CP2102 - mais il en existe d'autres. Bien entendu, cela ne fonctionne que si le pilote approprié est installé, ce qui devrait être très simple sous Windows 11 grâce à l'installation automatique. Si ce n'est pas le cas, des liens adéquats sont disponibles sur le site d'Espressif. Le flashage avec le Mac avec Chrome se fait également facilement après l'installation des pilotes CP2102 ou CH340.

Cette étape peut prendre beaucoup de temps avant que vous puissiez avoir un résultat. Sur le BoB ESP32, il peut y avoir un ou deux petit(s) bouton(s). Vous devez d'abord appuyer sur le bouton *Boot*, puis brièvement sur *EN* simultanément. Pour le mini-BoB doté d'un seul bouton, vous n'avez qu'un seul choix. Ainsi, l'outil se connecte à l'ESP32 et l'écran de la **figure 15** apparaît. Par sécurité, effacez d'abord la mémoire avec Erase Flash et sélectionnez ensuite le fichier du micrologiciel *opendtu-generic.factory.bin* [8] sous File. Ce fichier est le bon pour le premier flashage, car il contient également un chargeur de démarrage (bootloader). Ensuite, vous

pouvez effectuer la mise à jour directement par Wifi ou à distance (*OTA, over the air*). Avant de cliquer sur *Program*, vous devez définir l'adresse 0x0000 comme adresse de la mémoire flash, comme indiqué. Ce qui se passe (avec un peu de chance) après avoir cliqué sur *Program* est montré dans la **figure 16** : après une minute et demie, la programmation de l'ESP32 est terminée.

Après le redémarrage de l'ESP32, la liste des réseaux locaux sans fil disponibles devrait en afficher un avec le SSID "OpenDTU-*". L'astérisque représente une séquence de chiffres. La connexion à ce réseau fonctionne avec le mot de passe par défaut "openDTU42". Si vous entrez maintenant <http://192.168.4.1> dans la barre d'adresse du navigateur, le site web de DTU apparaît (**figure 17**). Cela fonctionne également avec Safari.

Vous pouvez maintenant vous connecter en tant qu'administrateur avec le mot de passe openDTU42. Il va sans dire que vous devriez modifier ce mot de passe dans les paramètres dès que possible. Une fois que vous êtes connecté, vous pouvez effectuer les modifications nécessaires dans le menu *Settings*. En plus de l'anglais, la langue du site peut être changée en allemand ou en français. Tout d'abord, nous allons dans *Network Settings* (**figure 18**). Ici, vous entrez le SSID et le mot de passe de votre propre réseau Wifi, pour que le DTU se connecte à votre propre réseau au prochain démarrage, et que vous puissiez y accéder aisément. Pour ce faire, vous devez vérifier l'adresse IP attribuée au DTU sur votre propre routeur Wifi. La plupart des routeurs permettent d'attribuer toujours la même adresse IP au DTU. C'est utile car cela vous permet de marquer le DTU d'un signet. Plus bas, le temps en minutes pendant lequel le DTU agit comme un point d'accès après le démarrage et fournit au WLAN le SSID "OpenDTU-*" est spécifié.

Sous *NTP Settings*, vous pouvez spécifier un serveur de temps et entrer la latitude et la longitude de la position de la centrale du balcon [9] afin qu'aucune requête inutile de l'onduleur n'ait lieu pendant la nuit. Dans les réglages de l'onduleur, vous devez entrer son numéro de série et lui attribuer un nom (**figure 19**). Un onduleur ne peut être adressé et identifié qu'avec ce numéro. Il convient de noter le numéro de série attaché à l'onduleur ou de le photographier avant de procéder à l'installation. Vous pouvez également spécifier plusieurs onduleurs. En cliquant sur le crayon à côté de l'onduleur que vous avez créé, vous accédez à une page de configuration où vous pouvez activer des éléments tels que l'arrêt nocturne de le DTU et entrer les désignations et la puissance des modules solaires connectés.

Enfin, il y a le *Device Manager*. La **figure 20** montre la configuration des broches et des appareils. Cependant, vous ne pouvez pas les régler ici, mais seulement les sélectionner comme profil en haut si un profil adéquat a été lu au préalable. Un clic sur *Display* conduit à la fenêtre des réglages explicites de la **figure 21**. Mais comment obtenir les profils nécessaires si l'on veut activer un écran, par exemple ? Les profils sont disponibles sur la page

```

Connected to device: ESP32-D0WD-V3 (revision 3)
Flash Address File
0x0000 Select file opendtu....factory.bin
Add File Program

esptool.py
WebSerial VendorID 0x10c4 ProductID 0xeaa0
Connecting...
Detected chip type...: ESP32
chip is ESP32-D0WD-V3 (revision 3)
Features: WI-FI, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None
Crystal: 40MHz
MAC: 9C:D8:42:00:00:0f:ec
Uploading stub...
Running stub...
Stub loading...
Starting flash (this may take a while)...
Chip erase completed successfully in 3.208s
Compressed 159232 bytes to 1023489...
Writing at 0x00000000... (1)
Writing at 0x000019fc... (34)
Writing at 0x0000194c... (44)
Writing at 0x00001a67... (64)
Writing at 0x0178fd9... (964)
Writing at 0x017e149... (998)
Writing at 0x0183dd0... (100%)
Wrote 159232 bytes (1023489 compressed) at 0x0 in 92.191 seconds.
Hash of data verified.
Leaving...
  
```

Figure 16. Après 1,5 minute, le micrologiciel est chargé sur le microcontrôleur et le DTU redémarre.

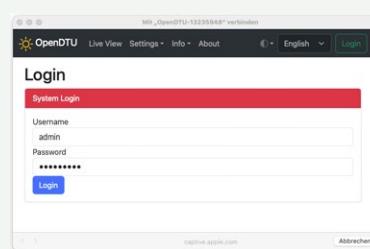


Figure 17. Après vous être connecté au réseau Wifi des DTU, vous pouvez accéder à la page de configuration des DTU à l'adresse <http://192.168.4.1>, où vous devez vous connecter.

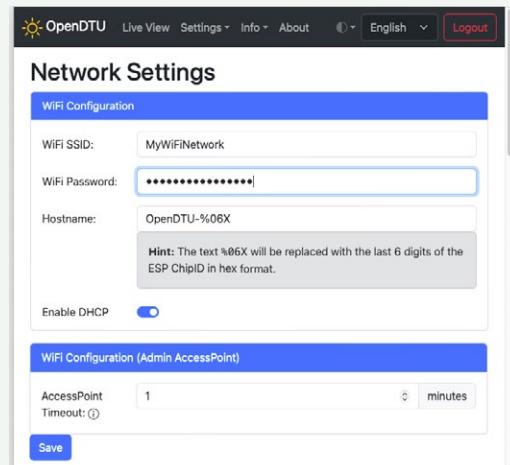


Figure 18. Il est maintenant temps de commencer les réglages. Tout d'abord, entrez le SSID et le mot de passe de votre propre réseau Wifi.

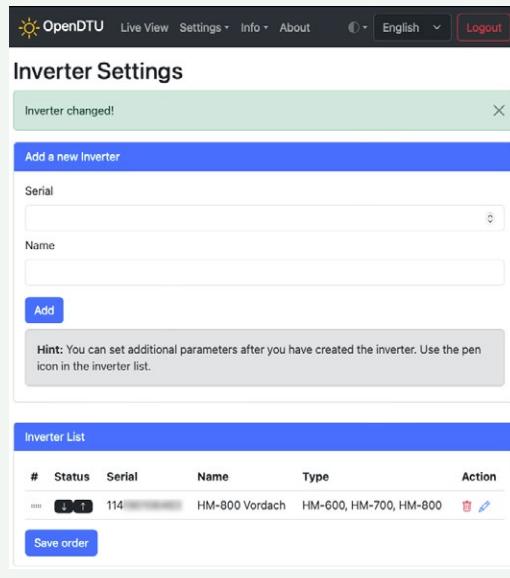


Figure 19. Vous entrez ici le numéro de série de votre onduleur ainsi que la durée de la fonction de point d'accès et vous ajoutez les données des panneaux solaires connectés via l'icône du crayon pour l'onduleur concerné.

Figure 20. Un fichier de profils permet de spécifier l'affectation des ports d'E/S aux appareils connectés. Ici, le module radio et l'écran sont intéressants.

The screenshot shows the 'Device-Manager' interface under the 'Connection settings' tab. A dropdown menu at the top right lists profiles: 'NRF24', 'NRF24 with SH1106', and 'NRF24 with SSD1306' (which is selected). A note below says: 'Your device may stop responding if you select an incompatible profile. In this case, you must perform a deletion via the serial interface.' Below this is a table titled 'Connection overview' with columns: Category, Name, Selected, and Active. It lists connections for 'Display' (clk, cs, data, reset, type) and 'Nrf24' (clk, cs, en, irq, miso, mosi). A 'Save' button is at the bottom.

The screenshot shows the 'Device-Manager' interface under the 'Display' tab. It includes settings for 'Enable Power Safe' (on), 'Enable Screensaver' (on), 'Display language' (English), 'Rotation' (180 degree rotation), and 'Contrast (60%)'. A 'Save' button is at the bottom. A caption to the right reads: 'Figure 21. Quelques réglages pour l'écran connecté.'

Figure 22. Ici, vous pouvez ici enregistrer la configuration dans un fichier json et spécifier l'affectation des broches selon la figure 20 avec un fichier de profil importé. En outre, tous les paramètres peuvent être réinitialisés à l'"état de livraison " ici.

The screenshot shows the 'Config Management' interface. It has three main sections: 'Backup: Configuration File Backup' (Backup the configuration file to 'config.json'), 'Restore: Restore the Configuration File' (Pin Mapping dropdown set to 'pin_mapping.json', 'Select file' dropdown set to 'OpenDTU_ES...OLED.json', and a 'Restore' button), and 'Initialize: Perform Factory Reset' (a 'Restore Factory-Default Settings' button). Notes in the interface mention replacing the configuration file with the restored configuration and restarting the DTU.

GitHub [10], entre autres, où vous pouvez en télécharger un. Il s'agit de fichiers json facilement éditables. Vous pouvez télécharger un profil *OpenDTU_ESP32_NRF24_OLED.json*, adapté au module radio nRF24L01+ et aux écrans OLED – avec les options présentées sur la **figure 20** – ainsi que les fichiers du circuit imprimé sur la page de cet article sur le site d'Elektor [6]. Vous importez le fichier de profil via *Config Management* (**figure 22**) en passant en mode *Pin Mapping* au centre gauche, en sélectionnant le fichier json avec le profil désiré et en cliquant sur *Restore* à côté sur la droite. Ensuite, le DTU redémarre et vous pouvez faire la sélection décrite de l'affichage, etc. montrée dans la **figure 20**.

Dans la figure 23, vous pouvez admirer le prototype dans son boîtier en plastique transparent. Ce DTU est situé directement sur l'enceinte gauche de mon bureau, près du moniteur. Grâce au grand écran de 2,42 puces, je suis toujours informé de la récolte solaire en cours.

En savoir plus

Dans les paramètres d'OpenDTU, vous pouvez configurer beaucoup de détails intéressants. Vous le découvrirez rapidement. Le système récupère automatiquement la date et l'heure via un serveur NTP. Si le serveur prédefini ne fonctionne pas, vous pouvez entrer l'adresse IP de votre propre routeur. Une autre chose intéressante est que vous pouvez sauvegarder la configuration (voir **figure 20**) de sorte que vous gardez une version de votre travail au cas où vous auriez fait une erreur dans votre configuration.

Si vous cliquez sur le bouton rouge dans l'outil de mesure en *Live View*, une fenêtre s'ouvre où vous pouvez définir une limitation de puissance de manière temporaire (jusqu'au redémarrage de l'onduleur) ou permanente (par exemple, si vous avez un onduleur plus puissant que ce que la loi autorise actuellement). La **figure 24** montre mon onduleur de 800 W limité à 75% = 600 W. Une fois que la situation juridique en Allemagne aura changé (cette année, espérons-le), je changerai la valeur à 100%. En outre, avec d'un logiciel externe (système domotique ou similaire) et de capteurs de courant appropriés, il est possible de réaliser un suivi automatique de la puissance générée, de sorte qu'une alimentation zéro *feed-in* (ou autoconsommation pure) est atteinte.

OpenDTU est capable d'envoyer les données via MQTT à un broker MQTT. De telles solutions sont disponibles dans le cloud et localement (c'est-à-dire sur de nano-ordinateurs comme un Raspberry Pi). Avec les outils appropriés, il est possible de visualiser graphiquement les données entrantes à distance, via internet. Vous n'êtes alors pas dépendant de clouds propriétaires et vous avez un contrôle total sur vos données. Le DTU lui-même nécessite environ 70 à 110 mA sans écran à 5 V et un courant un peu plus élevé avec écran. Une valeur très faible ! J'ai également essayé l'autre solution open-source - AhoyDTU – et je n'arrive pas à décider laquelle je préfère.

OpenDTU offre plus d'informations, mais n'est pas aussi facile à installer et à configurer qu'AhoyDTU. La méthode d'installation de ce dernier logiciel est décrite dans un numéro spécial du magazine allemand Elektor consacré à la technologie photovoltaïque. [11] ↪

230500-04

Questions ou commentaires ?

Contactez Elektor (redaction@elektor.fr).



À propos de l'auteur

Thomas Scherer a d'abord suivi une formation de technicien en électronique de télécommunication avant de travailler à la rédaction d'Elektor à partir de 1980. Après avoir étudié la psychologie et passé plusieurs années dans la recherche fondamentale, il est rédacteur indépendant pour Elektor depuis des décennies. Il a un grand labo d'électronique et, outre les neurones, il s'intéresse beaucoup aux électrons, c'est-à-dire à tout ce qui a trait à l'électricité.



Figure 23. Un prototype avec un écran OLED de 2,42 pouces est placé sur le haut-parleur gauche de mon poste de travail. Comme vous pouvez le constater, le temps était légèrement nuageux.

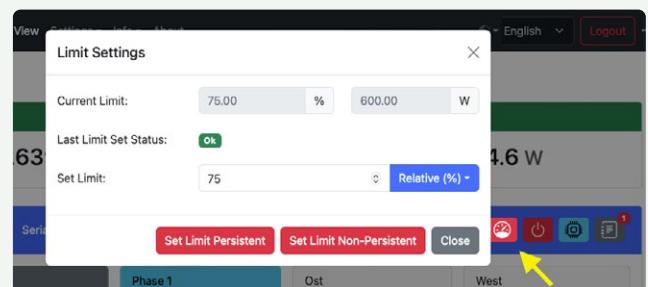
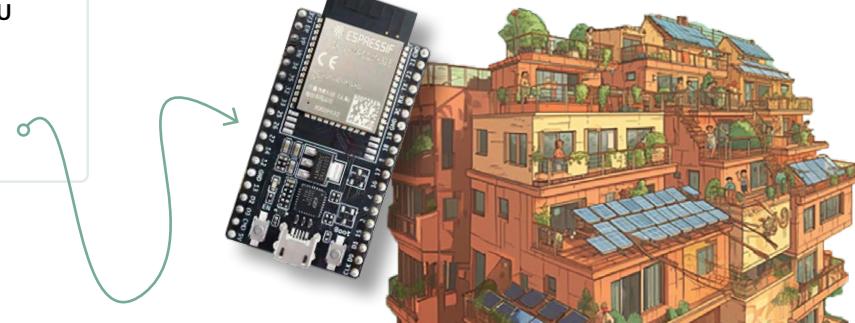


Figure 24. En cliquant sur l'icône rouge avec le compteur, vous accédez au réglage de la puissance maximale de l'onduleur. Vous pouvez régler la puissance en pourcentage ou en valeur absolue en watts, de manière temporaire ou permanente.



LIENS

- [1] T. Scherer, "centrale solaire sur balcon", Elektor 9-10/2021 : <https://www.elektormagazine.fr/magazine/elektor-184/59877>
- [2] Forum Mikrocontroller.net forum (en allemand) : <https://mikrocontroller.net/topic/525778>
- [3] OpenDTU sur GitHub : <https://github.com/tbnobody/OpenDTU>
- [4] AhoyDTU sur Github : <https://github.com/lumapu/ahoy>
- [5] Carte ESP32 d'Espressif : <https://espressif.com/en/products/devkits/esp32-devkitc>
- [6] Téléchargements pour cet article : <https://elektormagazine.fr/230500-04>
- [7] Programmateur en ligne pour ESP32 : <https://espressif.github.io/esptool-js>
- [8] Fichier du micrologiciel OpenDTU : <https://github.com/tbnobody/OpenDTU/releases>
- [9] Coordinate finder : <https://latlong.net>
- [10] Device profiles : <https://tinyurl.com/opendtugithub>
- [11] Elektor-Sonderheft Solartechnik : <https://elektor.de/20596>