

enregistreur de données énergétiques

mesurer et enregistrer la consommation d'énergie



Georg Luber (Allemagne)

Il existe potentiellement de nombreuses applications pour lesquelles il peut être utile d'enregistrer en continu la production ou la consommation d'énergie électrique. Il peut s'agir par exemple de la consommation d'énergie de certains circuits domestiques, de l'énergie produite par une centrale solaire de balcon, ou encore d'une installation photovoltaïque plus importante. Il existe évidemment des solutions d'enregistrement prêtes à l'emploi, mais vous pouvez également construire vous-même un appareil approprié.

L'enregistreur présenté ici utilise un dispositif de mesure doté d'une sortie conforme à l'interface So pour enregistrer la consommation d'énergie électrique. Des appareils de mesure adaptés existent

Caractéristiques

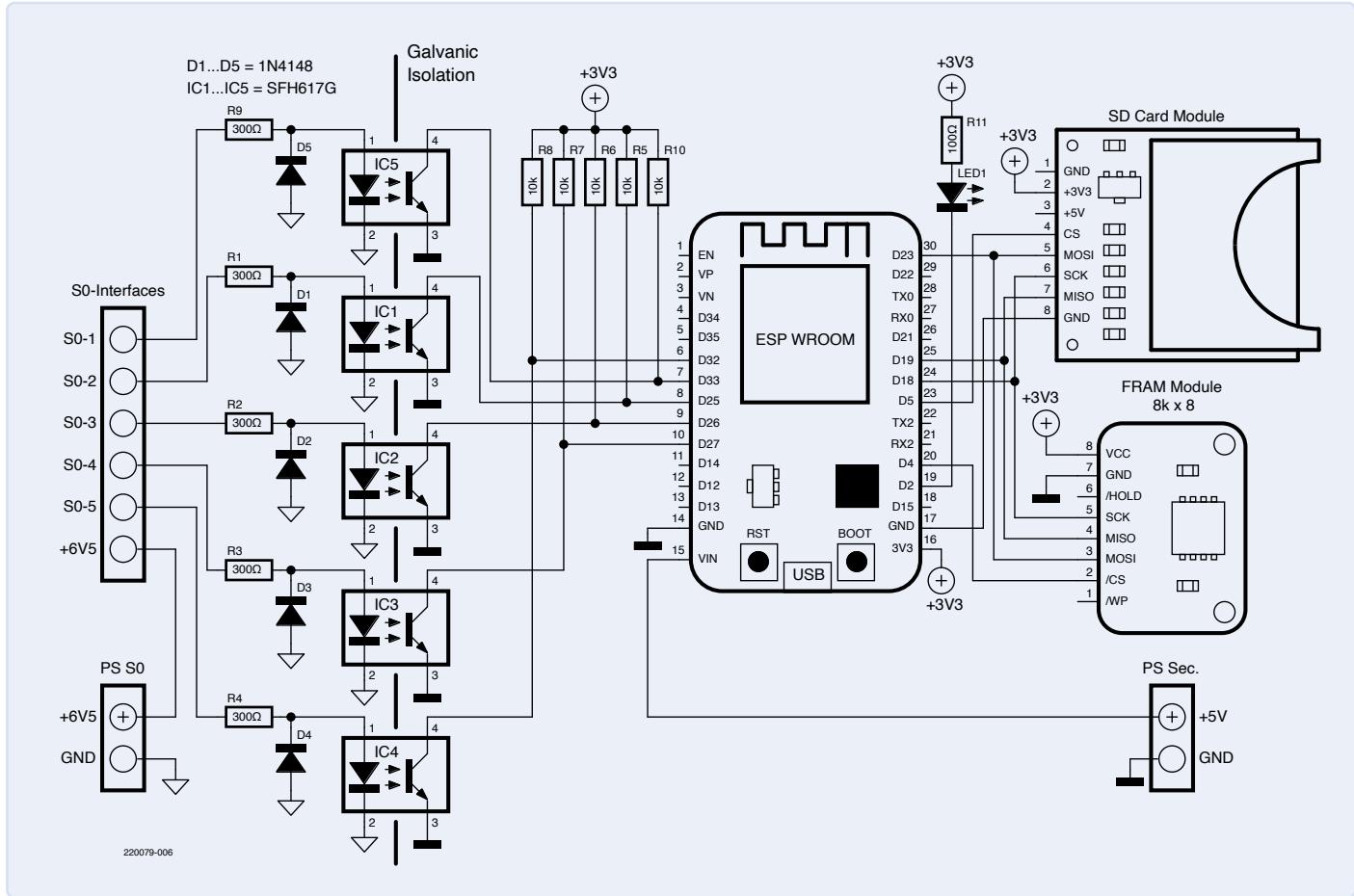
- Précision et sécurité avec un appareil de mesure doté d'une interface So et d'une isolation électrique.
- Acquisition configurable pour intégrer jusqu'à cinq appareils de mesure.
- Mise en mémoire tampon des données dans le module FRAM pour protéger la carte SD.
- Enregistrement sur la carte SD au format CSV.
- Résolution temporelle : 5 minutes.
- Un fichier distinct est enregistré chaque jour.
- Affichage des données individuelles et agrégées (mesures du jour et du jour précédent en kWh).
- Affichage de l'intensité du champ Wi-Fi et du nombre de réinitialisations du µC (compteur de démarrage).
- Téléchargement des données en mode FTP.
- Mise à jour en mode OTA : possibilité de mise à jour par le Wi-Fi.
- Environnement de développement : VSCode avec PlatformIO.

pour moins de 20 € (par exemple, sur Amazon, eBay ou directement auprès de fournisseurs situés en Extrême-Orient). Vous pouvez ainsi mesurer la consommation d'énergie des circuits électriques raccordés au réseau ou le rendement des systèmes photovoltaïques, puis « consigner » ou enregistrer les valeurs à l'aide de l'électronique décrite ici.

Après tout, pourquoi les microcontrôleurs ont-ils été inventés ? À peu de chose près pour ce type d'application. Avant l'ère de l'informatique, l'enregistrement de la consommation ou de la production d'énergie au fil du temps pour un usage domestique aurait été difficile et pratiquement impossible. Grâce aux faibles coûts des microcontrôleurs et à leurs exceptionnelles capacités, non seulement l'enregistrement numérique ne pose plus de problème aujourd'hui, mais même des démarches sophistiquées, par exemple une connexion Wi-Fi, deviennent possibles. Les possibilités de l'enregistreur de données énergétiques sont décrites dans l'encadré **Fonctionnalités**. La **figure 1** donne un aperçu de la solution finalisée. Elle montre comment combiner un dispositif de mesure et un enregistreur, notamment les blocs d'alimentation, installés dans un boîtier de distribution en plastique.



Figure 1. L'enregistreur de données énergétiques finalisé avec l'appareil de mesure et le bloc d'alimentation, installé dans un petit boîtier en matière plastique.



Circuit

Grâce au microcontrôleur, le circuit de l'enregistreur de données énergétiques (**figure 2**) est assez simple. Sur le côté gauche, se trouvent un certain nombre d'optocoupleurs servant à l'isolation électrique, par le biais desquels les données de l'interface So de l'appareil de mesure sont envoyées aux entrées correspondantes du microcontrôleur. Celles-ci sont contrôlées par des interruptions pour qu'aucune impulsion ne soit perdue. J'ai utilisé un module ESP32 bon marché car il possède une puissance de calcul suffisante et une interface Wi-Fi. Sur le côté droit, se trouvent deux modules de mémoire — un emplacement pour une carte SD servant au stockage de masse et un module FRAM supplémentaire, qui met temporairement en mémoire tampon les données correspondant à une période de 5 minutes pour réduire le nombre de cycles d'écriture sur la carte SD.

Le circuit étant très simple, je l'ai construit sur une platine d'essai. Comme vous pouvez le voir sur les deux blocs bleu clair, deux alimentations séparées sont prévues pour des raisons de sécurité. Pour les interfaces So de gauche, une tension de 6,5 V est suffisante. Le reste du circuit reçoit sa tension d'une alimentation en 5 V (à droite). Une capacité de charge de 0,5 A chacun est suffisante. Pour des raisons d'isolation électrique, les deux lignes de masse GND des alimentations ne doivent en aucun cas être connectées. En outre, au-dessous des optocoupleurs, et entre leurs entrées et leurs sorties, tout le cuivre doit être

retiré sur une longueur d'au moins 4 mm. La **figure 3** montre la présentation obtenue.

L'isolation électrique au moyen de deux alimentations séparées permet également de connecter l'ESP32 à l'interface USB d'un PC (par exemple, pour permettre l'envoi de futures mises à jour au contrôleur installé). Au lieu d'utiliser deux alimentations, une solution mono-alimentation associée à une charge de courant plus élevée, avec un convertisseur continu/continu isolant et un régulateur de tension, aurait également été possible, mais sans que cela rende les choses plus faciles ou moins chères.

Figure 2. Le circuit de l'enregistreur de données énergétiques est assez simple.

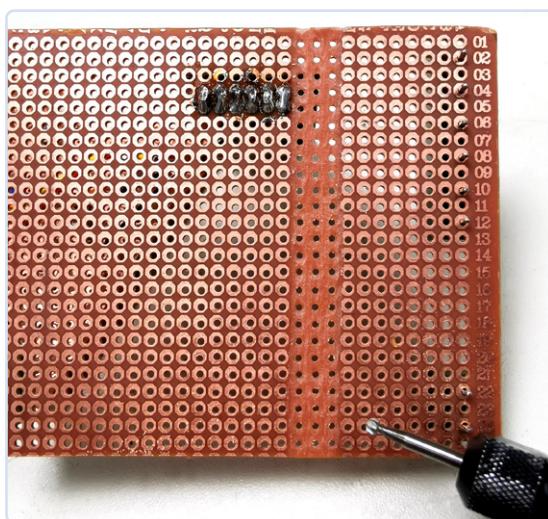


Figure 3. Le cuivre de la face inférieure de la platine d'essai, entre les entrées et les sorties des optocoupleurs, a été enlevé avec un petit cutter.

A	B	C	D	E
1 /2021/07/31.csv - 00:00				
2 426572	415390	132309	129324	218375
3 05:55				
4 >				
5 426572	415390	132309	129324	218375
6 426574	415391	132309	129324	218375
7 426575	415393	132309	129324	218375
8 426575	415393	132309	129324	218375
9 426576	415394	132309	129325	218375
10 426577	415396	132310	129326	218377
11 426579	415398	132310	129326	218378
12 426583	415401	132310	129326	218380
13 426587	415405	132310	129326	218382

Figure 4. La structure des données devient visible après importation d'un fichier CSV dans une feuille de calcul.

Stockage et transmission des données

Comme mentionné précédemment, l'utilisation de deux modules de mémoire n'est pas un luxe. Les interfaces So peuvent parfois délivrer plusieurs impulsions par seconde. Si les valeurs mesurées étaient recueillies dans la mémoire interne du microcontrôleur, les données pourraient être perdues lors d'une réinitialisation. D'autre part, les enregistrer immédiatement dans une carte SD réduirait massivement la durée de vie de la carte. En effet, avec seulement une valeur par seconde, il se produirait 31,5 millions de cycles d'écriture par an. Or, les cellules de mémoire d'une carte SD atteignent leur fin de vie au-delà de seulement 1000 à 3000 cycles d'écriture. Même une carte à haute capacité présenterait donc très certainement des défauts dans la première année.

Pour éviter cette situation, un tampon stable est prévu. Le module FRAM externe utilisé ici n'a qu'une capacité de stockage de 8 Ko, mais il est largement suffisant pour collecter un grand nombre de valeurs. Le principal avantage d'une mémoire FRAM est qu'elle autorise au moins 10^{10} écritures, selon le fabricant - typiquement, ce type de mémoire peut même accepter plusieurs millions de milliards de cycles d'écriture. Toutes les cinq minutes, les données collectées dans la mémoire FRAM sont transférées vers la carte SD. Ce mouvement correspond à un peu plus de 100 000 opérations d'écriture par an, et comme une carte SD comporte de nombreuses cellules de mémoire, elle peut être utilisée pendant plusieurs années sans problème.

De plus, le nombre de redémarrages de l'enregistreur est stocké dans la mémoire FRAM, ce qui vous permet de vérifier à tout moment la fréquence de déclenchement des réinitialisations. De plus, chaque jour, un nouveau fichier est créé sur le support SD, et les données du jour y sont enregistrées. Pour économiser de l'espace mémoire, l'enregistrement n'est lancé que si l'une des valeurs du compteur a changé depuis minuit. Si l'enregistreur de données énergétiques sert à consigner la consommation électrique, cette fonction est en fait superflue. En revanche, s'il s'agit de surveiller la production d'énergie d'une installation solaire, il vaut mieux que l'enregistrement ne commence qu'après le lever du soleil (c'est-à-dire au moment où commence la production électrique du jour).

Structure des données, et autres

La **figure 4** montre une section d'une feuille de calcul dans laquelle les données ont été importées au format CSV. La cellule A1 contient la date et l'heure de création du fichier. La ligne 2 contient ensuite les dernières valeurs de la journée précédente servant de nouveau point de départ. Le premier changement d'une valeur mesurée a eu lieu à 05h55 (cellule A3). À partir de la ligne 5, les nouvelles données du jour suivent. Les colonnes contiennent les données des différents appareils de mesure. Chaque ligne comporte un écart de temps de cinq minutes par rapport à la ligne suivante. Les données sont enregistrées pour la journée jusqu'à minuit. Un nouveau fichier est créé chaque jour. La quantité d'énergie peut alors être calculée à partir de la différence entre les mesures relevées à l'aide de la valeur associée à l'impulsion (par exemple, 0,5 Wh/impulsion). Le format CSV est économique et permet l'importation dans n'importe quel tableur comme Excel, OpenOffice/LibreOffice Calc

Name	Größe	Geändert
2021	9 Elemente	07.01.22 21:06
2022	11 Elemente	02.11.22 22:25
2023	1 Element	08.01.23 10:06
01	17 Elemente	17.01.23 17:21
01.CSV	7,8 kB	08.01.23 10:07
02.CSV	7,8 kB	08.01.23 10:07
03.CSV	7,7 kB	08.01.23 10:07
04.CSV	7,8 kB	08.01.23 10:07

Figure 5. La structure du dossier, avec une répartition par année, mois et jour.

ou Numbers sur Mac. Il est ensuite possible de traiter les données comme vous le souhaitez.

Sur la carte SD, les données sont stockées selon une structure de dossiers temporels. Au niveau supérieur, il y a des dossiers pour chaque année, chacun d'entre eux contenant des sous-dossiers pour les mois, qui contiennent à leur tour les fichiers individuels de chaque jour. La **figure 5** représente cette structure. Dans cet exemple, les mesures issues de cinq appareils sont enregistrées au format entier (voir la **figure 4**). Grâce au format CSV, la quantité de mémoire nécessaire est relativement modeste. Avec seulement quelques kilo-octets par appareil de mesure et par jour. Le transfert des données de l'enregistreur vers un PC se fait au moyen du protocole FTP. Les clients FTP habituels, comme FileZilla, sont adaptés à cette opération. Comme l'enregistreur est intégré au réseau domestique grâce au Wi-Fi, cette méthode est probablement la plus simple.

J'ai créé le code source de l'ESP32 en utilisant l'IDE PlatformIO pour VSCode [1]. Le code est disponible en téléchargement gratuit sur la page web de cet article [2].

Les données suivantes doivent être personnalisées dans le fichier *main.cpp* : à partir de la ligne 30, le SSID et le mot de passe du réseau Wi-Fi doivent être saisis. Le nom d'utilisateur et le mot de passe pour le FTP sont spécifiés à la ligne 813. Par défaut, « esp32 » sert de nom d'utilisateur et de mot de passe. Bien entendu, le client FTP a également besoin de l'adresse IP de l'enregistreur de données énergétiques. Cette adresse est fournie par l'interface USB après réinitialisation. L'idéal serait de configurer votre routeur de manière à ce que l'enregistreur de données ait toujours la même adresse IP.

Interface Web intégrée

Les données actuelles ainsi que la valeur totale de la journée et de la journée précédente sont affichées à l'aide du serveur web intégré. Le serveur héberge deux pages. Une page affiche la puissance actuelle par appareil de mesure ainsi que le total de tous les appareils connectés. En outre, l'énergie totale du jour et de la veille est affichée en kWh. L'autre page indique l'intensité du champ du réseau Wi-Fi et le nombre de redémarrages de l'enregistreur.

Les **figures 6 et 7** sont des captures d'écran d'un smartphone. Comme vous le remarquez immédiatement, je n'utilise pas l'enregistreur pour mesurer la consommation, mais pour surveiller mon installation photovoltaïque, qui compte cinq onduleurs.

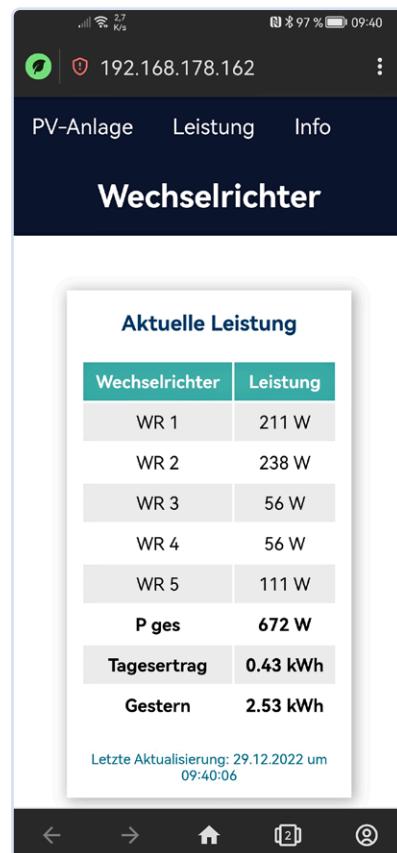


Figure 6. L'interface web intégrée affiche les résultats des mesures.



Figure 7. Si vous ajoutez la mention « / about » à l'adresse IP, ou si vous cliquez sur Info, vous obtiendrez ces informations.

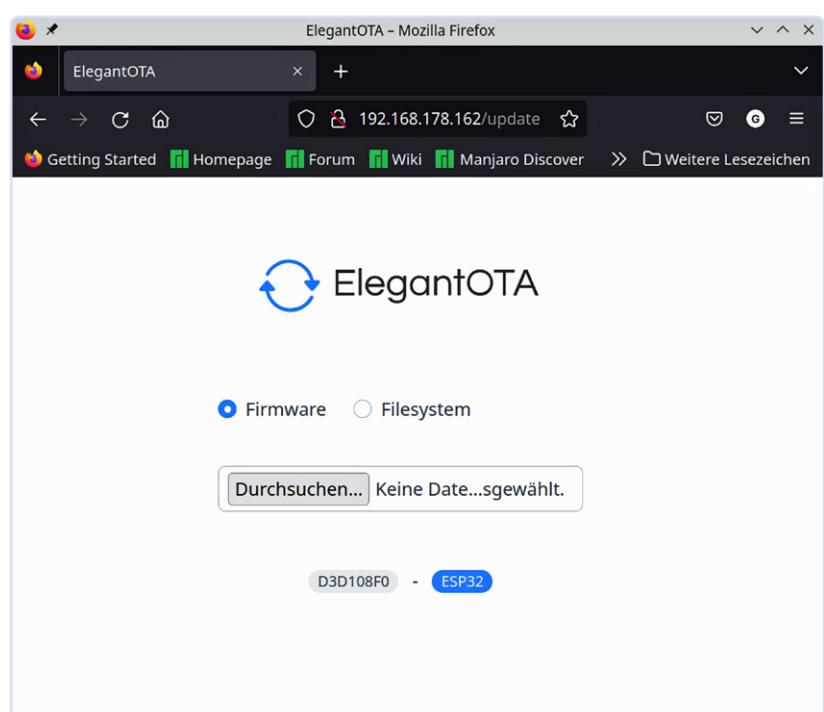


Figure 8. Les mises à jour du micrologiciel et de la structure des données sont possibles grâce au Wi-Fi.



►
Figure 9. Quatre dispositifs de mesure de type DDS5188 en fonctionnement.

C'est pourquoi j'ai adapté les étiquettes des différentes valeurs à l'utilisation que j'en fais. Cependant, vous pouvez facilement modifier cela dans le logiciel si nécessaire. Lorsque vous vous connectez à l'adresse IP dans le navigateur, vous obtenez l'affichage représenté sur la **figure 6**. En ajoutant « /about » à l'adresse ou en cliquant sur le bouton Info, l'affichage illustré par la **figure 7** apparaît.

Une remarque supplémentaire concernant la connexion Wi-Fi : si vous installez l'enregistreur dans un boîtier métallique, vous aurez très certainement des problèmes de réception.

Mise à jour en mode Wi-Fi

Une fois l'enregistreur de données énergétiques intégré dans un boîtier et installé, vous ne souhaiterez probablement pas démonter à nouveau l'électronique en cas de changement de logiciel. Vous pouvez donc mettre à jour l'enregistreur via le Wi-Fi (mode OTA, ou Over The Air). Il est ainsi possible, si nécessaire, de charger de nouveau le logiciel complet ainsi que le système de fichiers - les pages Web avec JavaScript et le fichier CSS.

Pour ce faire, il vous suffit de saisir l'adresse IP en ajoutant le suffixe « /update ». La **figure 8** illustre ma configuration. J'ai réalisé la mise à jour en mode OTA grâce à la bibliothèque d'Ayush Sharma [3].

Un circuit simple

Le circuit est si simple qu'une liste de pièces n'est pas nécessaire. Outre la carte de développement ESP32, que l'on trouve partout, les composants importants sont un emplacement pour support SD, que l'on peut trouver sur des cartes de liaison bon marché, et le module FRAM, également disponible sur des cartes de liaison. Ces cartes sont disponibles avec différentes implantations de broches. Vous ne devez donc pas

prêter attention aux numéros de broches, mais plutôt à leurs désignations lors de la connexion. Concernant le module de carte SD, MOSI peut être connecté à DI ou SI, et MISO à DO ou SO. Concernant les appareils de mesure, n'importe quel modèle bon marché pour montage sur rail DIN avec une interface SO peut convenir. Le modèle DDS5188 type I que j'ai utilisé (**figure 9**) est facilement disponible et très bon marché.

S'il va de soi que vous devez faire preuve de la prudence nécessaire lorsque vous travaillez sur le secteur, une remarque s'impose : un certain nombre de lois nationales s'appliquent aux interventions sur le tableau contenant les fusibles. En Allemagne, vous devez être enregistré auprès de l'exploitant du réseau en tant qu'électricien qualifié. De même, les interventions sur les sous-répartiteurs externes connectés au tableau contenant les fusibles ne sont autorisées que pour les électriciens qualifiés formés et possédant les connaissances appropriées. ◀

VF : Pascal Godart — 220079-04

À propos de l'auteur

Électricien de formation, Georg Luber a étudié l'électrotechnique et travaillé de nombreuses années dans les installations et la sécurité électriques. Il est intervenu au sein de différents comités de normalisation (DKE/VDE, CENELEC, IEC et ISO) aussi bien au niveau national qu'international. Il s'est également intéressé à l'automatisation des bâtiments, en particulier avec le protocole et les systèmes KNX. Georg Luber est intervenu dans le développement de logiciels et de projets électroniques dans ces domaines.

Des questions, des commentaires ?

Contactez Elektor à l'adresse redaction@elektor.fr.



Produits

➤ **ESP32-DevKitC-32D (SKU 18701)**
www.elektor.fr/18701

➤ **Compteur à pince PeakTech 4350 (SKU 18161)**
www.elektor.fr/18161

LIENS

- [1] PlatformIO pour VSCode : <https://platformio.org/platformio-ide>
- [2] Page web de l'article : <https://www.elektrormagazine.fr/220079-04>
- [3] Bibliothèque OTA d'Ayush Sharma : <https://github.com/ayushsharma82/ElegantOTA>
- [4] Projets ESP32 : <https://randomnerdtutorials.com/projects-esp32>