



mise à jour du projet #3 : **compteur d'énergie** basé sur l'ESP32

intégration et test avec Home Assistant

Saad Imtiaz (Elektor)

Dans la précédente mise à jour du projet, nous vous avons présenté les améliorations apportées au schéma et au circuit imprimé du compteur d'énergie ESP32. Dans cet article, nous nous concentrerons sur la réalisation pratique et l'intégration de cette nouvelle version. Vous trouverez un guide détaillé sur la configuration du compteur avec ESPHome et Home Assistant pour un contrôle efficace de la consommation d'énergie. En outre, nous abordons le calibrage de l'appareil.

Dans l'épisode précédent [1], nous avons exploré les améliorations apportées au schéma et au circuit imprimé du compteur d'énergie ESP32, en mettant l'accent sur les améliorations de la modularité et des caractéristiques de sécurité. Avant de nous pencher sur la prochaine mise à jour du projet, faisons un bref rappel.

Dans les dernières avancées de notre projet de compteur d'énergie ESP32, nous avons opté pour le microcontrôleur ESP32-S3, ce qui a

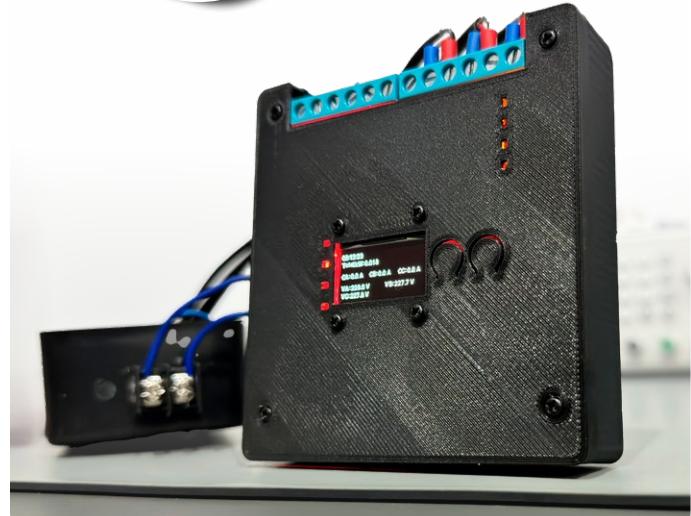


Figure 1. Le compteur d'énergie ESP32 assemblé avec son écran OLED et ses indicateurs d'état en direct.

permis d'améliorer la puissance de traitement et d'élargir les fonctionnalités. Cette mise à jour nous a permis de réduire la taille du circuit imprimé et d'intégrer un système d'alimentation basé sur un transformateur. Nous utilisons un transformateur abaissageur de tension 230 V à 12 V pour l'échantillonnage de la tension et pour alimenter le système. Cela renforce considérablement la sécurité tout en assurant la compatibilité avec les installations monophasées et triphasées.

Nous avons également introduit d'autres améliorations notables, telles que l'intégration d'un convertisseur buck AP63203WU-7 plus efficace, la modularité du circuit imprimé et le calibrage du circuit d'échantillonage du transformateur de courant. Ces améliorations permettent non seulement d'optimiser les performances et les fonctionnalités du compteur d'énergie, mais aussi de réduire son coût et sa taille.

Dans cet article, nous discuterons des étapes suivies pour rendre ce compteur d'énergie opérationnel et du chemin qu'il a parcouru depuis les étapes initiales sur l'établi jusqu'à son intégration dans la boîte à disjoncteurs. En outre, nous aborderons aussi la calibration et la configuration du système, et enfin, son intégration à Home Assistant avec ESP Home pour afficher et surveiller les données collectées à partir du compteur d'énergie. La **figure 1** montre le projet en action, logé dans un boîtier imprimé en 3D et équipé d'un écran OLED. L'image met en évidence des indicateurs d'état qui suivent et affichent la consommation d'énergie en temps réel de manière claire et instantanée.

Assemblage

Le nouveau circuit imprimé a été conçu pour être plus compact et plus facile à souder. La disposition assure un espace adéquat pour chaque composant, rendant ainsi le processus de soudure plus aisés. Pour faciliter la reproduction du projet et permettre des modifications tant que pour les passionnés que pour les professionnels, nous avons partagé la liste complète des composants (BOM) au format Mouser et les fichiers de production sur le dépôt GitHub d'Elektor Lab [2].

Pour les connexions d'échantillonnage de tension et de courant, nous avons utilisé des borniers à vis de CUI Devices. La qualité de ces borniers est bien meilleure que celle des borniers bleus bon marché souvent utilisés sur de nombreux modules de capteurs. Comme nous travaillons avec des tensions alternatives et des compteurs d'énergie, il est vital d'avoir des connexions sûres et fiables.

La réduction du bruit est un aspect critique dans la conception de ce circuit imprimé. Pour y répondre, nous avons intégré des condensateurs électrolytiques et céramiques autour de la puce de mesure de l'énergie ATM90E32S. Cette disposition permet de filtrer les bruits de basse et de haute fréquence, assurant ainsi une mesure d'énergie plus précise et plus stable. La carte est représentée dans la **figure 2**.

Comme mentionné précédemment, nous avons décidé d'utiliser un transformateur abaisseur de tension pour l'échantillonnage de la tension et comme source principale d'alimentation de l'ensemble du système. Trouver un tel transformateur est facile et abordable, mais la plupart de ces transformateurs abaisseurs prennent beaucoup de place lorsqu'ils sont utilisés dans un boîtier personnalisé, comme le montre la **figure 3**. Il est donc préférable d'utiliser des transformateurs DIN Rail Bell dans ce cas pour rendre l'installation plus propre et plus sûre ; de tels transformateurs peuvent être facilement trouvés en ligne. En outre, la précision des mesures de tension dépend des caractéristiques des transformateurs, notamment de la précision de leur rapport de tension, du déphasage et de la linéarité.

Vous avez peut-être remarqué dans les images qu'un unique transformateur est connecté au compteur d'énergie. Le compteur d'énergie a été configuré pour être utilisé en mode monophasé, en court-circuitant le cavalier JP8 situé sur la face arrière du circuit imprimé, comme le montre la **figure 4**. Pour faire fonctionner le compteur d'énergie en mode triphasé ou pour mesurer la tension de chaque phase dans un système triphasé avec trois transformateurs abaisseurs, vous devez brancher les côtés primaires de trois transformateurs aux phases correspondantes (L1, L2, L3). Du côté secondaire, relier une extrémité de l'enroulement de chaque transformateur à un point neutre commun pour former une configuration en étoile (Y). Les extrémités libres des enroulements secondaires (V1, V2, V3) fourniront alors les sorties de tension (UA, UB et UC) sur le circuit imprimé pour chaque phase. Il est essentiel de s'assurer que les transformateurs sont adaptés à la tension et au courant du système, de maintenir une isolation rigoureuse entre les circuits primaires et secondaires pour des raisons de sécurité et d'assurer une connexion neutre stable et équilibrée afin d'éviter toute imprécision dans les mesures.

Configuration dans ESPHome et Home Assistant

Dans le cadre de notre plan de développement, nous sommes en train de développer un micrologiciel spécialisé pour exploiter les capacités de la puce de mesure de l'énergie et les fonctions avancées d'intelligence artificielle de l'ESP32-S3. Bien que le développement d'un tel micrologiciel nécessite beaucoup de temps et soit toujours en cours, cela ne limite pas l'usage du compteur d'énergie. L'appareil est déjà pleinement fonctionnel avec les plateformes existantes telles que Home Assistant, offrant ainsi une solution immédiate pour la gestion de l'énergie. Dans cet article, nous allons nous concentrer sur

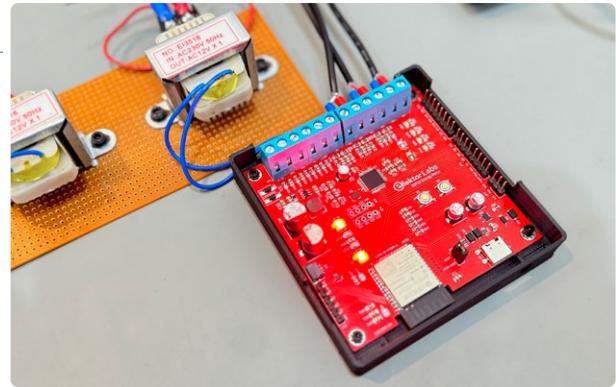


Figure 2. Circuit imprimé du compteur d'énergie ESP32 entièrement assemblé.

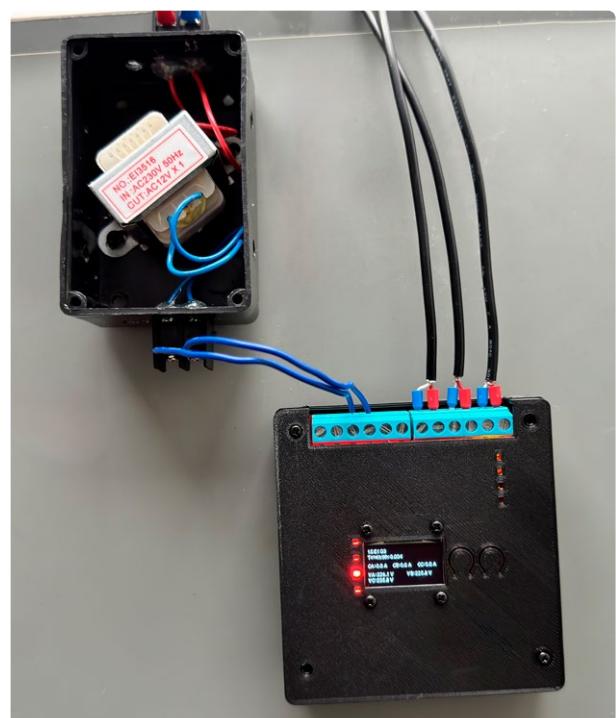


Figure 3. Transformateur abaisseur de tension 220 V à 12 V dans un boîtier personnalisé



Figure 4. Configuration des cavaliers pour un fonctionnement monophasé sur la carte du compteur d'énergie ESP32.

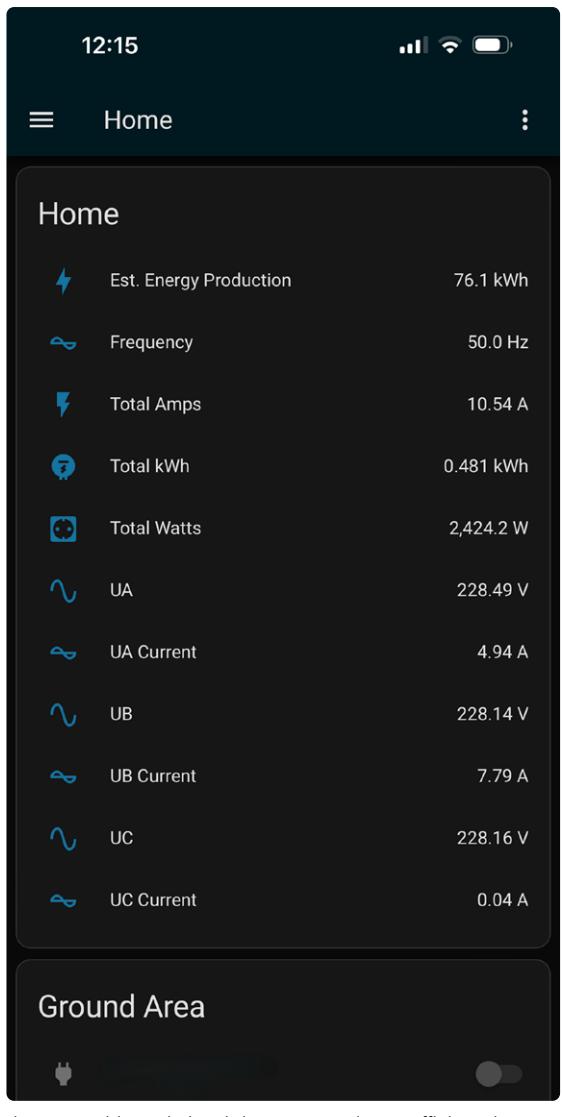


Figure 5. Tableau de bord du Home Assistant affichant les données énergétiques en temps réel du compteur d'énergie ESP32.

l'intégration du compteur d'énergie ESP32 dans Home Assistant [3] et ESPHome [4]. Cette section vous guidera à travers les étapes de la configuration du compteur d'énergie dans l'environnement Home Assistant afin d'utiliser toutes ses fonctionnalités.

Pour configurer le compteur d'énergie ESP32 avec le firmware ESPHome et l'intégrer dans Home Assistant, commencez par installer Home Assistant (consultez le guide détaillé proposé par notre collègue Clemens Valens [5]). Une fois installé, ajoutez l'intégration ESPHome via *Add-on Store*. Ensuite, créez un nouveau projet dans ESPHome pour votre appareil ESP32. Cela génère automatiquement un fichier de configuration YAML initial (un tel fichier YAML est spécifique à un projet ESPHome distinct avec tous les capteurs utilisés et de nombreuses autres options). Assurez-vous de télécharger ce fichier par défaut avant de passer aux étapes suivantes.

Connectez votre ESP32 à votre ordinateur et sélectionnez le bon port COM pour l'ESP32-S3. Dans le tableau de bord ESPHome, cliquez sur *Install* et choisissez le fichier *.bin* pour charger le micrologiciel. Une fois le chargement réussi et que votre compteur d'énergie ESP32 est reconnu par Home Assistant, vous pouvez modifier la configuration YAML initiale. Pour ce faire, ouvrez le tableau de bord ESPHome dans Home Assistant, trouvez votre appareil et cliquez sur *Edit* sur la carte du compteur d'énergie. Remplacez la configuration existante par le contenu YAML fourni dans le dépôt GitHub [2]. Assurez-vous de configurer correctement vos identifiants API, OTA et WiFi dans cette nouvelle configuration YAML.

Installez la nouvelle configuration sans fil sur votre compteur d'énergie ESP32. Une fois cette opération effectuée, l'appareil sera opérationnel et connecté. Pour afficher les données du compteur d'énergie sur le



Figure 6. Graphe détaillé de l'historique de la consommation d'énergie dans Home Assistant.

tableau de bord de votre Home Assistant, il suffit d'attribuer l'appareil ESPHome à une zone spécifique dans Home Assistant. Cela permet d'organiser votre tableau de bord en regroupant les appareils selon leur emplacement physique ou logique dans votre domicile. Pour une illustration de ce que vous pouvez obtenir, consultez la **figure 5**, qui montre les données du compteur d'énergie sur le tableau de bord de Home Assistant.

L'intégration du compteur d'énergie ESP32 dans Home Assistant simplifie non seulement le processus de gestion de la consommation d'énergie, mais ouvre également la voie à de nombreuses fonctions puissantes offertes par la plateforme. Home Assistant offre une interface intuitive pour la visualisation des données en temps réel, l'automatisation de la gestion et l'intégration transparente avec d'autres appareils connectés dans votre maison. Grâce à cette intégration, il est possible de créer des graphiques temporels détaillés et d'effectuer des analyses dans Home Assistant, offrant une compréhension approfondie sur les modèles de consommation d'énergie au fil du temps, comme le montre la **figure 6**. Ces informations permettent aux utilisateurs de prendre des décisions éclairées concernant leur consommation d'énergie, d'identifier les économies potentielles et d'optimiser l'efficacité énergétique de leur maison.

En suivant le processus d'installation décrit ci-dessus, les utilisateurs peuvent tirer pleinement parti de ces capacités, transformant le compteur d'énergie ESP32 en un élément essentiel de leur écosystème de maison intelligente. Cette intégration ne se contente pas d'améliorer le fonctionnement du compteur d'énergie, mais enrichit également l'expérience globale de la maison intelligente grâce à des outils complets de surveillance et de gestion de l'énergie.

Configuration YAML

La configuration YAML proposée permet de configurer le compteur d'énergie ESP32 avec ESPHome, ce qui permet de surveiller les paramètres électriques clés tels que la tension, le courant et la puissance sur les trois phases. Elle tire parti des capacités du capteur ATM90E32, avec des définitions détaillées pour la communication SPI



Figure 7. Configuration pour tester et calibrer le compteur d'énergie ESP32 en utilisant une charge variable.



Figure 8. Pince ampèremétrique pour le calibrage.

et les capteurs spécifiques pour chaque phase. Cette configuration permet non seulement de mesurer mais aussi de calculer la consommation totale, en intégrant un compteur d'énergie quotidien en kWh et un écran OLED pour la visualisation des données en temps réel. Ces configurations sont réalisées conformément aux instructions de la page ESPHome pour le capteur ATM90E32 [6].

Pour garantir la précision des données fournies par le compteur d'énergie ESP32, le calibrage est une étape cruciale. Les détails de l'ajustement des paramètres de gain pour les transformateurs de courant et les entrées de tension seront détaillés dans la section suivante.

Configuration de test et calibrage

Lors de nos tests, nous avons utilisé un sèche-cheveux à plusieurs niveaux de chaleur et de vitesse comme charge, couvrant une gamme de 0,7 A à 8 A. Pour faciliter la mise en place des transformateurs de courant ouverts sur le fil sous tension ou le fil neutre, le cordon d'alimentation d'une rallonge électrique a été dénudé, facilitant ainsi la surveillance directe dans diverses conditions, comme le montre la **figure 7**.

J'ai réalisé le calibrage en courant et en tension du compteur d'énergie ESP32 à l'aide de mon multimètre UT201+. Ce multimètre offre une résolution de 0,001 A et précision de $\pm 4\%$ +10 chiffres pour le courant et une résolution de 0,001 V avec une précision de $\pm 1\%$ +5 chiffres pour la tension. Bien que ce niveau de précision soit suffisant pour la plupart des projets, mais il est légèrement moins précis que celui offert par les modèles de qualité professionnelle.

Lors du calibrage, nous avons comparé les mesures de courant : la pince ampèremétrique indiquait 1,692 A, comme le montre la **figure 8**, tandis que les valeurs calculées par le compteur d'énergie avaient

```

10:54
ESPHome
ESPHome

Logs esp32-energymeter.yaml

preferences to flash...
[22:53:59] [D][esp32.preferences:143]: Saving 1
preferences to flash: 0 cached, 1 written, 0 failed
[22:53:59] [D][sensor:094]: 'Total Watts': Sending
state 1207.07874 W with 1 decimals of accuracy
[22:53:59] [D][sensor:094]: 'Total kWh': Sending
state 0.02185 kWh with 3 decimals of accuracy
[22:53:59] [W][atm90e32:366]: SPI power factor read
error.
[22:53:59] [D][sensor:094]: 'Line Voltage A':
Sending state 233.89999 V with 2 decimals of
accuracy
[22:53:59] [D][sensor:094]: 'Line Voltage B':
Sending state 0.42000 V with 2 decimals of accuracy
[22:53:59] [D][sensor:094]: 'Line Voltage C':
Sending state 0.15000 V with 2 decimals of accuracy
[22:53:59] [D][sensor:094]: 'CT1 Current': Sending
state 1.73400 A with 2 decimals of accuracy
[22:53:59] [D][sensor:094]: 'CT2 Current': Sending
state 1.72100 A with 2 decimals of accuracy
[22:53:59] [D][sensor:094]: 'CT3 Current': Sending
state 1.70300 A with 2 decimals of accuracy
[22:53:59] [D][sensor:094]: 'Active Power CT1':
Sending state -392.60928 W with 2 decimals of
accuracy
[22:53:59] [D][sensor:094]: 'Active Power CT2':
Sending state 0.11808 W with 2 decimals of accuracy
[22:53:59] [D][sensor:094]: 'Active Power CT3':
Sending state 0.08352 W with 2 decimals of accuracy
[22:53:59] [D][sensor:094]: 'Power Factor CT1':
Sending state -0.96700 with 2 decimals of accuracy
[22:53:59] [D][sensor:094]: 'Power Factor CT2':
Sending state 0.15500 with 2 decimals of accuracy

```

DOWNLOAD LOGS EDIT CLOSE

NEW DEVICE

Figure 9. Résultats du calibrage final montrant une amélioration de la précision des mesures.



ATTENTION : Travailler avec un tableau électrique comporte des risques inhérents, notamment un risque d'électrocution ou d'incendie. Il est essentiel de couper l'alimentation électrique avant de commencer l'installation. Dans la plupart des pays, ce travail ne peut être effectué que par un électricien certifié !

entre 1,70 et 1,73 A après le calibrage, comme le montre la **figure 9**. Compte tenu des spécifications de l'UT201+ et du SCT-013-000, un transformateur de courant ouvrant de classe 1 qui garantit une précision de 1% par rapport à la valeur réelle, ce faible écart reste dans la marge d'erreur prévue. Toutefois, pour une précision encore plus grande, il est possible d'utiliser une pince ampèremétrique plus précise.

Pour améliorer encore la précision du compteur d'énergie ESP32, des ajustements ont été apportés aux paramètres de gain pour les mesures de tension et de courant. Pour la tension, le capteur a été calibré en utilisant la formule suivante :

$$\text{New gain_voltage} = (\text{your voltage reading} / \text{ESPHome voltage reading}) * \text{existing gain_voltage value}$$

De même, pour les ajustements de courant :

$$\text{New gain_ct} = (\text{your current reading} / \text{ESPHome current reading}) * \text{existing gain_ct value}$$

Nous avons ensuite mis à jour ces nouvelles valeurs de gain dans le fichier de configuration YAML de l'ESPHome, puis nous avons recompilé et chargé le micrologiciel. Vous pouvez répéter ce processus autant de fois que nécessaire pour garantir une précision optimale. Ces valeurs calibrées permettent d'affiner les mesures et sont cruciales pour l'établissement de rapports et d'analyses précis dans toute installation de surveillance de la consommation d'énergie.



Figure 10. Compteur d'énergie ESP32 installé dans un tableau de disjoncteurs, surveillant la consommation d'énergie en temps réel.

Installation du compteur ESP32 dans le tableau électrique

L'installation du compteur d'énergie ESP32 dans mon tableau de disjoncteurs a été un processus gérable qui a exigé une attention méticuleuse aux détails pour garantir la sécurité et le bon fonctionnement du système. J'ai commencé par sélectionner un circuit avec la limite d'ampérage la plus basse. Ce choix stratégique offrait une marge de sécurité ; le disjoncteur se déclencherait en cas de surtension imprévue ou de défaillance du transformateur, offrant ainsi une protection au système.

L'utilisation de transformateurs de courant ouvrants offre un avantage notable grâce à leur simplicité d'installation. Il est possible de fixer rapidement ces transformateurs sur n'importe quelle charge, mais il est crucial de prêter attention au sens du courant pour garantir la précision des mesures. Il est important de noter que si le sens du courant et l'orientation du transformateur de courant ne sont pas correctement alignés, les relevés de puissance apparaîtront négatifs, ce qui indique une installation incorrecte.

La **figure 10** présente une démonstration visuelle du compteur d'énergie ESP32 en action dans le tableau électrique. Cette image montre le compteur d'énergie affichant en temps réel le courant, les mesures de tension et la charge correspondante en kilowatts, illustrant ainsi son fonctionnement en pratique.

Développement et perspectives

Bien que la configuration logicielle actuelle soit fonctionnelle sur ESPHome, nous poursuivons le développement pour étendre les capacités du compteur d'énergie ESP32. Nous travaillons actuellement sur l'intégration d'un nouveau micrologiciel spécialement conçu pour exploiter tout le potentiel de la puce ESP32-S3. Ce micrologiciel devrait inclure des fonctions avancées telles que l'analyse détaillée de l'énergie et des fonctionnalités IA/ML potentiellement révolutionnaires qui pourraient prédire les schémas de consommation énergétique et identifier les appareils à partir de l'empreinte de leur charge.

Bien que les aspects fondamentaux et opérationnels du projet soient achevés, le développement de fonctionnalités plus sophistiquées est une tâche complexe qui prend du temps. Je suis enthousiasmé par les possibilités qui s'offrent et je suis déterminé à repousser les limites de ce que ce compteur d'énergie peut réaliser.

Le projet de compteur d'énergie ESP32 continue d'évoluer, intégrant de nouvelles fonctions à chaque mise à jour. Nous encourageons vivement les membres de notre communauté qui sont intéressés par les futures fonctionnalités d'IA et de ML, ou ceux qui souhaitent contribuer au développement à s'impliquer. La collaboration est essentielle pour accélérer les progrès et aboutir à une solution de surveillance d'énergie plus robuste et plus riche en fonctionnalités. Restez attentifs aux prochaines avancées, nous sommes déterminés à perfectionner et à améliorer cet outil polyvalent de gestion de l'énergie. ↗

240244-04

Questions ou commentaires ?

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



À propos de l'auteur

Saad Imtiaz (ingénieur senior, Elektor) est un ingénieur mécatronicien expérimenté, spécialisé dans les systèmes embarqués, la mécatronique et le développement de produits.

Saad a également travaillé dans l'industrie aéronautique et a dirigé une startup technologique. Chez Elektor, il supervise le développement de projets dans les domaines du logiciel et du matériel.



Produits

➤ **PeakTech 4350 Pince ampèremétrique**
www.elektor.fr/18161

➤ **Siglent SDM3045X Multimètre**
www.elektor.fr/17892

LIENS

- [1] Saad Imtiaz, « mise à jour #2 : compteur d'énergie basé sur l'ESP32 », Elektor 5-6/2024 :
<https://www.elektormagazine.fr/magazine/elektor-344/62819>
- [2] Compteur d'énergie ESP32 | Dépôt Github : <https://github.com/ElektorLabs/esp32-energymeter>
- [3] Home Assistant : <https://home-assistant.io/>
- [4] ESPHome : <http://esphome.io>
- [5] Clemens Valens, « la domotique, c'est facile avec ESPHome, Home Assistant et MySensors », Elektor Magazine 9-10/2020 :
<https://www.elektormagazine.fr/magazine/elektor-156/58991>
- [6] ATM90E32 Power Sensor : <https://esphome.io/components/sensor/atm90e32.html>