

VOS PREMIERS PAS AVEC L'ESP32-C3 ET L'IDO

Bouton Wi-Fi
+ relais Wi-Fi

p. 6

FOCUS SUR

Internet des Objets

p. 54 **Les défis de la commercialisation des solutions IdO**

Problématiques de sécurité, d'évolutivité et de concurrence

p. 12 **Cloud IdO à la sauce Arduino**

Solution pratique de mise en œuvre d'un serveur du nuage

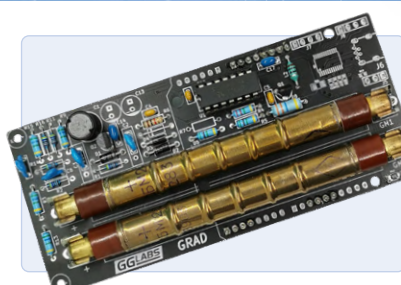
p. 74 **Passerelle intérieure Dragino LPS8**

Configuration rapide de la passerelle LoRaWAN



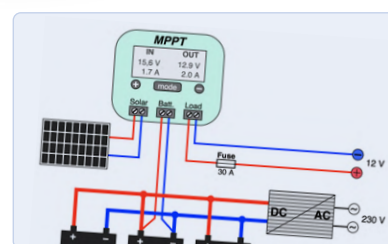
Détecteur de CO₂
Une solution « maison » pour surveiller la qualité de l'air

p. 24



Détecteur Geiger-Müller à double tube (extension pour Arduino)
Capteur très sensible pour très faible rayonnement

p. 18

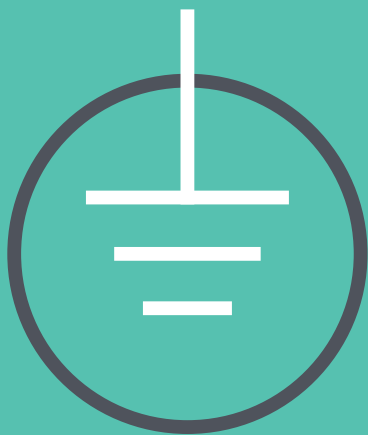


Systèmes solaires autonomes
Production d'électricité indépendante du réseau

p. 104

L 19624 - 495 - F. 15,50 € - RD





NOS PRIX LA MEILLEURE ARME CONTRE LES COÛTS ÉLEVÉS

The best part of your project:
www.reichelt.com

Avec reichelt, optimisez votre budget

L'efficacité de nos services logistique et informatique, développés par nos soins, ainsi que la concentration de nos achats sur des produits de qualité triés sur le volet nous permettent de vous faire bénéficier de prix très avantageux sur de petites quantités. Cela fait de nous le meilleur choix pour les projets de développement, la maintenance et les petites séries, et ce malgré les frais d'expédition.



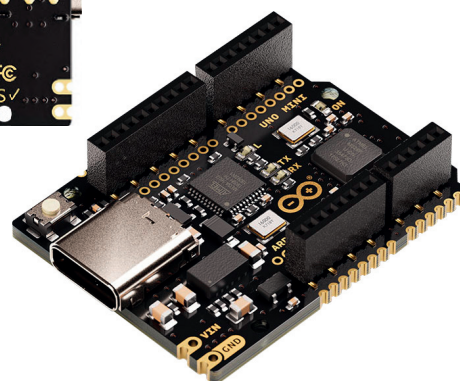
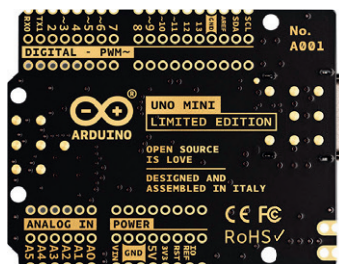
Arduino UNO Mini

ATmega328P, USB-C

L'Arduino UNO mini est une carte à microcontrôleur de 27 x 34 mm équipée d'un microcontrôleur ATmega328P.

Édition limitée – tout dans cette version de l'Arduino UNO est unique : noir et or, design élégant. Chaque circuit imprimé est numéroté et accompagné d'une lettre signée par les fondateurs.

- 14 E/S numériques - interfaces
(6 d'entre elles peuvent être utilisées comme sortie PWM)
- 6 entrées analogiques
- Oscillateur à cristal 16 MHz
- Connexion USB-C



ÉDITION LIMITÉE

N° de commande :
ARDUINO UNO MINI

39,50

Arduino — Microcontrôleurs puissants
pour les tâches de commutation et de commande

Découvrir tout de suite ► <https://rch.it/ard22-fr>



Arduino dans notre magazine en ligne:
Projets, guides pratiques et tutoriels Arduino: lancez-vous!

Découvrir tout de suite ►
<https://rch.it/m-arduino-fr>



Types de paiement :



PRIX DU JOUR! Prix à la date du: 6. 4. 2022

■ Excellent rapport qualité prix

■ Plus de 120 000 produits sélectionnés

■ Livraison fiable - depuis l'Allemagne dans le monde entier

www.reichelt.com

Assistance téléphonique: +33 97 518 03 04

reichelt
elektronik – Tirer le meilleur parti de votre projet

Les réglementations légales en matière de résiliation sont applicables. Tous les prix sont indiqués en € TVA légale incluse, frais d'envoi pour l'ensemble du panier en sus. Seules nos CGV sont applicables (sur le site <https://rch.it/CG-FR> ou sur demande). Semblables aux illustrations. Sous réserve de coquilles, d'erreurs et de modifications de prix. reichelt elektronik GmbH & Co. KG, Elektronikring 1, 26452 Sande (Allemagne), tél. +33 97 518 03 04

45^{ème} année
n° 495 – mai-juin 2022

ISSN 0181-7450
Dépôt légal : mai 2022
CPPAP 1125 T 83713
Directeur de la publication : Donatus Akkermans

Elektor est édité par :
PUBLITRONIC SARL
c/o Regus Roissy CDG
1, rue de la Haye
BP 12910
FR - 95731 Roissy CDG Cedex

Pour toutes vos questions :
service@elektor.fr

www.elektor.fr | www.elektormagazine.fr

Banque ABN AMRO : Paris
IBAN : FR76 1873 9000 0100 2007 9702 603
BIC : ABNAFRPP

Publicité :
Raoul Morreau
Tél. : +31 (0)6 4403 9907
Courriel : raoul.morreau@elektor.com

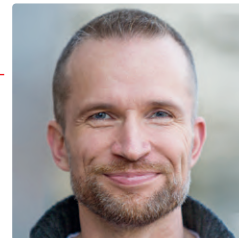
DROITS D'AUTEUR :
© 2022 Elektor International Media B.V.

Toute reproduction ou représentation intégrale ou partielle, par quelque procédé que ce soit, des pages publiées dans la présente publication, faite sans l'autorisation de l'éditeur est illicite et constitue une contrefaçon. Seules sont autorisées, d'une part, les reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective, et, d'autre part, les analyses et courtes citations justifiées par le caractère scientifique ou d'information de l'oeuvre dans laquelle elles sont incorporées (Loi du 11 mars 1957 - art. 40 et 41 et Code Pénal art. 425).

Certains circuits, dispositifs, composants, etc. décrits dans cette revue peuvent bénéficier de droits propres aux brevets; la Société éditrice n'accepte aucune responsabilité du fait de l'absence de mention à ce sujet. Conformément à l'art. 30 de la Loi sur les Brevets, les circuits et schémas publiés dans Elektor ne peuvent être réalisés que dans des buts privés ou scientifiques et non commerciaux. L'utilisation des schémas n'implique aucune responsabilité de la part de la Société éditrice. La Société éditrice n'est pas tenue de renvoyer des articles qui lui parviennent sans demande de sa part et qu'elle n'accepte pas pour publication. Si la Société éditrice accepte pour publication un article qui lui est envoyé, elle est en droit de l'amender et/ou de le faire amender à ses frais; la Société éditrice est de même en droit de traduire et/ou de faire traduire un article et de l'utiliser pour ses autres éditions et activités, contre la rémunération en usage chez elle.

Imprimé aux Pays-Bas par
Senefelder Misset – Doetinchem
Distribué en France par M.L.P. et en Belgique par A.M.P.

rédacteur en chef d'Elektor Magazine



Apprendre de ses erreurs

Écrire un éditorial signifie avoir toujours quelques semaines d'avance sur la date de publication. Lorsque vous lirez ces lignes, j'espère vivement que vous aurez déjà vu les premiers signes de paix en Ukraine. Et peut-être que les discussions sur les mesures liées à la COVID-19 et sur les taux d'incidence élevés se seront calmées.

Quel est le rapport entre les nouvelles (souvent mauvaises) du monde entier et l'électronique ? Au cours des derniers mois, nous avons tous dû nous rendre compte que notre monde est hautement interconnecté et mondialisé. Avant les récents problèmes géopolitiques et la crise de la COVID-19, les choses semblaient aller bien depuis plusieurs décennies ! Les grandes entreprises comme les constructeurs automobiles internationaux s'étaient tellement habituées à cette situation confortable que peu d'entre elles avaient un plan B dans le tiroir, au cas où une catastrophe mondiale se produirait. Une fois les crises actuelles surmontées, de nombreuses entreprises ne commettront certainement plus les mêmes erreurs. L'augmentation de la disponibilité des produits et le raccourcissement des itinéraires de transport seront des priorités plus importantes que la simple obtention du prix le plus bas. Cela sera également bénéfique pour l'industrie électronique.

Dans de nombreux domaines de l'électronique, la croissance permanente ralentit, mais les choses continuent d'avancer – par exemple, dans l'Internet des Objets, le thème de ce numéro. Mon collègue Robert van der Zwan a réalisé des infographies intéressantes à ce sujet (p. 60). Bien entendu, nous vous proposons également de nombreux articles de fond et des projets intéressants ! Restez à l'écoute !



Notre collègue de longue date, Ralf Schmiedel, est décédé en mars 2022. Cela a été un choc pour moi et toute notre équipe. Ralf était en charge du service aux lecteurs d'Elektor depuis 2014. Avec une grande patience et de nombreuses idées bien à lui, cet ingénieur s'est occupé des préoccupations les plus diverses des lecteurs et des auteurs, a compilé les corrections des projets et a aidé notre équipe de rédaction dans d'autres tâches, comme les DVD annuels. De nombreux courriels de remerciement de la part des lecteurs ont montré à quel point son soutien était précieux. Ralf, nous garderons tous un bon souvenir de toi !

notre équipe



Rédacteur en chef :	Jens Nickel
Rédaction :	Eric Bogers, Jan Buiting, Rolf Gerstendorf, Thomas Scherer, Clemens Valens, Marilene Thiebaut-Brodier (coordination)
Service aux lecteurs :	Ralf Schmiedel
Correcteur technique :	Malte Fischer
Laboratoire :	Mathias Claussen, Ton Giesberts, Luc Lemmens, Clemens Valens (responsable)
Maquette :	Giel Dols, Harmen Heida



Elektor est membre de la FIPP, une organisation qui « se développe depuis presque 100 ans pour réunir des propriétaires de médias et des créateurs de contenu du monde entier ».



Elektor est membre de VdZ (association d'éditeurs de magazines allemands) qui « représente les intérêts communs de 500 éditeurs allemands grand public et B2B. »

Vos premiers pas avec l'ESP32-C3 et l'IdO

Bouton Wi-Fi + relais Wi-Fi



Rubriques

3 Édito

36 Démarrer en électronique... (13)

Entrée en scène de la diode

40 Zone D

Trucs et astuces pour tester les composants

68 Drôles de composants

Tubes à ondes progressives

82 Projet 2.0

Corrections, mises à jour et courriers des lecteurs

90 Sur le vif

Façon de pincer

111 Questions d'éthique

10 ans avec le même smartphone ?

114 Hexadoku

Casse-tête pour elektorniciens

FOCUS

69 Narrowband Internet of Things

Normes, couvertures, conventions et modules

FOCUS

74 Passerelle intérieure Dragino LPS8

Configuration rapide de la passerelle LoRaWAN

86 Simulation de circuit avec TINA Design Suite & TINACloud

Extrait : oscillateurs sinusoïdaux

92 Infrastructure graphique WinUI pour les applications Windows

Application de démonstration

100 Création d'interfaces graphiques en Python

4^e partie : la pire des interfaces

104 Systèmes solaires autonomes

Production d'électricité indépendante du réseau

Industrie

FOCUS

54 Les défis de la commercialisation des solutions IdO

Problématiques de sécurité, d'évolutivité et de concurrence

FOCUS

60 Elektor infographie

Faits et chiffres sur l'Internet des Objets

Articles de fond

FOCUS

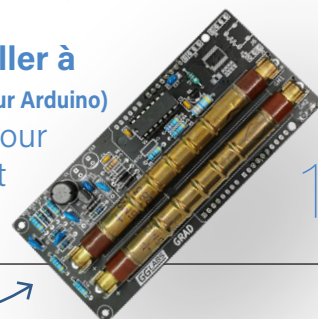
12 Cloud IdO à la sauce Arduino

Solution pratique pour les développeurs d'applications

32 Kit de mesure de la qualité de l'air pour Raspberry Pi de MonkMakes

Mesure de température et de CO₂e

Détecteur Geiger-Müller à double tube (extension pour Arduino)
Capteur très sensible pour très faible rayonnement

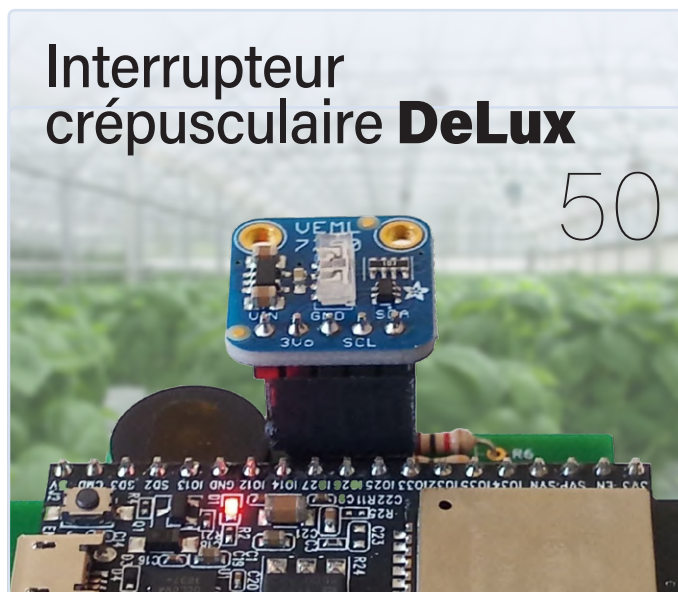




Trucs & Astuces pour tester les composants

Sans appareils de test coûteux

40



Interrupteur crépusculaire DeLux

50

- 62 « Je préfère quand même être en filaire »
Conseils pour développer une interface 1 Gbit/s
dans un environnement industriel
- 66 La détection d'objets en temps réel pour microcontrôleurs
grâce à Edge Impulse FOMO

Réalisations

- FOCUS**
- 6 Vos premiers pas avec l'ESP32-C3 et l'IdO
Bouton Wi-Fi + relais Wi-Fi
- 18 Détecteur Geiger-Müller à double tube
(extension pour Arduino)
Capteur très sensible pour très faible rayonnement
- FOCUS**
- 24 Détecteur de CO₂
Une solution « maison » pour surveiller la qualité de l'air
- 46 Réduction de la consommation d'énergie
de votre repousse-taupes
Grâce au remplacement du 555 par un ATtiny13
- 50 Interrupteur crépusculaire DeLux
Une solution pour une commutation de haute précision,
commandée par la lumière
- 77 Exploration des microcontrôleurs ATtiny
en C et assembleur
Extrait : ports d'E/S d'ATtiny
- FOCUS**
- 83 Mise à jour de la balise GPS LoRa
Recevoir et afficher la localisation à l'aide d'un Raspberry Pi

Bientôt dans ces pages

Le numéro de juillet-août 2022 d'Elektor

Vous retrouverez dans le prochain magazine Elektor l'habituel mélange stimulant de réalisations originales, de circuits soigneusement étudiés, d'articles de fond, de sujets nouveaux, de trucs et d'astuces pour les électroniciens actifs. Le thème de ce numéro sera « Tests et mesures ».

Quelques-uns des points forts :

- > Inductancemètre autonome
- > Mesure de CO₂ avec Sigfox
- > Outils de conception de filtres analogiques
- > Prises intelligentes : dissection et détournement
- > Instrument de mesure de résistance série équivalente, simple et analogique
- > Se lancer avec l'oscilloscope
- > Raspberry Pi Pico en récepteur de signal temporel MSF et SDR
- > Fréquence-mètre pour le secteur

et bien d'autres choses encore !

Le numéro de juillet-août 2022 du magazine Elektor sera publié aux alentours du 7 juillet 2022. La date d'arrivée du magazine papier chez les abonnés dépend des aléas d'acheminement. Le contenu et les titres des articles peuvent être modifiés.



vos premiers pas avec l'ESP32-C3 et l'IdO

Bouton Wi-Fi + relais Wi-Fi



Mathias Claußen (Elektor)

L'IdO n'est pas un livre fermé sur des secrets. De puissants contrôleurs, par ex. le nouvel ESP32-C3, et des environnements conviviaux pour les débutants comme l'EDI Arduino font du développement de petits projets un jeu d'enfant.

En parlant de l'IdO (en anglais IoT = *Internet of Things*), nous admettons que de plus en plus de choses de notre quotidien sont connectées à l'internet. Cela va de l'éclairage aux capteurs domotiques en passant par les voitures, feux de circulation, conteneurs d'expédition et bien plus encore. Dans ces *objets* connectés sont installées des puces interfaçables avec le réseau, ce qui permet d'échanger des informations. La meilleure façon d'aborder la connexion de vos propres applications à l'IdO est de construire un montage pratique simple qui, par ex., crée un lien entre un bouton-poussoir Wi-Fi et un relais Wi-Fi : le bouton peut activer à distance le relais qui renvoie son état au bouton.

Sélection des composants

Soyons ordonnés, choisissons d'abord les composants adéquats. Le kit ESP-C3-12F (**fig. 1**), disponible dans la boutique Elektor s'impose de lui-même. La carte est équipée d'un µcontrôleur ESP32-C3 d'Espressif avec fonction Wi-Fi. L'ESP32-C3 succède à l'ESP8266, qui a fait ses preuves. Outre un noyau CPU moderne, la puce offre un bon choix de périphériques intégrés à la fois faciles à utiliser et puissants. (Consultez notre article sur l'ESP32-C3 [1].) La **figure 2** donne un aperçu des blocs matériels intégrés dans la puce. En plus de l'ESP32-C3, la carte accueille une LED RVB et un convertisseur série-USB. Notre projet

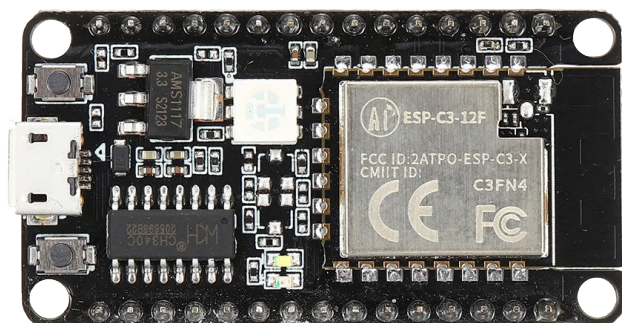


Figure 1. Kit ESP32-C3-12F.

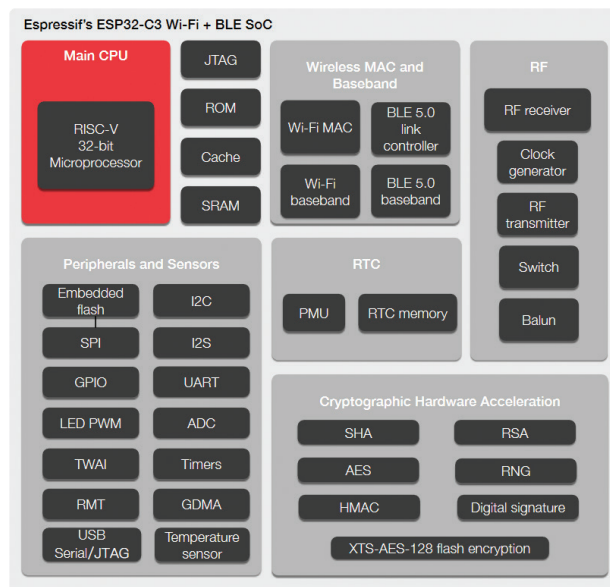


Figure 2. Blocs fonctionnels de l'ESP32-C3 (source : fiche technique de l'ESP32-C3).

nécessite deux kits ESP-C3-12F.

Un capteur et un actionneur sont également nécessaires. Le kit de capteurs 37-en-1 d'Elektor tombe à pic ! Il comprend 35 capteurs (au départ il y en avait 37, mais deux d'entre eux contenaient du mercure et ont été retirés pour des raisons de sécurité). La **figure 3** montre la boîte du kit, la **figure 4** et le document d'information [2] détaillent son contenu. Nous prendrons d'abord le module joystick et utiliserons sa fonction bouton-poussoir comme entrée de commande du système. Une fois connecté à l'autre carte à microcontrôleur, le module relais peut agir comme un actionneur dans le système. Des fils volants (femelle/femelle) sont nécessaires pour connecter les modules. Le kit de fabrication Pimoroni « Mini Breadboards & Jumpers » en contient. (Voir l'encadré **Produits**).

Il nous faut aussi un *serveur local* pour que les appareils IdO puissent échanger leurs données ; un Raspberry Pi sera parfait. Un Raspberry Pi original version 1 conviendrait, mais, pour cette application, nous conseillons une version 2 ou ultérieure. Le Wi-Fi ne fut pas intégré avant le Raspberry Pi 3B, donc pour utiliser les versions antérieures,

il faut ajouter soit une simple clé Wi-Fi, soit un câble Ethernet. Si le petit, mais puissant, Raspberry Pi Zero vous séduit, jetez un coup d'œil au kit Raspberry Pi Zero 2 W (carte avec boîtier et câble). Pour ce projet, il n'est toutefois pas obligatoire d'utiliser un Raspberry Pi. N'importe quel PC sous Linux (distribution Ubuntu par ex. [3]) fera tout à fait l'affaire.

Avant d'entrer dans le vif du sujet, voyons de plus près comment se passent le contrôle et l'échange de données dans cette configuration.

MQTT

Tout dispositif IdO (capteur ou actionneur) doit transférer des données. Pour ce faire, nous pouvons réinventer la roue en développant notre propre protocole de communication, ou bien utiliser un protocole standard éprouvé. Très utilisé, MQTT répond à cette demande. Cet acronyme signifie *Message Queuing Telemetry Transport* (transport de télémétrie par file d'attente de messages), mais cela ne rend plus compte de la fonction actuelle du système qui a évolué. En 2013, il fut officiellement décidé que MQTT serait une norme [4].



Figure 3. Kit de capteurs Elektor 37-en-1.

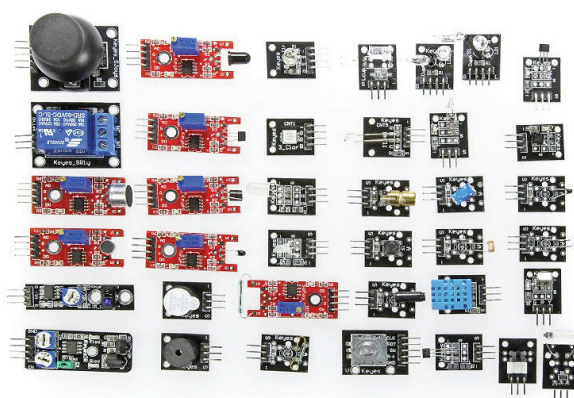


Figure 4. Toute une série de capteurs et d'actionneurs est incluse.

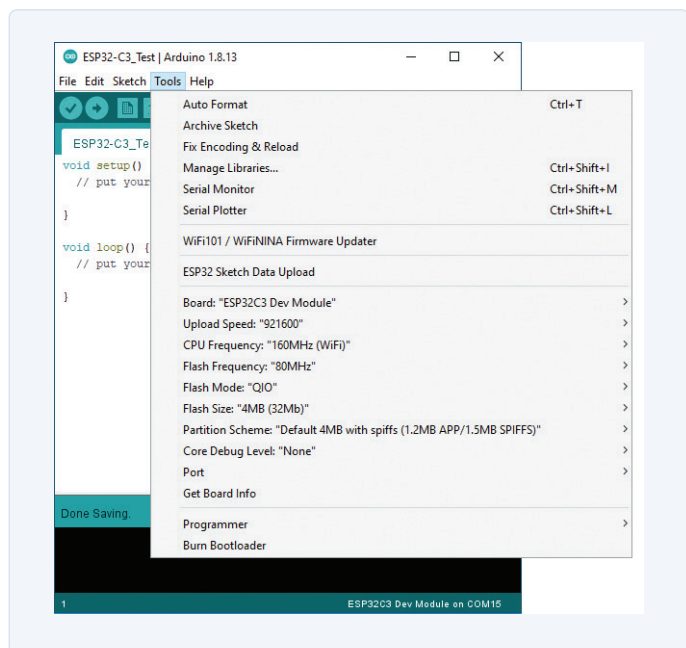


Figure 5. Configuration de l'ESP32-C3 dans l'EDI Arduino.

MQTT prend en charge l'échange de messages à l'aide d'un *broker* (courtier), notre serveur local, sans spécifier le contenu des messages. Cela peut se comparer à l'envoi d'une lettre : la logistique et le format de l'enveloppe sont spécifiés par l'entreprise postale, mais son contenu (message et langage employé) est du ressort exclusif de l'utilisateur. Quel « langage » choisir pour envoyer nos messages ? Il y a diverses possibilités, mais le format JSON (*JavaScript Object Notation*) s'est imposé (pas seulement pour MQTT).

JSON

C'est un format léger d'échange de données pour transférer des messages. Le code est facilement produit et interprété, même par de petits µcontrôleurs. En outre, toute personne comprend, mais

rédige aussi facilement un script JSON. Le site de la norme JSON [5] vous donne un aperçu de ses spécifications. L'emploi de JSON ne se limite pas à MQTT, mais concerne bien d'autres domaines. JavaScript, le langage de programmation dont JSON est dérivé, est l'un des piliers sur lesquels repose aujourd'hui le World Wide Web. Le site de Mozilla [6] présente agréablement JSON avec des exemples pratiques.

Configuration de l'environnement IdO : le courtier MQTT

Comme dans tout projet, une préparation adéquate évite les mauvaises surprises. Pour traiter les messages MQTT, il nous faut un périphérique de courtage situé soit sur l'internet, soit sur notre réseau local. Un courtier local implique que la fonction IdO n'aura pas besoin de services en nuage car, dans notre cas, elle ne servira qu'à transférer des messages entre appareils connectés localement. Un PC retiré du service ou un Raspberry Pi pourront faire l'affaire. Node-RED nous fournit une boîte à outils complète pour développer des applications réseau non limitées au traitement des messages MQTT. Chez Elektor, nous utilisons fréquemment Node-RED [7] pour traiter les données MQTT. Des instructions détaillées permettent une installation rapide sur Raspberry Pi [8] ou sur PC [9].

L'EDI Arduino

L'EDI Arduino constitue un bon environnement de développement. Il n'est pas le meilleur de sa catégorie, mais offre actuellement la prise en charge de l'ESP32-C3 la plus stable. Gratuit, l'EDI Arduino [10] peut être téléchargé et installé depuis la page d'accueil Arduino. Installons ensuite la prise en charge de l'ESP32 par Arduino comme décrit dans la documentation Espressif [11]. Les réglages de la carte sont effectués comme indiqué à la **figure 5**.

Pour nos premiers pas, outre la prise en charge de l'ESP32, nous aurons besoin de quelques bibliothèques. Pour que l'ESP32-C3 puisse envoyer des données via MQTT/JSON, notre exemple fait appel aux bibliothèques *PubSubClient* de Nick O'Leary et *ArduinoJson* de Benoit Blanchon. Installons-les à l'aide du gestionnaire de bibliothèques de l'EDI Arduino (**fig. 6 et 7**).



Figure 6. PubSubClient dans le gestionnaire de bibliothèque Arduino.

Figure 7. ArduinoJson dans le gestionnaire de bibliothèque Arduino.

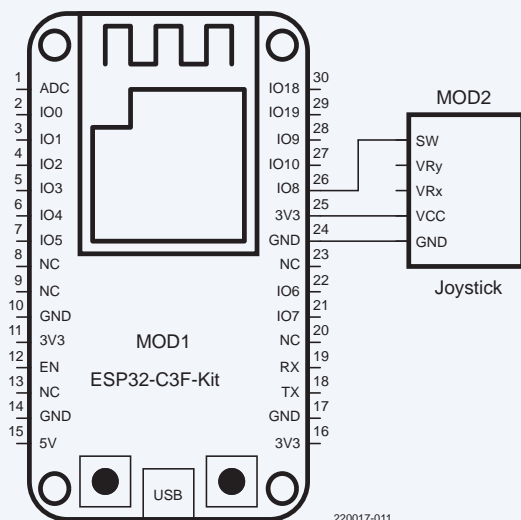


Figure 8. Schéma de l'ESP32-C3 et du joystick.

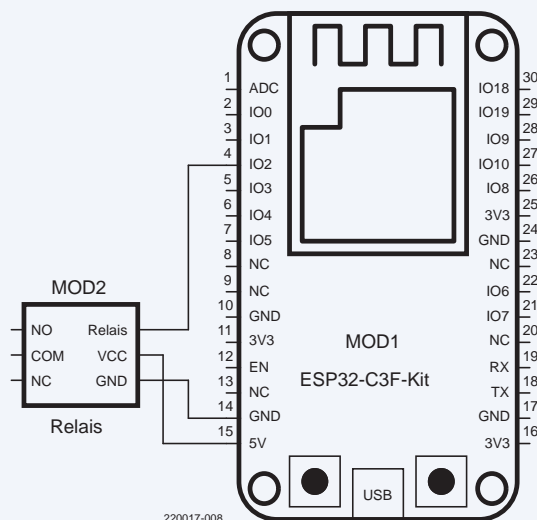


Figure 9. Schéma de l'ESP32-C3 et du relais.

Assemblage du matériel

Les **figures 8 et 9** donnent le schéma de connexion des deux cartes à ESP32-C3. Il n'y a que trois fils de liaison entre le module joystick ou relais et la carte ESP32-C3 correspondante. Notez que le relais s'alimente sur le 5 V de l'USB de la carte ESP32. La **figure 10** montre les modules et cartes câblés.

Configuration du logiciel

Le code source de ce projet est également disponible sur GitHub [12]. Vous y trouverez aussi les croquis pour les deux contrôleurs ESP32-C3. Avant de les transférer sur les ESP32, il faudra entrer quelques informations sur votre réseau local dans ces fichiers. Tout doit être réglé correctement pour que les deux contrôleurs puissent échanger des données avec le courtier MQTT local. C'est le rôle des directives `#define` présentes au début des deux croquis Arduino :

```
#define WIFI_SSID "changeme"
#define WIFI_PASS "changeme"
#define MQTT_SERVER "test.mosquitto.org"
```

Ces trois `#define` doivent refléter votre propre réseau. Le SSID et le PASSWORD de votre réseau doivent être saisis entre guillemets. L'adresse IP interne au réseau de l'ordinateur Node-RED est spécifiée pour le courtier MQTT. Une fois les deux croquis (relais et bouton) modifiés, ils peuvent être *téléversés* sur leur ESP32-C3 respectif. On peut alors mettre les deux ESP32 sous tension et la grande LED de chacun doit commencer à clignoter en blanc. Cela indique que l'ESP32-C3 tente de se connecter au réseau Wi-Fi. Dès qu'une carte y parvient, sa LED s'allume en continu. La couleur de la LED dépend de la fonction de la carte : celle de la carte du relais connecté s'allume en blanc ; selon que le relais est activé ou non, celle de la carte du

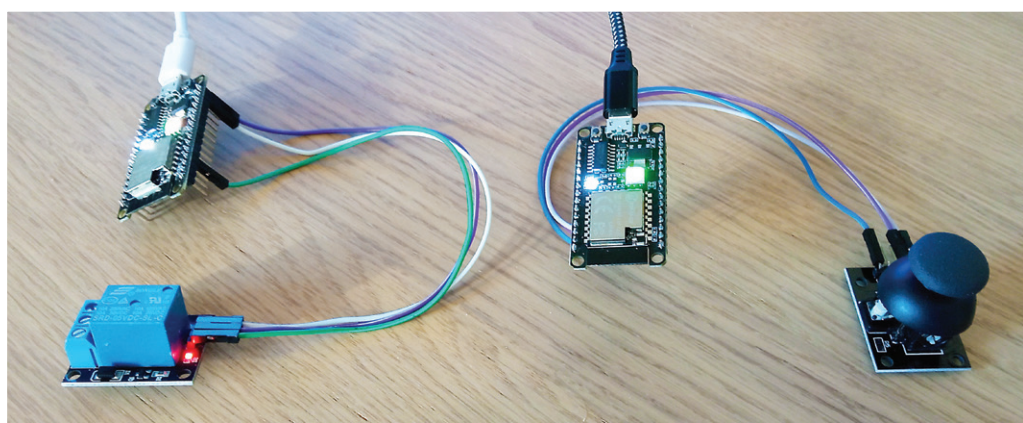


Figure 10. Câblage complet du matériel.

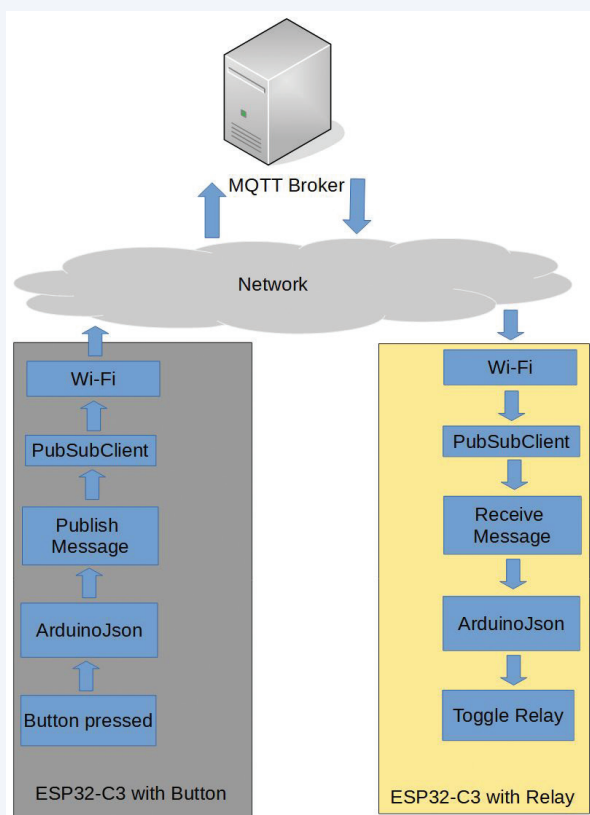


Figure 11. Transfert de données bouton-courtier-relais.

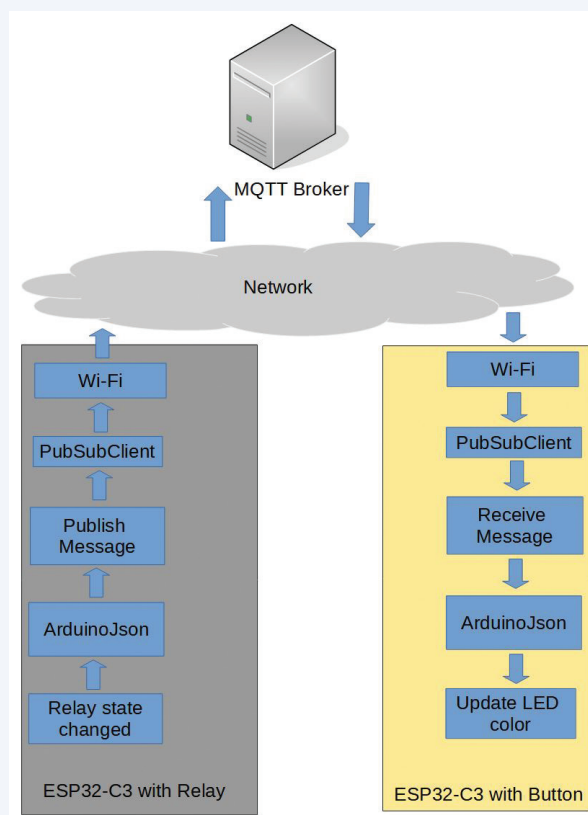


Figure 12. Retour relais-courtier-bouton.

bouton-poussoir connecté s'allume en rouge ou en vert. Si c'est le cas, les deux ESP32-C3 fonctionnent correctement. En appuyant sur le bouton, l'état du relais change et la couleur de la LED passe du rouge au vert et vice versa. Le bouton d'un ESP32-C3 peut donc commander le relais de l'autre ESP32-C3, et il reçoit également des informations sur l'état du relais. Il est temps de fêter ça. Votre première application IdO fonctionne ! Mais comment l'échange de données se fait-il précisément ?

Aller-retour relais-bouton

Commençons par le chemin suivi jusqu'au relais quand on presse le bouton. La **figure 11** illustre comment le message est emballé couche par couche, puis envoyé au courtier par Wi-Fi. L'instruction `client.publish(MQTT_TOPIC_OUT, (const uint8_t*)buffer, n, true)`; du code source s'en charge. Pourquoi cette fonction s'appelle-t-elle `publish` (publier) et non `send` (envoyer) ? Cela vient du mode ultérieur de distribution des données dans MQTT. Le courtier MQTT distribue les messages selon un **TOPIC** (sujet) ; ici, le sujet (`MQTT_TOPIC_OUT`) est « **BUTTON** ». En se connectant au courtier MQTT, le client (c'est-à-dire l'ESP32-C3 de notre relais) peut indiquer le sujet qui l'intéresse (c.-à-d. s'abonner à ce canal d'information). Chaque participant ayant informé le courtier MQTT qu'il est intéressé


par un sujet donné recevra les messages publiés sur ledit sujet. L'expéditeur, quant à lui, ne s'occupe pas de la distribution. Il ne fait qu'envoyer (« publier ») ses messages au courtier. L'ESP32-C3 du relais s'abonne au sujet **BUTTON** dans son code en utilisant `client.subscribe(MQTT_TOPIC_IN)` ; et ici `MQTT_TOPIC_IN` est « **BUTTON** ». Quand le bouton déclenche l'envoi d'un message, il arrive au courtier MQTT, puis l'ESP32-C3 du relais le reçoit et fait basculer ce relais.

Lorsque le relais change d'état, son contrôleur envoie au courtier MQTT un message dans lequel figure cet état (activé ou non) sous le sujet « **RELAIS** » (**RELAY**). L'ESP32-C3 du bouton, qui s'est abonné au sujet « **RELAIS** » auprès du courtier MQTT reçoit donc le message avec le nouvel état, comme illustré à la **figure 12**. La couleur de la LED passe alors au rouge ou au vert.

L'intérêt de cette structure est qu'un 2^e couple bouton-contrôleur peut être mis en réseau pour publier lui aussi des messages sous le sujet « **BUTTON** ». L'ESP32-C3 du relais répondra alors aux données de commande de l'un ou l'autre bouton et effectuera la commutation correspondante. Si vous souhaitez expérimenter davantage avec MQTT, regardez les autres projets d'Elektor utilisant MQTT pour véhiculer des données. Par ex. la station météo [13] et l'horloge à LED géante avec capteur de température externe [14]. Pour en savoir plus sur MQTT

et sur son utilisation dans l'envoi de données à des plateformes du nuage, consultez par ex. la série « Mon voyage dans le nuage » [15].

Une plateforme idéale pour l'IdO

L'IdO ça peut être simple ! Comme exposé ici, la technologie est assez mature, et vous pouvez rapidement développer des dispositifs IdO à l'aide d'outils conviviaux et de composants appropriés. Bien sûr, les applications peuvent être plus complexes qu'un bête bouton-poussoir et un relais. Du pilotage du chauffage domestique à la sonnette d'entrée, de nombreuses applications pratiques peuvent exploiter la connectivité IdO. L'ESP32-C3 est une carte puissante et peu coûteuse qui constitue une plateforme idéale pour donner vie à vos idées ! 

220017-04

Contributeurs

Idée et texte : Mathias Claußen

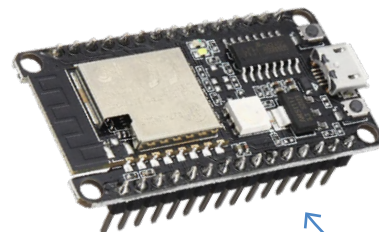
Rédaction : Jens Nickel

Traduction : Yves Georges

Mise en page : Giel Dols

Des questions, des commentaires ?

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



PRODUITS

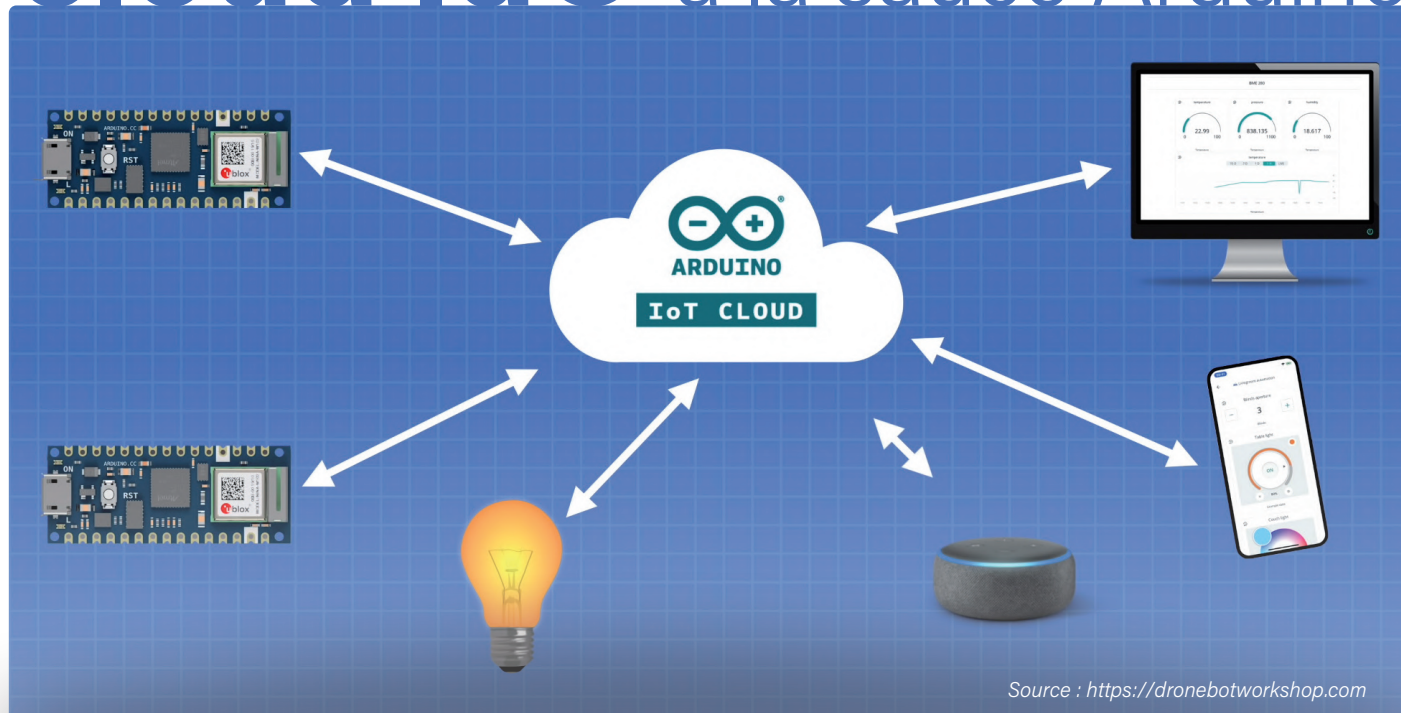
- > Carte de développement ESP-C3-12F avec 4 Mo de flash
www.elektor.fr/19855
- > Kit de capteurs Elektor 37-en-1
www.elektor.fr/16843
- > Kit Raspberry Pi Zero 2 W
www.elektor.fr/19952
- > Les indispensables du DIY, Pimoroni - Miniplaques d'essai et câbles de liaison
www.elektor.fr/18430



LIENS

- [1] « Prise en main du microcontrôleur ESP32-C3 RISC-V », M. Claußen, Elektor 01-02/2022 : www.elektormagazine.fr/210466-04
- [2] Documentation (en anglais) du kit de capteurs Elektor 37-en-1 : www.elektor.com/amfile/file/download/file/1170/product/6171/
- [3] Distribution Linux Ubuntu : <https://ubuntu.com/>
- [4] Procès-verbal du TC OASIS MQTT du 25/04/2013 : www.oasis-open.org/committees/download.php/49028/OASIS_MQTT_TC_minutes_25042013.pdf
- [5] JSON.org : www.json.org/json-fr.html
- [6] « Travailler avec JSON », Mozilla Web Docs : <https://developer.mozilla.org/fr/docs/Learn/JavaScript/Objects/JSON>
- [7] Node-RED : <https://nodered.org/>
- [8] Installation de Node-RED sur le Raspberry Pi : <https://nodered.org/docs/getting-started/raspberrypi>
- [9] Installation de Node-RED sur un PC : <https://nodered.org/docs/getting-started/local>
- [10] Téléchargement de l'EDI Arduino : www.arduino.cc/en/software
- [11] Instructions d'installation Espressif Arduino-ESP32 : <https://docs.espressif.com/projects/arduino-esp32/en/latest/installing.html>
- [12] Dépôt GitHub d'Elektor : https://github.com/ElektorLabs/220017-ESP32-C3-and-IdO_First-steps
- [13] « Station météo à ESP32 », R. Aarts, Elektor 01-02/2019 : www.elektormagazine.fr/180468-04
- [14] « Horloge à LED géante avec Wi-Fi et mesures météo », M. Claußen, Elektor 05-06/2019 : www.elektormagazine.fr/180254-04
- [15] « Mon voyage dans le nuage IoT », J. Nickel : www.elektormagazine.fr/search?query=mon+voyage+dans+le+nuage

cloud IdO à la sauce Arduino



Source : <https://dronebotworkshop.com>

Tam Hanna (Slovaquie)

L'Arduino IoT Cloud offre aux développeurs d'applications IdO une solution pratique de mise en œuvre d'un serveur du nuage sans les inconvénients de MQTT. Intéressé ? Allons voir !

De nombreuses applications à microcontrôleurs s'articulent aujourd'hui autour de l'Internet des Objets (IdO) où des informations sont diffusées via des services du nuage IdO et un courtier MQTT. Créer ce type d'application sous un environnement de développement local comme l'EDI Arduino traditionnel peut parfois s'avérer délicat. La plateforme Arduino IoT Cloud transpose l'EDI dans le nuage de sorte que votre navigateur devient une fenêtre de l'EDI. Nous avons essayé cet environnement en envoyant une valeur variable au nuage pour faire clignoter une LED sur notre banc d'essai. Puis nous avons tenté de le prendre en défaut. La base de tous les appareils IdO est bien sûr l'Objet. Dans l'environnement de développement Arduino IoT Cloud, l'Objet est un objet virtuel qui existe dans le cloud. Dans le monde réel, il s'agit d'un objet tel qu'un serveur, une carte à contrôleur ou un dispositif « intelligent » similaire [1]. Ici, votre Objet est construit dans le nuage avec un éditeur en ligne pour écrire un croquis qui décrit son comportement et ses réponses grâce à toute une série de variables.

Qui utilisera l'Arduino IoT Cloud ?

Avant de commencer, il est important de reconnaître que l'Arduino IoT Cloud n'est pas une alternative aux autres plateformes informa-

tiques dédiées au nuage, telles qu'Amazon AWS IoT Core, Microsoft IoT Hub ou Yandex IoT Core. Avec un grand nombre de dispositifs et de données à gérer, ces services bien établis du nuage IdO sont plus indiqués.

Lors de l'introduction de la dernière mouture de l'Arduino IoT Cloud, Massimo Banzi, CTO d'Arduino, a exprimé ses ambitions pour la plateforme en disant que : « Arduino propose désormais une plateforme complète avec la famille MKR et rationalise la création des nœuds IdO locaux et des dispositifs de périphérie. Ceux-ci utilisent une gamme d'options de connectivité et de compatibilité avec du matériel tiers, des passerelles et des systèmes du nuage. L'Arduino IoT Cloud permet aux utilisateurs de gérer, configurer et connecter, non seulement le matériel Arduino, mais aussi la grande majorité des appareils basés sur Linux – démocratisant véritablement le développement IdO ».

La prise en charge de la gamme de cartes Arduino MKR dédiées à l'IdO et de quelques autres cartes tierces répandues est un plus bienvenu et devrait inciter de nombreux développeurs de systèmes IdO à reconsidérer cette plateforme de développement accessible.

Configuration du matériel

Seul le forfait le plus rudimentaire des quatre versions possibles de l'Arduino IoT Cloud est gratuit. Vous pouvez consulter les différents forfaits et leurs caractéristiques dans la **figure 1** et choisir celui qui répond le mieux à vos besoins. La communauté des makers est surtout intéressée par la prise en charge par l'Arduino IoT Cloud de cartes tierces, telles que celles des fameuses familles ESP8266 et ESP32 (**tableau 1**), et d'autres kits (voir [2]). Les bibliothèques de

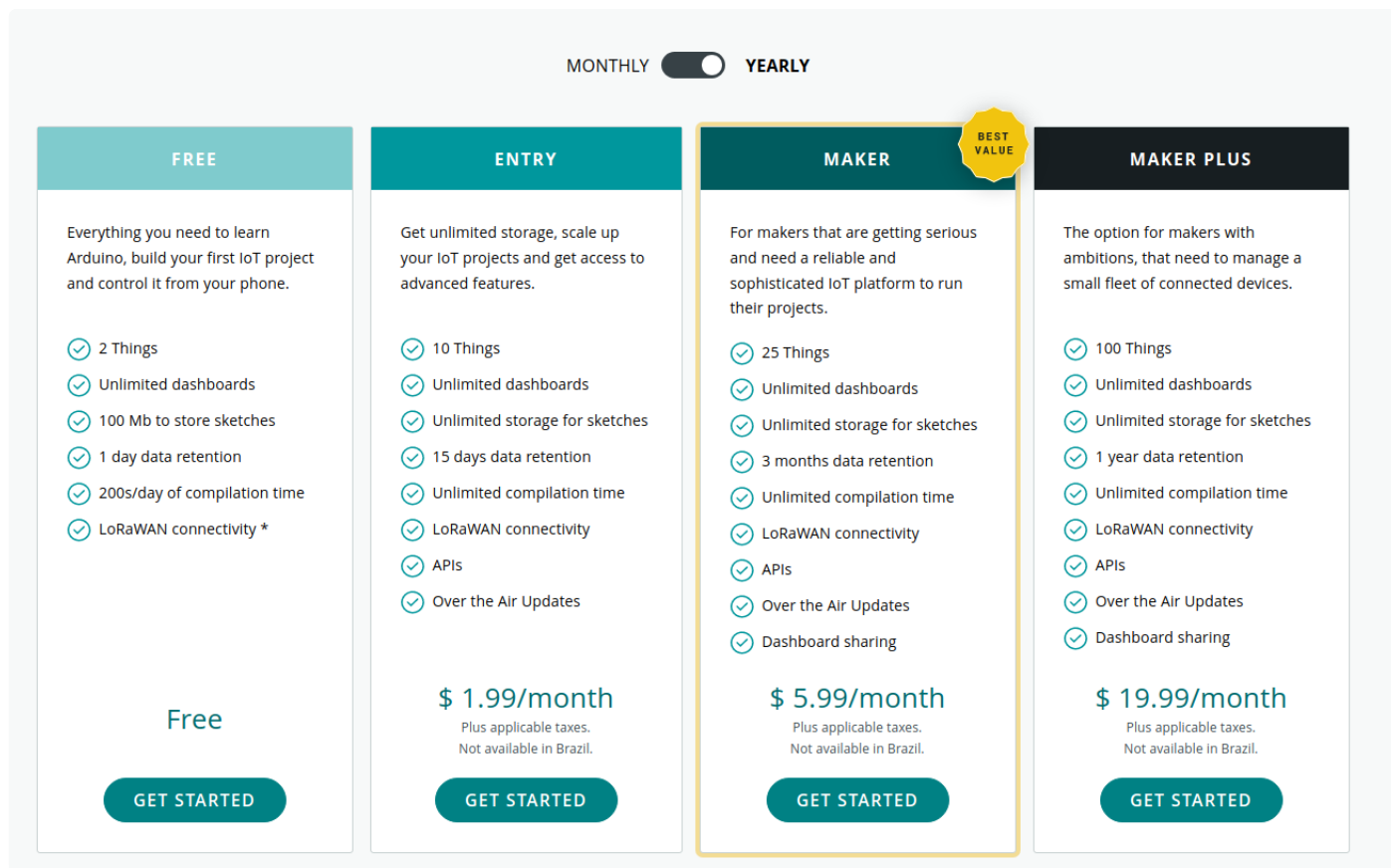


Figure 1. Quel est le meilleur forfait pour vous ? (Prix des forfaits en date du 20/01/2022).

pilotes permettent aussi à divers systèmes basés sur Linux d'envoyer et de récupérer des informations sur le nuage Arduino. Pour tester les fonctions de l'Arduino IoT Cloud, on utilise ici comme carte cible un module Arduino Nano RP2040 Connect. Cette carte est basée sur le microcontrôleur RP2040 de la fondation Raspberry Pi et comprend un module Wi-Fi u-blox. Avant de configurer le nuage Arduino, la carte est connectée à un ordinateur via un câble USB. Visitons d'abord le site web [3] pour créer un nouveau compte Arduino. Pour nos besoins, nous nous en tiendrons au forfait de base gratuit qui est explicitement destiné aux nouveaux venus dans le système. Commençons par cliquer sur le bouton *Create Thing* pour créer un nouvel objet.

L'aperçu de la **figure 2** montre alors la configuration des trois composants de base de l'environnement de développement. J'ai supprimé un précédent compte Arduino et tous les paramètres de configuration afin de commencer avec une configuration de périphérique vierge. Les étapes suivantes sont réalisées sur une machine sous Windows 10, mais le processus est identique sous Ubuntu (Linux), et j'ai constaté que la détection du matériel fonctionne en général mieux sous Unix.

Nous pouvons maintenant cliquer sur l'icône de raccourci dans la section *Device* et choisir *Set Up an Arduino Device*. Quelques secondes après avoir cliqué sur cette option, le serveur signalera l'absence du composant appelé *Arduino Create Agent*. Cliquez sur le bouton *Download* pour télécharger le logiciel et l'installer de la manière habituelle.

Notez que *Create Agent* est spécifique au navigateur : si vous l'installez sous Chrome, vous devrez le réinstaller. Vous devez tenir

Tableau 1.
Cartes prises en charge par l'Arduino IoT Cloud.

WLAN

- MKR 1000 WiFi
- MKR WiFi 1010
- Nano RP2040 Connect
- Nano 33 IoT
- Portenta H7

LoRaWAN

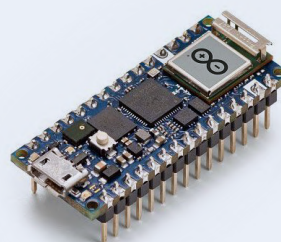
- MKR WAN 1300
- MKR WAN 1310

GSM/NB-IoT

- MKR GSM 1400
- MKR NB 1500

ESP32/ESP8266

- Large gamme de cartes tierces



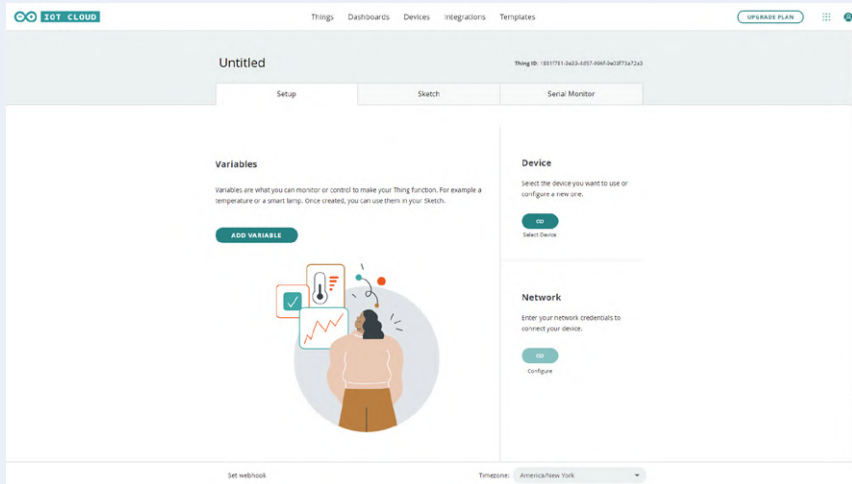


Figure 2. L'Arduino IoT Cloud guide le développeur pas à pas jusqu'à l'objectif.



Figure 3. Cet Arduino est connecté au nuage.

compte des avertissements du pare-feu qui s'affichent. Assurez-vous que vous autorisez l'accès aux réseaux privés et publics. *Arduino Create Agent* réside désormais dans votre barre de tâches – dans certains cas, il peut être nécessaire de le rappeler à partir du menu de démarrage, mais ainsi se termine l'installation du pilote.

Pour continuer, nous allons rafraîchir la vue jusqu'à ce que le nuage Arduino nous informe que notre carte Nano RP2040 Connect a été reconnue. Ensuite cliquez sur le bouton *Configure* pour lancer l'assistant de configuration – il vous demandera d'entrer un nom « convivial », puis il initialisera l'élément sécurisé du système cible avec le logiciel de communication de base.

Des problèmes surviennent parfois lors de l'affectation du réseau sous Windows. La méthode la plus fiable et la plus efficace est de le faire sous Linux. Par ailleurs, la section *Network* n'est pas automatiquement activée par Arduino IoT Cloud. Elle n'est disponible que lorsque vous créez l'une des variables destinées à l'échange de données.

Cliquons maintenant sur *Variables*, ce qui ouvre une fenêtre pour ajouter une nouvelle variable. Nous commençons par la nommer **ledIntenBool** en précisant *Boolean* dans le champ du type de données. Il est amusant de constater que l'Arduino IoT Cloud ne prend pas seulement en charge les unités de programmation en C, mais implémente également des encapsulations autour des variables du monde réel.

Si vous souhaitez vous limiter au C, nous vous recommandons de sélectionner l'option *Basic Types*. En théorie, nous pourrions alors définir des paramètres dans les champs *Variable Permission* et *Variable Update Policy*, mais les paramètres par défaut seront suffisants pour nos besoins, c'est pourquoi nous fermons maintenant la boîte de dialogue. En suivant, nous créons un champ de type *Integer Number*, que nous nommons **ledIntenInt**.

Préparation du code

Après avoir créé les variables, des points rouges dans l'onglet *Sketch* indiquent les changements apportés à la structure du programme. Il est maintenant possible de cliquer sur l'icône de raccourci dans la section *Network* pour saisir les paramètres Wi-Fi. Ma préférence est de saisir les valeurs à l'aide d'un outil en ligne de commande tel que *iwlist* sur une machine Linux, puis de les copier dans le presse-papiers.

Passons ensuite à l'onglet *Sketch* et cliquons sur le bouton *Verify and Upload*. L'Arduino IoT Cloud commence alors à compiler le code et l'envoi à la carte RP2040 connectée en utilisant l'*Arduino Cloud Agent*. Si le code compilé est chargé avec succès, le message « *Untitled_dec25a uploaded successfully on board Arduino Nano RP2040 Connect (/dev/ttyACM0)* » est affiché.

Après la nécessaire réinitialisation, le RP2040 va se connecter via son émetteur Wi-Fi. Après un certain temps et en appuyant plusieurs fois sur *F5*, l'appareil apparaîtra avec « *Status: online* » (fig. 3).

Si vous choisissez de vous abonner à l'un des forfaits Arduino IoT Cloud les plus complets, vous pouvez recevoir les mises à jour logicielles directement via le Wi-Fi ; mais pour nos expériences simples avec le forfait gratuit, il faut une connexion filaire.

Plus en détails

L'outil d'édition de base de l'onglet *Sketch* n'est pas très utile, mais en cliquant sur le bouton *Open full editor*, nous obtenons un EDI complet basé sur le nuage qui nous permet d'éditer de petits projets plus confortablement. Examinons d'abord le contenu du fichier *thingProperties.h*, qui contient les éléments structurels du croquis. Au début, nous voyons les déclarations suivantes qui fournissent les éléments requis pour l'accès Wi-Fi :

```
const char SSID[] = SECRET_SSID; // SSID du réseau (nom)
const char PASS[] = SECRET_PASS; // Mot de passe réseau
// (à utiliser pour WPA, ou comme clé pour WEP)
```

Arduino IoT Cloud se charge de saisir le nom et le mot de passe en utilisant les paramètres déjà saisis dans la section *Network*. Viennent ensuite les deux variables suivantes dont les noms devraient vous être familiers depuis leur déclaration dans la section *Variables* :

```
int ledIntenInt;
bool ledIntenBool;
```

Arduino IoT Cloud implémente les variables dans le *back-end* comme des variables C standard dotées de propriétés supplémentaires. Ces propriétés se trouvent, avec d'autres, dans la méthode *initProperties*, qui assure la mise en place des primitives et des structures nécessaires à la communication avec le nuage selon le schéma suivant :



```
void initProperties() {
  ArduinoCloud.setThingId(THING_ID);
  ArduinoCloud.addProperty(ledIntenInt, READWRITE,
    ON_CHANGE, onLedIntenIntChange);
  ArduinoCloud.addProperty(ledIntenBool, READWRITE,
    ON_CHANGE, onLedIntenBoolChange);
}
```

Il est intéressant de noter que la méthode `addProperty` se charge « d'enregistrer » l'attribut. Remarquez le passage des pointeurs de fonction `onLedIntenIntChange` et `onLedIntenBoolChange` – ils joueront un rôle important par la suite.

La fonction de pilotage de l'application est décrite dans le croquis, qui commence par l'inclusion de l'en-tête (non représenté ici). Elle est suivie par l'initialisation du croquis, qui se déroule selon le schéma suivant :

```
void setup() {
  Serial.begin(9600);
  delay(1500);

  initProperties();

  ArduinoCloud.begin(ArduinoIoTPreferredConnection);
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();
}
```

Du point de vue de l'environnement de programmation Arduino, l'Arduino IoT Cloud est un pilote matériel comme un autre. L'objet global `ArduinoCloud` expose un ensemble de fonctions que votre code utilise pour communiquer avec le pilote du nuage. L'appel à `setDebugMessageLevel`, qui définit la « verbosité » du pilote, est particulièrement important. Plus la valeur est élevée, plus le pilote du nuage émet d'informations de débogage sur le port série de la carte. Dans le cas de pilotes « compliqués », la question se pose toujours de savoir comment la puissance de calcul est allouée. Le squelette de projet créé pour nous par l'Arduino IoT Cloud peut répondre à cette question dans la méthode `loop`, qui se charge de l'allocation de la puissance de calcul selon le schéma suivant :

```
void loop() {
  ArduinoCloud.update();
}
```

L'Arduino IoT Cloud nous offre par défaut les trois méthodes d'écoute suivantes :

```
void onTestScheduleChange() {
}
void onLedIntenBoolChange() {
}
void onLedIntenIntChange() {
}
```

`onLedIntenBoolChange` et `onLedIntenIntChange` sont responsables des variables créées à distance dans le *back-end*, tandis que `onTestScheduleChange` aide à mettre en œuvre les fonctions « internes » du nuage Arduino.

Nous pouvons nous en occuper à l'étape suivante, en utilisant les capacités intégrées « intelligentes » de l'Arduino IoT Cloud. La carte Arduino utilisée ici est dotée d'une LED standard (rouge) sur la broche 13 et d'une LED RVB (canaux LEDR, LEDG et LEDB), qui peut être commandée par trois signaux PWM pour en modifier la couleur.

Revenons à la fonction `setup`, et initialisons les broches nécessaires :

```
void setup() {
  ...
  ArduinoCloud.printDebugInfo();

  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(LEDB, OUTPUT);
}
```

Les changements déclenchés dans le nuage activent l'écouteur d'évènement, qui écrit les valeurs entrantes sur le matériel :

```
void onLedIntenBoolChange() {
  digitalWrite(LED_BUILTIN, ledIntenBool);
}
void onLedIntenIntChange() {
  analogWrite(LEDB, ledIntenInt);
}
```

À ce stade, vous pouvez à nouveau transférer le croquis sur la carte. La LED RVB utilise des connexions à anode commune, de sorte que les couleurs individuelles sont contrôlées par la sortie d'un 0 pour allumer la LED correspondante. Les variables sont initialisées comme indiqué à la **figure 4**, pour que la diode bleue de la LED RVB s'allume après une initialisation réussie.

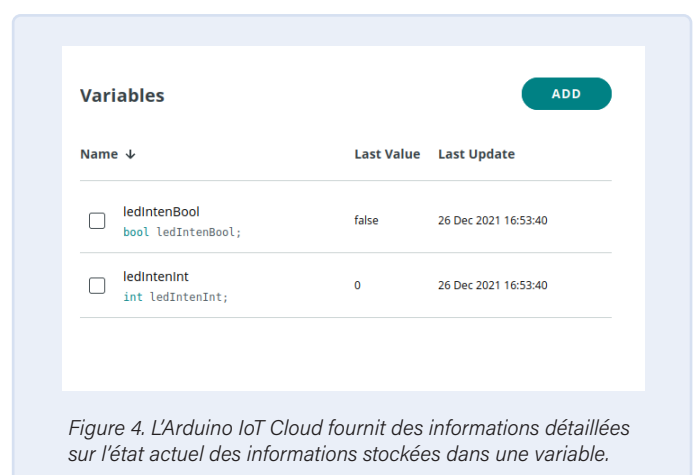


Figure 4. L'Arduino IoT Cloud fournit des informations détaillées sur l'état actuel des informations stockées dans une variable.

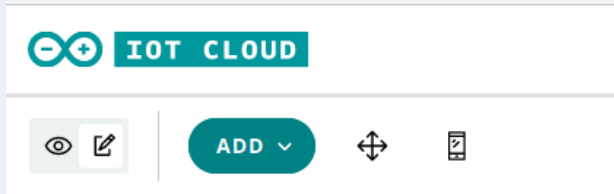


Figure 5. Passez le tableau de bord en mode édition en cliquant sur l'icône crayon en haut à gauche.

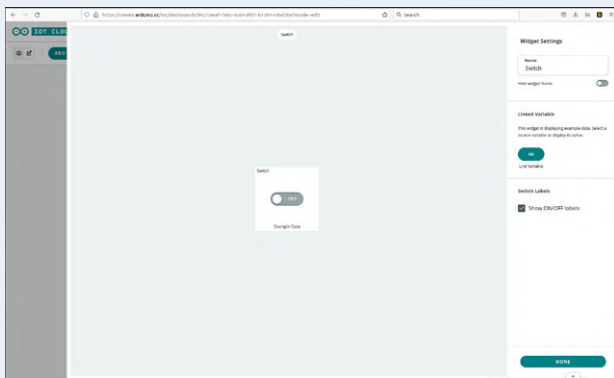


Figure 6. L'interface d'édition des commandes est « modale ».

Add variable

Name
TamsSchedule

Sync with other Things

Schedule eg. Every MON at 8:00 AM

Declaration
CloudSchedule tamsSchedule ;

Variable Permission

☒ Read & Write ☐ Read Only

Variable Update Policy

☒ On change ☐ Periodically

ADD VARIABLE CANCEL

Figure 7. CloudScheduler est un type de données comme tous les autres.

Modification du contenu des variables

Si nous revenons au panneau principal d'Arduino IoT Cloud, nous pouvons cliquer sur l'onglet *Dashboards*, qui, s'il s'agit d'un nouveau compte, vous invitera à créer un nouveau tableau de bord. Cliquez maintenant sur le bouton *Build Dashboard* pour lancer l'éditeur, ce qui prendra un bon moment même si vous avez une connexion internet rapide.

Pour modifier et ajouter des éléments au tableau de bord, cliquez sur l'icône d'édition (le crayon) en haut à gauche de l'écran (fig. 5). Une fois ce mode sélectionné, le bouton bleu **ADD** apparaît, que vous pouvez utiliser pour afficher la liste déroulante des widgets disponibles. Nous sélectionnons d'abord un widget *Switch*, qui apparaît alors dans l'interface d'édition présentée dans la figure 6. On a maintenant à droite de l'écran un champ *Linked Variable* avec un bouton de liaison. Cliquez dessus pour activer une liste de tous les objets et variables contenus dans le compte du nuage. Ici, nous pouvons choisir la variable `ledIntenBool` et la lier en utilisant le bouton *Link Variable*. L'état de `ledIntenBool` sera maintenant associé à l'état de l'interrupteur. Un clic sur le bouton *DONE* ferme l'interface d'édition et l'interrupteur est maintenant incorporé dans le tableau de bord. Puis nous pouvons cliquer sur l'icône en forme d'œil pour relâcher l'interrupteur et l'activer. La manœuvre de l'interrupteur commande maintenant la LED rouge située à côté de la prise micro-USB.

Afin de pouvoir régler la luminosité de la LED bleue, il faut remettre l'éditeur du tableau de bord en mode édition et ajouter une nouvelle commande avec *Add -> Widgets*. Cette fois, j'ai choisi le type *Slider*. Dans son interface d'édition, nous définissons sa *Value range* de 0 à 255. Le lien est fait par la variable `letIntenInt`, qui représente la « commande de luminosité » de la LED RVB. Nous repassons enfin en mode d'activation et constatons que les changements de position du curseur affectent désormais la luminosité de la LED bleue.

Utilisation du planificateur

Au moment où j'écris cet article, une nouvelle fonction permet de configurer des tâches planifiées ou *web cron-jobs* : elle utilise un type de variable appelé *CloudSchedule* que vous pouvez définir pour qu'elle soit vraie ou fausse à un moment précis et pour une durée donnée. Il n'est pas nécessaire d'invoquer une fonction de minuterie, car cette variable est définie ou réinitialisée automatiquement dans l'Arduino IoT Cloud, selon la façon dont vous la configurez. Les tâches peuvent ensuite être déclenchées en vérifiant l'état de cette variable. Pour en démontrer les possibilités, créons une nouvelle variable de type *Schedule*. La figure 7 montre la configuration souhaitée. Nous pouvons maintenant voir le nouveau type de variable dans le code :

```
CloudSchedule tamsSchedule;
```

Les paramètres temporels de cette variable sont configurés via le tableau de bord. La page de configuration présentée dans la figure 8 montre les paramètres de réglage à fournir.

À l'étape suivante, nous devons nous occuper du traitement « local » des valeurs contenues dans `tamsSchedule` :

Figure 8. Le planificateur est configuré à l'aide du tableau de bord.

```
void setup() {
  ...

  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(LED_B, OUTPUT);
  pinMode(LED_R, OUTPUT);
}

void loop() {
  ArduinoCloud.update();
  // Your code here
  if(tamsSchedule.isActive()){
    digitalWrite(LED_R, HIGH);
  }
  else{
    digitalWrite(LED_R, LOW);
  }
}
```

Il est important de noter ici que la « rédaction » des informations fournies dans `tamsSchedule` est à la charge exclusive du développeur. Le nuage se limite simplement à mettre à jour périodiquement la valeur contenue dans `tamsSchedule`. La procédure d'interrogation continue présentée ici dans la structure `loop` n'est peut-être pas optimale du point de vue des ressources, mais elle fonctionne sans problème. Le programme peut maintenant être envoyé à l'Arduino où vous pouvez observer les clignotements rouges périodiques provenant de la LED RVB.

À ce stade, je n'ai pas pu résister à un autre petit test du système pour voir ce qui se passe lorsque la liaison RF tombe en panne. Le « lab Wi-Fi » a été éteint et l'Arduino a commencé à agir de manière erratique en commutant de manière aléatoire l'élément bleu de la LED RVB ainsi que la LED sur la broche 13 pour finalement effectuer un redémarrage complet après quelques secondes.

À la mise sous presse de cet article, je n'ai pas vraiment compris comment précisément l'Arduino IoT Cloud se remettait de la perte de la liaison radio entre l'appareil final et le serveur.

Une option pratique

Il est clair que l'Arduino IoT Cloud est encore en cours de développement et d'amélioration. Il a cependant un grand potentiel et offre au développeur d'applications IdO une option pratique et accessible pour mettre en œuvre un *back-end* dans le nuage, sans les inconvénients de MQTT et Cie. En dépit du petit couac, je recommande vivement ce produit !

210550-04

Contributeurs

Idée, illustrations et texte : **Tam Hanna**

Rédaction : **Rolf Gerstendorf**

Traduction : **Denis Lafourcade**

Mise en page : **Harmen Heida**

Des questions, des commentaires ?

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

LIENS

- [1] Jumeau numérique : https://en.wikipedia.org/wiki/Digital_twin
- [2] Cartes prises en charge par l'Arduino IoT Cloud : <https://bit.ly/3t8Vl3W>
- [3] Objets Arduino : <https://create.arduino.cc/iot/things>



PRODUITS

- > **Arduino MKR WiFi 1010**
www.elektor.fr/19935
- > **Arduino Nano RP2040 connect**
www.elektor.fr/19754
- > **Arduino Nano 33 IoT**
www.elektor.fr/19937

détecteur Geiger-Müller à double tube (extension pour Arduino)

Capteur très sensible pour très faible rayonnement



Gabriele Gorla (Italie/USA)

Avec deux tubes, ce *shield* Geiger-Müller a une sensibilité accrue : votre Arduino Uno devient un appareil de mesure et d'enregistrement des radiations nucléaires. Ajoutez-y un *shield* Dragino LoRa/GPS et détectez les radiations ambiantes sur le terrain. Les données seront accessibles où que vous soyez !

La radioactivité et les capteurs de radiations m'ont toujours fasciné. Les tubes Geiger-Müller [1] sont des capteurs répandus et assez bon marché. Mon projet « GRAD » est un compteur de radiations complet au format Arduino. Il se caractérise par deux tubes (pour augmenter la sensibilité) et une très faible consommation.

Introduction aux compteurs Geiger-Müller

Un compteur Geiger-Müller comporte quatre blocs fonctionnels essentiels.

- **Tube Geiger-Müller.** Rempli d'un mélange gazeux à basse pression, il a deux électrodes. Polarisé avec la tension appropriée, le gaz s'ionise et conduit brièvement l'électricité à chaque impact de radiation. Selon son type, il est peut détecter des particules alpha, bêta et gamma. Le GRAD utilise des SBM-20 sensibles aux rayons gamma et bêta à haute énergie.
- **Alimentation à haute tension.** Le tube doit fonctionner dans ce qu'on appelle le plateau de Geiger. Il s'agit d'une région où le nombre d'impulsions est quasi indépendant de la polarisation HT.

Pour les tubes courants, elle se situe entre 400 et 500 V. Le point optimal du SBM-20 se situe vers 400 V.

- **Détecteur.** En sortie de tube, les impulsions sont très courtes et de tension variable. Le détecteur met le signal en forme pour faciliter le comptage.
- **Compteur.** Il compte les impulsions sur une durée fixe pour calculer une valeur en CPS (coups par seconde) ou CPM (coups par minute). Le facteur donné sur la fiche technique du tube sert à convertir cette valeur en débit de dose approximatif. La carte Arduino est chargée du comptage, de l'affichage et de l'enregistrement.

Alimentation

Sur la toile, on trouve de nombreux circuits d'alimentation de tubes Geiger-Müller. Beaucoup sont des convertisseurs élévateurs autour du *timer 555*, avec ou sans rétroaction. Je n'ai pas retenu de schéma à boucle ouverte parce que ça implique le réglage de chaque carte et que la tension à nombre élevé d'impulsions est instable. Les schémas à boucle fermée ont une meilleure stabilité. Cependant, pour que la consommation d'énergie globale soit faible, la boucle de rétroaction (12 μ A de courant à 400 V, soit ~5 mW) doit être bien étudiée.

Pour limiter la puissance perdue par la rétroaction à haute tension, le plus élégant serait d'utiliser un transformateur élévateur non isolé, par ex. le *LT3420* d'Analog Devices qui détecte la tension au primaire.

Cependant, nous avons conçu un convertisseur élévateur à découpage très simple à faible courant de rétroaction. La **figure 1** l'illustre ; il est fortement inspiré des adaptateurs Theremino Geiger [2]. Les versions CMS, DIY et « Pierrafeu » sont mélangées ici. La carte n'a que des composants traversants. Des diodes Zener font la rétroaction.

L'inverseur à trigger de Schmitt U1C, R4 et C5 forment un oscillateur. Mis en parallèle, U1A, U1B et U1D augmentent le courant de base de l'interrupteur principal Q2.

La rétroaction utilise des diodes Zener à très faible courant de fuite

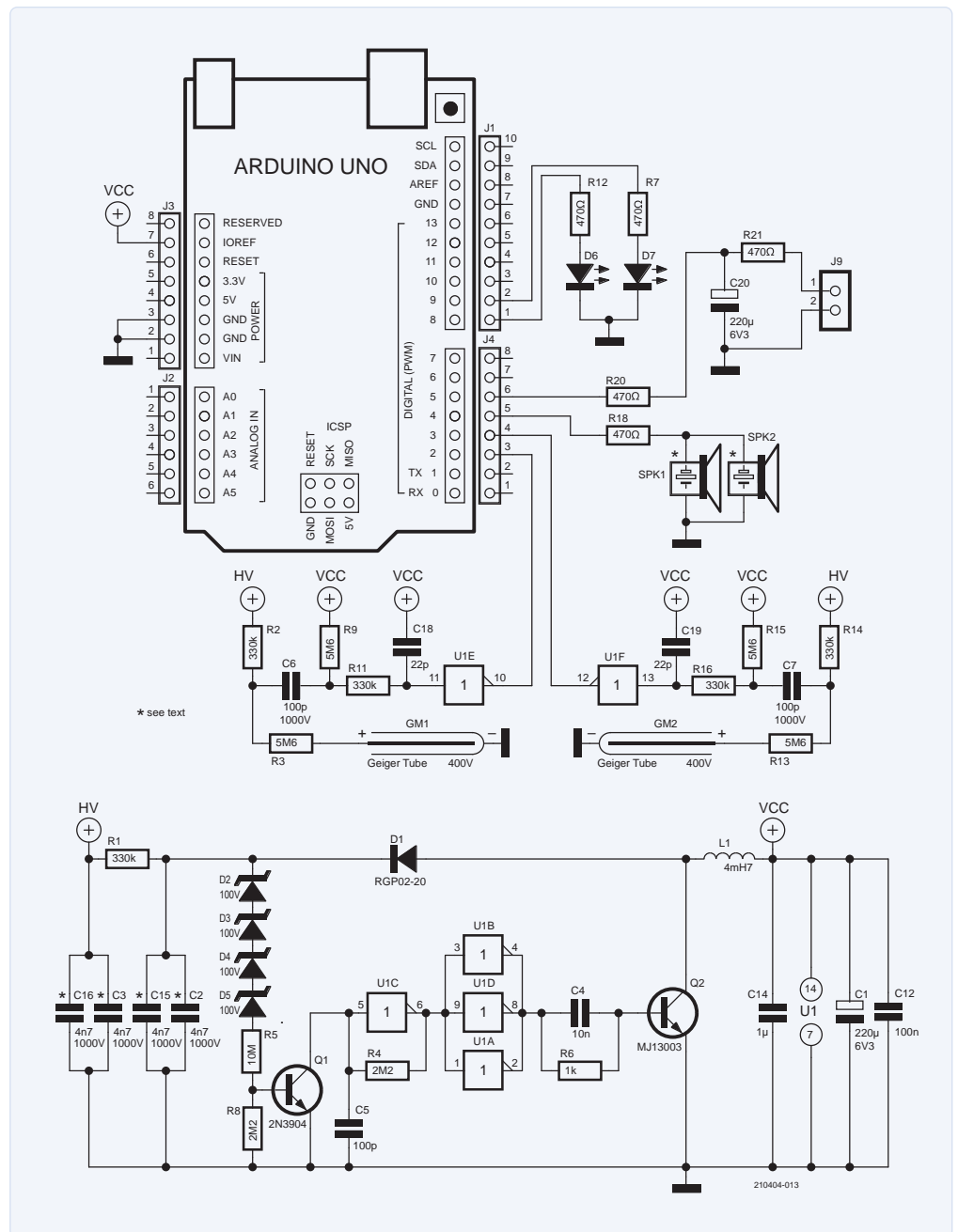


Figure 1. Schéma de principe du projet GRAD03 [3].

(D2 à D5) en série. Dès que la tension de sortie dépasse la tension Zener, Q1 conduit et stoppe l'oscillateur. Dès que la tension de sortie redescend, Q1 se coupe et libère l'oscillateur. R1 et C3 achèvent d'atténuer l'ondulation de sortie. Les résistances série (R2 et R3 pour GM1, R13 et R14 pour GM2) alimentent chaque tube séparément.

Comptage

Côté HT des tubes, un condensateur (C6 resp. C7) bloque le courant continu et laisse passer les impulsions qui sont mises en forme par les inverseurs à trigger de Schmitt U1E/U1F (voir **fig. 2**).

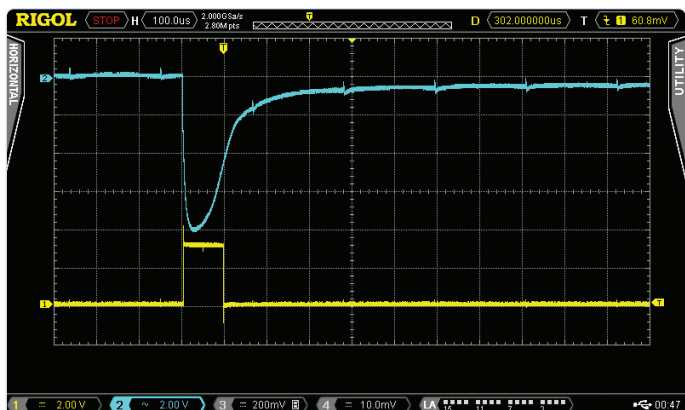


Figure 2. Oscillogramme d'une radiation ionisante frappant le tube et impulsion mise en forme.

J1 à J4 sont les connecteurs standard d'Arduino, GRAD n'en utilise que quelques broches. Les entrées numériques D2 et D3 furent choisies pour leur capacité d'interruption. Cela permet à l'Arduino de compter les impulsions en arrière-plan tout en réalisant d'autres tâches. Les sorties numériques D8 et D9 de l'Arduino pilotent une paire de LED signalant les rayons ionisants frappant les tubes respectifs. La broche D4 est reliée au haut-parleur piézoélectrique SPK1 qui fournit un retour sonore.

Enfin, la sortie numérique D5 est connectée à un filtre passe-bas, formé par R20, R21 et C20, qui pilote un galvanomètre analogique de 10 mA (J9).

CAO des circuits imprimés

Le schéma et le circuit imprimé (fig. 3) ont été conçus avec KiCad. Les fichiers de CAO sont téléchargeables depuis la page *Elektor Labs* de ce projet [3], rubrique *Elements*. Les fichiers Gerber et de perçage de la carte s'y trouvent également. Utilisez-les pour la commander à votre fournisseur préféré. Enfin, vous y trouverez une feuille Excel de nomenclature très détaillée, en particulier pour les composants les plus critiques (HT !) de ce projet, avec les coordonnées, les références et même les codes des fabricants.

La figure 4 montre la carte montée, tubes SBM-20 en place. Notez que les paires de composants SPK1/SPK2, C3/C16, et C2/C15 offrent deux empreintes alternatives sur la carte. En outre, un seul des composants de chacune de ces paires sera installé. J5 et J6 offrent aussi deux empreintes alternatives pour connecter les bornes négatives des tubes Geiger plus courts.

Options de tubes

La carte est conçue pour fonctionner avec le SBM-20 soviétique



LISTE DES COMPOSANTS

Résistances

R16 = 330 kΩ
R15 = 5,6 MΩ
R8 = 2,2 MΩ
R5 = 10 MΩ
R6 = 1 kΩ
R21 = 470 Ω

Condensateurs

C20 = 220 μF, 6,3 V
C3 = 4700 pF, 1000 V
C4 = 10 nF, 6,3 V
C5 = 100 pF, 50 V
C7 = 100 pF, 1000 V
C12 = 100 nF, 50 V
C14 = 1 μF, 6,3 V
C16 = 4700 pF, 1000 V
C19 = 22 pF, 50 V

Inductances

L1 = 4,7 mH

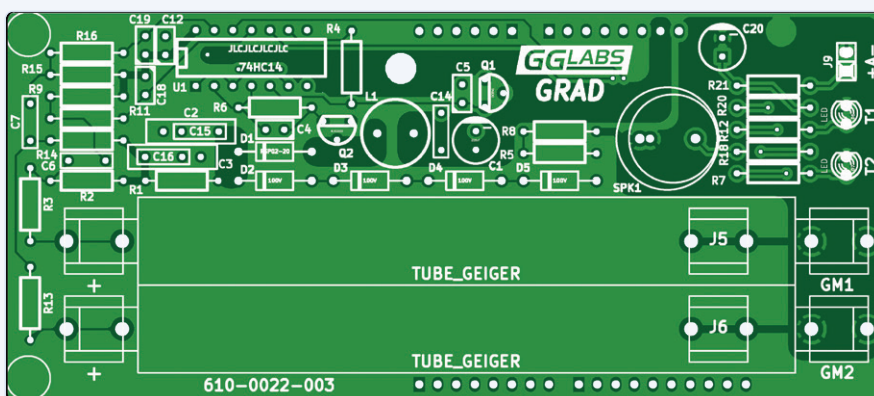


Figure 3. Circuit imprimé nu.

Semi-conducteurs

D1 = diode 1 A, 800 V
D5 = diode Zener 100 V 1,5 W
D7 = LED rouge 3 mm (marquées T1 et T2 sur le circuit imprimé)
Q1 = 2N3904
Q2 = MJ13003
U1 = 74HC14

Divers

GM2 = tube Geiger-Müller, par ex. SBM-20
J4 = jeu de connecteurs du shield Arduino Uno
SPK1 = buzzer piézoélectrique AC, par ex. AC-1205G-N1LF

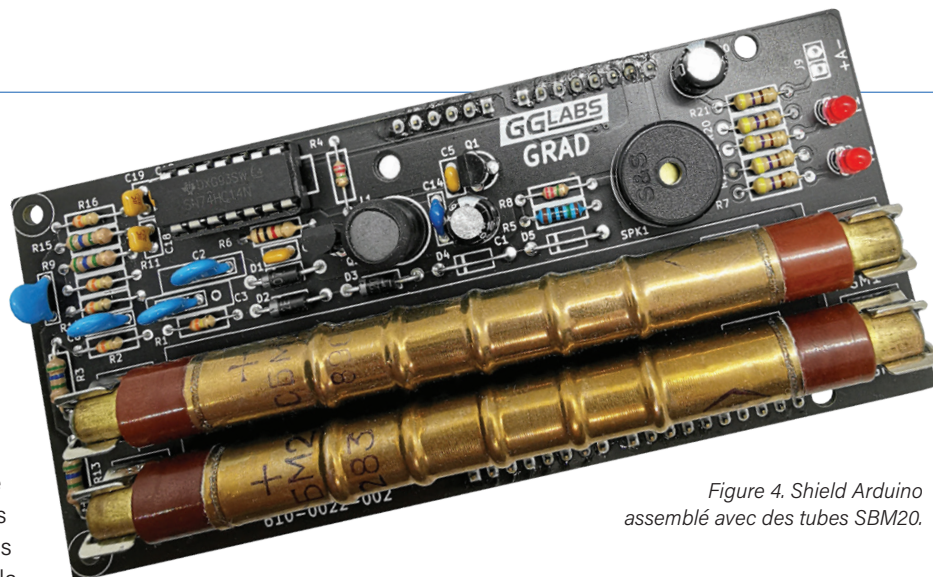


Figure 4. Shield Arduino assemblé avec des tubes SBM20.

(105 mm), le STS-5 et le J305 chinois ou les J305 et M4011 de 90 mm. Tout autre tube 400 V fonctionnera également avec une adaptation du montage. Dans ce cas, il faudra prendre soin de garder le fil positif aussi court que possible pour réduire la capacité parasite. Pour les tubes nécessitant une HT différente, la ou les diodes Zener doivent être remplacées pour obtenir la tension requise. Toute diode ayant une fuite $\leq 0,5 \mu\text{A}$ devrait fonctionner.

Performance

L'alimentation consomme $325 \mu\text{W}$ ($65 \mu\text{A}$) sous 5 V. En fonctionnement sur batterie, la carte peut aussi être alimentée en 3,3 V. À cette tension plus faible, la consommation tombe à $150 \mu\text{W}$ ($45 \mu\text{A}$).

Exemple de code Arduino

Pour piloter le compteur, un croquis Arduino simple est disponible sous la rubrique *Elements* de la page du projet Elektor Labs. Toutes les 60 s, le logiciel émet via le port série une ligne de valeurs séparées par des virgules (CSV) : numéro séquentiel, comptage brut de chaque tube, moyenne mobile de `cnt1 + cnt2` et débit de dose en $\mu\text{Sv/h}$ déduit du comptage moyen. L'extrait de sortie ci-après montre le début d'une mesure de radiation de fond effectuée à Santa Clara (Californie) avec un GRAD doté de deux tubes SBM-20.

```
Seq , cnt1 , cnt2 , avg10 ,  $\mu\text{Sv/h}$ 
1 , 14 , 19 , 33.0 , 0.075
2 , 10 , 17 , 32.3 , 0.073
3 , 16 , 12 , 31.8 , 0.072
4 , 16 , 24 , 32.5 , 0.074
5 , 13 , 11 , 31.6 , 0.072
```

Le graphe de la **figure 5** présente la mesure complète sur 15 heures. Au début figure une série de définitions modifiables en fonction de la configuration de la carte et des préférences de l'utilisateur. La 1^{ère} série configure les paramètres du tube Geiger-Müller et la taille de la fenêtre mobile pour le suivi.

```
#define CPM2USV 220 // nbre de CPM pour 1  $\mu\text{Sv/h}$ 
// du tube
#define TUBES 2 // nbre de tubes installés
#define WSIZE 10 // fenêtre de la moyenne mobile
// (en min)
```

`CPM2USV` est le comptage/min pour 1 $\mu\text{Sv/h}$ du tube utilisé. Malheureusement, aucun chiffre « correct » n'est publié pour ce dernier.

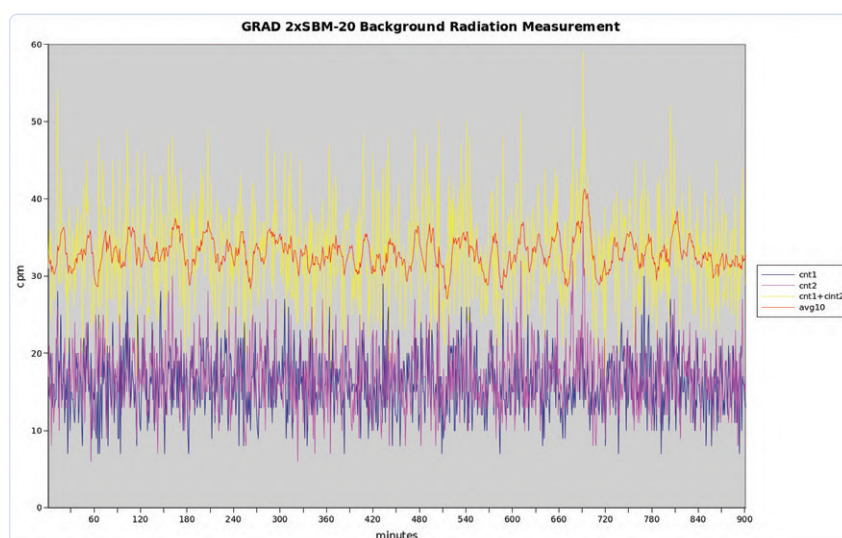


Figure 5. Rayonnement de fond à Santa Clara (Californie).

Pour le SBM-20, les amateurs du web utilisent des valeurs allant de 130 à 220 (facteur de conversion de 0,0075 à 0,0045).

`TUBES`, comme son nom l'indique, définit le nombre de tubes installés. Les valeurs permises sont 1 et 2. Enfin, `WSIZE` définit la taille de la fenêtre de moyenne mobile pour les comptages. 10 (val. par défaut) définit une fenêtre de 10 min.

La 2^e série de paramètres définit les attributs des fonctions matérielles optionnelles :

```
#define LED1_PIN 8 // broche de la LED du TUBE1
#define LED2_PIN 9 // broche de la LED de TUBE2
#define SPKR_PIN 4 // broche du haut-parleur
#define LED_BLINK_MS 20 // durée du clignotement de la
// LED pour chaque compte
```

Les trois `...PIN` définissent la broche Arduino à laquelle les LED et le HP sont connectés. Le `LED_BLINK_MS` définit le temps d'allumage de la LED pour chaque impulsion que le tube reçoit.

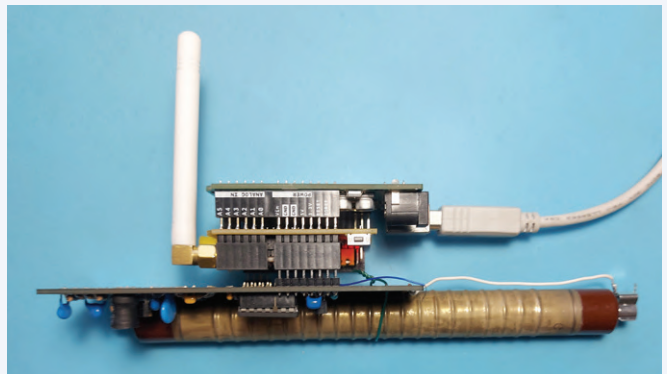
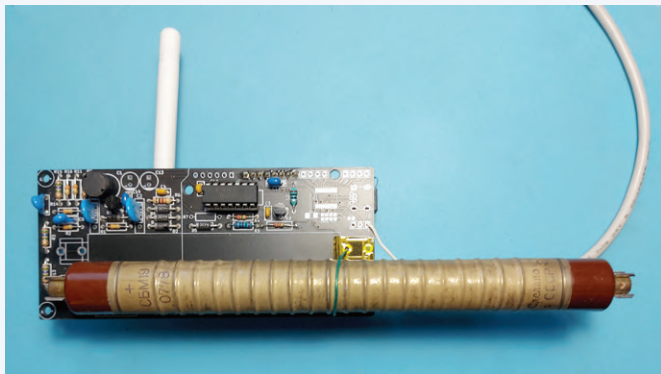


Figure 6. Carte GRAD03 et shield Dragino LoRa/GPS combinés sur l'Arduino Uno.

Mise en réseau LoRaWAN

Le croquis de base implique la connexion par USB de l'unité à un ordinateur. Cependant ce capteur a pour vocation la mesure de terrain donc loin de l'ordinateur, sans connexion USB.

On peut y ajouter un shield Dragino LoRa/GPS et bénéficier de sa connectivité. Malheureusement, le shield Dragino utilise aussi la broche D2 de l'Arduino, il faut donc pouvoir choisir entre un tube et deux tubes.

Option monotube : dans ce cas, seul le tube GM2 peut être utilisé et la broche 10 du trigger de Schmitt U1 doit être supprimée pour éviter tout parasitage de la communication LoRa. La **figure 6** montre l'empilement avec un seul SBM-19 soviétique.

Option bi-tube : dans ce cas, il faut modifier le shield Dragino et la carte GRAD. Pour le shield Dragino, R5 et J_DIO0 doivent être retirés. Reliez par un fil la broche radio DIO0 à D7 de l'Arduino.

Pour la carte GRAD, il faut enlever R7, R12, C20 et les deux LED. R21 doit être court-circuitée, et une seule LED doit être placée à l'emplacement J9 (remplace la sortie analogique du compteur).

Un croquis téléchargeable de liaison avec le réseau *The Things Network* (TTN) poste le comptage du tube toutes les 60 s. Il utilise la bibliothèque LMIC Arduino pour piloter le shield Dragino LoRa/GPS et se connecte au réseau TTN en utilisant l'activation par voie aérienne (OTAA). Un simple flux Node-RED est utilisé pour afficher les données (voir **fig. 7**).

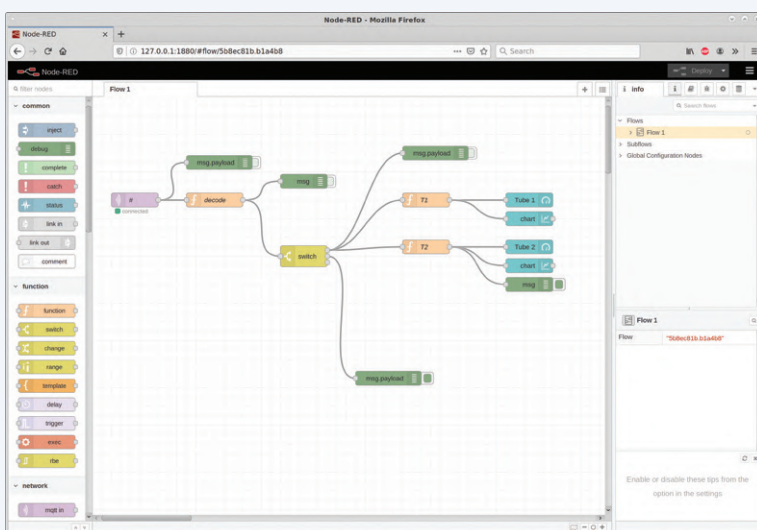
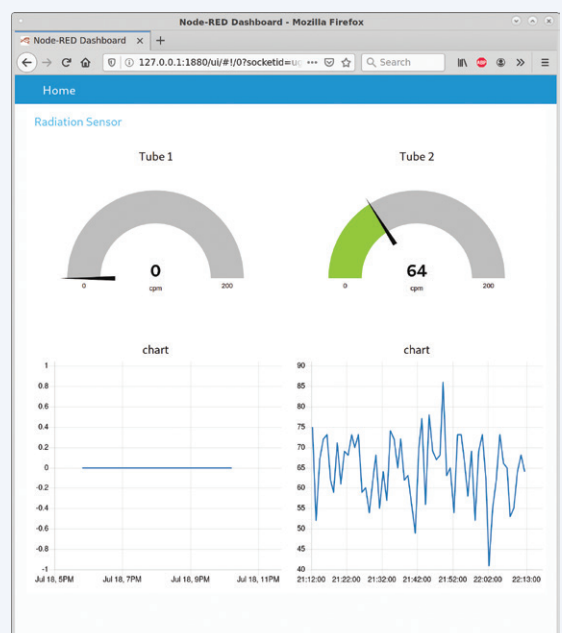


Figure 7. Flux et tableau de bord Node-Red.



Mesure de radiation

Pour vérifier le bon fonctionnement du compteur Geiger, une source de radiation est nécessaire. On peut en acheter en ligne chez des fournisseurs spécialisés. Il est aussi possible de trouver en friperie ou sur Ebay des articles *vintage* contenant une petite quantité de matériau radioactif. Parmi les objets radioactifs les plus courants, citons : objets en verre à l'uranium, manchons de lanterne au thorium et certaines couleurs d'objets de fête vintage. Alternativement, on peut mesurer le produit de désintégration du radon capturé par le filtre à air d'un climatiseur ou d'un purificateur d'air (**fig. 8**). Ces produits ont une demi-vie assez courte (qq. dizaines de minutes). Faites fonctionner la climatisation pendant 2 ou 3 h, puis mesurez immédiatement le filtre. Si le filtre est assez fin, il émettra un taux de radiation plusieurs fois supérieur au bruit de fond de votre région. Le graphe ci-après montre que le taux atteint presque dix fois le bruit de fond et décroît ensuite exponentiellement, caractérisant la désintégration radioactive. ◀

210404-04

Cet article est basé sur le matériel présenté sur la page Elektor Labs de ce projet [3]. Vous y trouverez tous les téléchargements pour GRAD03, ainsi que des discussions et des remarques sur ce sujet.

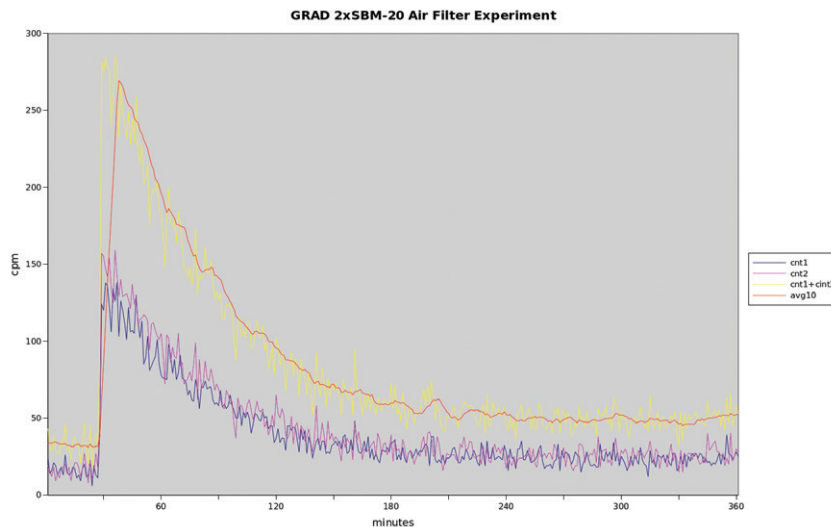


Figure 8. Rayonnement de désintégration du radon du filtre d'un climatiseur.

Des questions, des commentaires ?

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

Contributeurs

Conception : Gabriele Gorla
Texte : Gabriele Gorla, Luc Lemmens
Illustrations : Gabriele Gorla, Patrick Wielders
Rédaction : Jens Nickel, C. J. Abate
Mise en page : Giel Dols
Traduction : Yves Georges

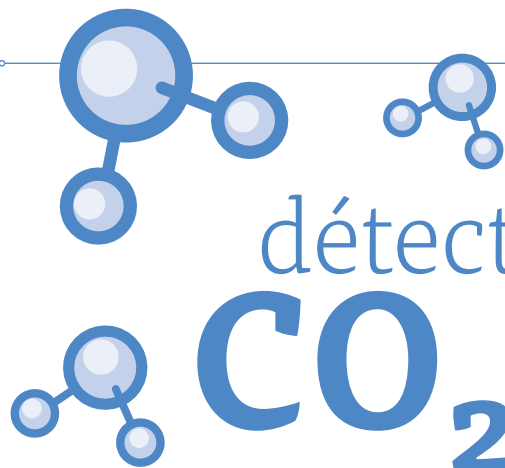


PRODUITS

- **Arduino Uno SMD Rev3**
www.elektor.fr/19938
- **Kit compteur Geiger MightyOhm (avec boîtier)**
www.elektor.fr/18509

LIENS

- [1] Tubes Geiger-Müller : https://france2.wiki/wiki/Geiger%E2%80%93M%C3%BCller_tube
- [2] Adaptations Theremino Geiger (voir les trois derniers modèles de cette page) : www.theremino.com/en/technical/schematics
- [3] Ce projet sur Elektor Labs : www.elektormagazine.fr/labs/dual-geiger-muller-tube-radiation-sensor-for-arduino



détecteur de CO₂ : CO₂ Guard

Une solution « maison »
pour surveiller la qualité de l'air

Mathias Claußen, Ton Giesberts et Luc Lemmens (Elektor)
Conception Florian Schäffer (Allemagne)

Les concentrations élevées de CO₂ peuvent causer des problèmes dans les locaux mal aérés. Ce détecteur de CO₂ (CO₂ Guard) offre une protection dans ce domaine. Construit avec des composants à trous traversants, il avertit les utilisateurs à l'aide de trois LED et d'une alarme sonore lorsqu'il est temps de renouveler l'air. Les fichiers Gerber du circuit imprimé et les fichiers de conception de la face avant du boîtier peuvent être téléchargés gratuitement.



Le CO₂ Guard est un projet à réaliser vous-même pour la surveillance et l'évaluation de la qualité de l'air intérieur en mesurant la concentration de dioxyde de carbone (Indoor Air Quality, IAQ). Le dispositif est basé sur un capteur infrarouge non dispersif (NDIR). Il indique la qualité de l'air selon cinq plages, sur une échelle composée de trois LED. En outre, une alarme retentit si la concentration en CO₂ est trop élevée. Il est possible de transférer les valeurs mesurées par Wi-Fi (valeur de CO₂ et température) vers la plateforme d'analyse Ido ThingSpeak et d'analyser les données graphiquement.

La qualité de l'air dépend de nombreux facteurs, notamment la concentration en CO₂ dans une pièce, qui est ressentie presque immédiatement. Un taux trop élevé de ce gaz provoque de la fatigue, une baisse de la concentration, voire des nausées, sans parler des effets négatifs durables causés par son excès dans votre environnement. Ce dernier cas est (espérons-le) peu susceptible de se produire dans une maison ou un bureau, mais le détecteur CO₂ Guard peut vous alerter au bon moment de la nécessité de ventiler votre local.

Circuit

Le cœur du schéma illustré à la **figure 1** est le circuit MOD2, Wemos D1 Mini, qui est un module Wi-Fi basé sur le SoC ESP8266EX avec une mémoire de programme flash de 4 Mo. Il collecte les données d'un capteur de CO₂ MH-Z19C (MOD1) via un logiciel UART, pilote trois LED pour indiquer les niveaux de CO₂ (LED 1... 3), et contrôle le buzzer BZ1, qui sert d'alarme sonore lorsque le niveau de CO₂ est excessif. Le CO₂ Guard est alimenté via le connecteur micro-USB du module Wemos, par un adaptateur secteur standard de 5 V CC. Cette tension d'alimentation est acheminée vers la broche 9 du MOD2 via une diode et un fusible internes, et vers un régulateur de tension 3,3 V embarqué. La sortie de ce régulateur à faible chute de tension (LDO) est utilisée pour l'alimentation interne du module Wemos lui-même et est également connectée à la broche 8 du module. La tension de 3,3 V peut être utilisée pour alimenter un petit ventilateur connecté à K1, comme nous le verrons plus tard. La sortie 5 V du circuit MOD2 sert à alimenter le capteur de CO₂ en plus d'une LED à l'intérieur du commutateur S1 servant à indiquer que la connexion Wi-Fi est active. Elle peut



également être connectée à K1 (via R6) comme source d'alimentation alternative pour le ventilateur. Le bouton-poussoir S1 peut déclencher le capteur de CO₂ pour lancer l'étalonnage automatique.

Alimenter le ventilateur

Dans ce projet, le ventilateur n'est pas utilisé pour le refroidissement, mais pour assurer un flux d'air autour du capteur. Bien qu'il semble évident d'alimenter un ventilateur de 5 V à sa tension nominale, si le débit d'air maximal n'est pas le critère de conception le plus important — comme dans ce projet — il est recommandé de minimiser le bruit de cet appareil en réduisant le débit d'air via la tension d'alimentation. Avec notre CO₂ Guard, il existe quelques options à cet égard.

Le rail d'alimentation de 3,3 V du module Wemos est régulé par un régulateur LDO dans un boîtier SOT23-5. Le courant de sortie maximal de ce régulateur est limité à 150 mA grâce à son boîtier. L'ESP8266 du module Wemos D1 Mini lui-même utilise une partie importante du courant de sortie. Une surcharge du LDO peut entraîner une instabilité de l'ESP8266 et donc des crashes aléatoires. Pour éviter ce cas, vous pouvez connecter le ventilateur au 5 V via une résistance (R6, l'empreinte sur le circuit imprimé correspond à une résistance de 1 W). Cela permet de réduire encore plus le flux d'air et, surtout, le bruit. Notez que les ventilateurs fonctionnant à une tension inférieure à celle pour laquelle ils sont conçus ont tendance à avoir des problèmes au démarrage ou à ne pas démarrer du tout. Si le ventilateur doit être directement connecté au 5 V, vous pouvez insérer un pontage à la place de la résistance R6.

Autre matériel

Le buzzer est également alimenté en 5 V. Le niveau de sortie haut des E/S du D1 Mini est de 3,3 V. Pour désactiver le buzzer, le diviseur de tension R4/R7 est donc connecté à la base du transistor PNP T1. Certains buzzers sont inductifs et susceptibles de produire des pics de tension, notamment lors de la mise hors tension. La diode D1 les empêche d'endommager le transistor. Le circuit imprimé comporte des empreintes pour les buzzers de 6,5 mm et 7,62 mm.

Nous avons utilisé un interrupteur doté d'un indicateur optique sous la forme d'un anneau bleu pour S1. Aucune résistance externe n'est nécessaire pour la LED à l'intérieur de l'interrupteur. Les broches marquées + et - sur le circuit imprimé (PCB), à côté de celles de l'interrupteur, sont les deux connexions pour la LED. Quatre fils sont reliés au connecteur à quatre voies S1 (ou soudés directement sur le circuit imprimé). Bien sûr, vous pouvez aussi utiliser un interrupteur et une LED séparément. R5 définit alors le courant pour la LED et la conception du panneau avant doit être modifiée pour tenir compte de cet indicateur optique. Avec l'interrupteur proposé dans la liste des composants, la résistance R5 n'est pas requise et doit être remplacée par un pontage.

Circuit imprimé

La conception et les fichiers Gerber du circuit imprimé illustrés à la **figure 2** peuvent être téléchargés à partir du référentiel GitHub [1] de ce projet. Avec les fichiers Gerber, vous pouvez commander la carte auprès de votre fabricant préféré. Tous les composants sont à trous

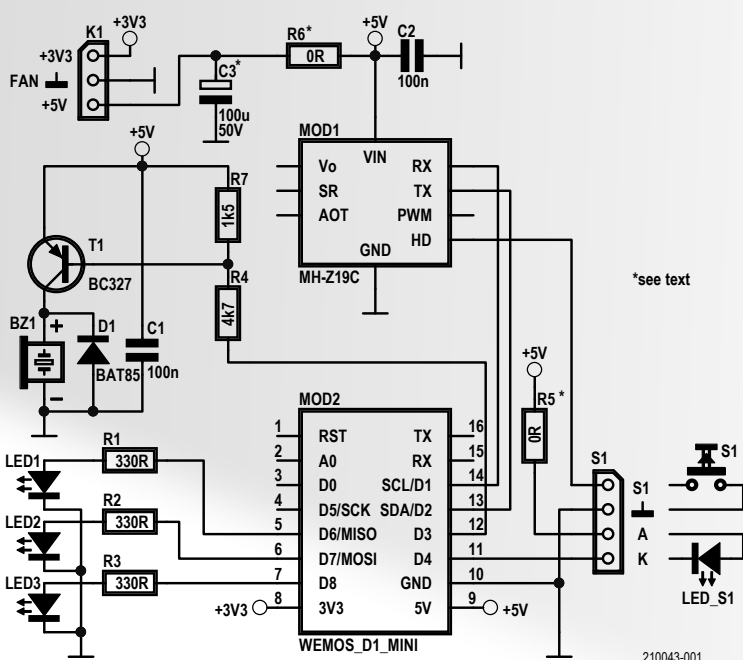


Figure 1. Schéma du CO₂ Guard.

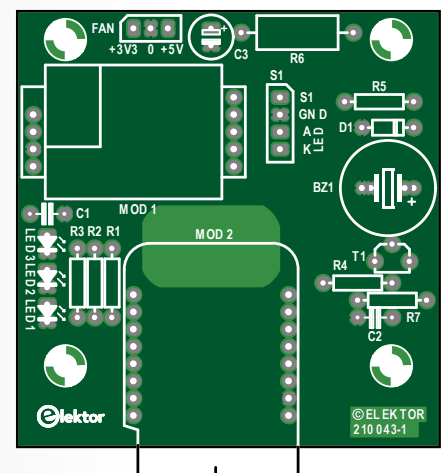


Figure 2. Circuit imprimé su CO₂ Guard.

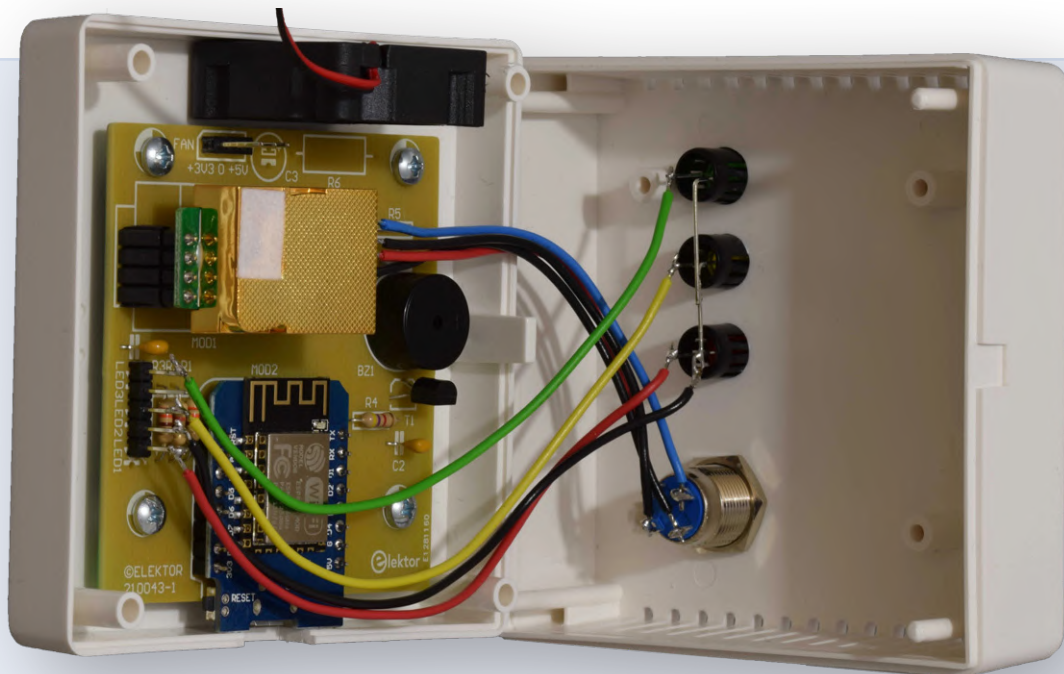


Figure 3. Ventilateur monté.

traversants et le circuit imprimé peut être à une seule face ; la couche supérieure est seulement constituée de cuivre pour éviter les coûts additionnels des circuits imprimés monocouche. Ainsi, même pour les makers débutants, le soudage ne sera pas trop difficile en suivant les directives données. Et avec toutes les traces sur la couche inférieure de ce circuit, il est relativement facile de dessouder les composants en cas d'erreur. Commencez par assembler les composants les plus petits, les résistances et la diode, puis les condensateurs, le transistor, le buzzer et les connecteurs (supports pour le capteur de CO₂ MOD1, connecteurs optionnels pour l'interrupteur et les LED). Pour monter le module D1 Mini (MOD2), soudez d'abord les deux connecteurs mâles à huit broches sur le côté du module où se trouvent le connecteur USB et la plupart des composants (voir **figure 3**). Il est recommandé de ne pas utiliser de supports de circuits (sockets) sur le circuit pour MOD2, car le circuit subit une certaine force lors du branchement et du débranchement du connecteur USB. Il est préférable de souder D1 Mini sur le circuit. De plus, l'utilisation de supports de circuits pourrait limiter le flux d'air à travers les trous de ventilation du boîtier.

Lorsque le circuit imprimé est fixé à l'aide de quatre vis sur la moitié inférieure du boîtier et le ventilateur collé (voir ci-dessous), le module du capteur peut être branché dans les supports correspondants. **Ne touchez jamais le dessus du capteur**, comme indiqué dans la fiche technique [2] : « Veuillez éviter la pression de son enceinte en plastique doré, quelle que soit la direction, pendant la soudure, l'installation et l'utilisation. En particulier, ne touchez jamais l'entrée d'air (filtre à poussière blanche). »

Les supports à quatre et cinq broches du capteur seront probablement fournis sous la forme d'un support SIL à 10 broches. Coupez ce bloc au niveau de la cinquième broche pour obtenir deux supports distincts à quatre et cinq broches.

Préparer le boîtier

Commencez par utiliser une copie du schéma du panneau pour les positionner correctement dans les quatre trous (trois LEDs et l'interrupteur), et utilisez un poinçon pour marquer les quatre centres. Sur la page de ce projet sur Elektor Labs [3], vous trouverez des conseils et astuces pour éviter d'endommager le boîtier lors du perçage de plus

gros trous dans le panneau avant en plastique. Avec ce type de boîtier, faites surtout attention aux longues broches qui servent à le fermer ; elles risquent de se casser si elles sont mal manipulées !

Une entaille doit être faite sur un côté de la moitié inférieure du boîtier pour accéder au connecteur micro-USB du module D1 Mini, qui sert d'alimentation pour le CO₂ Guard. Assurez-vous que l'entaille est suffisamment grande pour la prise USB de votre alimentation. Placez le mini module D1 sur le circuit imprimé pour marquer l'emplacement correct. À ce stade, il n'est pas encore nécessaire de souder le module avec les connecteurs mâles correspondants. Avec le module placé sur le circuit imprimé et ce dernier posé correctement sur les vis de fixation (vous pouvez utiliser deux vis temporaires), la position de l'encoche peut être repérée. Elle est légèrement décentrée.

Vous n'avez pas besoin de faire de trous pour le flux d'air du ventilateur ; celui-ci est collé sur la face inférieure du boîtier avec de la super colle ou de la colle époxy (**figure 3**). Pour éviter toute résonance, le ventilateur ne doit pas toucher le couvercle de l'enceinte. Le ventilateur doit souffler l'air hors du boîtier. Avant de le monter, vérifiez la direction du courant d'air en le connectant à une alimentation de 3,3 V à 5 V ou vérifiez s'il y a une flèche sur le côté du cadre du ventilateur indiquant la direction du flux. Faites attention à la polarité des fils. Une mauvaise polarité endommagera le ventilateur. (Les modèles modernes comportent souvent des circuits à l'intérieur).

Connexion des composants du panneau avant

Pour connecter les LED, quatre fils suffisent. La connexion à la masse entre le circuit imprimé et les trois LED sur la face avant peut être commune, il suffit de connecter les trois cathodes des LED et d'utiliser un seul fil.

Assurez-vous qu'aucun fil ne touche l'antenne du circuit MOD2 lorsque le couvercle du boîtier est finalement déposé sur la partie inférieure. Gardez-les aussi loin que possible du module Wemos, mais ne les allongez pas plus que nécessaire. Les fils connectés à l'interrupteur doivent avoir une longueur de 8,5 cm seulement et ceux connectés aux LED d'environ 12 cm. Avec cette longueur, vous pouvez enlever le haut du boîtier et le placer sur le côté, vers le bas, tout en gardant tous les fils connectés. Pas besoin de vis pour fermer le boîtier ; quatre

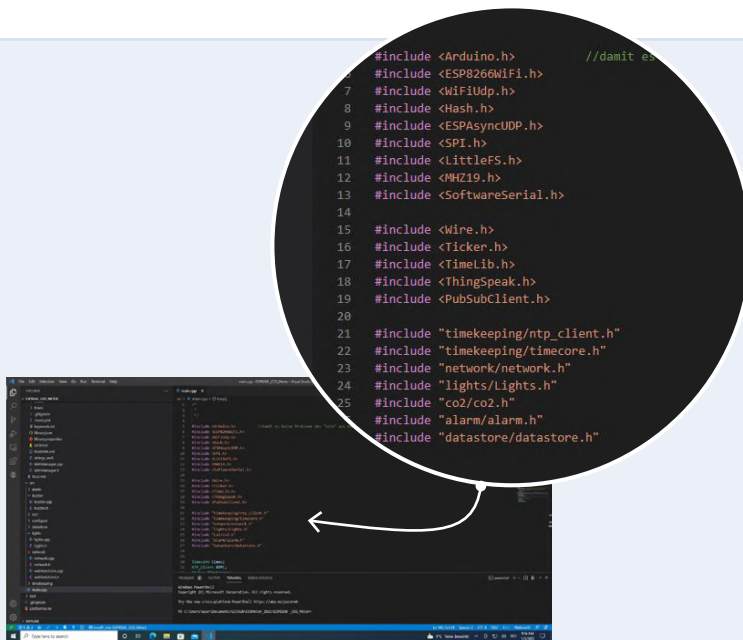


Figure 4. Code chargé dans l'IDE PlatformIO.

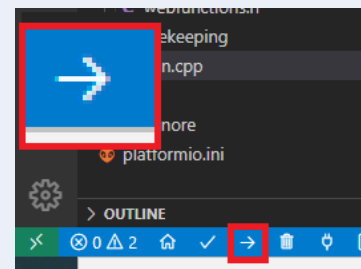


Figure 5. Bouton pour le téléversement du micrologiciel.

longues broches en plastique du couvercle glissent dans quatre longs bossages de montage.

Bien entendu, n'hésitez pas à créer votre propre design de panneau avant. Vous pouvez par exemple utiliser une LED bleue et un interrupteur séparés. Mais faites attention à l'emplacement des LED et de l'interrupteur ; ils ne doivent pas être situés trop près des modules.

Téléversement du micrologiciel sur Wemos

Le micrologiciel de l'ESP8266 sur le module Wemos est créé à l'aide du framework Arduino, vous avez donc besoin du sketch et d'un ensemble de bibliothèques. Pour le développement du code de CO₂ Guard, nous utilisons PlatformIO et Visual Studio Code. Les bibliothèques, le projet et les paramètres de téléversement sont combinés afin de faciliter la compilation du code.

Pour installer Visual Studio Code et PlatformIO, vous pouvez suivre le guide [4] fourni par PlatformIO. Après l'installation, vous devez utiliser le code source du dépôt GitHub d'Elektor. Si vous voulez en savoir plus sur la manière d'obtenir le code, regardez la vidéo de notre collègue Clemens Valens [5]. De plus, si vous souhaitez en savoir plus sur GitHub et l'utilisation des dépôts, visionnez le webinaire Elektor sur GitHub [6]. Une fois que vous avez le code et que vous l'ouvrez avec l'IDE PlatformIO, vous devriez voir ce qui est reproduit sur la **figure 4**. Connectez Wemos à votre ordinateur et appuyez sur le bouton de téléversement (**figure 5**). PlatformIO téléchargera toutes les bibliothèques requises, les outils, les packages de gestion de la carte et commencera à effectuer des compilations, puis des téléversements. À ce stade, PlatformIO a établi et lancé le micrologiciel. Comme pour la plupart des projets basés sur l'ESP8266, il y a une deuxième étape : comme le micrologiciel offre une configuration via un serveur Web, nous devons également créer les pages Web sur l'ESP8266. Cliquez dans Visual Studio Code sur l'icône de la petite fourmi et vous devriez voir apparaître une liste de tâches (**figure 6**). Cliquez d'abord sur *Build Filesystem Image*, puis sur *Upload Filesystem Image*. Cela permettra d'écrire le contenu du serveur Web sur l'ESP8266. Si vous obtenez des erreurs du type «*can't access COM port*», assurez-vous que vous n'avez pas de terminal série ouvert ce qui bloquerait le port. Ensuite, le micrologiciel est prêt à être configuré et utilisé.

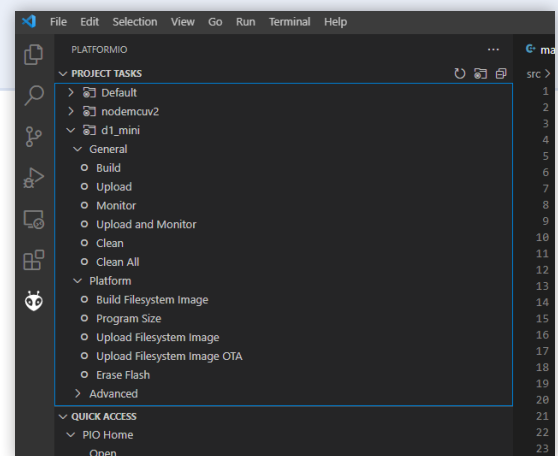


Figure 6. L'icône de la petite tête de fourmi montre plus d'options.

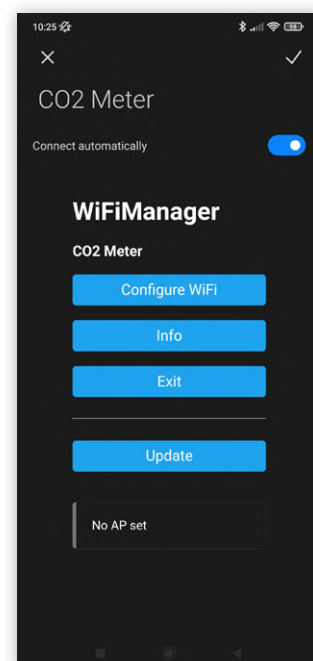


Figure 7. Paramètres du Wi-Fi.

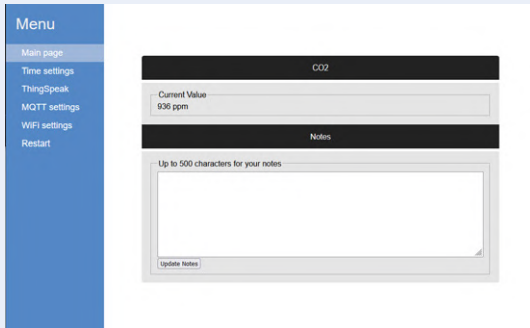


Figure 8. Interface web du CO₂ Guard.

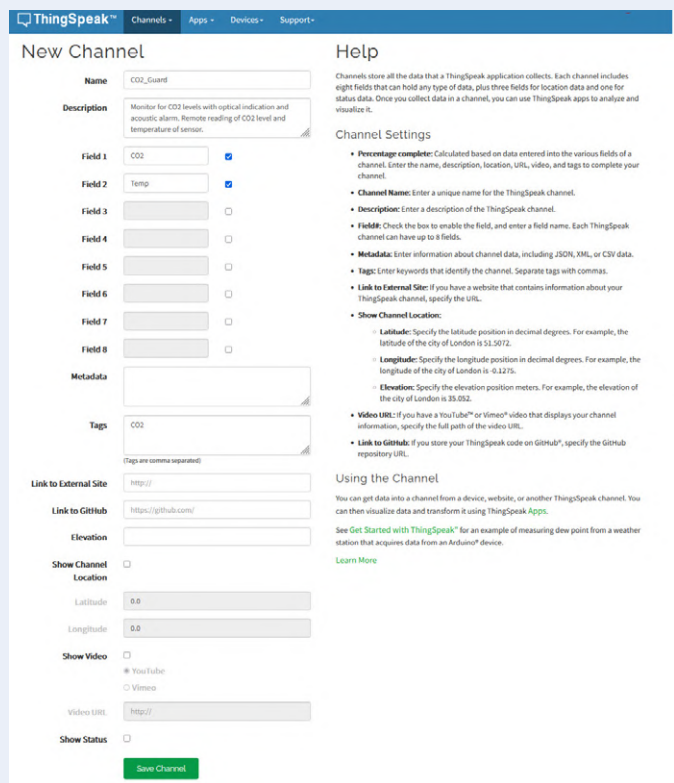
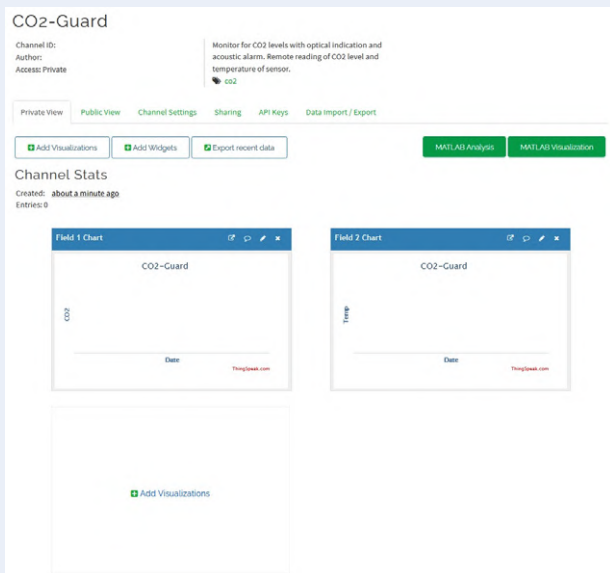


Figure 9. Paramètres de ThingSpeak.

Figure 10. Nouveau canal ThingSpeak généré.

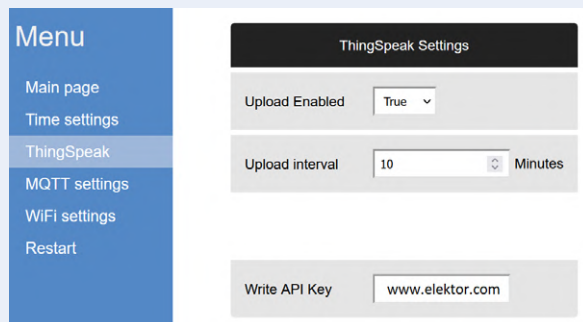


Figure 11. Paramètres de ThingSpeak.

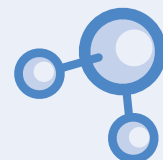
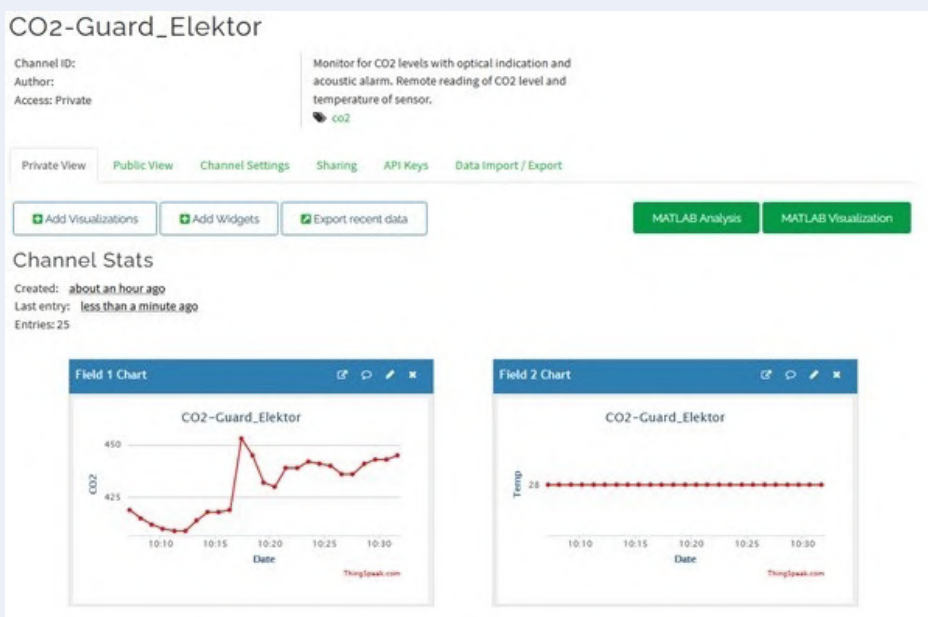


Figure 12. Données reçues.



Connectivité Wi-Fi et Cloud

Si votre ESP8266 sur le module Wemos ne parvient pas à se connecter à un réseau Wi-Fi, il lancera un point d'accès appelé « CO₂ Meter ». Connectez-vous à ce point et ouvrez un navigateur pointant vers <http://192.168.4.1>. Vous devriez voir l'interface de configuration WiFi Manager (**figure 7**). Cliquez sur Configure WiFi, sélectionnez votre réseau et appliquez les paramètres requis avant de cliquer sur Save. L'ESP8266 tentera de se connecter à votre réseau. S'il peut établir une connexion, vous pouvez utiliser votre réseau <http://co2guard.local> pour accéder à l'interface Web (**figure 8**). À partir de là, vous pouvez commencer à configurer le CO₂ Guard.

Si vous le souhaitez, vous pouvez connecter le CO₂ Guard à une solution de stockage de données Web comme ThingSpeak ou un courtier MQTT, comme celui utilisé dans Node-RED. Avec ce dernier, vous pouvez automatiser encore plus le détecteur, par exemple en envoyant un message à votre téléphone portable si le CO₂ dépasse un certain niveau. Pour démarrer avec ThingSpeak, nous vous proposons un guide rapide de la configuration du compte.

ThingSpeak : création d'un compte MathWorks

Si vous souhaitez uniquement surveiller la concentration actuelle en CO₂ dans votre pièce, vous n'avez pas besoin d'une connexion Wi-Fi et certainement pas d'un accès à un compte ThingSpeak. Les LED (et dans les cas extrêmes, le buzzer) donnent alors une indication claire de la qualité de l'air, du moins en ce qui concerne le CO₂. Mais si vous souhaitez, par exemple, enregistrer l'évolution de ces mesures et analyser les données au cours du temps, ou pouvoir visualiser de loin la concentration actuelle, alors la connexion à ThingSpeak via le Wi-Fi est un choix évident. Vous pouvez également suivre la température, qui est mesurée par le module de capteur, mais qui n'est pas affichée sur le CO₂ Guard lui-même.

Pour créer un canal en ligne pour vos mesures, accédez au site Web de ThingSpeak [7] et choisissez *Get Started For Free*. Sur la page suivante, vous pouvez créer un compte MathWorks (si vous n'en avez pas déjà un). Ensuite, créez un nouveau canal et entrez les données requises comme indiqué dans la **figure 9**. Cliquez sur *Save Channel*. Un aperçu du canal qui vient d'être créé est présenté comme illustré à la **figure 10**.

Dans l'onglet *API Keys*, nous devons copier la valeur *Write API Key* pour une utilisation ultérieure. Gardez-la privée et ne la montrez à personne. Accédez à l'interface web du CO₂ Guard et sélectionnez l'option *ThingSpeak* dans le menu latéral. Une page contenant les paramètres de ThingSpeak s'affiche. Saisissez l'*API Key*, sélectionnez un intervalle de téléchargement et activez ThingSpeak (**figure 11**). Vous revenez maintenant sur le site ThingSpeak de MathWorks. Dans le menu *Channels*, choisissez *My Channels*, puis l'onglet *Private View*. Les premières mesures devraient apparaître au bout de quelque temps, comme dans la **figure 12**.

Pour partager vos données avec d'autres personnes, vous pouvez ajouter des utilisateurs individuels avec une autorisation de visualisation. Vous pouvez également rendre le canal public pour que vos données soient accessibles. Les utilisateurs individuels recevront un courrier électronique à l'adresse e-mail saisie dans l'onglet *Sharing*. Le partage des données est limité à trois autres utilisateurs si vous

utilisez un plan gratuit (non payant) pour ThingSpeak. Pour plus de détails sur le partage des données, vous pouvez consulter le centre d'aide de MathWorks [8].

Connexion MQTT

MQTT est un moyen flexible de distribuer vos données à d'autres systèmes qui peuvent les stocker ou les traiter. Vous n'avez pas besoin d'un service externe pour collecter les données ou les traiter, et vous pouvez conserver les données dans votre propre réseau. Un des outils qui offre une connectivité MQTT est Node-RED (par exemple, sur un Raspberry Pi). Nous avons couvert beaucoup d'installations pour Node-RED dans le passé, s'il existe un guide d'installation rapide pour votre Raspberry Pi sur le site web de Node-RED [9]. Pour MQTT, dans l'interface web du CO₂ Guard, vous pouvez indiquer un serveur et une rubrique où les données doivent être publiées (**figure 13**). Les données elles-mêmes sont transférées sous la forme d'une chaîne JSON, comportant les objets « CO₂Value » et « Temp », présentant la valeur du CO₂ en ppm et la température en °C. Si vous souhaitez en savoir plus sur Node-RED en utilisant un Raspberry Pi, consultez le livre de Dogan Ibrahim, *Programming with Node-RED* (Elektor 2020) [10].

Fichiers de conception de circuits imprimés KiCad

À l'origine, le circuit imprimé a été conçu dans Altium, et les fichiers Gerber fournis sont basés sur cette démarche éprouvée. La conception est assez simple, comme vous pouvez le voir sur les schémas. Comme le schéma n'est pas très complexe (**figure 1**), il a été reproduit dans KiCad. Outre le schéma, le circuit imprimé doit aussi être transféré. Le projet ayant été entamé dans la dernière version stable

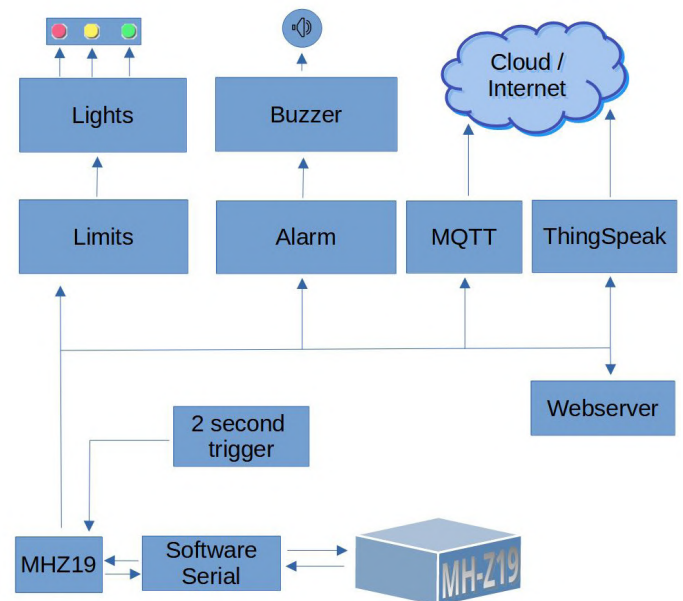


Figure 13. Paramètres MQTT.




de KiCad 5, il n'y avait pas d'importation native (intégrée) pour les cartes Altium. L'astuce est d'utiliser les fichiers Gerber fournis et de les importer dans KiCad. Le visualiseur Gerber de KiCad est capable de générer des fichiers de carte que vous pouvez utiliser dans l'outil de dessin. Il fournit non seulement les contours de la carte, mais aussi les empreintes et les trous de montage à utiliser pour le placement des composants. Cette méthode peut être utilisée pour importer vos projets dans KiCad, mais il est préférable d'utiliser la version 6 de KiCad et l'importation native de cartes Altium si vous en avez des fichiers. Mais pourquoi faire cette conversion ? Tout d'abord, cela permet à d'autres personnes de travailler directement avec les fichiers KiCad et de les modifier si elles le souhaitent. De plus, avec certains services de production de PCB, il suffit de glisser-déposer un fichier de carte KiCad pour que votre carte soit fabriquée (par exemple, chez Aisler) [11]. Vous n'avez pas besoin d'effectuer l'exportation vers des fichiers Gerber et de perçage pour générer un fichier ZIP. Cependant, si votre service PCB a besoin de ces fichiers, vous pouvez toujours les générer dans KiCad selon les spécifications requises par votre service.

La combinaison de KiCad et FreeCad est une solution très efficace si vous souhaitez concevoir votre propre boîtier pour le circuit imprimé, comme le montre le projet de Noël [12]. Imprimer un boîtier est pour certains plus pratique que de percer des trous dans un autre préfabriqué. Si vous avez chez vous une imprimante 3D inutilisée, pourquoi ne pas vous en servir ? De plus, cela fera un composant de moins à commander. Au moment de la rédaction de cet article, le boîtier imprimable en 3D est en cours de conception. En attendant, si vous voulez utiliser un boîtier, vous devez percer quelques trous.

Construire le vôtre

Construire un détecteur de CO₂ pour votre maison ou votre espace de travail n'est pas si difficile, et comme vous le savez, de nombreux dispositifs de mesure de CO₂ ont été développés sous la forme de

projets « maison » et commerciaux. Le capteur MH-Z19 utilisé dans ce projet constitue un outil de mesure abordable et fiable. L'ESP8266 est un microcontrôleur Wi-Fi éprouvé qui peut non seulement être programmé avec le framework Arduino, mais qui prend également en charge le Home Assistant.

Vous voulez construire le vôtre ? Le capteur vous coûtera environ 30 € (en Europe). L'interrupteur Wemod D1 mini, le circuit imprimé, les LED, le boîtier, le ventilateur et les frais d'expédition s'ajouteront à la facture. Le coût total des composants s'élèvera donc à environ 60 €. D'autre part, la surveillance de la concentration en CO₂ dans vos pièces présente de nombreux avantages. Connaissant la concentration en CO₂, vous pouvez vous assurer que le renouvellement de l'air est suffisant. 

210043-04

Des questions, des commentaires ?

Contactez Elektor (redaction@elektor.fr).

Contributeurs

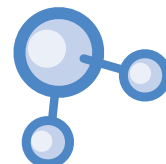
Conception originale : **Florian Schäffer**

Conception et texte : **Mathias Claußen, Ton Giesberts, Luc Lemmens**

Rédaction : **Jens Nickel, C. J. Abate**

Traduction : **Asma Adhimi**

Mise en page : **Harmen Heida**



Logiciel

Certains aspects du logiciel initialement fourni par le développeur sur la plate-forme Elektor Labs pourraient être améliorés. Par exemple, les identifiants codés en dur pour votre réseau Wi-Fi ne fonctionnent que si vous ne changez jamais le nom du réseau ou le mot de passe auquel l'appareil est connecté. De plus, garder vos identifiants quelque part dans des fichiers de code n'est pas le moyen le plus sûr de les stocker. Il en va de même pour l'API Key ThingSpeak. Si vous souhaitez modifier cette clé, vous devez recompiler tout le code et le télécharger sur l'appareil.

Elektor Lab n'a gardé dans le code (sous forme de `#defines` dans `/configure/configure.h`) que l'affectation des broches, ainsi que les seuils de CO₂ pour les trois LED et le buzzer. Dans le fichier `/configure/configure.h`, vous avez tous les paramètres codés en dur à un seul emplacement et vous pouvez les modifier si nécessaire. Bien sûr, après modification, vous devrez recompiler le micrologiciel et le télécharger sur l'ESP8266.

Dans le logiciel, le flux de données est assez simple, comme vous pouvez le voir sur l'image. Toutes les deux secondes, les valeurs des capteurs sont vérifiées et renvoyées aux différents modules de code. Pour la surveillance locale du niveau de CO₂, nous avons la classe `limits`, qui contrôlera finalement nos trois LED, et nous avons la classe `alarm`, qui fera sonner le buzzer si un certain niveau de CO₂ est dépassé. Ces classes sont créées de manière générique, ce qui permet de changer facilement la disposition des LED et la façon d'afficher les valeurs. Il en va de même pour le buzzer. Si vous souhaitez utiliser un buzzer actif ou même un module sonore, cette classe est suffisamment flexible pour le faire. Pour la connectivité avec le cloud, nous avons utilisé, dans la mesure du possible, des bibliothèques prédéfinies, notamment pour ThingSpeak. Cela simplifie considérablement l'échange de données avec ce service par rapport à la station météorologique ESP32 [13] que nous avons publiée [14] il y a quelques années.



LISTE DES COMPOSANTS

Résistances

R1, R2, R3 = 330 Ω, 0,25 W, 5 %

R4 = 4,7 kΩ, 0,25 W, 5 %

R5 = 0 Ω, connecter un fil, voir le texte

R6 = 0 Ω, connecter un fil pour 5 V ou laisser ouvert, voir le texte

R7 = 1,5 kΩ, 0,25 W, 5 %

Condensateurs

C1, C2 = 100 nF, 50 V, 10 %, X7R, espacement des fils 5 mm

C3 = 100 µF, 25 V, optionnel, non monté, voir texte

Semi-conducteurs

D1 = BAT85, DO-35

T1 = BC327, PNP, TO-92

LED1 = LED rouge, 5 mm

LED2 = LED jaune, 5 mm

LED3 = LED verte, 5 mm

Divers

MOD1 = connecteur 1x10 SIL, femelle, divisé en 1x4 et 1x5 (voir Texte)

MOD1 = MH-Z19C, version 400-5000 ppm, broches en bas

MOD2 = Wemos D1 mini avec 2x connecteur 1x8 SIL

S1 = interrupteur avec LED intégrée (marche momentanée, auto-réinitialisation, cercle bleu)

Boîtier PP73BL, Supertronic

BZ1 = buzzer CC, 5 V, pas 5 or 7,62 mm, diam. 14 mm max tube thermorétractable, I.D 2.4 mm, 20 cm

Boîtier de LED, pour LED 5 mm (LED1..3), noire

Ventilateur 5 V CC, 30x30x10,5 mm

Connecteur SIL mâle à 12 broches

Circuit imprimé 210043-1 v2.0

Fil fin pour connecter les LED et l'interrupteur :

fil torsadé, 0,25 mm², noir (2x8,5+12 cm)

fil torsadé, 0,25 mm², rouge (8,5+12 cm)

fil torsadé, 0,25 mm², jaune (12 cm)

fil torsadé, 0,25 mm², vert (12 cm)

fil torsadé, 0,25 mm², bleu (8,5 cm)



PRODUITS

- **WeMos D1 mini Pro – ESP8266 based WiFi Module** www.elektor.fr/19185
- **PCB V2.1 with Components (excluding WeMos D1 and MH-Z19)** www.elektormagazine.de/aisler/co2guard
- **H. Henrik Skovgaard, IoT Home Hacks with ESP8266** www.elektor.fr/19158
- **D. Ibrahim, Programming with Node-RED (E-book)** www.elektor.fr/19225



LIENS

- [1] Référentiel GitHub d'Elektor: <https://github.com/ElektorLabs/210043-CO2-Guard>
- [2] Fiche technique de Winsen MHZ-19: www.winsen-sensor.com/d/files/infrared-gas-sensor/mh-z19b-co2-ver1_0.pdf
- [3] « CO2 Guard » Elektor-Labs.com: www.elektormagazine.fr/labs/co2-guard
- [4] Guide d'installation de PlatformIO : <https://platformio.org/install/ide?install=vscode>
- [5] « Comment coder (télécharger) ou cloner depuis GitHub » Elektor TV: www.youtube.com/watch?v=X5e3xQBeqf8
- [6] « Webinaire GitHub : GitHub pour les débutants, » Elektor TV: www.youtube.com/watch?v=j_LgvVhBdwQ
- [7] MathWorks ThingSpeak: <http://thingspeak.com>
- [8] Aide MathWorks ThingSpeak: <https://bit.ly/ChannelPropertiesMATLAB>
- [9] Guide d'installation de Node-RED : <https://nodered.org/docs/getting-started/raspberrypi>
- [10] D. Ibrahim, Programming with Node-RED, Elektor 2020: www.elektor.fr/programming-with-node-red-e-book
- [11] AISLER B.V: <http://aisler.net>
- [12] M. Claußen, « DIY Christmas Fireplace: A 3D Puzzle with PCBs, LEDs, and Raspberry Pi Pico » ElektorMagazine.com: www.elektormagazine.com/articles/diy-christmas-fireplace-3d-puzzle-pcbs-leds-raspberry-pi-pico
- [13] « ESP32 Weatherstation » Elektor-Labs.com: www.elektormagazine.fr/labs/esp32-weather-station-180468
- [14] R. Aarts, « station météo à ESP32 » ElektorMag 1-2/2019: www.elektormagazine.fr/magazine/elektor-71/42320

kit de mesure de la qualité de l'air pour Raspberry Pi de MonkMakes

Mesure de température et de CO₂e



Luc Lemmens (Elektor)

Nous sommes très nombreux à être confinés dans des locaux (privés), d'où la popularité croissante des modules de mesure de la qualité de l'air peu coûteux. Le kit de MonkMakes mesure ainsi le taux d'équivalent CO₂ et la température. Il est spécialement conçu pour être utilisé avec un Raspberry Pi 400, mais il peut également être connecté à d'autres modèles à l'aide de fils de pontage et d'un gabarit GPIO inclus.

Les thermomètres sont utilisés de longue date pour surveiller la température ambiante. Ces dernières années, les appareils de mesure de CO₂ sont devenus de plus en plus courants pour la surveillance de la qualité de l'air. Un excès de dioxyde de carbone (CO₂) a un effet négatif sur la concentration et ce gaz est nocif à des niveaux encore plus élevés. Ce kit mesure la qualité de l'air dans une pièce (niveau de pollution) ainsi que la température. Il est conçu comme un greffon pour le Raspberry Pi, mais il peut également être utilisé comme dispositif autonome. La carte est dotée d'un buzzer et d'une barre de six LED (deux vertes, deux oranges et deux rouges) qui indiquent la qualité de l'air. Les mesures de température et de qualité de l'air peuvent être traitées par un Raspberry Pi. Le buzzer et l'affichage LED sont éventuellement commandés par le système hôte.

Le kit ne contient pas de documentation imprimée, mais il propose un lien vers le site web de MonkMakes où il est possible de télécharger la fiche technique et les instructions [1]. Ces documents contiennent toutes les informations nécessaires pour connecter la carte et l'utiliser. Des exemples d'applications en Python sont disponibles en téléchargement sur Github [2].

Matériel

En plus des six voyants lumineux et du buzzer – le gros composant carré au milieu du circuit imprimé de la **figure 1** – la carte comporte un voyant d'alimentation, un capteur de température, un capteur de CO₂,

un microcontrôleur et, bien sûr, un connecteur à 40 broches qui s'adapte directement au connecteur d'extension d'un Raspberry Pi 400 (fig. 2). Il va sans dire que les autres cartes Raspberry Pi ne peuvent pas être connectées directement. Pour cela, des fils de connexion sont inclus. Les quatre connexions nécessaires (deux pour l'alimentation et deux pour la connexion série) sont indiquées par des inscriptions sur la carte MonkMakes, mais aussi sur le gabarit fourni, pour les associer aux broches correspondantes du connecteur GPIO du Raspberry Pi, comme illustré sur la figure 3. La LED d'alimentation s'allume dès que la tension d'alimentation de 3,3 V est activée, ainsi que l'une des LED du niveau de CO₂.

Le capteur de température est un TMP235 [3] de Texas Instruments. Sa tension de sortie est proportionnelle à la température. Pour la mesure du CO₂, la carte MonkMakes utilise un capteur CCS811 de composés organiques volatils totaux (TVOC) [4]. Ce capteur ne mesure pas réellement la concentration de CO₂, mais plutôt le niveau d'un groupe de gaz appelés composés organiques volatils (COV). À l'intérieur, le niveau de ces gaz augmente à un taux comparable à celui du CO₂, et peut donc être utilisé pour estimer le niveau de CO₂ (appelé équivalent CO₂ ou CO₂e).

Le microcontrôleur ATtiny1614 intégré lit les deux capteurs et pilote l'affichage à barre de LED et le buzzer. Par le biais d'un protocole série, un système hôte peut demander les lectures des capteurs, ou allumer et éteindre les LED et le buzzer. La fiche technique du kit documente ce protocole simple, il ne sera donc pas trop difficile d'écrire votre propre logiciel pour le prendre en charge. Comme son nom l'indique, ce kit est conçu pour le Raspberry Pi, mais rien ne vous empêche de l'utiliser avec d'autres cartes ou systèmes dotés d'un UART 3,3 V. Le micrologiciel de l'ATtiny offre également un mode automatique (activé par défaut) qui affiche le niveau de CO₂e sur la barre de LED, sans aucune commande externe ; une simple alimentation de 3,3 V est nécessaire. Ainsi, même sans système hôte, le kit de mesure de la qualité de l'air peut servir de moniteur de CO₂e.

Logiciel

Comme mentionné précédemment, MonkMakes propose le téléchargement de quelques exemples de programmes en Python pour commander son kit, ce qui permet de tester et montrer toutes ses fonctions. Dans la section « Getting Started » (prise en main) de la documentation, les instructions décrivent clairement comment utiliser le logiciel sur une carte Raspberry Pi pour mettre en œuvre un détecteur de CO₂e, un détecteur de CO₂e avec alarme acoustique (fig. 4) et une application d'enregistrement de données.

En examinant les exemples, comme dans la figure 5, vous remarquerez que l'ATtiny et l'API vous évitent de récupérer et d'évaluer les données des capteurs : une simple instruction de l'hôte (Raspberry Pi) activera la carte Air Quality pour qu'elle renvoie la température ambiante actuelle (en °C) ou la concentration de CO₂ (en ppm), respectivement. Il existe des commandes similaires pour activer et désactiver le buzzer, mais aussi pour piloter les LED d'affichage du niveau de CO₂e.

Un beau design

Il suffit de quelques informations élémentaires sur le Raspberry Pi pour pouvoir exploiter ce kit de mesure de la qualité de l'air. Avantage important pour certains, mais moins intéressant pour d'autres : la connaissance des capteurs et la commande du buzzer et des LED ne sont pas obligatoires. Le micrologiciel (code source) de l'ATtiny1614 intégré n'est pas publié (nous ignorons donc ce qui se passe exactement dans ce microcontrôleur). Cependant, le protocole de communication avec la carte est assez simple et bien documenté, et le développement de

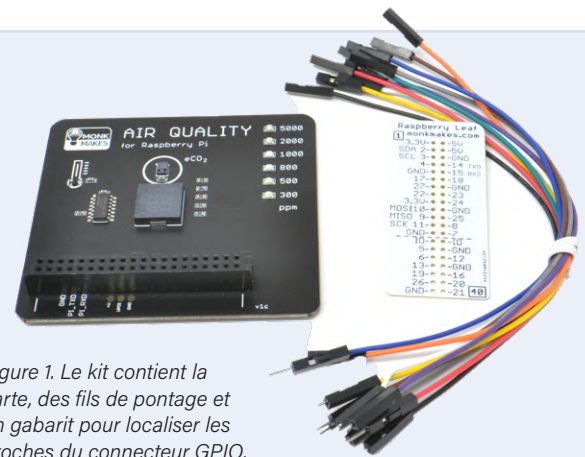


Figure 1. Le kit contient la carte, des fils de pontage et un gabarit pour localiser les broches du connecteur GPIO. (Source : MonkMakes)



Figure 2. Kit de qualité de l'air connecté à un Raspberry Pi 400. (Source : MonkMakes)

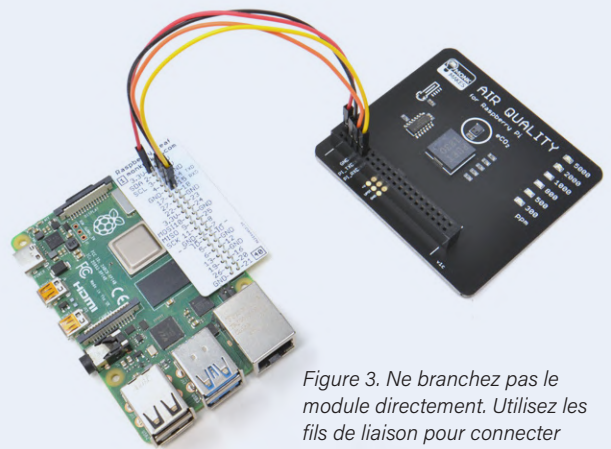


Figure 3. Ne branchez pas le module directement. Utilisez les fils de liaison pour connecter le kit à d'autres modèles de Raspberry Pi ! (Source : MonkMakes)

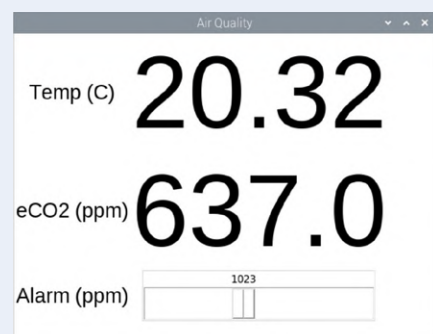


Figure 4. Sortie de l'écran du Raspberry Pi pour l'un des exemples. (Source : MonkMakes)

```

1 import threading
2 import time
3 from guizero import App, Text
4 from aq import AQ
5
6 aq = AQ()
7
8 app = App(title="Air Quality", width=550, height=300, layout="grid")
9
10 def update_readings(): # update fields with new temp and eCO2 readings
11     while True:
12         temp_c_field.value = str(aq.get_temp())
13         eco2_field.value = str(aq.get_eco2())
14         time.sleep(0.5)
15
16 t1 = threading.Thread(target=update_readings)
17 t1.start() # start the thread that updates the readings
18
19 aq.leds_automatic()
20
21 # define the user interface
22 Text(app, text="Temp (C)", grid=[0,0], size=20)

```

Figure 5. Code source Python montrant que seules des instructions simples et courtes sont nécessaires pour communiquer avec le kit de mesure de la qualité de l'air.

vos propres applications – même pour des systèmes cibles autres que les cartes Raspberry Pi – sera relativement facile. Le kit de mesure de la qualité de l'air MonkMakes pour Raspberry Pi est une carte bien conçue et correctement documentée qui, avec les exemples, convient également aux débutants souhaitant se lancer dans les mesures de température et de CO₂e. ◀

210681-04

Des questions, des commentaires ?

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

Contributeurs

Texte : Luc Lemmens

Illustrations : MonkMakes, Luc Lemmens

Rédaction : Jens Nickel, C. J. Abate

Mise en page : Harmen Heida

Traduction : Asma Adhimi



PRODUITS

> Kit de mesure de la qualité de l'air pour Raspberry Pi de MonkMakes

www.elektor.fr/19913

> Raspberry Pi 400 – PC à base de Raspberry Pi 4 (version US) + extension GPIO offerte

www.elektor.fr/19429

> Raspberry Pi 4 B (1 Go RAM)

www.elektor.fr/18966

Caractéristiques du capteur

Lecture minimale de CO ₂ e	400	ppm
Lecture maximale de CO ₂ e	4095	ppm
Résolution de mesure du CO ₂ e	1	ppm
Précision de mesure du CO ₂ e	non spécifiée	
Lecture minimale de la température	-10	°C
Lecture maximale de la température	100	°C
Précision de la température	±2	°C

À propos des concentrations de CO₂

La concentration de CO₂ dans l'air que nous respirons a une influence directe sur notre bien-être. Les taux de CO₂ présentent un intérêt particulier du point de vue de la santé publique. Pour faire simple, il s'agit d'une mesure de la quantité d'air que nous respirons provenant des expirations des autres. Nous, les humains, expirons du CO₂, et si plusieurs personnes se trouvent dans une pièce mal ventilée, la concentration de CO₂ augmentera progressivement, ainsi que la concentration des aérosols qui propagent les rhumes, les gripes et les coronavirus. Un autre impact important du niveau de CO₂ est celui influant sur les fonctions cognitives, c'est-à-dire sur la capacité de concentration et de réflexion.

Le tableau ci-dessous indique les concentrations auxquelles le CO₂ peut devenir nocif pour la santé. Les valeurs de CO₂ sont exprimées en ppm (parties par million).

250-400	Concentration normale dans l'air ambiant.
400-1000	Concentrations typiques dans les espaces intérieurs occupés avec un bon flux d'air.
1000-2000	Manifestations de somnolence et de manque d'air.
2000-5000	Maux de tête, somnolence et air stagnant, vicié et étouffant. Une mauvaise concentration, une perte d'attention, une accélération du rythme cardiaque et de légères nausées peuvent également survenir.
5000	Limite d'exposition en milieu de travail dans la plupart des pays.
>40000	L'exposition peut conduire à une grave privation d'oxygène entraînant des lésions cérébrales permanentes, un coma, voire la mort.

LIENS

[1] Page d'instructions de MonkMakes : http://monkmakes.com/pi_aq

[2] Logiciel sur Github : https://github.com/monkmakes/pi_aq

[3] Feuille de caractéristiques du TMP235 : www.ti.com/product/TMP235

[4] Feuille de caractéristiques du CCS811 : www.sciosense.com/products/environmental-sensors/ccs811-gas-sensor-solution/

e!ektor e-zine

Your dose of electronics



Chaque semaine où vous n'êtes pas abonné à l'e-zine d'Elektor est une semaine de grands articles et de projets électroniques qui vous manquent !

Alors, pourquoi attendre plus longtemps ? Abonnez-vous dès aujourd'hui à www.elektor.fr/ezine et recevez également le livre gratuit du projet Raspberry Pi !



À quoi pouvez-vous vous attendre ?

Éditorial

Chaque vendredi, vous recevrez les meilleurs articles et projets de la semaine. Nous couvrons les projets basés sur les MCU, l'IdO, la programmation, l'IA, et plus encore !

Promotionnel

Ne manquez pas les promotions de notre magasin, chaque mardi et jeudi nous avons une promotion spéciale pour vous.

Envoi des partenaires

Vous souhaitez rester informé des activités en cours dans le secteur ? Alors ce courriel vous donnera les meilleures informations. Non régulier, mais toujours le mercredi.

démarrer en électronique... (13)

Entrée en scène de la diode

Eric Bogers (Elektor)

Cela a pris un moment, mais nous avons enfin terminé notre tour des composants « passifs ». Dans cet article, nous nous intéresserons à quelques composants « actifs », et plus précisément aux semi-conducteurs. Voilà qui devient vraiment intéressant !



Nos premiers composants actifs : la diode et les composants associés. Une diode est-elle un composant actif ou passif ? On appelle composants actifs ceux qui sont capables d'amplifier un signal, ce qu'une diode ne fait pas. Mais parce qu'elle fait partie des composants semi-conducteurs (ou, dans le passé, des tubes électroniques, mais ceux-ci sortent du cadre de cette série d'articles), nous la rangeons dans la catégorie des transistors et autres composants actifs.

Si nous écrivions un « vrai » manuel, nous le commencerions avec de nombreuses pages sur les matériaux semi-conducteurs (silicium, germanium, sélénium et autres), le dopage, les jonctions PN et bien d'autres choses encore ; mais cette série d'articles s'adresse aux amateurs d'électronique débutants et non aux physiciens ou aux fabricants de semi-conducteurs. Le passionné d'électronique débutant est surtout intéressé par la fonction d'un composant particulier et comment il peut l'utiliser dans ses circuits – et beaucoup moins par la question de savoir comment et pourquoi un composant particulier fait ce qu'il fait.

Sur la **figure 1**, vous pouvez voir quelques modèles courants de diodes. En haut à gauche, deux diodes « ordinaires » : une 1N4148 et une 1N4001. En bas à gauche, trois LED (diodes électroluminescentes) de 3 mm, 5 mm et 10 mm de diamètre. À droite, un ensemble de quatre redresseurs en pont – ce sont des combinaisons de quatre diodes dans un seul boîtier. Ils sont utilisés en grand nombre dans les alimentations électriques à partir du secteur. Celui d'en bas à droite a un boîtier métallique avec un trou de vis de fixation à un radiateur. Son courant nominal dépasse 25 A.

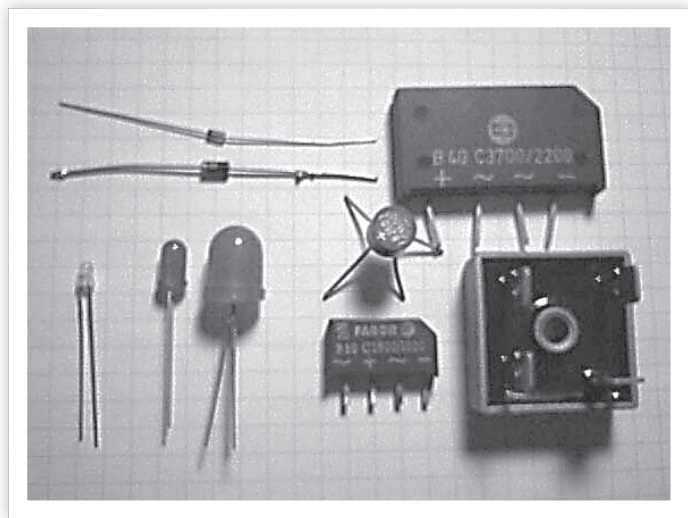


Figure 1. Échantillons de diodes et de ponts redresseurs.

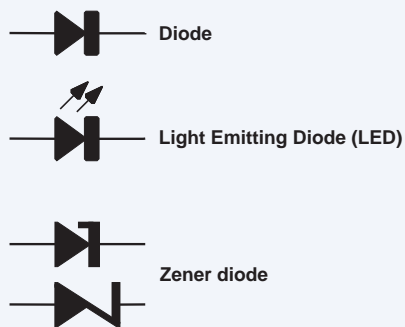


Figure 2. Symboles schématiques de quelques diodes courantes.

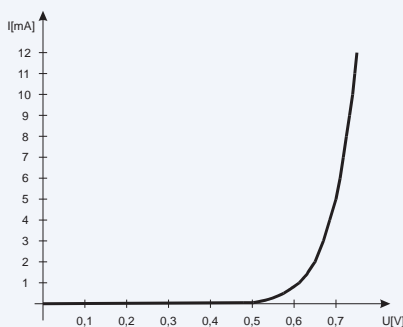


Figure 3. La caractéristique V/I d'une diode.

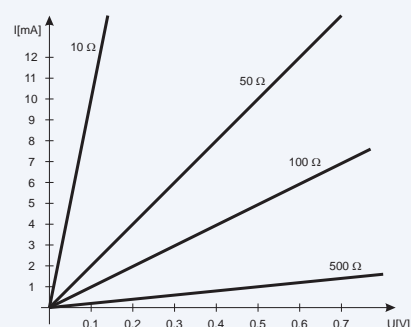


Figure 4. Caractéristiques V/I de quatre résistances.

La diode

Une diode fonctionne comme un redresseur de courant électrique : elle est conductrice dans un sens, mais pas dans l'autre. La **figure 2** présente les symboles schématiques de quelques diodes ; de haut en bas : une diode ordinaire, une diode électroluminescente (LED) et en dessous, deux versions du symbole de la diode Zener (dans le symbole du bas, le 'Z' de Zener est reconnaissable). Le symbole de la diode ressemble à une flèche qui pointe dans la direction de circulation du courant. La borne de gauche (positive) dans la figure 2 est appelée l'anode et celle de droite (négative) est appelée la cathode.

Une diode (au silicium) commence à être conductrice lorsque la tension sur l'anode dépasse celle sur la cathode d'environ 0,6 V – un peu comme une soupape mécanique où une certaine différence de pression est nécessaire avant qu'elle ne s'ouvre. Cette valeur est caractéristique du matériau semi-conducteur, dans ce cas le silicium. Pour une diode au germanium, elle est d'environ 0,3 V. Un physicien peut sans doute en fournir une explication détaillée, mais pour notre propos, c'est sans importance. Nous ne retenons que ces deux tensions : 0,6 V pour le silicium et 0,3 V pour le germanium.

Cela devient plus intéressant lorsque nous faisons un graphique du courant traversant une diode en fonction de la tension à ses bornes. La **figure 3** en donne un exemple. Dans ce diagramme V/I, l'axe horizontal représente la tension et l'axe vertical le courant. À titre de comparaison, la **figure 4** montre les caractéristiques V/I de quelques résistances ordinaires, des droites dont la pente est inversement proportionnelle à la valeur de la résistance.

L'examen attentif de la figure 3 montre que jusqu'à une tension de 0,5 V, il ne se passe presque rien – pratiquement aucun courant ne traverse la diode. Au-delà de 0,5 V, la pente de la courbe augmente rapidement ; à partir de ce point, elle est principalement déterminée par la résistance interne de la diode.

Pour être complet, la courbe de la figure 3 a été mesurée sur une diode au silicium de type 1N4148, mais la plupart des autres diodes au silicium présentent une courbe très voisine.

Spécifications d'une diode

Le fabricant d'une diode spécifiera au moins le courant maximal dans le sens direct et la tension maximale dans le sens inverse (sens de blocage, où la tension de la cathode est plus positive que celle de l'anode). Souvent, la puissance maximale admissible (la dissipation maximale) d'une diode est également mentionnée.

Le courant maximum de la 1N4148 (une diode à petits signaux « typique ») utilisée comme exemple ici, s'élève à 100 mA, tandis que la dissipation maximale est de 500 mW.

D'autre part, une diode n'est pas non plus capable de bloquer une tension inverse excessivement élevée. Lorsque le seuil de cette tension est dépassé, la diode commence à conduire dans le sens inverse. Si le courant n'est alors pas fortement limité, la diode surchauffe, provoquant la fusion du cristal de silicium, ce qui rend permanente la conduction dans les deux sens. Cette tension plus élevée entraîne également une dissipation beaucoup plus grande, de sorte que la diode sera détruite par un courant bien plus faible que dans le sens direct.

Régulation des tensions

En utilisant des diodes ordinaires, on peut obtenir une tension raisonnablement stable de manière très simple (voir **fig. 5**). Tout d'abord, le circuit à gauche de la figure 5 : ici, une diode est connectée via une résistance série de 1 kΩ à une source de tension continue. Pour une tension de la source de 7,5 V, un courant de 6,68 mA circule, et la chute de tension aux bornes de la diode s'élève à 0,715 V. Doublons maintenant la tension de la source à 15 V : le courant passe à 14,12 mA, mais la tension aux bornes de la diode n'augmente que faiblement, à 0,761 V. Conclusion : en doublant la tension d'alimentation, la tension aux bornes de la diode n'augmente que d'environ 6 %.

En utilisant un « circuit à deux étages » (figure 5, à droite), la tension est mieux régulée. La tension aux bornes des trois diodes en série est d'environ 2 V. Cette tension, déjà raisonnablement régulée, sert à alimenter la diode du deuxième étage.

Avec une tension d'entrée de 7,5 V, nous mesurons une tension de 0,617 V aux bornes de cette diode, qui passe à 0,662 V pour une tension d'entrée de 15 V. Ainsi, pour un doublement de la tension d'entrée (soit une augmentation de 100 %), la tension aux bornes de cette diode n'augmente que de 7 %.

La régulation de tensions à l'aide de diodes ordinaires n'est utilisée que pour des tensions très basses. Pour des tensions régulées plus élevées, nous utilisons des diodes Zener, que nous verrons dans le prochain article.

Circuits d'alimentation

L'une des applications les plus importantes des diodes est celle de redresseur dans les sources d'alimentation. Les circuits électroniques ont besoin pour leur alimentation d'une tension

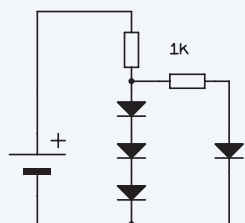
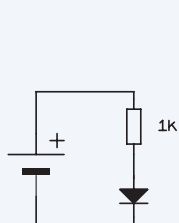


Figure 5. Régulation de tension avec des diodes ordinaires.



Figure 6. Redresseur simple alternance (demi-onde).

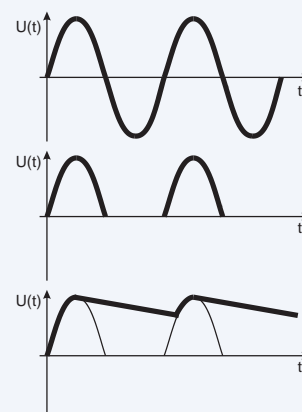


Figure 7. Formes d'onde de la tension pour le redresseur simple alternance (demi-onde).

continue qui, selon l'appareil, peut être comprise entre 1,5 V et 150 V. Cependant, le secteur ne fournit qu'une tension alternative de 230 V. Cette tension doit donc d'abord être transformée en une basse tension adéquate, puis être redressée et lissée et, dans la plupart des cas, régulée.

Redresseurs simple alternance

Simple alternance (fig. 6) signifie que, des deux demi-périodes de la tension d'entrée alternative, seule la demi-période positive est laissée passer et que pendant le temps relativement court où la tension de sortie du transformateur est supérieure à la tension aux bornes du condensateur tampon plus la chute de tension directe de la diode (qui s'élève à environ 0,7 V). Ce type de circuit est aussi appelé un redresseur demi-onde.

Sur la figure 7, le graphique du haut montre la tension alternative telle qu'elle est fournie par le transformateur. En l'absence du condensateur tampon, il y aura aux bornes de la résistance (la charge) une tension continue pulsée, comme le montre le graphique du milieu.

Le graphique du bas, le plus intéressant, montre la tension aux bornes du condensateur tampon. Au début de chaque demi-période positive de la tension alternative, ce condensateur est chargé très brièvement ; pendant le reste du cycle, il se décharge en alimentant la charge. En pratique, cette charge n'est généralement pas une simple résistance, mais un circuit électronique complet qu'il faut alimenter par une tension continue.

Dans le prochain article, nous verrons comment évaluer certaines caractéristiques de ces redresseurs (et d'autres) par le calcul. ❏

220003-04

La série d'articles « démarrer en électronique » est basée sur le livre « Basic Electronics Course » de Michael Ebner, publié par Elektor.

Des questions, des commentaires ?

Contactez Elektor (redaction@elektor.fr).

Contributeurs

Idee et illustrations : Michel Ebner

Texte et rédaction : Eric Bogers

Traduction : Helmut Müller

Mise en page : Giel Dols



PRODUITS

➤ B. Kainka, *Initiation à l'électronique et programmation de montages pour débutants*
www.elektor.fr/19339

➤ R. Mallard, *L'électronique pour les débutants*
www.elektor.fr/15662



Ils nous font confiance, n'est-ce pas ?

"Got the goods quickly and in pristine condition. Will buy again from them."

★★★★★ by Alexander Shopov

Rated 4 / 5 | 60 reviews



"Simply Great for EEE & Computer Science or other IoT Makers"

★★★★★ by FARGES Gerard

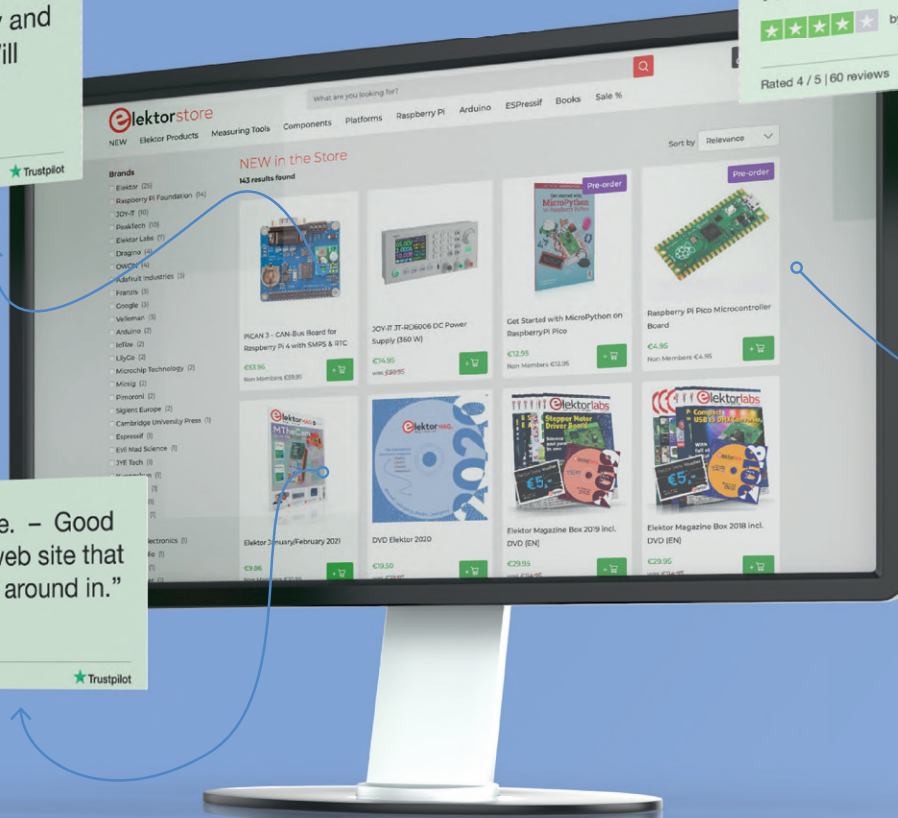
Rated 4 / 5 | 60 reviews



"Service & Web use. – Good service and clear web site that is easy to navigate around in."

★★★★★ by C. Buttmer

Rated 4 / 5 | 60 reviews



Nous aimons l'électronique et les projets, et nous faisons tout notre possible pour répondre aux besoins de nos clients.

Le magasin Elektor : **Jamais cher, toujours surprenant**

Elektor is rated **Excellent**

Based on 8 reviews



★★★★★ by Trustpilot

Consultez d'autres avis sur notre page Trustpilot : www.elektor.fr/TP



Vous pouvez également vous faire votre propre opinion en visitant notre Elektor Store, www.elektor.fr

zone D

Astuces, bonnes pratiques et autres informations pertinentes

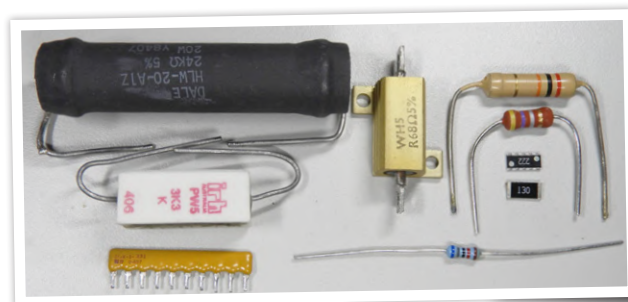


trucs et astuces pour tester les composants

Sans appareils de test coûteux

David Ashton (Australie)

Dans mon article sur l'identification des composants, j'ai évoqué l'importance du savoir-faire en matière de test des composants [1]. Généralement, en tant qu'amateur, il vous suffit de vérifier le fonctionnement d'un composant avant de le souder sur votre circuit. Il y a de nombreuses techniques pour tester des composants piochés dans votre stock de pièces, récupérés sur des cartes ou de provenance douteuse, sans avoir besoin d'un instrument de test sophistiqué ou coûteux.



Résistances

Les multimètres numériques sont désormais si répandus et bon marché qu'il n'y a guère d'intérêt à fabriquer ses propres testeurs. Il y a des exceptions – les résistances de très faible valeur (moins de 1 Ω environ) utilisées pour la détection du courant et d'autres applications ne peuvent pas être testées de manière fiable avec des multimètres numériques bon marché, car la résistance des cordons de test est du même ordre que celle de la résistance à tester.

Une méthode plus fiable consiste à faire passer un courant connu dans la résistance et à mesurer la tension à ses bornes. On trouve de nombreux modèles de ce que l'on appelle des milliohm-mètres, dans d'anciens articles d'Elektor et sur l'internet. Certains sont simples et

donnent des résultats acceptables pour un test rapide.

En général, on ne doit pas conserver les résistances qui ont chauffé au point de décolorer le corps ou les marquages, mais il est parfois possible de les tester pour les garder.



Condensateurs et inductances

J'avais environ 20 ans (il y a bien longtemps), quand j'ai construit moi-même un capacimètre. Un astable 555 déclenchait un monostable 74121 avec le condensateur à tester pour donner un train d'impulsions avec un rapport cyclique proportionnel à sa capacité. J'utilisais un potard pré-réglé pour chaque gamme et des condensateurs à 1% pour l'étalonner. Il mesurait d'environ 1 pF à 100 µF et c'est l'un des gadgets les plus utiles que j'ai jamais construits.

De nombreux multimètres numériques disposent désormais d'une fonction de test de capacité, et j'en ai trouvé un dans un magazine d'électronique récent pour environ 40 €, qui mesure la capacité et l'inductance, avec des plages malgré tout assez limitées. J'avais un LCR-mètre rudimentaire, mais convenable qui était à peu près au même prix, mais j'ai récemment investi dans un appareil plus polyvalent. Avec ce type d'appareil de test, vous en avez pour votre argent et le résultat dépend de votre budget et de vos exigences. Cela ne vaut plus vraiment la peine de construire vos propres testeurs. Mais quelquefois, vous pouvez avoir besoin d'effectuer des mesures plus étonnantes, telles que la résistance série équivalente (ESR) d'un condensateur, ou le Q d'une inductance, et pour cela, un instrument de test plus avancé peut s'avérer nécessaire. Au fil des ans, Elektor a publié de nombreux montages aptes à mesurer toutes les caractéristiques d'une capacité ou d'une inductance, et en cherchant sur le site du magazine, vous trouverez quelque chose qui répondra à vos besoins. La boutique d'Elektor propose également des testeurs à des prix raisonnables (voir encadré).

Les supercondensateurs ne sont qu'un cas particulier de condensateurs électrolytiques, mais ils dépassent la capacité de la plupart des testeurs. Leur tenue en tension est en général de 5,5 V, il faut donc veiller à ne pas dépasser cette valeur. Testez-les en utilisant une résistance de 100 Ω sur une alimentation de 5 V, et comptez le nombre de secondes pour qu'un condensateur entièrement déchargé atteigne 3,5 V. Divisez ce nombre par 100 et vous obtiendrez une estimation approximative de la capacité en farads. Cette méthode utilise la constante de temps RC, c'est-à-dire le temps nécessaire au condensateur pour atteindre 70% de la charge complète ($5 \text{ V} \times 0,7 = 3,5 \text{ V}$). Un condensateur de 1 F mettra $100 \Omega \times 1 \text{ F} = 100 \text{ s}$ pour atteindre 3,5 V. $100/100 = 1$, soit 1 F.

Transformateurs

Les transformateurs ne sont que des inductances particulières. Identifiez les enroulements avec un multimètre, puis mesurez les inductances.

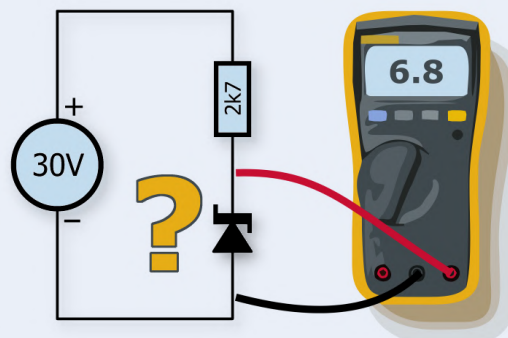
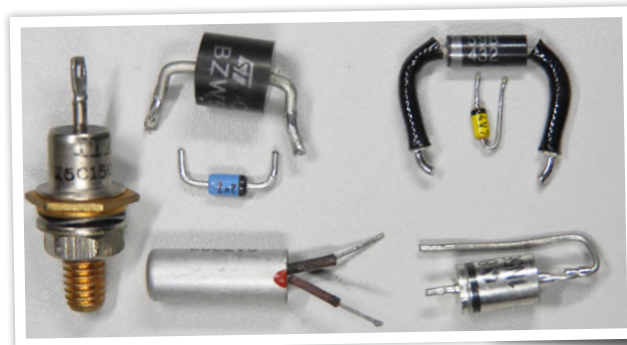


Figure 1. Une façon de tester rapidement les diodes Zener.

Pour un petit transformateur de puissance, tout enroulement dont la résistance est supérieure à environ 50 Ω et l'inductance supérieure à environ 10 H doit être un enroulement côté secteur. Et comme l'inductance est proportionnelle au carré du nombre de spires, prenez la racine carrée du rapport des inductances des enroulements pour obtenir le rapport de transformation.

Par exemple, un transformateur avec des inductances de 10 H et 50 mH a un rapport de transformation d'environ $(10 / 0,05)^{0,5} = 14$, donc s'il s'agit d'un transformateur secteur de 240 VCA, on peut s'attendre à ce que le secondaire donne environ 15 V. Mais de nombreux enroulements secteur ont une inductance plus élevée que ce que votre LCR-mètre peut mesurer, donc vous ne pourrez pas toujours faire cela.



Diodes

Les diodes sont faciles à tester avec n'importe quel multimètre : elles conduisent dans un sens et pas dans l'autre. Certains multimètres numériques ont une fonction de test de diode qui vous donnera une indication de la tension aux bornes de la diode lorsqu'elle est conductrice – environ 0,6 à 0,7 V signifie qu'il s'agit d'une diode au silicium standard, 0,3 à 0,5 V d'une diode Schottky, et 0,2 V et moins d'une diode au germanium. La position du point décimal peut varier, mais c'est le premier chiffre qu'il faut regarder.

C'est un peu plus difficile avec les diodes Zener. Elles se comportent comme des diodes normales lorsqu'elles sont testées avec un multimètre, car il est peu probable qu'un multimètre applique une tension suffisante pour les faire conduire en mode inverse (Zener). Pour tester les Zener, utilisez votre alimentation de laboratoire à sa tension maximale, avec une limitation de courant très basse (disons 10 mA), ou une résistance en série qui laisse passer environ 10 mA à la pleine tension. Appliquez ceci à la Zener et mesurez la tension à ses bornes avec votre multimètre (fig. 1). Cela ne fonctionnera que pour les Zener dont la tension est inférieure à celle de votre alimentation, mais une alimentation de 30 V couvrira la plupart des Zener.

Les LED peuvent être testées de la même manière que les Zener, mais

zone D

Astuces, bonnes pratiques et autres informations pertinentes

en abaissant la tension d'alimentation à environ 5 V – les LED ont une tension de claquage inverse assez faible. Testez toute LED avec un boîtier transparent dans les deux sens – il peut s'agir d'un type bicolore. Limitez le courant à 10 mA ; c'est suffisant pour que n'importe quelle LED s'allume et sans danger même pour les plus petites.

Transistors

J'avais environ 13 ans quand j'ai vu quelque part un article sur un testeur de transistors simple qui vérifiait le fonctionnement et donnait une valeur de H_{fe} (gain). Sa précision était raisonnable pour les transistors au silicium à petits signaux, mais insuffisante pour les transistors au germanium – ce qui de nos jours n'est guère un problème ! Il a beaucoup servi et je l'utilise encore aujourd'hui. Mais encore une fois, la plupart des multimètres ont un testeur de transistors intégré qui, en général, fait du bon travail.

N'oubliez pas non plus qu'un transistor se comporte comme deux diodes, l'identification de la base est donc assez facile. Les transistors comme les Darlington et les transistors de puissance sont plus difficiles à tester, car ils nécessitent des tensions base-émetteur plus élevées ou des courants de collecteur plus intenses pour effectuer un test valable. Mais pour l'amateur, un test fonctionnel de base « bon/pas bon » suffit.

MOSFET

Les MOSFET sont devenus omniprésents, et un simple testeur de transistors ne pourra pas les tester. La plupart d'entre eux nécessitent des tensions grille-source de 5 V et plus pour être complètement passants. Heureusement, il existe un moyen facile de réaliser un test fonctionnel d'un MOSFET en utilisant uniquement une alimentation à limitation de courant (ou une alimentation et une résistance, ou encore mieux une ampoule électrique) et vos doigts ou une résistance de grande valeur (un mégohm ou plus). Le fait de placer la résistance (ou vos doigts) entre la grille et le plus devrait l'allumer et entre la grille et la source devrait l'éteindre. Étant donné que les MOSFET ont une capacité de grille appréciable, l'allumage et l'extinction peuvent prendre une seconde ou plus. Utilisez 12 V pour appliquer une tension de grille suffisante, et limitez le courant à 100 mA ou moins, ou vous risquez de vous brûler les doigts si le MOSFET n'est pas sur un dissipateur thermique !

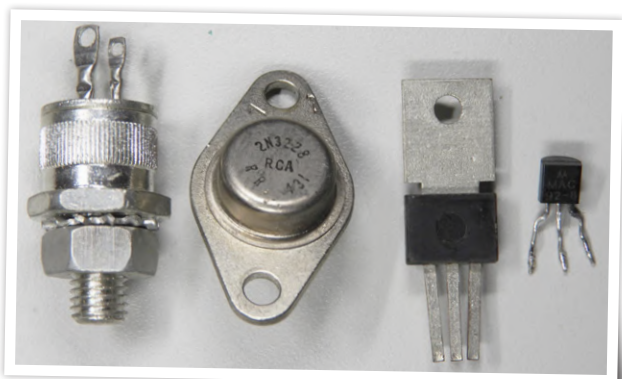


Figure 2. Un simple testeur de semi-conducteurs de puissance teste les MOSFET, les transistors, les thyristors et les triacs.

Thyristors/SCR/triacs

Pour ces derniers, vous pouvez utiliser la même configuration que pour les MOSFET ci-dessus, mais vous aurez besoin d'une résistance plus petite pour la grille – environ 470 Ω à 1 k Ω devrait convenir. Lorsque vous connectez la grille, via la résistance, à l'anode ou à l'alimentation positive, le SCR doit s'allumer (absorber du courant) et le rester même après la suppression de la commande de la grille. Vous devez interrompre la connexion de l'anode pour le désactiver. Idem pour les triacs, mais ils fonctionneront avec l'alimentation dans les deux sens. Les exigences de test pour les MOSFET, les transistors et les SCR sont similaires, et vous pouvez construire vous-même un testeur rudimentaire qui les testera tous. Voyez celui que j'ai construit à la **figure 2**.

VDR

Les résistances dépendant de la tension (VDR) et autres protections contre les surtensions telles que les types à gaz et les diodes P6KE font partie des composants les plus difficiles à tester. Une VDR de 460 V ne commencera à conduire qu'à partir de 500 V. Mais si vous avez accès à un mégohmmètre ou à un testeur d'isolement d'électricien, vous pouvez les vérifier rapidement.

La plupart des testeurs d'isolement ont trois tensions d'essai : 250 V, 500 V et 1000 V. Un VDR de 460 V ne devrait donner ni déviation ni lecture à 250 V, puis montrer une petite déviation à 500 V et une faible résistance à 1000 V. Les résultats peuvent varier en fonction du modèle de testeur utilisé, mais ils devraient être suffisants pour un test bon/pas bon. Les VDR qui ne fonctionnent pas sont soit en court-circuit, soit en circuit complètement ouvert, et ce test montrera l'un ou l'autre de ces états. Et gardez vos doigts loin de ces tensions !

Fusibles réarmables ou résistances CTP

Il s'agit de composants qui, en général, ont une faible résistance (quelques ohms), mais qui, lorsqu'ils sont traversés par un courant supérieur au courant nominal, chauffent et passent à une résistance

élevée, à priori maintenue jusqu'à ce que la source de la surintensité disparaisse. Testez-les en les plaçant sur une alimentation à limitation de courant et en augmentant lentement le courant. Au départ, la tension à leurs bornes sera faible, mais à mesure que le courant augmente, leur résistance augmente et la tension à leurs bornes augmente assez rapidement. En utilisant cette technique, vous pouvez avoir une idée de leur courant nominal (qui sera inférieur au point qui les fait passer en haute résistance). Et ne vous brûlez pas les doigts ; ils augmentent leur résistance en chauffant !

Interrupteurs DIP

Les commutateurs DIP usagés peuvent avoir certains éléments défectueux, il faut donc les tester avant de les utiliser. Vous pouvez tester n'importe quel interrupteur avec un multimètre bien sûr, mais si vous en avez beaucoup à tester, vous pouvez fabriquer un testeur d'interrupteurs DIP avec un support de circuit intégré, des résistances et des LED pour indiquer si chaque pôle de l'interrupteur est ouvert ou fermé. Vous pouvez aussi l'utiliser pour tester les LED. Placez une LED à l'endroit où l'un des pôles de l'interrupteur devrait se trouver et si elle est bonne, cette LED ainsi que celle du testeur devraient s'allumer. Voyez le modèle que j'ai construit à la **figure 3**.

Circuits intégrés

Ceux-ci sont plus difficiles à tester, car il en existe de nombreuses variétés. Mais vous pouvez effectuer des tests simples. Les amplificateurs opérationnels ont généralement le même brochage, vous pouvez donc fabriquer un testeur en le configurant pour qu'il fonctionne comme un multivibrateur astable avec quelques LED sur la sortie. Cela fonctionnera également pour de nombreux comparateurs, mais sachez que certains comparateurs n'ont qu'un simple transistor en sortie qui ne fournira pas de courant.

Vous pourriez également réaliser des testeurs pour les amplificateurs opérationnels doubles et quadruples. Utilisez un support de bonne qualité – un de type ZIF (à force d'insertion nulle) si votre budget vous le permet. Vous pouvez faire la même chose pour le très apprécié 555 – le faire fonctionner comme un astable avec des LED sur la sortie.



Figure 3. Un testeur de commutateurs DIP et de LED bricolé qui teste ici un commutateur DIP à 4 voies et une paire de LED.

Programmeur et testeur de circuits intégrés numériques TL866



Source: David Ashton

Le TL866 est avant tout un programmeur pour une très large gamme de composants. À ce titre, il semble raisonnablement efficace et on en trouve de nombreuses analyses, y compris celle d'Elektor [2]. Je vais toutefois me concentrer sur sa capacité à tester les circuits intégrés logiques.

Le TL866-II – la version actuelle – se présente dans un boîtier en plastique avec un support ZIF à 40 broches de bonne qualité, quelques LED et une connexion USB. J'ai dû télécharger le logiciel à partir du site du fabricant – ce qui ne m'enchantait pas, mais c'est le site du fabricant, donc le risque est minime. Le TL866 a un bon manuel PDF imprimable. Le logiciel est assez bien conçu et vous pouvez trouver rapidement la plupart des composants et des fonctions.

On trouve l'item *Logic Test* dans l'onglet *Device* du menu principal. Il fait apparaître une longue liste de types de circuits intégrés CMOS 4000 et TTL de la série 74 parmi lesquels choisir, et même quelques périphériques Intel 8080. J'ai passé une bonne demi-heure avec mes boîtes de circuits intégrés CMOS et TTL à en essayer un bon nombre et le TL866 a testé la plupart d'entre eux sans problème. Cependant, les 74LS21 ont systématiquement donné lieu à une erreur – un peu décevant pour un circuit intégré aussi élémentaire. Mais maintenant, la partie amusante. Le logiciel du TL866 affiche la séquence de test qu'il utilise, et vous pouvez la copier et la modifier (bien que vous deviez l'enregistrer avec un nouveau nom). En regardant la fiche technique du 7421 et la séquence de test programmée, j'ai remarqué que les broches 3 et 11 avaient « H » (niveau haut attendu) marqué pour ces broches. Ces broches ne sont pas utilisées, elles doivent donc être ignorées. En changeant ces broches en « x » (ignorer), le test a fonctionné et j'ai testé avec succès tous mes 74LS21 sans aucune erreur. Je suis maintenant sûr de pouvoir tester à peu près tous les circuits intégrés numériques, même si je dois écrire mes propres séquences de test pour certains d'entre eux. D'autres circuits intégrés ont donné des erreurs, principalement des multivibrateurs monostables qui seraient difficiles à tester sans composants de synchronisation.

Pour tester les circuits intégrés CMS, vous pouvez trouver des jeux d'adaptateurs CMS vers DIL sur eBay et d'autres sites, et vous pouvez même acheter un TL866 avec divers adaptateurs inclus pour moins de 50 € – un testeur très pratique pour un bon prix, et un bon ajout à la collection de matériel de test de tout amateur.

zone D

Astuces, bonnes pratiques et autres informations pertinentes

C'est plus difficile avec les circuits intégrés numériques. Si vous devez en tester souvent, le mieux est d'acheter un testeur approprié. Il existe un appareil appelé TL866 [2] (voir encadré), généralement disponible à moins de 50 €, qui est un programmeur d'EPROM, mais qui teste également les circuits intégrés logiques. Il y a plusieurs modèles disponibles, alors regardez les spécifications avant d'acheter. Le TL866II est la version actuelle.

Ils sont équipés d'un support ZIF pour la plupart des tailles de circuits intégrés de type traversant, mais sont souvent livrés avec des adaptateurs pratiques pour tester également divers composants en boîtier CMS. Certains modèles permettent la programmation ICSP si cela vous est utile, mais les plus récents ne programment pas les très vieilles EPROM (pour un ancien comme moi, c'est important !). Pour le simple test de circuits intégrés numériques, rien de tout cela n'est très important. Vous pouvez aussi écrire vos propres programmes de test, pour les CI qui ne sont pas dans la base de données ou pour lesquels les programmes inclus ne fonctionnent pas (il y en a quelques-uns).

Résonateurs à quartz

Les résonateurs à quartz et les oscillateurs font partie des composants les plus courants que vous trouverez lors du démontage d'un appareil professionnel. Généralement, ces composants sont marqués avec leur fréquence et un simple test bon/pas bon est nécessaire. Cela peut être fait assez simplement. Pour vérifier la fréquence, ou pour tester les résonateurs qui ne sont pas marqués, un fréquencemètre sera nécessaire. Vous n'avez besoin que de deux portes inverseuses pour faire osciller un quartz, et d'une autre pour détecter qu'il oscille. Vous pouvez utiliser la 4^e porte d'un circuit intégré à quatre portes comme tampon pour alimenter un fréquencemètre.

Les horloges en temps réel utilisent souvent des quartz de 32 768 Hz, mais les quartz des microcontrôleurs sont généralement de l'ordre du MHz. Vous aurez donc peut-être besoin de deux oscillateurs pour vérifier tous les quartz. Ajoutez alors un support DIP pour vérifier les oscillateurs ; vous avez juste besoin d'une alimentation et d'une connexion à la sortie. C'est plus difficile avec les quartz et les oscillateurs CMS. Un adaptateur de test est nécessaire pour eux.

Relais

On utilise encore beaucoup les relais. Pour l'isolement et la gestion des courants élevés, ils sont imbattables. Testez les contacts à l'aide d'un multimètre équipé d'un buzzer de continuité qui indiquera une faible résistance. Faites fonctionner le relais avec votre alimentation réglée sur la tension appropriée et vérifiez que les contacts s'ouvrent ou se ferment. Si vous avez beaucoup de relais d'un même type, il peut être utile de fabriquer un banc d'essai pour eux, en utilisant une charge de quelques ampères (par ex. une lampe de voiture) pour éliminer ceux dont les contacts ont une résistance élevée.

Autres outils

Si vous ne pouvez vous offrir qu'un seul bon appareil de test, prenez un multimètre de qualité, avec test des transistors, capacité et autres grandeurs (fréquence, température...). Il sera vraiment utile jusqu'à ce que vous puissiez acheter ou construire un instrument de test meilleur. L'une des choses les plus utiles à acheter si vous travaillez sur des composants CMS est une sonde de type pince (fig. 4). Elle se branche sur votre multimètre et vous permet de tester très rapidement et facile-



Figure 4. Sonde à pince pour multimètre. (Source : Sparkfun)

ment les résistances, les condensateurs, les inductances et les diodes CMS. Elles sont fabriquées par Sparkfun et se trouvent partout. Si vous effectuez beaucoup de tests de CMS, vous pouvez trouver des pinces avec appareil de mesure intégré. Le DT71 par ex. est disponible dans la boutique Elektor (voir l'encadré) avec un banc d'essai en [3]. Peak fabrique la gamme de testeurs de composants portatifs « Atlas », saluée par de bonnes critiques et proposée à un prix raisonnable. Nous recommandons leur « analyseur perfectionné pour semi-conducteurs DCA75 Pro » (voir encadré) qui identifiera à peu près n'importe quel semi-conducteur à 2 ou 3 bornes. Si vous avez besoin de faire des tests plus poussés que les tests simples décrits ci-dessus, cela vaut la peine d'investir dans un instrument de test comme celui-ci. Elektor propose depuis peu le testeur de composants multifonctions Joy-IT LCR-T7 [5] qui, bien que n'offrant pas les spécifications du testeur de Peak, est d'un niveau exceptionnel pour environ un quart du prix. Une loupe de bureau ou à main avec éclairage LED et/ou un microscope USB sont pratiques pour vérifier la valeur, la polarité ou l'orientation des petits composants.

On a mentionné une alimentation dans de nombreux tests ci-dessus. Au minimum, une alimentation variable 0-30 V avec limitation de courant de 10 mA (ou moins) à 500 mA (ou plus) conviendra.

L'utilisation de ces techniques simples et de quelques appareils de test bricolés vous facilitera grandement la vie si vous réutilisez des composants pour vos projets. ◀

210279-04

Contributeurs

Texte et photographies : David Ashton
Rédaction : Clemens Valens
Mise en page : Harmen Heida
Traduction : Denis Lafourcade

Des questions, des commentaires ?

Contactez Elektor (redaction@elektor.fr).

Banc d'essai : analyseur perfectionné pour semi-conducteurs DCA75 de Peak



Il y a quelque temps, je suis tombé sur l'analyseur perfectionné pour semi-conducteurs DCA75 de Peak. J'étais un peu hésitant, je l'admets, car je me demandais comment un si petit testeur avec seulement deux boutons pouvait

valoir ce prix-là (un peu plus de 100 €). Mais je n'aurais pas pu être plus agréablement surpris. Il fait tout ce qu'il prétend et il est très simple à utiliser. L'interface est tout simplement géniale et ne nécessite pratiquement aucune intervention de votre part. Vous connectez votre composant, dans le sens que vous voulez, et le DCA75 identifie ce que vous avez connecté, vous en donne quelques caractéristiques de base et vous en indique le branchement.

J'ai eu beau essayer, je n'ai pas pu tromper le DCA75. L'un des premiers composants que j'ai analysés était un « transistor » 0V8F de mon article sur l'identification des composants [1]. Je n'avais pas été capable de les identifier ou de les tester – je pensais que ce pouvaient être des FET. Le DCA75 les a instantanément identifiés comme des triacs. Ce seul fait

m'a convaincu que le DCA75 vaut son pesant d'or !

Le DCA75 a quelques limitations mineures – il ne peut pas tester les Zeners et les régulateurs de tension de plus de 8 ou 9 V – et n'identifie pas certains des semi-conducteurs les plus étonnants comme les transistors unijonction, mais ce ne sont pas des inconvénients majeurs pour l'utilisateur moyen.

En utilisant un PC, le câble USB et le logiciel inclus, le DCA75 peut également produire les courbes caractéristiques de nombreux composants. Dans l'ensemble, c'est un appareil de test très polyvalent à ajouter à votre arsenal. Il existe un petit frère, le DCA55, qui coûte environ la moitié du prix, mais je vous conseille d'acheter le DCA75. Je ne saurais trop recommander ce testeur.



PRODUITS

- **Analyseur perfectionné pour semi-conducteurs Atlas DCA75 Pro de Peak**
www.elektor.fr/17567
- **Brucelles numériques DT71 de Miniware**
www.elektor.fr/19422
- **Testeur de composants multifonctions LCR-T7 de Joy-IT**
www.elektor.fr/19709
- **Kit du LCR-mètre 2 MHz d'Elektor**
www.elektor.fr/19883

LIENS

- [1] « Identification des composants », D. Ashton, Elektor 03-04/2022 : www.elektormagazine.fr/210024-04
- [2] « Banc d'essai : programmeur MiniPro TL866A » : www.elektormagazine.fr/news/yg-review-minipro-tl866a-programmer
- [3] « Banc d'essai : DT71 de Miniware - brucelles de mesure numériques », H. Baggen, Elektor 07-08/2021 : www.elektormagazine.fr/210182-04
- [4] « Analyseur perfectionné pour semi-conducteurs PEAK Atlas » : www.elektormagazine.fr/news/analyseur-pour-semi-conducteurs-Atlas-DCA-75
- [5] « Testeur multifonctions JOY-IT LCR-T7 » : www.elektormagazine.fr/news/testeur-multifonctions-joy-it-lcr-t7

réduction de la consommation d'énergie de votre repousse-taupes

Grâce au remplacement du 555 par un ATtiny13



Gerhard Dürr (Allemagne) et Luc Lemmens (Elektor)

Les taupinières peuvent ruiner une pelouse entretenue avec amour en un rien de temps. Si vous protégez votre précieux jardin avec un repousse-taupes électronique, cette mise à niveau matérielle peut multiplier par dix la durée de vie de ses piles !

Si vous avez un jardin avec une pelouse, surtout si vous habitez à la campagne, vous courez le risque d'être désagréablement surpris par la partie visible de l'activité des taupes : les taupinières. La bonne nouvelle c'est que la terre de votre jardin est sans doute propre et fertile. Vous ne verrez qu'exceptionnellement les taupes elles-mêmes, qui ne quittent que rarement les galeries qu'elles creusent en quête de vers et d'insectes, leur menu favori. La terre qu'elles déplacent pendant qu'elles creusent doit bien sûr aller quelque part, et elles la poussent vers le haut, formant ainsi les monticules qui signent leur présence. C'est tellement peu apprécié que de nombreuses méthodes ont été mises au point pour leur barrer la route ou pour les déloger une fois installées. L'une des solutions est ce que l'on appelle le repousse-taupes : un appareil qui produit des vibrations (sonores) pour empêcher ou mettre

fin à ces activités souterraines. Il s'agit d'un tube pointu que l'on insère dans le sol, pourvu à son extrémité d'une source sonore (généralement un vibreur électronique à courant continu) censée chasser les taupes et autres vermines comme les campagnols et les rats.

Certains fabricants prétendent que les répulsifs antiparasites qu'ils fabriquent produisent des vibrations *ultrasoniques*. Peut-être en existe-t-il, mais les spécifications de la plupart des appareils que nous avons évalués précisent qu'ils produisent des sons dont les fréquences se situent dans la zone inférieure des kilohertz ou même plus bas, bien en dessous de la gamme des ultrasons. Dans la plupart des cas, le terme repousse-taupes sonore (sans ultra) sera plus approprié. Le son est émis pendant environ une demi-seconde, à des intervalles de quelque 30 s. Ces impulsions seraient si désagréables pour une

taupe qu'elles la feraient décamper et/ou rester à distance. Certaines personnes sont absolument convaincues de l'efficacité de ces appareils, d'autres prétendent qu'il s'agit d'une arnaque qui ne profite qu'aux seuls fabricants et revendeurs ; nous ne nous prononcerons pas sur ce point. Quelques réflexions sur l'éloignement des taupes en général sont présentées dans l'encadré. Dans cet article, nous n'aborderons qu'un moyen de réduire la consommation d'énergie d'une variante produisant du son et fonctionnant sur piles.

Comment ça marche

Ce type de répulsif contient un circuit minuteur et une sorte de sonnette constituée le plus souvent d'un vibreur piézoélectrique, mais nous avons également trouvé des dispositifs avec des moteurs vibrants, comme ceux utilisés dans les téléphones portables. En électronique, « circuit minuteur » est synonyme de « 555 », peut-être le plus célèbre des circuits intégrés, qui vient tout juste de fêter son 50^e anniversaire. Il y a toujours des exceptions, mais le concepteur du circuit proposé ici a également découvert un appareil dans lequel le 555 est utilisé comme multivibrateur bistable. La majeure partie de l'énergie sert à produire le son, mais il en faut un peu pour maintenir le 555 actif lorsque le vibreur est silencieux. Bien qu'une modification de ce circuit puisse procurer une économie d'énergie et donc prolonger la durée de vie des piles (par exemple, en remplaçant un 555 standard par une version CMOS ou en modifiant les valeurs de ses composants externes), le concepteur a choisi une approche différente. Le circuit complet de la minuterie est remplacé par un petit microcontrôleur qui reste en mode veille la plupart du temps ; sa minuterie de chien de garde le réveille environ toutes les 30 s et fait retentir le vibreur pendant une seconde, puis le repasse en mode veille. La consommation de courant est réduite de près d'un facteur dix : de 4 mA avec le circuit original de la minuterie à 555 à une moyenne de 430 µA avec le microcontrôleur, soit une durée de vie des piles portée de moins de 200 jours à environ 1450 jours ! En pratique, cette durée sera plus courte, en raison de facteurs tels que l'autodécharge des piles, mais il s'agit tout de même d'une grande amélioration par rapport à la consommation de courant du circuit original.

Le nouveau matériel

Le schéma du circuit à microcontrôleur est représenté sur la **figure 1**. IC1, un ATtiny13 de Microchip (anciennement Atmel), est le cœur du circuit de la minuterie. Sa mémoire de programme interne peut être flashée avec une interface de programmation ISP connectée à K2. Si le microcontrôleur est programmé avant d'être inséré dans son support, on peut omettre ce port à six broches.

Le circuit est alimenté par des piles, dans ce cas quatre piles de type D (LR20) en série – c'est-à-dire 6 V au total – connectées entre K1 et K3 (les bornes positive et négative, respectivement). Avec quatre piles de 1,5 V neuves, à vide ou avec une charge très faible, la tension, de l'ordre de 6,5 V, dépasse la tension d'alimentation maximale absolue de 5,5 V de l'ATtiny13. D1 est une LED rouge standard (de préférence pas un type à haut rendement) qui a une tension directe d'environ 1,3 V, même à un courant direct aussi bas que 1 µA ; la tension d'alimentation du microcontrôleur s'en trouve réduite à une valeur sûre, bien en dessous du maximum de 5,5 V. D1 peut aussi être constituée de deux ou trois diodes standard 1N4148 en série. Un régulateur de tension linéaire n'est pas une bonne solution, car il ajouterait sa propre consommation. C1 et C2 sont des condensateurs de découplage pour l'alimentation du microcontrôleur.

Le vibreur est récupéré du matériel d'origine et connecté aux broches

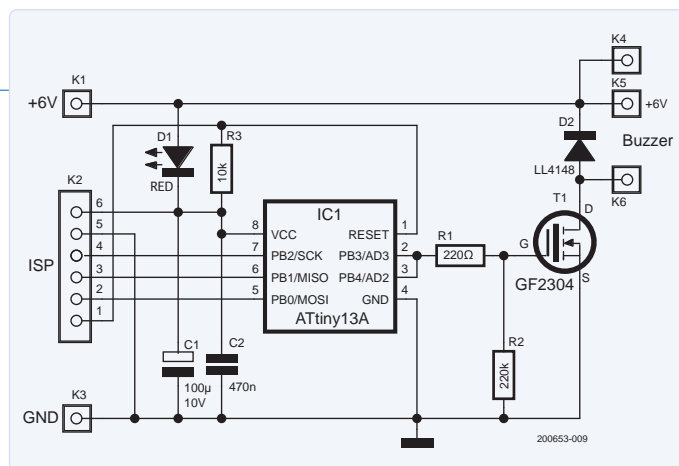


Figure 1. Le schéma du circuit de mise à niveau du minuteur.

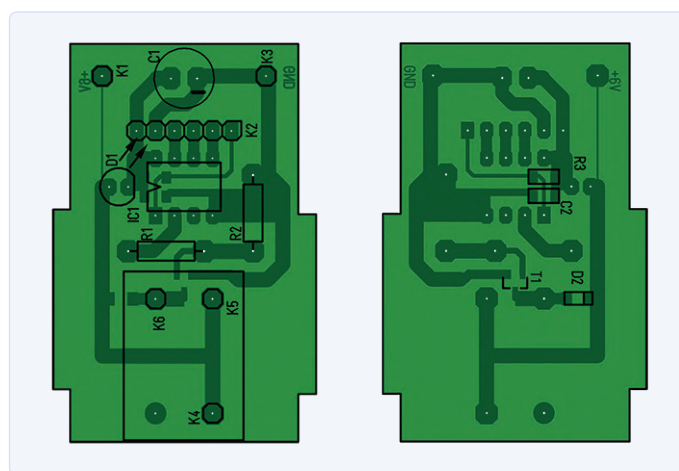


Figure 2. Implantation sur le circuit imprimé.

K4/K5 (positif) et K6 (négatif). Certains vibreurs sont inductifs et susceptibles de produire des pics, notamment lors de la mise hors tension. D2 est une diode de roue libre qui protège le transistor T1 contre ces pics. Le choix de ce transistor de commutation n'est pas du tout critique. Le courant traversant le vibreur étant inférieur à 10 mA, un BC547 standard peut convenir avec 4,7 kΩ pour R1 et 10 kΩ pour R2. N'importe quel NMOS-FET à niveau logique fera également l'affaire, en remplaçant R1 par un fil de liaison et en supprimant R2.

Il y a même un circuit imprimé

Le schéma a été conçu avec une version gratuite de Target3001! et l'auteur a même réalisé un petit circuit imprimé pour le passage au microcontrôleur (**fig. 2**). Les fichiers de conception sont disponibles en téléchargement sur la page Elektor Labs de ce projet [1]. Compte tenu du grand nombre de marques différentes de repousse-taupes disponibles sur le marché, ce serait une pure coïncidence si, comme sur les photos des auteurs (**fig. 3**), le circuit imprimé s'adaptait exactement au vôtre. Si vous désirez modifier le circuit imprimé, voilà une belle occasion de vous faire la main sur Target3001! avec ce projet relativement simple, mais refaire le schéma et le circuit imprimé avec un autre programme de CAO ne serait pas plus compliqué. Il y a quelques composants CMS soudés sur la face inférieure en cuivre du circuit imprimé qui peuvent facilement être remplacés par des composants traversants, même sans changer le circuit imprimé.



Figure 3a. Vue d'ensemble.

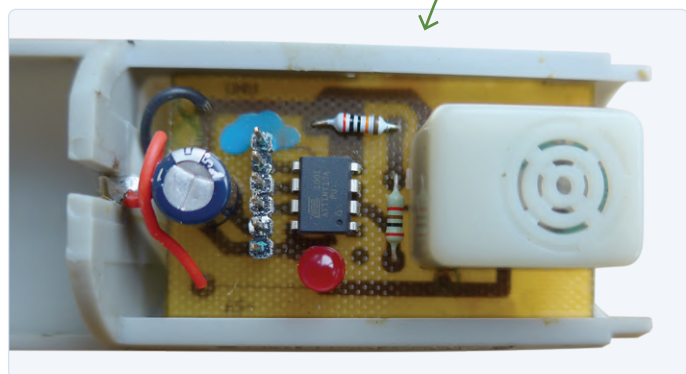


Figure 3b. Le nouveau circuit imprimé installé dans le repousse-taupes initial.

Rien ne fonctionne sans micrologiciel

Le code source (**listage 1**) pour l'ATtiny13 est développé sous AVR Studio 4.19 avec le compilateur AVR-GCC installé. Bien que cet EDI soit dépassé, il est toujours disponible en téléchargement, mais il ne sera pas difficile d'adapter le code source à des outils de développement plus récents. Le code source, le projet AVR Studio et le fichier HEX peuvent être téléchargés depuis la page Elektor Labs du projet [1]. Ce dernier vous permet de flasher la mémoire de programme de l'ATtiny13 sans avoir à compiler. Les fusibles de configuration de ce

microcontrôleur resteront à leur valeur d'usine (c'est-à-dire qu'ils n'ont pas besoin d'être programmés).

Le programme est essentiellement constitué d'instructions en assembleur insérées dans un module écrit en C. Après la mise sous tension, les broches GPIO sont configurées ; les minuteries, le convertisseur A/N, les résistances de rappel et le comparateur analogique sont désactivés et le mode veille est réglé sur *Powerdown*. Le code entre ensuite dans une boucle sans fin (`while(1)`), qui effectue les opérations suivantes :

- Le vibreur est activé.
- Le chien de garde est réglé pour produire une interruption après 128k cycles, avec une horloge de 128 kHz, cela donne un délai d'interruption d'une seconde (environ).
- L'interruption est activée et l'ATtiny passe en mode *Powerdown* avec la commande assembleur `sleep`.
- Après une seconde, l'interruption du chien de garde réveille le microcontrôleur de son état *Powerdown*
- Le vibreur est éteint et l'interruption du chien de garde est portée à son intervalle maximum de 1024k cycles, c'est-à-dire 8 s. Comme nous voulons que le repousse-taupes soit silencieux pendant environ une demi-minute, le chien de garde doit produire trois interruptions avant que le `while(1)` ne recommence avec l'activation du vibreur.

La fin des taupes ?

Cette mise à niveau du repousse-taupes prolonge la durée de vie des quatre piles de type D, mais ne modifie pas l'efficacité de l'appareil lui-même. Comme nous l'avons mentionné, nombreux sont les débats entre ceux qui croient en l'efficacité des repousse-taupes et ceux qui pensent ou même sont convaincus que leur effet est nul. La profusion de marques et de revendeurs suggère que beaucoup de gens y croient ou s'interrogent sur leur efficacité. En tout cas, c'est un remède qui ne nuit pas à l'environnement. La chose la plus négative que nous ayons trouvée, ce sont des témoignages de voisins gênés par les sons doux et répétitifs de ces répulsifs. ❏

200653-04

Des taupes dans votre jardin ? Que faire ou ne pas faire ?

Si vous cherchez un remède sur l'internet contre les taupes dans votre jardin ou pour leur barrer l'accès à votre propriété, vous serez surpris de voir tout ce qui a été inventé au fil des ans pour leur pourrir la vie. Les experts ne donnent pas cher de l'efficacité de nombre de ces solutions, mais vous pourrez en être certain si votre magnifique pelouse est soudain défigurée par des taupinières.

Nous ne sommes pas du tout des experts en taupes, mais, d'après les informations dont nous disposons, le mieux est de ne rien faire du tout. S'il y a des taupinières, c'est qu'une taupe a déjà creusé son terrier et celui-ci ne s'agrandit généralement que lorsqu'il n'y a plus assez de nourriture. Si ces mini-montagnes vous dérangent, la meilleure chose à faire est d'enlever la terre avec précaution à l'aide d'une pelle. Si vous les cognez ou les aplatissez, il y a de fortes chances que l'animal répare l'effondrement de ses galeries en produisant de nouvelles taupinières.



LISTE DES COMPOSANTS

Résistances

R1 = 220 Ω
R2 = 220 kΩ
R3 = 10 kΩ

Condensateurs

C1 = 220 µF/25V
C2 = 470 nF

Semi-conducteurs

D1 = LED rouge
D2 = LL4148
T1 = GF2304 (voir texte)
IC1 = ATtiny13A (style DIL-8)

Divers

K2 = barrette à 6 broches

Contributeurs

Conception : Gerhard Dürr

Texte : Gerhard Dürr,

Luc Lemmens

Illustrations : Gerhard Dürr,

Patrick Wielders, Luc Lemmens

Rédaction : Jens Nickel, C. J. Abate

Mise en page : Giel Dols

Traduction : Helmut Müller

Des questions, des commentaires ?

Contactez Elektor
(redaction@elektor.fr).



PRODUITS

➤ **Programmeur avec accessoires EXA-Prog de Diamex**
www.elektor.fr/19912



Listage 1. Code source complet.

```
// Controlling the molechaser-buzzer with the watchdog-timer
// * MC ATtiny13A, all fuse bytes default
// * PB3 and PB4 HIGH: buzzer ON
// * Configurable with watchdog-timer-prescaler
// * Selected: 128K-cycles(~1s) ON, 3*1024K-cycles(~24s) OFF
// * Current consumption: powerdown ~6µA, buzzing ~7.2mA, average 0.43mA
#include <avr/io.h>
#ifndef F_CPU
#warning "Defining F_CPU 1.2MHz"
#define F_CPU 1200000UL
#endif
#include <avr/interrupt.h>
int main(void) {
    uint8_t i=0;
    DDRB = 1<<PB0|1<<PB1|1<<PB2|1<<PB3|1<<PB4;
    PRR = 1<<PRTIM0|1<<PRADC; // Power Reduction Timer0, ADC
    MCUCR = 1<<PUD|1<<SE|1<<SM1|0<<SM0; // Pullup Disable, Sleepmode: Powerdown
    ACSR = 1<<ACD; // Analog Comparator Disable
    while(1) {
        i = 0;
        PORTB |= 1<<PB3|1<<PB4; // buzzer ON
        // change Watchdog-Interrupt to 1s (measured: 1.14s @ 4.5V, 22°C)
        MCUSR = 0; // clear Reset-Flags
        WDTCSR = 1<<WDCE |0<<WDE; // Watchdog Change Enable
        // Watchdog Timer Prescaler: 128K-cycles(~1s), Watchdog Timer Interrupt Enable
        WDTCSR = 1<<WDTIE|0<<WDE|1<<WDP2|1<<WDP1;
        sei();
        asm volatile("sleep::");
        asm volatile("wdr::"); // Watchdog Reset, wait for WDT-ISR(~1s)
        PORTB &= ~(1<<PB3|1<<PB4); // buzzer OFF
        // change Watchdog-Interrupt to 3*8s: (measured: 27.9s @ 4.5V, 22°C)
        asm volatile("wdr::"); // Watchdog Reset
        MCUSR = 0; // clear Reset-Flags
        WDTCSR = 1<<WDCE|0<<WDE; // Watchdog Change Enable
        // Watchdog Timer Prescaler: 1024K-cycles(~8s), Watchdog Timer Interrupt Enable
        WDTCSR = 1<<WDTIE|0<<WDE|1<<WDP3|1<<WDP0;
        while(i < 3) { // wait 24s
            i++;
            asm volatile("sleep::");
            asm volatile("wdr::"); // Watchdog Reset, wait for WDT-ISR(~8s)
        }
    }
}
EMPTY_INTERRUPT(WDT_vect); // awaking from powerdown-mode
```

LIEN

[1] Ce projet sur Elektor Labs : <https://bit.ly/3JWCXBk>



interrupteur crépusculaire DeLux

Une solution pour une commutation de haute précision,
commandée par la lumière

Clemens Valens (Elektor)

On trouve de nombreux interrupteurs crépusculaires pour environ 10 €, mais pour la plupart le changement d'état s'effectue quelque part à la tombée du jour. Parfois, les applications exigent plus de précision et de maîtrise que ne le permettent ces interrupteurs bon marché.

Vous avez besoin d'un contrôle de luminosité précis au lux près ? Si oui, ce projet est pour vous.

Il existe de très nombreux modèles d'interrupteurs crépusculaires et la plupart fonctionnent parfaitement dans l'application pour laquelle ils ont été conçus. Il s'agit généralement d'allumer un éclairage lorsque le niveau de luminosité ambiante descend en dessous d'un certain seuil et de l'éteindre à nouveau lorsque le niveau de luminosité augmente. On y ajoute aussi parfois une minuterie.

Vous pourriez penser qu'il est possible de couvrir toutes les applications avec ces modèles, mais ce n'est pas le cas. La raison en est qu'ils manquent tous de précision.

Basés sur une LDR (photorésistance) ou un phototransistor, ils ont tendance à commuter quelque part dans la zone crépusculaire. Or, les niveaux de luminosité varient bien plus que cela.

La luminosité est subjective

Pour les humains, l'intensité ou la luminosité de la lumière du jour est assez constante. Bien sûr, nous remarquons les variations dues aux nuages et au soleil, mais nous n'y sommes pas très sensibles. Cela s'explique par la réponse logarithmique de l'œil à la luminosité. Par temps nuageux,

la luminosité peut varier entre 5 000 et 10 000 lux, alors qu'elle nous semble presque identique. Au soleil, la lumière peut atteindre des niveaux supérieurs à 25 000 lux, ce que nous percevons évidemment, mais sans réaliser que la luminosité est au moins trois fois plus élevée. Les plantes, quant à elles, sont beaucoup plus sensibles à l'intensité lumineuse que les humains. Les agriculteurs le savent, et ils éclairent artificiellement certaines de leurs cultures, même pendant la journée, pour améliorer leur rendement. Par temps ensoleillé, cela n'est généralement pas nécessaire, mais en cas de nuages, cela peut être utile. Pour faire cela de manière économique, ils ont donc besoin de commutateurs commandés par la lumière capables de détecter les différences de luminosité avec plus de précision qu'une LDR ou un phototransistor.

Il existe aujourd'hui des capteurs lumineux qui convertissent directement la luminosité en une valeur en lux avec des résolutions allant jusqu'à 16 bits. Certains de ces capteurs ne mesurent pas seulement les lux, mais aussi l'intensité des rayons UV et de la lumière blanche. Avec un tel



capteur, il est assez facile de construire un interrupteur de haute précision commandé par la lumière.

Capteur de lumière ambiante de haute précision

Un capteur de lumière courant est le VEML7700 de Vishay. Il s'agit d'un capteur de lumière ambiante de haute précision avec interface I²C que l'on peut trouver monté sur un petit module pour quelques euros. De la même famille, on peut également citer le VEML6075, un capteur de lumière UVA et UVB également avec interface I²C.

Grâce à son interface numérique I²C, le capteur ne nécessite pas de convertisseur analogique-numérique et peut être connecté directement à la plupart des microcontrôleurs. Ses données de sortie sont disponibles dans deux registres de 16 bits : lumière ambiante (également appelée « ALS », « Ambient Light Sensor ») et lumière blanche. La lumière blanche couvre un large spectre allant de 250 nm à 950 nm. Le spectre ALS est beaucoup plus étroit, d'environ 450 nm à 650 nm, car il est optimisé pour la perception humaine. Les plantes n'étant pas des êtres humains, la sortie à utiliser dépendra de l'application. La sensibilité du capteur est également importante. Le VEML7700 a une résolution de 0,005 lx par bit et un niveau de détection maximal de 167 000 lx (0,01 lx pour le minimum). Une gamme aussi étendue nécessiterait des valeurs sur 25 bits, alors que le composant ne délivre que 16 bits ; c'est pourquoi on peut spécifier une valeur de sensibilité (parfois appelée « gain ») pour ramener les choses dans la gamme. Sa haute sensibilité permet d'utiliser le capteur derrière des surfaces à faible transmittance (c.-à-d. sombres) et d'obtenir néanmoins de bons résultats.

Pour les conditions de faible luminosité, le capteur dispose d'un paramètre d'intégration allant jusqu'à 800 ms. Enfin, il est possible de définir des seuils bas et hauts qui peuvent déclencher des interruptions, ce qui facilite la création d'alarmes ou permet une commutation automatique.

Associons-le à un ESP32

Le module VEML7700 choisi pour mes expériences a été acheté chez Adafruit. Une bonne plateforme pour l'utiliser est le thermostat connecté ESP32 [1] (alias Automator, voir [2]). Le module possède

cinq broches, mais comme il dispose de deux options d'alimentation, nous n'en avons besoin que de quatre. Ces quatre broches ont le même ordre que sur le connecteur de l'écran OLED du thermostat, et nous pouvons donc brancher le module sur K9. Assurez-vous que la broche VIN ne soit pas connectée et que le module soit orienté vers le haut (contrairement à la façon dont l'écran OLED serait branché, voir **figure 1**).

Le logiciel avec ESPHome

Après avoir connecté le capteur à l'ESP32, nous devons ajouter du logiciel pour que tout fonctionne. Comme l'objectif est une sorte d'interrupteur commandé par la lumière, un choix logique serait d'utiliser une plateforme domotique et ma préférée est ESPHome. Elektor a publié plusieurs projets utilisant ESPHome [3][4], et donc vous savez probablement comment l'utiliser et le configurer, mais ce projet introduit un concept que je n'avais pas encore traité : la création d'un capteur personnalisé. Si vous ne connaissez pas ESPHome, je vous recommande de lire et de regarder d'abord [3] et [4].

Il nous faut un capteur personnalisé

ESPHome prend en charge de nombreux capteurs, mais (au moment de la rédaction de cet article) pas le VEML7700. Il connaît d'autres capteurs de lux, mais pas celui-ci. Cependant ce n'est pas un problème, car ESPHome propose une méthode pour ajouter votre propre capteur. Pour cela il

faut écrire du code C++, ce qui rend les choses un peu plus complexes.

Comme précédemment (voir [3] et [4]), nous devons déclarer une section *sensor* dans le fichier de configuration de ESPHome pour ce capteur. On doit maintenant préciser *custom* dans le champ *platform* du capteur. Cela indique à ESPHome que vous allez lui fournir tous les détails.

Suit une section *lambda* qui consiste en quelques lignes de code C++ pour indiquer à ESPHome d'inscrire un capteur de notre conception (que nous avons nommé *veml7700*). Nous devons également spécifier le(s) flux de données que notre capteur produit. Il a trois sorties ici : ALS, lux et lumière blanche. L'ordre est important et doit être respecté chaque fois qu'elles sont référencées ailleurs dans le fichier YAML.

Nous continuons avec des déclarations YAML normales pour spécifier davantage les sorties des capteurs. Cela se fait avec une section *sensors* (au pluriel !) où nous pouvons spécifier pour chaque flux de données son nom, ses unités et le nombre de décimales à utiliser. L'ordre est le même que dans l'instruction *return* de la section *lambda*.

Seul le flux de lux a des unités (« lx »), et aucun flux ne nécessite de décimales, donc nous les mettons à zéro.

La dernière chose à faire est d'indiquer à ESPHome où trouver le pilote pour le capteur personnalisé. Nous le faisons dans la section *esphome* en haut du fichier de configuration où nous ajoutons un fichier (*veml7700.h*) à la sous-section *includes*. Sur

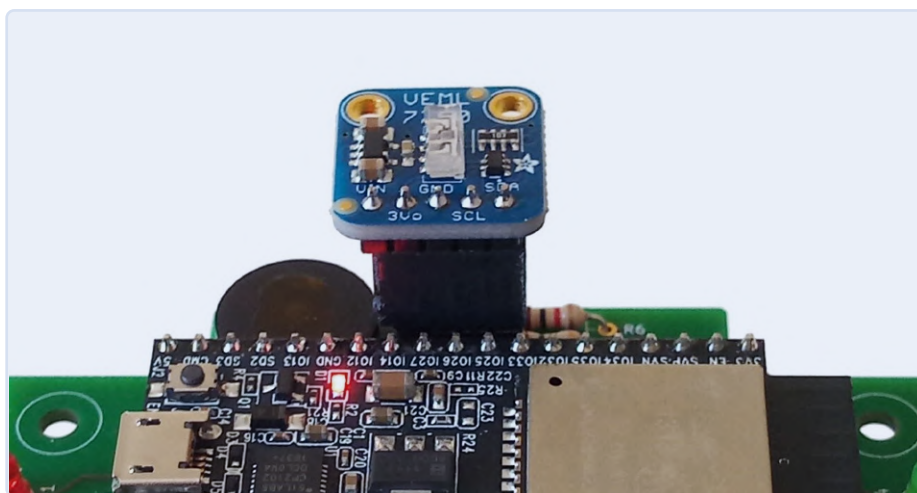


Figure 1. Le module VEML7700 à 5 broches se branche sur le connecteur à 4 voies K9. Sa broche VIN n'est pas connectée, même si elle en a l'air.

Install espiic.yaml

```
Library Manager: Installing esphome/AsyncTCP-esphome @ 1.2.2
Library Manager: AsyncTCP-esphome @ 1.2.2 has been installed!
Library Manager: Installing Adafruit_VEML7700_Library
Library Manager: Adafruit_VEML7700_Library @ 1.1.1 has been installed!
Library Manager: Installing dependencies...
Library Manager: Installing Adafruit_BusIO
Library Manager: Warning! More than one package has been found by Adafruit_BusIO requirements:
- adafruit/Adafruit_BusIO @ 1.9.1
- merbanian/Adafruit_BusIO @ 1.7.3
Library Manager: Please specify detailed REQUIREMENTS using package owner and version (showed above) to avoid name
conflicts
Library Manager: Adafruit_BusIO @ 1.9.1 has been installed!
Library Manager: Installing Hash
Library Manager: Already installed, built-in library
Dependency Graph
|-- <AsyncTCP>-esphome @ 1.2.2
|-- <Adafruit_VEML7700_Library> 1.1.1
|   |-- <Adafruit_BusIO> 1.9.1
|   |   |-- <Wire> 1.0.1
|   |   |-- <SPI> 1.0
|   |   |-- <Wire> 1.0.1
|   |   |-- <SPI> 1.0
|   |-- <ESP8266> 1.0
|   |-- <ESP8266> 1.0
|   |-- <ESP8266> 1.0
|   |-- <WiFi> 1.0
```

Figure 2. ESPHome installe automatiquement la bibliothèque Adafruit.

le système qui exécute ESPHome, ce fichier doit se trouver dans le même dossier que le fichier YAML du périphérique.

La partie difficile

C'est fini ? Pas vraiment. Vient maintenant la partie la plus difficile, qui consiste, bien sûr, à créer le pilote pour le capteur personnalisé. Ce pilote doit être conforme aux normes ESPHome pour les composants (un capteur est un composant), c'est-à-dire qu'il doit fournir certaines fonctions qu'ESPHome attend de ses composants, et il doit communiquer avec le capteur lui-même.

Pour la deuxième partie, nous pouvons compter sur Adafruit qui, en plus de fabriquer le module VEML7700, a également créé une bibliothèque Arduino pour celui-ci [5]. Notre pilote doit fournir l'interface entre ESPHome et la bibliothèque Arduino.

Installation d'une bibliothèque tierce

ESPHome fournit un mécanisme pour ajouter des bibliothèques communautaires *open source* : il suffit d'ajouter le nom exact (!) de la bibliothèque à la sous-section *libraries* de la section *esphome*. Dans notre cas, il s'agit de « Adafruit VEML7700 Library ». Maintenant, lorsque ESPHome traitera le fichier de configuration, il installera d'abord la bibliothèque (s'il ne l'a pas déjà fait, voir **figure 2**) avant de tout compiler. Dans votre pilote personnalisé, il suffit d'inclure la bibliothèque comme n'importe quelle autre.

Je ne me suis pas aventuré plus loin sur ce terrain, donc je ne connais pas les critères qu'une bibliothèque tierce doit remplir pour être utilisée de cette façon. Pour plus

de détails, consultez la documentation de platformIO, la chaîne d'outils utilisée par ESPHome.

Un croquis Arduino encapsulé

Notre pilote est une classe C++ qui doit fournir au moins un constructeur et une fonction nommée *update*. Nous avons également besoin d'une fonction *setup* afin d'initialiser le pilote Adafruit. Les fonctions *setup* et *update* de notre classe font ce qui serait normalement fait dans les fonctions *setup* et *loop* d'un croquis Arduino typique utilisant le pilote d'Adafruit. En fait, notre classe encapsule un croquis Arduino incluant des variables globales et y ajoute des éléments spécifiques à ESPHome. Pour ESPHome, nous devons insérer des appels à *publish_state* à la fonction *update* pour chaque flux de données (ALS, lux et lumière blanche). Cela permettra de mettre les données à la disposition du reste du monde. L'ordre des appels doit être le même que dans l'instruction de retour de la sous-section *lambda* de la section *sensors* (voir ci-dessus).

Comme notre classe hérite de la classe *PollingComponent* qui elle-même hérite de la classe *Component*, vous disposez d'autres fonctions qui peuvent vous servir. Un composant *polling* est un composant qui est appelé périodiquement, et la fréquence d'appel peut être spécifiée (par ex. dans notre constructeur). Reportez-vous à [6] pour plus de détails.

Ajout du bus I²C

Il ne reste plus maintenant qu'à connecter le capteur au bus I²C au sein du logiciel (c.-à-d. créer une connexion logique entre les deux, puisque nous avons déjà une connexion physique). La bibliothèque Adafruit utilise

à cet effet la bibliothèque *Wire* d'Arduino. Dans le fichier YAML de ESPHome, nous ajoutons donc une section *i2c* pour que ESPHome sache qu'elle est nécessaire.

L'ESP32 possède deux bus I²C avec des signaux SDA et SCL qui peuvent être connectés à presque toutes les broches de la puce. ESPHome prend donc en charge plusieurs bus I²C avec des broches librement assignables, ce qui permet de spécifier où va quoi. Mais la bibliothèque Arduino *Wire* ne supporte qu'un seul bus I²C, alors comment lui dire d'utiliser le bus que vous voulez ? Eh bien, vous ne pouvez pas, car il utilisera toujours le bus par défaut. Dans ESPHome, le bus I²C par défaut est le premier bus spécifié dans la section *i2c*. Il aurait été bien de pouvoir spécifier un bus I²C en utilisant son ID, et ESPHome vous aurait volontiers permis de le faire, mais la bibliothèque Arduino *Wire* sous-jacente ne le permet pas.

C'est fini

C'est ici que se termine cet article. Si l'Automator est programmé par ESPHome avec le code décrit ci-dessus, vous obtiendrez un dispositif qui peut être autonome ou commandé par un assistant domotique comme Home Assistant. Il n'y a pas de règles d'automatisation dans le fichier YAML d'ESPHome, donc sans assistant, il se contentera de mesurer l'intensité de la lumière ambiante. Les règles d'automatisation peuvent être ajoutées au fichier YAML lui-même, ou créées dans Home Assistant, par exemple. Pour une description du reste du code YAML, qui n'a rien de particulier, référez-vous à [3] et [4]. Le fichier de configuration YAML et le code C++ peuvent être téléchargés depuis la page web de cet article [7]. ◀

210190-04

Contributeurs

Conception, texte et photographies :

Clemens Valens

Rédaction : **Jens Nickel, C. J. Abate**

Mise en page : **Harmen Heida**

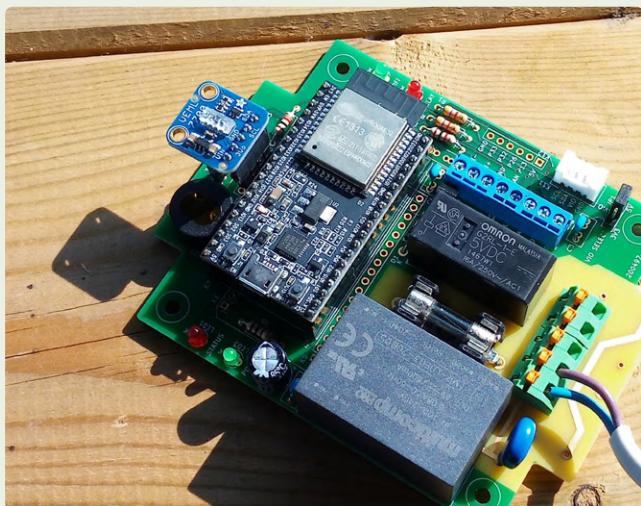
Traduction : **Denis Lafourcade**

Des questions, des commentaires ?

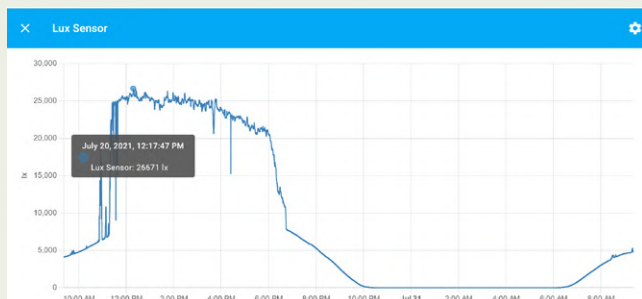
Envoyez un courriel à l'auteur (clemens.valens@elektor.com) ou contactez Elektor (redaction@elektor.fr).



Quelques résultats

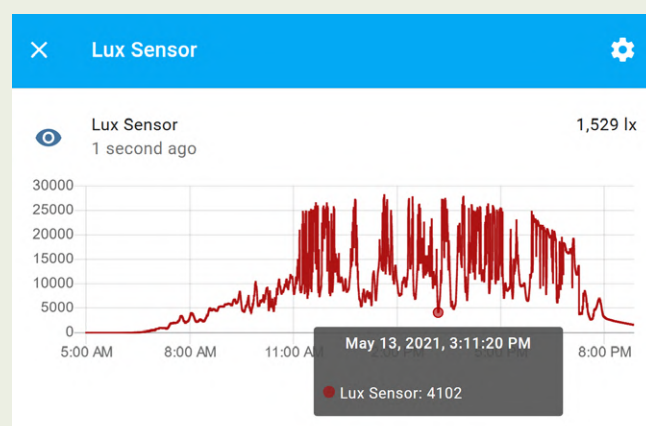


Le capteur de luminosité placé en plein soleil d'été. À midi, avec une sensibilité réglée sur 0,125 et un temps d'intégration de 100 ms, le capteur a indiqué des valeurs d'environ 48 000 pour l'ALS et d'un peu plus de 30 000 pour la lumière blanche. Ces valeurs correspondent à des intensités lumineuses d'environ 22 000 lx. À 10h30, dans les mêmes conditions, j'ai mesuré 16 800 lx, soit une différence de 5 200 lx. Cependant, à mes yeux, la luminosité était à peu près la même dans les deux situations.



Évolution de la luminosité par une journée ensoleillée de juillet. Le « rebond » sur la gauche est dû à l'ombre d'un arbre. De midi à 18h, la luminosité diminue lentement, mais on ne le remarque pas vraiment. Les creux sont causés par les nuages. L'ombre de

la maison commence à couvrir le capteur à 18h, ce qui explique la forte baisse. Ensuite, l'intensité lumineuse continue de diminuer jusqu'à la nuit. Pour les humains, 5 000 lx est déjà une forte luminosité.



Par un jour partiellement nuageux en mai, les niveaux de luminosité peuvent varier entre 4 000 et 27 000 lx environ.

Tous les graphiques ont été créés à l'aide de Home Assistant à une latitude de 47°N.



PRODUITS

- > **ESP32 DevKitC**
www.elektor.fr/18701
- > **Écran OLED, I²C, 128x64 de 2,5 cm**
www.elektor.fr/18747
- > **Livre en anglais, « Getting Started with ESPHome », Koen Vervloesem**
www.elektor.fr/19738

LIENS

- [1] « Thermostat connecté à ESP32 », Y. Bourdon, Elektor 09-10/2021 : www.elektormagazine.fr/200497-04
- [2] Elektor Automator : www.elektormagazine.fr/labs/automator
- [3] Guide Home Assistant et ESPHome (en anglais) : www.elektormagazine.fr/labs/how-to-home-assistant-esphome
- [4] « la domotique, c'est facile avec... », C. Valens, Elektor 09-10/2020 : www.elektormagazine.fr/200019-02
- [5] Bibliothèque Adafruit VEML7700 : https://github.com/adafruit/Adafruit_VEML7700
- [6] ESPHome sur GitHub : <https://github.com/esphome>
- [7] Téléchargements pour cet article : www.elektormagazine.fr/210190-04

les défis de la commercialisation des **solutions IdO**

Problématiques de sécurité, d'évolutivité et de concurrence

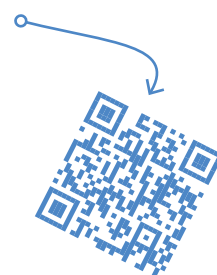


Figure 1. La Commission européenne a examiné si les grands acteurs du secteur des assistants vocaux, interface utilisateur préférée des maisons intelligentes, étouffaient la concurrence. (Source : Shutterstock/Gorodenkoff)

Stuart Cording (Elektor)

L'Internet des objets (IdO) existe depuis plus de 20 ans, et une multitude de technologies sans fil ont émergé pour favoriser son déploiement. À la maison, les assistants vocaux sont devenus la principale interface utilisateur de nombreux appareils intelligents. Malgré ces progrès, plus d'un tiers des projets IdO ne dépassent jamais la phase de démonstration de la faisabilité. En outre, la Commission européenne craint que l'absence de concurrence dans certains espaces d'application n'empêche l'entrée sur le marché des entreprises de l'UE. Alors, quels sont les véritables défis, et comment se déroule le déploiement d'une solution IdO en Europe ?

Si vous souhaitez vous familiariser avec les technologies de l'Internet des objets (IdO), vous pouvez trouver facilement un grand nombre de projets adaptés. Il suffit de rechercher « IdO » avec votre plateforme de prototypage préférée, et vous serez submergé par des pages de projets, de plateformes en cloud, de comparaisons technologiques et de listes d'idées. Elektor est également une excellente source d'informations. Depuis que le terme IdO a été introduit il y a plus de 20 ans, notre site web a rassemblé plus de 600 articles sur l'IdO ou liés à ce sujet [1].



Cependant, il existe un fossé entre la démonstration du concept de base d'une solution IdO à l'aide d'une plateforme prototype et le déploiement d'une version réelle. Le rapport *IoT Insights* de Microsoft [2] est basé sur des entretiens avec plus de 3 000 professionnels de l'IdO en 2021. Ils ont constaté que 35 % des projets IdO connaissaient un échec pendant la phase d'essai ou de démonstration de faisabilité, soit une augmentation de 5 % par rapport à l'enquête réalisée un an plus tôt. Le coût élevé de la mise à l'échelle (déploiement) est la raison la plus citée pour expliquer l'échec à ce stade. Parmi les autres causes, figurent le grand nombre de plateformes à tester, la multitude de cas d'utilisation à étudier et le manque de ressources. Une autre étude réalisée par Cisco [3] a révélé que seulement 26 % des entreprises participantes pensaient que leurs initiatives IdO avaient été couronnées de succès. Les réponses à cette étude ont montré que, même si la plupart des projets IdO semblent bien conçus sur le papier, ils s'avèrent plus complexes que prévu dans la pratique.

En dépit de ces impressions négatives sur l'IdO, il existe des secteurs où cette infrastructure fait d'énormes progrès et génère des revenus croissants.

La Commission européenne examine le secteur de l'IdO grand public

Les citoyens de l'UE ont accueilli favorablement les solutions IdO grand public proposées ces dernières années. À tel point qu'un rapport sur le marché des maisons intelligentes publié par Statista [4] prévoit que le chiffre d'affaires associé doublera, passant d'environ 17 milliards d'euros en 2020 à environ 38,1 milliards d'euros en 2025. Préoccupée par le fait que la concurrence dans ce secteur pourrait être étouffée, la Commission européenne a réalisé une enquête dans le cadre de sa stratégie numérique [5]. Le rapport, publié en janvier 2022, a recueilli les contributions des fabricants de technologies portables, d'appareils grand public connectés utilisés dans la maison intelligente et de ceux qui fournissent des services par le biais de ces appareils intelligents. En outre, la Commission a demandé la contribution d'organismes de normalisation. Cependant, on constate à la lecture que de nombreux points de leur analyse sont consacrés aux assistants vocaux (*voice assistants* - VA) qui constituent l'interface utilisateur de nombreux produits et services IdO (figure 1).

Après avoir analysé le paysage de la maison intelligente, le rapport montre clairement qu'en Europe, Google Assistant de Google, Alexa d'Amazon et Siri d'Apple sont les principaux assistants vocaux polyvalents. D'autres sont disponibles, mais ils ont tendance à disposer de fonctionnalités plus limitées et à se concentrer sur la prise en charge d'un seul produit ou de l'application d'un fournisseur de services. Selon ZDNet, parmi les solutions des trois grands groupes, c'est Amazon qui offre le plus haut niveau de compatibilité, en prenant en charge environ 7 400 marques [7]. En comparaison avec Google qui en reconnaît environ 1 000, Apple reste le plus exclusif avec une cinquantaine de marques.

Peu de place pour les nouveaux venus dans le domaine des assistants vocaux

Avec de tels acteurs industriels puissants et déjà présents sur le marché, il y a peu de place pour les nouveaux venus, et la courbe technologique pour développer un VA compétitif est importante. Ainsi, si vous voulez tirer parti du contrôle vocal pour votre solution IdO, vous devez jouer selon les règles établies par les trois géants. Une approche alternative serait d'accorder une licence à un VA. Cependant, certains fabricants ont signalé que les conditions d'octroi des licences restreignaient leurs choix. Ces limitations pouvaient aller de l'exclusivité ou de restrictions empêchant l'utilisation simultanée de plusieurs VA jusqu'à des licences imposant l'intégration d'autres types de logiciels ou d'applications, ce qui signifie que la technologie VA ne peut pas être utilisée de manière autonome.

Une autre grande préoccupation est l'accès aux données. En tant que tiers recourant à un VA, vous n'avez qu'un accès limité aux données collectées. Le fournisseur de VA a accès aux enregistrements audio et sait également combien de tentatives ont échoué pour exécuter les commandes sélectionnées pour votre appareil. Cependant, votre équipe n'aura pas cet accès, et il vous faudra plus probablement attendre les retours des utilisateurs pour découvrir que votre choix de commandes vocales est sous-optimal par rapport au groupe plus expansif que celui utilisé pour les tests. En outre, comme le fournisseur de VA peut analyser tout ce qui est dit, il pourrait éventuellement utiliser ces données pour développer une solution concurrente à la vôtre ou tirer parti de l'expérience des utilisateurs fournie par votre solution IdO pour améliorer ses propres services.

Un autre problème se pose lorsque le fournisseur de VA propose également des services de publicité. En théorie, les données vocales fournies par vos utilisateurs pourraient aider le fournisseur à améliorer le ciblage de la publicité en fonction de la population représentée par votre base de clients. Enfin, il y a la diminution de la reconnaissance de la marque et de l'expérience. Votre solution soigneusement élaborée est à la merci des VA. Toute modification importante, telle que la voix utilisée, le mot d'appel ou même le déploiement de fonctions entraînant une baisse du nombre d'utilisateurs et aura inévitablement des conséquences pour vous.

Le rapport examine également de nombreux autres domaines pertinents, notamment les interfaces de programmation d'applications (API), les normes, l'interopérabilité, le déséquilibre de puissance entre de nombreux développeurs tiers de dispositifs IdO et les grands fournisseurs de services de plateforme en cloud, ainsi que les clauses de résiliation de contrat. Le rapport ne contient aucune recommandation. Toutefois, les conclusions précisent que le contenu du rapport contribuera à la stratégie de normalisation de la Commission et enrichira le débat de la législation sur les marchés numériques (DMA).



Si l'on examine le paysage de l'IdO, il est clair que les opportunités commerciales s'offrent à vous, et ce, qu'il s'agisse de solutions pour les consommateurs ou à l'industrie.

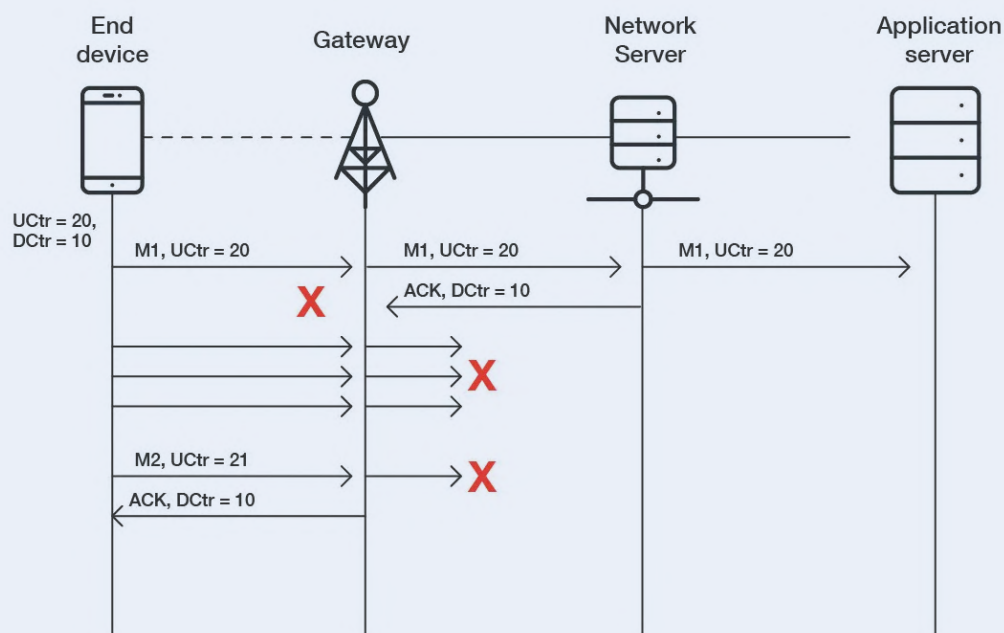


Figure 2. Les chercheurs ont découvert une attaque par déni de service (DoS) dans LoRa 1.0. En répétant un précédent transfert de données réussi, un nœud LoRaWAN est empêché d'envoyer d'autres paquets de données. (Source : Trend Micro)

Les préoccupations en matière de sécurité inquiètent les concepteurs de solutions IdO

À l'examen des données recueillies dans le rapport *IoT Insights* de Microsoft, on constate que la sécurité de l'IdO est en tête de la liste des préoccupations, notamment pour ceux qui envisagent des solutions IdO. 29 % d'entre eux ont indiqué que les risques de sécurité associés les empêchaient d'utiliser davantage l'IdO. Le rapport explique également qu'environ un tiers des organisations sont préoccupées par les risques de sécurité de l'IdO, notamment les violations de données. Pour lutter contre cela, l'externalisation est considérée comme le meilleur moyen de gagner en sérénité. Même si de nombreux ingénieurs connaissent la notion de « sécurité par l'obscurité » (non-divulgaration d'informations relatives au procédé considéré), peu d'entre eux sont qualifiés dans ce domaine pour garantir qu'une solution est sécurisée de bout en bout contre les attaquants. Et si les fournisseurs de semi-conducteurs proposent toute une gamme de solutions de sécurité monochips, les développeurs doivent encore comprendre comment les utiliser correctement afin d'éviter de créer par inadvertance de nouvelles failles de sécurité.

Au cours des dernières décennies, les solutions de réseaux étendus à basse consommation (LPWAN) telles que LoRa et Sigfox se sont imposées comme des technologies IdO sans fil clés prenant en charge les communications longue distance. Avec des portées de plusieurs dizaines de kilomètres, elles constituent une alternative au réseau cellulaire sans fil tel que le LTE Cat-M1 et le NB-IoT grâce à leurs performances exceptionnelles à basse consommation pour de faibles volumes de données [8]. Pour autant, à quel point sont-ils sécurisés ?

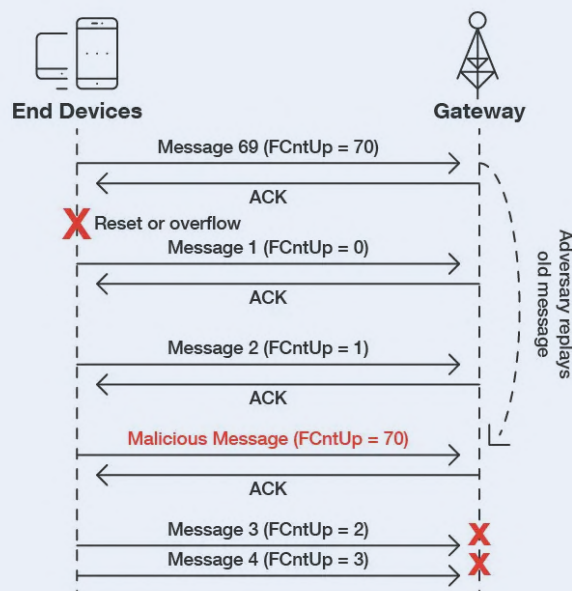
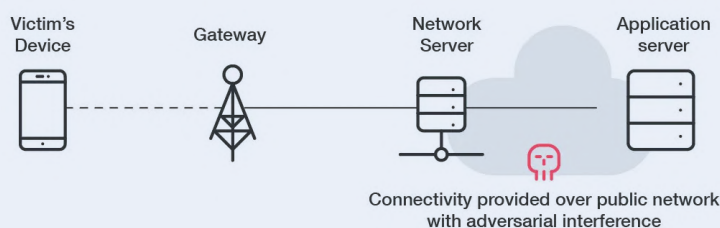
LoRaWAN à la loupe

Les protocoles LoRa et LoRaWAN ont fait l'objet d'une attention particulière de la part de la communauté de la sécurité et du

piratage. Sébastien Dudek, de Trend Micro, entreprise spécialisée dans les solutions de sécurité informatique, est l'un des nombreux chercheurs qui ont écrit de manière approfondie sur certains des problèmes potentiels. Dans une série de trois notes techniques (brief1, brief2, brief3), il évoque un ensemble de problèmes de mise en œuvre et d'attaques potentielles. Celles-ci vont du déni de service (DoS) (**figure 2**) et de l'écoute clandestine au basculement binaire (**figure 3**) et à la mystification des accusés de réception (**figure 4**). Les conséquences de ces attaques vont de l'impossibilité de communiquer avec les nœuds jusqu'à l'altération des données des applications, en passant par la réduction de l'autonomie de la batterie. Un grand nombre des vulnérabilités signalées ont été résolues entre les versions 1.0.2 et 1.1 de la norme LoRaWAN. Cependant, d'autres défis se posent lors de l'exploitation de nœuds LoRaWAN avec des passerelles utilisant différentes versions de la spécification. Dans de tels cas, il est nécessaire d'apporter des modifications pour assurer une rétrocompatibilité sécurisée entre les dispositifs finaux et le back-end, comme le souligne un article de 2018 de Tahsin C. M. Dönmez [12].

Le problème ne se limite pas au piratage de la liaison sans fil. Le risque que de mauvais acteurs volent et attaquent directement le matériel existe aussi. Sébastien Dudek étudie également cet aspect de la sécurité. Dans le cas du protocole LoRaWAN, de nombreuses solutions utilisent un microcontrôleur et un module sans fil de Semtech. Comme ceux-ci sont connectés via SPI, les données passant entre les deux peuvent être facilement acquises et analysées.

En outre, il faut également tenir compte de la sécurité du microcontrôleur lui-même. Une méthode d'attaque consiste simplement à extraire le micrologiciel de la mémoire flash, ce qui permet d'analyser le code. Si les clés de sécurité se trouvent également dans le programme, un attaquant peut les utiliser pour développer des nœuds qui usurpent des dispositifs finaux authentiques. Pour lutter



elektor mai/juin 2022 57

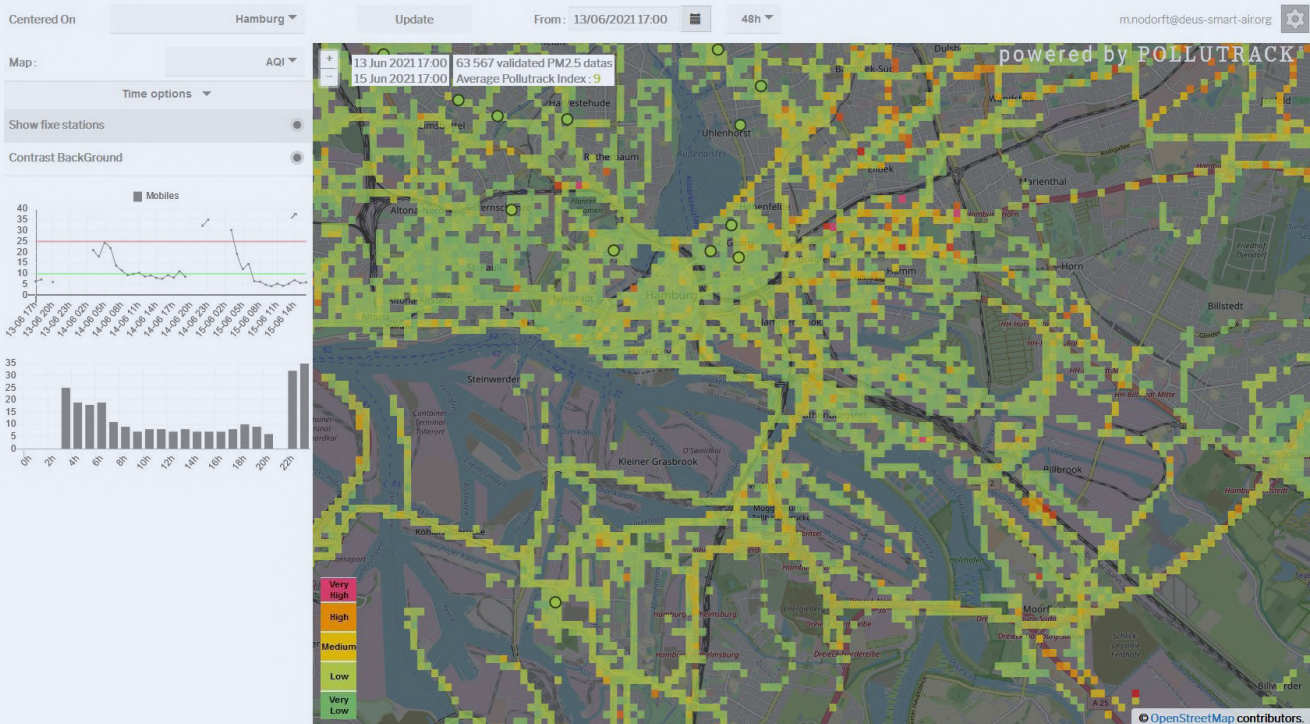


Figure 6. Un tableau de bord basé sur le cloud montre le niveau des polluants atmosphériques, comme illustré ici pour la ville allemande de Hambourg. Les autorités locales utilisent ces données pour élaborer des décisions sur les solutions de transport. (Source : DEUS POLLUTRACK)

Telekom [15], il est indiqué que la mise en œuvre de Sigfox et LoRaWAN « sans [un SE] peut [même] rendre inutile le chiffrement de bout en bout ». Le document explique qu'en revanche, le protocole NB-IoT bénéficie de fonctions de sécurité LTE éprouvées, telles que l'authentification et la génération, et l'échange de clés sécurisées. Cependant, il précise également que le chiffrement de bout en bout n'est pas standard et que, s'il est jugé nécessaire, il doit être discuté avec l'opérateur réseau.

Offrir des solutions IdO à l'échelle de la ville

Les problématiques relatives à la sécurité des réseaux LPWAN ont influencé les choix technologiques de DEUS POLLUTRACK Smart City GmbH i.G. pour sa plateforme IdO [16]. Leur équipe développe depuis plus de dix ans des réseaux de capteurs IdO pour surveiller les particules dans les villes. La technologie étant déployée dans plus de 15 villes européennes, elle permet aux responsables locaux des municipalités de prendre des décisions environnementales éclairées concernant la pollution atmosphérique. Leurs compteurs optiques de particules (OPC) brevetés sont capables de surveiller jusqu'à la classification des particules ultrafines (UFP) (moins de $0,1 \mu\text{m}$). Alors que les particules plus volumineuses, comme les PM_{10} , sont considérées comme dangereuses pour les poumons, les UFP peuvent pénétrer dans la circulation sanguine et passer à d'autres organes par l'air inhalé.

La technologie de capteurs de DEUS (figure 5) utilise une combinaison de capteurs fixes et mobiles, reliés à des tableaux de bord en back-end qui visualisent les données collectées. Des villes comme Marseille et Paris utilisent 40 capteurs fixes ainsi que 300 capteurs mobiles [17]. Les capteurs mobiles sont installés sur les véhicules des partenaires, tels que les fourgons de livraison de DPD, qui

circulent régulièrement dans la ville cible. Ces capteurs se réactualisent en fonction des données acquises par les capteurs fixes qu'ils croisent pour garantir la précision requise sur la base. Tout cela nécessite un choix de réseau LPWAN robuste, fiable et sécurisé. Le cofondateur Marc Nodorft a expliqué que les normes Sigfox et LoRaWAN ont l'une et l'autre été considérées lors des premières étapes du développement. Sigfox offrait une infrastructure de connectivité, ce qui simplifiait le déploiement du système, mais aucun des deux ne fournissait le débit de données requis. Le protocole LoRaWAN, à l'époque, n'était pas suffisamment sécurisé et, en l'absence de partenaires d'infrastructure dans les villes où la technologie devait être déployée, il était nécessaire de mettre en place des passerelles qui se connectaient au back-end via des réseaux cellulaires. La 4G et, plus tard, la 5G cellulaire, ont donc été choisies, résolvant ainsi les problèmes de couverture, de fiabilité et de sécurité selon les niveaux requis.

Marc Nodorft nous explique également que, bien qu'il existe de nombreuses solutions électroniques bon marché pour l'IdO, celles-ci ne sont pas suffisamment robustes pour un déploiement à long terme dans les environnements où leurs produits sont installés. C'est pourquoi le choix s'est porté sur un développement conforme aux normes industrielles, une autre considération pour ceux qui planifient leurs propres produits IdO.

Autre aspect, les activités de back-end, développées spécifiquement selon les besoins de leur implémentation IdO (figure 6). Pour progresser, il est nécessaire de prendre en charge des tableaux de bord de reporting open source pour permettre aux organismes publics utilisant le système et aux citoyens d'accéder aux données, ce qui nécessite un fournisseur de services dans le cloud. Et, bien que les choix soient nombreux, la sélection du fournisseur est


aussi importante que la solution technique. Il faut donc chercher un fournisseur capable d'offrir une assistance personnelle, bien au-delà d'un chatbot de service client impersonnel.

Avec mon expérience dans des déploiements IdO importants et ayant beaucoup appris sur les défis techniques, je me suis demandé quels autres conseils Marc Nodorft pourrait donner à ceux qui cherchent à mettre en œuvre des solutions IdO. « Nous sommes toujours restés fidèles à notre vision, répond-il, ce qui nous a souvent obligés à modifier notre approche. » Cela a impliqué l'évaluation de différentes technologies, la collaboration avec différents partenaires et la modification de la stratégie de vente sur leur chemin vers le succès.

Équipes d'experts et partenariats nécessaires

Si l'on examine le paysage de l'IdO, il est clair que les opportunités commerciales s'offrent à vous, et ce, qu'il s'agisse de solutions pour les consommateurs ou l'industrie. Cependant, du concept au déploiement, le parcours est parsemé de défis. Si les développeurs de systèmes embarqués peuvent être bien versés dans le développement de matériel et de micrologiciels, et peuvent même avoir de l'expérience dans les technologies sans fil, l'IdO et ses défis en matière de sécurité et d'évolutivité peuvent être trop importants pour qu'une organisation s'y attaque toute seule.

Selon le rapport de la Commission européenne, les grandes organisations dominent également les relations commerciales en matière de services et de plateformes. Les petits opérateurs et les jeunes pousses (start-up) auront du mal à obtenir le soutien dont ils ont

besoin dans ces relations asymétriques s'ils font cavalier seul. Sans aucun doute, l'expertise, au travers d'embauches ou de financements, est essentielle pour aller au-delà des exemples d'applications, des tableaux de bord de démonstration et des tests de services IdO. Enfin, il est vital de fixer votre vision, mais en restant agile dans tous les domaines de la mise en œuvre, depuis les choix technologiques jusqu'au marché visé, pour la concrétiser. 

220053-04

Contributeurs

Texte : **Stuart Cording**

Rédaction : **Jens Nickel, C. J. Abate**

Traduction : **Asma Adhimi**

Mise en page : **Harmen Heida**



Des questions, des commentaires ?

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

LIENS

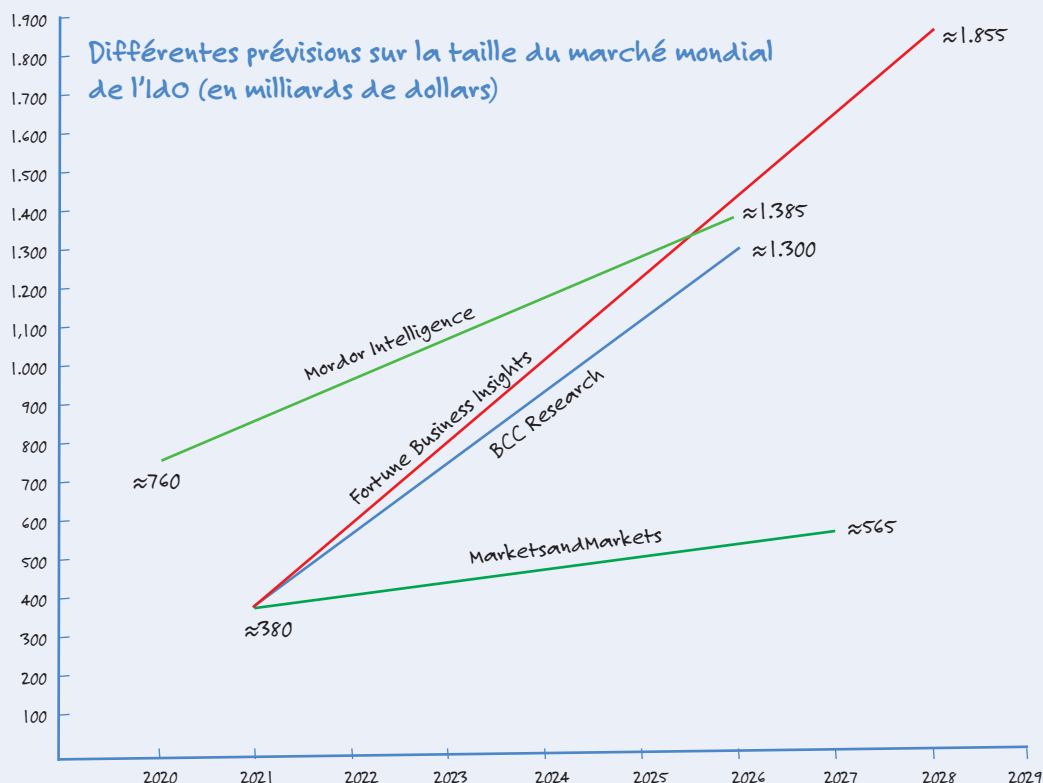
- [1] Articles d'Elektor sur l'IdO : www.elektormagazine.com/select/internet-of-things-iot
- [2] « IoT Insights, Edition 3 », Microsoft/Hypothesis, octobre 2021: <https://bit.ly/3rxMk3a>
- [3] « The Journey to IoT Value », Cisco, mai 2017: <https://bit.ly/3GzdJWS>
- [4] J. Lasquety-Reyes, « Smart Home - revenue forecast in Europe from 2017 to 2025 », Statista, juin 2021: <https://bit.ly/3LIgiuG>
- [5] « Sector inquiry into the Consumer Internet of Things », Commission européenne, janvier 2022 : <https://bit.ly/3Lgw9iE>
- [6] « Final report - sector inquiry into consumer Internet of Things », Commission européenne, janvier 2022 : <https://bit.ly/3B2Htu9>
- [7] «The best voice assistant», ZDNet, septembre 2021 :: <https://zd.net/3rxf6Rt>
- [8] L. Tan, « Comparison of LoRa and NB-IoT in Terms of Power Consumption », Institut royal de technologie KTH : <https://bit.ly/3JafsUb>
- [9] S. Dudek, « Low Powered and High Risk : Possible Attacks on LoRaWAN Devices », Trend Micro, janvier 2021 : <https://bit.ly/3rA02Tg>
- [10] S. Dudek, « Gauging LoRaWAN Communication Security with LoraPWN », Trend Micro, février 2021 : <https://bit.ly/3LhV0T5>
- [11] S. Dudek, « Protecting LoRaWAN Hardware from Attacks in the Wild », Trend Micro, mars 2021 : <https://bit.ly/3rxquge>
- [12] T.C.M. Dönmez, « Security of LoRaWAN v1.1 in Backward Compatibility Scenarios », Elsevier, 2018 : <https://bit.ly/3GtzKq0>
- [13] Page du produit Microchip - ATECC608A :: <https://bit.ly/3B7zIm5>
- [14] F.L. Coman et ass., « Security issues in internet of things : Vulnerability analysis of LoRaWAN, sigfox and NB-IoT », IEEE, juin 2019 : <https://bit.ly/3uwhUQX>
- [15] « NB-IoT, LoRaWAN, Sigfox : An up-to-date comparison », Deutsche Telekom AG, avril 2021 : <https://bit.ly/3uyUydj>
- [16] Site web de DEUS Pollutrack : <https://bit.ly/3sHL9O5>
- [17] Réseau de mesure des capteurs DEUS : <https://bit.ly/3Gzjbcc>

IoT : le nouveau Graal ?

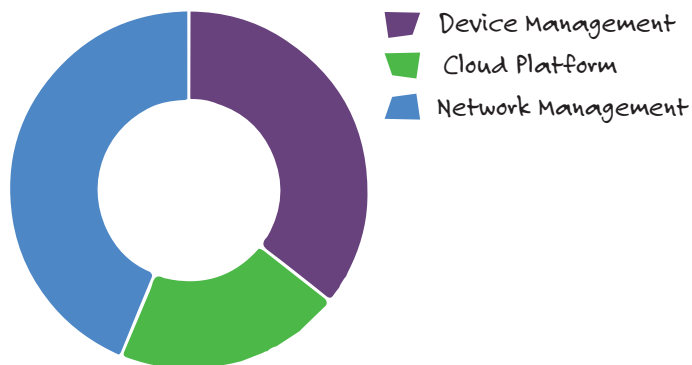


Qu'est-ce que l'IoT ? Nous savons tous ce que signifie IoT ou en français IdO. La question devrait plutôt être : que doit-on considérer comme l'Internet des Objets ? Le marché de l'IdO inclut-il le stockage de données en tant que service (*Dwaas, Data Warehouse as a Service*) ? L'IdO inclut-il la cybersécurité ? Ou ces secteurs sont-ils autonomes ? En dehors du débat sur l'inclusion et l'exclusion, il reste surprenant de constater à quel point les prévisions sur la taille du marché mondial de l'IdO peuvent être différentes. Les prévisions (oui, au pluriel) des sociétés d'études de marché sont si différentes que le verre n'est pas seulement à moitié plein ou à moitié vide, mais aussi complètement plein au point de déborder. Il doit y avoir quelque chose de proche du Graal ici.

(Sources : BCC Research, Fortune Business Insights, MarketsandMarkets, Mordor Intelligence)



Trois ingrédients ? Non, il y en a quatre.



Part du marché mondial de l'IdO, par secteur, en 2020

Quand on pense à l'IdO, on pense aux appareils et à leurs connexions à un ordinateur central. Mais une vision aussi simple n'est plus en phase avec la réalité. Outre un appareil, une connexion et un ordinateur, il existe un autre ingrédient : une plateforme dans le nuage, animée par des centres de données. Les appareils envoient leurs informations au nuage, où elles sont stockées et analysées. Un ordinateur du client se connecte au nuage pour obtenir les résultats finaux. Ce stockage de données en tant que service (*DWaaS*) ou cette plateforme en tant que service (*PaaS, Platform as a Service*) représente environ 20% du chiffre d'affaires mondial de l'IdO. Les services de plateforme dans le nuage devraient connaître une croissance d'environ 22% entre 2021 et 2026.

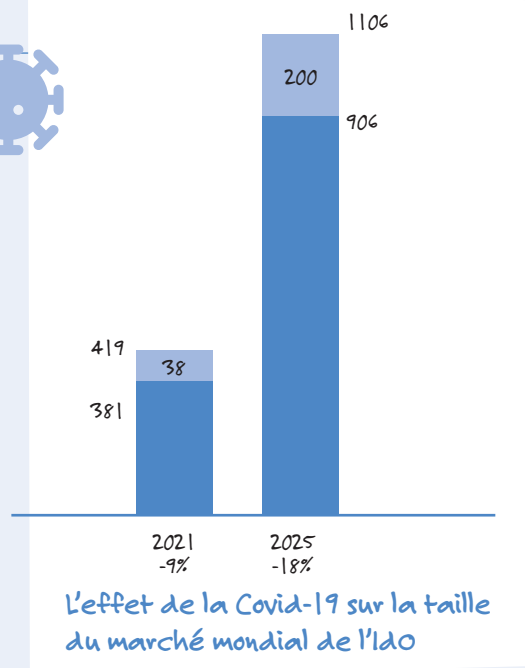
(Source : BCC Publishing, Fortune Business Insights)



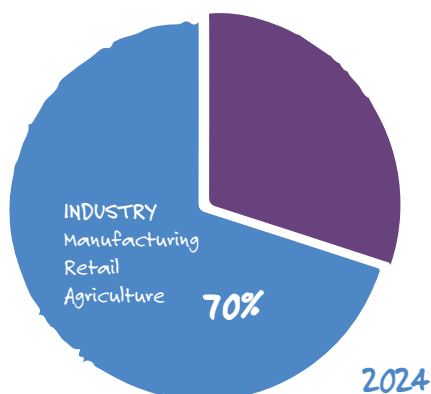
L'effet négatif de la Covid-19 augmente au fil des ans

Vodafone ne veut pas empêcher les entreprises de l'IdO de brosser un tableau optimiste de leurs attentes sur le marché. Cependant la société de télécommunications a effectué des recherches sur l'effet de la Covid-19 sur le marché mondial de l'IdO. La multinationale a conclu que les effets de la Covid-19 se feront sentir pendant un certain temps. Alors que l'année dernière, le marché mondial de l'IdO a perdu 9% de son chiffre d'affaires à cause de la pandémie, ce pourcentage doublera pour atteindre 18% en 2025. Bien que Vodafone soit conscient de l'ombre que projette la Covid-19 sur les années à venir, son rapport sur l'IdO indique clairement que le soleil sera au rendez-vous pendant cette même période.

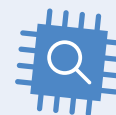
(Sources : Saft Batteries, Vodafone)



Contribution du secteur industriel au total des connexions IdO



À qui profitent les connexions de l'IdO ?



Quels secteurs de la société bénéficieront le plus de l'IdO ? Si l'on avait posé cette question il y a quelques années, la réponse aurait été : les consommateurs. Les appareils domestiques intelligents sont toujours très présents si l'on considère qu'en 2022, un ménage moyen en Amérique du Nord possède au moins neuf appareils IdO, allant de la caméra de la porte d'entrée au pilotage de la température. Toutefois, les applications industrielles (notamment le commerce de détail et l'agriculture) représenteront 70% de toutes les connexions IdO en 2024. Les 30% restants peuvent être répartis entre l'électronique grand public et les services publics comme le stationnement intelligent, les déchets intelligents, l'éclairage public intelligent et la gestion intelligente du trafic.

(Sources : Saft Batteries, Global Information, Inc.)

Pourquoi l'IdO fait-il battre les cœurs ?



Une chose est très claire : l'IdO fera de plus en plus partie de notre quotidien. Mais pourquoi ? Est-ce parce que les appareils IdO sont de moins en moins chers ? C'est peut-être vrai, mais cela pose bien sûr la question de savoir pourquoi les fournisseurs misent sur un marché de masse et peuvent baisser autant leurs prix. La réponse est en fait double : (1) l'IdO permet à ses utilisateurs de réaliser des économies substantielles et (2) il créera de nouvelles sources de revenus, également dans une très large mesure. En ce qui concerne les économies, on peut penser à une augmentation de la productivité des employés et du temps de fonctionnement des installations. La création de nouveaux flux de revenus ou l'élargissement des flux existants est une autre façon de gagner de l'argent.

(Sources : Saft Batteries, Vodafone)

Principaux avantages des déploiements de l'IdO

50%

Productivité des employés

42%

Temps de fonctionnement des installations (cohérence et fiabilité)

34%

Ont vu l'IdO générer directement de nouvelles sources de revenus

34%

Ont vu leurs sources de revenus existantes augmenter de 24% en moyenne.

« Je préfère quand même être en filaire »

Conseils pour développer une interface 1 Gbit/s dans un environnement industriel



Heinz Zenkner (collaborateur indépendant chez Würth Elektronik)

Les réseaux sans fil jouissent d'une grande popularité, notamment pour les applications industrielles. Toutefois, dans de nombreux cas un câblage robuste par Ethernet reste l'alternative la plus fiable et la plus sûre. Cet article montre comment une interface de 1 Gbit/s peut être facilement mise en œuvre.

Les capteurs et compteurs intelligents, qui utilisent des techniques de modulation et de codage efficaces avec de bonnes propriétés de propagation et de faibles bandes passantes permettent la mise en place de réseaux de capteurs industriels sans fil. Cependant, la plupart des cas d'utilisation considérés se limitent à des applications à faible débit de données. S'agissant d'applications, le débit réel dépasse rarement 1 Mbit/s.

Un réseau sans fil n'a pas de frontières fixes. À titre d'exemple, même de petits ajustements de la position de l'antenne du point d'accès peuvent avoir une influence considérable sur la puissance du signal dans les autres stations. Les murs, les plafonds et les sols atténuent le signal et les objets métalliques provoquent des réflexions. Il peut arriver qu'une station puisse recevoir le signal d'un point d'accès, mais que le point d'accès ne puisse pas recevoir le signal de la station. En outre, un accès extérieur au

réseau est toujours possible, ou la transmission du signal radio peut être perturbée. Par conséquent, la transmission de données sans fil est, par nature, moins fiable que via un réseau filaire. C'est pourquoi il peut arriver qu'il ne soit pas possible de se passer d'un réseau Ethernet filaire, particulièrement dans un environnement industriel.

Réseau Ethernet filaire

Les réseaux filaires sont similaires aux réseaux sans fil en ce sens qu'ils fonctionnent par l'échange de trames Ethernet entre les points d'extrémité. Pour éviter des problèmes lors de la mise en place d'un réseau, il convient de respecter un grand nombre de règles. Les violations de règles sont la cause la plus fréquente des problèmes de réseau. Par exemple, l'Ethernet ne permet pas d'utiliser n'importe quelle longueur de câble. Lors de la mise en cascade, c'est-à-dire la connexion en série

de hubs, il ne faut pas en utiliser n'importe quelle quantité et une structure de réseau mal choisie peut également entraîner des erreurs dans le réseau ou le surcharger inutilement. Mais en fonction de la qualité des câbles et des performances du matériel, il n'est souvent pas possible d'atteindre les débits de données souhaités.

Actuellement, le 100Base-TX (l'Ethernet rapide, 100 Mbit/s), le Gigabit-Ethernet (1 Gbit/s), le 10-Gigabit-Ethernet (10 Gbit/s) et le 100-Gigabit-Ethernet (100 Gbit/s) existent. Dans la plupart des applications, le Gigabit Ethernet fonctionne bien avec un câble Ethernet ordinaire, particulièrement avec les câbles de catégories CAT5e et CAT6. Ces types de câbles suivent la norme de câblage 1000BASE-T, également appelée IEEE 802.3ab.

L'interface Ethernet 1 Go fonctionne selon la norme 802.3ab-1999 (CL40) et nécessite quatre paires de fils par canaux pour la transmission des signaux. On obtient ainsi un débit de symboles de 125 mégabaud (MBd) avec une bande passante de 62,5 MHz par canal (2 bits par symbole). La tension de signal pour le 1000BASE-T (GB-Ethernet) est en moyenne de 750 mV en différentiel, avec pour limites : $820 \text{ mV} > U_{\text{Signal}} > 670 \text{ mV}$ pour une charge de 100Ω .

Ethernet 1-Gbit/s - Front - Ends

Une « extrémité frontale » dite Front End typique pour Ethernet est équipée d'une interface RJ45. Elle est conçue pour des transmissions en duplex intégral (Full Duplex), c'est-à-dire pour une transmission simultanée des données d'émission et de réception. Cela est possible parce que le connecteur comprend deux paires de fils, une paire étant toujours nécessaire par direction (principe de tension différentielle). Pour chaque connexion RJ45, la norme IEEE exige une isolation galvanique par transformateur. Ce transformateur protège les appareils des dommages causés par la haute tension sur la ligne et empêche les décalages de tension qui peuvent se produire en raison des différences de potentiel entre les appareils. La **figure 1** montre le circuit de principe d'une interface Gigabit Ethernet.

Câblage discret de l'interface Gigabit Ethernet

Le transformateur Ethernet (Transformateur LAN) est l'interface entre l'appareil et le câble Ethernet. Le transformateur assure la séparation galvanique entre l'appareil et le câble, essentielle pour la sécurité, et en même temps l'adaptation de l'impé-

dance entre, d'une part la logique interne et d'autre part les paires de fils symétriques. En outre, le transformateur protège l'appareil contre les perturbations transitoires, supprime les signaux de mode commun entre l'IC (Phy) et le câble, aussi bien de l'appareil vers l'extérieur que du câble extérieur vers l'électronique dans l'appareil. Cependant, le composant doit également transmettre les données à un débit allant jusqu'à 1 Gbit/s sans atténuer sensiblement les signaux d'émission et de réception. Des composants supplémentaires sont nécessaires pour répondre à l'adaptation et aux exigences CEM.

Le circuit de la **figure 2** montre un câblage de l'interface Gigabit Ethernet avec des composants discrets. Le transmetteur LAN assure une isolation DC entre l'électronique et le câble réseau. Le point milieu de l'enroulement côté primaire présente la terminaison dite de « Bob Smith ». Pour chaque paire de fils, une résistance de 75Ω est connectée pour former un « point étoile ». Le tout est ensuite isolé galvaniquement et connecté à la masse du boîtier au moyen de deux condensateurs de $100 \text{ pF}/2\text{kV}$ montés en parallèle. Les self de filtrage en mode commun intégrées en plus dans le module X3 réduisent les perturba-

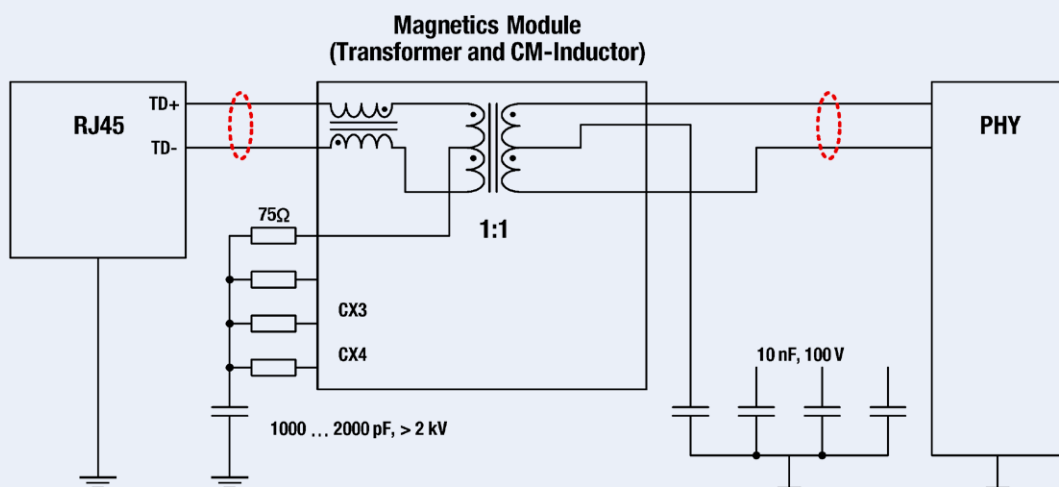


Figure 1. Circuit de base d'une interface Gigabit Ethernet. Représentation d'une transmission d'un total de quatre canaux.
(Source : Würth Elektronik)

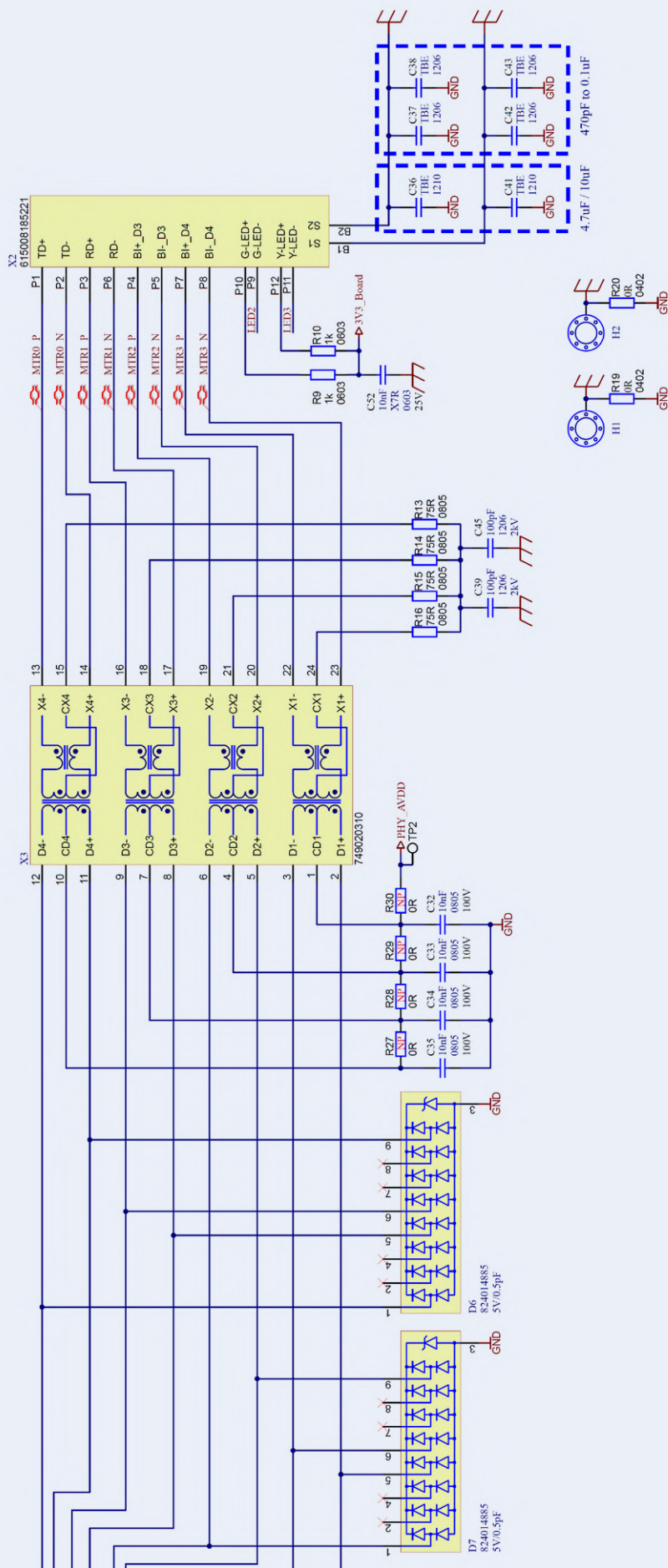


Figure 2. Circuit discret d'une interface Gigabit Ethernet. Le module X3 comprend les transformateurs LAN ainsi que des selfs de mode commun.

tions qui sont couplées de manière capacitive et inductive via les longs câbles Ethernet et qui pourraient ainsi déséquilibrer la communication différentielle de données Ethernet et la transformer en perturbations de mode commun.

R9, R10 et C52 sur la figure 2 servent à alimenter les LED typiquement intégrées dans le connecteur. Les condensateurs C36 à C38 et C41 à C43 permettent de relier le blindage de la prise Ethernet à la masse de la carte (GND). Pour les boîtiers en tôle, il est préférable de ne pas équiper ces condensateurs et de raccorder directement la masse (GND) de l'électronique au boîtier via des raccords à vis. Pour les boîtiers en plastique, il convient d'équiper les condensateurs afin de raccorder le blindage du câble Ethernet à la masse de référence. Les résistances 0 Ω R19 et R20 ont le même objectif. Toutefois, il n'existe pas ici de séparation galvanique, comme c'est le cas avec les condensateurs. Les équipements alternatifs ont été prévus ici à des fins « expérimentales » afin de comparer la qualité de blindage de différents câbles Ethernet. Sur le côté secondaire des transformateurs, les condensateurs C32 à C35 relient les points milieu de ces derniers à la masse (GND) d'un point de vu HF. Pour éviter les courants de compensation DC du PHY, une séparation galvanique par condensateurs est nécessaire. Les résistances R27 à R30 sont prévues en raison des exigences de certains fabricants de PHY (Option Driving de ligne en mode courant), mais ne sont généralement pas nécessaires si le PHY fonctionne en « mode de tension standard ». En revanche, les diodes TVS D6 et D7 sont indispensables pour limiter les perturbations transitoires (Surtensions, ESD) apparaissant côté interface vers le PHY par rapport à la masse du circuit (GND). Côté secondaire, c'est-à-dire après les transformateurs du module X3, les transitoires se produisent en mode commun, raison pour laquelle une diode TVS doit être connectée à la masse de référence à chaque connexion des transformateurs. Les niveaux de bruit sont toutefois plus faibles du côté secondaire du transformateur que du côté primaire. Pour que les diodes TVS fonctionnent proprement, il est important de les connecter avec une faible impédance, d'une part dans les lignes de signal et d'autre part à la masse.

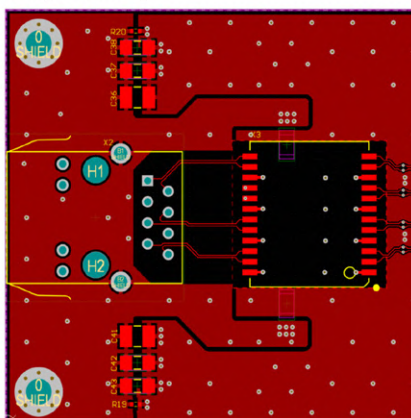
Le « Layout » des quatre couches de la carte

de la zone d'interface Ethernet est illustré à la **figure 3**. L'empreinte de masse du boîtier est séparée du GND de l'électronique dans les quatre couches. Les surfaces de la masse du boîtier ne se chevauchent donc pas avec d'autres couches afin de réduire au maximum le couplage capacitif. Les plans de masse ont été l'objet de connexions transversales tous les 4 mm environ. Les lignes de signaux provenant du connecteur Ethernet sont symétriques, routées avec une impédance différentielle de 100 Ω par rapport à la masse de référence. Les paires de conducteurs présentent une largeur de piste de 0,154 mm et sont espacées de 0,125 mm. La prise Ethernet est positionnée sur le bord du circuit imprimé afin de garantir une connexion à basse impédance avec un boîtier métallique le cas échéant.. Le Transfo LAN (X3) est placé à proximité immédiate afin de limiter les influences de couplage électrique ou les perturbations dues aux longues pistes conductrices. Comme côté primaire, une impédance différentielle de 100 Ω par rapport à la masse de référence doit être respectée sur le côté secondaire du module de transmission au niveau des pistes conductrices. Les diodes TVS doivent être connectées directement au signal et au GND afin d'éviter une chute de tension due à des inductances parasites.

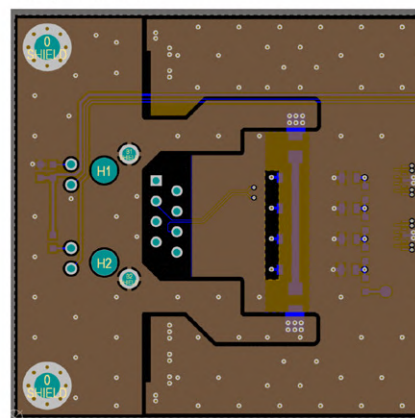
Conformité CEM

Du point de vue de la CEM (Compatibilité Electromagnétique), la carte respecte les tests en immunité selon la norme industrielle (EN61000-6-2) et les tests en Emissions selon la EN55032 classe B pour les appareils multimédia. Pour réussir la conception d'une interface Ethernet 1 Gbit/s, il faut tenir compte de nombreux points, ceux-ci étant une conception de circuit et routage adaptée à la RF, un concept de masse dépendant du système et le bon choix des composants. Ce n'est que lorsque tous ces points sont pris en compte ensemble qu'il est possible de développer un produit qui fonctionne de manière fiable et qui répond aux exigences élevées.

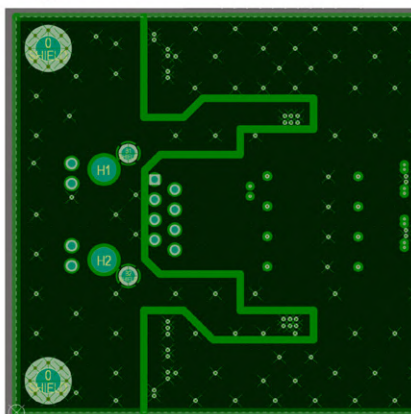
Layout Top



Layout V_{CC}



Layout GND



Layout Bottom

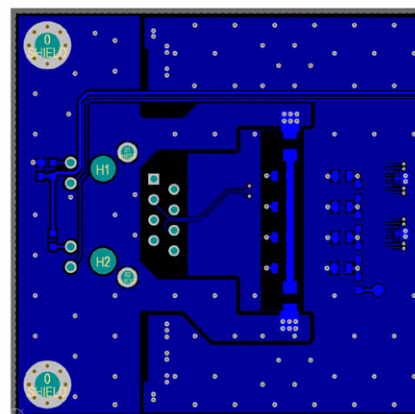


Figure 3. Routage des quatre couches de la carte dans la zone d'interface Ethernet.

De plus amples informations sur ces aspects et sur d'autres normes d'interface sont disponibles dans diverses notes d'applications de Würth Elektronik [1]. ◀

220182-04

L'auteur



Heinz Zenkner, Docteur Ingénieur, est collaborateur indépendant chez Würth Elektronik dans les domaines du marketing technique et de l'ingénierie d'application et donne des cours dans le domaine de la CEM à l'académie technique du groupe. Parallèlement, le Dr Zenkner est expert CEM assermenté. M. Zenkner est de longue date auteur de plusieurs revues et livres spécialisés. En outre, il a travaillé en tant que chargé de cours dans différentes universités, à la Chambre de commerce et d'industrie et lors de nombreux séminaires.

LIEN

[1] Notes sur les applications de Würth Elektronik: www.we-online.com/applicationguide/en

la détection d'objets en temps réel pour microcontrôleurs grâce à Edge Impulse FOMO

Jan Jongboom, Edge Impulse

Nous, les humains, dépendons largement de la vue pour accomplir de nombreuses tâches quotidiennes, des plus simples aux plus complexes. D'un seul coup d'œil, nous savons s'il y a des gens autour de nous, s'il y a un éléphant à proximité ou combien de places de parking sont disponibles. En dépit de l'importance de la vision, de nombreux appareils embarqués n'ont pas la capacité de la perception visuelle. Ne serait-il pas merveilleux de pouvoir apprendre à tous nos appareils à voir le monde comme nous le voyons ?

Ces dernières années, la vision par ordinateur a connu des développements surprenants, qui ont permis des avancées dans des domaines tels que les voitures autonomes et les systèmes d'authentification biométrique (très utiles si, comme moi, vous voyagez beaucoup !). Mais ces cas d'utilisation sont incroyablement gourmands en calcul, car ils nécessitent des processeurs graphiques coûteux ou des accélérateurs spéciaux pour fonctionner.

La bonne nouvelle est que les tâches de vision par ordinateur ne nécessitent pas toujours un calcul aussi puissant. Toute question fermée (« Est-ce que je vois un éléphant ? », « Est-ce que cette étiquette est bien collée sur la bouteille ? ») peut apporter une valeur considérable aux dispositifs embarqués limités. De plus, ces problèmes de classification d'images peuvent même

être résolus par les microcontrôleurs modernes.

Et si nous pouvons ajouter des capacités de vision encore plus avancées à chaque dispositif embarqué ?

Classification result

Summary	
Name	Parking_data_244.png.2tgq40ai
CATEGORY	COUNT
F1 score	91.43%
car	16

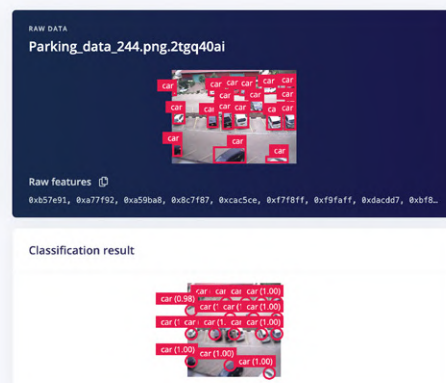
Figure 1. Classification FOMO dans Edge Impulse Studio.

Bonjour FOMO

Nous concrétisons cette idée. Nous avons développé une nouvelle architecture de réseau neuronal pour la détection d'objets, appelée Faster Objects, More Objects, ou FOMO (figure 1). Elle a été conçue dès le départ pour fonctionner en temps réel sur des microcontrôleurs, de sorte que les ingénieurs embarqués peuvent éviter la peur de manquer le coche en matière de vision par ordinateur.

Rapide, léger et flexible

FOMO est capable de fonctionner sur un microcontrôleur à 32 bits, comme un Arm Cortex-M7, avec une fréquence de 30 images par seconde. Et si vous optez pour un Raspberry Pi 4, vous serez en mesure de détecter des objets à une fréquence d'environ 60 images par seconde. C'est à peu près 30 fois plus rapide que



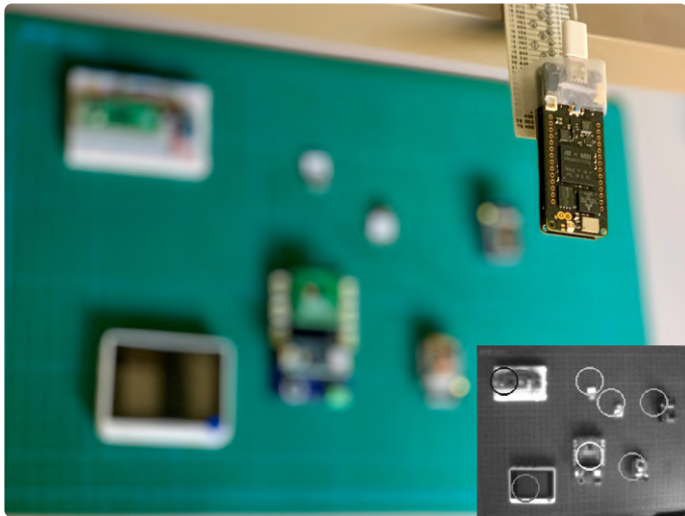


Figure 2. Exécution de la détection d'objets sur une grande variété de cartes de développement, y compris l'Arduino Portenta.



Figure 3. Voici une ancienne itération de l'approche FOMO utilisée pour compter les abeilles individuelles.

MobileNet SSD ou YOLOv5.

La capacité de FOMO est réduite à environ 100 Ko de RAM, ce qui permet d'exécuter les applications de détection d'objets en temps réel sur des cœurs Arm Cortex-M4 très contraignants, sur des cœurs plus puissants, comme les noyaux Cortex-M7 de l'Arduino Portenta H7 (figure 2), sur le nouvel Arduino Nicla Vision (un autre processeur double Arm Cortex-M7/M4), ou même sur des DSP spécialisés comme le WE-I de Himax.

FOMO peut s'adapter aux plus petits microcontrôleurs tout comme aux passerelles ou aux processeurs graphiques. Ce haut degré de flexibilité le rend également utile lorsque la détection de défauts nécessite d'identifier des variations les plus petites dans une image.

Dans un microcontrôleur dont la capacité de calcul et de mémoire est strictement limitée, il est préférable d'utiliser une taille d'image d'environ 96x96 pixels. Mais avec un microcontrôleur plus puissant, une image de 160x160 pixels est probablement adéquate. Le plus important est que FOMO est entièrement convolutif, il fonctionne donc avec n'importe quelle taille d'entrée arbitraire. Si vous souhaitez avoir une plus grande granularité, plus de détails ou plus d'objets, vous pouvez simplement augmenter la résolution d'entrée.

Il perçoit les petits détails

Cette nouvelle architecture peut même repérer et compter de nombreux objets

très petits de manière très efficace tant que les éléments présents dans le cadre sont de taille similaire et ne se chevauchent pas (figure 3). C'est quelque chose que le MobileNet SSD et le YOLOv5 ne peuvent pas réaliser très bien, malgré le fait qu'ils soient des modèles plus grands et plus performants.

Fini les ratés

FOMO est disponible dès aujourd'hui, fonctionne sur une grande variété de plateformes informatiques et est compatible avec les systèmes Linux, les microcontrôleurs Cortex-M et les DSP spécialisés. Il suffit d'ajouter une caméra et d'utiliser Edge Impulse, et le tour est joué. Avec FOMO, vous pouvez rapidement

intégrer la détection d'objets à n'importe quel appareil doté d'une caméra. Ne ratez plus ce que les ingénieurs en systèmes embarqués ont dû affronter jusqu'à présent en matière de vision par ordinateur (figure 4).

Pour en savoir plus sur FOMO et expérimenter votre propre algorithme, rendez-vous sur edgeimpulse.com/fomo. ◀

220207-04



Jan Jongboom est un ingénieur de systèmes embarqués et un défenseur de l'apprentissage automatique,

toujours en quête de moyens de collecter davantage d'informations dans le monde réel. Il a livré des appareils, travaillé sur les dernières technologies de réseau, simulé des microcontrôleurs. Il y a même un monument à San Francisco qui porte son nom. Il est actuellement cofondateur et directeur technique de Edge Impulse, la principale plateforme de développement pour l'apprentissage automatique embarqué, qui compte plus de 80 000 projets.

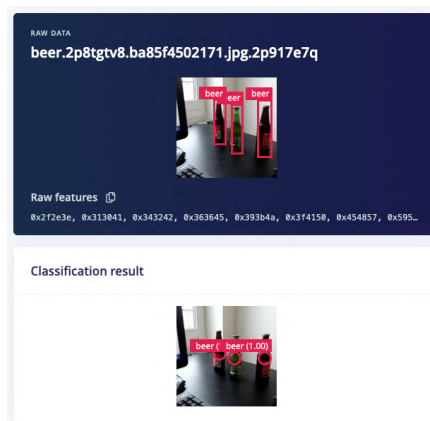


Figure 4. Formation sur les centroïdes de bouteilles de bière. En haut les étiquettes sources, en bas le résultat de l'inférence.

tubes à ondes progressives

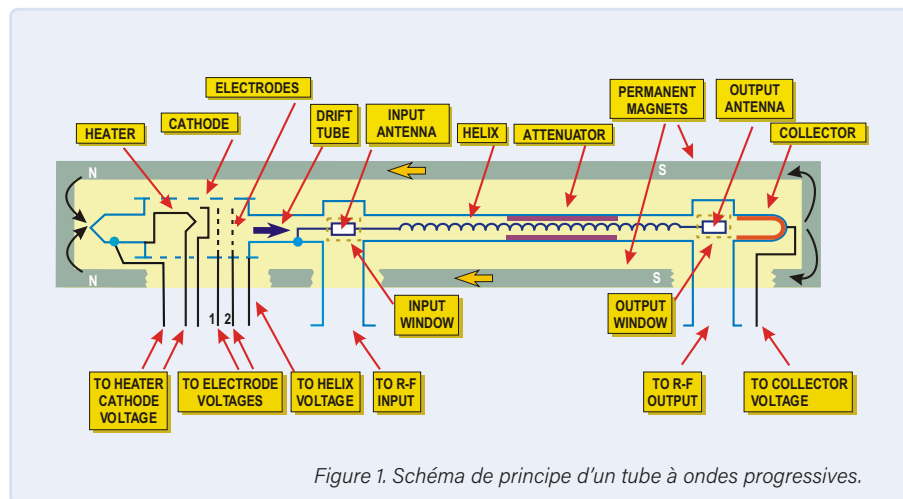


Figure 1. Schéma de principe d'un tube à ondes progressives.

Neil Gruending (Canada)

Le monde des amplificateurs RF est fascinant par la multiplicité de leurs conceptions. Si aujourd'hui, les semi-conducteurs se sont imposés à peu près partout, il subsiste encore quelques cas où les tubes à vide restent incontournables. Après avoir examiné les klystrons dans le passé, tournons-nous aujourd'hui vers l'amplificateur à tube à ondes progressives, un autre de ces héros électroniques méconnus.

L'un des aspects les plus fascinants des amplificateurs à tube à ondes progressives est leur mode de fonctionnement. Ces tubes comportent un élément chauffant, une cathode émettrice d'électrons et des électrodes accélératrices formant un canon à électrons, similaire à celui d'un tube cathodique, qui envoie un faisceau d'électrons vers le collecteur (fig. 1). Ce faisceau est focalisé par un champ magnétique externe, généralement constitué d'aimants permanents. En utilisant la modulation de vitesse des électrons du faisceau par le signal d'entrée RF, ceux-ci amplifient ce signal en lui cédant une partie de leur énergie.

Les électrons du faisceau étant beaucoup moins rapides que le signal RF (dont la vitesse est voisine de celle de la lumière), celui-ci est ralenti en allongeant son trajet par passage dans un fil enroulé en spirale, l'hélice, de sorte que la composante axiale de sa vitesse soit légèrement inférieure à celle du flux d'électrons.

En se propageant le long de l'hélice, le signal RF module la vitesse des électrons du faisceau, ceux en phase accélérant, les autres ralentissant, ce qui a pour effet de les regrouper en paquets. Les électrons ralentis, en majorité, induisent un signal amplifié dans l'hélice qui est ensuite prélevé à l'extrémité de l'hélice à l'aide d'un coupleur directionnel. Comparés aux klystrons [1], les tubes à ondes progressives se distinguent par une bande passante plus large. Dépourvus de composants résonnants, ils sont idéaux pour les applications à micro-ondes de faible puissance comme les radars ou les communications spatiales. Un excellent exemple est l'amplificateur Collins Radio en bande S (figures 2 et 3) utilisé dans le programme spatial Apollo [2]. Il s'agissait d'un amplificateur compact de 20 W, pesant 15 kg, qui transmettait l'ensemble des signaux voix, données et télévision au réseau d'antennes paraboliques terrestres de 26 m de la NASA. En comparaison, la station terrestre utilisait un émetteur directionnel de 10 kW pour communiquer avec le vaisseau.

Bien que l'usage des tubes à ondes progressives soit essentiellement du domaine commercial, il y a toujours un petit groupe de passionnés expérimentant ces remar-

quables petits amplificateurs dans des émetteurs à micro-ondes amateurs [3]. Leur plus grand défi, cependant, est d'en trouver, des tubes !

210418-04 – VF : Helmut Müller

Des questions, des commentaires ?

Contactez Elektor (redaction@elektor.fr).



Figure 2. Amplificateur en bande S Collins Radio à tube à ondes progressives utilisé pour la communication avec la Terre pendant la mission Apollo. (Source : Ken Shirriff)

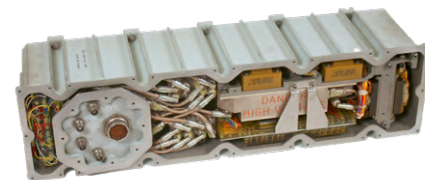


Figure 3. Fonctionnant à plusieurs milliers de volts, l'amplificateur de Collins Radio était un enchevêtrement compact de câbles coaxiaux. (Source : Ken Shirriff)

LIENS

[1] N. Gruending, « le klystron, drôle de composant n°13 », Elektor 03/2015 : <https://www.elektormagazine.fr/140535>

[2] K. Shirriff, « Inside a 20-Watt Traveling-Wave Tube Amplifier from Apollo », blog de Ken Shirriff, 07/2021 : <https://bit.ly/3ea8lOn>.

[3] H. Griffiths, « Travelling Wave Tube Amplifiers », The National Valve Museum, 09/1980 : <https://bit.ly/3wA8aCn>

Narrowband Internet of Things

Normes, couvertures, conventions et modules



Tam Hanna (Slovaquie)

Le NB-IoT vous intéresse ?
Est-il fait pour vous ?
Jetons un coup d'œil.

source: shutterstock.com

En plus de LoRa et Sigfox, les réseaux de communication mobile constituent également une bonne option pour la transmission des données des capteurs IdO. Le passage de l'EDGE à l'UMTS a rendu cette option encore plus attrayante, puisque l'utilisation d'un système de transmission plus rapide peut se révéler préférable à l'utilisation d'un système moins énergivore, mais plus lent. Cependant, cette règle générale n'est pas tout à fait valable avec la large bande passante et la grande consommation de la 4G/LTE. La consommation d'énergie des émetteurs est nettement plus élevée, et en plus, les modules sont plus chers. Néanmoins, cela peut être avantageux.

Dans le cadre de la spécification 3GPP Release 13, désignée par LTE comme « informative », la GSM Association définit deux systèmes pour l'IdO. Le premier est *Narrowband Internet of Things* (NB-IoT), et le second est le LTE-M, également connu sous le nom de LTE Cat-M1 ou eMTC.

Le LTE-M est en fait une extension « légère » du LTE (4G) avec une bande passante de 1,4 MHz, tandis que le NB-IoT est une norme de communication sans fil dédiée à l'Internet des Objets. La principale différence est que le LTE-M prend en charge la transmission de la voix avec VoLTE, tandis que le système NB-IoT transmet uniquement des messages de données.

Les voies de NB-IoT, chacune d'une largeur de 180 kHz, utilisent un sous-ensemble des méthodes implémentées dans la version complète de LTE. La liaison montante a recours à une version simple de la méthode d'accès multiple par répartition en fréquence FDMA (*frequency division multiple access*), tandis que la liaison descendante applique la méthode OFDMA (*orthogonal frequency-division multiple access*). La méthode de modulation QPSK (*quadrature phase shift keying*) ne nécessite pas de matériel particulièrement complexe en termes de puissance de traitement.

Tableau 1. Bandes de fréquences

Région	Bandes
Europe	3, 8, 20
(anciennement) pays de la CEI	3, 8, 20
Amérique du Nord	2, 4, 5, 12, 66, 71, 26
Asie-Pacifique (APAC)	1, 3, 5, 8, 18, 20, 26, 28
Afrique sub-saharienne	3, 8
Moyen-Orient et certaines parties de l'Amérique du Nord	8,20
Amérique Latine	2, 3, 5, 29

Toutefois, il faut noter que le déploiement du NB-IoT entraîne généralement des frais supplémentaires pour l'opérateur, liés au nouveau matériel. Grâce à sa bande passante extrêmement étroite, NB-IoT peut facilement être placé dans la bande de garde entourant les paquets de la fréquence LTE. D'autre part, il est également possible d'utiliser NB-IoT en mode autonome.

Performance

Même la norme de communication sans fil la plus attrayante sur le plan technique n'est d'aucune utilité si la capacité de transmission est insuffisante pour la tâche prévue. Dans le cas de NB-IoT, la version est une considération importante, car il existe des différences entre LTE Cat NB1 (version 13) et LTE Cat NB2 (version 14). La version précédente ne peut atteindre que 26 kbit/s en voie montante, mais Cat NB2 est considérablement plus rapide avec 127 kbit/s en voie montante et 159 kbit/s en voie descendante. À titre de comparaison, la 3G classique (non HSDPA) atteignait initialement 380 kbit/s. LTE Cat M1 fonctionne actuellement à environ 1 Mbit/s en montée et en descente. La version 14 augmente ce débit à 4 Mbit/s en montée et 7 Mbit/s en descente.

Les différences entre les temps de latence sont énormes. LTE-M peut généralement atteindre 15 ms, alors qu'avec NB-IoT, la « plage de fonctionnement » recommandée est comprise entre 1,6 s et 10 s. Le fabricant de modules Sierra Wireless, surtout populaire aux États-Unis, décrit la situation comme suit :

« Un autre fait important à prendre en compte est que LTE-M peut prendre en charge tous les cas d'utilisation de NB-IoT. En d'autres termes, LTE-M prend en charge toute application LPWA, alors que NB-IoT est conçu pour des applications plus simples à capteur statique. » [1]

En outre, seule la version 2 de la norme NB-IoT prend en charge la transmission de données de localisation par l'opérateur réseau. Si le module n'a pas de fonction GPS ou si vous voulez éviter de recourir à une antenne externe, vous pouvez utiliser cette approche pour obtenir des données de position de base. La version 14 accélère également la recherche de nouvelles cellules, ce qui est principalement utile pour les appareils mobiles. Malgré ces nouveaux avantages du Cat NB2, le LTE-M reste le meilleur choix pour les applications automobiles et mobiles, car il permet un transfert cellulaire plus intelligent. La dernière amélioration est relative à la puissance d'émission : les émetteurs à très faible puissance [2],

qui peuvent fonctionner avec seulement 14 dBm, ne sont autorisés que dans la version 14.

Si jamais vous mettez la main sur un module 4G de Verizon, vous vous demanderez naturellement quelles bandes il utilise. La bande 13, qui n'est importante qu'en Amérique du Nord, a posé des problèmes à de nombreux fournisseurs de modules asiatiques et européens. Le **tableau 1** est extrait du *Deployment Guide* [3] de la GSM Association. Vous devez vous assurer que le module que vous choisissez prend en charge toutes les bandes utilisées par votre opérateur préféré.

Disponibilité et conventions

Il va sans dire que les normes de communication sans fil ne sont utiles que si elles sont disponibles dans la pratique. Dans le cas des deux normes de communication sans fil de l'IdO, il est recommandé de consulter la carte mondiale interactive de la GSM Association dans la figure 1 [3] (situation en septembre 2021). Comme vous pouvez le constater, le Mexique est l'unique pays où seul le CAT-M est disponible (probablement pour sa portée plus grande), tandis que le « NB-IoT only » est plus répandu dans les zones rurales des pays asiatiques et notamment en Europe de l'Est. Dans les régions fortement industrialisées d'Europe, d'Amérique du Nord, d'Asie, d'Australie et d'Océanie, les deux versions sont disponibles.

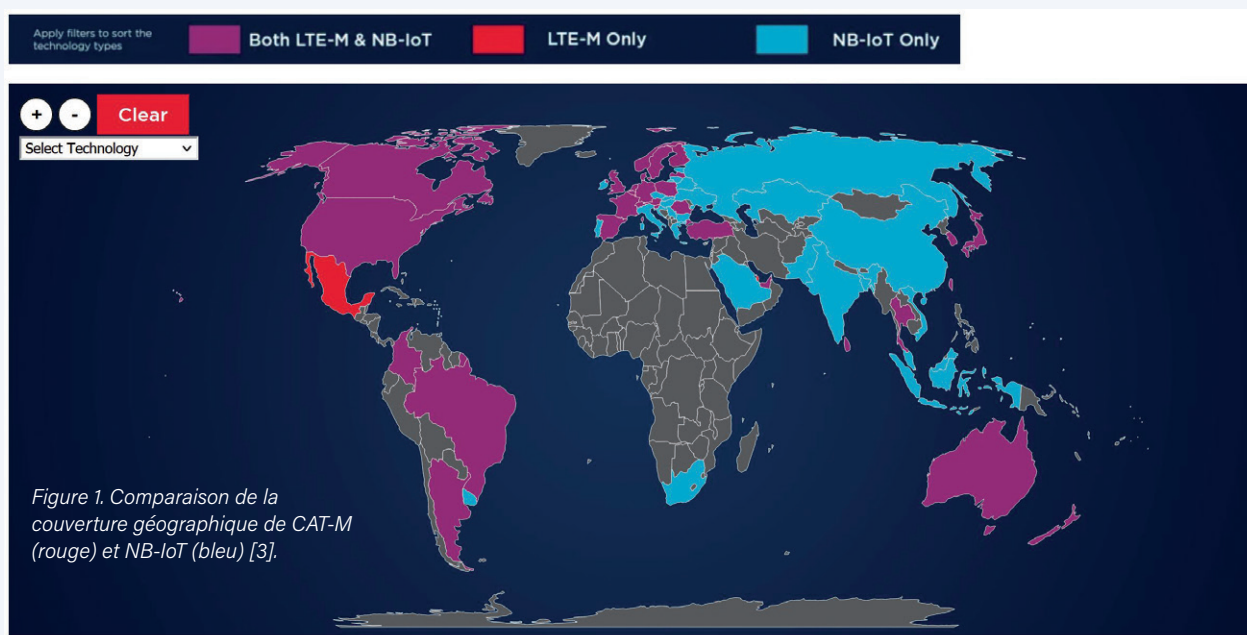
Les conventions de CAT-M sont généralement des accords ordinaires dans lesquels le volume total d'utilisation et le nombre de cartes SIM déterminent le coût global. Par souci d'exhaustivité, notez qu'en ce qui concerne le coût, il vaut mieux acheter une carte SIM prépayée chez un fournisseur IdO tel que PodGroup que sur le marché libre.

D'après l'auteur, un consultant ayant une expérience pratique, la non-soumission de NB-IoT à des restrictions de cycle d'utilisation n'est pas confirmée. La connectivité IoT restera toujours une question de négociation avec votre opérateur mobile. Souvent, il imposera un nombre de paquets limité par intervalle de temps donné. Les opérateurs publient rarement leurs conditions exactes à cet égard, ce qui rend la déclaration suivante de T-Mobile USA encore plus remarquable :

« Rejoignez le premier réseau NB-IoT national pour optimiser le suivi de biens, de villes connectées et bien plus encore. Offre d'une durée limitée, sujette à modification. Des taxes et des frais peuvent être appliqués. Le plan comprend 10 transactions par paquet par heure jusqu'à 64 Kbps, jusqu'à 12 Mo. Le paiement intégral du service est dû à l'activation. » [4]

Il est intéressant de noter qu'il ne s'agit que d'une opinion individuelle et que Hutchison Holding Ltd a confirmé que le volume total du trafic de données (dans les limites du volume convenu) peut être épuisé en une journée. Tom Tesch, le porte-parole autrichien de Hutchinson, déclare à ce propos :

« Le débit de données de NB-IoT — conformément à la norme — est très faible et convient principalement à la transmission de mesures individuelles ou d'indicateurs d'état. Pour cette raison, il est rare que les appareils NB-IoT requièrent plus de 5 à 10 Mo par mois. Pour les applications exigeant une bande passante large, telle que la transmission de photos ou de vidéos, la 3G/4G et bien sûr la 5G sont des technologies plus adaptées. Il n'y a actuellement aucune limite quant



au volume ou à la durée l'utilisation, ce qui signifie que la totalité du volume peut même être consommée en une journée. »

Comment commencer ?

Après ces considérations de base, il est temps de réfléchir comment intégrer NB-IoT dans vos systèmes pratiques. Bien sûr, le développement de modems personnalisés n'est pas réalisable pour la plupart des entreprises. Mais nous avons déjà décrit en détail le processus de « conception » des modules sans fil, par exemple dans l'article [5]. Si vous ne souhaitez pas commencer par concevoir votre propre carte directement, vous pouvez utiliser une carte d'évaluation « turnkey », quoique les circuits intégrés Qualcomm ne soient pas toujours disponibles.

Deux solutions sont disponibles : la *NBIOT-BG96-SHIELD* d'Avnet, qui intègre un module Quectel BG96, et la *5 G NB IoT click board* de MikroElektronika, qui est dotée d'un module Cinterion. Arduino propose également une petite carte de développement : la *MKR NB 1500*. Cependant, ces deux cartes coûtent plus de 50 dollars. Dans de nombreux cas, il n'est plus permis de fournir des cartes d'évaluation équipées de cartes SIM. Alors le déploiement massif d'applications basées sur NB-IoT n'est pas si facile. Cela est dû au fait que les opérateurs de réseau n'ont pas encore mis la technologie à la disposition des utilisateurs finaux. Les opérateurs l'admettent d'ailleurs ouvertement, comme en témoigne la déclaration suivante de Hutchison : « NB-IoT est un réseau très moderne et innovant. Comme il n'y a pratiquement pas d'appareils disponibles sur le marché, le groupe cible se compose principalement de clients professionnels qui développent du matériel et du logiciel. Cela signifie que notre offre

est pour le moment exclusivement destinée aux clients professionnels, pour lesquels nous créons une offre sur mesure lors d'un processus de consultation. »

En travaillant avec des systèmes 2G/3G/4G « ordinaires », le recours à un fournisseur de communication mobile « virtuel » tel que PodGroup constitue une façon de contourner ce problème. En réponse à cette question, ils ont précisé que NB-IoT n'est pas encore vraiment adapté, en particulier pour les solutions « globales » qui doivent fonctionner avec une seule carte SIM.

Deux raisons : premièrement, le déploiement de NB-IoT est encore relativement limité. Deuxièmement, les accords d'itinérance entre les différents opérateurs de réseau ne sont pas encore adaptés à la nouvelle norme de communication sans fil NB-IoT. Tout comme pour les conventions fiscales entre pays, la réalisation de telles adaptations prend beaucoup de temps. En résumé : l'itinérance internationale du NB-IoT est encore à ses débuts.

Est-ce que ça vaut la peine ?

La recherche d'un module pratique prenant uniquement en charge NB-IoT est certainement une tâche très délicate. Quectel, par exemple, propose deux versions, même pour la plus petite série (BC660) : une avec uniquement NB-IoT, et l'autre avec eMTC et NB-IoT. Les deux normes sans fil sont également présentes dans des familles plus grandes, comme les fameux BG95 et BG96. Ces modules ne sont disponibles à des prix abordables que chez *SOS Electronic* : le BC660K-GL coûte 7,63 € en petites quantités, tandis que la version avec LTE-M et NB-IoT n'est pas répertoriée. Le prix du BG96 y est de 19 €.






LTE-M / NB-IoT	Cat 1	4G	3G	2G
 <p>ALEX-R5 series Ultra-small LTE-M / NB-IoT SIP with Secure Cloud</p>	 <p>UBX-R5 series Multi-band LTE-M / NB-IoT chipset</p>	 <p>SARA-R5 series LTE-M / NB-IoT modules with secure cloud</p>	 <p>SARA-N3 series Multi-band NB-IoT (LTE Cat NB2) modules</p>	
 <p>SARA-R4 series LTE-M / NB-IoT / EGPRS modules with Secure Cloud</p>				

Figure 2. Les résultats de la recherche d'un module purement CAT-M ne sont pas nombreux. [6].



Figure 3. Un moulin à vent traditionnel stiérien servant d'épouvantail. (Source : Martin Geisler, CC BY-SA 4.0 [7]).

Une recherche sur u-blox [6] donne plus de résultats. La famille SARA-N3 comprend un module exclusivement destiné à l'ensemble des protocoles NB-IoT, mais la société suisse ne propose pas de dispositif CAT-M exclusif (voir la **figure 2**).

Chez Gemalto, dont le site web est devenu encore plus confus qu'auparavant après son rachat par Thales, on trouve un module CAT-M pur de la forme de l'EMS31, ainsi qu'un module NB-IoT pur (ENS22) ayant le même facteur de forme. Chez le distributeur tchèque Sectron, vous pouvez comparer les prix : l'EMS31 coûte 14 €, l'ENS22 seulement 8 €.

Les informations sur la consommation de courant (sous une forme détournée) se trouvent dans les fiches techniques, désignées par « Hardware Interface Description ». La consommation de courant de l'EMS31 est maximale lorsqu'il fonctionne dans la bande 4. Elle est de 239 mA à une tension d'alimentation de 3,8 V. Pour l'ENS22, le courant maximal répertorié est de 404 mA dans la bande 28, mais il faut noter que les modules sans fil ne requièrent souvent des courants de pointe de ce type que pendant un temps très court.


LIENS

- [1] LTE-M vs. NB-IoT : quelles sont les différences ? : www.sierrawireless.com/iot-blog/lte-m-vs-nb-iot/
- [2] Article de Wikipedia sur NB-IoT : <https://fr.wikipedia.org/wiki/NB-IoT>
- [3] GSMA : carte mondiale des normes de communication sans fil de l'IdO : www.gsma.com/iot/deployment-map/
- [4] Page web de T-Mobile Narrowband IoT : <https://t-mo.co/3EC5Jo4>
- [5] Tom Hanna, « module cellulaire, même pas peur ! », Elektor 06-05/2021 : www.elektormagazine.fr/magazine/elektor-176/59583
- [6] Modules de communication mobile : www.u-blox.com/en/cellular-modules
- [7] Klapotetz : <https://bit.ly/3nOr0Fb>

Avantages

D'un point de vue technique, NB-IoT fonctionne parfaitement. Une fois arrivé à un accord avec un opérateur, il suffit de passer un coup de fil à votre avocat pour pouvoir exploiter le réseau, contrairement à la situation d'un WAN LoRa domestique. La consommation de courant de crête et de repos relativement faible des modules contribue également à réduire votre facture d'électricité.

En fin de compte, c'est une question d'échelle, tout comme les paradis fiscaux tels que Dubaï ou Monaco. Si vous utilisez cinq modems en un an, un module 4G « integral » ou, mieux encore, un module doté d'un adaptateur d'alimentation plus puissant et un peu plus cher, d'après l'expérience désagréable de l'auteur dans la pratique, vous aurez souvent besoin de recourir à « l'autre » norme sans fil. Ne serait-ce que parce que certaines stations de base ne prennent pas en charge toutes les normes de communication sans fil.

Évidemment, la situation change si vous achetez 50 000 modems qui seront tous destinés au même client. Si la mairie de Großdorf am Klapotetz (**figure 3**) a besoin de NB-IoT, l'opérateur local mettra probablement son réseau à niveau, et les économies réalisées grâce au grand nombre d'appareils y contribueront également. 

180021-04

Questions, commentaires ?

Contactez l'auteur (tamhan@tamoggemon.com) ou contactez Elektor (editor@elektor.com).

Contributeurs

Texte : Tam Hanna

Rédaction : Rolf Gerstendorf

Traduction : Asma Adhimi

Mise en page : Giel Dols



PRODUITS

► Livre en anglais, « IoT Home Hacks with ESP8266 », H. Henrik Skovgaard, (Elektor, 2020)
www.elektor.fr/19159

Publicité

Notre garantie : des produits 100 % authentiques

Mouser, premier distributeur agréé selon la norme SAE AS6496

La plus large sélection de
composants électroniques en stock™

mouser.fr/authentic



passerelle intérieure Dragino LPS8

Configuration rapide de la passerelle LoRaWAN



Figure 1. La passerelle intérieure Dragino LPS8. (Source : Dragino [5])

Mathias Claußen (Elektor)

Nous avons déjà abordé la manière dont vous pouvez interconnecter vos appareils électroniques à l'aide d'une liaison LoRaWAN. Si vous n'êtes pas à portée d'une passerelle LoRaWAN existante, ou si vous souhaitez simplement explorer le sujet plus en profondeur, vous pouvez configurer et exploiter votre propre passerelle. Nous avons expérimenté cette solution en utilisant la passerelle intérieure Dragino LPS8 à prix modéré.

Nous avons abordé le sujet LoRaWAN à plusieurs reprises dans Elektor. Il est relativement facile de créer un nœud LoRaWAN de base auquel est associé un capteur ou un actionneur. Dans ce type de configuration, un module LoRaWAN (qui gère la communication avec le réseau) est connecté à une carte à microcontrôleur, telle que le STM32 de STMicroelectronics ou le Raspberry Pi Pico [1, 2], qui communique avec le capteur. Pour que les données envoyées vers et depuis le nœud via LoRa soient transmises plus loin, il faut disposer d'une station distante. Dans ce cas, une passerelle LoRaWAN recevra les données par voie aérienne via LoRa et les acheminera sur Internet vers The Things Network (TTN) par exemple. Vous pouvez utiliser une passerelle déjà installée dans votre région (beaucoup sont gérées par des bénévoles), ou vous pouvez créer votre propre passerelle. J'utilise un Dragino LPS8 comme passerelle intérieure depuis plus d'un an maintenant.

Dragino LPS8

L'électronique intégrée est basée sur un petit SoC Wi-Fi Atheros (désormais Qualcomm) AR9331 cadencé à 400 MHz, spécialement conçu pour être utilisé dans les routeurs et points d'accès. Avec 64 Mo de RAM et 16 Mo de Flash, sa puissance de traitement n'est pas impressionnante en comparaison avec celle d'un Raspberry Pi Zero 2 W, mais elle est plus que suffisante pour les fonctions que la passerelle doit exécuter. Le SoC dispose également du Wi-Fi conformément à la norme 802,11 b/g/n et d'un port LAN 10/100 Mbit. Les vitesses de communication disponibles sont largement suffisantes pour le débit de données relativement lent utilisé par LoRaWAN. La passerelle non plus n'a pas besoin d'être très puissante, car elle s'occupe uniquement du module émetteur-récepteur LoRa intégré et transmet les données à l'internet. Un schéma fonctionnel est présenté à la **figure 2**.

LPS8 System Overview:

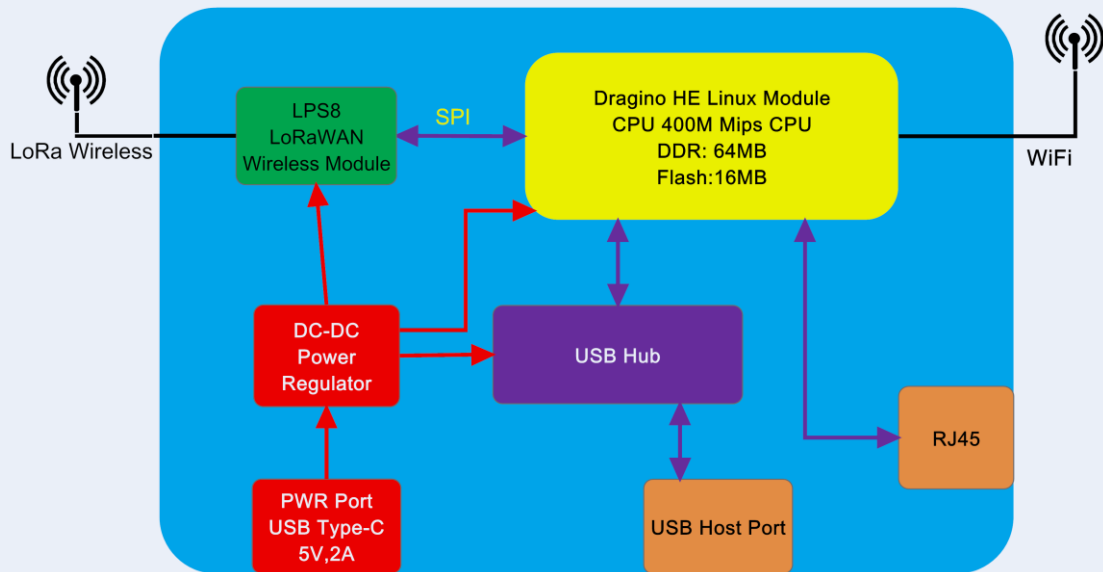


Figure 2. Schéma fonctionnel de la Dragino LPS8. (Source : Dragino [6])

L'émetteur-récepteur LoRa rassemble une puce bande de base LoRa SX1308 de Semtech (figure 3) et deux modules frontaux SX1257 (figure 4). Cette configuration assure la conversion de l'interface radio en Ethernet. La passerelle est alimentée par son port USB de type C et nécessite un adaptateur secteur de 5 V/2 A (10 W).

Comme le nom le suggère, la passerelle n'est pas étanche. Elle est destinée à être utilisée à l'intérieur d'un bâtiment. L'environnement doit donc être sec et relativement exempt de poussière. La structure du bâtiment et les murs intérieurs réduiront la couverture radio en comparaison avec un dispositif équivalent installé à l'extérieur dans un espace ouvert avec une antenne montée sur mât.

Manuel, micrologiciel et configuration du LPS8

La version la plus récente du manuel Dragino (disponible en ligne [3]) décrit la méthode de configuration de la passerelle. Le manuel a été continuellement mis à jour depuis la sortie du produit. Il présente les caractéristiques et les mises à jour du micrologiciel actuel. C'est tout à fait louable ; je souhaite seulement que d'autres fabricants de produits accordent autant d'attention aux détails dans leurs documentations.

Le micrologiciel est également bien maintenu. La version actuelle est datée du 4 novembre 2021 (à partir du 15 décembre 2021) [4]. Il est conseillé d'effectuer la mise à jour vers la version la

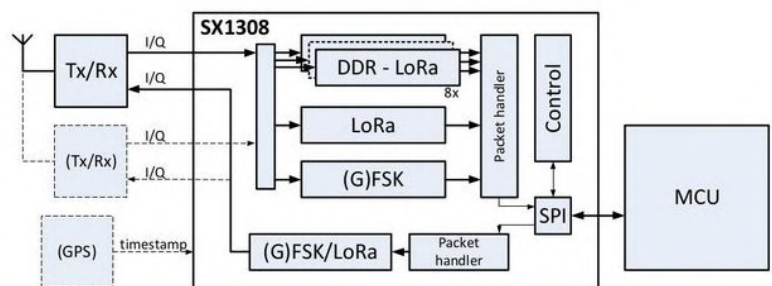


Figure 3. Schéma fonctionnel de la puce bande de base SX1308. (Source : Semtech [7])

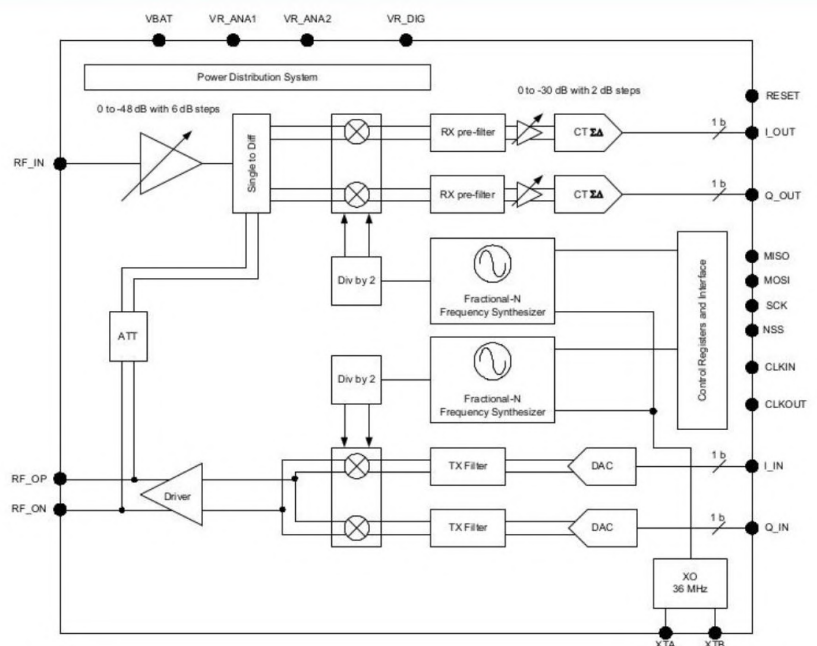


Figure 4. Schéma fonctionnel du module frontal SX1257. (Source : Semtech [8])

plus récente avant de mettre la passerelle en service. Ainsi, les bogues ou les faiblesses de sécurité éventuels seront corrigés autant que possible.

Le manuel vous guide dans l'installation. Il vous suffit de configurer le réseau de manière appropriée et d'effectuer les réglages pour la liaison LoRaWAN (par exemple, The Things Network). À ce stade, la passerelle LoRaWAN est prête à être utilisée (**figure 5**).

Sous-structure OpenWRT

Même si la première page de l'interface web ne le suggère pas, le micrologiciel OpenWRT (Open Wireless Router) basé sur Linux sert de base à la passerelle. Celui-ci prend non seulement en charge la fonction de passerelle LoRaWAN, mais fournit également d'autres paramètres pour le routeur (adresses IP, acheminement, Wi-Fi).

Grâce à la sous-structure OpenWRT, il est possible de connecter un modem LTE ou 5G au port USB de la passerelle si aucune autre liaison à Internet n'est disponible. Si vous le souhaitez, vous pouvez également accéder à la ligne de commande Linux par SSH. (Faites-le à vos risques et périls !) Pour ajouter plus de fonctions à l'appareil, vous pouvez installer des paquets supplémentaires via l'interface Web ou la ligne de commande.



Figure 5. Les voies de communication disponibles via la passerelle LPS8.

Solution fiable

Cela fait maintenant plus d'un an que j'utilise un Dragino LPS8. Pendant cette période, elle s'est avérée être une passerelle LoRaWAN fiable qui nécessite peu de maintenance. C'est tout ce que l'on demande d'un tel appareil. Elle continue à bien desservir mes différents nœuds LoRaWAN et offre une excellente couverture dans tout le bâtiment (et aux alentours). Si vous envisagez d'installer une passerelle LoRaWAN peu coûteuse dans un environnement domestique, vous devriez considérer de plus près la passerelle intérieure Dragino LPS8, qui est actuellement disponible dans l'e-choppe Elektor [5].

210680-04

Des questions, des commentaires ?

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

Contributeurs

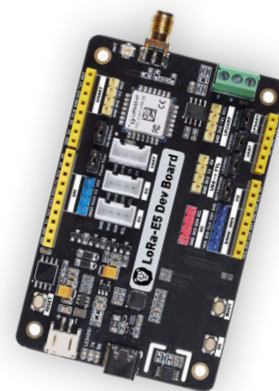
Texte : Mathias Claussen
Rédaction : Jens Nickel, C.J. Abate
Traduction : Asma Adhimi
Mise en page : Harmen Heida



PRODUITS

> **Dragino LPS8 Indoor LoRaWAN Gateway (868 MHz)**
www.elektor.fr/19094

> **Seeed Studio LoRa-E5 STM32WLE5JC Development Kit**
www.elektor.fr/19956



LIENS

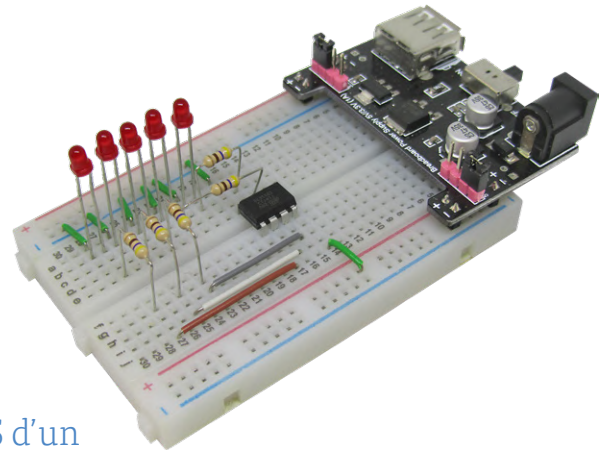
- [1] M. Claussen, « LoRaWAN : décollage facile », ElektorMag 3-4/2020: www.elektormagazine.fr/magazine/elektor-142/57188
- [2] M. Claussen, « LoRa avec le Raspberry Pi Pico » ElektorMag 7-8/2020: www.elektormagazine.fr/magazine/elektor-180/59777
- [3] Manuel de la passerelle intérieure Dragino LPS8: www.dragino.com/downloads/index.php?dir=LoRa_Gateway/LPS8/
- [4] Téléchargement du micrologiciel de la passerelle intérieure Dragino LPS8: <https://bit.ly/LPS8-firmware-release>
- [5] Ressource pour les images de la passerelle : www.dragino.com/media/k2/galleries/148/LPS8-10.jpg
- [6] Manuel de la passerelle intérieure Dragino LPS8: <https://bit.ly/LPS8-user-manual>
- [7] Fiche technique du module Semtech Front-end SX1257: <https://sforce.co/3fZmy1f>
- [8] Fiche technique de l'émetteur-récepteur Semtech SX1308: <https://sforce.co/32zxAqV>

Explore ATtiny Microcontrollers using C and Assembly Language

Extrait : ports d'E/S d'ATtiny

Warwick A. Smith (Afrique du Sud)

Les ports d'E/S contrôlent les broches d'un microcontrôleur et leur permettent d'être configurées un à un en entrée ou en sortie. C'est une notion si générale, valable dans n'importe quel cours de débutant ou d'introduction à la programmation des microcontrôleurs. Cependant, afin de comprendre et d'exploiter les capacités d'E/S d'un microcontrôleur, vous devez étudier plus profondément le circuit. Dans cet article, Warwick Smith, auteur d'un livre publié par Elektor, en fait une démonstration avec des exemples de programmation en assembleur du célèbre micro ATtiny. Intéressé ? Jetons un œil !



Note de l'éditeur. Cet article est un extrait du livre de 376 pages *Explore ATtiny Microcontrollers using C and Assembly Language* (W. Smith, Elektor, 2021) formaté et légèrement modifié pour correspondre aux normes éditoriales et à la mise en page du magazine Elektor. Puisque cet article est extrait d'une publication plus vaste, certains termes peuvent faire référence à des passages du livre d'origine situés ailleurs. L'auteur et l'éditeur ont fait de leur mieux pour l'éviter et seront heureux de répondre aux questions. Pour les contacter, voir l'encadré « Des questions, des commentaires ? ».

Les ports d'E/S sont configurés et contrôlés en utilisant un ensemble de quatre registres dans ATtiny13(A) et ATtiny25/45/85. Il est possible de lire le niveau logique d'une broche configurée en entrée à l'aide d'un programme exécuté sur l'AVR. Si un interrupteur y est connecté, le niveau logique déterminera s'il est ouvert ou fermé. Lorsqu'une broche est configurée en sortie, elle peut être utilisée pour commuter le niveau logique de la broche vers le haut (niveau logique 1) ou vers le bas (niveau logique 0). Une broche de sortie peut être utilisée pour piloter une LED, comme dans le projet de clignotement de LED décrit ailleurs dans ce livre.

Configuration des broches en sorties en assembleur

Dans cette section, nous configurons plus qu'une broche en sortie en utilisant un ATtiny 8 broches avec cinq LED connectées avec des résistances en série. Ensuite, nous créerons un code pour réaliser un compteur binaire de 5 bits qui compte de zéro avec ces LED.

La figure 1 montre un schéma de circuit d'un microcontrôleur AVR ATtiny13(A) ou ATtiny25/45/85 avec cinq LED connectées aux broches E/S PB0 à PB4. La broche PB5 du microcontrôleur ATtiny dans le circuit est utilisée comme broche debugWIRE pour

la programmation et le débogage. Notez que pour utiliser cette configuration, il faut utiliser un programmeur/débogueur, tel que Atmel-ICE, ou AVR Dragon.

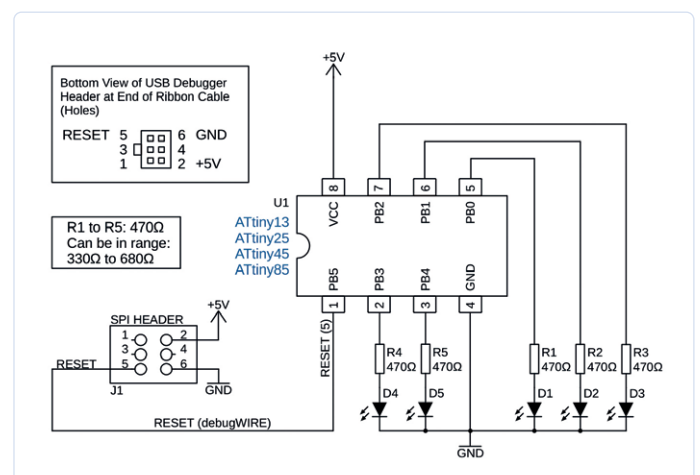


Figure 1. Schéma du circuit du compteur à cinq LED.

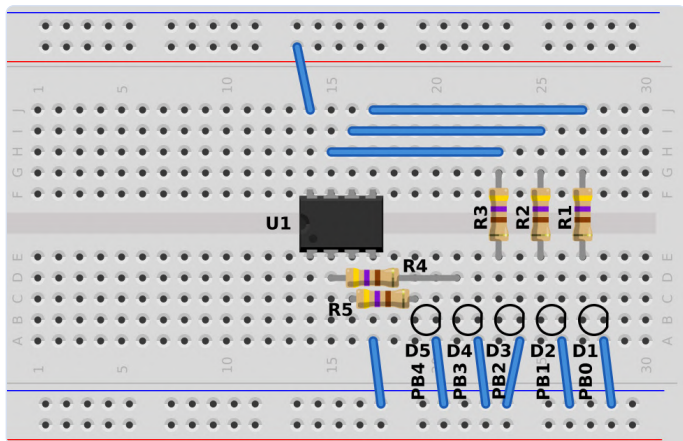


Figure 2. Disposition du circuit de compteur à cinq LED sur la platine à essai.

Un programmeur USB (program-only) ne fonctionnera pas en mode debugWIRE, et ne pourra pas effectuer de débogage.

Note : N'activez pas le fusible DWEN avec un programmeur USB (program-only), car l'AVR ne pourra pas quitter le mode debugWIRE. L'utilité du mode debugWIRE avec cet exemple de circuit est de libérer les autres broches de l'AVR qui sont normalement utilisées en mode ISP/SPI.

Programmateurs program-only

Les programmeurs USB avec uniquement des capacités de programmation peuvent être utilisés avec le circuit de la **figure 1** pour charger le programme d'exemple qui suit et voir la valeur du compteur affichée sur les LED.

Connectez les LED comme indiqué dans la **figure 1** et la **figure 2**. Le design original du USBasp et celui de USBtinyISP sont munis de résistances de protection sur les lignes qui sortent de ces programmeurs, pour les protéger, ainsi que la puce AVR cible. L'Arduino Uno programmé comme un ArduinoISP n'a pas de résistances de protection, mais il est possible d'en ajouter sur la ligne MOSI et la ligne SCK de l'Arduino Uno dédié à l'AVR cible. Des résistances de protection de 270 Ω sont utilisées dans la conception originale de l'USBasp, et sont placées sur les lignes MOSI, SCK et RESET. Des résistances de 1,5 k Ω sont utilisées sur les lignes MOSI et RESET de la conception originale de l'USBtinyISP.

Les résistances de protection empêchent un court-circuit si l'une des broches de l'AVR cible conduit en même temps une tension de sortie de polarité inverse que celle du programmeur.

Périphériques interférant avec la programmation

Bien que le circuit de la **figure 1** puisse être programmé à l'aide de l'interface ISP/SPI avec les LED et les résistances en série, d'autres circuits peuvent avoir du matériel qui interfère avec la programmation. Il existe différentes solutions à ce problème. Bien sûr, les lecteurs qui disposent d'un programmeur/débogueur USB doté de debugWIRE, tel que l'Atmel-ICE, peuvent simplement mettre l'AVR cible en mode debugWIRE, et ainsi utiliser une seule broche pour la programmation. Une autre solution est de programmer l'AVR sur une autre platine d'essai en utilisant ISP/SPI, et de le

brancher ensuite sur le circuit cible.

On peut aussi utiliser un AVR avec plus de broches, mais celles du port libre ne correspondent pas toujours à celles du même port d'un ATtiny PDIP 8 broches. Par exemple, si l'on utilise les broches de port PBO à PB4 comme dans le circuit de la **figure 1**, on ne dispose pas de cinq broches libres consécutives sur la gamme d'AVRs ATtiny24/44/84 à 14 broches, parce que les broches ISP/SPI utilisent celles du port A et du port B. Sur la gamme ATtiny26/261/461/861 20 broches, le port A est entièrement libre avec un programmeur ISP/SPI connecté, mais cela signifie que le logiciel doit être modifié pour utiliser le port A au lieu du port B. Heureusement, la gamme ATtiny2313/4313 20 broches a les broches PBO à PB4 libres avec un programmeur ISP/SPI connecté.

Mettre l'AVR en mode debugWIRE

Pour que le microcontrôleur puisse être programmé à l'aide d'une seule ligne debugWIRE comme le montre la **figure 1**, il est nécessaire de connecter d'abord toutes les lignes du connecteur ISP au microcontrôleur à partir du programmeur/débogueur USB, puis de configurer le fusible **DWEN** de l'AVR pour le mettre en mode debugWIRE. Votre AVR est maintenant en mode debugWIRE. Si ce n'est pas le cas, connectez votre programmeur/débogueur, tel qu'un Atmel-ICE ou un AVR Dragon, et assurez-vous que vous êtes prêt et capable de régler le fusible **DWEN**. Une fois le que le fusible **DWEN** est programmé, toutes les connexions du connecteur ISP peuvent être démontées du circuit à l'exception de RESET, +5 V (Vcc) et GND comme le montre la **figure 1**.

Construire le circuit ATtiny 5-LED sur une platine d'essai

Si vous disposez du matériel nécessaire, construisez le circuit de la **figure 1** sur une plaque d'essai. Assurez-vous que les cinq LED sont connectées en une ligne avec les broches PBO à PB4 connectées dans l'ordre en commençant par PBO à droite. La LED D1 sera donc à droite de la rangée et D5 à gauche. Nous voulons avoir une seule rangée de LED avec PBO connecté à la LED de droite, PB1 connecté à la LED voisine de gauche, PB2 connecté à la troisième LED en commençant par la droite, et ainsi de suite, comme on peut le voir sur le schéma de la **figure 2**. L'image ne montre que les contours des LED afin qu'elles ne masquent pas les connexions des fils et des résistances. Si vous ne disposez pas du matériel nécessaire, suivez les programmes à l'aide d'un simulateur.

Code assembleur pour le circuit du compteur à 5 LED

Créez un nouveau projet en assembleur AVR sur Microchip Studio nommé `led_count_asm`. Tapez le code montré dans le **Listage 1** dans le fichier `main.asm` du projet en remplaçant le programme de base. Si vous utilisez le matériel de la **figure 1**, sélectionnez l'outil matériel (votre débogueur), par exemple Atmel-ICE sous Microchip Studio, avec debugWIRE comme interface. Pour ce faire, cliquez sur l'icône *hammer* dans la deuxième barre d'outils supérieure. Si vous utilisez le simulateur, sélectionnez *simulator* comme outil. Si vous utilisez un programmeur « maison », poursuivez la lecture pour voir comment télécharger le programme sur l'ATtiny cible. Le programme `led_count_asm` configure les broches PBO à PB4 en

sortie de sorte que les LED associées puissent être commandées par le code. Il affiche un nombre binaire croissant, ou compte, sur les LED en commençant par 0 (représenté par toutes les LED éteintes). Lorsque le compteur atteint sa valeur maximale (toutes les LED allumées), il revient à zéro et recommence à compter. Chaque LED « éteinte » représente un chiffre binaire 0 ou un niveau bas, et chaque LED « allumée » représente un chiffre binaire 1 ou un niveau haut. Il est important de disposer les LED comme indiqué sur la **figure 2** afin que le compte s'affiche correctement avec PBo/D1 comme LSB (bit de poids faible) et PB4/D5 comme MSB (bit de poids fort) de la valeur du compteur.

Si vous utilisez le matériel de la **figure 1** et de la **figure 2**, créez le programme et téléversez-le sur l'AVR. Si vous utilisez un Atmel-ICE ou un AVR Dragon, utilisez l'icône Start Without Debugging dans la barre d'outils supérieure de Microchip Studio ou le raccourci clavier **Ctrl + Alt + F5** pour télécharger le programme dans l'AVR. Si l'interface du débogueur est correctement configurée et que la puce est en mode debugWIRE, le programme sera chargé et commencera à fonctionner. Le compte binaire incrémentiel sera indiqué par les LED. Si vous utilisez un programmeur USB *program-only*, ou un que vous avez conçu vous-même, chargez alors le programme sur l'AVR cible en utilisant la fonction appropriée. Si le circuit a été câblé convenablement, et le programme a été saisi sauvegardé et compilé correctement, vous verrez la valeur du compte binaire s'incrémenter sur les LED et vous pourrez utiliser le simulateur dans la suite.

Si vous ne disposez pas des composants nécessaires, vous pouvez toujours voir la valeur du comptage en utilisant le simulateur de Microchip Studio. Le programme peut être exécuté progressivement en utilisant les icônes *Start Debugging* et *Break*. Après avoir démarré le simulateur, ouvrez la fenêtre I/O dans le menu supérieur en cliquant sur *Debug Windows I/O*, puis cliquez sur l'élément I/O Port (PORTB). Exécutez le programme en utilisant l'icône *Step Over* ou la touche **F10** du clavier. Regardez l'élément PORTB au bas de la fenêtre I/O pour voir la valeur de compte qui aurait été affichée sur les LED si elles étaient connectées. Le compteur est mis à jour dans PORTB chaque fois que l'instruction **OUT** est exécutée dans la boucle principale.

Comment fonctionne le programme en assembleur LED Count

La moitié du code *led_count_asm* est constituée du sous-programme de délai qui a été utilisé dans le programme *LED blink* situé ailleurs dans ce livre. Ce sous-programme est appelé une fois dans la boucle principale afin que le comptage sur les LED soit visible à l'œil sans un clignotement trop rapide. Lancez le projet *led_count_asm* dans Microchip Studio, avec le code de *main.asm* également ouvert, tout en suivant l'explication du code qui suit.

Les deux premières instructions du programme sont utilisées pour configurer les broches PBo à PB4 en sortie pour piloter les LED en configurant les bits du registre DDRB. La **figure 3** montre le registre DDRB en haut. Chaque bit de ce registre correspond à une broche du microcontrôleur. Par exemple, le bit DDB0 correspond à la broche PBo, DDB1 à la broche PB1, et ainsi de suite.

Lorsqu'un bit de DDRB est mis à 1 logique, la broche correspondante est configurée en sortie. Si un bit dans DDRB est mis à 0 logique, la



Listage 1 : *led_count_asm* : *main.asm*

```
; Set up pins PB0 to PB4 as output pins
ldi    r16, 0b0001_1111
out    DDRB, r16
clr    r18                ; Clear count register
loop:
out    PORTB, r18         ; Display count on LEDs
rcall  delay
inc    r18                ; Increment count
andi   r18, 0b0001_1111 ; Clear unused bits
rjmp   loop
; Delay subroutine
delay:
ldi    r16, 0xff
deloop1:
ldi    r17, 0xff
deloop2:
dec    r17
brb    SREG_Z, deloop2
dec    r16
brbc   SREG_Z, deloop1
ret
```

DDRB : Port B Data Direction Register. Configures pin direction: 0 = Input, 1 = Output. Address: 0x17 Initial Value: 0b0000_0000 Bits 5 to 0: Read/Write (R/W)							
7	6	5	4	3	2	1	0
—	—	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0

PORTB : Port B Data Register. Drives output pins. Enables input pin pull-ups with logic 1. Address: 0x18 Initial Value: 0b0000_0000 Bits 5 to 0: Read/Write (R/W)							
7	6	5	4	3	2	1	0
—	—	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0

PINB : Port B Input Pins Register. Read input pin state. Write logic 1 to toggle output pin. Address: 0x16 Initial Value: Depends on level on pin. Bits 5 to 0: Read/Write (R/W)							
7	6	5	4	3	2	1	0
—	—	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0

Figure 3. Registres des ports d'E/S d'ATtiny13(A) et ATtiny25/45/85.

broche correspondante est configurée en entrée qui est la configuration par défaut de toutes les broches à la mise sous tension ou à la réinitialisation. Au début du programme, 0b0001_1111 est écrit dans le registre R16, puis écrit à partir de R16 dans le registre DDRB en utilisant la commande **OUT**. Il est nécessaire de charger d'abord la valeur constante dans R16 car il n'y a pas d'instruction pour le faire directement dans un registre d'E/S tel que DDRB. L'écriture de 0b0001_1111 dans DDRB active les bits DDB0 à DDB4 ce qui définit les broches PBo à PB4 en sortie.

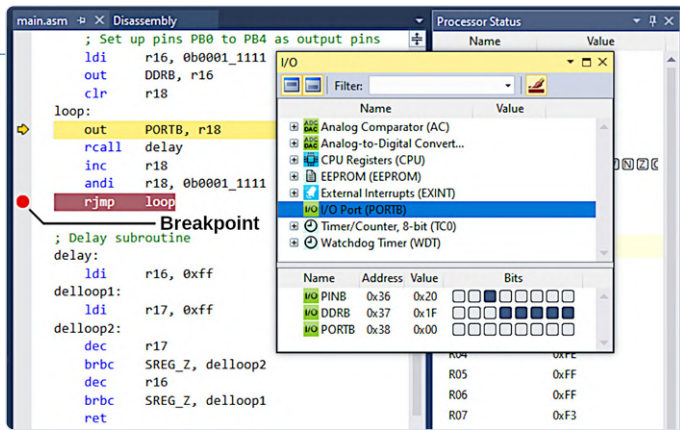


Figure 4. Insertion d'un point d'arrêt dans le débogueur de Microchip Studio.

Bien qu'il y ait quatre registres pour contrôler le port d'E/S B, seuls trois sont représentés sur la **figure 3**. Le quatrième registre, MCUCR, n'a qu'un seul bit utilisé par le port B. Ce bit est un bit global de pull-up que nous n'utilisons pas dans ce chapitre. Référez-vous à la partie description des registres de la fiche technique dans la section ports d'E/S pour voir ce registre - consultez la fiche technique de l'ATtiny13, ATtiny13A, ou ATtiny25/45/85.

R18 est utilisé dans le programme pour maintenir une valeur de compte incrémentielle qui est affichée sur les LED. R18 est mis à 0 à l'aide de **CLR** avant d'entrer dans la boucle principale afin que le compte commence à partir de 0.

Dans la boucle principale, la commande **OUT** permet d'envoyer le contenu de R18 au registre PORTB. Ce dernier est représenté au milieu de la **figure 3**. Encore une fois, chaque bit dans ce registre correspond à une broche sur le microcontrôleur. Les niveaux logiques écrits dans le registre PORTB apparaissent sur les broches qui ont été configurées en sortie en utilisant DDR. Pour les broches qui sont configurées en entrée, 1 logique dans PORTB active une résistance de rappel haut interne sur ces broches. 0 logique désactive la résistance de rappel haut sur la broche correspondante.

Comme les broches PBo à PB4 sont configurées en sortie, la valeur de compte écrite au PORTB est représentée par les valeurs logiques qui allument et éteignent les LED connectées. Le niveau logique 1 dans la valeur de comptage allume la LED correspondante, et le niveau logique 0 dans la valeur de comptage éteint la LED correspondante.

Après que la valeur de compteur soit écrite sur PORTB en utilisant la commande **OUT**, le sous-programme de délai est appelé pour la maintenir affichée sur les LED pendant une courte durée. La valeur sur le registre R18 est alors incrémentée de 1 en utilisant la commande **INC**. **ANDI** est utilisée pour effacer les trois bits supérieurs de la valeur de compte dans R18, en utilisant une valeur de masque de 0b0001_1111. Cela garantit que ces bits de poids fort sont toujours à zéro. De retour au début de la boucle, lorsque R18 est à nouveau écrit dans PORTB, 0 est toujours écrit dans les trois bits de poids fort, car ils ont été effacés en utilisant **ANDI**. Lorsque la fin de la boucle est atteinte à l'instruction **RJMP**, l'exécution du programme recommence, avec la nouvelle valeur du compteur écrite dans PORTB, et affichée sur les LED.

Notez que dans ce programme : les registres d'E/S DDRB et PORTB sont tous deux utilisés pour configurer et contrôler le port B du microcontrôleur dans les deux programmes. Le programme LED blink n'a besoin de contrôler qu'une seule broche, il utilise donc les instructions **SBI** et **CBI** pour activer et désactiver un seul bit dans

les registres d'E/S directement — aucun des registres de travail R0 à R31 n'est nécessaire pour cela. Le programme de compteur à LED a besoin d'accéder à cinq LED ou broches en même temps et utilise donc l'instruction **OUT** pour écrire sur plusieurs bits dans un registre en même temps.

L'utilisation des registres dans un programme en langage assembleur est indispensable pour suivre son progrès. Par exemple, une valeur immédiate est chargée dans R16 au début du programme. R16 n'est utilisé que temporairement dans ce cas, et peut donc être utilisé plus tard sans avoir besoin de sauvegarder sa valeur au préalable. Il est à nouveau utilisé dans le sous-programme de délai. Comme R16 et R17 sont utilisés dans le sous-programme de retard, R18 a été choisi pour contenir la valeur de comptage uniquement. Cela garantit que la valeur du compteur n'est jamais écrasée. Si un programme devient très large et qu'il n'y a pas de registres libres à utiliser pour des objectifs dédiés, alors les commandes **PUSH** et **POP** peuvent être utilisées au début et à la fin du sous-programme pour maintenir les valeurs dans les registres en cours d'utilisation.

Utilisation d'un point d'arrêt dans le débogueur

En utilisant le simulateur, ou le matériel et un débogueur tel que l'Atmel-ICE, utiliser le débogueur pour progresser dans le programme en cliquant sur **Start Debugging** et **Break** comme nous l'avons fait précédemment. Visualisez les registres du port d'E/S en ouvrant **I/O Window** dans Microchip Studio. Lorsque le débogueur est en cours d'exécution, sélectionnez **Debug Windows I/O**, puis cliquez sur **I/O Port (PORTB)** dans **I/O window**, comme illustré à la **figure 4**.

Lorsque l'exécution du programme entre dans la boucle principale, il est pratique de placer un point d'arrêt sur la commande **RJMP** et d'exécuter la boucle entière en cliquant sur le bouton **Continue** de la barre d'outils (touche F5). Un point d'arrêt peut être placé sur la commande **RJMP** en cliquant sur la zone grise située à l'extrême gauche de l'instruction et de la fenêtre du Microchip Studio. Cela fait apparaître un point rouge dans cette zone, indiquant qu'un point d'arrêt a été placé sur l'instruction, comme on peut le voir sur la **figure 4**. Vous pouvez également cliquer sur la commande sur laquelle vous souhaitez placer un point d'arrêt afin que le curseur soit placé dessus, puis sélectionner **Debug Toggle Breakpoint** dans le menu supérieur ou utiliser la touche F9 du clavier. Une fois le point d'arrêt défini, utilisez le bouton **Continue** de la barre d'outils ou appuyez sur la touche F5 pour parcourir toute la boucle et ne l'interrompre qu'en bas. De cette façon, on peut voir le compte s'incrémenter de 1 dans le PORTB dans la fenêtre I/O ainsi que dans le registre R18 dans la fenêtre **Processor Status**, sans avoir à parcourir individuellement chaque instruction de la boucle.

Supprimez le point d'arrêt de la commande **RJMP** en cliquant sur le point rouge à gauche de l'instruction, ou désactivez-le en utilisant le menu ou la touche F9, nous supposons que le curseur est sur la ligne de la commande **RJMP**. Placez un point d'arrêt sur la commande **INC R18**. Utilisez maintenant F5 pour progresser dans la boucle. Le nouveau point d'arrêt garantit que le registre PORTB et R18 contiennent la même valeur lorsque l'exécution du programme s'arrête. Si l'exécution du programme s'arrête sur **RJMP**, alors R18 est incrémenté avant d'écrire sa nouvelle valeur à PORTB, donc ils sont désynchronisés lors de l'observation de ces valeurs dans les

fenêtres du débogueur dans Microchip Studio.

Le logiciel associé au livre publié par l'auteur peut être téléchargé gratuitement. Visitez [1], descendez jusqu'aux téléchargements, et cliquez sur le nom du fichier : *Software_Explore ATtiny Microcontrollers using C and Assembly Language*. Save the ZIP archive file locally (environ 29 kb) et extrayez-le. ❏

220045-04

Contributeurs

Texte et image : Warwick Smith

Éditeur : Jan Buiting

Mise en page : Giel Dols

Traduction : Asma Adhimi

Des questions, des commentaires ?

Envoyez un courriel à l'auteur (warwsmi@axxess.co.za) ou contactez Elektor (redaction@elektor.fr).

LIEN

[1] Ressources et informations sur le livre: www.elektor.fr/20007



PRODUITS

➤ Livre en anglais « Explore ATtiny Microcontrollers using C and Assembly Language » de Warwick A. Smith

Version papier : www.elektor.fr/20007

Version numérique : www.elektor.fr/20008



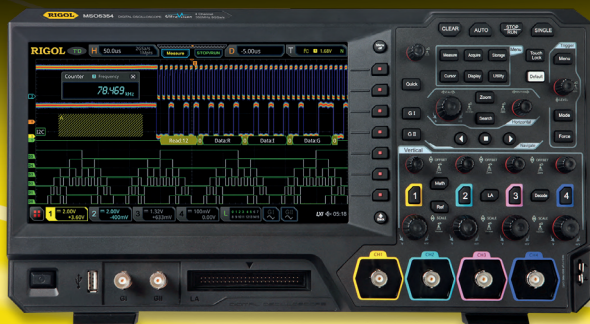
Publicité

RIGOL

Possibilities and More

Oscilloscopes numériques : Puissants et économes

UltraVision II Technology



Disponible immédiatement → à partir de € 809,- HT

Profitez vite de prix réduits sur une sélection de modèles de la Série!

Série MSO5000

Oscilloscopes numériques haut de gamme

- Incluant la représentation des diagrammes de Bode
- Bande passante analogique de 70, 100, 200 et 350 MHz (par le biais d'une mise à niveau du logiciel)
- 2 (70/100 MHz) ou 4 canaux analogiques (Mise à niveau) + 16 canaux numériques (MSO)
- Taux d'échantillonnage temps réel jusqu'à 8 GS/s
- Profondeur mémoire jusqu'à 200 Mpts *
- Taux de capture 500 000 formes d'onde/s

Offre Speciale → offerts jusqu'au 30.06.2022 :

Analyse de protocole, générateur de formes d'onde, analyse de puissance

* Option

X-IN-1
WORKSTATION



RIGOL Technologies EU GmbH
Téléphone +49 8105 27292-0
info-europe@rigol.com
<https://rigolshop.eu>

www.rigol.eu



projet 2.0

Corrections, mises à jour et courriers des lecteurs

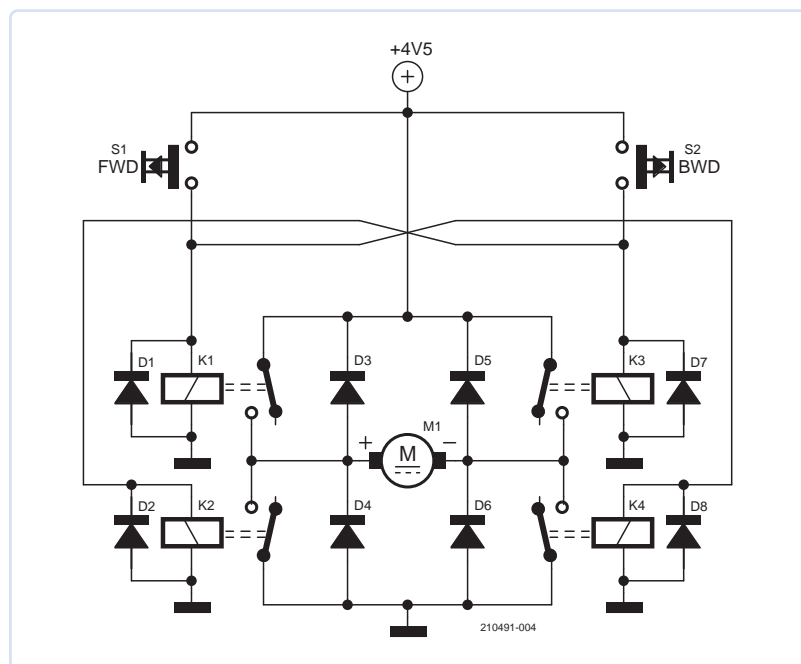
Ralf Schmiedel et Jens Nickel (Elektor)

Commande de moteurs : les ponts en H

Elektor 01-02/2022, p. 6 (210491-04)

Malheureusement, le schéma du pont en H avec relais de cet article (fig. 4) ne comporte pas de connexion entre la borne M- du moteur et les contacts communs des relais K3 et K4. Nous avons inclus ici une version corrigée du schéma. Nous remercions les lecteurs les plus attentifs qui ont remarqué cette erreur.

Une autre erreur s'est glissée dans les descriptions des figures 14 et 20. La puce représentée est en fait un L298N et non un L294N comme le suggère la description. Les deux derniers schémas de la figure 23 ne montrent pas la séquence correcte d'excitations du champ et de positions du rotor. Voici l'image correcte :



Traceur de fuites à la terre simple

Elektor 01-02/2022, p. 106 (200576-04)

Il est essentiel que le traceur de détection des fuites à la terre décrit dans cet article ne soit utilisé que pour tester des appareils complètement isolés du réseau électrique.

Il peut s'avérer mortel s'il est utilisé sur un appareil alimenté par le secteur ! Le traceur de fuites à la terre fonctionne bien pour dépanner les appareils portables, mais uniquement lorsqu'ils sont hors tension et isolés.

Jörg Stäudle

Merci de souligner ce point, Jörg. Vous avez tout à fait raison.
L'équipe de rédaction

Une rose des vents avec le capteur GY-271

Elektor 07-08/2021, p. 78 (200597-04)

J'ai trouvé votre article concernant le module boussole GY-271 très intéressant, mais je ne comprends pas pourquoi il faut faire des huit avec le capteur.

Jac Hettema

J'ai fait quelques recherches pour savoir s'il existait un moyen d'améliorer la précision de la boussole d'un téléphone intelligent et j'ai trouvé plusieurs sites web, dont celui-ci : <https://bit.ly/3rQfdYJ>. Certains experts en téléphonie mobile affirment qu'effectuer des huit avec le capteur permet de le faire tourner deux fois sur 360° et de réaliser un étalonnage.
Rolf Hase (auteur de l'article)

Charge électronique en CC et CA

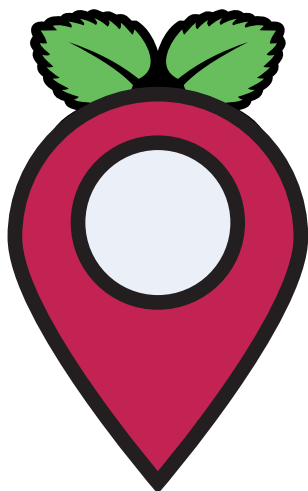
Elektor 07-08/2021, p. 20 (191206-04)

Le schéma (fig. 2) indique que la diode Zener D3 présentée à la page 23 a un seuil de tension inverse de 3,3 V, mais elle devrait en fait avoir un seuil de 10,0 V.

220052-04

Des questions, des commentaires ?

Contactez Elektor (redaction@elektor.fr).



mise à jour de la balise GPS LoRa

Recevoir et afficher la localisation à l'aide d'un Raspberry Pi

Hans Schneider (Allemagne)

L'article intitulé « balise GPS LoRa » publié dans *Elektor* (11-12/2020) décrit comment récupérer les données de suivi sur un serveur TTN à l'aide de Node-RED, puis les visualiser sur une carte. En principe, cela devrait fonctionner, non seulement avec un PC, mais aussi avec un Raspberry Pi. C'est effectivement possible, mais il y a un ou deux obstacles à franchir. Un de nos lecteurs a trouvé une bonne solution et décrit comment il a porté le flux Node-RED sur l'ordinateur monocarte.

Dans l'article « balise GPS LoRa » (*Elektor*, 11-12/2020), Mathias Claussen décrit le matériel et le logiciel utilisables pour suivre des objets en mouvement. Un module de suivi compact envoie les données d'un capteur GPS via LoRa à une passerelle, qui les transmet à un serveur *The Things Network*, d'où les données brutes peuvent être lues manuellement via l'internet, sous réserve d'avoir le droit d'accès. Bien sûr, c'est plus pratique d'y accéder automatiquement via un *framework* tel que Node-RED, qui peut également gérer l'affichage sur une carte (du monde). Le flux Node-RED décrit dans l'article convient très bien pour un PC, mais, pour un fonctionnement automatique, il a le grave inconvénient de nécessiter que le serveur Node-RED et un PC assoiffé d'énergie soient tous deux actifs en permanence.

Il est probable que de nombreux particuliers férus de technologie possèdent déjà au moins un Raspberry Pi sur lequel tourne un logiciel tel que Pi-Hole [1], Homebridge [2] ou Flightradar24 [3], en attente permanente d'échange de données. Pourquoi ne pas lui confier une tâche de plus ? Mais attention : les ressources nécessitées par Node-RED peuvent parfois entrer en conflit avec une installation de Homebridge. En outre, la procédure d'installation n'est pas non plus totalement exempte de difficultés.

Installation de Node-RED sur le RPi

Pour commencer, nous devons installer Node-RED sur le Raspberry Pi. Le lecteur est supposé savoir saisir des commandes dans la fenêtre de terminal du RPi. Ensuite, nous devons nous connecter au terminal via SSH, en utilisant la ligne de commande suivante : `bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)`. Vous devrez peut-être entrer `sudo apt install build-essential git` pour permettre à `npm` de compiler les modules binaires qui doivent être installés. L'exécution du script prend un certain temps, de l'ordre de la demi-heure.

Lorsque le script est terminé, la commande `sudo systemctl enable nodered.service` provoquera le démarrage automatique de Node-RED à la mise sous tension ou au redémarrage du RPi. Cette commande peut être désactivée en entrant `sudo systemctl disable nodered.service`.

Le serveur Node-RED peut être démarré avec `node-red-start` et arrêté avec `node-red-stop`. Pour arrêter et redémarrer le serveur, utilisez `node-red-restart`. Le terminal peut être quitté après avoir démarré Node-RED, et le serveur continuera à fonctionner. Lorsque le terminal est redémarré, vous pouvez démarrer le service de sortie du serveur en utilisant `node-red-log` et voir la sortie dans le terminal. Ceci est utile si vous voulez vérifier si tout fonctionne toujours sans démarrer une nouvelle instance du serveur.

Installation des modules requis dans Node-RED

Les modules peuvent être installés dans Node-RED en utilisant soit le « gestionnaire de palette » dans la fenêtre du navigateur, soit le gestionnaire de paquets `npm` depuis le terminal. Je recommande d'utiliser `npm` dont le fonctionnement me paraît plus fiable, auquel cas Node-RED doit être arrêté au préalable.

Par contre, si le service Node-RED est actif sur le RPi, vous pouvez vous connecter à l'interface utilisateur en utilisant un navigateur sur le PC et en saisissant le lien : `http://<adresse IP du RPi>:1880`, ce qui vous connectera à l'« éditeur de flux ». Les modules requis sont installés en premier ; dans l'« éditeur de flux », cela se fait via le « gestionnaire de palette ». Les modules dont vous aurez besoin



Figure 1. Nœud MQTT dans Node-RED.

Figure 2. Paramétrage des propriétés du serveur du nœud MQTT.

Figure 3. Ajouter un nouveau serveur.

Figure 4. Configuration TLS du nouveau serveur.

(dont certains ont déjà été installés lors de l'installation de Node-RED) sont les suivants :

- > *node-red*
- > *node-red-contrib-worldmap*
- > *node-red-node-rbe*
- > *node-red-node-sqlite*
- > *node-red-node-base64*
- > *node-red-tail*

L'installation via le « gestionnaire de palette » demande un peu de patience ; si le moniteur d'activité a été masqué, il faut un certain temps avant que le message de succès n'apparaisse en haut de la fenêtre. Si l'installation via le « gestionnaire de palette » échoue, quittez l'éditeur et, dans le terminal, arrêtez le service Node-RED sur le RPi et procédez manuellement. Par exemple, la commande `npm i --unsafe-perm node-red-node-sqlite` installe le module *node-red-node-sqlite*, et vous pouvez faire de même pour les autres modules.

Si *node-red-node-sqlite* n'installe pas le module *node-red-sqlitedb*, il faudra l'installer par la suite. Ceci n'est possible que via *npm*, car dans ce cas le « gestionnaire de palette » détecte un conflit avec le module *node-red-node-sqlite*.

Le module TTN manquant

Tout devrait s'être bien passé jusqu'ici, sauf que vous avez peut-être remarqué l'absence du module TTN (*The Things Network*) présenté dans l'article original. En effet il n'est pas pris en charge par l'environnement Raspberry Pi. Par exemple, une fois que *node-red-contrib-ttn* a été chargé, il n'est plus possible de démarrer le service Node-RED sur le RPi. Il m'a fallu un moment pour le comprendre, alors il m'est agréable de penser au temps que cette information va vous faire gagner !

La solution réside dans le module *mqtt* intégré à Node-RED. Il est inclus dans le flux d'origine plutôt que dans le module de liaison montante TTN. Vous pouvez utiliser mon flux, spécialement adapté au RPi et téléchargeable sur la page de la balise GPS LoRa [5] d'Elektor Labs. Le flux est importé en faisant simplement glisser le fichier téléchargé JSON dans la fenêtre de l'« éditeur de flux ». D'ailleurs, ce flux adapté au RPi fonctionne également sur le PC à la place des flux avec les nœuds TTN spéciaux.

Avec l'« éditeur de flux », vous pouvez maintenant double-cliquer sur le symbole *mqtt* (fig. 1) et modifier les paramètres dans la fenêtre qui s'ouvre (fig. 2). Maintenant, « v3/+/devices/+/up » est saisi sous *Topic* comme indiqué sur la figure afin de recevoir les données de tous les appareils de l'application TTN. Pour « output » (sortie), vous pouvez conserver la configuration par défaut « a parsed JSON object ». Un nouveau serveur peut être saisi dans le champ « Server » à l'aide du symbole du crayon ce qui ouvre la fenêtre illustrée à la figure 3. L'ID d'application de la console TTN est utilisé comme « name » (nom), et *eu1.cloud.thethings.network* est le « server » (serveur). Le port par défaut est 1833 (MQTT sans TLS) qui doit être changé en 8883. Enfin, il manque la coche à côté de « Enable secure (SSL/TLS) connection ». Un clic sur le symbole du crayon du champ « Configuration TLS » ouvre la fenêtre suivante (fig. 4).

Maintenant, les paramètres peuvent être acceptés en cliquant sur



Figure 5. Gestion des clés d'accès des nœuds MQTT.

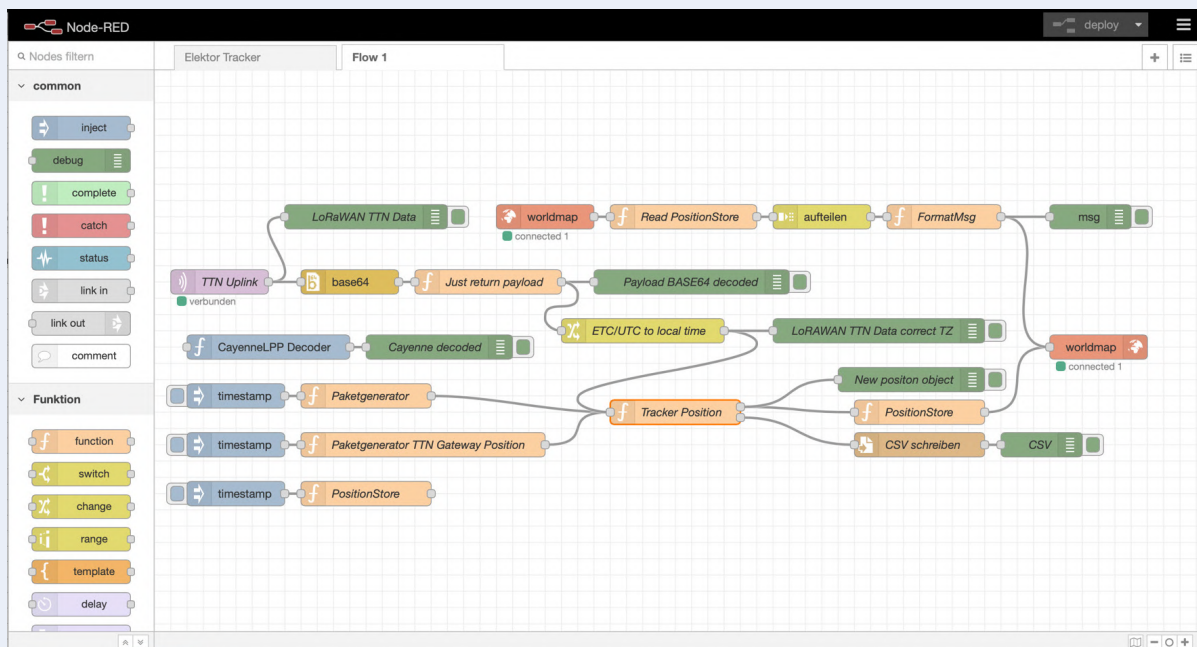



Figure 6. Le diagramme de flux terminé dans Node-RED.

« Add ». De retour dans la fenêtre « mqtt-broker config node », sélectionnez l'onglet « Security » (**fig. 5**). Le « Username » est l'ID d'application de la console TTN suivi de @ttn. Une clé d'accès avec au moins « Messages Rights » doit être produite comme mot de passe dans la console TTN. La clé d'accès copiée de la console TTN est ensuite collée sous « Password » et confirmée en cliquant sur « Add ». De retour dans l'élément *mqtt in*, vous pouvez donner un nom à cet élément (par ex. *TTN uplink*) ; enfin, un clic sur « Done » termine la tâche. Le **figure 6** montre le diagramme de flux Node-RED mis à jour. Le décodeur CayenneLPP présenté dans le schéma est inclus pour une extension future du système. 

210120-04



PRODUITS

- **Elektor LoRa Node** - module partiellement assemblé
www.elektor.fr/19175
- **RFM95** - module émetteur-récepteur LoRa à très longue portée (868 MHz) de SeeedStudio
www.elektor.fr/18715
- **PG1301** - concentrateur GPS LoRaWAN pour Raspberry Pi (868 MHz) de Dragino
www.elektor.fr/19367

Contributeurs

Idée et texte : Hans Schneider

Rédaction : Mathias Claußen, Jens Nickel, C.J. Abate

Traduction : Helmut Müller

Mise en page : Harmen Heida

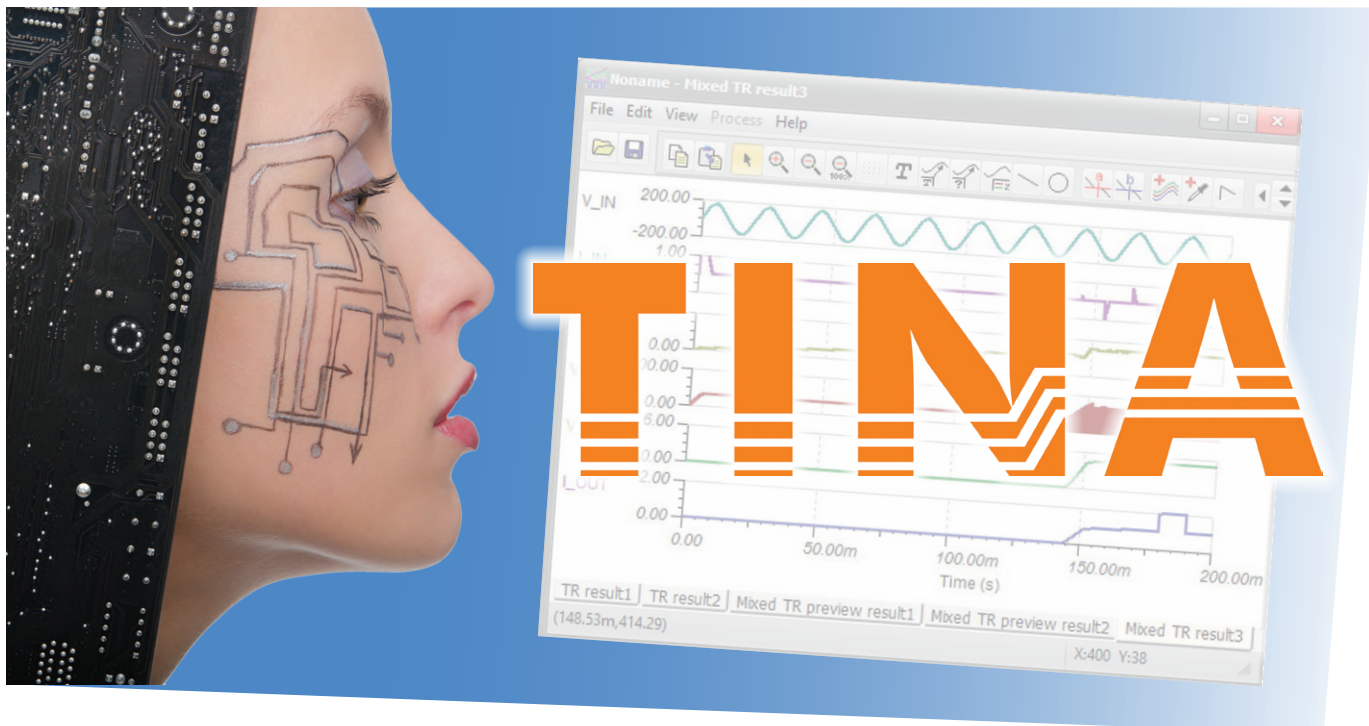
Des questions, des commentaires ?

Envoyez un courriel à l'auteur (hans.schneider@belgacom.net) ou contactez Elektor (redaction@elektor.fr).

LIENS

- [1] « balise GPS LoRa », M. Claußen, Elektor 11-12/2020 : www.elektormagazine.fr/200096-02
- [2] Pi-hole : <https://pi-hole.net>
- [3] Homebridge : <https://homebridge.io>
- [4] Flightradar24 : www.flightradar24.com/
- [5] « LoRa GPS Tracker », Elektor Labs : www.elektormagazine.fr/labs/lora-gps-tracker





Circuit Simulation with **TINA Design Suite & TINACloud**

Extrait : oscillateurs sinusoïdaux

Dogan Ibrahim (Royaume-Uni)

Selon certains, l'une des meilleures choses de TINA sera la possibilité de transposer facilement un circuit simulé sur un circuit imprimé grâce aux fonctions de placement et de routage automatiques. En effet, les options de traçage gerber et de perçage CNC permettent d'utiliser TINA pour concevoir et mettre en œuvre des prototypes. C'est bien sûr génial, mais avant de se lancer dans la conception des platines, il faut d'abord apprendre à maîtriser la simulation de circuits élémentaires. Voici une introduction à TINA à ne pas manquer.

Un oscillateur sinusoïdal est constitué d'un amplificateur et d'un réseau de rétroaction (**figure 1**). Les deux conditions suivantes doivent être remplies pour que l'oscillateur fonctionne :

- le gain de la boucle ($A \times B$) de la **figure 1** doit être supérieur ou égal à 1 ;
- le déphasage total dans le circuit doit être de 0 ou 360°.

Quelques simulations de circuits oscillateurs à base d'amplificateurs opérationnels sont proposées dans cette section.

Simulation 1 — Oscillateur à déphasage

On le désigne parfois sous le nom d'oscillateur RC. Chaque paire RC introduit un déphasage de 60°. Trois résistances et condensateurs sont utilisés ici pour introduire un déphasage de 180° dans la boucle de retour. Le déphasage total de la boucle est donc de 0° comme requis pour l'oscillation.

Schéma de TINA

La **figure 2** montre le schéma du circuit. Le réseau RC est connecté à l'entrée d'inversion de l'amplificateur opérationnel. En supposant que les résistances et les condensateurs sont identiques, l'exigence est que le gain en tension de l'amplificateur doit être supérieur ou égal à 29, c'est-à-dire :

$$\text{Gain} = \frac{R_f}{R} \geq 29$$

La fréquence d'oscillation est donnée par :

$$f = \frac{1}{2\pi RC\sqrt{6}}$$

Dans cet exemple, la fréquence requise est de 4 kHz. En choisissant $C = 2 \text{ nF}$, on peut trouver la valeur requise de R à partir de :

Note de l'éditeur. Cet article est un extrait du livre de 440 pages *Circuit Simulation with TINA Design Suite & TINACloud* (Elektor 2022), formaté et légèrement modifié pour correspondre aux normes éditoriales et à la mise en page du magazine Elektor. Puisque cet article est extrait d'une publication plus vaste, certains termes peuvent faire référence à des passages du livre d'origine situés ailleurs. L'auteur et l'éditeur ont fait de leur mieux pour l'éviter et seront heureux de répondre aux questions. Pour les contacter, voir l'encadré « Des questions, des commentaires ? ».

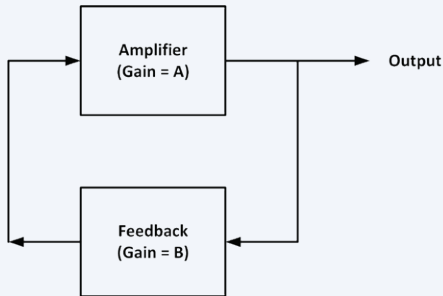


Figure 1. Principe de l'oscillateur.

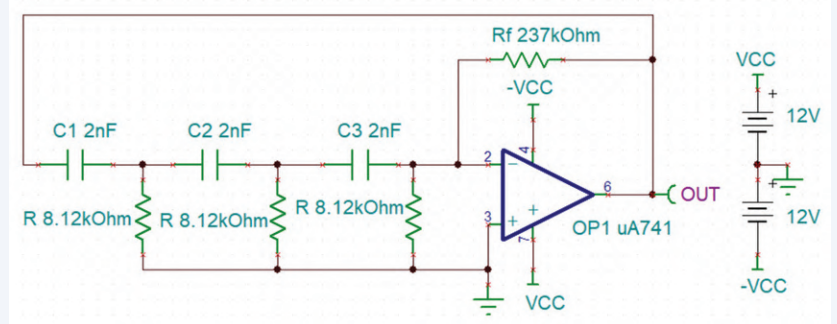


Figure 2. Schéma du circuit.

$$R = \frac{1}{2\pi f C \sqrt{6}} = \frac{1}{2\pi \times 4 \times 10^3 \times 2 \times 10^{-9} \times \sqrt{6}}$$

ce qui donne $R = 8,12 \text{ k}\Omega$. Alors,

$$\frac{R_f}{R} \geq 29$$

et nous choisissons $R_f = 237 \text{ k}\Omega$. Dans ce projet, nous utilisons un amplificateur opérationnel de type UA741. Le circuit TINA est disponible sous forme d'un fichier sim9 (voir fin de l'article).

Simulation TINA

Voici les étapes à suivre pour lancer la simulation :

- cliquez sur T&M -> Oscilloscope et Run. Réglez Time/div sur 100 u.
- vérifiez la forme d'onde de sortie (figure 3). La période est de 250 μs , ce qui correspond à 4 kHz. Notez la pointe dans la forme d'onde de sortie sur l'oscilloscope. L'oscilloscope peut être synchronisé pour créer une image stable. Pour ce faire, sélectionnez Normal sous Mode et synchronizing signal sous Source. Vous devrez peut-être aussi régler le niveau de déclenchement (Trigger).

Simulation 2 — Oscillateur à pont de Wien

Il s'agit de l'un des oscillateurs sinusoïdaux les plus simples. L'oscillateur à pont de Wien est un circuit couplé RC à deux étages qui présente une bonne stabilité à sa fréquence de résonance, une faible distorsion et est très facile à accorder, ce qui en fait un circuit populaire en tant qu'oscillateur de fréquence audio.

Le circuit utilise un circuit RC en série connecté à un circuit RC en parallèle. Le déphasage du circuit est de 0° à la fréquence de résonance et le circuit est connecté à l'entrée positive de l'amplificateur opérationnel de sorte que le déphasage global est de 0° . Habituellement, on utilise une configuration d'amplificateur non-inverseur.

Pour obtenir une oscillation, il faut que le gain en tension soit supérieur ou égal à 3.

En supposant que les mêmes résistances et condensateurs sont utilisés, la fréquence d'oscillation est donnée par :

$$f = \frac{1}{2\pi RC}$$

Dans cet exemple, la fréquence requise est de 5 kHz. En choisissant $C = 3 \text{ nF}$, nous pouvons déterminer la valeur requise de R à partir de :

$$R = \frac{1}{2\pi f C} = \frac{1}{2\pi \times 5 \times 10^3 \times 3 \times 10^{-9}}$$

ce qui donne $R = 10,6 \text{ k}\Omega$.

Pour remplir la condition de gain, pour un amplificateur non-inverseur (voir la figure 4) :

$$\text{Gain} = 1 + \frac{R_f}{R_2}$$

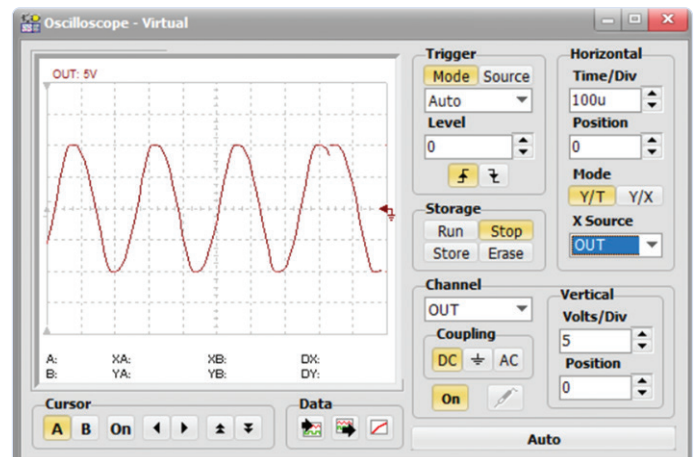


Figure 3. Forme d'onde de sortie.

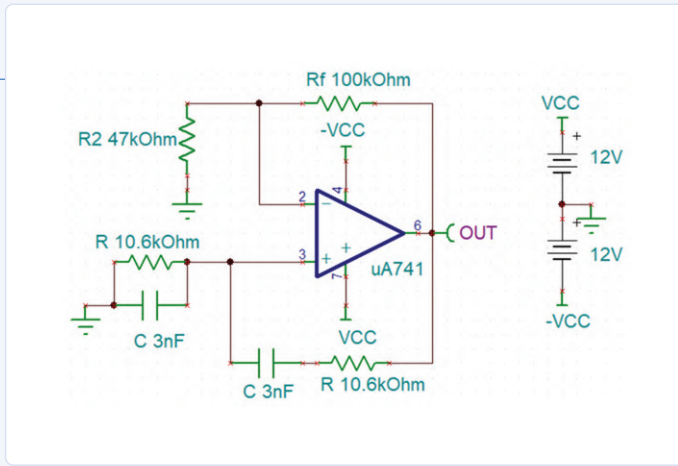


Figure 4. Schéma du circuit.

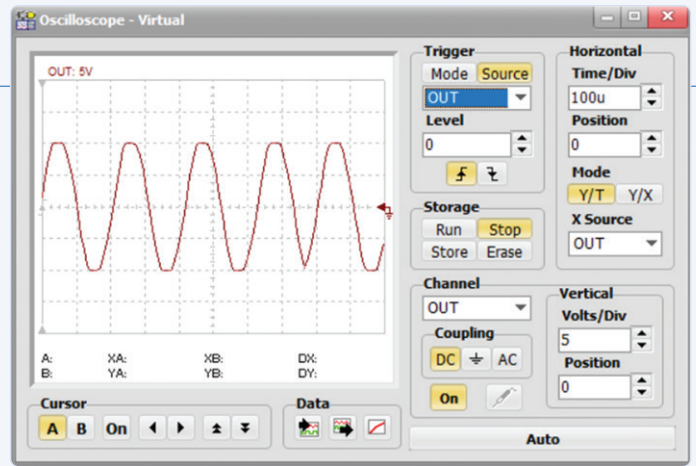


Figure 5. Forme d'onde de sortie.

Choisissez $R_f = 100 \text{ k}\Omega$:

$$1 + \frac{R_f}{R_2} \geq 3$$

Ce qui donne :

$$R_2 = \frac{R_f}{3-1} \leq 50 \text{ k}\Omega$$

Choisissez $R_2 = 47 \text{ k}\Omega$.

Schéma TINA

La **figure 4** montre le schéma du circuit. Le circuit de rétroaction est connecté à l'entrée non inverseuse de l'amplificateur opérationnel et le gain est réglé par R_f et R_2 . Dans ce projet, nous utilisons un amplificateur opérationnel de type UA741. Le circuit est disponible sous forme d'un fichier : *sim10*.

Simulation TINA

Les étapes pour lancer la simulation sont :

- cliquez T&M -> Oscilloscope et cliquez Run. Réglez Time/div sur 100u.
- vérifiez la forme d'onde de sortie (**figure 5**). La période est de 200 μs , ce qui correspond à 5 KHz.

Simulation 3 – Oscillateur de Colpitts

L'oscillateur de Colpitts utilise un réseau diviseur de tension capacitif comme source de rétroaction. Deux condensateurs, C_1 et C_2 , sont placés aux bornes d'une seule inductance commune, L , où C_1 , C_2 et L forment le circuit LC. Le circuit de rétroaction est généralement connecté à l'entrée négative. C_1 , C_2 et L assurent le déphasage supplémentaire de 180° nécessaire pour que le déphasage total soit de 0° .

La fréquence d'oscillation est donnée par :

$$f = \frac{1}{2\pi\sqrt{LC_T}}$$

Où C_T est l'équivalent série de C_1 et C_2 , c'est-à-dire :

$$C_T = \frac{C_1 C_2}{C_1 + C_2}$$

Le taux de rétroaction dépend des valeurs de C_1 et C_2 . Par conséquent, en modifiant les valeurs de ces condensateurs, nous pouvons ajuster la valeur de la tension de rétroaction renvoyée vers le circuit LC. Le rapport entre C_1 et C_2 représente le taux de rétroaction B :

$$B = C_1 / C_2$$

Pour les oscillations, $AB \geq 1$, où A est le gain de l'amplificateur.

$$\text{Ou, } A \geq C_2 / C_1.$$

Dans cet exemple, la fréquence requise est de 10 KHz. En choisissant $L = 10 \text{ mH}$, nous pouvons trouver le taux de rétroaction de :

$$C_T = \frac{1}{4\pi^2 f^2 L} = \frac{1}{4\pi^2 \times 10^8 \times 10 \times 10^{-3}}$$

ce qui donne 25,3 nF.

En choisissant $C_2 = 250 \text{ nF}$, on obtient $C_1 = 28,15 \text{ nF}$, ce qui correspond à un taux de rétroaction de $B = C_2/C_1 = 250/28,15 = 8,88$. Nous pouvons donc choisir le gain de l'amplificateur à environ 10 (voir la **figure 6**), ce qui donne :

$$\text{Gain} = \frac{R_f}{R_1} = 10$$

En choisissant $R_1 = 5 \text{ k}\Omega$, on obtient $R_f = 50 \text{ k}\Omega$.

Schéma TINA

La **figure 6** montre le schéma du circuit. Le circuit de rétroaction est connecté à l'entrée inverseuse de l'amplificateur opérationnel et le gain est réglé par R_f et R_1 . Un amplificateur opérationnel de type UA741 est utilisé dans ce projet. Le circuit est disponible sous forme de fichier : *sim11*.

Simulation TINA

Les étapes pour lancer la simulation sont :

- cliquez sur T&M -> Oscilloscope et Run. Réglez Time/div sur 50u.
- vérifiez la forme d'onde de sortie (**figure 7**). La période est de 100 μs (100u) ce qui correspond à 10 KHz.

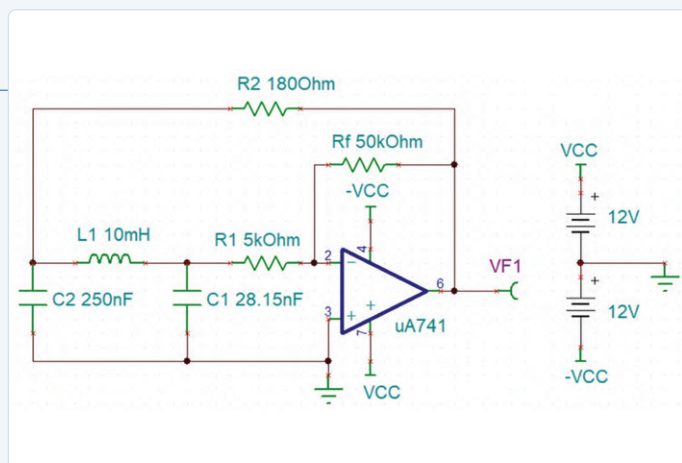


Figure 6. Schéma du circuit.

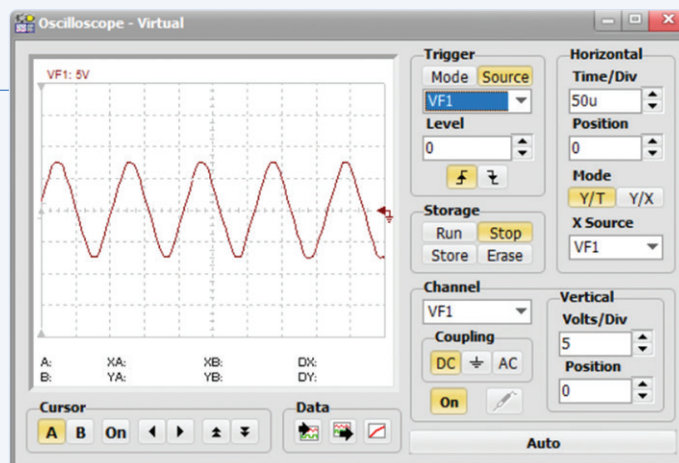


Figure 7. Forme d'onde de sortie.

Où l'obtenir ?

Les fichiers de simulation TINA mentionnés dans cet article sont contenus dans le paquet de logiciels publié par l'auteur et DesignSoft à l'appui du livre. Le logiciel peut être téléchargé gratuitement. Rendez-vous sur le site [1], faites défiler la page jusqu'aux téléchargements et cliquez sur le nom de ce fichier :

Contents_Circuit Simulation with TINA Design Suite & TINACloud.

Enregistrez localement le fichier ZIP (2,45 Mo) puis extrayez-le. Ouvrez votre version de TINA et chargez les fichiers *sim9*, *sim10*, *sim11* tels que mentionnés dans cet article. N'hésitez pas à les modifier en fonction de vos propres applications. ◀

Offre à durée limitée : Le livre TINA est fourni avec une licence gratuite d'un an de TINA Cloud Basic Edition.

220025-04

Contributeurs

Texte et images : Dogan Ibrahim

Éditeur : Jan Buiting

Mise en page : Giel Dols

Traduction : Asma Adhimi

Des questions, des commentaires?

Envoyez un courriel à l'auteur (d.ibrahim@btinternet.com) ou contactez Elektor (redaction@elektor.fr).

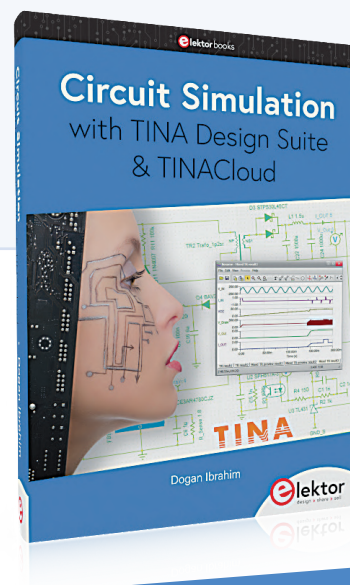


PRODUITS

► Livre en anglais « Circuit Simulation with TINA Design Suite & TINACloud » de Dogan Ibrahim

Version papier : www.elektor.fr/19977

Version numérique : www.elektor.fr/19978



Les deux livres sont accompagnés d'une licence d'un an pour TINACloud Basic Edition. (Offre à durée limitée)

LIEN

[1] Page de ressources/informations sur le livre: www.elektor.fr/circuit-simulation-with-tina-design-suite-tinacloud

zone D

Astuces, bonnes pratiques et autres informations pertinentes

sur le vif

Façon de pincer

Ilse Joostens (Belgique)

Il y a quelque temps, j'ai décidé d'abandonner ces « bons vieux » traversants au profit des CMS. Je craignais un peu leur formidable petitesse, mais finalement ils se sont révélés moins diaboliques que je ne l'imaginais. Je me suis même mise à rêver de circuits conçus par moi-même et « *made in CMS* ». J'ai vite déchanté. Avant même que je ne le réalise, mes journées étaient devenues une longue suite d'heures passées à assembler des circuits avec des pincettes. La corvée, quoi. L'emprise était telle que le soir je voyais encore défiler devant mes yeux des rangées de CMS. C'est sans doute ce qui explique ma réaction légèrement irritée lorsque je découvris mon cadeau d'anniversaire : une « broderie diamant », visiblement choisie pour contribuer à ma relaxation générale. Car, selon le site du vendeur, la broderie diamant réduit le stress et permet de réaliser des chefs-d'œuvre à couper le souffle. Sans doute, mais pour moi cela ressemblait furieusement à de l'assemblage de CMS.

Veiller au grain

Il y a un peu plus de 10 ans, je travaillais pour une petite entreprise allemande faisant assembler tous ses circuits imprimés en Chine. Nous employions le terme *chicken feed* (ou *grit*, « grain ») dans nos discussions avec nos sous-traitants. Cet euphémisme gallinacé désigne les composants passifs

bon marché montés en grand nombre sur un circuit, p. ex. les condensateurs en céramique servant au découplage des rails d'alimentation. Hélas pour nous, l'assembleur chinois d'un de nos contrats dut un jour prendre au pied de la lettre le terme *grit*. Les composants semblaient avoir été jetés sur les cartes par un ancien éleveur de poules, beaucoup

étaient de travers ou montés à l'envers, et je ne parle même pas des soudures. Certains étaient même fissurés, comme s'ils avaient été placés à l'aide d'une emboutisseuse. D'où, nécessairement, un long re-travail sur les circuits, doublé du stress de savoir que des clients attendaient nos produits.

Les cartes elles-mêmes étaient d'une qualité douteuse. Il était presque impossible d'en dessouder les composants sans soulever en même temps les pastilles. Autre souci, la présence de pistes interrompues, et parfois de courts-circuits indésirables entre les pistes. C'en était trop. J'ai fait part de mon exaspération à notre fournisseur allemand. Il a haussé les épaules et s'est contenté d'un désinvolte : « Qualité chinoise ! » Je ne souhaitais pas revivre ça avec mes produits et décidai de m'adresser à des fournisseurs européens. La qualité européenne est irréprochable, mais d'autres désillusions m'attendaient.

Une cuillère pour maman

Pour éviter les délais d'attente trop longs, je commande habituellement les circuits impri-

Photo : Yekaterina Netuk / Shutterstock.

més de mes prototypes auprès d'un fabricant européen. Bien que plus élevé, le prix reste encore acceptable pour des volumes plus grands, car le coût d'un CI ne représente qu'une petite partie du coût total d'un produit. À condition, bien sûr, d'éviter si possible les cartes multicouches et les masques de soudure de couleurs fantaisistes – mais le vert classique est un peu ennuyeux à la longue, et j'enviais secrètement les CI colorés de la concurrence.

La découverte des coûts liés à l'assemblage automatisé des CMS a vite douché mes espoirs. La seule fabrication de pochoirs pour l'imprimante de pâte à braser se chiffrait en centaines d'euros, sans compter les coûts de documentation, de configuration de la ligne d'assemblage des CMS, et ceux relatifs à l'inspection optique automatisée.

Mes volumes étaient vraiment insuffisants pour couvrir de tels coûts. Seule issue possible : l'assemblage manuel, avec en perspective de longues soirées solitaires passées dans une mansarde à chercher et saisir de façon mécanique des composants minuscules. Le plus fastidieux aura toutefois été l'application de la pâte à braser. Pour la faciliter, je prends en général une plaque de plexiglas de 3 mm d'épaisseur et y découpe une fenêtre de mêmes dimensions que celles du CI. Je superpose ensuite deux CI de façon à ce que celui du dessus soit de niveau avec le plexiglas. Je peux alors positionner le pochoir et le fixer avec du ruban adhésif. Pratique, mais arrive ensuite l'inévitable désastre. Quel que soit le soin mis dans l'application de la pâte à braser, elle finit toujours par atterrir sur la table et vos mains, voire, si votre bonne étoile a décidé d'assister à un autre spectacle ce jour-là, sur vos vêtements. Une fois mon travail achevé, j'ai l'impression d'être un de ces bébés qui vient de terminer un dessert au chocolat et en a mis plus sur lui et autour de lui que dans son ventre [1]. Alors je pousse un gros soupir et nettoie.

Conscience géopolitique

Aujourd'hui nos circuits imprimés viennent d'Extrême-Orient, essentiellement pour répondre à la pression économique exercée par nos clients. Et, oui, tout se passe bien. Peut-être la qualité en souffre-t-elle légèrement, mais elle reste assurément acceptable. Qui plus est nous pouvons choisir des couleurs fantaisistes sans surcoût excessif. Malgré les déboires passés, nous avons, ces dernières années, fait assembler en Chine certains de nos produits, dont l'horloge de sable et la carte d'extension *Swiss Pi* pour le Raspberry Pi [2]. Il faut dire que la qualité chinoise, malgré des prix restés bas, s'est grandement améliorée avec le temps. L'arrivée d'autres services, comme l'impression 3D et l'usinage numérique, rend également très tentant le passage de commandes dans cette partie du monde.

Néanmoins tout cela me gêne un peu. Car les prix mis à part, très peu de choses

sont démocratiques en Chine. La volonté d'expansion mondiale du gouvernement et les développements géopolitiques de ces dernières années rendent le paysage chinois un peu moins sympathique. Je vais donc encore attendre un peu avant de remettre définitivement mes pincettes. ◀

220031-04

Des questions, des commentaires ?

Contactez Elektor (redaction@elektor.fr).

Contributeurs

Texte : Ilse Joostens

Rédaction : Eric Bogers

Mise en page : Giel Dols

Traduction : Hervé Moreau

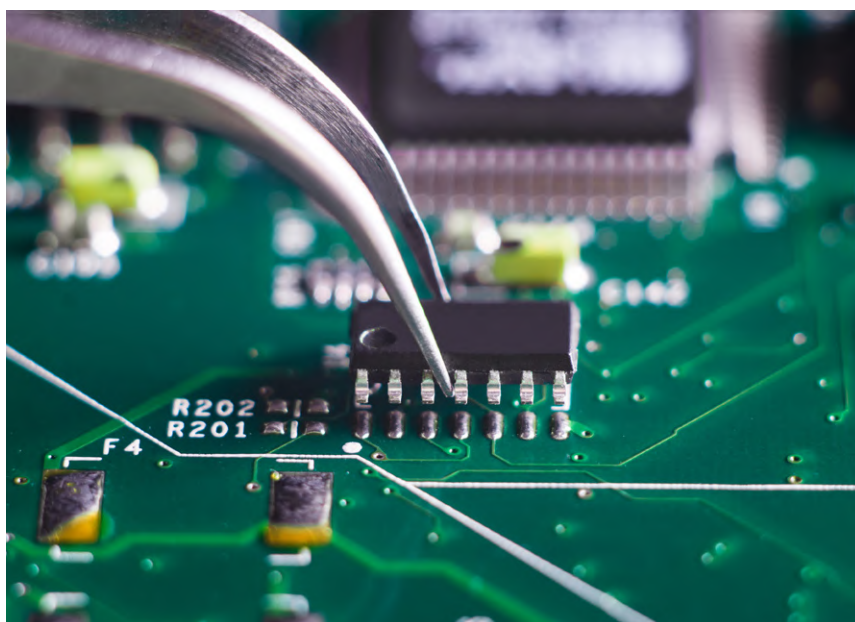


Photo: Oleg Shvydiuk / Shutterstock.

LIENS

[1] « Is cake smashing the dumbest cake trend ever? », Angès de Sucre/Angès Bakery : <https://bit.ly/3nyQeGD>

[2] « Swiss Pi, couteau suisse pour Raspberry Pi », P. S'heeren, Elektor 09-10/2016 : www.elektormagazine.fr/magazine/elektor-201609/39774

Infrastructure graphique WinUI pour les applications Windows

Application de démonstration

Veikko Krypczyk (Allemagne)

Les logiciels conçus pour le contrôle des applications d'électronique fonctionnent souvent sous un environnement Windows. L'utilisation d'une application de bureau installée localement donne un accès direct à toutes les interfaces système du PC. Ainsi, Microsoft rationalise actuellement son assistance technique aux développeurs utilisateurs d'infrastructures (ou frameworks) d'interface Windows. La nouvelle interface graphique WinUI 3 est une approche d'avenir. Dans cet article, nous allons aborder le contexte technique et montrer comment l'utiliser en créant une simple application de démonstration pour les électroniciens.

Dans de nombreux projets d'électronique, un PC Windows est indispensable pour assurer les fonctions de contrôle ainsi que l'enregistrement des données et d'autres tâches ; en général, il est nécessaire de développer une interface utilisateur graphique exécutable sous Windows pour remplir cette fonction. Les applications de bureau exécutées localement sont souvent la méthode à privilégier dans ce cas ; elles donnent un accès complet à l'environnement du système et procurent également des liens vers les périphériques connectés via les pilotes appropriés.

Développeur du système d'exploitation Windows, Microsoft connaît actuellement un bouleversement technologique majeur, qui concerne essentiellement la connectivité et la conception d'interfaces utilisateur. Les développements sont coordonnés sous les projets *WinUI 3* et *Windows App SDK*. Cet article décrit en quoi consiste cette nouvelle interface graphique et comment l'utiliser. Vous y trouverez d'abord un aperçu des possibilités de programmation d'applications Windows avec une interface utilisateur graphique. La finalité de *WinUI 3* sera précisée grâce à l'utilisation de différents types d'applications et technologies. Ne souhaitant pas nous contenter d'une analyse purement théorique de ce type d'application relativement nouveau, nous créerons également notre première application pratique.

Technologie

Pour l'essentiel, les applications actuelles dotées d'une interface utilisateur graphique pour le système d'exploitation Windows se répartissent en deux catégories. D'une part, nous avons les applications de bureau (*Desktop Applications*). Elles sont essentiellement basées sur l'utilisation de l'*API Win32* et leur développement s'appuie sur différentes approches, infrastructures et langages de programmation. Microsoft a proposé les technologies Windows Forms (WinForms) et Windows

Presentation Foundation (WPF). WinForms est basé sur l'interface Windows GDI. WPF est basé sur DirectX et était à l'origine destiné à remplacer WinForms. Les deux infrastructures graphiques étaient basées sur l'infrastructure .NET, destinée aux programmes Windows. Son développement s'est arrêté à la version 4.8.

La version .NET 5, quant à elle, est le successeur technologique de .NET Core. Cette infrastructure ne se limite pas à Windows, et peut également être utilisée avec d'autres systèmes d'exploitation. De manière inattendue, Microsoft a transformé WinForms et WPF en .NET Core. Si vous optez aujourd'hui pour WinForms ou WPF en développant une application pour Windows, vous avez le choix entre l'ancien environnement .NET et .NET Core. Il est également possible d'effectuer une migration des applications existantes, mais, comme c'est souvent le cas pour ce genre de projets, cette opération s'accompagne d'un certain nombre de problèmes. Les autres outils de développement de différents fabricants sont le plus souvent basés sur l'interface graphique du système d'exploitation (GDI) et encapsulés dans leur propre infrastructure.

La deuxième catégorie d'applications Windows concerne la plateforme d'application universelle Windows (*UWP*). Ces applications s'exécutent au sein d'une couche isolée du système d'exploitation et n'ont qu'un accès limité au système. Les utilisateurs les installent via le magasin d'applications. Cependant, ce type d'application n'a pas eu le succès escompté et leur adoption a été assez réduite. L'une des causes est l'accès très limité au système. La plateforme *UWP* a toutefois pour avantage que l'infrastructure graphique WinUI 2 utilisée ici est considérablement plus moderne que les technologies WinForms et WPF. Ses points forts résident dans un design attrayant, de nouveaux composants visuels, l'utilisation de matériaux et l'orientation vers le langage de conception *Fluent Design System*. En d'autres termes, les applications *UWP* ont un aspect moderne, contemporain et inédit, mais

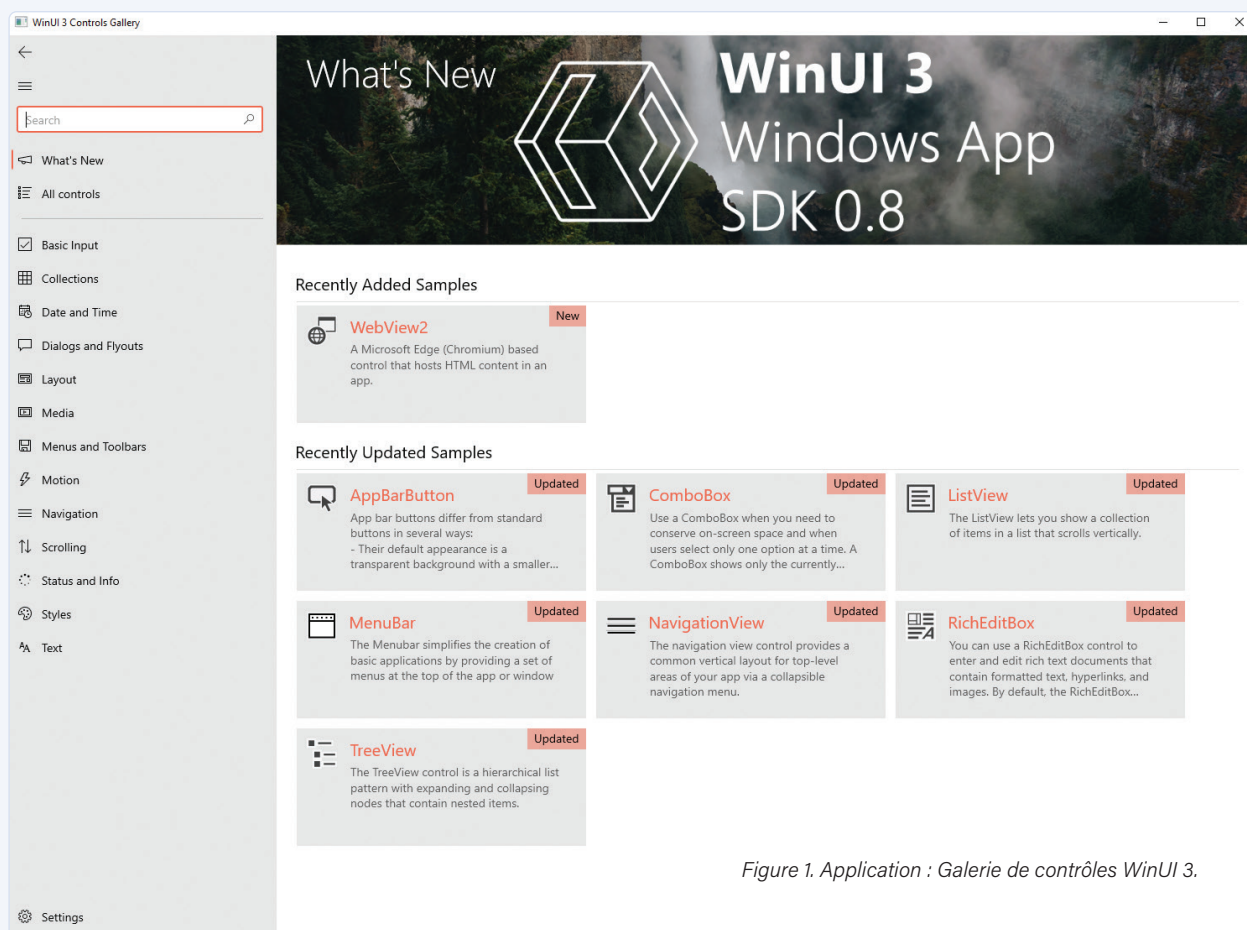


Figure 1. Application : Galerie de contrôles WinUI 3.

leur convivialité est assez limitée. Pour obtenir des effets similaires avec les technologies WinForms ou WPF, nous devons pousser les capacités dans leurs retranchements, utiliser de nombreux composants tiers ou « combiner » les technologies WinForms/WPF avec la plateforme UWP. Cette approche conduit toutefois rapidement à une structure d'application plus complexe et présente les inconvénients habituels, notamment une plus grande vulnérabilité aux erreurs et une plus faible maintenabilité.

Les logiciels conçus pour la commande de projet électronique, le développement, etc., sont presque tous des applications classiques de bureau. Vous pouvez également les construire en utilisant d'autres outils et cadres. Pour le langage de programmation Java, il existe, par exemple, le cadre graphique Swing qui est basé en interne sur l'interface du système d'exploitation GDI sous Windows.

Les logiciels conçus pour la commande de projets électroniques, le développement, etc., sont presque tous des applications classiques de bureau. Vous pouvez également les construire en utilisant d'autres outils et infrastructures. Pour le langage de programmation Java, il existe, par exemple, l'infrastructure graphique Swing, basée en interne sur l'interface de système d'exploitation GDI de Windows.

Infrastructure graphique WinUI 3

Avec l'introduction de WinUI 3, Microsoft souhaite permettre à toutes les applications fonctionnant sous Windows d'utiliser une interface graphique moderne. WinUI 3 est le successeur de WinUI 2 [1]. Cependant, elle est disponible pour tous les types d'applications Windows et n'est donc pas uniquement utilisée dans celles fonctionnant avec la plateforme UWP. WinUI 3 fait partie du nouveau SDK (kit de développement) Windows APP, proposé parallèlement à l'introduction de Windows 11. Ce SDK offre de nouvelles fonctionnalités pour le

développement d'applications Windows. Il n'est pas seulement destiné à Windows 11, mais peut être utilisé aussi sous les versions actuelles de Windows 10. Le développement du SDK Windows App est toujours en cours, mais une première version est disponible et peut déjà être utilisée dans les nouvelles applications.

L'infrastructure WinUI 3 est techniquement et conceptuellement basée sur WinUI 2. Si vous avez déjà développé une application pour la plateforme UWP, vous vous familiariserez rapidement avec son utilisation. Elle est fondée sur les principes suivants :

- **Séparation entre le code et la conception** : l'interface utilisateur est créée de manière déclarative dans des fichiers séparés à l'aide du langage XAML, basé sur XML.
- **Éléments de contrôle de l'interface utilisateur** : il existe un ensemble de contrôles pour concevoir l'interface utilisateur. Il s'agit d'éléments de base, comme des boutons et des champs de saisie de texte, mais aussi des éléments plus complexes et avancés comme un élément de contrôle de calendrier, un composant *WebView* ou un élément d'affichage de données personnelles, que nous pouvons utiliser, par exemple, pour l'administration des utilisateurs. Si vous souhaitez explorer davantage toute la gamme des éléments de contrôle, accédez au Microsoft Store et téléchargez l'application *WinUI 3 Controls Gallery*, qui donne un aperçu des contrôles disponibles pour WinUI 3, avec une démonstration de leur utilisation et de leur intégration dans le code source (XAML). Des liens correspondants vers la documentation sont également disponibles (figure 1).
- **Couplage souple par liaison de données** : les propriétés et les événements des éléments de contrôle sont liés au code source par liaison de données. Les données sont ainsi échangées dans

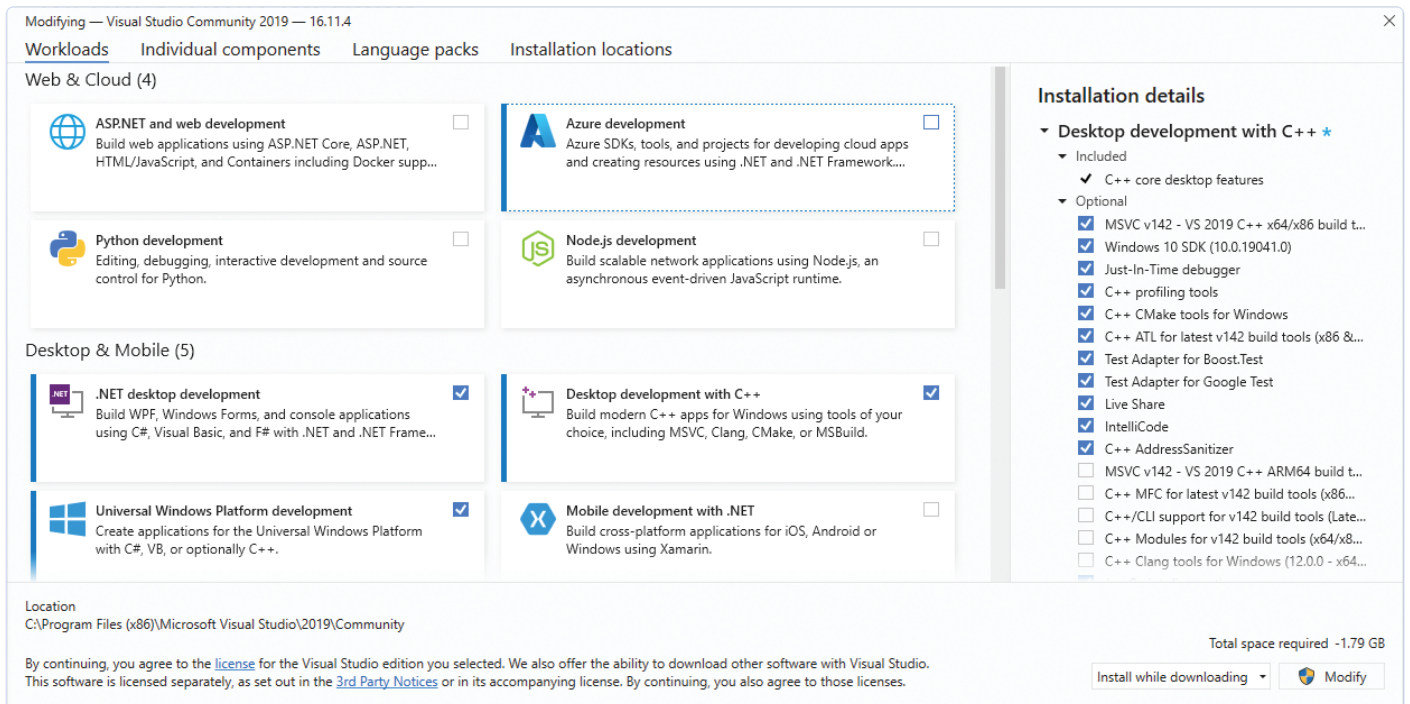


Figure 2. Sélectionnez les Workloads nécessaires pour Visual Studio.

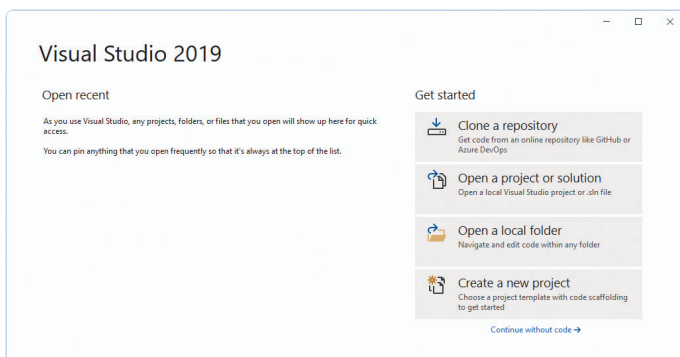


Figure 3. Premier démarrage de Visual Studio.

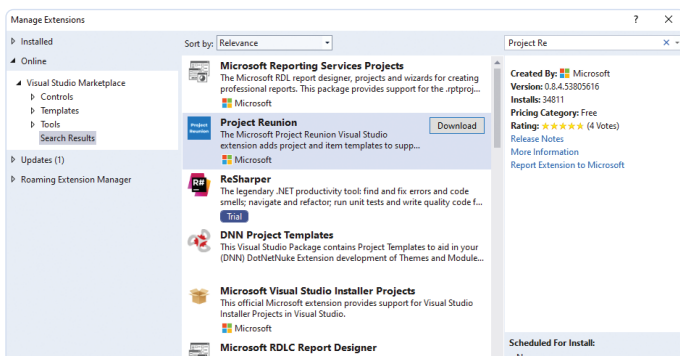


Figure 4. Installation de Windows SDK App (Projet « Reunion »).

les deux directions entre le code du programme et le contrôle de l'interface utilisateur. Les événements des éléments de contrôle, par exemple le clic sur un bouton, sont également transmis à l'algorithme correspondant, de la même manière.

- **Design moderne** : l'infrastructure WinUI 3 offre un aspect contemporain. Elle inclut le langage de conception *Fluent Design System* de Microsoft, ainsi que le matériau *Mica*, intégré dans Windows 11. Le langage *Fluent Design System* fournit les éléments d'interface utilisateur suivants : utilisation raisonnée de la géométrie et de la couleur, superposition des surfaces, utilisation de matériaux sélectionnés et application d'une iconographie et d'une typographie spécifiques pour la conception visuelle à l'aide d'images, de symboles et de polices. Les mouvements entre les éléments de l'interface utilisateur sont aussi pris en charge.

Examinons la procédure de développement d'applications avec l'infrastructure *WinUI 3*. Nous devons tout d'abord configurer l'environnement de développement.

Environnement de développement et configuration

La version actuelle de Visual Studio 2019 [2] est l'environnement de développement que nous utiliserons. L'édition Community sera suffisante pour nos besoins. Au moment où vous lirez cet article, une version stable de Visual Studio 2022 sera peut-être déjà disponible, auquel cas vous devrez l'utiliser. Il est conseillé d'installer au préalable les dernières mises à jour du système d'exploitation. Lors de l'installation de Visual Studio, il vous sera demandé de sélectionner les packages d'installation. Vous pouvez également, par la suite, appeler à tout moment le programme d'installation de Visual Studio via le

menu de démarrage. Sélectionnez maintenant les packages d'installation suivants (Workloads) : *.NET desktop development*, *Desktop development with C++* et *Universal Windows Platform development* (**figure 2**). Après l'installation, démarrez Visual Studio et dans l'écran de démarrage, sélectionnez l'option *Continue without code* -> (continuer sans code) (**figure 3**).

Nous devons maintenant installer le modèle pour le développement avec WinUI 3. Pour ce faire, dans Visual Studio, choisissez l'option de menu *Extensions | Manage Extensions* (Extensions | Gérer les extensions). Recherchez *Project Reunion* (le nom de développement du nouveau *SDK Windows App*) et installez la version actuelle (**figure 4**). De même, installez l'extension *Windows Template Studio*. Celle-ci offre des modèles avancés pour la création d'une nouvelle application. *Visual Studio* doit être redémarré après le téléchargement des extensions ; l'installation se fera alors automatiquement.

Application pour WinUI 3

Commençons par créer un nouveau projet. Ici, comme l'indique la **figure 5**, nous sélectionnons le modèle *App (WinUI 3 in Desktop)* (*Windows Template Studio*). Dans *Windows Template Studio* (**figure 6**), nous pouvons configurer le projet :

- **Project type** : on spécifie ici le type de navigation, par exemple avec une barre de menu ou une barre de navigation latérale (menu *Hamburger*).
- **Design pattern** : installation et configuration directe de la boîte à outils *MVVM*. Celle-ci associe les éléments de l'interface utilisateur (définis dans *XAML*) à la logique du programme (langage de programmation *C#*).
- **Pages** : nous pouvons ajouter un certain nombre de pages au projet. Nous pouvons choisir parmi différents modèles, par exemple une page pour entrer les paramètres du programme.
- **Features** : ici, nous pouvons sélectionner des exemples de thèmes ou enregistrer les paramètres du programme.

En cliquant sur le bouton *Create*, nous générons l'application de bureau que *WinUI 3* utilise. *Windows Template Studio* génère un dossier avec trois projets :

- **App** : il contient le code source de l'application de bureau. Dans le sous-dossier *View*, par exemple, vous trouverez les fichiers *XAML* pour les pages qui ont été créées par *Windows Template Studio*. La logique du programme est stockée dans les fichiers du dossier de programme *ViewModel*.
- **Package** : le projet est chargé de fournir l'application de bureau. Actuellement, les applications *WinUI 3* sont installées sur l'ordinateur cible sous la forme d'un paquet d'applications. Ce format a jusqu'à présent été utilisé pour les applications *UWP*. Les paquets générés peuvent également être distribués en utilisant *Store*. Les futures versions du *SDK Windows App* devraient également permettre l'installation sans package d'application.
- **Core** : ce projet contient l'ensemble des services et des classes qui fournissent des services à l'application. Il n'est pas obligatoire et peut être exclu.

Démarrez l'application directement à partir de Visual Studio en utilisant la flèche verte de la barre d'outils. Félicitations - vous avez créé

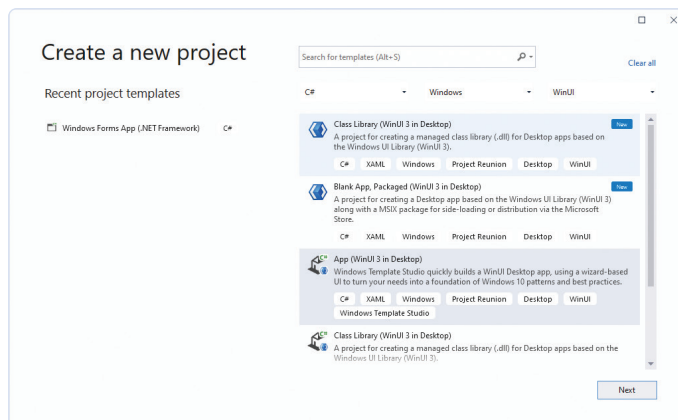


Figure 5. Modèle de projet WinUI 3 Desktop.

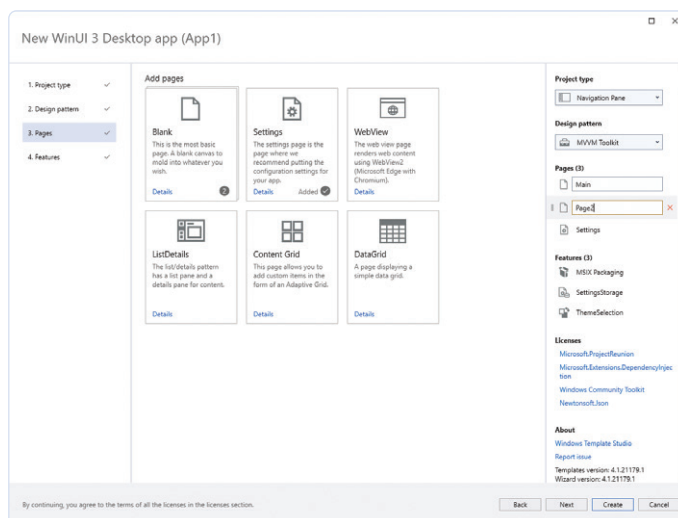


Figure 6. Windows Template Studio.

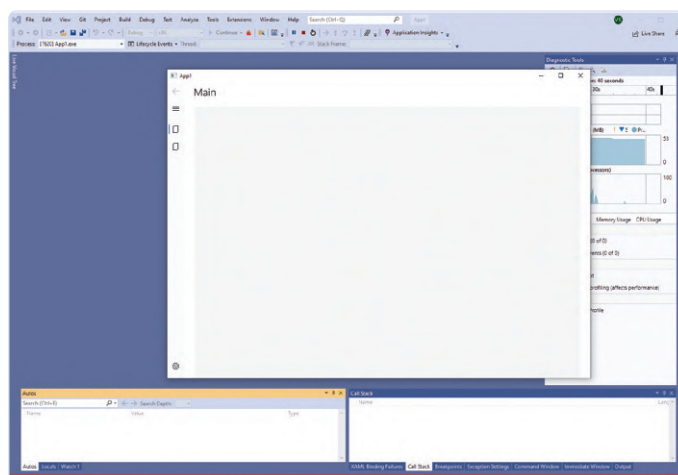


Figure 7. Une première application de bureau avec WinUI 3.

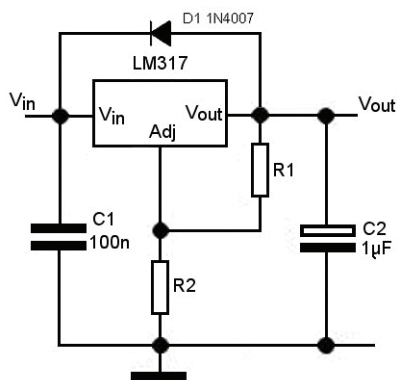


Figure 8. Schéma du circuit du régulateur de tension LM317.

vosre première application avec *WinUI 3* (figure 7). Je dois insister à nouveau sur le fait qu'il s'agit d'une application de bureau avec un accès complet au système. Comme nous l'avons déjà mentionné, c'est un aspect important pour les logiciels destinés à contrôler des composants électroniques externes, par exemple. Il est possible de créer une barre de navigation latérale, des pages initiales et d'adapter la conception de l'application. Vous disposez de toutes les options d'accès au système, y compris la communication avec les bibliothèques et les pilotes du système. Nous pouvons maintenant commencer à expérimenter la conception de l'interface utilisateur.

Application de démonstration

La meilleure façon de se familiariser avec un nouveau système est de le tester. Nous allons concevoir une interface utilisateur simple pour notre première application (le code source de cet exemple se trouve sur la page Web de l'article [3]). Le point de départ est le fichier *XAML* de la page concernée. À titre d'expérience, nous pouvons créer un outil de calcul pratique utilisable avec le régulateur de tension linéaire ajustable

Tableau 1 : Éléments de contrôle

Éléments de contrôle	Propriété	Valeur	Description
TextBoxR1	Width	200	Textbox width
	Margin	20, 20, 0, 0	Margin size: left, top, right, bottom
	HorizontalAlignment	Left	Left justified
	Header	R1:	Caption
	Text	x:Bind ViewModel.R1	Binding the property to the variable R1 in C#.
TextBoxU (out)	Width	200	Textbox width
	Margin	20, 10	Margin size: left, top, right, bottom
	HorizontalAlignment	Left	Left justified
	Header	U (out):	Caption
	Text	x:Bind ViewModel.UOut	Binding the property to the variable UOut in C#.
TextBoxR2	Width	200	Textbox width
	Background	LightGray	Background colour
	Margin	20, 10	Margin size: left, top, right, bottom
	HorizontalAlignment	Left	Left justified
	Header	R2:	Caption
	IsReadOnly	True	Read only protection
Button	Text	x:Bind ViewModel.R2	Binding the property to the variable R2 in C#.
	Width	200	Button area width
	Margin	20, 10	Margin size: left, top, right, bottom
	Background	LightGreen	Background colour
	Command	x:Bind ViewModel.CalcCommand	Binding to the CalcCommand Method in C#.
	Content	Calc	Caption
	FontWeight	Bold	Font properties

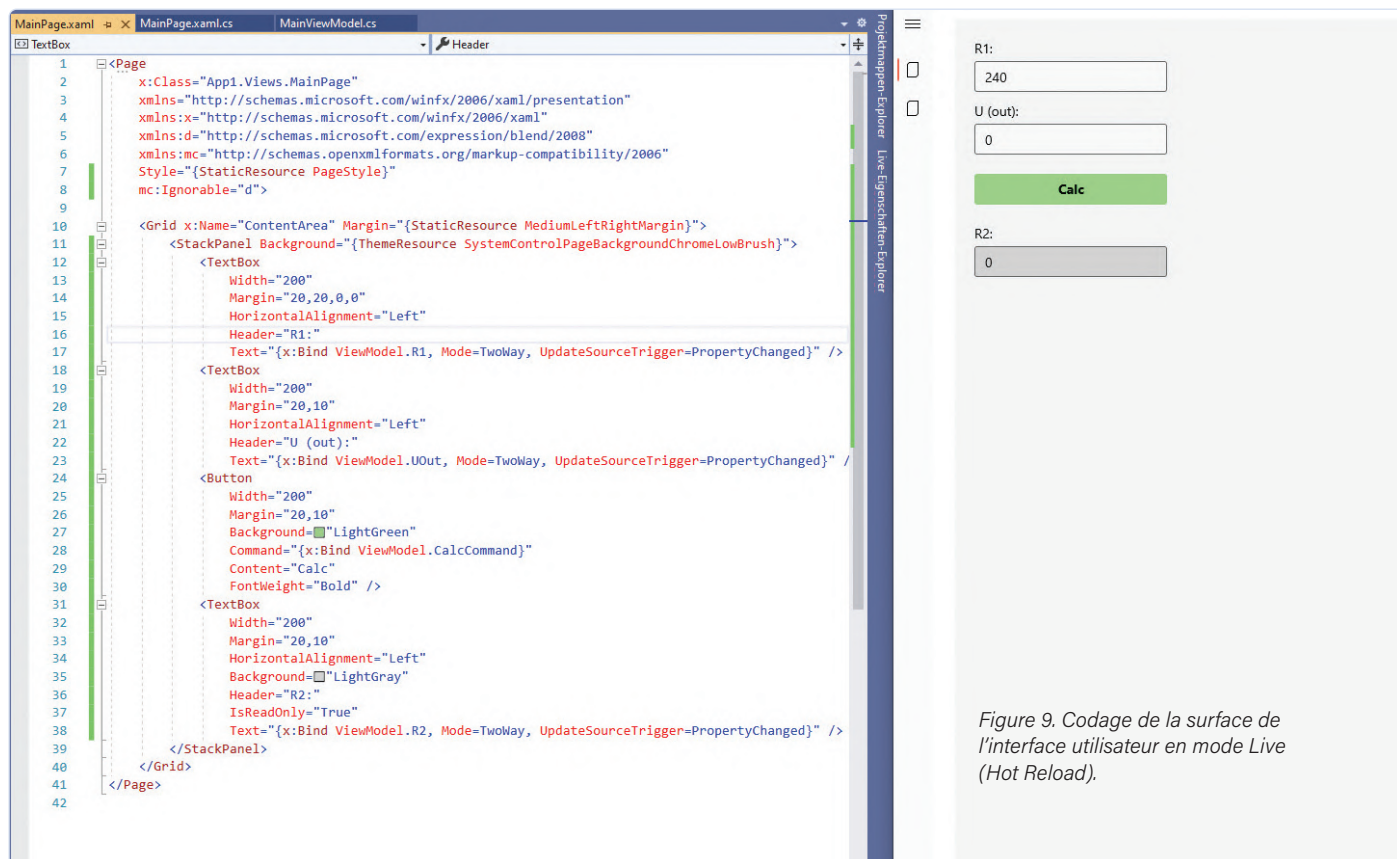


Figure 9. Codage de la surface de l'interface utilisateur en mode Live (Hot Reload).

LM317 (**figure 8**). La tension de sortie de ce dispositif est donnée par la formule $V_{out} = 1,25 (1 + R2/R1)$. Nous pouvons résoudre cette équation pour $R2$ et ainsi calculer sa valeur pour donner la tension de sortie souhaitée. À l'aide de cet exemple, nous pouvons démontrer la marche à suivre pour programmer des applications avec *WinUI 3*. Cette procédure comprendra les étapes suivantes :

- Définir l'interface utilisateur en XAML.
- Coder la logique du programme en C#.
- Lier l'interface utilisateur à la logique du programme.

- Transférer l'interaction de l'utilisateur entre l'interface utilisateur et la logique du programme.
- Générer les données vers le formulaire.

Commençons par définir la surface. Nous avons besoin de deux champs de texte pour enregistrer les valeurs de $R1$ et V_{out} , ainsi que d'un champ de texte pour la valeur de $R2$. Il faudra un bouton (*Button*) pour lancer le calcul. Nous utilisons des contrôles *TextBox* pour l'entrée et la sortie. Tous les éléments doivent être disposés sous la forme d'une pile verticale les uns au-dessus des autres, ils sont donc insérés dans un



Listage 1. Définition des interfaces utilisateur

```
<Page
  x:Class="App1.Views.MainPage"
  ...>

  <Grid x:Name="ContentArea" Margin="">
    <StackPanel Background="">
      <TextBox
        Width="200"
        Margin="20,20,0,0"
        HorizontalAlignment="Left"
        Header="R1:"
        Text="" />
      <TextBox
        Width="200"
        Margin="20,10"
        HorizontalAlignment="Left"
        Header="U (out):"
        Text="" />
    </StackPanel>
  </Grid>
</Page>
```

```
<Button
  Width="200"
  Margin="20,10"
  Background="LightGreen"
  Command=""
  Content="Calc"
  FontWeight="Bold" />
<TextBox
  Width="200"
  Margin="20,10"
  HorizontalAlignment="Left"
  Background="LightGray"
  Header="R2:"
  IsReadOnly="True"
  Text="" />
</StackPanel>
</Grid>
</Page>
```

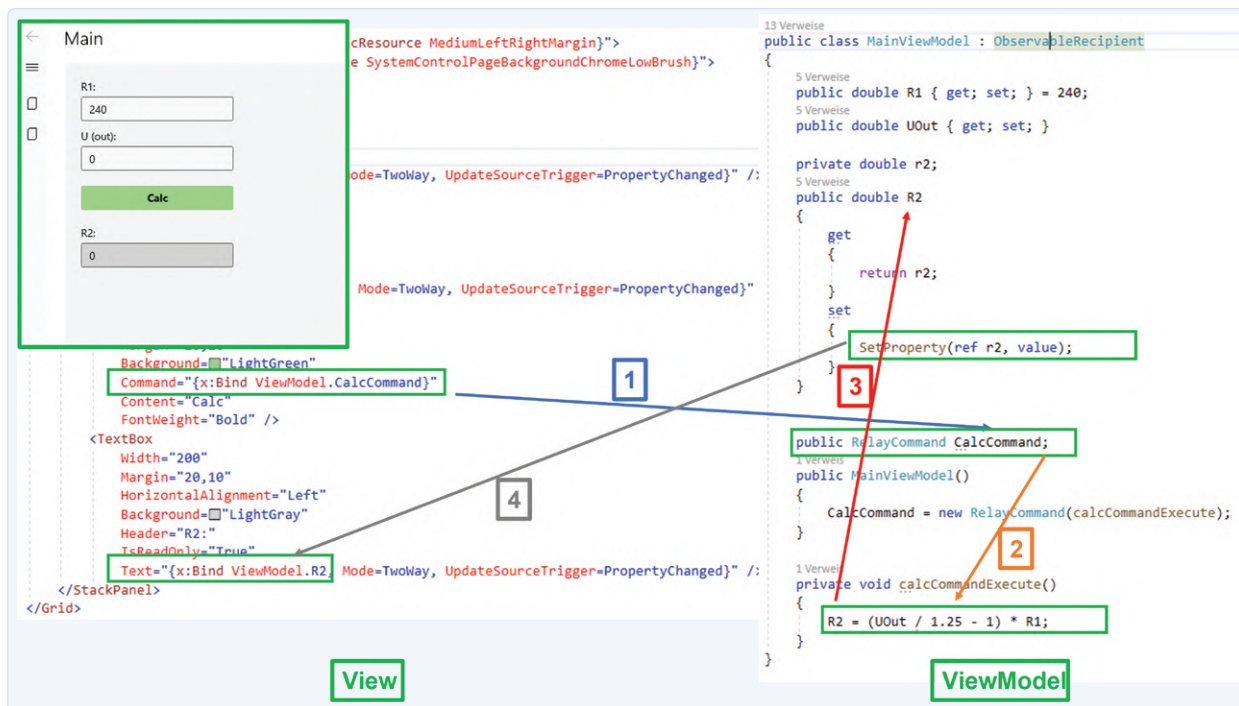


Figure 10. Exemple de relation entre View et ViewModel.

conteneur de mise en page de type `<StackPanel />`. Tous les éléments seront disposés les uns au-dessus des autres avec `StackPanel` sans nécessiter aucune autre configuration. Les contrôles sont configurés via le code `XAML`, avec les propriétés selon le **tableau 1**.

Le code source associé est présenté dans le **listage 1**. Vous pouvez coder la surface de manière interactive. Démarrez l'application et placez le fichier `XAML` correspondant dans Visual Studio et l'application en parallèle sur l'écran (**figure 9**). Les modifications apportées au code `XAML` sont immédiatement reprises au démarrage de l'application, sans enregistrement, et produisent un affichage actualisé. Cette fonctionnalité, appelée *Hot Reload*, est standard dans la création d'interfaces utilisateur graphiques.

Ce qui est intéressant ici est la liaison des éléments de contrôle de type `TextBox` avec les propriétés de `Text`. Vous trouverez ici une expression conforme au modèle dans le code `XAML` :

```
Text="{x:Bind ViewModel.R1, Mode=TwoWay,
UpdateSourceTrigger=PropertyChanged}"
```

Cela signifie que la propriété `Text` est liée à la variable `R1`. Cette liaison, basée sur le concept MVVM, est définie dans la page `ViewModel`. Les événements de l'interface utilisateur sont traités dans la couche `View` et les données sont gérées dans la couche `Model`. `ViewModel` représente la connexion entre les deux couches. Grâce au concept MVVM,



Listage 2. Logique de programme pour le calcul en C#

```
using CommunityToolkit.Mvvm.ComponentModel;
using CommunityToolkit.Mvvm.Input;

namespace App1.ViewModels
{
    public class MainViewModel : ObservableRecipient
    {
        public double R1 { get; set; } = 240;
        public double UOut { get; set; }

        private double r2;
        public double R2
        {
            get
            {
                return r2;
            }
            set
            {
                r2 = value;
            }
        }

        public RelayCommand CalcCommand;

        private void CalcCommandExecute()
        {
            R2 = (UOut / 1.25 - 1) * R1;
        }
    }
}
```

```
{
    SetProperty(ref r2, value);
}

public RelayCommand CalcCommand;
public MainViewModel()
{
    CalcCommand = new RelayCommand(CalcCommandExecute);
}

private void CalcCommandExecute()
{
    R2 = (UOut / 1.25 - 1) * R1;
}
}
```


toutes les couches sont découplées et peuvent être développées et maintenues indépendamment les unes des autres. Vous trouverez des informations sur le modèle MVVM dans le document [4].

Logique du programme

La logique du programme est mise en œuvre en langage C# (listage 2). À cette fin, un fichier de programme (*ViewModel*) est attribué à chaque fenêtre de l'interface utilisateur (*View*). Dans notre exemple, il s'agit du fichier *MainViewModel* attribué à la vue *MainPage*. Quelques remarques concernant le code :

- **Importation de bibliothèques** : elle se fait par l'instruction *uses*. Dans notre cas, nous avons besoin de deux bibliothèques pour le modèle MVVM.
- **Définition des propriétés** : celles-ci doivent être publiques, car nous accédons aux propriétés depuis l'extérieur, dans ce cas depuis *View*.
- **Mise à jour automatique de l'interface utilisateur** : la classe *MainViewModel* est dérivée de la classe de base *ObservableRecipient*, générée par l'assistant lorsque le projet a été créé. Cette classe implémente à son tour l'événement *OnPropertyChanged*. Cela garantit que lorsque la valeur d'une propriété change, tous les éléments liés sont avertis de cette modification. Dans notre cas, c'est la propriété *R2* qui nous intéresse. La valeur de *R2* est calculée dans le code du programme. Le code *Setter* de la propriété est appelé et l'événement *OnPropertyChanged* décrit ci-dessus est déclenché par la méthode *SetProperty (...)*. La propriété *Text* de *TextBox* *R2* est liée à *R2* (dans *ViewModel*). Si *R2* est modifié, la valeur affichée est automatiquement mise à jour dans la *TextBox* associée. Cela fonctionne grâce à la liaison de données.
- **Transmettre les actions de l'utilisateur au moyen de commandes** : si l'utilisateur appuie sur le bouton, il déclenche une commande. La méthode de calcul est liée à celle-ci. Ici aussi, l'interface utilisateur et le code de programme ne sont reliés entre eux que par la liaison de données.
- **Attribution de View et ViewModel** : le code du programme (fichier : *MainViewModel.cs*) est attribué à l'interface utilisateur (fichier : *MainPage.xaml*). Cela se fait dans le fichier *code-behind* de la page (fichier : *MainPage.xaml.cs*). Vous pouvez le constater en examinant le code source.
- **Calcul** : le calcul de la valeur de *R2* se fait par la méthode *calcCommandExecute (...)* selon la formule ci-dessus, qui est ensuite affectée à *R2*.

L'interface utilisateur est donc liée de manière « souple » au code du programme au moyen de la liaison de données. Les connexions à la liaison de données qui viennent d'être décrites sont visualisées

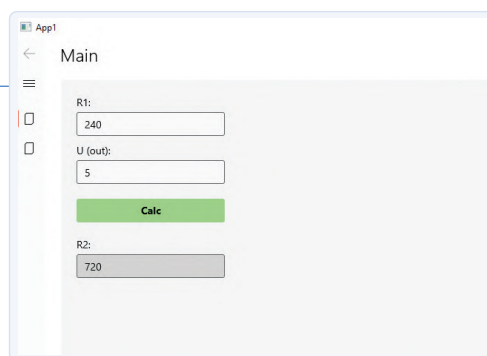


Figure 11. L'exemple complet de l'application.

à l'aide de l'exemple de la figure 10. Lancez l'application et testez-la. La valeur de la deuxième résistance *R2* est calculée après avoir saisi les valeurs de *R1* et de *V_{Out}* (figure 11).

Nous avons ainsi décrit le modèle de développement de base des applications de bureau avec l'infrastructure graphique WinUI 3. Dans cette perspective, elle deviendra un nouveau standard sous Windows et pourra également être utilisée par d'autres environnements et langages de développement. La variété des options graphiques permettant de créer des applications modernes est impressionnante.

Conclusions et perspectives

Vous avez le choix entre plusieurs technologies pour créer une application Windows. La tendance – avec Windows 11 en perspective – est à l'utilisation de *WinUI 3*, qui vous permettra de créer une interface utilisateur attrayante et moderne. De ce point de vue, il est intéressant de considérer *WinUI 3* pour développer une nouvelle application Windows et d'envisager les options de migration pour une application existante. Vous obtiendrez ainsi une interface contemporaine avec une bonne expérience utilisateur. Il n'y a pas non plus de limitations en termes d'accès au système comme c'est le cas avec le modèle d'application UWP. ❏

210407-04

Contributeurs

Texte et images : Veikko Krypczyk

Rédaction : Jens Nickel

Traduction : Asma Adhimi

Mise en page : Giel Dols

Des questions, des commentaires ?

Contactez Elektor (redaction@elektor.fr).

LIENS

[1] Informations sur WinUI 3: <https://docs.microsoft.com/en-us/windows/apps/winui/>

[2] Visual Studio 2019: <https://visualstudio.microsoft.com/>

[3] Page du projet pour cet article: www.elektormagazine.fr/210407-04

[4] Informations sur le modèle MVVM: https://fr.wikipedia.org/wiki/Mod%C3%A8le-vue-vue_mod%C3%A8le

Création d'interfaces graphiques en Python avec guizero

La pire des interfaces

Apprenez les bonnes pratiques de conception en faisant tout de travers !



Laura Sach

Laura dirige l'équipe A Level de la Fondation Raspberry Pi chargée des ressources pédagogiques en informatique à destination des étudiants.

@CodeBoom



Martin O'Hanlon

Martin crée des cours, des projets et des ressources en ligne au sein de l'équipe Learning de la Fondation Raspberry Pi.

@martinohanlon

Grisé par votre nouveau savoir, vous pourriez être tenté de vous lancer tête baissée dans la création d'interfaces graphiques en mélangeant frénétiquement widgets, couleurs, polices et fonctions de votre cru. Vous apprendriez de vos erreurs, mais cela vous prendrait du temps. Alors gagnons-en ensemble en découvrant dès maintenant tout ce qu'il ne faut pas faire !

Mais que fait donc la police ?

Un mauvais contraste entre l'arrière-plan et la police rend tout texte difficilement lisible. Autrement dit il ne faut jamais utiliser deux couleurs de teintes trop proches.

Importez ces widgets en début de code :

```
from guizero import App, Text
```

Créez une fenêtre **app** et un texte :

```
app = App("it's all gone wrong")
title = Text(app, text="Some hard to read text")

app.display()
```

Jouez avec les couleurs, la police et la taille du texte (listage **worst1.py**). Nos choix ne sont *visiblement* pas les bons !

```
app = App("it's all gone wrong", bg="dark green")
title = Text(app, text="Some hard-to-read text", size="14", font="Comic Sans", color="green")
```

Pour être bien lisible, il va de soi qu'un texte doit rester affiché suffisamment longtemps à l'écran et ne pas se prendre pour un clignotant.

Tous les widgets de *guizero* peuvent être rendus invisibles (ou de nouveau visibles) avec les fonctions

hide() et **show()**. La fonction **repeat()** permet quant à elle d'exécuter une fonction toutes les *n* secondes. Combinons ces trois fonctions pour faire clignoter le texte.

La fonction **flash_text()** masque le texte s'il est visible, et le montre s'il ne l'est pas :

```
def flash_text():
    if title.visible:
        title.hide()
    else:
        title.show()
```

Utilisons ensuite **repeat()** pour l'exécuter toutes les 1000 millisecondes (1 s) :

```
app.repeat(1000, flash_text)

app.display()
```

Lancez votre code (**worst2.py**) : le texte est masqué et réaffiché toutes les secondes.

Mauvais choix de widgets

La frontière entre une bonne et une mauvaise interface tient parfois à un seul widget.

Lequel utiliser pour la saisie d'une date ? Un



Figure 2

► **Figure 2** Sélection de lettres par listes déroulantes.

TextBox serait flexible, mais exigerait validation et formatage de la date. Trois listes déroulantes (widgets Combo) pour le jour, le mois et l'année éviteraient l'étape de validation, mais seraient d'utilisation fastidieuse.

Pour la saisie d'une date et d'une heure, pourquoi pas une glissière (fig. 1) ? Le widget **Slider** (code **worst3.py**) retourne un nombre compris entre 0 et 999.999.999, soit le nombre de secondes écoulées depuis le 1^{er} janvier 1970, puis le convertit en date et en heure avec la fonction `ctime()`. Essayez !

User d'un TextBox simple ou multilignes est une façon simple de demander du texte à l'utilisateur. Mais n'est-ce pas justement *trop* simple ? N'exige-t-on pas trop de saisie de la part de l'utilisateur ? Et quid de celui qui ne voudra pas lâcher sa souris ? Et si nous lui proposons de choisir les lettres à entrer au moyen de listes déroulantes (fig. 2) ? Testons cette idée géniale. Nous aurons besoin de la constante de chaîne `ascii_letters` :

```
from guizero import App, Combo
from string import ascii_letters
```

La constante `ascii_letters` est une liste contenant les lettres de l'alphabet en minuscules et en majuscules. Fournissons-la au paramètre **options** d'un widget **Combo** :

```
a_letter = Combo(app, options=" " + ascii_letters, align="left")

app.display()
```

Lancez le code (**worst4.py**) : la liste déroulante est alignée à gauche de la fenêtre et permet d'afficher une lettre suivie d'une espace. Pour que notre fanatique de souris puisse p. ex. entrer un mot de trois lettres, créons trois widgets **Combo** :

```
a_letter = Combo(app, options=" " + ascii_letters, align="left")
b_letter = Combo(app, options=" " + ascii_letters, align="left")
c_letter = Combo(app, options=" " + ascii_letters, align="left")
```

Le paramètre `align="left"` les place les uns contre les autres en partant du bord gauche.

Plutôt que de définir explicitement ces widgets un par un, le code **worst5.py** utilise une boucle **for**. Une boucle **for** apporte plus de souplesse au code, car il

worst1.py

Langage : Python 3

TÉLÉCHARGEZ
LE CODE COMPLET :

 magpi.cc/guizero/code

```
001. # Imports -----
002.
003. from guizero import App, Text
004.
005.
006. # App -----
007.
008. app = App("it's all gone wrong", bg="dark green")
009.
010. title = Text(app, text="Hard to read", size="14", font="Comic
011. Sans", color="green")
012.
013. app.display()
```

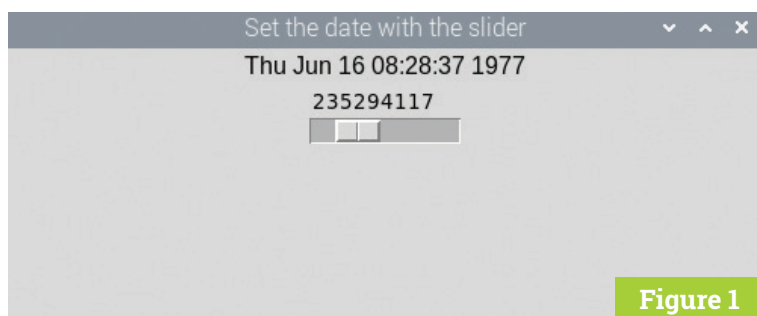


Figure 1

▲ Figure 1 Sélection (impossible) de la date et de l'heure.

worst2.py

Langage : Python 3

```
001. # Imports -----
002.
003. from guizero import App, Text
004.
005.
006. # Functions -----
007.
008. def flash_text():
009.     if title.visible:
010.         title.hide()
011.     else:
012.         title.show()
013.
014.
015. # App -----
016.
017. app = App("it's all gone wrong", bg="dark green")
018.
019. title = Text(app, text="Hard to read", size="14", font="Comic
020. Sans", color="green")
021.
022. app.repeat(1000, flash_text)
023.
024. app.display()
```


worst3.py

► Langage : Python 3

```
001. # Imports -----
002.
003. from guizero import App, Slider, Text
004. from time import ctime
005.
006.
007. # Functions -----
008.
009. def update_date():
010.     the_date.value = ctime(date_slider.value)
011.
012.
013. # App -----
014.
015. app = App("Set the date with the slider")
016. the_date = Text(app)
017. date_slider = Slider(app, start=0, end=999999999,
018.     command=update_date)
019.
020. app.display()
```

worst4.py

► Langage : Python 3

```
001. # Imports -----
002. from guizero import App, Combo
003. from string import ascii_letters
004.
005.
006. # App -----
007.
008. app = App("Enter your name")
009.
010. a_letter = Combo(app, options=" " + ascii_letters, align="left")
011.
012. app.display()
```

► **Figure 3**
Une pop-up aussi inutile qu'agaçante.

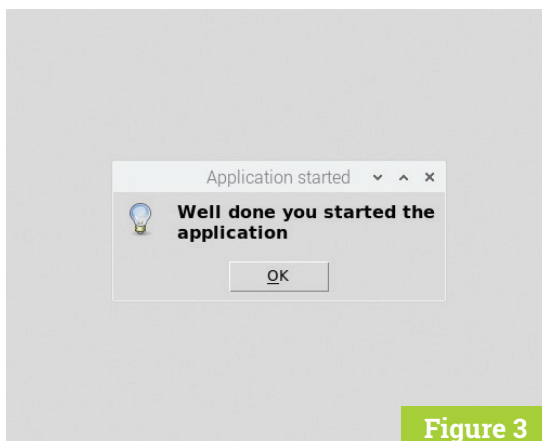


Figure 3

Widget Window

Les pop-ups permettent de questionner l'utilisateur, mais sont d'emploi limité. Pour afficher plus d'informations ou acquérir des données supplémentaires, vous pouvez créer plusieurs fenêtres à l'aide du widget **Window**.

Un objet (une fenêtre) **Window** s'utilise comme l'objet **App** et partage avec lui bon nombre de ses méthodes (fonctions).

```
from guizero import App, Window
```

```
app = App("Main window")
window = Window(app, "2nd Window")
```

```
app.display()
```

Un objet **Window** se masque avec **hide()** et se réaffiche avec **show()** :

```
window.show()
window.hide()
```

Pour qu'un objet **Window** devienne la fenêtre principale tant qu'il n'a pas été fermé, utilisez le paramètre **wait** de la méthode **show** et mettez-le sur **True** :

```
window.show(wait=True)
```

Vous trouverez plus d'informations sur l'utilisation des fenêtres multiples à la page lawsie.github.io/guizero/multiple_windows.

suffit de modifier la valeur de **range()** pour afficher plus ou moins de widgets.

Widgets pop-up

La bibliothèque **guizero** offre plusieurs types de pop-ups. Ces fenêtres surgissantes servent à interrompre l'utilisateur pour lui poser une question ou lui fournir une information, mais attention à leur abus !

Un bon moyen d'irriter votre prochain serait p. ex. de l'informer qu'il vient de lancer votre application (**fig. 3**) :

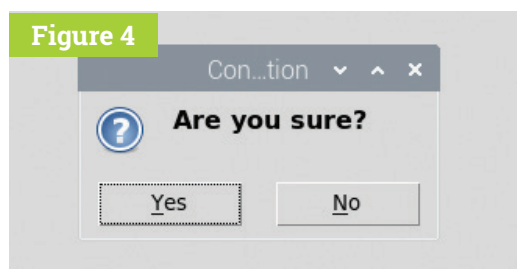
```
from guizero import App

app = App(title="pointless pop-ups")

app.info("Application started", "Well done
you started the application")

app.display()
```

Note de l'éditeur : cet article est paru initialement dans le magazine MagPi (le magazine officiel du Raspberry Pi).
La maison d'édition Elektor publie ce magazine en néerlandais, allemand et français (www.magpi.fr).



▲ Figure 4 Yes, we are sûrs !

Le premier paramètre passé à la méthode `info` est le titre de la fenêtre, le second est le message lui-même. Les méthodes `warn` et `error` s'utilisent comme `info` et n'en diffèrent que par l'icône affichée dans la pop-up.

Certaines pop-ups servent à obtenir une information. La plus simple à cet égard est la pop-up `yesno` présentant les options de réponse `Yes` et `No`. L'option `Yes` retourne `True`, `No` retourne `False`. Cette pop-up est utile pour demander confirmation d'une action, p. ex. la suppression d'un fichier. Cela dit, ne sollicitez pas une confirmation à chaque fois que l'utilisateur appuie sur un bouton ! Pour notre exemple, importez le widget `PushButton` :

```
from guizero import App, PushButton
```

Créez une fonction demandant confirmation avec la pop-up `yesno` :

```
def are_you_sure():
    if app.yesno("Confirmation", "Are you
sure?"):
        app.info("Thanks", "Button
pressed")
    else:
        app.error("Ok", "Cancelling")
```

Ajoutez un bouton appelant `are_you_sure` lorsqu'il est pressé :

```
button = PushButton(app, command=are_you_
sure)
```

Lancez le code (`05-worlds-worst-gui.py`) et cliquez sur le bouton : une pop-up attend votre décision (fig. 4).

Toutes les pop-ups de `guizero` sont décrites sur lawsie.github.io/guizero/alerts. (VF : Hervé Moreau)

worst5.py

Langage : Python 3

```
001. # Imports -----
002.
003. from guizero import App, Combo
004. from string import ascii_letters
005.
006.
007. # App -----
008.
009. app = App("Enter your name")
010.
011. name_letters = []
012. for count in range(10):
013.     a_letter = Combo(app, options=" " + ascii_letters,
014. align="left")
015.     name_letters.append(a_letter)
016.
017. app.display()
```

05-worlds-worst-gui.py

Langage : Python 3

```
001. from guizero import App, PushButton
002.
003. def are_you_sure():
004.     if app.yesno("Confirmation", "Are you sure?"):
005.         app.info("Thanks", "Button pressed")
006.     else:
007.         app.error("Ok", "Cancelling")
008.
009. app = App(title="pointless pop-ups")
010.
011. button = PushButton(app, command=are_you_sure)
012.
013. app.info("Application started", "Well done you started the
014. application")
015.
016. app.display()
```

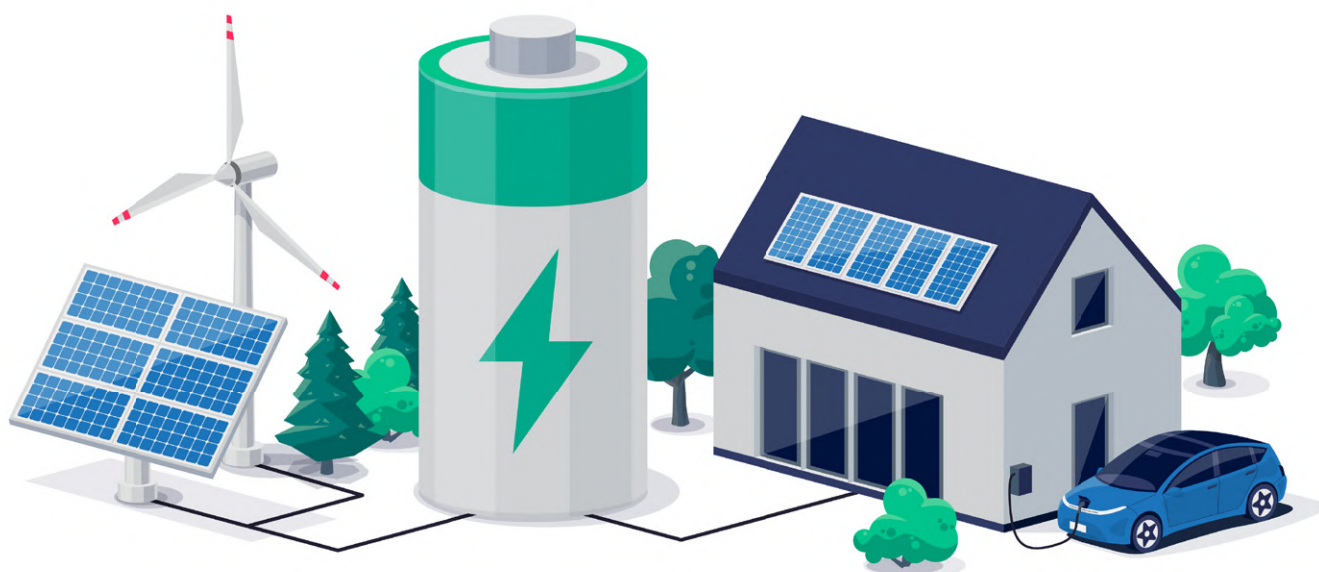
Python 3 Programming and GUIs

Destiné aux ingénieurs, scientifiques et amateurs, ce livre (en anglais) explique comment interfacier des PC et des projets matériels au moyen d'interfaces graphiques. L'écriture d'applications pour environnements de bureau et web est également abordée. Python 3 est un langage éminemment lisible, une qualité essentielle pour écrire rapidement des programmes.

Guide simple et pratique, cette seconde édition révisée et mise à jour entend mettre le pied à l'étrier aux débutants.

www.elektor.fr/python-3-programming-and-guis





systemes solaires autonomes

Production d'électricité indépendante du réseau

Thomas Scherer (Allemagne)

Qu'est-ce qu'un système solaire autonome ? Où un tel système est-il nécessaire ou pratique ? Quels sont les points principaux à considérer ? Cet article répondra à ces questions et à bien d'autres.

Dans le n° de sept./oct. 2021 d'*Elektor*, nous avons fait le tour d'une installation photovoltaïque raccordée au réseau [1]. Ici, nous aborderons principalement les installations solaires autonomes et isolées du réseau public. Elles permettent de produire de l'électricité là où le raccordement au réseau serait trop coûteux, par ex. dans un abri de jardin en habitat collectif, ou impossible, comme sur un bateau à moteur ou à voile. Ce sont en

général des systèmes de faible puissance de crête, de quelques watts à quelques kilowatts. En outre, les tarifs de rachat continuant de baisser, les installations solaires domestiques fixes plus simples uniquement à usage privé, stockant l'énergie produite dans des batteries au lieu de l'injecter dans le réseau public, commencent à prendre tout leur sens. Ces installations ont généralement une puissance maximale de quelques kWc (c pour crête). Voyons maintenant ces systèmes à petite échelle de plus près.

Principe de fonctionnement

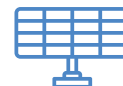
Une installation solaire autonome comporte au moins trois éléments : le panneau solaire, le stockage d'énergie sous forme de batterie et le régulateur de charge qui veille à ce que la batterie ne soit pas surchargée. Pour les petits systèmes, fonctionnant en 12 ou 24 V, en théorie cela suffit. Toutefois, s'il faut une sortie de 230 V_{CA} 50 ou 60 Hz, un 4^e élément entre en jeu : l'onduleur. La **figure 1** montre une solution type à quatre éléments. De prime abord, elle semble très simple, mais attention

aux détails. Dans les paragraphes ci-dessous, nous nous intéresserons à ces éléments.

Voici un exemple concret : *Klaus*, un de mes bons amis, décide d'installer un système 12 V dans son abri de jardin en raison du faible prix et de la taille raisonnable des composants. Pour concevoir le système et spécifier les éléments, il faut d'abord répondre à deux questions.

Énergie et puissance

La 1^{ère} question concerne la quantité d'énergie totale que le système doit stocker. Elle conditionne la capacité minimale de la batterie. Il faut donc estimer la charge moyenne de l'installation. Le nombre de jours de temps couvert que le système doit subir sans faiblir influe sur ce calcul. Dans son abri, Klaus aimerait de temps en temps utiliser une perceuse électrique et préparer une tasse de thé, mais d'utilisation occasionnelle, ces charges n'influent que peu sur le calcul de la charge moyenne. La nécessité d'avoir des boissons fraîches sous la main est plus importante : il faut donc un réfrigérateur 12 V



fonctionnant en continu, soit une consommation moyenne de 20 W. Il faut que le système puisse fonctionner pendant au moins un jour sans soleil.

La 2^e question concerne la puissance crête requise. À partir de là, nous pouvons déterminer le courant maximal qui sera tiré de la batterie et donc aussi spécifier les paramètres du régulateur de charge (et de l'onduleur, s'il est utilisé). En général, il est facile de répondre à cette question : pour l'abri de jardin de mon ami, la réponse est de 1 kW ; cela permet d'alimenter la bouilloire, une perceuse standard et peut-être une pompe à eau, le tout fonctionnant en 230 V.

Le panneau solaire

Sur 24 heures, le réfrigérateur de l'abri de Klaus consomme au plus 500 Wh. Bien situé dans une région ensoleillée du sud-ouest de l'Allemagne, le toit de l'abri est malheureusement à l'ombre d'un arbre et le panneau ne peut donc pas y être installé. Il faut le disposer verticalement sur le mur sud de l'abri, ce qui réduit sa puissance d'environ 30 % par rapport à un montage à l'angle optimal par rapport au soleil. Il faut surdimensionner le panneau d'environ 40 % pour compenser cette perte. Par chance, l'espace disponible est grand et le prix des panneaux a considérablement baissé ces dernières années. Le montage vertical a aussi des avantages : en hiver, la neige ne s'accumule pas ; de plus, comme le soleil est à un angle plus bas, le rendement augmente : dans le meilleur des cas, la bière restera fraîche même pendant les journées ensoleillées d'hiver.

Nous pouvons maintenant calculer la puissance requise pour le panneau. Dans cette région, nous pouvons compter sur une énergie incidente annuelle moyenne $\geq 1200 \text{ kWh/m}^2$ sur une année. Outre la consommation quotidienne d'énergie prévue, il faut prévoir une marge de sécurité de 100 %. Ainsi, pour 500 Wh/jour (du printemps à l'automne), il faut chercher à produire 1 kWh/jour. Sur la base de 8 h d'ensoleillement par jour, nous arrivons à une puissance requise du panneau d'environ 125 Wc. Pour compenser le montage vertical, nous arrivons à 175 Wc. Soit un panneau de 180 W, qui s'adaptera confortablement au mur de l'abri : voir **figure 2**.

La ou les batteries rechargeables

L'énergie nécessaire pour constituer une journée de réserve est de 500 Wh au moins.

En 12 V, cela équivaut à une batterie d'au moins 40 Ah de capacité. Comme notre onduleur est spécifié pour une puissance de sortie de 1 kW, il nous faut bien voir qu'à charge maximale, il exigera un courant d'entrée d'au moins 85 A. C'est une considération importante dans le choix d'une batterie. Nous devons d'abord décider du couple

électrochimique de la batterie. Un pack de batteries au lithium de 40 Ah peut facilement gérer ce courant (environ '2C', ou deux fois le courant que la batterie peut fournir pendant une heure) en raison de sa faible résistance interne. Cependant, un tel pack peut facilement coûter plus de 250 € et nécessite une gestion attentive. Klaus a préféré opter pour

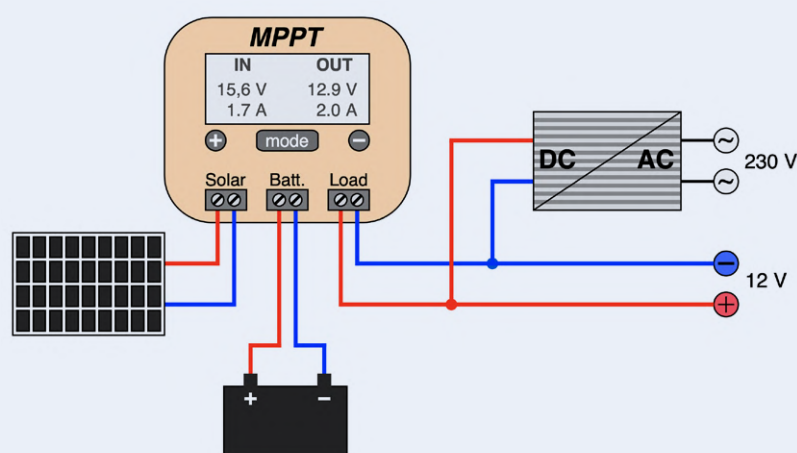


Figure 1. Câblage standard des quatre éléments composant une installation solaire autonome. L'onduleur à droite n'est nécessaire que pour faire fonctionner des appareils alimentés en 230 V_{CA}.



Figure 2. Le panneau solaire 12 V monté verticalement sur le mur de l'abri de Klaus. Il a une puissance de 180 Wc.



Figure 3. Trois batteries AGM 12 V, chacune de 36 Ah, sont câblées en parallèle pour stocker l'énergie dans l'abri de Klaus.

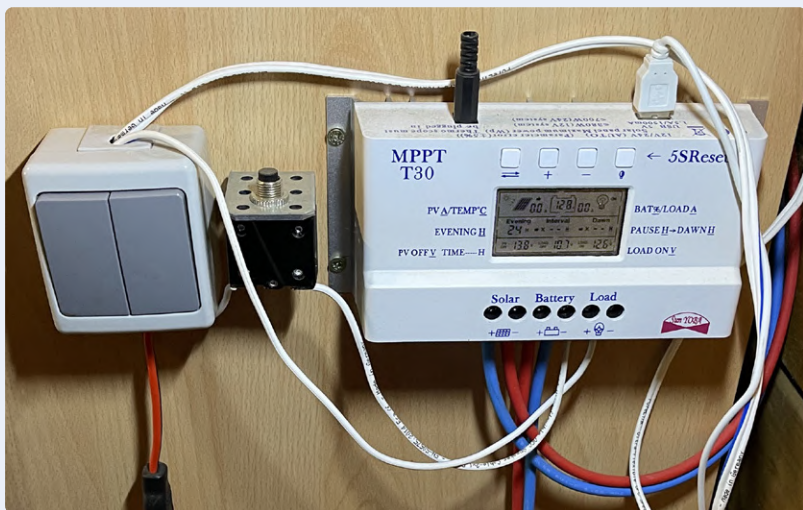


Figure 4. De gauche à droite : interrupteur d'éclairage, disjoncteur électromagnétique 30 A, et régulateur MPPT.



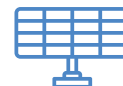
Figure 5. Les régulateurs de charge similaires à celui-ci n'ont certainement pas la fonctionnalité MPPT (même s'ils portent un autocollant MPPT !). (Source : Ministère de l'Énergie des États-Unis)

une classique batterie au plomb coûtant une fraction de ce prix. Il a porté son choix sur une batterie de voiture, car elle supporte des courants de pointe élevés. Cependant, elle présente des inconvénients : rendement et durée de vie faibles ; taux d'auto-décharge élevé. Il a donc choisi une batterie au gel, mais avec ce type de batterie le courant de décharge nominal est moindre. Pour compenser, deux batteries de 36 Ah chacune ont donc été connectées en parallèle. Cette combinaison offre une énergie nominale stockée de 864 Wh et coûte un peu moins de 150 €.

En plein soleil, le panneau solaire choisi fournit assez d'énergie pour charger complètement les batteries en une seule journée, et leur capacité permet de couvrir 1,5 jour sans soleil. J'avais quelques doutes quant à la forte consommation de courant à l'entrée de l'onduleur, mais Klaus a décidé de tenter le coup et, si deux batteries ne suffisaient de compléter par une 3^e en parallèle. Une fois le système installé et les batteries complètement chargées, nous nous sommes amusés à faire un 1^{er} essai avec une bouilloire de 1 kW. La tension de la batterie tomba à 11,7 V, mais il fut néanmoins facile de faire bouillir un demi-litre d'eau. À ce courant élevé, une estimation rapide donne un rendement de batterie (énergie restituée/énergie de recharge) de 50 % au plus et ce courant n'est pas favorable à la longévité de la batterie. Une batterie supplémentaire de 36 Ah a donc été ajoutée : voir **figure 3**. Lors d'une décharge à 85 A, la tension initiale est maintenant de 12,6 V, ce qui est plus acceptable ; la capacité totale atteint maintenant près de 1,3 kWh, ce qui garantit plus de deux jours de réserve.

Le contrôleur de charge

Une recherche sur eBay ou chez un distributeur spécialisé permet d'en trouver un large éventail. Les régulateurs 10 A ne coûtent pas plus de 15 €. Cependant, un panneau de 180 Wc fournit 15 A sous 12 V, il faut donc un régulateur d'au moins 20 A : ces régulateurs coûtent environ 20 €. Si le régulateur de charge doit être connecté comme indiqué sur la figure 1, il est préférable de choisir une version 100 A, coûtant peut-être un peu moins de 50 €. Entrons maintenant dans les détails. Le régulateur a pour mission de charger la batterie grâce à l'énergie fournie par le panneau, et de stopper la charge lorsqu'une



tension de seuil est atteinte. Ainsi la batterie connectée ne sera pas surchargée et donc non endommagée. Presque tous les régulateurs commandent la connexion de la charge et, pour empêcher la décharge profonde de la batterie, la déconnectent quand elle atteint son seuil inférieur de tension. Ils utilisent toujours un μ contrôleur et la plupart d'entre eux sont configurables pour s'adapter à différents types de batteries, notamment plomb-acide, plomb-gel et lithium. Ils s'adaptent aussi automatiquement à une tension nominale aux bornes de 12 ou 24 V. Souvent, il est possible de configurer manuellement les seuils de sous-tension et de surtension.

La topologie du chargeur est un point central. Tous les appareils bon marché utilisent une commande MLI (en anglais PWM = *Pulse Width Modulation*) même si MPPT (*Maximum Power Point Tracking*) est mentionné : une étiquette ne coûte rien, mais un vrai chargeur MPPT est meilleur, plus complexe et donc plus cher !

En MLI, le courant de charge est régulé de sorte que la tension de sortie du panneau soit juste au-dessus de la tension instantanée aux bornes de la batterie. La batterie est donc chargée au courant maximal possible résultant du flux lumineux total reçu par le panneau, et de l'état de charge de la batterie dans une large gamme de conditions. Le circuit nécessaire n'emploie qu'un μ contrôleur simple et un MOSFET de puissance : une solution peu coûteuse,

mais non optimale.

Or, la puissance de sortie d'un panneau est égale au produit de sa tension par son courant de sortie. Pour un panneau et un niveau d'éclairage donnés, il existe un point où ce produit est maximal ; en ce point de puissance maximale, on a presque toujours : *tension de sortie du panneau > tension de la batterie*. Un contrôleur MPPT détermine en permanence où se trouve ce *point idéal* et pilote le régulateur de tension abaisseur de sorte qu'il consomme le courant optimal et donc délivre la puissance maximale possible. Dans le cas le plus favorable, la puissance de sortie d'un régulateur MPPT dépasse de 30 % celle d'un MLI. Cependant, cela a un coût : même bon marché, un contrôleur MPPT coûte plus de 50 €, et une unité de marque au moins 100 €. Le régulateur de charge 30 A de la **figure 4** est un régulateur MPPT bon marché (environ 60 €) et Klaus a estimé que son gain en puissance de sortie en valait la peine. Si vous voulez un régulateur MPPT, évitez ceux qui ressemblent à celui de la **figure 5** : ce type de régulateur existe en diverses couleurs et avec différents marquages.

L'onduleur

S'il faut produire une sortie de 230 V_{CA}, alors un onduleur est essentiel. Les modèles bon marché avec des spécifications de puissance peu crédibles et des formes d'onde de sortie très éloignées d'une sinusoïde sont à éviter. Un point important à noter est que la puissance



Figure 6. Cet onduleur de 1 kW fabriqué par Ective s'est révélé très stable et fiable au fil du temps.

continue maximale spécifiée est donnée pour une charge ohmique. L'onduleur de 1 kW de Klaus convient parfaitement pour alimenter une bouilloire de 1 kW et 500 ml, mais la situation est tout à fait différente avec une charge inductive ou, cas plus rare, capacitive. Il faut alors aussi vérifier la puissance réactive : notez que la puissance apparente est toujours \geq à la puissance réelle. Parmi les charges les plus difficiles, on trouve les moteurs électriques. Leur courant d'appel élevé peut déclencher

Publicité

PERFORMANCE. RELIABILITY. SERVICE.

Optocouplers by Würth Elektronik



WÜRTH
ELEKTRONIK
MORE THAN
YOU EXPECT

Optocouplers by Würth Elektronik

With the new optocouplers, Würth Elektronik presents one of the latest additions to its optoelectronic product portfolio. The innovative design features a coplanar structure and high-grade silicon for total internal reflection. The coplanar design ensures the isolation gap stay fixed during the production process and provide perfect isolation and protection for your application. The total internal reflection provide stable CTR over the whole temperature range and high CTR even at low current operation.

Provided in all industry standard packages. Available with all binnings ex stock. Samples free of charge: www.we-online.com/optocoupler

- Innovative coplanar design
- High grade silicon encapsulation
- Copper leadframe for high reliability
- Stable CTR over whole temperature range
- High CTR in low current operation

WE meet @ PCIM Europe
Hall 6, Booth 402

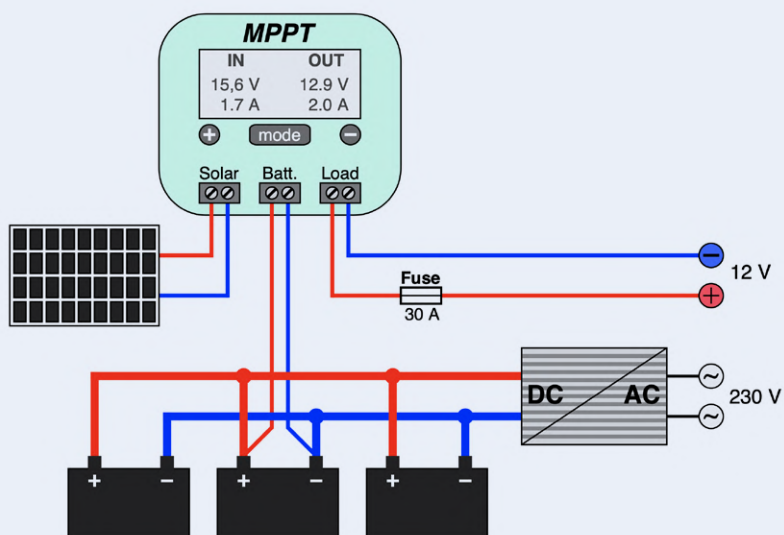


Figure 7. Dans l'installation de Klaus, l'onduleur est connecté directement à la batterie, et la sortie 12 V du régulateur bénéficie d'une protection supplémentaire.

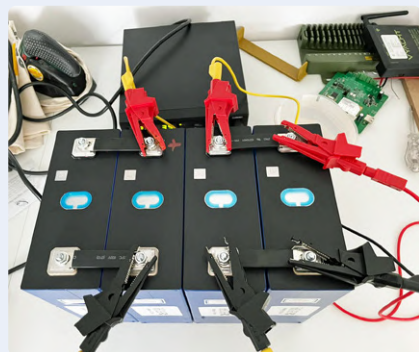


Figure 8. Essai de batteries LiFePO₄ avant l'installation dans le bateau de Martin. (Source : Martin Jepkens)



Figure 9. De puissance nominale 120 Wc, le panneau solaire pliant choisi par Martin et Detlev peut être rangé sous le pont en navigation. (Source : Martin Jepkens)

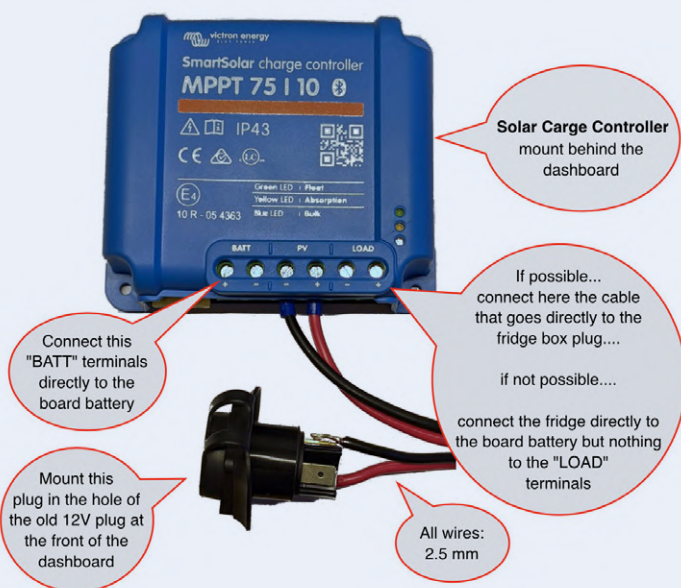


Figure 10. Guide d'installation du régulateur dans le bateau de Detlev.

le circuit de protection antisurcharge d'un onduleur mal dimensionné. Une marge de sécurité de 100 % pour utiliser un moteur n'est pas excessive, même pour un onduleur de haute qualité. L'onduleur de 1 kW de Klaus, illustré à la **figure 6**, gère confortablement une perceuse électrique et une pompe à eau de 450 W. Il coûte plus de 200 €.

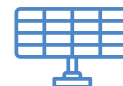
Câblage

Comme vous l'avez peut-être deviné en regardant les illustrations, le câblage entre le panneau solaire, le régulateur et la batterie est réalisé à l'aide de fil multibrin de 6 mm² de section ; les connexions aux prises allume-cigare 12 V ne sont pas représentées. La mise en parallèle des batteries elles-mêmes est réalisée avec du fil de 16 mm². La section de fil utilisée doit être adaptée au courant à transporter : tenter de faire des économies sur ce point serait une erreur.

L'onduleur est connecté directement à la batterie à l'aide d'un fil de 16 mm² pour réduire les pertes. Seule une protection contre la sous-tension permet la connexion directe de l'onduleur (comme c'est le cas ici) à la batterie pour la protéger contre une décharge profonde. L'onduleur n'est activé que lorsque la sortie 230 V est réellement requise : sinon, au courant de repos, sa consommation (quelques dizaines de milliampères) serait un gaspillage inutile d'énergie. La configuration finale est donc celle de la **figure 7**.

Autres systèmes autonomes

L'alimentation électrique de l'abri de Klaus est un exemple type d'installation solaire autonome. Divers fournisseurs proposent des packs *panneau solaire + régulateur de charge + onduleur* prêts à l'emploi de différentes puissances nominales. Si vous optez pour l'énergie éolienne plutôt que solaire, il existe également des générateurs et régulateurs de charge adaptés, basés sur les mêmes principes. Pour ma part, l'an passé, j'ai modifié ma tondeuse robot pour la rendre autonome [2]. Un panneau de 50 W et un simple contrôleur de charge MLI firent l'affaire ; aucun onduleur n'était nécessaire. Depuis, je suis passé d'une batterie de 12 Ah à une de 30 Ah, pour aider à couvrir les périodes pluvieuses que nous avons eues. J'ai aussi récemment remplacé le contrôleur MLI par un meilleur (et plus cher) contrôleur MPPT. Le système produit maintenant assez d'électricité pour tondre la pelouse même tard en automne.



L'alimentation autonome a bien sûr beaucoup d'autres applications. Deux autres de mes amis ont un bateau : *Martin* navigue sur un grand bateau en acier à travers les paysages fluviaux des Pays-Bas, tandis que *Detlev* écume la Méditerranée sur son bateau de sport à carène planante. Tous deux passent souvent des jours loin d'un port ou autre lieu d'amarrage avec électricité disponible et aimeraient donc avoir plus d'autonomie, surtout en ce qui concerne la réfrigération : pour la boisson, mais aussi pour conserver des aliments. Il serait peu écologique, voire inefficace, de faire régulièrement tourner le moteur pour charger la batterie de bord, les deux bateaux ont donc été équipés en systèmes solaires.

Ingénieur intelligent, Martin n'a pas besoin de mes conseils. Néanmoins, de temps en temps, il discute de ses idées avec moi. Il se demandait si le générateur de son bateau pouvait être surchargé en y connectant une grosse batterie LiFePO₄ de 200 Ah. Les dangers

d'une telle démarche sont analysés dans une vidéo YouTube [3]. La **figure 8** montre la configuration des batteries pour l'essai de capacité : il choisit une batterie LiFePO₄ principalement en raison de sa longévité, mais aussi de sa compacité par rapport à une batterie au plomb. Dans le bateau de Martin, les batteries de bord et de démarrage sont distinctes. Afin d'alléger la charge de l'alternateur, les batteries sont chacune chargées par leur propre régulateur quand le moteur tourne. Un panneau solaire pliant de 120 Wc et un régulateur sont également installés pour la charge en cas d'arrêt dans un endroit sans accès à une prise de courant (voir **fig. 9**).

En raison du manque d'espace disponible, une installation fixe de panneaux solaires ne convient pas sur le bateau de sport de Detlev. Il opta pour le même type de panneau solaire que Martin, même si aucun des deux ne savait ce que l'autre avait choisi ! Detlev n'est pas féru d'électronique, et au début, il voulait utiliser sa batterie de bord supplémentaire de 120 A,

car elle était plutôt récente. J'ai fait quelques calculs pour lui et je lui ai dit que question fiabilité, utiliser la prise allume-cigare du pont de son bateau pour connecter le panneau solaire n'était pas une bonne idée : je lui ai suggéré d'utiliser plutôt un connecteur étanche Neutrik. J'ai précâblé le connecteur et rédigé un guide d'installation (**fig. 10**) à l'adresse du constructeur de son bateau, afin que le système soit installé dans le port d'Istrie. La combinaison panneau pliant de 120 Wc/contrôleur de charge MPPT de *Victron Energy* a coûté un peu moins de 500 €. Le régulateur de charge dispose d'une connexion *Bluetooth* et une application pour smartphone affiche tous les paramètres et graphiques.

La maison semi-autonome

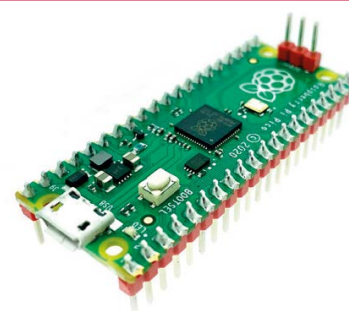
En ces temps de baisse des tarifs de rachat, l'installation solaire fixe qui, dans la mesure du possible, consacre toute l'énergie produite à la demande de la maison connaît un regain d'intérêt. Par ex., un ensemble de dix panneaux

Publicité

ABONNEZ-VOUS ET RECEVEZ

Raspberry Pi + Headers GRATUIT

TOUS LES 2 MOIS, LES DERNIÈRES NOUVELLES
DU RASPBERRY PI ET LES MEILLEURS PROJETS !



SEULEMENT
54,95 €
PAR AN
(6 NUMÉROS)

Vos avantages :

- Une carte Raspberry avec headers
- Prix au numéro réduit
- Chaque numéro directement dans votre boîte aux lettres
- Tous les numéros disponibles sous forme numérique (PDF)
- Découverte de chaque nouveau numéro avant sa sortie en kiosque



ABONNEZ-VOUS : WWW.MAGPI.FR

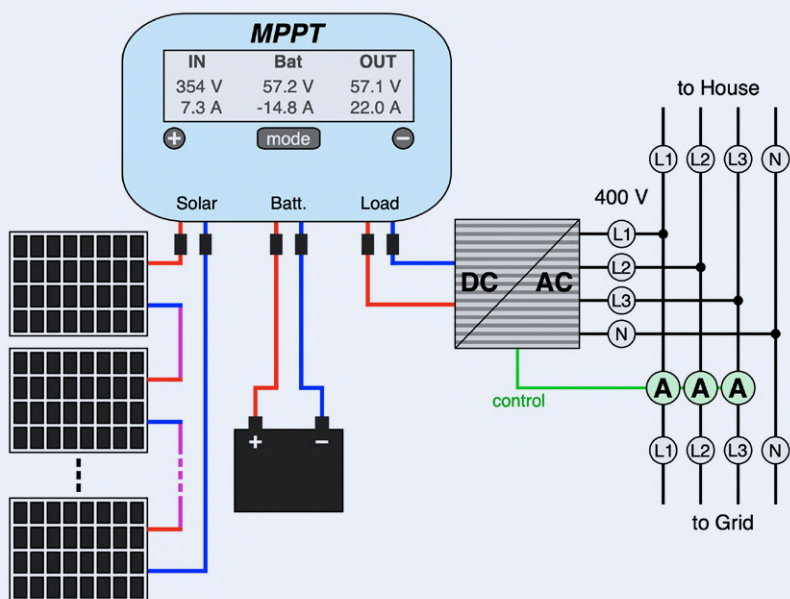


Figure 11. Installation solaire semi-autonome pour une maison. La mesure du courant triphasé commande l'onduleur de manière à n'envoyer aucune énergie électrique sur le réseau.

solaires modernes produit environ 3,75 kWc ; un régulateur de charge solaire MPPT approprié charge ensuite une batterie LiFePO₄ d'une capacité de 6,5 kWh ; un circuit de surveillance du courant (les trois ampèremètres en bas à droite de la **figure 11**) pilote un onduleur triphasé pour s'assurer qu'en aucun cas l'énergie électrique n'est envoyée sur le réseau. Tout le « courant » est donc utilisé localement. Avec en 2022, un coût de l'électricité en Europe d'environ 0,35 €/kWh, cette option est très intéressante : non seulement elle évite d'avoir un onduleur complexe et coûteux relié au réseau avec une électronique de charge intégrée pour la batterie, mais elle dispense aussi de formalités compliquées. C'est une considération non négligeable ! (Notez qu'un tel montage peut ne pas être légal dans tous les pays).

Avec le schéma de la figure 11, les économies réalisées grâce à l'utilisation de la conception simplifiée vont de 1 000 € à 2 000 €. Au tarif de rachat, il faudrait quelques années pour récupérer ce montant. La partie la plus coûteuse du système est la batterie :

une batterie LiFePO₄ de 6,5 kWh coûte plus de 3 000 €. Avec une garantie de 6 000 cycles de charge à une profondeur de décharge de 90 %, cela signifie qu'environ 36 MWh d'énergie auront transité par la batterie, ce qui fait que le coût de la batterie est d'environ 0,09 €/kWh. De plus, à ce stade, la capacité de la batterie est encore appréciable et le coût

effectif du kWh sera donc encore plus bas. Sur la durée de vie de la batterie, une solution de ce type permet d'économiser environ 13 000 € grâce à l'énergie produite localement. En rechargeant aussi un véhicule électrique (à un faible taux de charge), une installation de ce type peut s'amortir en quelques années. ◀

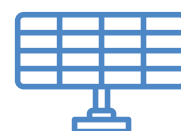
210644-04

Contributeurs

Texte et illustrations (sauf indication contraire) : **Thomas Scherer**
Rédaction : **Jens Nickel**
Traduction : **Yves Georges**
Mise en page : **Harmen Heida**

Des questions, des commentaires ?

Contactez Elektor (redaction@elektor.fr).



PRODUITS

- **PeakTech 4350 - pince ampèremétrique**
www.elektor.fr/18161
- **Pokit Meter - multimètre, oscilloscope et enregistreur portable**
www.elektor.fr/19854
- **PeakTech 3445 - multimètre numérique à valeur efficace, avec Bluetooth**
www.elektor.fr/18774

LIENS

- [1] « Centrale solaire sur balcon », T. Scherer, Elektor, 09-10/2021 : www.elektormagazine.fr/210326-04
- [2] « Énergie solaire pour les robots de tonte », T. Scherer, Elektor 07-08/2021 : www.elektormagazine.fr/200553-04
- [3] « How to not blow up your alternator when charging lithium », Victron Energy, YouTube : www.youtube.com/watch?v=jgolocPgOug.



10 ans avec le même smartphone ?

Et si c'était possible !

Priscilla Haring-Kuipers (Pays-Bas)

Ce que nous attendons de nos appareils détermine ce que nous acceptons d'eux, notamment leur longévité. Qu'en est-il de nos smartphones ?



Avez-vous vu cette publicité en ligne annonçant un smartphone d'une durée de vie de 10 ans ? Si vous avez cliqué dessus, sans doute avez-vous été déçu de constater que l'appareil n'existait pas... encore. L'annonce est de fait l'outil d'une campagne plus large, baptisée « 10 Year Smartphone », visant à attirer l'attention des consommateurs et législateurs européens sur la réparabilité en général. Un monde dans lequel un « smartphone de 10 ans » serait la norme nécessiterait une législation sur la réparabilité, la durée de la prise en charge logicielle et la disponibilité des pièces de rechange – en particulier la batterie. Le consommateur devrait connaître le score de réparabilité de son appareil et disposer d'informations expliquant comment le réparer soi-même [1].

Attrait fatal

Il semblerait qu'au fil du temps nous ayons accepté que nos indispensables et coûteux mobiles ne durent que de deux à quatre ans (en parfait état de marche). Je suppose que nous ne réfléchissons que rarement à ce que devrait être la longévité de nos appareils.

Quelle devrait être celle d'un écran plat 4k ? D'une machine à laver ? D'un réveil ? D'un four à refusion ?

Deux à quatre ans de vie, ce n'est pas si mal pour la plupart des appareils si l'on considère notre attrait pour les choses nouvelles. Mais cet attrait n'est-il pas étrange ? D'où vient ce besoin d'être « à jour » ? Ne serait-il pas temps de s'en défaire ? La planète subit notre frénésie d'achat, et la fabrication de bon nombre d'appareils peut aussi être source de souffrances humaines. Et si nous adaptions nos modèles mentaux et nos appareils afin qu'ils durent plus longtemps ?

Nouveau réflexe

Mon smartphone a cinq ans et sa batterie passe de 42 % à « vide » en un rien de temps. Alors que je furetais sur l'internet en quête d'un nouveau modèle, j'ai réalisé que l'actuel me satisfaisait encore, batterie mise à part. Je ne suis pas la seule à avoir ce genre de réflexe : seulement 11 % d'Européens font réparer leur téléphone lorsqu'il tombe en panne. Aurais-je spontanément cherché une nouvelle machine à laver si elle avait refusé d'effectuer son cycle

d'essorage ? Je ne le pense pas. Mon réflexe aurait plutôt été d'essayer une solution trouvée en ligne, comme déboucher je ne sais quelle buse. C'est le genre d'appareil que je cherche à entretenir et réparer moi-même. Je n'aurais appelé un professionnel que si la machine avait eu moins de 10 ans.

Machines à laver et smartphones sont dans la même gamme de prix, pourtant nous n'attendons pas de nos mobiles qu'ils durent 10 ans. Il est temps de se montrer plus intelligents.

L'initiative 10 Year Smartphone a été lancée par Right to Repair, une coalition regroupant plus de 80 organisations européennes actives sur le sujet de la réparation. ▶

210714-04 – VF : Hervé Moreau

LIEN

[1] Campagne « 10 Year Smartphone » : <https://10yearphone.com/>

e-choppe Elektor

des produits et des prix surprenants

L'e-choppe Elektor a évolué. D'abord simple commerce de proximité pour nos productions (livres, magazines, kits et modules), c'est désormais une boutique en ligne bien rodée qui propose des produits surprenants à des prix très étudiés. Ce sont les produits que nous

aimons et testons nous-mêmes. Si vous avez une suggestion, n'hésitez pas : sale@elektor.fr. Seule exigence :
jamais cher, toujours surprenant !

Four de refusion à infrarouge T-962 (puhui)

Prix : 229,00 €

Prix (membres) : 206,10 €

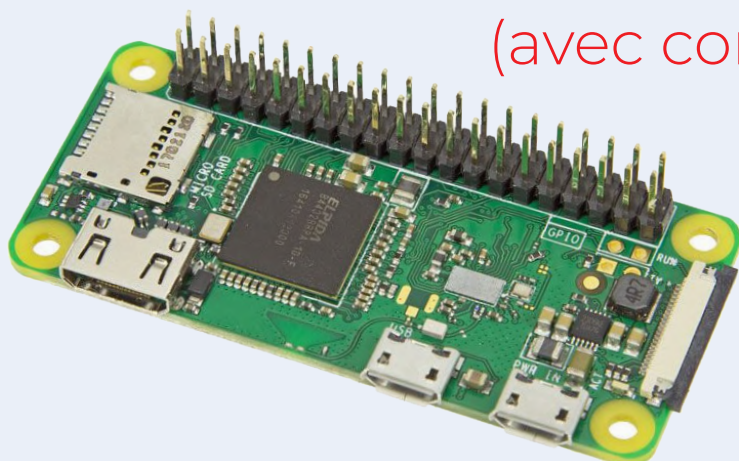
 www.elektor.fr/20135



Raspberry Pi Zero 2 WH (avec connecteur GPIO soudé)

Prix : 34,95 €

Prix spécial : 29,95 €



 www.elektor.fr/20157



Kit de développement MakePython ESP32

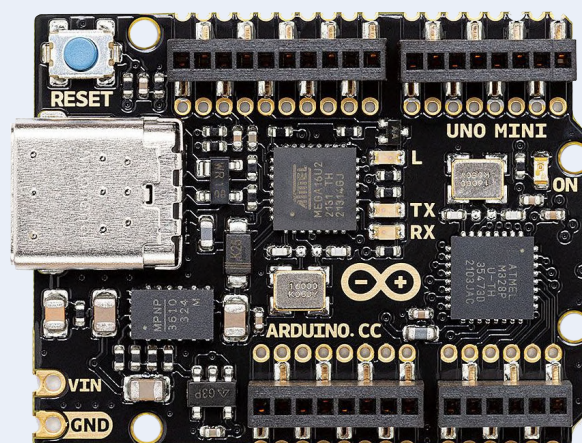


Prix : 74,95 €

Prix spécial : 64,95 €

www.elektor.fr/20137

Arduino Uno Mini (Limited Edition)

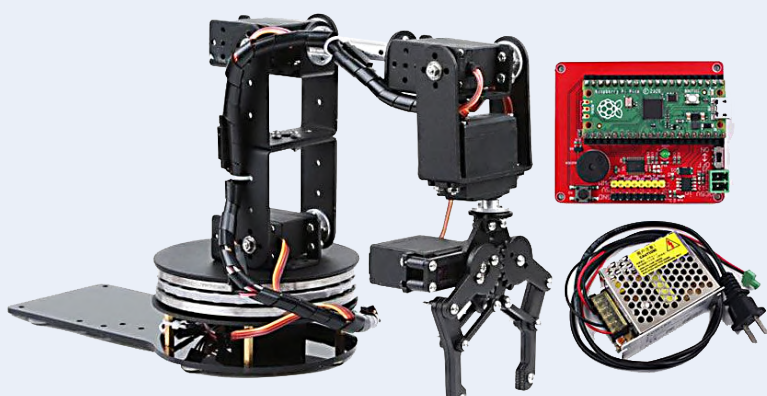


Prix : 54,95 €

Prix (membres) : 49,46 €

www.elektor.fr/20098

Bras de robot 6 DOF (Makerfabs) avec Raspberry Pi Pico

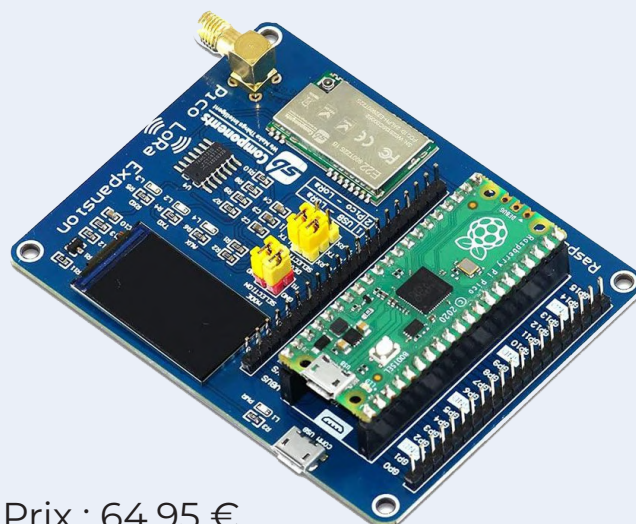


Prix : 139,95 €

Prix (membres) : 125,96 €

www.elektor.fr/20130

Raspberry Pi Pico LoRa Expansion + Raspberry Pi Pico GRATUIT



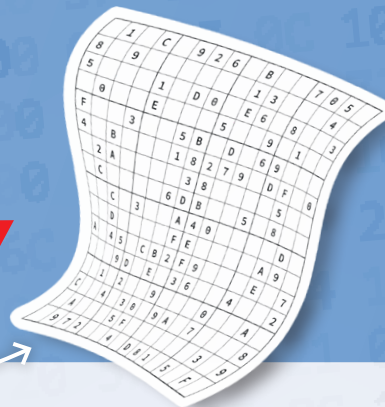
Prix : 64,95 €

Prix (membres) : 58,46 €

www.elektor.fr/20096

hexadoku

casse-tête pour elektorniciens



La dernière page de votre magazine propose toujours une grille de sudoku particulière. Éteignez le fer à souder, oubliez les écrans, rangez les pinces et les fils. Faites travailler votre matière grise d'une autre façon : attrapez un crayon pour remplir la grille d'hexadoku. N'oubliez pas d'envoyer votre réponse pour tenter de gagner un bon d'achat.

Une grille hexadoku est composée de chiffres du système hexadécimal, de 0 à F. Remplissez le diagramme de 16 x 16 cases de telle façon que **tous** les chiffres hexadécimaux de 0 à F (0 à 9 et A à F) n'apparaissent **qu'une seule et unique fois** dans chaque rangée, colonne et carré de 4 x 4 cases (délimités par un filet gras). Certains chiffres, déjà placés dans la grille, en définissent la situation de départ.

Pour participer, inutile de nous envoyer toute la grille, il suffit de nous donner **la série de chiffres** sur fond grisé.



Participez et gagnez !

Nous tirons au sort cinq des réponses internationales correctes reçues dans les délais ; leurs auteurs recevront chacun un bon d'achat Elektor d'une valeur de 50 €.

Où envoyer votre réponse ?

Envoyez votre réponse (les chiffres sur fond grisé) avec vos coordonnées par courriel, avant le **15 juin 2022** à l'adresse **hexadoku@elektor.fr**

LES GAGNANTS

La solution de la grille du numéro de mars/avril 2022 est **C73B8**.

La liste des gagnants est publiée ici : www.elektormagazine.fr/hexadoku

Bravo à tous les participants et félicitations aux gagnants !

	D	7		1	6					B		8	0
4	6			D	5			0				C	F
2				F	4				B				E
			3	8		2		A	9		D		5
D	E	B	5	2			6			8			3
8	0	3				1			2	F	C	D	
			4		8	0			7		F		
				3	4			A		5			
	1		E						B		6		F
			C		D	8		7			E		9
		2			E	0				F	5	3	7
			7	B	F		3	E		1	4		
6					1	A			C	D		3	9
	A							1		5		4	
9	C	5	D	7					4	2			1
B		4		6				9	E			C	7

E	2	6	9	0	3	C	A	F	5	7	B	1	4	8	D
4	C	7	3	B	8	5	D	E	1	9	6	A	F	2	0
5	B	0	D	E	F	4	1	C	2	8	A	3	6	7	9
8	F	1	A	2	6	7	9	3	D	0	4	5	C	B	E
A	3	2	0	5	C	F	8	4	6	D	9	B	7	E	1
1	D	B	8	4	2	E	6	0	7	A	5	C	3	9	F
7	5	C	6	1	9	A	B	2	3	E	F	4	0	D	8
F	E	9	4	3	7	D	0	1	8	B	C	2	A	5	6
3	0	D	E	6	A	9	F	B	C	1	2	7	8	4	5
2	6	F	1	C	B	8	E	5	4	3	7	9	D	0	A
9	4	5	7	D	0	1	2	8	A	F	E	6	B	3	C
B	8	A	C	7	5	3	4	9	0	6	D	E	1	F	2
0	7	3	2	8	4	6	5	D	E	C	1	F	9	A	B
6	A	4	B	F	E	0	C	7	9	5	8	D	2	1	3
C	1	E	F	9	D	2	3	A	B	4	0	8	5	6	7
D	9	8	5	A	1	B	7	6	F	2	3	0	E	C	4

Tout recours est exclu, de même que le sont, de ce jeu, les personnels d'Elektor International Media et leur famille. Un seul gagnant par foyer.



AISLER

Votre design en tant que **prototype puissant**

Jusqu'à 30 € offerts sur votre 1er PCB AISLER



Beautiful
Boards



Stellar
Stencil



Amazing
Assembly

Commandez votre PCB maintenant et

GAGNEZ

une Voltera V-One
imprimante de PCB
de bureau
(d'une valeur de 4150 €)



UTILISEZ LE CODE

ELCCBBL

Valable jusqu'au 30 juin 2022



COMMANDEZ MAINTENANT !

www.elektormagazine.fr/gagnez-une-voltera-v-one

Rejoignez les électroniciens de la communauté Elektor

Devenez membre



maintenant !



- ✓ accès à l'archive numérique depuis 1978 !
- ✓ 6x magazine imprimé Elektor
- ✓ 9x magazine numérique (PDF) dont Elektor Industry (EN)
- ✓ 10 % de remise dans l'e-choppe et des offres exclusives pour les membres
- ✓ le DVD annuel d'Elektor

- ✓ accès à plus de 1000 fichiers Gerber, collaboration avec les milliers d'électroniciens d'Elektor LAB, et une ligne directe avec nos experts !
- ✓ possibilité de voir votre projet publié ou vendu par notre boutique en ligne

Également disponible

abonnement



sans papier !

- ✓ accès à l'archive numérique d'Elektor
- ✓ 10 % de remise dans l'e-choppe
- ✓ 6x magazine Elektor (PDF)
- ✓ offres exclusives
- ✓ accès à plus de 1000 fichiers Gerber



www.elektormagazine.fr/membres