

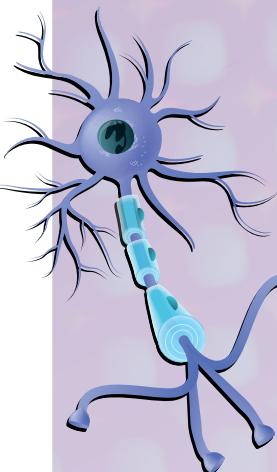


PRISE DE VUE ET DIFFUSION VIDÉO EN CONTINU



avec un Raspberry Pi 4

p. 50



nouvelle série
voyage dans les réseaux neuronaux

p. 6



TRAITEMENT D'IMAGES
avec le module Jetson Nano de Nvidia

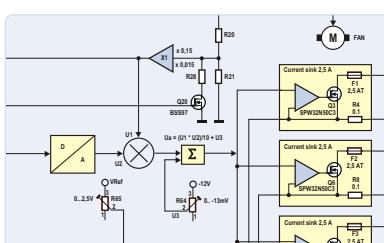
p. 30



dans ce numéro :

- **60 ans d'Elektor :** regard sur septembre
- préconformité CEM pour un projet alimenté en courant continu
- lévitation magnétique : version numérique
- Propeller 2 de Parallax : envoi de chaînes de caractères
- zoom sur les transformateurs secteur
- module de bus CAN pour le Raspberry Pi 4
- rose des vents avec le capteur GY-271
- questions d'éthique : mesurez votre empreinte carbone
- démarer en électronique : les selfs

et bien d'avantage !



charge électronique
en CC et CA
jusqu'à 400 V et 10 A (crête)

p. 20



centrale solaire sur balcon
un investissement vite amorti

p. 44

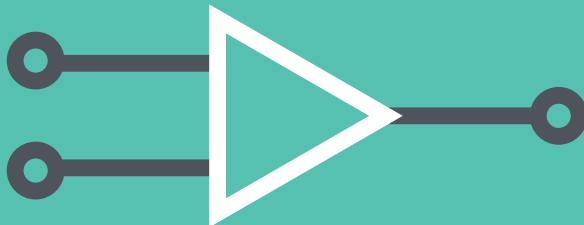


thermostat connecté à ESP32
conservez votre vin à la bonne température !

p. 84



NOTRE SERVICE DE DEVIS VOTRE COMPARATEUR DE PRIX PERSONNALISÉ



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

Dès aujourd'hui, optimisez votre budget rapidement et en toute simplicité

Vous avez besoin de plus grandes quantités, vous souhaitez regrouper vos achats ou vous recherchez des produits ne figurant pas dans notre programme ? Nous nous ferons un plaisir de comparer pour vous les meilleurs fournisseurs sur le marché afin de vous proposer le meilleur prix reichelt possible dès la première offre.

Multimètre numérique TRMS, série UT 161

Polyvalent et économique

Grâce à son fonctionnement simple et à ses importantes fonctions de mesure de base, ce multimètre convient pour les loisirs, les formations et les études, mais aussi pour le développement. Le port USB permet de connecter le multimètre à un ordinateur et de le commander à distance, ainsi que d'archiver facilement des valeurs de mesure et les exporter au format XLS pour une utilisation ultérieure.

- Tension : 1 000 V c.a. / c.c.
- Intensité : 0 - 10 A c.a. / c.c.
(20 A pour les modèles D et E)
- Fréquence : 10 Hz - 10 MHz
(220 MHz pour les modèles D et E)
- Sélection de plage automatique et manuelle
- Test de diode et test de continuité
- Mesure de la capacité et la résistance



UT 161D:
avec mesure
de température

UNI-T



à partir de **70,72**
(58,94)

N° de commande :	Chiffres / Graphique à barres	Précision de base	Capacité	Résistance	Version
UT 161B	70,72 (58,94)	6.000 / oui	0,10 %	jusqu'à 60 mF	0,1 Ohm - 60 MOhm
UT 161D	80,81 (67,34)	6.000 / oui	0,10 %	jusqu'à 60 mF	0,1 Ohm - 60 MOhm
UT 161E	100,97 (84,15)	20.000 / oui	0,10 %	jusqu'à 220 mF	0,1 Ohm - 220 MOhm

UT 161 E:
avec mesure hFE

Vous trouverez des appareils de mesure pour de nombreuses autres applications en ligne !



Découvrir tout de suite ► www.reichelt.com/technique-de-mesure

UTILISEZ LE TEMPS À VOTRE AVANTAGE
DES TECHNOLOGIES
EFFICACES POUR VOTRE
ENTRETIEN D'ÉTÉ



En savoir plus ►



Types de paiement :



PRIX DU JOUR! Prix à la date du: 11. 8. 2021

■ 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

44^{ème} année
n° 491 – septembre-octobre 2021

ISSN 0181-7450
Dépôt légal : septembre 2021
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

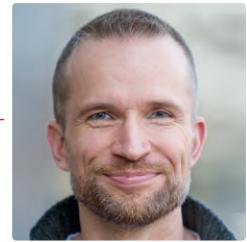
DROITS D'AUTEUR :
© 2021 Elektor International Media B.V.

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'œuvre 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 Pijper Media – Groningen
Distribué en France par M.L.P. et en Belgique par A.M.P.

rédacteur en chef d'Elektor Magazine



Testé et approuvé

Si j'osais, je vous dirais que moi aussi j'ai contracté le virus du Raspberry Pi. Mais je ne voudrais pas vous choquer. Malheureusement, à l'heure où j'écris ces lignes, le relâchement prudent des mesures de confinement et de protection est freiné, voire inversé dans certains pays. La situation reste encore très incertaine.

Heureusement, vous trouverez dans ce numéro une foule de projets passionnantes que vous pourrez reproduire, modifier et adapter sur votre paillasse. C'est une activité totalement compatible avec la pandémie.

La publication des circuits de nos lecteurs (par exemple à la page 20, « charge électronique en CC et CA ») est une tradition de longue date. Nos lecteurs imaginent des montages, les réalisent et les peaufinent généralement pendant leur temps libre. Toutefois j'ai remarqué que ces projets sont souvent liés aux activités électronico-professionnelles de nos lecteurs.

Un certain nombre de projets qui nous sont soumis par des auteurs extérieurs sont reproduits et testés par mes collègues du labo. C'est ce qui nous distingue de nombreux autres magazines et plateformes web ! Elektor a son propre labo d'électronique. Dorénavant nous signalerons ces projets avec notre tampon « Elektor Lab Tested ». Quant aux montages créés par notre laboratoire (depuis des décennies), ils seront estampillés du sceau « Elektor Lab Original ».

Si tout va bien, je publierai un article « Elektor Lab Original » dans l'un des prochains numéros. Comme je diffuse souvent des vidéos en direct (entre autres pour montrer des DJ), j'ai été inspiré par l'article de mon collègue Mathias Claussen, intitulé « prise de vue et diffusion vidéo en continu avec un Raspberry Pi 4 » (page 50). Il explique comment partager des images en direct sur le web avec un Raspberry Pi. Actuellement je cherche à réaliser un petit boîtier mobile pour diffuser le signal HDMI d'une caméra vidéo sur une plateforme comme Twitch. Pour de courtes vidéos en extérieur, l'ensemble doit bien sûr être compact, économique en énergie et raisonnablement bon marché. Vous en saurez plus bientôt sur www.elektormagazine.fr.

Restez à l'écoute et en pleine forme !

Elektor a 60 ans : joignez-vous à la fête !

Elektor célébrera ses 60 ans de publications et d'innovations dans le monde de l'électronique avec quelques projets spéciaux passionnants. Nous travaillons sur un livre anniversaire, un film, un événement en direct (*World Ethical Electronics Forum*) et le « labo domestique mobile d'Elektor ». Nous sommes certains que cela vous intéresse. Restez à l'écoute, nous vous donnerons plus d'informations dans les semaines à venir. Vous avez des idées pour des projets spéciaux ? Envoyez-moi vos idées : denise.bodrone@elektor.com

Denise Bodrone, coordinatrice du comité « Elektor 60 »



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 ».

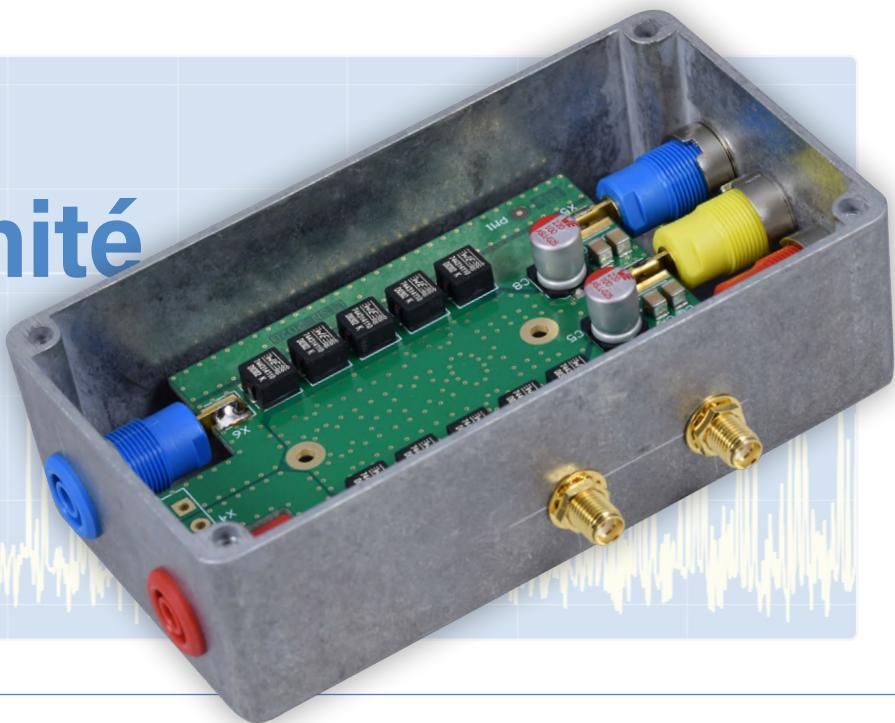


Elektor est membre de VDZ (association d'éditeurs de magazines allemands) qui « représente les intérêts communs de 500 éditeurs allemands grand public et B2B. »

Test de préconformité CEM

RSIL double continu

16

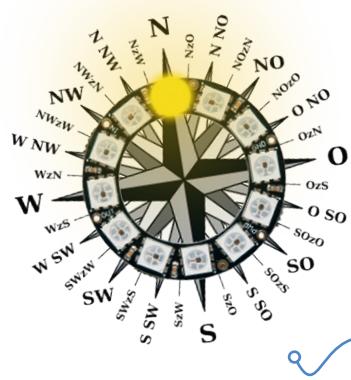


Rubriques

- 3 **Édito**
- 27 **Démarer en électronique... (9)**
N'ayez pas peur du choc... un peu de « self control » suffit !
- 73 **Dans l'antre de...**
Menno van der Veen et de ses tubes amplificateurs
- 76 **Drôles de composants**
Circuits hybrides
- 82 **Questions d'éthique**
Mesurez votre empreinte carbone
- 110 **Sur le vif**
Passions incertaines
- 114 **Hexadoku**
Casse-tête pour elektorniciens

Articles de fond

- 6 **Voyage dans les réseaux neuronaux (1^{ère} partie)**
Les neurones artificiels
- 30 **Traitement d'images avec le module Jetson Nano de Nvidia**
1^{ère} partie : matériel et logiciel
- 34 **Zoom sur les transformateurs secteur**
Comportement transitoire à l'allumage et l'extinction
- 41 **« Yes We CAN » avec PiCAN 3**
Module de bus CAN pour le Raspberry Pi 4
- 44 **Centrale solaire sur balcon**
Un investissement vite amorti
- 65 **Propeller 2 de Parallax (4)**
Envoi de chaînes de caractères
- 68 **60 ans d'Elektor**
Regard sur septembre
- 95 **Ultimate Arduino Uno Hardware Manual**
Extrait : chargeur d'amorçage du microcontrôleur principal
- 107 **Carte MadMachine SwiftIO**
Un langage récent pour du matériel tout nouveau



Rose de vents avec
le capteur GY-271

78

Zoom sur les transformateurs secteur



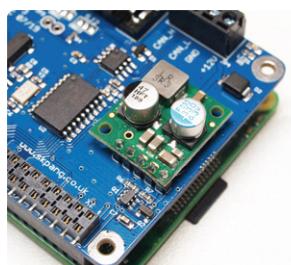
34

Lévitation magnétique version numérique



Réalisations

- 16 Test de préconformité CEM pour un projet alimenté en courant continu**
Partie 1 : RSIL double continu
- 20 Charge électronique en CC et CA**
Jusqu'à 400 V et 10 A (crête)
- 50 Prise de vue et diffusion vidéo en continu**
La caméra Raspberry Pi HQ en action
- 58 Utilisation d'écrans dans les projets Raspberry Pi**
Extrait : écrans à diodes électroluminescentes organiques (OLED)
- 78 Rose des vents avec le capteur GY-271**
Ou pourquoi il faut faire des huit pour calibrer un capteur
- 84 Thermostat connecté à ESP32**
Conservez votre vin à la bonne température !
- 90 Lévitation magnétique version numérique**
Comparateur analogique remplacé par un ESP32 Pico
- 100 MicroPython pour l'ESP32 et ses copains**
Partie 2 : piloter facilement les écrans matriciels



Module de bus CAN pour le Raspberry Pi 4

41

Bientôt dans ces pages

Le numéro de novembre-décembre 2021 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.

Quelques-uns des points forts :

- Pi-KVM : contrôle à distance d'autres ordinateurs
- Traceur GPS avec logiciel à code source ouvert
- Synthétiseur de musique avec bruit
- Mémoire analogique avec une ampoule au magnésium
- Kit de mesure du CO₂

et bien d'avantage !

Ce numéro paraîtra le 5 novembre 2021.

elektor
créer ➤ partager ➤ vendre

voyage dans les réseaux neuronaux (1ère partie)

Les neurones artificiels

Stuart Cording (Elektor)

L'intelligence artificielle (IA) et l'apprentissage automatique (ML) révolutionnent l'industrie électronique. Dans cette série d'articles, nous étudierons un élément central : le réseau neuronal. Ce premier article concerne notamment la recherche sur les neurones artificiels et la mise en œuvre d'un perceptron multicouche (MLP) à l'aide d'un logiciel.



L'intelligence artificielle (IA) et l'apprentissage automatique (*Machine Learning*) sont deux des sujets les plus brûlants de l'industrie électronique et informatique. Grâce aux succès de l'IA qui font la une des médias, lorsqu'elle réussit à battre les meilleurs joueurs de Go du monde [1], ou en dépit de ses échecs, avec les accidents de véhicules sans conducteurs [2], l'intelligence artificielle fait désormais partie de notre univers quotidien. Si le ML et l'IA se sont démocratisés grâce à des outils fonctionnant dans le nuage comme TensorFlow, ces plateformes aussi énormes que puissantes peuvent sembler intangibles si vous essayez de comprendre comment cela fonctionne de manière très pratique. Dans cette série consacrée aux réseaux neuronaux, nous revenons à l'essentiel en explorant la plupart des éléments de base de ces systèmes. Au fur et à mesure de notre progression, vous trouverez de nombreux exemples à essayer, des projets de ML parfois sympas, parfois obscurs, et à la fin nous doterons même un Arduino d'un cerveau. Plongeons maintenant dans le sujet passionnant des réseaux neuronaux.

Défis informatiques

L'IA et le ML sont « les » défis informatiques de notre époque. L'IA a pour finalité d'utiliser des ordinateurs pour imiter l'intelligence humaine. Le ML sert, pour sa part, à reconnaître des formes ou des modèles constitués de données structurées et semi-structurées. L'un et l'autre nécessitent de gros investissements annuels dans des projets de recherche et des développements technologiques de semi-conducteurs et de plateformes informatiques. Et, grâce au « nuage », la technologie est facilement accessible à ceux qui veulent explorer et tester leurs idées.

Mais en quoi consistent les réseaux neuronaux ? Comment fonctionnent tous ces algorithmes intelligents ? Comment apprennent-ils ? Quelles sont leurs limites ? Et est-il possible de s'amuser avec les rouages du ML sans s'inscrire à un énième service en nuage ? Ce sont précisément ces questions que cette courte série sur les réseaux neuronaux va aborder, en quatre parties :

- 1^{ère} partie — Neurones artificiels : dans cet article, nous commençons par remonter aux années 1950 pour examiner les premières recherches destinées à développer un neurone artificiel. De là, nous passerons rapidement à la mise en œuvre d'un logiciel de perceptron multicouche (**MLP**, *MultiLayer Perceptron*) qui utilise la rétropropagation pour « apprendre ».
- 2^e partie — Neurones logiques : l'un des problèmes des premiers neurones était leur incapacité à résoudre la fonction XOR (OU exclusif). Nous examinerons si notre MLP peut résoudre ce problème et visualiserons comment le neurone apprend.
- 3^e partie — Neurones pratiques : nous appliquerons notre MLP à une partie du problème de la conduite autonome. Il s'agira de reconnaître l'état des feux de signalisation à l'aide d'un programme sur PC.
- 4^e partie — Neurones embarqués : nous terminerons cette série en nous focalisant sur ce type de neurone. Si cela fonctionne sur un PC, cela devrait fonctionner sur un microcontrôleur, non ? Grâce à un Arduino et à un capteur RVB, nous déterminerons à nouveau les couleurs des feux de signalisation.

La petite histoire des neurones

Les premières tentatives de création d'un neurone numérique ou artificiel se sont inspirées de la nature. Un neurone biologique reçoit des entrées par ses dendrites et transmet la sortie résultante par son axone vers les terminaisons axonales (**fig. 1**). La décision d'émettre ou non un stimulus en sortie, connue sous le nom de décharge du neurone, est prise par un processus appelé « activation ». Si les entrées sont conformes à un modèle appris, le neurone s'active. Dans le cas contraire, il ne le fait pas. On peut rapidement constater qu'avec des chaînes de neurones biologiques interconnectés, il est possible de reconnaître des modèles très complexes.

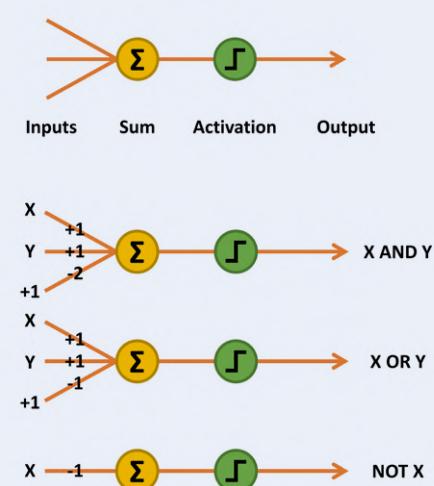
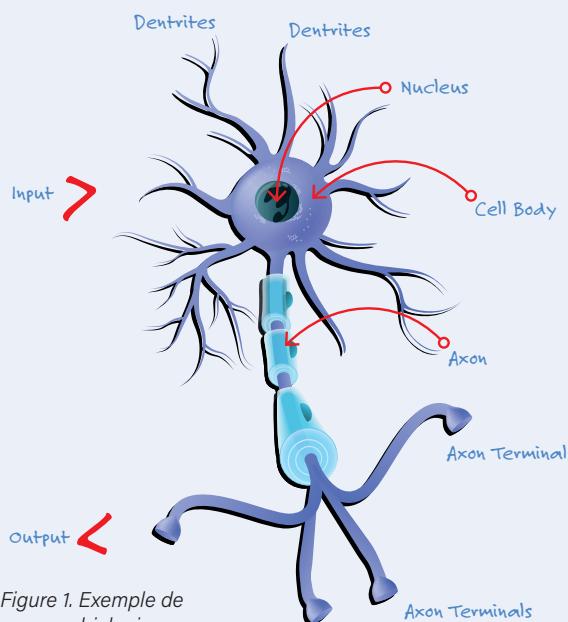


Figure 2. Un réseau McCulloch-Pitts additionne les entrées multipliées par leurs poids, en activant une sortie 1 si le résultat est égal ou supérieur à 0.

Les premiers neurones artificiels mis au point étaient des réseaux McCulloch-Pitts, également connus sous le nom d'unités logiques de seuil (TLU, *Threshold Logic Unit*). Il s'agissait de simples « machines décisionnelles » capables de reproduire les fonctions de portes logiques. Elles n'acceptaient et ne produisaient que des valeurs logiques 0 et 1. Pour mettre en œuvre leurs capacités de reconnaissance de formes, des valeurs de pondération devaient être déterminées mathématiquement ou heuristiquement pour chaque entrée. Parfois, une entrée supplémentaire était également nécessaire (fig. 2, fonctions ET et OU). Les entrées du réseau sont simplement multipliées par leurs poids et additionnées. La décision de produire un 1 en sortie, ou d'activer le neurone, est mise en œuvre à l'aide d'une unité de seuil linéaire. Si le résultat est égal ou supérieur à 0, un 1 est émis. Sinon, la sortie est 0 (tableau 1).

Entrées			Poids			Sortie pour X ET Y	
X	Y	+1	X	Y	+1	Σ (entrée \times poids)	Sortie
0	0	+1	+1	+1	-2	-2	0
0	1	+1				-1	0
1	0	+1				-1	0
1	1	+1				0	1

Tableau 1. Réseau McCulloch-Pitt pour mettre en œuvre une fonction ET.

Perceptrons

L'étape suivante du développement a eu lieu dans les années 1950 avec les travaux du psychologue Frank Rosenblatt [3]. Son perceptron conservait les entrées binaires et la prise de décision par l'unité de seuil linéaire de la TLU McCulloch-Pitts. La sortie était également une valeur binaire 0 ou 1. Ce perceptron se distinguait cependant de deux manières : le niveau de seuil (appelé θ ta, Θ) pour décider de la valeur de sortie était ajustable, et il acceptait une forme limitée d'apprentissage (fig. 3).

Le processus d'apprentissage fonctionne ainsi : le perceptron ne produit une valeur 1 que si la somme du produit des entrées et des poids est supérieure à Θ . Si la sortie par rapport à la combinaison des entrées est correcte, rien n'est modifié.

Si une valeur 1 sort alors qu'un 0 est requis, le niveau de seuil Θ est augmenté de 1. De plus, tous les poids associés aux entrées 1 sont réduits de 1. Si l'inverse se produit, c'est-à-dire si une valeur 0 est émise alors qu'un 1 est requis, tous les poids associés aux entrées 1 sont augmentés de 1.

Le raisonnement qui sous-tend ce processus est que seules les entrées ayant une valeur 1 peuvent contribuer à une sortie 1 non désirée. Il est donc logique de réduire leur impact en diminuant les poids correspondants. Inversement, seules les entrées ayant une valeur 1 peuvent contribuer à la sortie 1 souhaitée. Si la sortie est 0, et non 1 comme souhaité, les poids associés doivent être augmentés.

En 1958, le « perceptron Mark I » a été construit sous la forme d'un dispositif matériel, après avoir été implémenté à l'aide d'un logiciel sur un IBM 704 [4]. Connecté à 400 cellules photoélectriques au sulfure de cadmium formant une caméra rudimentaire et utilisant des moteurs connectés à des potentiomètres pour mettre à jour les poids pendant l'apprentissage, il pouvait, une fois entraîné, reconnaître la forme « triangle » [5].

Le problème des perceptrons

Bien que la démarche ait ouvert une nouvelle ère où un système électronique pouvait potentiellement apprendre, un problème essentiel se posait : le dispositif ne pouvait résoudre que des problèmes linéairement séparables. Pour en revenir à l'ancienne TLU McCulloch-Pitts, les fonctions simples ET, OU, NON, NON ET et NON OU sont toutes linéairement séparables. Cela signifie qu'une seule ligne peut séparer les sorties souhaitées (par rapport aux entrées) des sorties non souhaitées (fig. 4). Les fonctions OU exclusif (et la fonction complémentaire NON OU exclusif) sont différentes. Lorsque les entrées sont identiques (00 ou 11), la sortie est 0, mais lorsque les entrées sont différentes (01 ou 10), la sortie est 1. Il faut donc que la sortie souhaitée soit classée dans un groupe par rapport aux entrées. En d'autres termes, le perceptron ne peut pas être entraîné à apprendre comment fonctionne un OU exclusif ou un NON OU exclusif ou à reproduire sa fonction.

L'autre problème essentiel réside dans la fonction d'activation utilisée. L'unité de seuil linéaire pouvait effectuer un saut brusque entre l'état inactif et l'état actif. Les recherches qui ont abouti au

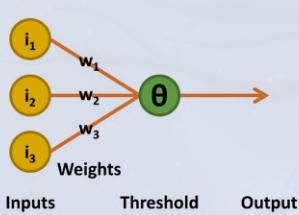


Figure 3. Le perceptron a introduit un niveau de seuil variable Θ et utilisait un processus d'apprentissage itératif.

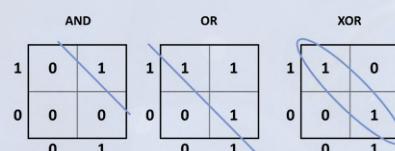


Figure 4. De gauche à droite, les fonctions ET, OU et OU exclusif. Les sorties 1 des deux premiers peuvent être séparées linéairement, à l'aide d'une ligne droite, des états de sortie 0. Pour le OU exclusif, les états de sortie 1 ne peuvent pas être séparés linéairement des états de sortie 0. C'est quelque chose qu'un perceptron ne peut pas apprendre.

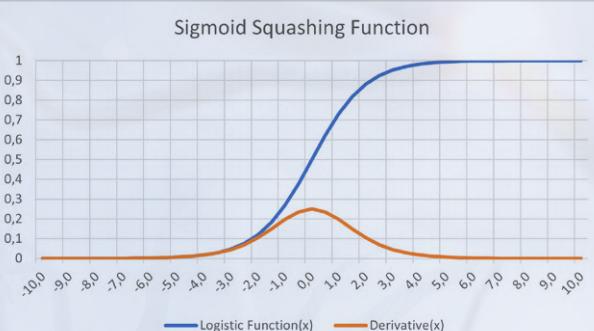


Figure 5. La fonction logistique délivre rapidement des valeurs d'entrée à 0 ou 1, ce qui facilite l'apprentissage par descente de gradient dans les réseaux neuronaux.

réseau de type « règle delta » [6] ont montré que la descente de gradient était un élément crucial du processus d'apprentissage des réseaux neuronaux. Cela signifiait également que toute fonction d'activation devait être différentiable. Le saut brusque de 0 à 1 dans l'unité de seuil linéaire n'est pas différentiable au point de transition (la pente devient infinie), et le reste de la fonction délivre simplement l'état 0 (la sortie reste inchangée).

Il a été proposé qu'un réseau multicouche comportant un ou plusieurs nœuds cachés entre les nœuds d'entrée et de sortie permettrait de résoudre le problème du OU exclusif. De plus, une fonction différentiable, telle que la fonction logistique (fig. 5), une courbe sigmoïde, pourrait constituer une fonction d'activation progressive acceptant l'apprentissage par descente de gradient. Le principal problème résidait dans l'apprentissage : comment effectuer l'entraînement pour tous les poids ?

L'approche de la « règle delta » avait montré qu'en calculant l'erreur quadratique du réseau (sortie souhaitée – sortie réelle) et qu'en mettant en œuvre un taux d'apprentissage, les poids du réseau pouvaient être successivement améliorés jusqu'à ce que l'ensemble optimal de vecteurs de poids ait été trouvé (fig. 6). L'ajout d'une couche de nœuds cachés entre l'entrée et la sortie a rendu le calcul plus compliqué, mais pas impossible, comme nous le verrons.

Perceptron multicouche

C'est l'ajout de la couche cachée qui a permis de créer le perceptron multicouche (MLP). La forme la plus simple de réseau neuronal MLP utilise une seule couche cachée. Tous les nœuds sont reliés (autrement dit « entièrement connectés ») et des poids sont attribués entre chaque nœud d'entrée et chaque nœud caché, ainsi qu'entre chaque nœud caché et chaque nœud de sortie (fig. 7). Les lignes entre les nœuds représentent les poids. Les entrées souhaitées sont appliquées aux nœuds d'entrée (valeurs comprises entre 0,0 et 1,0), et le réseau calcule la réponse de chaque nœud caché et de chaque nœud de sortie – une étape connue sous le nom de phase de **propagation avant**. Le résultat obtenu doit indiquer que les valeurs d'entrée correspondent à une catégorie de sortie que le réseau a apprise.

À titre d'exemple, les entrées pourraient être liées à une caméra de 28×28 pixels pointée sur des chiffres manuscrits. La base de

données MNIST de chiffres manuscrits contenant uniquement ces tailles d'images pourrait constituer l'ensemble d'apprentissage [7]. Chacune des sorties représenterait l'un des chiffres de 0 à 9. Le chiffre 7 étant tenu devant la caméra, chaque sortie indiquerait la probabilité que l'entrée ait cette valeur. La sortie 0 indiquerait, avec un peu de chance, qu'il est peu probable que le chiffre écrit à la main soit un 0, tout comme les huit autres sorties. Mais la sortie 7 devrait indiquer une forte probabilité que le nombre manuscrit soit un 7. Après entraînement, cette sortie devrait fonctionner comme prévu pour un 7 provenant des données d'entraînement ou écrit à la main et reconnaissable par un humain.

Avant cela, le réseau doit apprendre la tâche à accomplir. Pour ce faire, on applique l'entrée (des 7 écrits à la main) et on analyse les résultats fournis par les sorties. Comme il est peu probable qu'ils soient corrects dès la phase initiale, un cycle d'apprentissage est exécuté pour modifier les poids afin de réduire l'erreur. Ce processus itératif, appelé **rétropropagation**, est exécuté plusieurs milliers de fois jusqu'à ce que la précision du réseau réponde aux exigences de l'application. Dans l'univers de l'apprentissage automatique (ML), ce processus est appelé « apprentissage supervisé ».

Il existe deux autres facteurs importants à prendre en compte durant les phases de propagation avant et de rétropropagation. Le premier est le phénomène du **biais**. Une valeur de biais de 1,0 multipliée par un poids (entre 0,0 et 1,0) est appliquée aux nœuds des couches cachées et de sortie pendant la phase de propagation avant. Son rôle est d'améliorer la capacité du réseau à résoudre les problèmes et, en substance, de solliciter la fonction d'activation logistique (voir fig. 5) vers la gauche ou la droite. L'autre valeur est le **taux d'apprentissage**, là encore une valeur comprise entre 0,0 et 1,0. Comme son nom l'indique, cette valeur détermine la vitesse à laquelle le MLP apprend à résoudre le problème donné. Elle est également appelée « vitesse de convergence ». Trop basse, le réseau risque de ne jamais résoudre le problème avec un niveau de précision suffisant. Et si elle est trop élevée, le réseau risque d'osciller pendant l'apprentissage et de ne pas donner un résultat suffisamment précis (voir aussi l'encadré « Limites de l'apprentissage par gradient »).

Le MLP décrit ici pourrait être considéré comme une version standard. Mais la mise en œuvre des réseaux neuronaux est possible

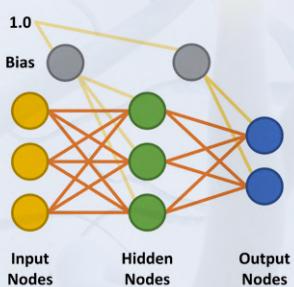


Figure 6. Exemple de MLP avec trois nœuds d'entrée, trois nœuds cachés et deux nœuds de sortie. Les nœuds cachés et de sortie peuvent utiliser une courbe sigmoïde pour déterminer leur sortie au cours de la phase de propagation avant.

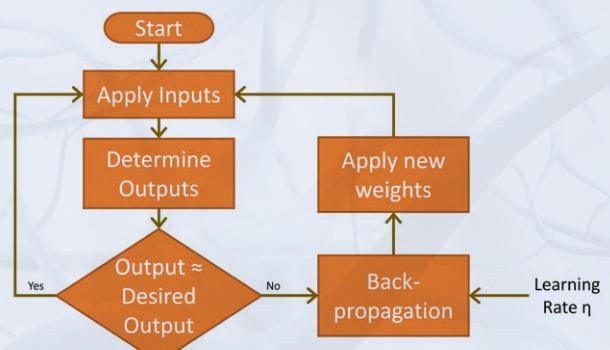


Figure 7. Organigramme montrant comment est mis en œuvre l'apprentissage par réseau neuronal.

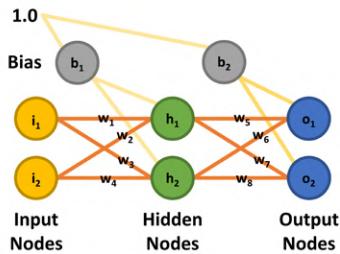


Figure 8. Configuration du MLP utilisée pour les exemples de calculs.

de multiples façons. Ils peuvent notamment posséder plusieurs couches cachées, ne pas connecter complètement les nœuds, relier les nœuds ultérieurs aux nœuds antérieurs et utiliser différentes fonctions d'activation [8].

Perceptron multicouche (MLP) en action

Le principe étant clair, nous l'espérons, passons à un exemple concret. Il fait suite à un excellent article de Matt Mazur, qui a pris le temps d'expliquer la rétropropagation dans un MLP à une seule couche cachée [9]. L'analyse menée ici sera à un niveau élargi, mais les personnes intéressées (et qui n'ont pas peur des mathématiques) peuvent étudier l'article de Matt pour plus de détails. Une fois le principe mathématique de fonctionnement abordé, nous examinerons la mise en œuvre du MLP sous forme de logiciel, avec les mêmes paramètres que ceux utilisés pour cette analyse théorique. Pour aller plus loin, il existe une feuille de calcul Excel qui correspond à l'exemple traité par Matt. Il suffit de télécharger ou de cloner le référentiel GitHub [10] et d'ouvrir le fichier *workedexample/Matt Mazur Example.xlsx* dans le dossier.

Pour simplifier les choses, nous utilisons un MLP à deux entrées et deux sorties avec deux nœuds cachés. Les nœuds d'entrée sont appelés i_1 et i_2 , les nœuds cachés h_1 et h_2 , et les nœuds de sortie o_1 et o_2 . Pour cet exercice, l'objectif est d'entraîner le réseau à produire 0,01 sur le nœud o_1 et 0,99 sur o_2 lorsque le nœud i_1 est à 0,05 et i_2 est à 0,10. Il y a huit poids (w_1 à w_8) et deux valeurs de biais (b_1 et b_2). Pour que les calculs puissent être reproduits, tous les nœuds d'entrée, les biais et les poids se voient attribuer les valeurs indiquées à la **figure 8** et correspondent à l'article de Matt Mazur.

Propagation avant

La phase de propagation avant pour calculer les sorties o_1 et o_2 fonctionne comme suit. Chaque nœud caché reçoit une entrée qui est la somme nette des entrées multipliées par les poids, plus l'entrée de biais ($b_1 = 0,35$). En utilisant le processus et les équations fournies par Matt Mazur, l'entrée de h_1 est la somme de i_1 multipliée par w_1 (0,15), i_2 multipliée par w_2 (0,20) et b_1 (0,35).

$$(input to node) net_{h1} = w_1 \times i_1 + w_2 \times i_2 + b_1 \times 1 \\ 0.15 \times 0.05 + 0.20 \times 0.10 + 0.35 \times 1 = 0.3775$$

La valeur de sortie de chaque nœud caché est déterminée par la fonction (d'activation) logistique, qui écrase la sortie du nœud caché entre 0,0 et 1,0. Cette valeur est calculée comme suit :

$$(output of node) out_{h1} = \frac{1}{1 + e^{-net_{h1}}} \\ = 0.593269992$$

En répétant le processus pour h_2 (avec $w_3 = 0,25$, $w_4 = 0,3$ et $b_2 = 0,35$), on obtient :

$$(output of node) out_{h2} = 0.596884378$$

Les entrées nettes des nœuds de sortie sont calculées de la même manière, en utilisant les valeurs de sortie calculées pour les nœuds cachés et en appliquant les poids 5 à 8 ($w_5 = 0,4$, $w_6 = 0,45$, $w_7 = 0,5$ et $w_8 = 0,55$) et la valeur du biais $b_2 = 0,60$.

$$(output of node) out_{o1} = 0.75136507$$

$$(output of node) out_{o2} = 0.772928465$$

Comme nous l'avons déjà dit, notre objectif est d'obtenir une sortie de 0,01 pour o_1 et de 0,99 pour o_2 , mais nous pouvons constater que nous sommes assez loin de ce résultat. L'étape suivante consiste à calculer l'erreur de chaque sortie en utilisant la fonction d'erreur quadratique, ainsi que l'erreur totale, comme suit :

$$E_{total} = \sum \frac{1}{2} (target - output)^2$$

Celle-ci est calculée pour chaque sortie comme suit :

$$E_{o1} = \frac{1}{2} (0.01 - 0.75136507)^2 = 0.274811083$$

$$E_{o2} = \frac{1}{2} (0.05 - 0.772928465)^2 = 0.023560026$$

Enfin, l'erreur totale du réseau peut être établie :

$$E_{total} = E_{o1} + E_{o2} = 0.274811083 + 0.023560026 = 0.298371109$$

L'étape suivante consiste à déterminer comment améliorer cette erreur.

Rétropropagation

La rétropropagation constitue le cœur de l'apprentissage. Le processus consiste ici à déterminer la contribution de chaque poids à l'erreur totale. Il commence par examiner les poids entre les nœuds cachés et les nœuds de sortie. Ce qui complique les choses, c'est que w_5 contribue à l'erreur totale par le biais de deux nœuds de sortie, o_1 et o_2 , qui sont également influencés par les poids w_6 à w_8 . Il convient également de noter que les valeurs de biais ne jouent aucun rôle dans ces calculs.

Les mathématiques requises pour définir ceci sont assez complexes, mais elles se limitent à quelques simples opérations de multiplication, d'addition et de soustraction. Le calcul de la nouvelle valeur de w_5 en tenant compte du taux d'apprentissage choisi, η (0,5), s'effectue comme suit :

$$w_5^+ = w_5 - \eta * \frac{\delta E_{total}}{\delta w_5}$$

À partir de là, nous pouvons revoir les anciens poids et les comparer avec les nouveaux poids pour w_5 à w_8 :

$$w_5 = 0.40 \quad w_5^+ = 0.35891648$$

$$w_6 = 0.45 \quad w_6^+ = 0.408666186$$

$$w_7 = 0.50 \quad w_7^+ = 0.511301270$$

$$w_8 = 0.55 \quad w_8^+ = 0.561370121$$

Un apprentissage nonchalant

Si l'apprentissage est difficile pour les humains, il semble l'être aussi pour l'IA. Dans le cadre de recherches menées par l'université technique de Berlin, l'Institut Fraunhofer pour les télécommunications et l'université de technologie et de design de Singapour (SUTD), les systèmes d'IA ont dû justifier leurs stratégies de prise de décision [13]. Les résultats ont été révélateurs. Si les systèmes d'intelligence artificielle ont tous accompli leur tâche de manière admirable, ils ont fait preuve d'un culot époustouflant dans leur prise de décision. Ainsi,

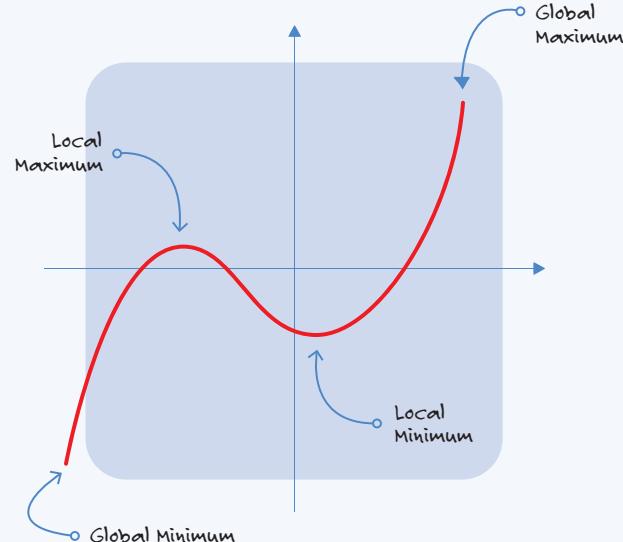
un algorithme a correctement déterminé la présence d'un navire sur une image, mais en fondant sa décision sur le fait que de l'eau apparaissait sur la photo. Un autre a détecté à juste titre que les images contenaient des chevaux à partir de la présence d'une indication de copyright, et non grâce aux caractéristiques visuelles de l'animal.



Graphic: cjmacer / Shutterstock.com

Limites de l'apprentissage par gradient

Il arrive qu'un réseau neuronal soit apparemment incapable d'apprendre, bien qu'il ait précédemment assimilé la fonction requise avec la même configuration de nœuds. Cette situation peut être due au fait que le réseau reste bloqué dans un « minimum local » au lieu de trouver un « minimum global » de la fonction d'erreur.



En effectuant un rapide contrôle de vraisemblance, nous pouvons constater que cela fonctionne. Nous voulons que o_1 soit diminué vers 0,01 par w_5 et w_6 , et que o_2 soit augmenté vers 0,99 par w_7 et w_8 . L'étape finale consiste à déterminer la contribution sur l'erreur en sortie des poids entre les entrées et les nœuds cachés. Comme précédemment, les mathématiques se réduisent à de simples opérations de multiplication, d'addition et de soustraction. En fait, l'équation est la même que celle utilisée pour calculer les nouveaux poids w_5 à w_8 . Ce qui est différent, c'est la façon dont est calculée l'erreur totale par rapport au poids (initialement w_1), car la sortie de la couche cachée dépend d'une autre entrée et d'un autre poids (pour h_1 , i_1 et w_1 ainsi que i_2 et w_2) :

$$w_1^+ = w_1 - \eta * \frac{\delta E_{total}}{\delta w_1}$$

Encore une fois, la valeur de biais ne joue aucun rôle dans ce calcul. Nous pouvons maintenant comparer les anciens poids w_1 à w_4 et les nouveaux :

$$w_1 = 0.15 \quad w_1^+ = 0.149780716$$

$$w_2 = 0.20 \quad w_2^+ = 0.19956143$$

$$w_3 = 0.25 \quad w_3^+ = 0.24975114$$

$$w_4 = 0.30 \quad w_4^+ = 0.29950229$$

Une fois les nouveaux poids déterminés, il est possible de remplacer les anciens et d'effectuer une nouvelle phase de propagation vers l'avant. Tant que l'erreur de sortie reste supérieure à la valeur souhaitée, les phases de rétropropagation peuvent être répétées comme décrit ici.

Mise en œuvre du MLP dans l'environnement Processing

Pour démontrer ce réseau neuronal simple, il a été codé de A à Z sous la forme d'une classe utilisable dans Processing [11], l'environnement de développement conçu pour promouvoir le codage dans les arts visuels. Les capacités visuelles de l'EDI permettent d'afficher aisément des graphiques 2D et 3D. Une console de sortie texte permet également de tester les idées rapidement et facilement. Le code ci-après figure dans le dépôt [10].

Le code pour mettre en œuvre le MLP se trouve dans le dossier *processing/neural/neural.pde*. Il suffit d'ajouter ce fichier à tout projet Processing qui doit l'utiliser. La classe *Neural* peut être instanciée pour prendre en charge un nombre quelconque de nœuds d'entrée, cachés et de sortie. Pour reproduire l'exemple déjà traité, le fichier *processing/nntest/nntest.pde* doit être ouvert dans Processing.

La création d'un réseau neuronal est assez simple. Tout d'abord, le constructeur de classe *Neural* (dans *nntest.pde*) est appelé pour créer un objet, appelé ici *network*, en définissant le nombre souhaité d'entrées, de nœuds cachés et de sorties (2, 2 et 2). Une fois l'objet créé, d'autres fonctions membres sont appelées pour définir le taux d'apprentissage et les biais des nœuds cachés et des nœuds de sortie, comme dans l'exemple de croquis.

Le constructeur initialise également les poids avec des valeurs aléatoires comprises entre 0,25 et 0,75. Pour correspondre à l'exemple, nous modifions les poids comme indiqué dans le **listage 1** où les valeurs d'entrée et les valeurs de sortie souhaitées sont également définies.

L'exemple de code active également un mode « verbeux » qui permet d'afficher le travail pour chaque étape des calculs. Ceux-ci doivent correspondre aux résultats affichés dans la feuille de calcul Excel. Après activation de **Run** d'un simple clic dans Processing, la console en mode texte doit afficher le résultat suivant :

```
...forwardpass complete. Results:  
For i1 = 0.05 and i2 = 0.1  
o1 = 0.75136507 (but we want: 0.01)  
o2 = 0.7729285 (but we want: 0.99)  
Total network error is: 0.2983711
```

Après cela, l'apprentissage est activé et une phase vers l'avant est à nouveau exécutée, suivie de l'étape de rétropropagation. Il en résulte la sortie des poids nouveaux et anciens ainsi que le calcul d'une nouvelle phase vers l'avant pour déterminer les nouvelles valeurs et les erreurs des noeuds de sortie :

```
New Hidden-To-Output Weight [ 0 ][ 0 ] = 0.3589165,  
Old Weight = 0.4  
New Hidden-To-Output Weight [ 1 ][ 0 ] = 0.40866616,  
Old Weight = 0.45  
New Hidden-To-Output Weight [ 0 ][ 1 ] = 0.5113013,  
Old Weight = 0.5  
New Hidden-To-Output Weight [ 1 ][ 1 ] = 0.56137013,  
Old Weight = 0.55  
New Input-To-Hidden Weight[ 0 ][ 0 ] = 0.14978072,  
Old Weight = 0.15  
New Input-To-Hidden Weight[ 1 ][ 0 ] = 0.19956143,  
Old Weight = 0.2  
New Input-To-Hidden Weight[ 0 ][ 1 ] = 0.24975115,  
Old Weight = 0.25  
New Input-To-Hidden Weight[ 1 ][ 1 ] = 0.2995023,  
Old Weight = 0.3
```

Nous pouvons voir que les valeurs représentent les poids 5 à 8, puis 1 à 4. Elles correspondent également aux calculs effectués à la main précédemment (à quelques petites exceptions près dues à des erreurs d'arrondi).

MLP et réseaux neuronaux : et ensuite ?

Vous avez envie d'en savoir plus sur les réseaux neuronaux ? Maintenant que vous comprenez bien le fonctionnement d'un réseau neuronal MLP et de l'exemple de classe **Neural** fourni ici pour Processing, vous êtes en position idéale pour entreprendre d'autres expériences en toute indépendance. Voici quelques suggestions :

- Exécuter l'apprentissage en boucle — Combien de périodes sont nécessaires pour obtenir une erreur de réseau de 0,001, 0,0005 ou 0,0001 ?
- Commencer avec différents biais et poids — Quel est l'impact sur le nombre de périodes nécessaires à l'apprentissage ? Le réseau neuronal échoue-t-il parfois dans un apprentissage ?
- Mettre les sorties en correspondance avec les entrées — Essayez de générer un graphique 3D de chaque sortie par rapport aux entrées une fois que le réseau a appris ce qu'il a à faire. Le résultat est-il conforme à vos attentes ? Peut-être préférerez-vous essayer Chart-Studio de Plotly plutôt que d'utiliser une feuille de calcul [12] pour tracer les données de sortie.

Dans le prochain article de cette série, nous apprendrons à notre réseau neuronal à mettre en œuvre des portes logiques et à visualiser son processus d'apprentissage. ↗

(210160-04)

Des questions, des commentaires ?

Envoyez un courriel à l'auteur (stuart.cording@elektor.com)

Contributeurs

Idée, texte et illustrations : **Stuart Cording**

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

Illustrations : **Patrick Wielders**

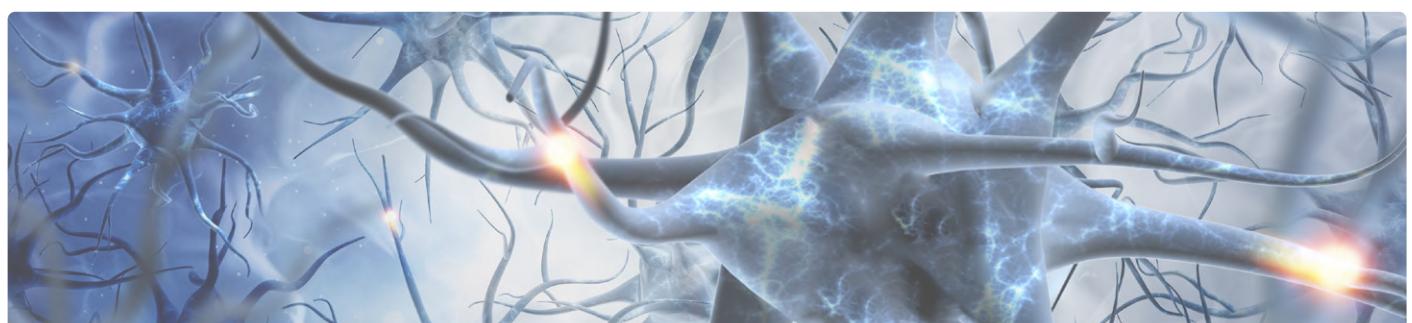
Mise en page : **Harman Heida**

Traduction : **Pascal Godart**



Produits

- B. van Dam, « **Artificial Intelligence** » (livre numérique, en anglais) www.elektor.fr/artificial-intelligence-e-book
- Kit Google AIY Vision pour Raspberry Pi www.elektor.fr/google-aiy-vision-kit-for-raspberry-pi
- Caméra IA de Huskylens avec boîtier en silicone www.elektor.fr/huskylens-ai-camera-with-silicone-case





Listage 1. Utilisation de la classe Neural du MLP dans nn_test.pde.

```
Neural network;

void setup() {
    // This neural network uses two inputs, two hidden nodes, and two output nodes.
    network = new Neural(2,2,2);

    // Set learning rate here
    network.setLearningRate(0.5);

    // Set Neural class to be 'verbose' so we can see what it is doing
    network.turnVerboseOn();

    println("Matt Mazur Neural Network Backpropagation Example");
    println("-----");
    println();
    println("Structure is:");
    println("Input nodes: ", network.getNoOfInputNodes(), "; Hidden nodes: ", network.getNoOfHiddenNodes(),
    "; Output nodes: ", network.getNoOfOutputNodes());

    // Set network biasing here
    // b1 = 0.35
    network.setBiasInputToHidden(0.35);
    // b2 = 0.60
    network.setBiasHiddenToOutput(0.6);

    // The Neural class constructor gives the weights random value.
    // Here we use the values from Matt Mazur's example.
    // We start with the input-to-hidden weights:
    // w1 = 0.15 (i1 to h1)
    network.setInputToHiddenWeight(0, 0, 0.15);
    // w3 = 0.25 (i1 to h2)
    network.setInputToHiddenWeight(0, 1, 0.25);
    // w2 = 0.20 (i2 to h1)
    network.setInputToHiddenWeight(1, 0, 0.2);
    // w4 = 0.30 (i2 to h2)
    network.setInputToHiddenWeight(1, 1, 0.3);

    // Next we configure the hidden-to-output weights:
    // w5 = 0.40 (h1 to o1)
    network.setHiddenToOutputWeight(0, 0, 0.4);
    // w7 = 0.50 (h1 to o2)
    network.setHiddenToOutputWeight(0, 1, 0.5);
    // w6 = 0.45 (h2 to o1)
    network.setHiddenToOutputWeight(1, 0, 0.45);
    // w8 = 0.55 (h2 to o2)
    network.setHiddenToOutputWeight(1, 1, 0.55);

    // Configure the inputs
    // i1 = 0.05
    network.setInputNode(0, 0.05);
    // i2 = 0.10
    network.setInputNode(1, 0.1);

    // Now declare the values we like to achieve at the outputs for this
    // input combination
    // o1 should be 0.01
    network.setOutputNodeDesired(0, 0.01);
    // o2 should be 0.99
    network.setOutputNodeDesired(1, 0.99);

    // We now perform a forwardpass using the configured inputs, weights
    // and bias value. Verbose is on, so you will see the working
```



```
println();
println("Calculating values for o1 and o2...");
println();
network.calculateOutput();
println();

// Let's summarise the current state
println("...forwardpass complete. Results:");
println("For i1 = ", network.getInputNode(0), " and i2 = ", network.getInputNode(1));
println("o1 = ", network.getOutputNode(0), " (but we want: ", network.getOutputNodeDesired(0), ")");
println("o2 = ", network.getOutputNode(1), " (but we want: ", network.getOutputNodeDesired(1), ")");
println();
println("Total network error is: ", network.getTotalNetworkError());
println();

// Now we'll perform a learning cycle using backpropagation
// This enables a backpropagation cycle everytime the calculateOutput() method is called
network.turnLearningOn();

println("Learning is ON");
println("Calculating outputs again and performing backpropagation...");
println();
network.calculateOutput();

// We'll turn learning off again...
network.turnLearningOff();
// ...and run another forwardpass without verbose on:
network.turnVerboseOff();
network.calculateOutput();

println("...backpropagation complete. Results:");
println("For i1 = ", network.getInputNode(0), " and i2 = ", network.getInputNode(1));
println("o1 = ", network.getOutputNode(0), " (but we want: ", network.getOutputNodeDesired(0), ")");
println("o2 = ", network.getOutputNode(1), " (but we want: ", network.getOutputNodeDesired(1), ")");
println();
println("Total network error is: ", network.getTotalNetworkError());
println();
}

void draw() {
}
```

LIENS

- [1] M. Reynolds, « DeepMind's AI Beats World's Best Go Player in Latest Face-Off », *NewScientist*, 05/2017 : <http://bit.ly/3up6Xy3>
- [2] « Uber's self-driving operator charged over fatal crash », *BBC News*, 09/2020 : <http://bbc.in/3pJLQ5R>
- [3] Frank Rosenblatt : https://en.wikipedia.org/wiki/Frank_Rosenblatt
- [4] Perceptron : <https://en.wikipedia.org/wiki/Perceptron>
- [5] H. Mason, D. Stewart, and B. Gill, « Rival », *The New Yorker*, 11/1958 : <http://bit.ly/2NS4hrU>
- [6] I. Russell, « The Delta Rule », Université de Hartford : <http://bit.ly/3kgFt91>
- [7] Y. LeCun and C. Cortes, « The MNIST Database », Université de New York : <http://bit.ly/3kkdN39>
- [8] A. Tch, « The mostly complete chart of Neural Networks, explained », *Towards Data Science*, 08/2017 : <http://bit.ly/2ZJnQoY>
- [9] M. Mazur, « A Step by Step Backpropagation Example », 03/2015 : <http://bit.ly/2NxLNx5>
- [10] Dépôt GitHub - simple-neural-network : <http://bit.ly/2ZHLv9p>
- [11] EDI Processing : <https://processing.org/>
- [12] Plotly Chart-Studio : <https://chart-studio.plotly.com>
- [13] « Paper published at Nature Communications: Scientists put AI systems to the test », *Fraunhofer HHI*, 03/2019 : <http://bit.ly/3bycjH>

elektor 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

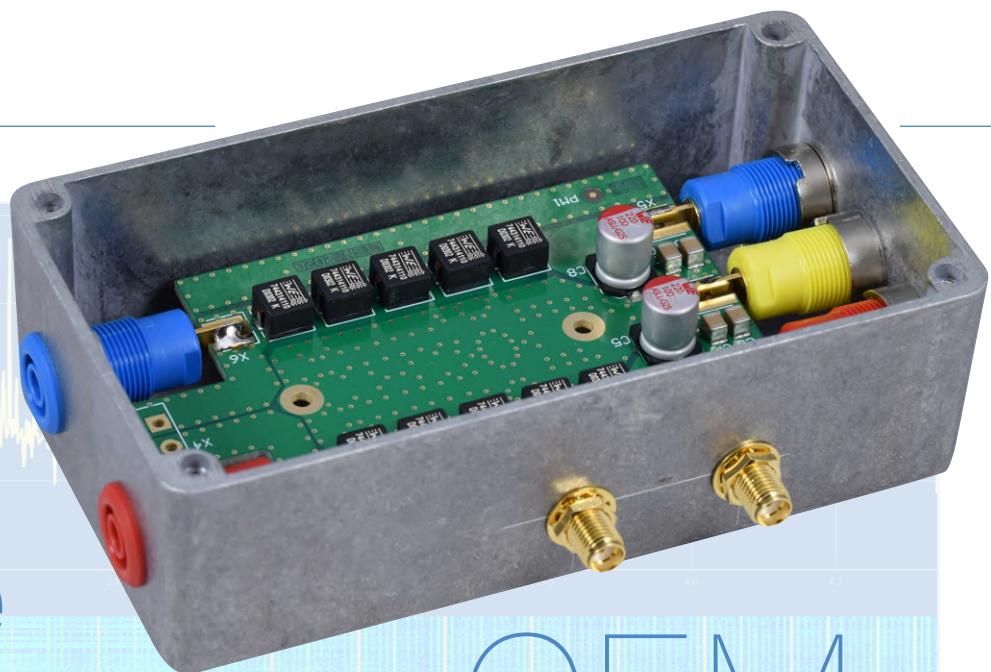
Chaque vendredi, vous recevrez les meilleurs articles et projets de la semaine. Nous couvrons les projets basés sur les MCU, l'IdO, la programmation, l'IA, et plus encore !

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.



test de préconformité CEM pour un projet alimenté en courant continu

Partie 1 : RSIL double continu

Ton Giesberts (Elektor) et **Robert Schillinger** (Würth Elektronik)

La mesure des émissions conduites est la méthode la plus simple et la moins chère pour savoir si un montage peut satisfaire aux exigences de CEM/IEM. Un réseau de stabilisation d'impédance de ligne (RSIL), tel que le projet que nous présentons ici en partenariat avec Würth Elektronik, en est un élément indispensable. Dans cet article, nous allons voir à quoi sert un RSIL. Dans la deuxième partie de cette série, nous verrons la construction du circuit et son utilisation dans un test de préconformité CEM.

Un réseau de stabilisation de l'impédance de ligne (RSIL) peut constituer un moyen relativement simple et peu coûteux de réaliser des tests d'émissions conduites sur les câbles d'alimentation d'appareils soumis à des tests de préconformité CEM/IEM. Utilisé pour des tests de préconformité dans un laboratoire ordinaire, un RSIL réduira très probablement les coûts des tests définitifs dans un laboratoire de certification, parce qu'il n'y aura que peu ou pas de modifications à apporter au circuit pour le rendre conforme à la directive CEM/IEM. Ces coûts étant élevés, tout problème résolu dans votre laboratoire peut vous faire gagner beaucoup de temps et d'argent, ainsi que réduire votre frustration.

Qu'est-ce qu'un RSIL ?

Le RSIL [1] (*Line Impedance Stabilization Network, LISN*) est connecté entre la source d'alimentation et le matériel à tester (MàT) (**fig. 1**) et fournit une impédance de ligne d'alimentation normalisée de 50Ω , rendant les tests indépendants de l'impédance physique de la source. De

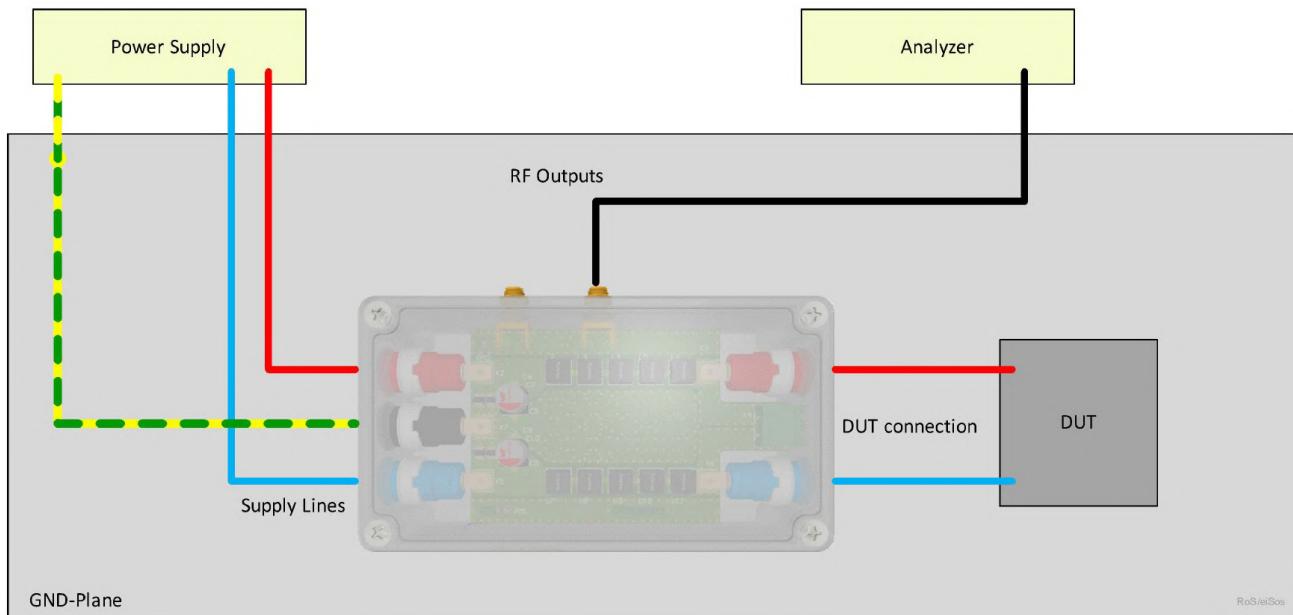


Figure 1. Circuit typique de test RSIL pour la mesure des émissions conduites (DUT = M&T).

plus, son filtre d'entrée passe-bas bloque la propagation vers le M&T les signaux RF (radio fréquences) provenant de la source d'alimentation et/ou du bruit sur les fils d'alimentation. Une troisième fonction du RSIL est d'offrir une ou plusieurs sorties 50 Ω connectables à un analyseur de spectre ou autre instrument de mesure pour évaluer les émissions conduites sur les câbles d'alimentation. Le plus souvent, ces sorties de mesure protègent les entrées de l'analyseur de spectre contre les transitoires et les surtensions. Tout bruit RF produit par le matériel à tester est isolé par le RSIL et transmis à l'analyseur de spectre pour mesure ou enregistrement.

Circuits RSIL

Le circuit le plus élémentaire pour le RSIL, comme recommandé par la FCC et le CISPR, est illustré sur la **figure 2**. Pour les normes militaires, une inductance de 50 µH est recommandée, une valeur qui représente un câblage de distribution d'énergie d'environ 50 m, comme sur les navires et les gros avions. Cependant, pour des plateformes plus petites – comme dans les applications automobiles – cette inductance peut être beaucoup plus faible et des RSIL de 5 µH sont recommandés. Le RSIL que nous présentons ici est de ce dernier type.

Dans la pratique, cependant, ce circuit simple ne fonctionnera pas comme prévu en raison des capacités et inductances parasites présentes dans sa structure

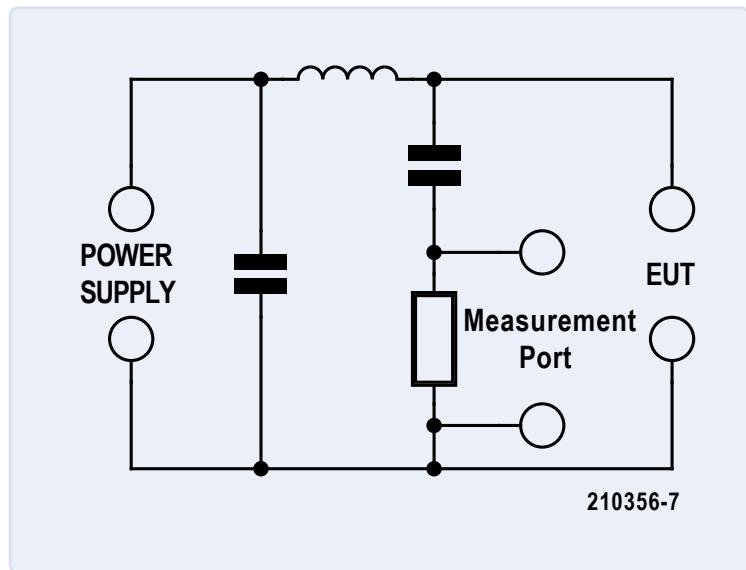


Figure 2. Schéma de base, simplifié du RSIL (EUT = M&T)



WEBINAIRE CEM

Elektor organisera prochainement un webinaire sur la CEM en partenariat avec Würth Elektronik et Rohde & Schwarz. L'heure et la date restent à déterminer et les modalités d'inscription au webinaire seront annoncées sur le site web du magazine Elektor et dans notre lettre d'information (e-zine). Consultez www.elektormagazine.fr.

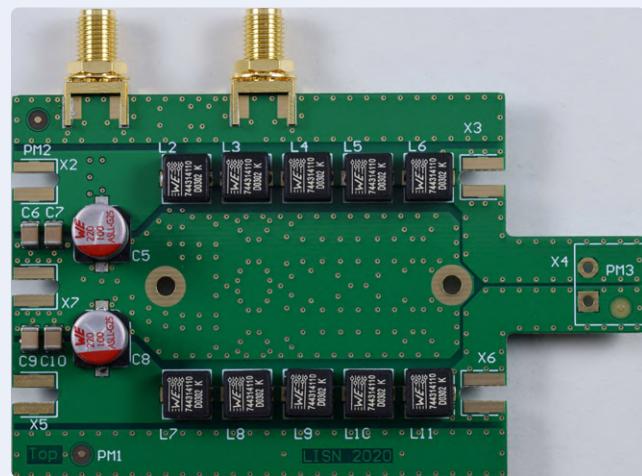
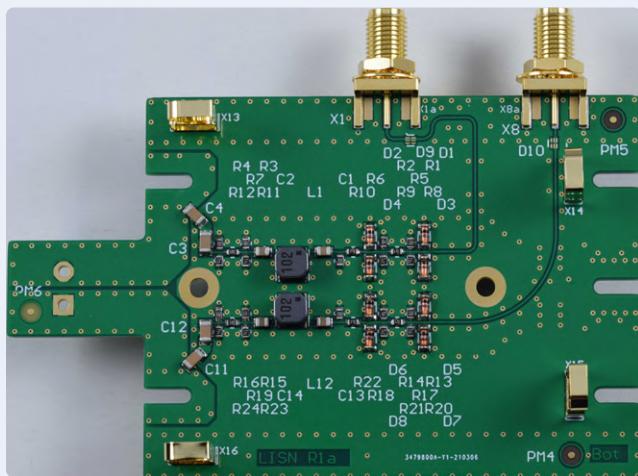
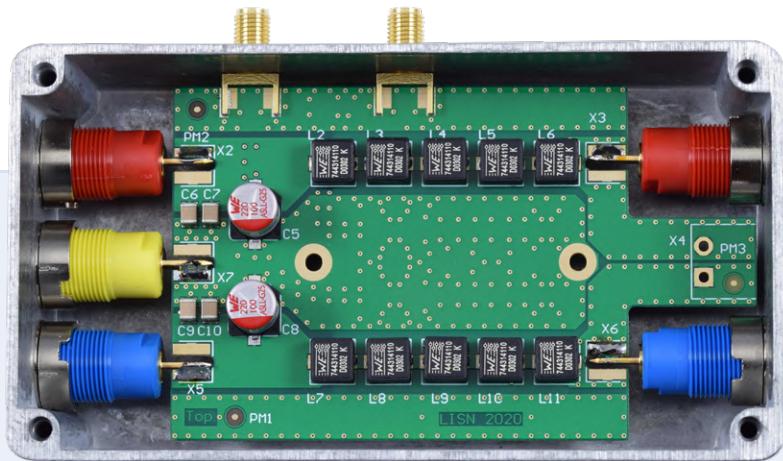


Figure 3. Photos du prototype de notre RSIL pour courant continu.

physique. Le véritable RSIL de ce projet, comme le montrent les photos de la **figure 3**, est donc un peu plus complexe. En outre, les circuits de protection des entrées de l'analyseur de spectre mentionnés précédemment sont omis dans le schéma de base. Les schémas complets et le circuit imprimé de notre RSIL seront décrits et expliqués plus en détail dans le deuxième article de ce projet.

Il existe de nombreux types différents de RSIL, spécialement conçus pour diverses normes de test, plages de fréquences, tensions et courants de fonctionnement, type de source d'alimentation (alternative ou continue) et dispositif à tester. Le modèle présenté dans ce projet Elektor/Würth Elektronik est basé sur un circuit de démonstration d'Analog Devices (DC2130A – anciennement Linear Technology). La simulation de réseau « dual DC RSIL » est utilisée pour mesurer les émissions conduites d'un objet à tester dans la gamme de fréquences allant jusqu'à 200 MHz, conformément à la norme CISPR 25 / ISO 7637 pour les systèmes électriques automobiles. Notre montage présente deux sorties pour mesurer les interférences RF

CARACTÉRISTIQUES TECHNIQUES

Chemin RF

Canaux :	2 (avec diodes de protection)
Bande passante :	200 MHz
Inductance :	$5 \mu\text{H} \parallel 50 \Omega$
Atténuation interne :	10 dB

Chemin DC

Courant max. :	$< 10 \text{ A}$
Tension max. :	$< 60 \text{ V}$
Résistance :	$< 2 \times 70 \text{ m}\Omega$
Taille du circuit imprimé :	121 mm x 66 mm x 35 mm (hors connecteurs)

ACHETER LE KIT RSIL

Un kit pour réaliser ce montage, avec tous les CMS soudés sur le circuit imprimé, sera bientôt disponible dans la boutique Elektor. Ce kit contient tous les connecteurs (que vous devez souder vous-même) et un boîtier prépercé en aluminium moulé sous pression. Le prix n'est pas encore fixé.



PRODUIT

➤ **STEMlab 125-14 (kit de démarrage)**
www.elektor.fr/17939

sur les deux bornes de puissance de ces systèmes, en utilisant des inductances de blocage de $5 \mu\text{H}$. Notre RSIL est adapté aux mesures de CEM en mode différentiel et en mode commun, ses propriétés et spécifications les plus importantes sont listées dans les caractéristiques techniques (voir encadré). D'autres informations sur ce projet sont déjà disponibles sur les pages d'Elektor Labs [2].

Que vous faut-il de plus ?

Un RSIL seul n'est pas un instrument de mesure complet, mais plutôt une sonde spéciale à connecter à un appareil du genre analyseur de spectre ou récepteur IEM pour mesurer les émissions conduites sur les bornes d'alimentation du M&T. Grâce au partenariat avec Würth Elektronik, nous pourrons prochainement proposer le kit RSIL à un prix intéressant dans notre

boutique en ligne. Mais même si une bande passante relativement modeste suffit pour les signaux de mesure, un analyseur de spectre professionnel sera toujours beaucoup plus cher que le RSIL. C'est pourquoi nous présenterons dans le deuxième article une alternative plus abordable utilisant une carte STEMlab 125-14 de Red Pitaya qui exécute une application de transformée de Fourier discrète (DFT). Bien que ce ne soit pas la solution idéale, les résultats des mesures contribueront à l'évaluation des émissions conduites du M&T.

Les instruments de mesure de CEM sont assez coûteux, mais l'investissement total dans ce projet de RSIL peut être amorti relativement rapidement par la réduction du coût de la certification complète grâce à la réalisation de tests de préconformité. 

(210356-04)

Des questions, des commentaires ?

Envoyez un courriel au rédacteur (luc.lemmens@elektor.com) ou contactez Elektor (redaction@elektor.fr)

Contributeurs

Idée et conception :

Robert Schillinger (Würth Elektronik)

Texte : Robert Schillinger, Ton Giesberts

Illustrations : Patrick Wielders

Rédaction : Luc Lemmens

Traduction : Helmut Müller

Mise en page : Sylvia Sopamena

LIENS

- [1] Réseau de stabilisation de l'impédance de ligne (RSIL) : https://en.wikipedia.org/wiki/Line_Impedance_Stabilization_Network
- [2] Ce projet sur Elektor Labs : www.elektormagazine.fr/labs/dual-dc-lisn-for-emc-pre-compliance-testing-210296

— Publicité



L'ingénierie est complexe

Ajoutez à vos favoris notre page conversion et calculatrice

Visitez nos ressources pour les calculs du secteur électronique
mouser.fr/conversion-calculators

MOUSER
ELECTRONICS

charge électronique

en CC et CA

Jusqu'à 400 V et 10 A (crête)



Rainer Schuster (Allemagne)

Concevoir et essayer une alimentation ainsi qu'en mesurer les caractéristiques nécessite une charge. Une résistance (de puissance) suffit, mais l'essai de l'alimentation à différents niveaux de puissance devient fastidieux si différentes résistances doivent être connectées en série et/ou en parallèle pour adapter la charge. La charge électronique ou e-charge (e-Load) est une solution : elle peut être contrôlée sans interventions répétées sur le circuit d'essai.

Outre les modes résistance constante et courant constant, notre e-charge peut adopter un mode dynamique pour étudier le comportement de la source à des courants de sortie variant rapidement. Cette dernière option nécessite une connexion USB (galvaniquement isolée) à un PC équipé d'une application de commande qui peut afficher les valeurs efficaces vraies (*true RMS*) de tension et de courant mesurées et calculées par l'e-charge. Contrairement à beaucoup d'autres e-charges, ce circuit est aussi adapté aux essais de sources de *courant alternatif*.

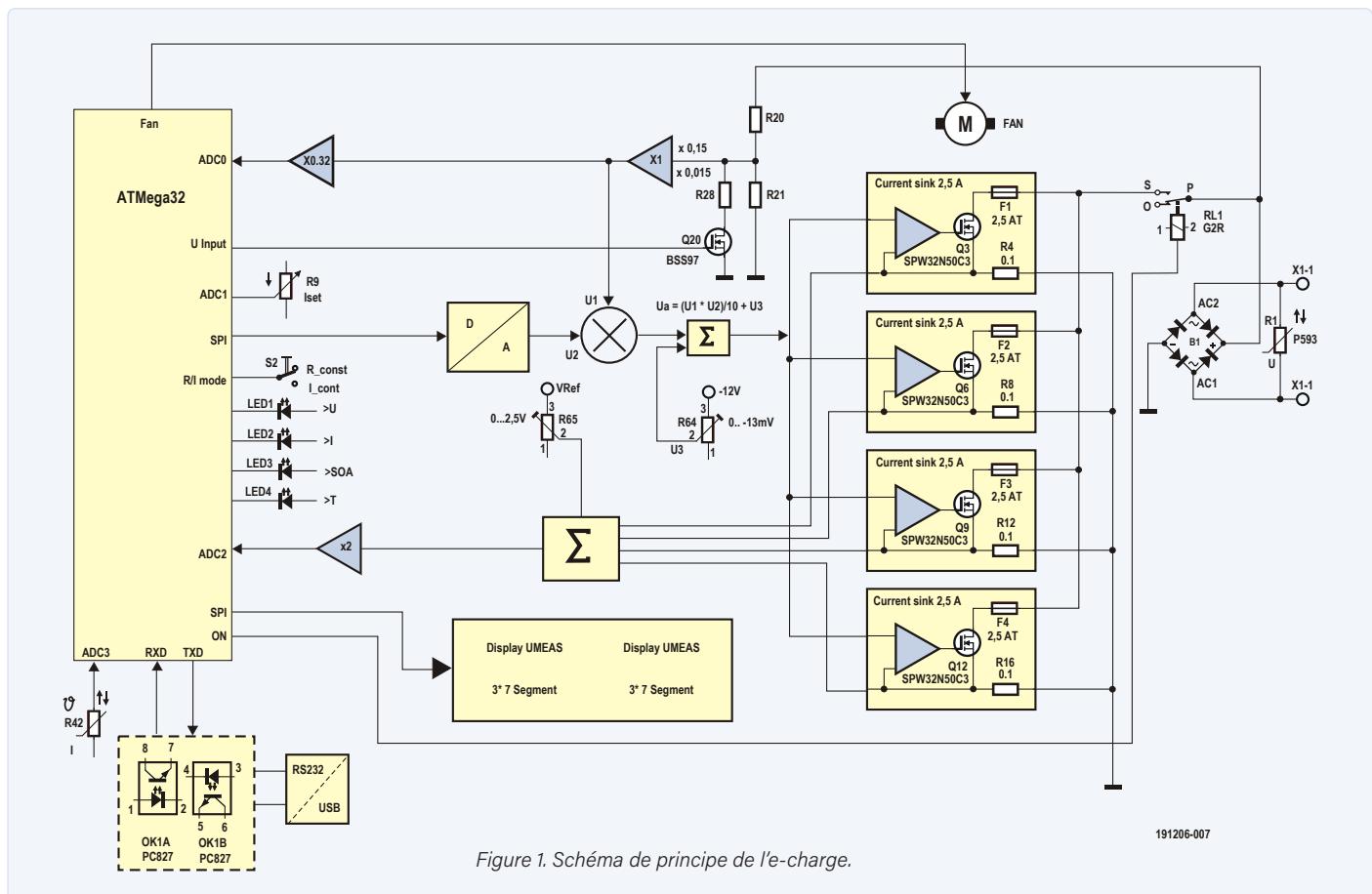
Le matériel

La **figure 1** montre le schéma fonctionnel de l'e-charge. Le principe et le fonctionnement seront discutés plus en détail ci-après. Quatre MOSFET (Q3, Q6, Q9 et Q12) en parallèle forment une charge capable d'absorber un courant de 10 A max. (voir **fig. 2**). La tension de commande est appliquée aux entrées non-inverseuses des ampli-op IC1 et IC2. Le courant de charge résultant pour chaque étage est égal à cette tension de commande divisée par la résistance de drain (R4, R8, R12, R16). Le courant maximal à travers chaque

unité est donc de 2,5 A.

Le radiateur des transistors de charge est un LK75 de *ELV Elektronik* secondé par un ventilateur de 12 V. La résistance thermique résultante du dissipateur vaut 0,44 K/W. Il faut ajouter la résistance thermique de la rondelle isolante (1,25 K/W) et de la jonction au boîtier du transistor (également 0,44 K/W). La résistance thermique totale est donc de 2,13 K/W. Pour chaque transistor, la puissance max. dissipée qui en résulte (à une température de jonction max. de 150 °C et une ambiante de 25 °C) est $P_v = (T_j - T_a) / R_{th}$

AVERTISSEMENT. Les hautes tensions présentent un risque létal. Le circuit décrit ici n'est pas destiné aux débutants. Vous ne devez pas le construire ni l'utiliser si vous n'êtes pas expérimenté en haute tension.



191206-007

Figure 1. Schéma de principe de l'e-charge.

= 58 W, ou 232 W pour quatre unités. Pour des raisons de sécurité, nous avons limité la puissance dissipée à 200 W.

Génération de la tension de commande

Dans le cahier des charges de cette charge électronique, nous avions inclus la gestion du courant alternatif. La tension de commande doit avoir la même forme d'onde que la tension d'entrée. Pour ce faire, la tension d'entrée (préalablement mise à une échelle convenable par Q2/IC3) est appliquée à l'entrée X1 du multiplicateur analogique IC5. La tension de commande est appliquée à Y1 du multiplicateur. La sortie W du multiplicateur est directement connectée à l'entrée des quatre étages de sortie MOSFET. Le multiplicateur utilisé est un AD633, les entrées X2 et Y2 sont mises à la masse et Z est connectée à une tension négative de 0 à 100 mV. La tension de sortie W est :

$$W = ((X1 * Y1) / 10) + Z$$

La tension Z est réglée (R64) pour compenser l'offset (décalage) de courant. La tension de contrôle Y1 est calculée par le microcontrôleur et envoyée au convertisseur N/A à 12 bits

MCP4921 (IC6) dont VR1 fixe la tension de référence à 2,5 V. C'est également sa tension de sortie maximale.

Mesure de la tension

La tension d'entrée de la charge est mesurée par le microcontrôleur. Pour couvrir la gamme de 0 à 400 V avec une résolution suffisante, le microcontrôleur se charge de la diviser en deux sous-gammes. Si Q20 est activé, la tension d'entrée est divisée par 66 (R20, R28||R21). Cette tension est appliquée au multiplicateur et la forme d'onde du courant peut suivre la tension d'entrée. Cette valeur est ensuite divisée par 10 (par R22 et R23) puis multipliée par 3,26 par IC3B. Si la tension d'entrée est inférieure à 50 V, Q20 est coupé et la tension d'entrée est divisée par 6,7 (R20/ R21). À une tension d'entrée de 50 V, la tension de sortie d'IC3B est de 2,77 V (valeur binaire 567 sur le CA/N à 10 bits), la résolution est de 100 mV. À une tension d'entrée de 400 V, la tension de sortie d'IC3B est de 1,956 V (valeur binaire 400), la résolution est de 1 V. Selon que l'entrée est en CA ou en CC, on mesure respectivement la tension CA redressée ou la tension CC. Le logiciel se charge de calculer la tension efficace vraie.

Mesure de courant

R31 à R34 et IC7A + IC7B effectuent la somme des courants des quatre MOSFET et à 2,5 A_{eff} chacun, l'amplitude de sortie est de 1 V_{eff}/10 A_{eff}. IC7D multiplie cette valeur par 4,3, on a donc une sortie de 0,43 V/A. Selon que l'entrée est en CA ou en CC, on mesure respectivement le courant CA ou le courant CC redressé. Le logiciel se charge de calculer le courant efficace vrai. R65 permet de régler l'offset de l'affichage du courant. Note : à 10 A_{CC}, il y a 4,3 V à l'entrée du CA/N. En CA, la valeur efficace ne doit donc pas dépasser 7 A.

Microcontrôleur

Un microcontrôleur ATMega32 commande la charge électronique. Ce contrôleur se programme via SV1 sur l'interface ISP (par ex. à l'aide d'un AVR/ISP MKII). Pendant la programmation, le contrôleur doit être alimenté par l'alimentation interne.

Note de sécurité importante : quand la charge électronique est reliée au programmeur, il n'y a pas d'isolation galvanique des entrées. Toujours déconnecter le programmeur avant de brancher les bornes d'entrée au secteur

Figure 2. Partie principale du schéma.

ET avant de déconnecter la terre de la borne d'entrée négative.

Afficheur

L'affichage de la tension et du courant est confié à des afficheurs LED à sept segments commandés par un *MAX7219 SPI*. Les schémas proviennent d'un ancien projet *Elektor* [2] dont l'agencement du circuit imprimé est légèrement modifié, voir la page du projet [1].

Interface USB

M2 gère la communication avec un PC (fig. 2). La carte reçoit un module provenant de *Conrad*, mais tout circuit USB-UART alimenté par USB peut être utilisé. Les optocoupleurs OK3 et OK4 assurent l'isolation galvanique de la connexion au microcontrôleur.

Alimentation électrique

Pour le microcontrôleur, la charge électronique a besoin de +5 V et pour la partie analogique, de +12 V et -12 V. Deux transformateurs montés sur le circuit imprimé y pourvoient. La régulation des tensions d'alimentation est confiée à des régulateurs standard (78XX et 79XX) (fig. 3).

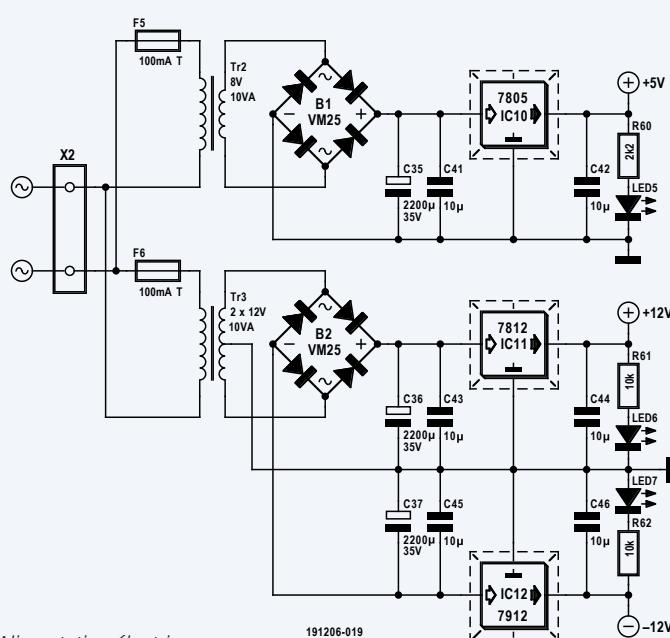
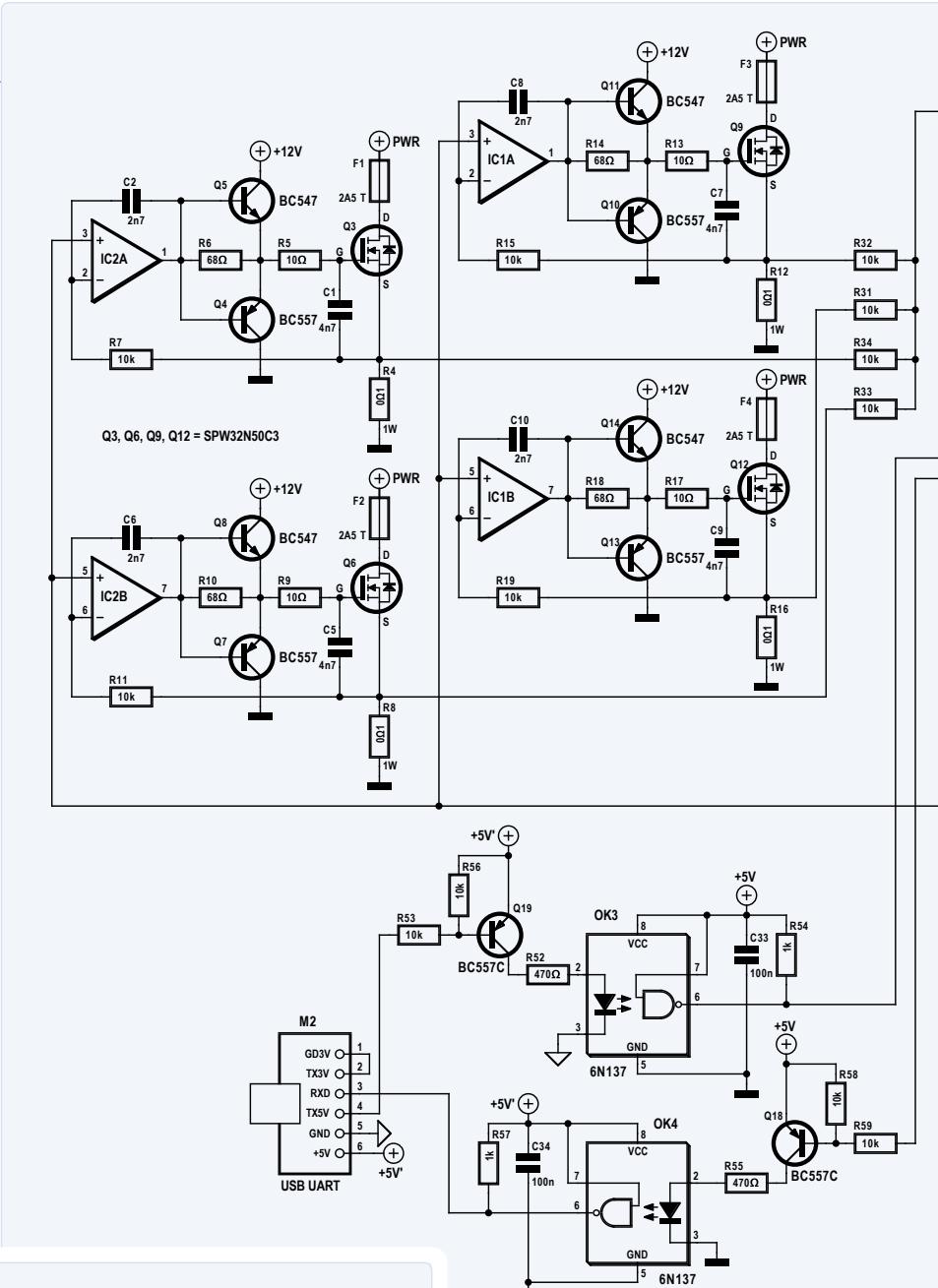
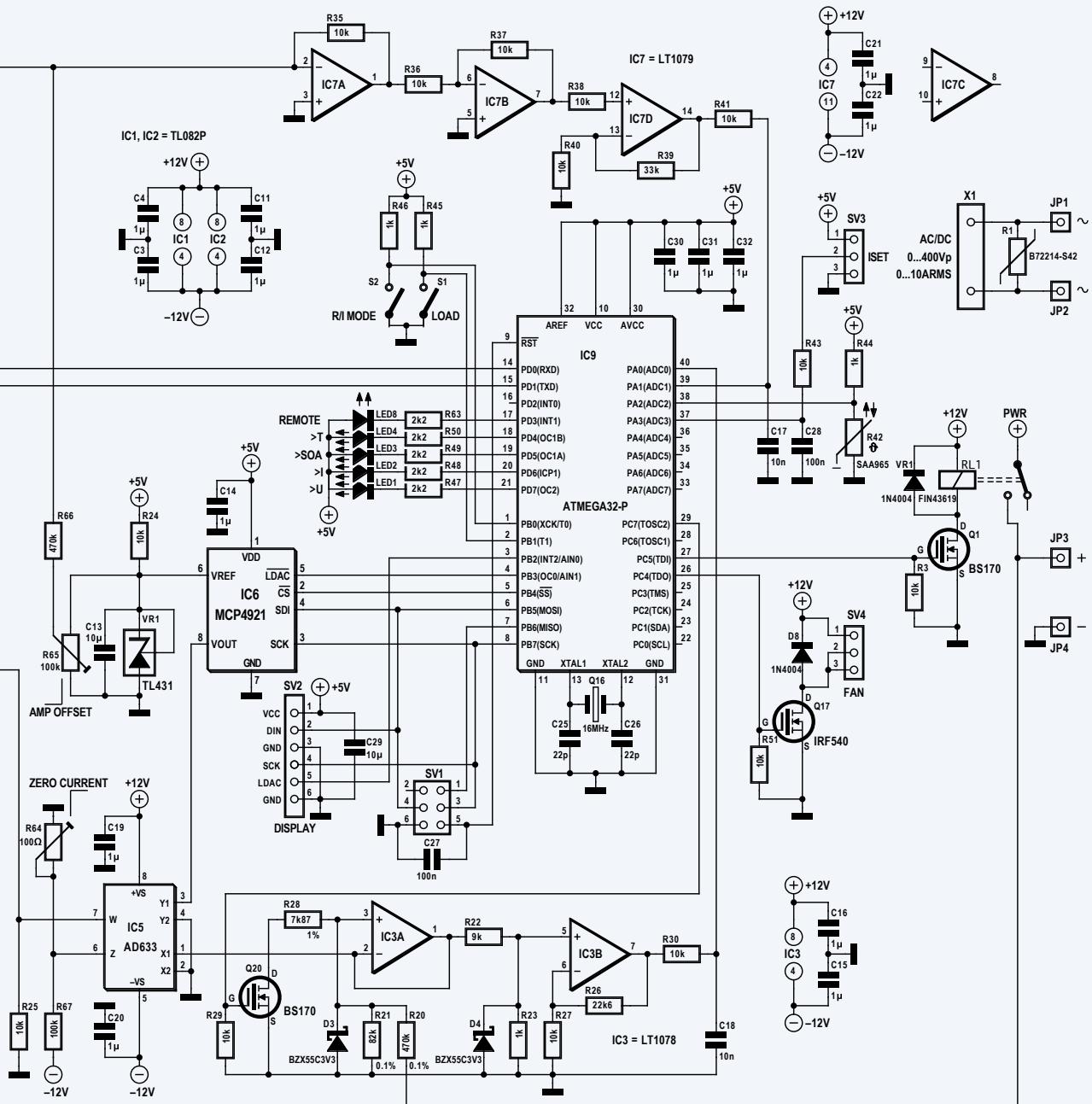


Figure 3. Alimentation électrique.

Micrologiciel

Le micrologiciel de l'ATmega32 a été développé avec *BASCOM AVR*. Si vous souhaitez le modifier, vous devrez acquérir une version sous licence. Le code source et le fichier HEX sont téléchargeables à l'adresse [1].

Tension et courant efficaces vrais sont calculés toutes les 20 ms. La bande passante de mesure atteint 100 Hz. Pour « lisser » les mesures, on calcule la moyenne sur dix mesures consécutives, de sorte qu'une mesure complète prend 200 ms. Le micrologiciel calcule le courant cible selon le réglage du commutateur R-const/I-const. En mode courant constant (I-const), le courant



doit être fixe. En mode résistance constante, celle-ci est simulée en faisant varier le courant I tel que $I = V_{input}/R\text{-const}$. Pour que le courant soit constant, la valeur du CN/A doit être calculée d'après la consigne de courant et la tension efficace instantanée.

En raison des tolérances des composants, le courant réel peut différer de la consigne. Pour y remédier, la fonction *Set_dest_current* corrige d'après la courbe caractéristique de I . Si le courant obtenu est supérieur à la consigne, la commande augmente de 5 mA. Si le courant obtenu est inférieur à la consigne, la commande diminue de 5 mA.

La boucle du programme principal vérifie d'abord les conditions d'erreur :

- Tension > 400 V : allume la LED $> U$ et coupe le relais de charge. Quand la tension < 400 V, la LED s'éteint et le relais est réactivé.
- Courant : 1 s après la modification de la consigne, vérifie si le courant obtenu est $< \text{consigne} + 100$ mA. Si ce n'est pas le cas : allume la LED $> I$, désactive le relais de charge. Pour la réactiver, il faut couper puis rallumer l'e-charge.
- Alimentation : une condition tension * courant > 200 W allume la LED $> SOA$ et désactive le relais de charge. Pour la réactiver, il faut couper puis rallumer l'e-charge.
- Température : si la température du radiateur (mesurée avec une CTN connectée au CA/N2) atteint 40 °C, le venti-

lateur démarre. C'est aussi le cas si la puissance dépasse 100 W. Si la température descend à moins de 35 °C et que la puissance est < 100 W pendant > 10 s, le ventilateur s'éteint. Si la température du radiateur dépasse 80 °C, la LED $> T$ s'allume et le relais de charge est désactivé. Si elle redescend à moins de 75 °C, le relais de charge est réactivé.

Ces conditions sont contrôlées toutes les 500 ms. L'affichage des mesures tension/courant est aussi rafraîchi toutes les 500 ms. Hors commande à distance, les interrupteurs *Load On/Off* et *I-const/R-const* sont lus ainsi que la valeur du potentiomètre de réglage du courant.

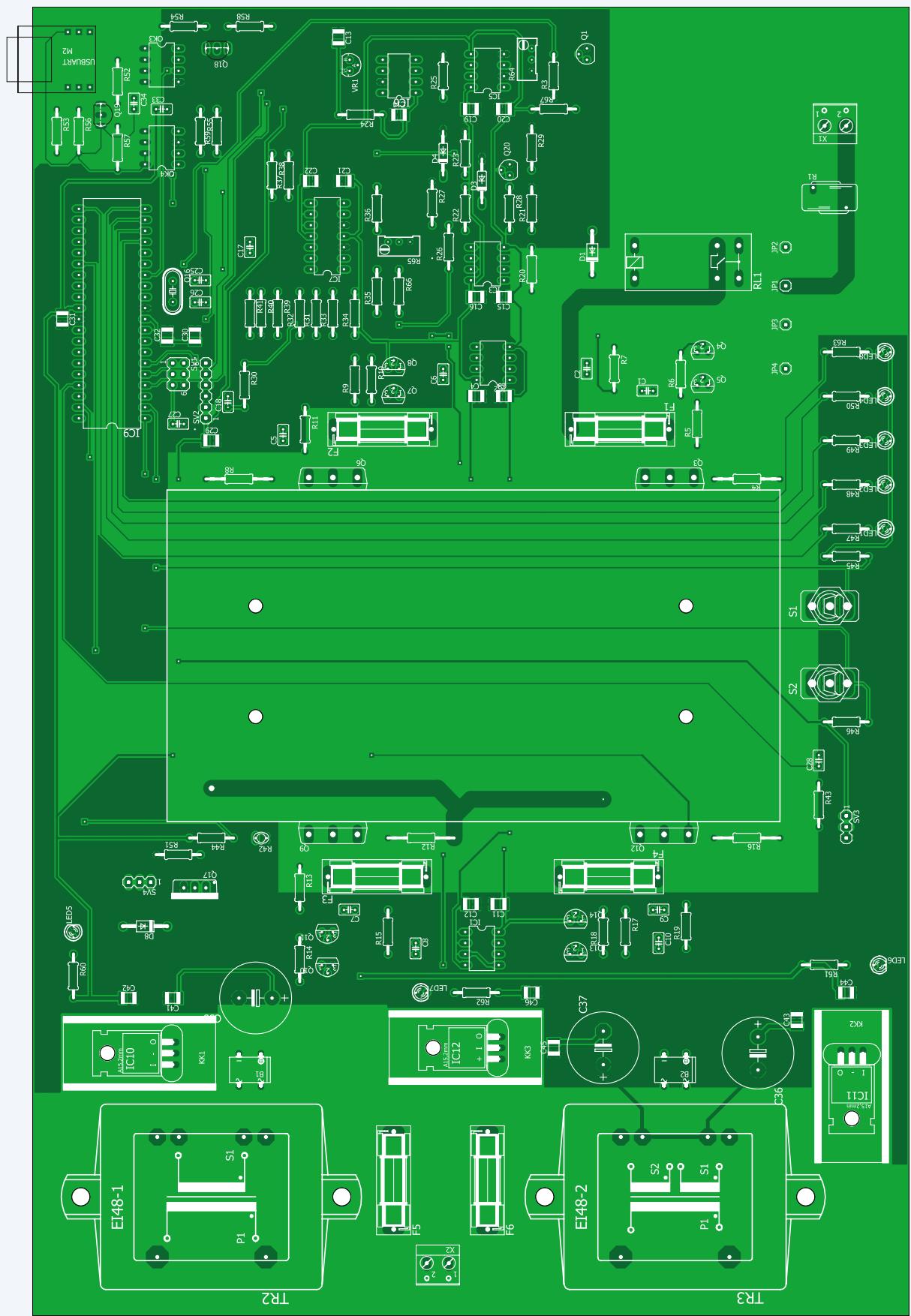


Figure 4. Vue du circuit imprimé de la carte principale. (reproduction à 85% de la taille réelle)

Réglage courant/résistance

Si la charge est en mode *I-const* puis mise sur *On*, la plage de courant est de 1 A. Si le commutateur de charge est mis sur *Off*, la plage de courant est définie par la tension d'entrée. Cette plage est de 10 A si la tension d'entrée est inférieure à 20 V ; sinon, la plage de courant vaut 200 W/tension réelle.

En mode *R-const*, la résistance minimale est elle aussi calculée quand le commutateur de charge est sur *Off* et vaut U/I = tension d'entrée/consigne de courant. Le réglage max. de résistance vaut cent fois la résistance minimale.

Note : en mode *I-const*, les butées antihoraire et horaire du potentiomètre correspondent respectivement au courant minimal et maximal. Pour éviter une surintensité au passage du mode *I-const* au mode *R-const*, la butée antihoraire du potentiomètre correspond à la valeur maximale de la résistance.

Réalisation matérielle

Munis des fichiers CAO *Eagle* et *Gerber* (téléchargeables depuis [1]), vous pourrez commander les cartes imprimées chez votre fournisseur préféré. La **figure 4** montre l'aspect de la carte principale. Pour la carte d'affichage, consultez la page Elektor Labs, vous y trouverez aussi les listes de composants des deux cartes. Un boîtier accueille les cartes. La **figure 5**

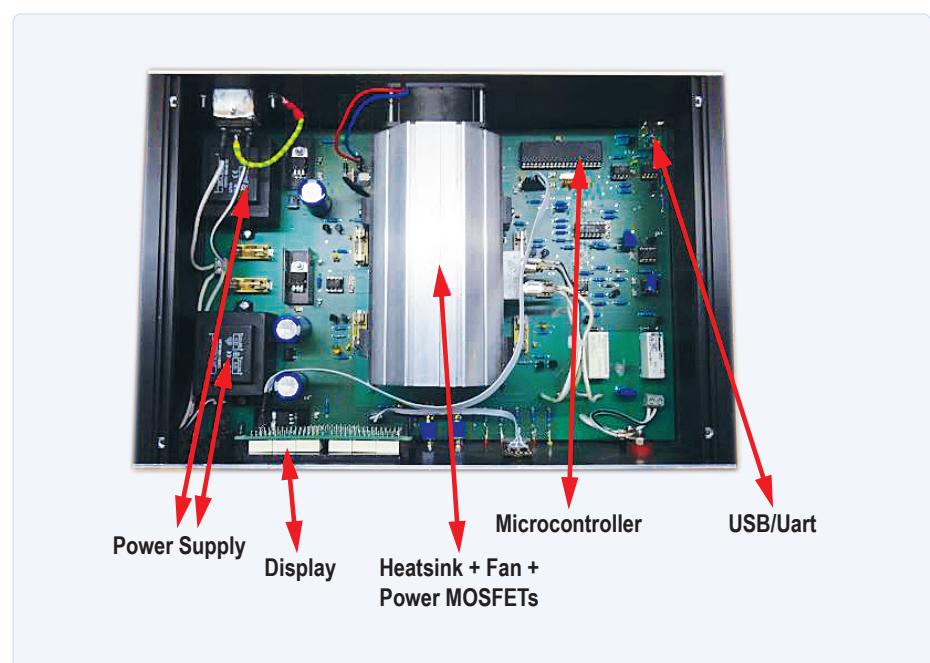


Figure 5. Vue de dessus de l'e-charge à l'intérieur du boîtier.

propose une vue de dessus. Les MOSFET de puissance sont montés sur le radiateur avec des rondelles thermiques TOP3 et de la pâte thermique. Le pont de diodes des mesures CA est également monté sur ce dissipateur et relié à JP1-JP4 sur le circuit imprimé principal. Le dissipateur lui-même est fixé au circuit imprimé avec des entretoises en plastique de 2 mm.

Logiciel de commande

Une application de commande de l'e-charge par USB, a été écrite en Visual Basic pour Visual Studio 2015. L'exécutable s'appelle *Electronic_ACDC_Load.exe*. La **figure 6** illustre l'écran principal. Un fichier zip des codes source + exécutable de l'application est téléchargeable depuis [1]. Sur cette page web, on trouve également la documentation

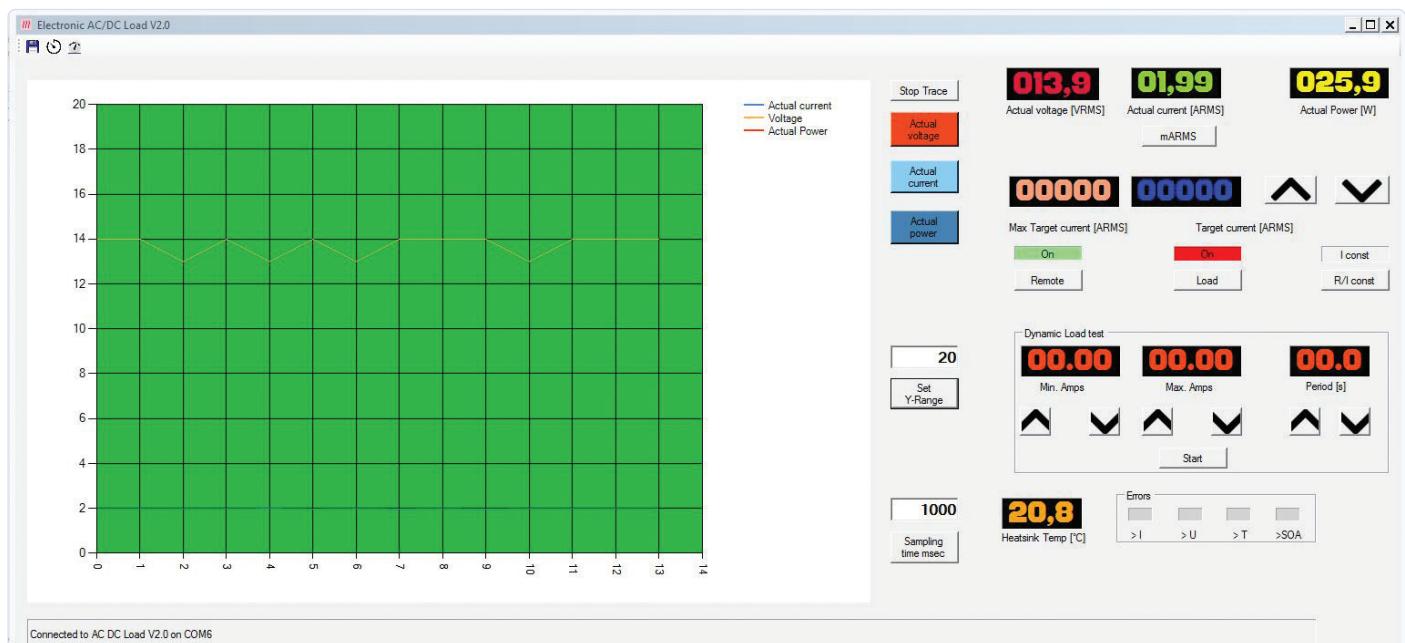


Figure 6. Écran principal de l'application PC.

complète du projet (*Description.PDF*). Elle comporte un tableau des commandes disponibles pour communiquer avec l'e-charge par l'interface USB. Ces informations seront utiles si vous souhaitez modifier le logiciel du PC.

Pour se connecter à la charge, le logiciel scrute les ports COM disponibles en demandant trois fois (au plus) à intervalle d'1 s la chaîne d'identification. Quand le programme reçoit la chaîne d'ID, celle-ci et le port COM s'affichent dans la ligne d'état. Puis les données (tension, courant, puissance et température du radiateur) sont scrutées et affichées à intervalle d'1 s. Pour envoyer les consignes de courant/résistance, le bouton *Remote* doit être sur *On*. Dès lors, le relais de charge (*Load On/Off*) et le mode *R-const/I-const* sont sous contrôle du logiciel. En mode *I-const*, le bouton *mARM* permet de faire passer l'affichage de *A* à *mA* et vice versa. Les boutons *Up/Down* permettent d'augmenter/diminuer le courant. Le pas du courant est de 10 mA jusqu'à 100 mA, puis 100 mA jusqu'à 1 A et enfin 1 A. Un fois les touches relâchées, le pas revient à 10 mA (uniquement en mode *I-const*).

En mode *R-const*, la consigne de résistance peut être augmentée (sans limitation) ou diminuée. La limite basse correspond à la résistance minimale calculée (v. ci-dessus). Le pas de résistance est de 1 Ω jusqu'à 10 Ω, puis 10 Ω jusqu'à 100 Ω et enfin 100 Ω. Une fois les touches relâchées, le pas revient à 1 Ω.

Essai de charge dynamique

Deux courants (mode *I-const*) ou deux résistances (mode *R-const*) peuvent être prérégulés, par ex. pour tester des alimentations. Bien que l'application affiche la forme d'onde de la source d'alimentation, il faut connecter un oscilloscope aux bornes de sortie de l'alimentation pour bien observer sa réponse dynamique. *Period* permet de définir un intervalle en secondes. *Start* répète en boucle successivement les deux consignes (courant ou résistance) avec cet intervalle. *Stop* met fin à ce processus. La consigne passe alors à 0 en mode *I-const* ou à la résistance max. en mode *R-const*.

Affichage des courbes de tension, de courant et de puissance

La touche *Start Trace* affiche les courbes *Actual Current*, *Actual Voltage* et *Actual Power* précédemment activées. La touche *Sampling time msec* permet de régler la vitesse d'enregistrement. Lorsque le

nombre de points affichés atteint 60 ou qu'une consigne est modifiée, les traces sont effacées et l'enregistrement recommence. *Set Y-Range* permet de modifier la résolution de l'axe vertical.

Journal

Le symbole de disquette permet d'ouvrir un journal d'enregistrement à long terme des données. Les données sont enregistrées au format *.CSV*, ce qui permet de les traiter dans un tableur standard. Le symbole de minuterie permet de définir la durée de collecte du journal. En cliquant à nouveau sur le symbole de la disquette le journal ouvert se ferme.

Étalonnage

Le symbole d'*instrument* donne accès à l'étalonnage de la charge électronique. Cela implique de se munir d'instruments déjà étalonnés pour mesurer tensions et courants. En modifiant les valeurs d'étalonnage de l'affichage du courant *Current display*

calibration value et de la tension *Voltage display calibration value*, vous pouvez caler les valeurs affichées sur celles des instruments étalonnés. Les valeurs sont stockées dans l'EEPROM du microcontrôleur.

Pour les ingénieurs, l'usage d'une charge électronique est moins fréquent que l'usage d'autres instruments. Elle sera cependant très utile pour construire, essayer ou réparer une alimentation. Des e-charges sont disponibles dans le commerce spécialisé, mais il est possible d'en construire une soi-même et la présente réalisation mérite que l'on s'y arrête.

Cet article est basé sur les informations présentées sur la page d'un projet Elektor Labs [1]. Cette page complète cet article. Vous y trouverez en téléchargement les logiciels, les fichiers de CAO des cartes et leur liste de composants. Une vidéo de présentation de cette e-charge [3] est proposée. 

(191206-04)

Des questions, des commentaires ?

Envoyez un courriel à l'auteur (rainerschuster@mnet-mail.de) ou contactez Elektor (redaction@elektor.fr).

Contributeurs

Idée, conception, texte et illustrations :

Rainer Schuster

Rédaction : **Luc Lemmens**

Schémas :

Patrick Wielders, Kurt Diedrich

Mise en page : **Giel Dols**

Traduction : **Yves Georges**



PRODUITS

➤ **JOY-iT JT-HD35, résistance de charge USB (35 W)**
www.elektor.fr/joy-it-hd35-usb-load-resistor-35-w

➤ **Siglent SDL1020X-E, charge électronique CC programmable (200 W)**
www.elektor.fr/siglent-sdl1020x-e-programmable-dc-electronic-load-200-w

LIENS

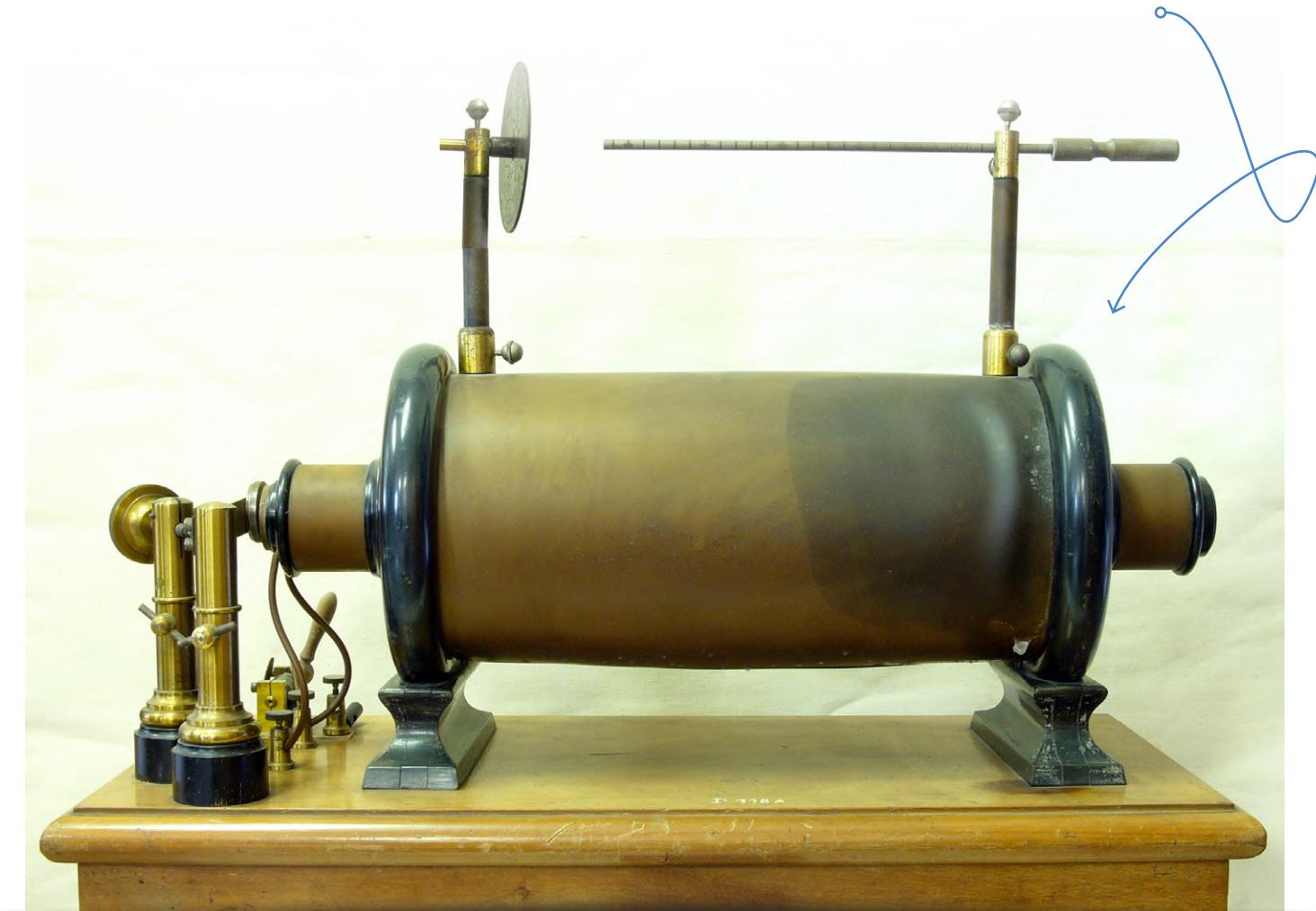
[1] **Page Elektor Labs de ce projet :**
www.elektormagazine.fr/labs/electronic-load-for-dc-and-ac

[2] **Afficheurs à six chiffres avec port SPI :**
www.elektormagazine.fr/magazine/elektor-200907/11294

[3] **Vidéo :** www.electronics-lab.com/ac-dc-200w-electronic-load/

démarrer en électronique... (9)

N'ayez pas peur du choc... un peu de « self control » suffit !



Bobine d'induction (source : https://commons.wikimedia.org/wiki/File:Induktionsapparat_hg.jpg).

Eric Bogers (Elektor)

Soyons honnêtes : les électroniciens amateurs détestent les selfs. Les selfs d'arrêt (dites de choc) prêtes à l'emploi sont tout juste tolérables, mais si nous devons les bobiner, c'est vite dissuasif. En général, obtenir les caractéristiques souhaitées demande d'innombrables essais.

À tous égards, les selfs (aussi connues sous le nom de bobines ou d'inductances) sont l'opposé des condensateurs. L'impédance (résistance en alternatif) d'un condensateur diminue quand la fréquence augmente ; au contraire, l'impédance d'une bobine augmente avec la fréquence. Et, alors que dans un condensateur le courant précède la tension (voir le précédent épisode de « démarrer en électronique »), dans une self, le courant court après la tension.

La **figure 1** illustre quelques « exemples classiques » de bobines. À l'extrême gauche, une bobine de filtrage (axiale) ; en raison de sa géométrie et des anneaux de couleur, une telle inductance* est facilement confondue avec une résistance. D'ailleurs, le code couleur de ces selfs est comparable à celui des résistances. Au milieu, en haut, se trouve une self torique ; ces selfs sont très

fréquentes dans les réseaux de suppression du bruit. Et en haut à droite, on trouve une self à air dans toute sa splendeur. On en voit souvent dans les filtres de recouvrement des haut-parleurs. Enfin, en bas, il s'agit d'une self à noyau en ferrite pour les applications haute fréquence. Ici, on doit soi-même bobiner le fil sur la carcasse et, dans certaines limites, on peut donc déterminer librement son inductance.

*NDT : en langage châtié, le mot inductance est préféré pour désigner une self. Ce mot désigne donc aussi bien la self que sa valeur.

La self comme réservoir d'énergie

Lorsqu'un courant traverse une self, un champ magnétique apparaît autour de celle-ci. Il stocke de l'énergie. Si on interrompt le courant qui traverse la self, cette énergie est restituée : il apparaît une

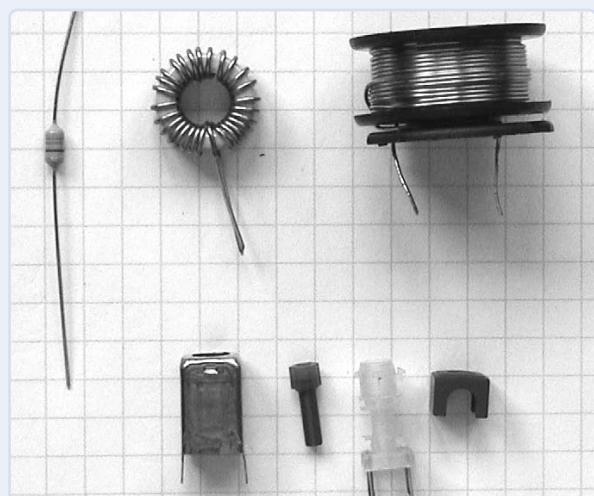


Figure 1. Quelques inductances.

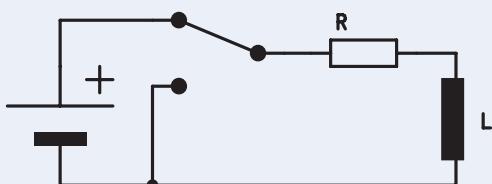


Figure 2. Inversion du courant dans une inductance.

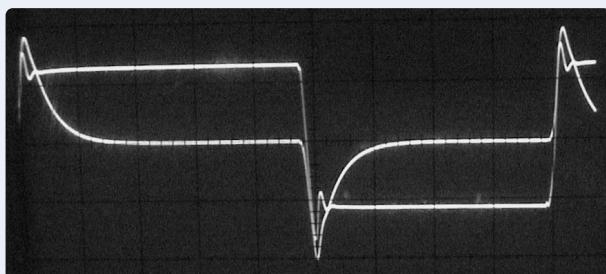


Figure 3. Allers-retours lors de l'allumage et de l'extinction.

tension qui tend à maintenir le courant d'avant la coupure. La mesure de l'aptitude d'une self à stocker de l'énergie s'appelle l'inductance* L (exprimée en henrys, symbole H). La quantité d'énergie stockée dans une self est donnée par la formule :

$$W = \frac{1}{2} \cdot L \cdot I^2$$

Vous remarquerez la similitude avec la formule correspondante du condensateur, mais pour la bobine c'est un *courant* I et pour le condensateur une *tension* U.

Comportement d'allumage et d'extinction

Les caractéristiques de démarrage et d'arrêt d'une self sont opposées à celles de charge/décharge d'un condensateur. Quand on applique une tension aux bornes d'une self, au début le courant est presque nul ; la self présente une impédance élevée et la tension à ses bornes est à son maximum. Ensuite le courant augmente lentement tandis que la tension diminue en conséquence. Le courant à travers la self s'approche asymptotiquement d'un maximum et la tension à ses bornes se réduit à un minimum, les deux valeurs étant déterminées par la résistance de l'enroulement.

Le comportement d'une self est intéressant à l'instant de la commutation : quand on inverse la polarité de la tension appliquée (fig. 2), le champ magnétique de la self s'effondre et une tension est induite dont la polarité tente de maintenir le courant dans la direction initiale – cette polarité est donc l'inverse de celle appliquée initialement. La conséquence est qu'au moment de la commutation, une « tension de retour » élevée apparaît (fig. 3).

À la coupure, une self peut produire une tension considérablement plus élevée que celle appliquée initialement. (Un condensateur produit un phénomène comparable : il peut délivrer un courant considérablement plus élevé que celui qui a servi à le charger). Cette caractéristique des selfs peut être mise à profit pour produire une tension plus élevée que celle de départ ; un exemple est la bobine d'allumage [1] que l'on rencontre sur un moteur à essence. La haute tension pour les bougies est fournie par la bobine d'allumage qui fonctionne exactement selon ce principe.

Un effet indésirable de ce phénomène est que la tension induite peut détruire d'autres composants du circuit. Le relais commandé par un transistor (interrupteur où le champ magnétique d'une bobine permet d'actionner des contacts de commutation) en est l'exemple type. À la mise hors tension, la tension induite peut détruire ce transistor, ce qui *arrive effectivement*. On résoud ce problème par ex. par une diode dite de *roue libre*, qui court-circuite la tension induite et la rend ainsi inoffensive (fig. 4).

Circuits série et parallèle

Tout comme les résistances et les condensateurs, les selfs peuvent être connectées en série et en parallèle. Elles se comportent exactement de la même manière que les résistances, ce qui va de soi quand on y réfléchit : connectées en série, les nombres de tours s'ajoutent. Dans le cas d'un montage en série, les règles suivantes s'appliquent donc :

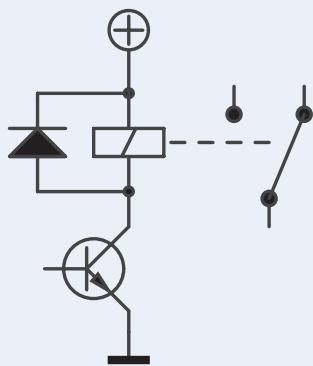


Figure 4. La diode protège le transistor quand le courant de la bobine du relais est coupé.

$$L_{tot} = L_1 + L_2 + L_3 + \dots + L_n$$

Et pour la connexion en parallèle :

$$L_{tot} = \frac{1}{\frac{1}{L_1} + \frac{1}{L_2} + \frac{1}{L_3} + \dots + \frac{1}{L_n}}$$

La bobine comme résistance en alternatif

Comme résistance en alternatif, la self se comporte encore une fois exactement à l'opposé du condensateur. Pour le condensateur, l'impédance diminue quand la fréquence augmente. Pour la self, l'impédance augmente avec la fréquence :

$$X_L = 2 \cdot \pi \cdot f \cdot L$$

La self introduit aussi un déphasage entre tension et courant : le courant est décalé de 90° par rapport à la tension.

Induction

Si une self est placée dans un champ magnétique variable, une tension y est induite. La formule ci-après permet de calculer la valeur de cette tension :

$$U_{ind} = n \cdot A \cdot B' \cdot \cos \alpha$$

où n est le nombre de spires de la self, A la section (moyenne) occupée par la self, B' la dérivée de l'induction magnétique par rapport au temps et α l'angle entre le plan des spires et le champ magnétique.

Si le champ magnétique varie selon une fonction sinusoïdale, nous pouvons écrire la dérivée de l'induction :

$$B' = 2 \cdot \pi \cdot f \cdot B \cdot \cos \phi$$

Sans l'induction, le monde serait totalement différent. Par ex. point de transformateur car une tension alternative dans l'enroulement primaire ne produirait pas de champ magnétique variable,

et n'induirait pas à son tour une tension dans le secondaire. Il n'y aurait pas de générateurs non plus (pensez à l'alternateur sous le capot).

Dans les applications audio, cette induction peut aussi être gênante : une boucle de terre involontaire (crée par ex. si l'équipement audio n'a pas été connecté correctement) peut produire un bourdonnement irritant.

Nous n'en avons pas fini avec les inductances. Nous en reparlerons dans le prochain article.

(210342-04)

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

Idée et illustrations : Michael Ebner

Texte et rédaction : Eric Bogers

Traduction : Yves Georges

Mise en page : Harmen Heida



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



LIEN

[1] Bobine d'induction :

https://en.wikipedia.org/wiki/Induction_coil

TRAITEMENT D'IMAGES

avec le module Jetson Nano de Nvidia

1^{ère} partie : matériel et logiciel

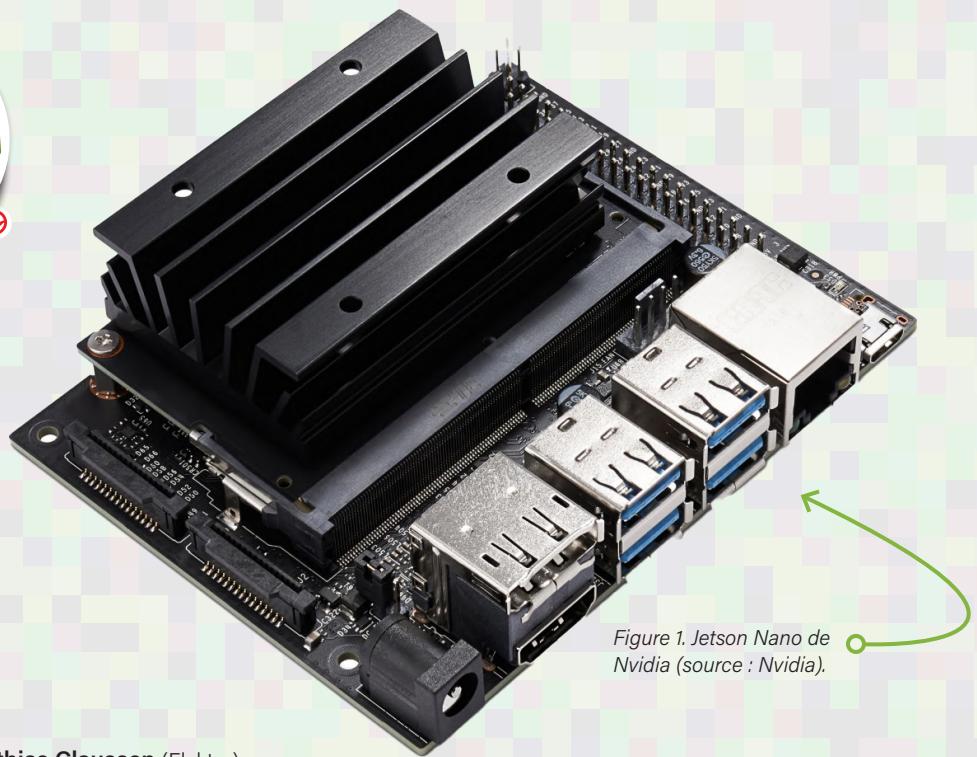


Figure 1. Jetson Nano de Nvidia (source : Nvidia).

Mathias Claussen (Elektor)

Des outils novateurs comme le module Jetson Nano de Nvidia permettent de se lancer facilement dans des projets d'intelligence artificielle (IA). Dans cette série d'articles, j'aborde en détail les applications possibles pour le traitement d'images ou la robotique. Commençons tout de suite par un aperçu du matériel et des logiciels...

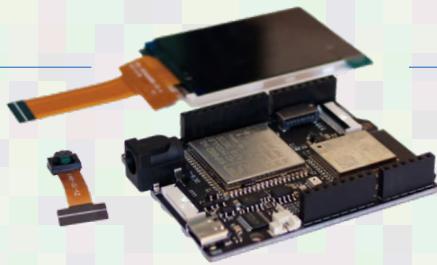


Figure 2. Carte Sipeed avec le microcontrôleur K210.



Figure 3. Carte FTHR MAX78000 avec accélérateur d'IA.



Figure 4. Carte Raspberry Pi 4.

Aucun développeur ne devrait ignorer le potentiel de l'intelligence artificielle (IA). Des outils comme le Maixduino de Sipeed et le MAX78000 de Maxim, déjà présentés dans le magazine Elektor, sont des options appropriées pour faire vos premiers pas dans l'univers de l'IA. Pour sa part, le module Jetson Nano de Nvidia (fig. 1) est un matériel beaucoup plus puissant de la classe des ordinateurs monocarte, doté d'un processeur ARM à quatre cœurs, de 4 Go de mémoire RAM et d'un processeur graphique assorti de 128 cœurs CUDA. Dans cette série d'articles, nous découvrons les applications possibles, notamment dans les domaines du traitement d'images ou de la robotique.

L'IA au quotidien

L'IA s'invite de plus en plus dans notre vie quotidienne. Les réseaux neuronaux permettent par ex. de reconnaître des objets, d'améliorer des images et de convertir les paroles d'une conversation en texte. Et ce domaine passionnant va contribuer à façonner notre avenir. Pour vous aider à faire vos premiers pas dans l'univers enthousiasmant de l'IA, nous avons déjà présenté la carte Maixduino de Sipeed (fig. 2) [1] et la carte FTHR MAX78000 de Maxim (fig. 3) [2].

Les deux cartes ont un point commun : elles contiennent des microcontrôleurs et des accélérateurs pour réseaux neuronaux, mais avec des ressources limitées. Le Jetson Nano de Nvidia (fig. 1) [3] est un matériel beaucoup plus puissant de la catégorie des ordinateurs monocarte (SBC).

Doté d'un processeur ARM quadricœur, de 4 Go de mémoire RAM et d'un processeur graphique avec 128 cœurs CUDA, le module Jetson Nano est une solution de choix pour étudier l'IA et les réseaux neuronaux. Les applications possibles vont de la reconnaissance vocale à la classification d'objets dans des images vidéo capturées, en passant par le contrôle intégral de robots. De nombreux « accessoires » sont disponibles

pour tous les domaines – par ex. le kit JetBot AI v2.1 de SparkFun, doté de la plateforme Jetson Nano.

Le processeur graphique fait toute la différence

Concernant le Jetson Nano de Nvidia, la comparaison avec le populaire Raspberry Pi 4B (fig. 4) va de soi. L'un et l'autre sont des ordinateurs monocarte, possèdent quatre cœurs de processeur et sont proposés avec 4 Go de mémoire RAM. Ils ont également comme points communs : la possibilité d'installer le système Ubuntu 20.04, la présence d'un connecteur d'extension. Le Raspberry Pi 4 possède un processeur graphique (GPU) Broadcom VideoCore6. Le Jetson Nano, pour sa part, dispose d'un GPU à base Maxwell avec 128 cœurs CUDA (GM20B). Ces cœurs sont à l'origine des performances supérieures du Jetson Nano. Les cœurs CUDA sont constitués d'unités arithmétiques programmables incorporées dans le processeur graphique, mais sans exclusivité de traitement puisqu'ils peuvent aussi exécuter des opérations arithmétiques

génériques en parallèle (par ex. celles nécessaires à l'entraînement et à la mise en œuvre d'un réseau neuronal).

Entraînement du réseau neuronal avec le Jetson Nano

C'est ici que vous pouvez voir la différence avec un MAX78000. L'entraînement du réseau neuronal doit être effectué sur un PC Linux, le mieux étant de disposer d'une carte Nvidia compatible CUDA. À cet effet, vous pouvez utiliser PyTorch et d'autres outils logiciels. Le Jetson Nano, quant à lui, possède déjà tous les ingrédients pour entraîner lui-même un réseau neuronal. Certes, ce n'est pas aussi rapide que sur un PC actuel équipé d'une carte Geforce 1660 ou 1050Ti. Mais le Jetson Nano est à vous pour à peine 130 €. Une carte graphique coûte deux fois plus cher, et encore faut-il qu'elle soit disponible.

Nvidia fournit un grand nombre de tutoriels [4, 5] et, évidemment, des logiciels [6]. Ces contenus montrent, entre autres, comment entraîner les réseaux neuronaux pour reconnaître/traiter des images à l'aide d'une webcam sur le Jetson Nano de Nvidia.

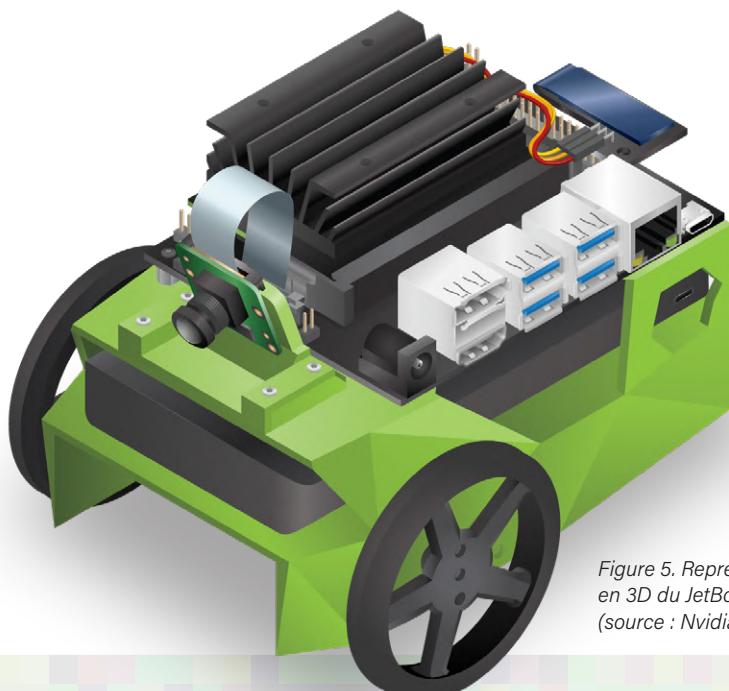


Figure 5. Représentation en 3D du JetBot de Nvidia (source : Nvidia).

Jetson Nano et le robot mobile JetBot

Une fois familiarisé avec l'environnement, il y a de fortes chances que l'envie d'une application concrète vous saisisse. En raison de l'accent mis sur le traitement des images, un robot autonome s'impose. Le JetBot (**fig. 5**) [7] appartient à cette catégorie. Adapté aux makers, il comporte une liste de pièces et de modèles pour l'impression 3D. Un robot mobile offrant la reconnaissance d'objets et les interactions permet de voir directement en action l'intelligence des réseaux neuronaux automatiquement entraînés. Le webinaire « AI for Makers: Learn with JetBot » montre comment entraîner un robot de ce type pour lui permettre d'interagir avec son environnement [8].

Développement dans un navigateur web

Pour développer le logiciel du JetBot, un simple navigateur web suffit. L'environnement de développement exécuté dans le navigateur est le JupyterLab (**fig. 6**) [9]. Il permet de créer, modifier et exécuter des scripts Python, mais aussi d'accéder à un grand nombre de bibliothèques tierces. Les pilotes de moteur ou les capteurs (fournis par ex. par Sparkfun ou Adafruit) peuvent être gérés directement en Python.

JupyterLab peut également lancer l'entraînement des réseaux neuronaux. Les nouvelles données d'entraînement produites pour les réseaux neuronaux servent à améliorer le comportement d'un robot grâce aux données issues du fonctionnement, mais aussi à ajouter de nouvelles fonctions. L'IDE est toujours accessible pour vous dans un navigateur.

Cependant, développer à partir de zéro la commande du robot et d'autres services est fastidieux et inutile. Vous pouvez faire appel pour cela au système d'exploitation ROS (*Robot Operation System*) [10] qui sert de base flexible pour tous les types de robots. Vous avez ainsi à votre disposition un noyau établi et une base logicielle bien documentée. Le système ROS, basé sur Linux, offre également des fonctions comme la création de cartes d'environnement. Une image appropriée du système d'exploitation avec ROS préinstallé est disponible pour le Jetson Nano de Nvidia et le robot JetBot.

Kit tout-en-un de SparkFun

Vous pouvez télécharger des images, préparer le Jetson Nano, vous procurer des composants pour le JetBot auprès de différentes sources, et éventuellement les imprimer en 3D. Il est également possible de configurer les différentes parties du logiciel et de n'aborder l'IA qu'ensuite. Il est en effet plus facile d'utiliser un kit prêt à l'emploi pour disposer immédiatement d'un matériel défini et fonctionnel et pouvoir ensuite se plonger pleinement dans l'IA. Le kit JetBot AI v2.1 de SparkFun (**fig. 7**) comprend tout le matériel dont vous avez besoin pour expérimenter l'IA et la robotique. Basé sur le robot JetBot de Nvidia, il donne accès à un vaste écosystème de tutoriels et de logiciels, renforcé par la documentation pratique de SparkFun à propos du matériel [11].

Il n'est pas non plus nécessaire d'assembler les composants du logiciel : le kit contient une carte SD prête à l'emploi où figure déjà tout le nécessaire. L'achat d'un tel kit peut sembler coûteux, puisqu'il faut débourser environ 270 €. Cependant, pour ce prix, vous avez le Jetson Nano, les pièces mécaniques du robot, les moteurs, les pilotes de moteur, un afficheur OLED, un adaptateur Wi-Fi, une caméra, une carte SD et toutes les pièces nécessaires pour vous lancer (**fig. 8**).

Et maintenant ?

Il y a beaucoup à apprendre et autant à découvrir. L'IA met en évidence des approches inédites et passionnantes pour doter un véhicule d'une commande autonome. Pour

la plupart des lecteurs, cependant, les notions de bibliothèque PyTorch et de système ROS doivent encore être replacées dans le bon contexte. PyTorch se charge de modifier et d'entraîner les processus d'IA. Comme tout logiciel, PyTorch doit être utilisé de manière appropriée et, de préférence, avec des fichiers de configuration, comme à l'accoutumée pour nombre de programmes de l'univers Linux. De même, le contrôle véhicules/robot est géré par le système ROS.

Le maniement de ces deux outils représente à lui seul un domaine suffisamment vaste pour écrire des livres entiers. Nous aborderons donc progressivement les différents composants et nous plongerons bientôt dans les détails !

(210318-04)

Contributeurs

Conception et texte : **Mathias Claußen**

Rédaction : Jens Nickel

Mise en page : Harmen Heida

Traduction : Pascal Godart

Des questions, des commentaires ?

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

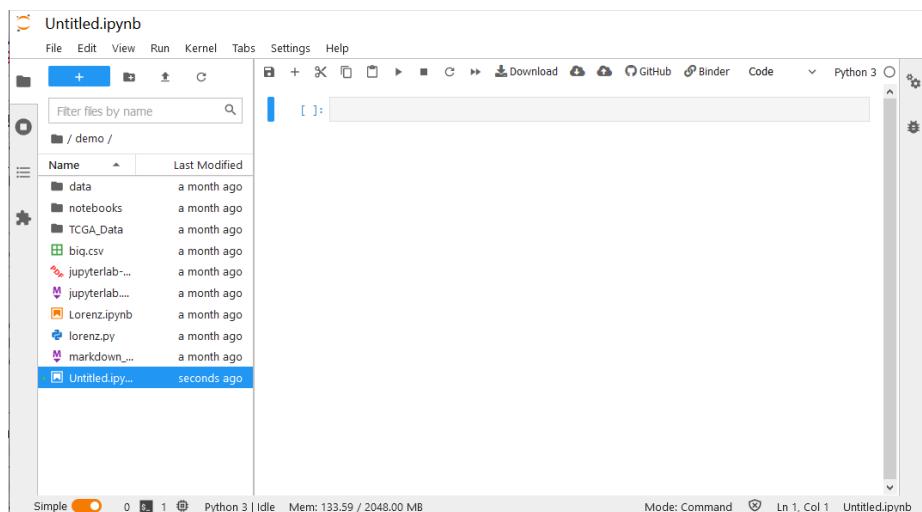


Figure 6. Jupyter Lab dans un navigateur.



Figure 7. Kit JetBot AI v2.1 de SparkFun
(source : SparkFun).



Figure 8. Composants du kit JetBot AI v2.1 de SparkFun
(source : SparkFun).



PRODUITS

- **Module Jetson Nano de Nvidia**
www.elektor.fr/nvidia-jetson-nano-developer-kit
- **Kit AI JetBot v2.1 de SparkFun (avec kit de développement NVIDIA Jetson Nano inclus)**
www.elektor.fr/sparkfun-jetbot-ai-kit-v2-1-incl-nvidia-jetson-nano-developer-kit
- **Kit AI JetBot v2.1 de SparkFun (sans kit de développement NVIDIA Jetson Nano inclus)**
www.elektor.fr/sparkfun-jetbot-ai-kit-v2-1-without-nvidia-jetson-nano-developer-kit

LIENS

- [1] **Walter Trojan, « Intelligence artificielle pour les \pm nuls (1) »**, ElektorMagazine.fr :
www.elektormagazine.fr/articles/intelligence-artificielle-pour-debutants-1
- [2] **Luc Lemmens, « Applis d'IA avec la carte MAX78000 au format Feather : les fondamentaux du matériel »**, ElektorMagazine.fr :
www.elektormagazine.fr/news/applis-ia-avec-la-carte-max78000-au-format-feather
- [3] **Nvidia Jetson Nano** : <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>
- [4] **Cours en ligne sur Jetson AI** :
https://developer.nvidia.com/embedded/learn/jetson-ai-certification-programs#course_outline
- [5] **Fondamentaux sur Jetson AI, Youtube** : www.youtube.com/watch?v=uvU8AXY1170&t=322s
- [6] **Centre de téléchargement Jetson** : <https://developer.nvidia.com/embedded/downloads#?search=Jetson%20Nano>
- [7] **Nvidia JetBot sur GitHub** : <https://github.com/NVIDIA-AI-IOT/jetbot>
- [8] **« AI for Makers - Learn with JetBot »** : www.youtube.com/watch?v=zOCSRzDUI-Y&t=2248s
- [9] **JupyterLab** : <https://jupyter.org/>
- [10] **ROS.org** : www.ros.org/
- [11] **Kit AI JetBot v2.1 de SparkFun, animé par Jetson Nano** : www.sparkfun.com/products/16417

zoom sur les transformateurs secteur

Comportement transitoire à l'allumage et l'extinction

Andreas R. Fecht (Allemagne)

À la mise sous tension d'un transformateur secteur, un courant d'appel circule. Son intensité dépend de l'angle de phase de la tension sinusoïdale d'entrée à cet instant précis.

Ce courant dépend aussi de l'angle de phase à la mise hors tension précédente. Le but de cet article n'est pas de concevoir un circuit palliant ce courant d'appel, mais seulement d'étudier cet effet en détail. Nous n'avons trouvé que très peu d'information détaillée sur ce sujet, et cette recherche a été menée afin de combler à l'avenir cette lacune.

La rémanence du noyau de fer entraîne quelques considérations théoriques sur le comportement des transformateurs. Seuls les essais permettent de déterminer le degré d'accord entre la théorie et la pratique. Les possibles différences de comportement entre transformateurs toriques et ceux à entrefer sont également abordées ici.

Banc d'essai

Pour étudier en détail les courants du transformateur, nous avons conçu un montage adapté. La **figure 1** en donne le schéma fonctionnel. Au cœur de ce montage, le microcontrôleur et le commutateur MOSFET commandé.

L'électronique de mesure et le microcontrôleur sont alimentés sous 3,3 V régulés. Un optocoupleur commande la grille du MOSFET avec une alimentation de 12 V à isolation galvanique.

L'électronique de mesure se compose de deux AOP rail à rail, de bande passante appropriée. Un diviseur résistif ramène la tension du secteur à une valeur convenable et une résistance shunt (dont la valeur permet d'aller jusqu'à ± 75 A) mesure le courant.

Doté de deux CA/N à 12 bits, le microcontrôleur acquiert simultanément les deux valeurs à une fréquence d'échantillonnage suffisamment élevée (environ 20-50 kHz). Quatre optocoupleurs



relient le microcontrôleur au PC et transmettent les mesures en RS232.

Le circuit d'essai en détail

Le schéma du banc d'essai est présenté à la **figure 2**. Trois tensions séparées galvaniquement sont nécessaires pour alimenter le circuit en énergie.

L'alimentation du circuit de mesure est confiée à un transformateur

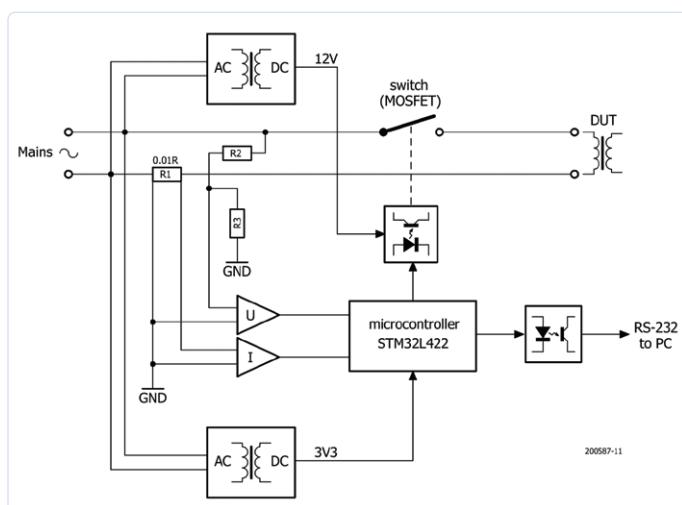


Figure 1. Schéma de principe du banc d'essai.

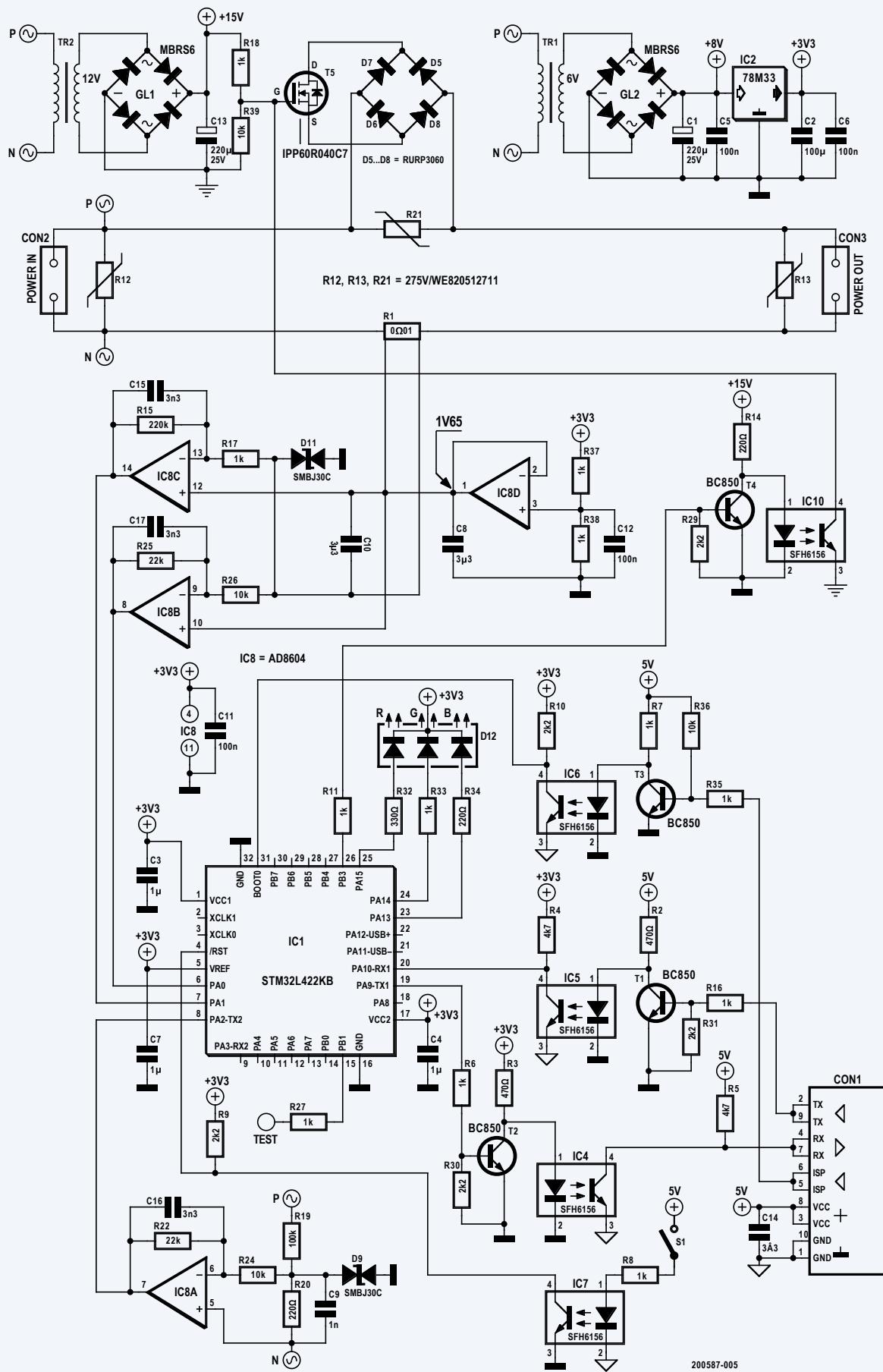


Figure 2. Le schéma du banc d'essai semble compliqué en raison des optocoupleurs qui assurent l'isolation galvanique.

zone D

Astuces, bonnes pratiques et autres informations pertinentes

Tableau 1 : Résistance de contact des bornes, mesurée pour différentes valeurs de courant.

Courant [A]	Résistance du commutateur fermé [Ω]
1	1,5
2	0,85
5	0,4 (100 mΩ différentiel)
10	0,24 (80 mΩ différentiel)

6 V / 1 W suivi d'un redresseur, un condensateur électrolytique, et un régulateur de tension 3,3 V.

Un transformateur 12 V / 1 W avec redresseur et filtrage produisant une tension continue non régulée d'environ 15 V alimente la commande de grille du MOSFET.

Le côté PC de l'interface RS-232 est alimenté en direct par un convertisseur USB-RS-232 5 V.

Interruuteur marche/arrêt

L'interrupteur marche/arrêt est un MOSFET (T5) avec un pont redresseur (D5 à D8). Un MOSFET peut être activé ou désactivé à tout moment, ce n'est pas le cas d'un thyristor ou d'un triac. Le MOSFET choisi peut supporter un courant continu de 73 A et des pointes de 211 A. Les diodes du pont redresseur supportent des courants continus de 30 A et des pointes de 70 A. Trois varistances avec une tension d'amorçage de 430 V assurent la protection contre les tensions induites.

Le MOSFET est commandé par un optocoupleur. Le temps de montée du front mesuré est de 20 μ s, c'est plus que suffisant pour ce type de mesure.

Les tableaux 1 et 2 donnent respectivement les valeurs mesurées et extrapolées de la résistance de contact aux bornes du circuit pour différentes intensités de courant.

Prise de mesures

Le microcontrôleur STM32L422KB choisi est doté de deux convertisseurs analogique-numérique de 12 bits. Ils permettent d'échantillonner simultanément la tension et le courant.

Pour mesurer des valeurs positives et négatives, IC8D crée une masse virtuelle à la moitié de la tension d'alimentation (1,65 V). C'est le point de référence des AOP IC8A/B/C. Pour l'amplificateur de tension IC8A, la tension du secteur est d'abord divisée par 455 par le pont R19/R20. L'amplificateur a un gain de 2,2 et décale en outre la tension de 1,65 V. La plage de tension d'entrée est donc de $\pm 341,6$ V. L'amplificateur de courant IC8B amplifie la tension aux bornes de la résistance shunt d'un facteur 2,2 et décale en outre la tension amplifiée de 1,65 V. Cela donne une plage de courant de ± 75 A avec une résolution de 36 mA environ. IC8C augmente la précision de mesure des courants les plus faibles en portant à 220 le gain en tension. Sa sortie est reliée à une autre entrée du CA/N du MCU. Ainsi on peut mesurer des courants jusqu'à $\pm 0,75$ A avec une résolution d'environ 0,37 mA. La plus petite puissance crête détectable à 230 VAC est donc d'environ 0,1 W.

Tableau 2. Valeurs extrapolées de la résistance de contact.

Courant [A]	Résistance du commutateur fermé [mΩ]
20	140 (40 mΩ différentiel)
50	68 (20 mΩ différentiel)
100	40 (12 mΩ différentiel)

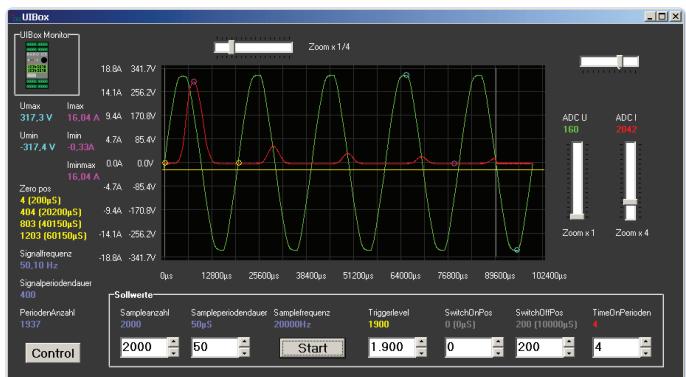


Figure 3. Le logiciel sur PC en action affiche la tension du réseau (vert) et le courant (rouge). Ici, la tension a été appliquée à la charge pendant un passage par zéro (ligne grise verticale de gauche) et retirée neuf passages par zéro plus tard (ligne grise verticale de droite).

Interface PC

L'interface PC est dotée de quatre optocoupleurs : deux pour les lignes TX et RX d'un port RS-232, un pour commander la broche de programmation in-system (ISP) du MCU et un pour réinitialiser le système de manière sûre. Les optocoupleurs utilisés permettent une vitesse de transmission maximale de 19200 bauds.

Circuit imprimé

Le schéma a été transposé sur un circuit imprimé double face. Sur la face inférieure, le chemin de fort courant a été renforcé de fils de cuivre d'1,5 mm². La résistance du circuit est de 11 mΩ environ pour le chemin de mesure du courant et 240 mΩ environ à 10 A entre les bornes de l'interrupteur. La résistance différentielle entre les bornes de l'interrupteur est d'environ 80 mΩ à 10 A.

Logiciel

Pour les expériences sur les transformateurs, le logiciel se compose d'un micrologiciel pour le MCU (enregistrement des données brutes) et d'un logiciel sur PC pour les traiter.

Le micrologiciel du MCU a été créé avec STM32Cube de STMicroelectronics. Le langage de programmation utilisé est le C99. Nous avons choisi le compilateur C ARM-GNU. Il est disponible comme dérivé d'Eclipse dans STM32Cube. Le logiciel pour PC (fig. 3) a été créé avec Delphi et donc écrit en Pascal.

Les données du MCU sont transférées par RS-232 vers un PC à l'aide

$$dW = \int [4(x_0, y_0, z_0)] dV = 44 dV$$

$$e^{ix} = \cos x + i \sin x$$

d'un convertisseur de niveau USB-RS-232. Pour chaque mesure individuelle, l'évolution de la tension et du courant est en outre affichée pour le contrôle. La valeur de la tension qui déclenche les mesures (ligne jaune) est réglée légèrement négative pour garantir l'enregistrement du 1^{er} passage par zéro de la tension.

Mesures

Le transfo a été allumé et éteint en couvrant tous les angles de phase de la tension du réseau CA, pour toutes les mesures. Paramètres utilisés :

- Pas de l'angle de phase : 5°
- Transformateur allumé pendant au moins 25 périodes (0,5 s)
- Pause entre deux mesures : 0,5 s
- Fréquence d'échantillonnage : 20 kHz

Le courant est enregistré sur deux périodes de tension complètes. Après chaque série de mesures, les deux maxima (positif et négatif) sont extraits et la valeur absolue la plus grande est retenue comme résultat. Les trois premières mesures de chaque cycle sont exclues pour écarter toute magnétisation antérieure résiduelle du noyau.

Pour un angle de phase final donné, le logiciel sur PC enregistre les 72 séries dans un fichier HTML en un tableau de 72 lignes que l'on peut ensuite facilement importer dans un tableur. Avec un pas angulaire de 5°, on obtient un ensemble de $72 \times 72 = 5184$ points qui s'affiche en 3D.

Exemple : mesure simple

Il s'agit d'une mesure sans charge pour l'impulsion de courant la plus forte (fig. 3). C'est le scénario le plus défavorable. Il se produit quand la tension est activée au début de la courbe à un passage par zéro (ligne grise verticale à gauche) et désactivée quatre périodes et demie plus tard, à nouveau à un passage par zéro (ligne grise verticale à droite).

L'impulsion de courant la plus forte se produit au 1^{er} passage par zéro. Cependant, elle n'est pas suffisante pour annuler la magnétisation du noyau du transformateur déjà présente. Les impulsions disparaissent progressivement après plusieurs passages par zéro. Cet effet est clairement audible : un bref bourdonnement dure environ une demi-seconde à la mise sous tension.

Transformateur torique avec charge résistive

Nous avons effectué cet essai avec un transformateur torique ELNA 100 VA (primaire de 230 V et secondaire de 2×24 V), voir figure 4. Comme charge résistive, nous avons connecté une résistance de 10 Ω / 50 W à chaque secondaire. Une plaque d'aluminium de montage assurait la dissipation thermique de cette charge totale d'environ 115 W (légère surcharge des résistances et du transfo). Les résultats sont présentés en 3D à la figure 5 (sans charge) et à la figure 6 (avec charge).

Transformateur EI avec charge résistive

Nous avons refait ces essais avec un transformateur à noyau EI de 120 VA (primaire 230 V et secondaire 2×15 V avec points



Figure 4. Transformateur torique de 100 W utilisé pour les expériences.

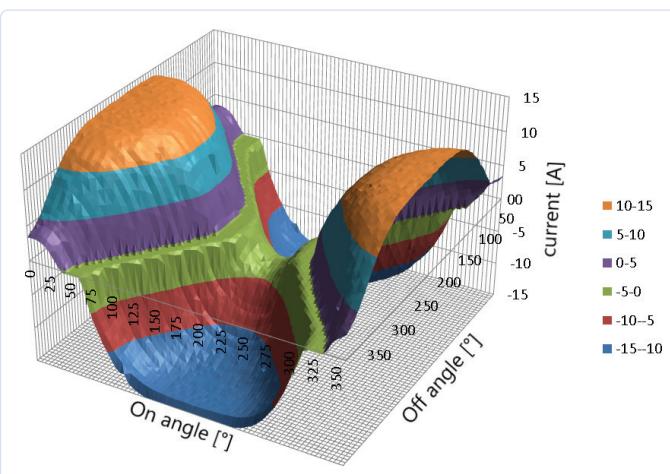


Figure 5. Tracé 3D du courant de phase pour le transformateur torique sans charge.

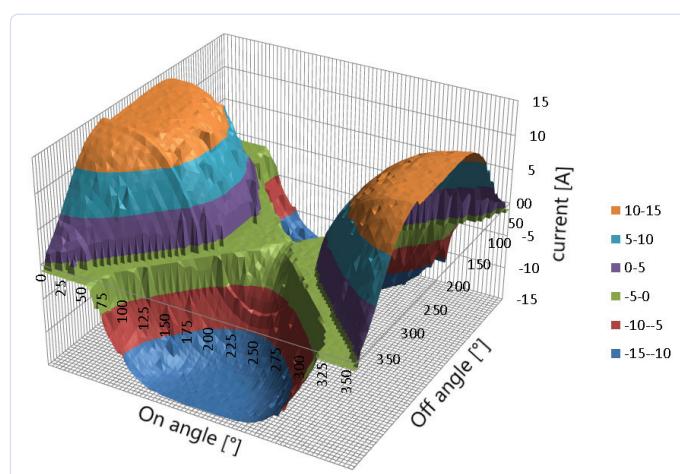


Figure 6. Charge résistive. Notez la discontinuité à l'angle de coupure d'environ 300°. Pour une raison inconnue, le fusible a sauté. Avant la poursuite de l'expérience, le transformateur s'était un peu refroidi, entraînant un courant légèrement plus élevé après le redémarrage. La raison de cette augmentation du courant pourrait être la résistance du fil de cuivre du primaire qui dépend de la température.

zone D

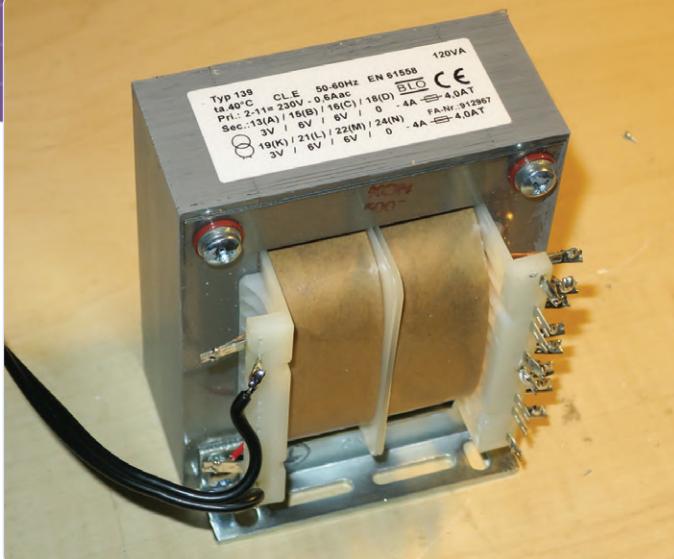


Figure 7. Ce transformateur à noyau EI de 120 W se comporte-t-il différemment du transformateur torique ?

$$F(h, k, \ell) = A + i \cdot B \quad F(h, k, \ell, t) \geq h = t$$

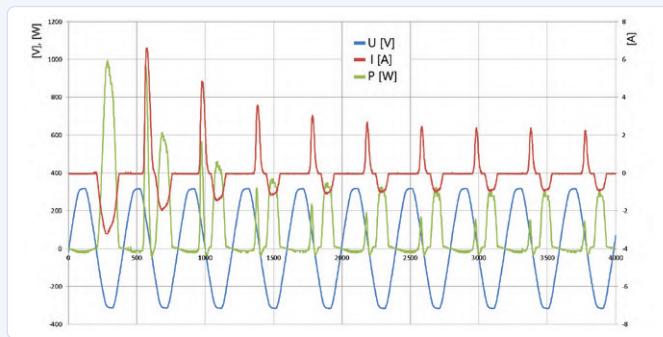
intermédiaires) du fabricant Block (fig. 7). Comme charge résistive, nous avons connecté deux résistances de $10 \Omega / 50 \text{ W}$ en parallèle (soit 5Ω) à chaque secondaire de 15 V. Comme ci-dessus, nous les avons vissées sur dissipateur thermique. La charge totale est de 90 W et le transfo est un peu sous-chargé. Les figures 8 (sans charge) et 9 (avec charge) donnent les résultats en 3D.

Redressement demi-onde des deux secondaires

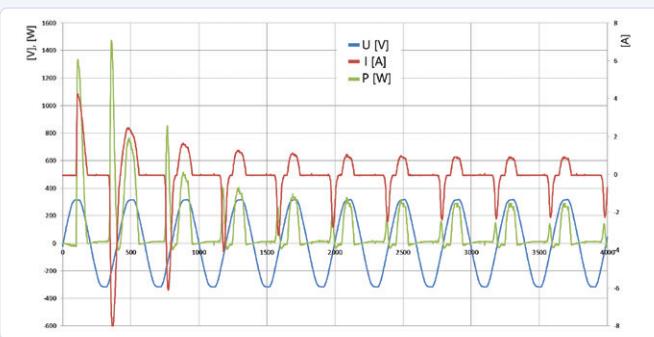
Toutes les autres expériences ont été réalisées sur le transformateur torique. Une diode (redresseur monoalternance) a été connectée à chaque secondaire du transformateur. Les enroulements sont redressés en polarités opposées, ce qui entraîne une charge magnétique symétrique sur le transformateur, comme ce serait le cas avec un redresseur bialternance. Chaque diode est connectée à une

CHARGE ASYMÉTRIQUE

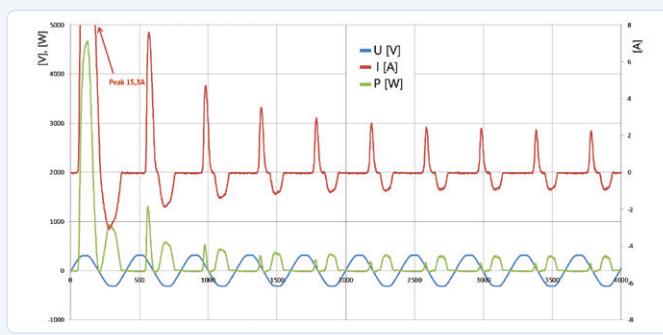
Les expériences suivantes utilisent le circuit de la figure 10, mais avec un seul secondaire chargé à la fois. Chaque expérience a été réalisée deux fois, une fois avec la charge sur S1 ('A') et ensuite avec la charge inversée sur S2 ('B'). Les conditions de départ étaient les mêmes que dans les expériences correspondantes avec charge symétrique.



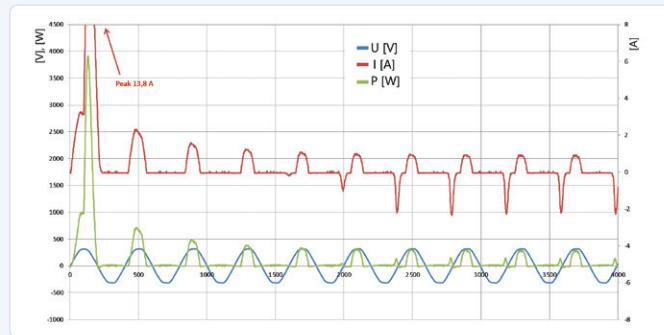
Meilleur cas A. Seule l'énergie des impulsions de courant négatifs circule dans la charge. Les impulsions de courant positives se stabilisent lentement à une valeur d'environ 2 A. La rémanence du noyau du transformateur est quelque peu perturbée par la charge asymétrique du côté secondaire. L'énergie transportée par les impulsions de courant positif n'est pas négative : elle est dissipée comme puissance active dans le transfo et non dans la charge.



Meilleur cas B. La plus grande impulsion de courant se produit pendant la 2^e alternance. L'énergie de la 1^e pointe passe dans la charge, l'énergie de la 2^e dans le transformateur. Ici aussi, les impulsions de courant se stabilisent progressivement à environ 2 A sans tomber à zéro.



Pire cas A. Là encore, le pic d'allumage est similaire à celui du circuit ouvert. La même condition que dans le meilleur cas s'installe progressivement.



Pire cas B. Ce scénario ne produit pas la plus grande impulsion de courant. Cela est probablement dû au fait que la charge asymétrique du côté secondaire compense un peu la rémanence stockée dans le noyau pendant quelques périodes. Cependant, au bout de cinq périodes, le montage revient au comportement précédent.

$$dW = \int [\psi(x_0, y_0, z_0)] dV = 44 dV$$

$$e^{ix} = \cos x + i \sin x$$

$$F(h, k, c)$$

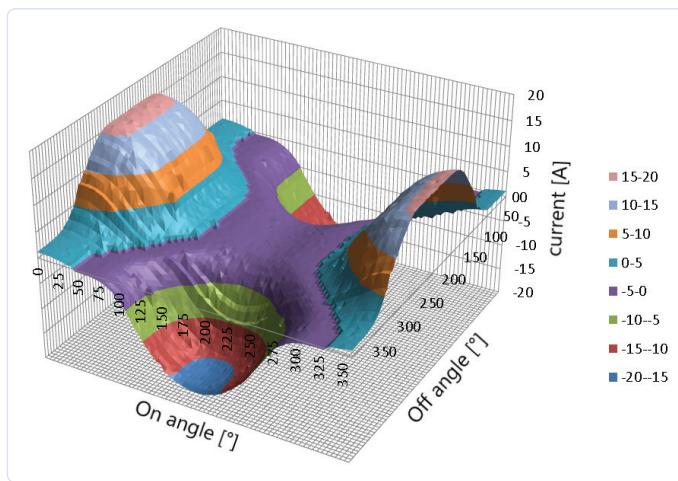


Figure 8. Tracé 3D pour le transformateur à noyau EI non chargé.

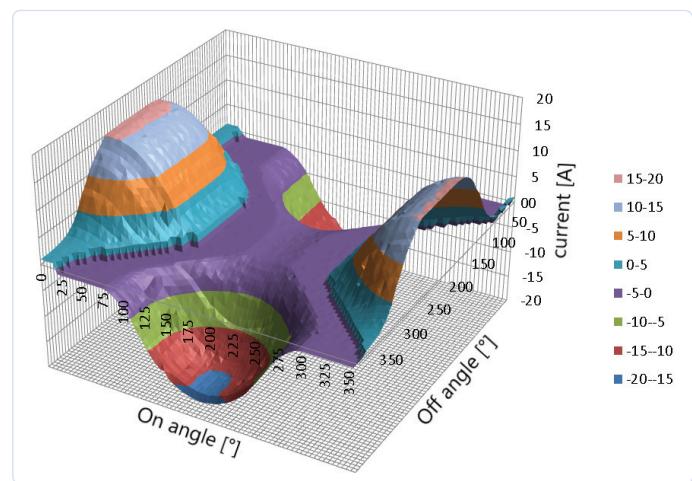


Figure 9. Transformateur à noyau EI avec charge résistive de 90 W.

résistance de $20 \Omega / 100 \text{ W}$ avec un condensateur électrolytique de $10\,000 \mu\text{F}$ en parallèle (fig. 10).

Meilleur cas : angles de commutation $90^\circ/90^\circ$

Allumage du transformateur au 1^{er} maximum de tension (après extinction exactement à un tel maximum). Ceci correspond au cas le plus favorable, où aucun effet magnétique supplémentaire ne devrait se produire comme dans les expériences précédentes.

Comme prévu, les résultats (fig. 11) montrent que le maximum de courant circule au maximum de tension (recharge du condensateur). La 1^{re} impulsion est la plus forte, car le condensateur est alors vide. Les impulsions décroissent lentement pour se stabiliser autour de $0,9 \text{ A}$ (environ deux fois le courant nominal à pleine charge). La courbe de puissance (verte) n'est jamais négative. La puissance réactive convertie est quasi nulle.

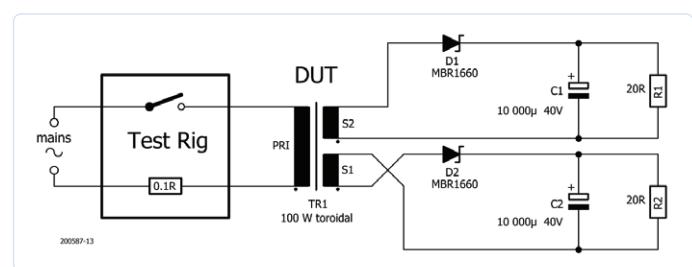


Figure 10. Circuit utilisé pour les expériences de charge symétrique. Pour les expériences de charge asymétrique, un seul secondaire était chargé à la fois.

Pire cas : angles de commutation $0^\circ/180^\circ$

Allumage du transformateur au 1^{er} passage à zéro de la tension après une extinction au passage à zéro opposé. Comme observé

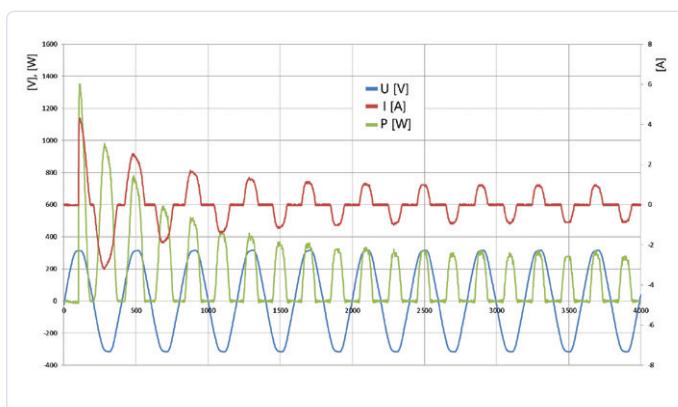


Figure 11. Charge symétrique dans le meilleur des cas. La courbe de puissance (verte) n'est jamais négative. La puissance réactive convertie est quasi nulle.

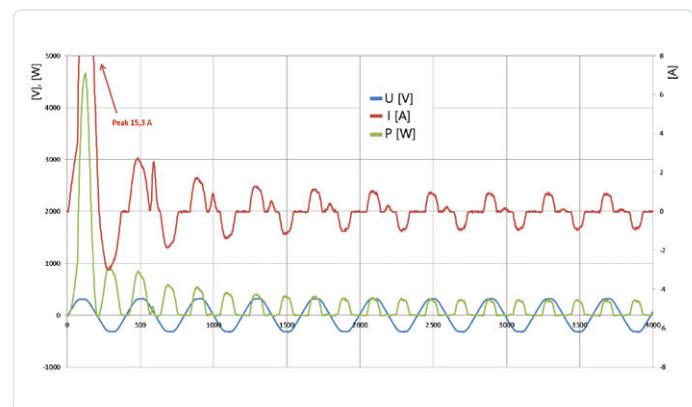


Figure 12. Charge symétrique dans le pire des cas. La puissance convertie dans le premier pic est remarquable pour un petit transfo de 100 W : presque 4,65 kW ! La valeur est positive, c'est une puissance active pure.

zone D

Astuces, bonnes pratiques et autres informations pertinentes

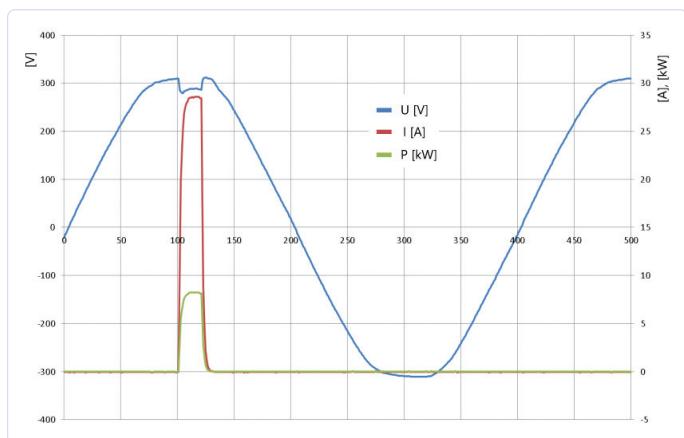


Figure 13. Pour l'impédance du réseau, une valeur d'environ 640 mΩ a été trouvée.

précédemment, c'est le cas le plus défavorable, où les effets de rémanence magnétique les plus forts sont à prévoir. Le pic de courant de 15,3 A à l'allumage (fig. 12) est proche du cas du circuit ouvert. Les pics qui suivent sont aussi similaires et disparaissent au bout de 10 à 20 périodes. La conversion de puissance réactive est quasi nulle.

Impédance du secteur

La mesure de l'impédance du secteur se fait par commutation d'une résistance de $10 \Omega / 50 \text{ W}$ pendant 1 ms environ sur le pic de tension du secteur. La chute de la tension secteur observée permet de calculer son impédance, voir figure 13.

La pointe de tension du réseau est mesurée peu avant la mise sous tension (310 V) et environ 0,5 ms après le front de mise sous tension (289 V). Le courant qui circule ici est de 28,5 A (puissance de crête maximale de 8,25 kW). L'impédance est alors de $(310 - 289) / 28,5 = 740 \text{ m}\Omega$. Le circuit lui-même présente une résistance de contact d'environ 100 mΩ à ce courant. Ainsi, la valeur de l'impédance du réseau est d'environ 640 mΩ.

Principaux points à retenir

Après analyse des résultats, on peut tirer les conclusions suivantes :

- Le courant d'appel dépend fortement de l'angle de phase auquel la tension a été précédemment coupée.

- La mise sous tension à un extremum est toujours convenable, quel que soit le type de transformateur. Cependant, il doit s'agir du même extremum (max. ou min.) que lors de la mise hors tension précédente. Sinon, le courant est à peine tolérable (environ 5× le courant normal à pleine charge). Les transformateurs toriques sont un peu plus brutaux.
- L'allumage lors d'un passage à zéro n'est pas souhaitable. L'extinction lors du passage par zéro de sens opposé produit l'impulsion de courant la plus forte (environ 30× le courant normal à pleine charge). Dans le même sens de passage par zéro, c'est moins critique (environ 10× le courant normal à pleine charge).
- Les transformateurs non chargés produisent des appels de courant beaucoup plus forts que lorsqu'ils sont chargés. Sous charge, la marge de tolérance sur l'angle de phase correct est nettement plus grande.
- Le comportement des transformateurs sous charge purement résistive ou redresseur + capacité de filtrage est comparable.

Si vous souhaitez approfondir les expériences présentées ici, toutes les données d'essai, les fichiers de conception du banc et le logiciel sont téléchargeables [1].

(200587-04)

Des questions, des commentaires ?

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

Contributeurs

Idée, conception, texte et illustrations : **Andreas R. Fecht**
Rédaction et illustrations : **Clemens Valens**
Schéma : **Patrick Wielders**
Mise en page : **Giel Dols**
Traduction : **Yves Georges**



Produit

- Livre en anglais, « **Advanced Programming with STM32 Microcontrollers** »
www.elektor.fr/19520

LIEN

- [1] Données d'essai, fichiers de conception et logiciels :
www.elektormagazine.fr/200587-04

« Yes We CAN » avec PiCAN 3

Module de bus CAN pour le Raspberry Pi 4

Tam Hanna (Slovaquie)

Beaucoup de données ont circulé depuis la publication par Bosch de la spécification du bus CAN (Controller Area Network) en 1986 et sa première utilisation par Mercedes-Benz dans son modèle Classe S en 1991. Depuis, la notion de réseau CAN fait partie intégrante du développement et de la conception des automobiles. Le bus CAN est robuste, d'où son utilité dans de nombreux autres domaines d'application. Un module de bus CAN mobile est utile pour analyser les communications, diagnostiquer les problèmes et commander des dispositifs. Le HAT PiCAN examiné ici fait parfaitement l'affaire et se branche sur la toute dernière version du Raspberry Pi. Voyons cela de plus près.

Les systèmes électriques des véhicules sont des environnements hostiles pour les appareils électroniques délicats. J'ai le souvenir de nombreuses interventions de recherche de pannes consistant à poser des câbles de 40 m de long pour alimenter des oscilloscopes et des équipements de test fonctionnant dans une voiture. Si la simple évaluation des données du bus CAN vous intéresse, le PiCAN 3 est une autre ressource de diagnostic avantageuse et portable.

Pourquoi utiliser un Raspberry Pi ?

Lorsque vous expérimentez un système de communication par bus (I2C, SPI, etc.), les problèmes ne découlent pas de la mauvaise qualité du signal dans la couche physique, mais de l'organisation et du séquençage des données transmises. Si vous envoyez des paquets mal formés ou si vous vous trompez sur la façon dont les octets doivent être séquencés, par exemple, vous ne serez guère surpris si le moteur ou tout autre dispositif que vous espérez commander refuse de coopérer.

Des outils tels que PiCAN 3, associés à un Raspberry Pi, permettent d'effectuer des mesures mobiles. Dans l'idéal, vous n'avez besoin que d'un Raspberry Pi 4 et d'un petit écran pour construire un banc d'essai portable. Si vous intégrez l'environnement de développement du système CAN dans la distribution ARMBian du système, vous serez opérationnel.

Le matériel

L'environnement électrique d'une voiture peut être assez hostile pour les circuits électroniques. Le fabricant SK Pang en a tenu

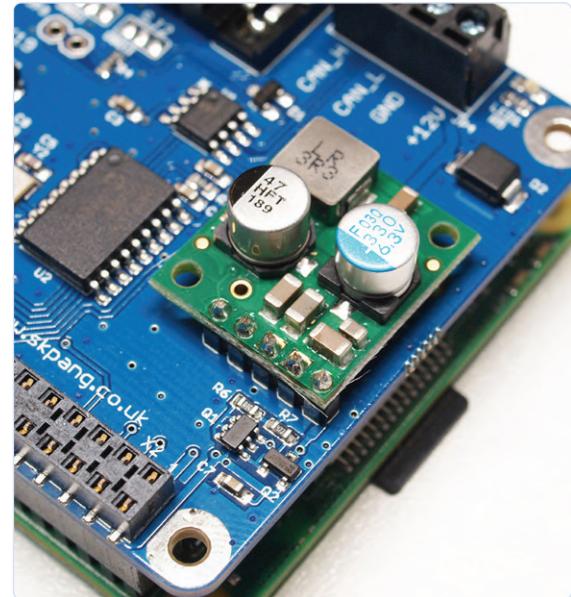


Figure 1. Cinq broches au pas de 0,1" relient la carte d'alimentation (verte) à la carte principale.

compte dans la conception de la carte en incorporant une alimentation à découpage (fig. 1) avec une large plage de tensions d'entrée (6 à 20 V continu). Elle se connecte à la carte via un connecteur à cinq broches. Le HAT PiCAN devrait être disponible sans l'alimentation à découpage – mais actuellement, ce n'est pas le cas et l'alimentation est fournie.

Ce tout dernier HAT CAN est baptisé « PiCAN 3 – carte à bus CAN pour Raspberry Pi 4 avec alim. à découpage 3 A et horloge en temps réel » pour le différencier de ses prédecesseurs. Cette dernière version inclut donc une alimentation à découpage de 3 A pour alimenter le Raspberry Pi 4, très gourmand en énergie. Il est possible de faire fonctionner le HAT directement à partir du Raspberry Pi lorsqu'il est alimenté par son port USB-C.

Le schéma de la carte peut être consulté en [1] et montre qu'elle fait appel à des circuits intégrés standard. Le manuel d'utilisation est disponible en [2].

L'interface du bus de signaux CAN est réalisée avec le MCP2515, qui communique avec le Raspberry Pi via SPI et une broche d'interruption GPIO. L'interface physique avec le bus est assurée par l'émetteur-récepteur de bus CAN MCP2562. Le schéma du circuit PiCAN 3 montre que l'alimentation de 3,3 V utilisée par certains composants de la carte provient du régulateur du Raspberry Pi, ce qui permet d'économiser un régulateur supplémentaire.

La carte possède deux connecteurs (fig. 2) pour relier la carte PiCAN 3 à un bus CAN externe. La première méthode fait appel à un bornier à quatre voies dont l'identification des signaux est imprimée sur la carte. La deuxième méthode utilise une prise sub-D



Figure 2. La carte PiCAN 3 permet de se connecter à un bus CAN de deux façons.

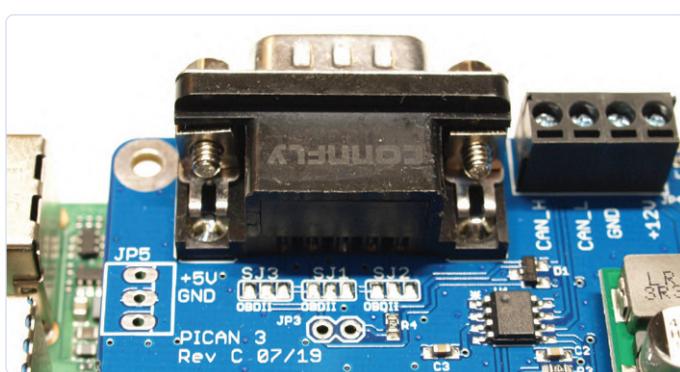
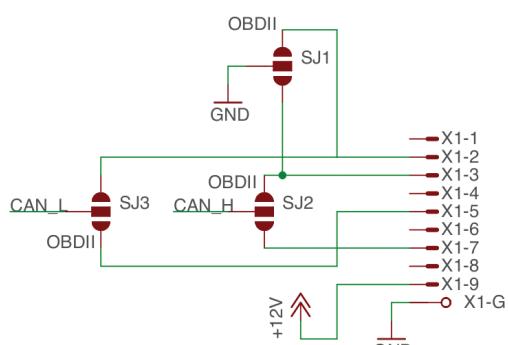


Figure 3. Grâce aux généreuses pastilles de soudure, le routage des signaux n'est pas trop compliqué.

à 9 broches, que les lecteurs d'un certain âge reconnaîtront comme celle des communications RS232 standard.

Le connecteur sub-D à 9 broches accepte un câble standard OBD-II vers DB9 pour se connecter à un système OBD. Malheureusement, il n'existe pas d'affectation standard des broches du connecteur et des signaux CAN. Pour parer à toutes les éventualités de câblage,



	CAN ANALYSER	OBDII	SOLDER-JOINT
CAN H	7	3	SJ2
CAN L	2	5	SJ3
GND	3	2	SJ1

Figure 4. Le tableau montre comment ponter les plages de soudure.

il est possible de ponter les pastilles de la carte (fig. 3) avec de la soudure. Un plan d'affectation (fig. 4) vous aide à décider lesquelles relier pour que le connecteur soit compatible avec le système auquel vous vous connectez. Ces pastilles sont complètement ouvertes sur la carte livrée. Ainsi, sans les ponts de soudure, les signaux ne seront pas acheminés vers les bonnes broches.

Les applications véhiculaires nécessitent souvent des informations précises sur l'heure du jour. Pour cela, le PiCAN 3 utilise une horloge en temps réel PCF8523. Celle-ci communique avec le Raspberry Pi via une interface I²C. La carte comporte un support pour pile bouton qui accepte une pile de type CR1220 pour alimenter l'horloge en temps réel.

L'écosystème CAN

Le système d'exploitation Linux a une certaine pertinence pour le secteur automobile. Il existe en fait tout un écosystème dans l'univers Linux permettant de prendre en charge les applications de bus CAN. Les utilitaires disponibles englobent aussi bien les pilotes de noyau et les logiciels de ligne de commande que de très nombreux autres puissants outils.

Pour commencer à communiquer avec le PiCAN 3, nous devons effectuer quelques adaptations dans le fichier `/boot/config.txt` d'une installation Raspbian toute neuve. D'abord il faut activer le bus SPI avec le bloc suivant (notez l'appel supplémentaire `overlay`) :

```
dtparam=spi=on
dtoverlay=mcp2515-can0,oscillator=16000000,interrupt=25
dtoverlay=spi-bcm2835-overlay
```

Les déclarations suivantes sont nécessaires pour utiliser l'horloge en temps réel :

```
dtparam=i2c_arm=on
dtoverlay=i2c-rtc,pcf8523
```

L'étape suivante consiste à télécharger les modules du noyau et quelques autres utilitaires. Fort à propos, il existe un package prêt à l'emploi disponible dans les référentiels :

```
sudo apt-get install can-utils
```

Après le déploiement des utilitaires CAN, il faut enregistrer l'interface sur le système d'exploitation. Pour cela, il suffit de créer une nouvelle interface selon le schéma suivant. La valeur 500 000 indique ici le débit maximal de données admis par le matériel :

```
sudo /sbin/ip link set can0 up type can bitrate 500000
```

Une fois l'interface prête, vous pouvez l'utiliser comme tout autre produit similaire pour communiquer à l'aide du bus CAN. De manière classique, vous pouvez utiliser l'outil `candump`, activable à l'aide de la ligne de commande, comme suit :

```
candump
```

Une fois en fonctionnement, elle affiche automatiquement et en continu tous les messages CAN du HAT. C'est particulièrement utile par ex. pour décortiquer les stratégies de commande de moteur inconnues dans les systèmes existants.

Il existe également une API CAN Python que vous pouvez installer :

```
git clone https://github.com/hardbyte/python-can
cd python-can
sudo python3 setup.py install
```

Ajout de l'horloge en temps réel

Le système d'exploitation Linux met en œuvre depuis longtemps des horloges en temps réel sur les ordinateurs portables et les PC. Les systèmes sans composant de type horloge en temps réel (pour des raisons de coût) utilisent le module émulateur `fake-hwclock` du système d'exploitation pour fournir des informations sur l'heure ; c'est également le cas du Raspberry Pi.

Pour pouvoir utiliser l'horloge en temps réel matérielle sur la carte CAN, il faut d'abord désactiver la « fausse » `hwclock` pour l'empêcher d'interférer avec l'heure fournie par la « vraie » :

```
sudo apt-get -y remove fake-hwclock
sudo update-rc.d -f fake-hwclock remove
sudo systemctl disable fake-hwclock
```

Ensuite, ouvrez le fichier `/lib/udev/hwclock-set` (en mode super-utilisateur pour disposer des droits nécessaires) et mettez en commentaires les lignes de code dans ces deux blocs :

```
#if [ -e /run/systemd/system ] ; then
# exit 0
#fi

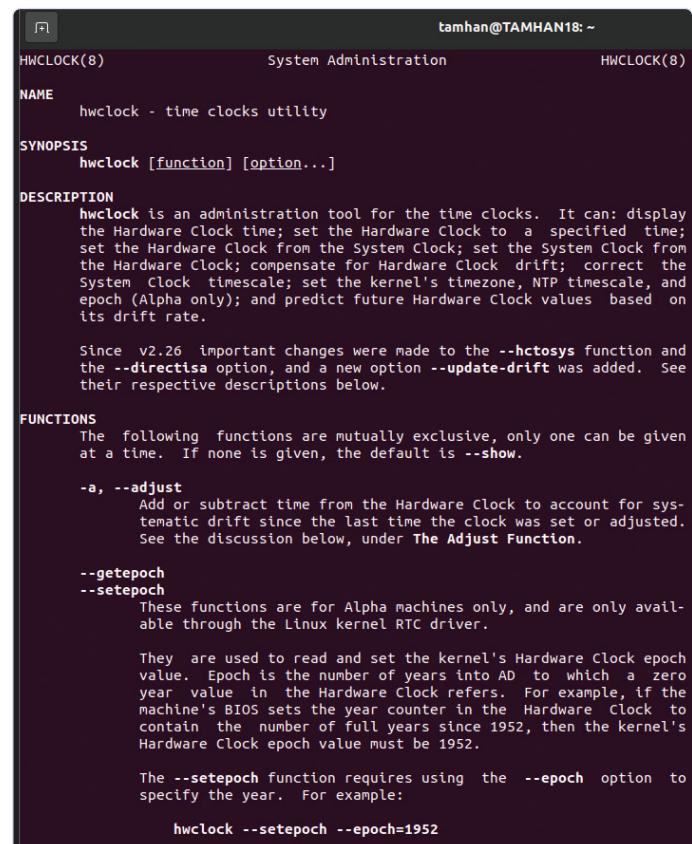
#/sbin/hwclock --rtc=$dev --systz --badyear
#/sbin/hwclock --rtc=$dev --systz
```

Les informations temporelles de l'horloge matérielle peuvent maintenant être lues à l'aide de `hwclock`. Les caractéristiques de cet outil d'administration de l'horloge matérielle sont présentées à la **figure 5**.

Fonctions de commande et de communication du bus CAN

Le HAT PiCAN 3 de SK Pang, avec son alimentation à découpage de 3 A embarquée, peut être utilisé avec un Raspberry Pi 4 pour offrir des capacités de communication et de commande par bus CAN. Avec le Raspberry Pi, il forme une plateforme mobile expérimentale compacte et relativement peu coûteuse, dotée de l'interface nécessaire pour se connecter directement à un bus CAN. Le matériel de la carte est *open source*, de sorte que le système peut être intégré à vos propres projets après évaluation, si nécessaire. 

(210303-04)



```
HW CLOCK(8)                               System Administration                               HW CLOCK(8)

NAME
    hwclock - time clocks utility

SYNOPSIS
    hwclock [function] [option...]

DESCRIPTION
    hwclock is an administration tool for the time clocks. It can: display
    the Hardware Clock time; set the Hardware Clock to a specified time;
    set the Hardware Clock from the System Clock; set the System Clock from
    the Hardware Clock; compensate for Hardware Clock drift; correct the
    System Clock timescale; set the kernel's timezone, NTP timescale, and
    epoch (Alpha only); and predict future Hardware Clock values based on
    its drift rate.

    Since v2.26 important changes were made to the --hctosys function and
    the --directisa option, and a new option --update-drift was added. See
    their respective descriptions below.

FUNCTIONS
    The following functions are mutually exclusive, only one can be given
    at a time. If none is given, the default is --show.

--a, --adjust
    Add or subtract time from the Hardware Clock to account for sys-
    tematic drift since the last time the clock was set or adjusted.
    See the discussion below, under The Adjust Function.

--getepoch
--setepoch
    These functions are for Alpha machines only, and are only avail-
    able through the Linux kernel RTC driver.

    They are used to read and set the kernel's Hardware Clock epoch
    value. Epoch is the number of years into AD to which a zero
    year value in the Hardware Clock refers. For example, if the
    machine's BIOS sets the year counter in the Hardware Clock to
    contain the number of full years since 1952, then the kernel's
    Hardware Clock epoch value must be 1952.

    The --setepoch function requires using the --epoch option to
    specify the year. For example:
        hwclock --setepoch --epoch=1952
```

Figure 5. Pour en savoir plus sur l'accès par la ligne de commande à l'horloge en temps réel matérielle, utilisez `man` (abréviation de « manuel ») <code>. Voici le résultat pour `hwclock`, avec des informations pour commander l'horloge en temps réel utilisée par la carte CAN.

Des questions, des commentaires ?

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

Contributeurs

Texte et illustrations : **Tam Hanna** Traduction : **Pascal Godart**
Rédaction : **Thomas Scherer** Mise en page : **Giel Dols**

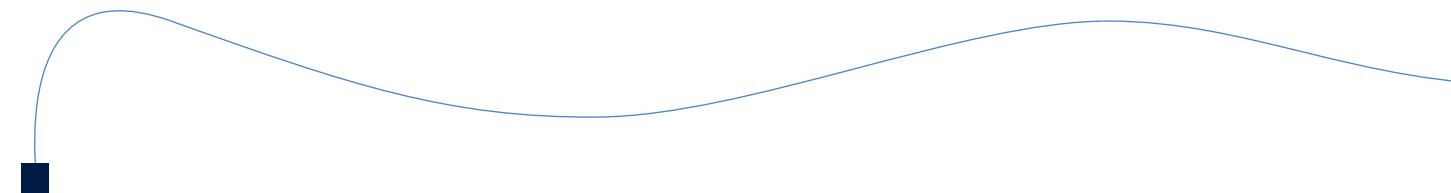


PRODUITS

- ▶ **PiCAN 3 - carte à bus CAN pour Raspberry Pi 4 avec alim. à découpage de 3 A et horloge en temps réel**
www.elektor.fr/19542

LIENS

- [1] **Schéma de la carte PiCAN 3** : https://cdn.shopify.com/s/files/1/0563/2029/5107/files/pican3_rev_C.pdf?v=1619981690
- [2] **Manuel de la carte PiCAN 3** : https://cdn.shopify.com/s/files/1/0563/2029/5107/files/PICAN3_UGA_10.pdf?v=1619981615



centrale solaire sur balcon

Un investissement vite amorti

Thomas Scherer (Allemagne)

Il n'est pas nécessaire de tapisser tout un toit de panneaux solaires pour avoir une installation économiquement et écologiquement viable. De nombreux employés de bureau travaillant désormais à domicile, l'idée d'une centrale solaire sur balcon couvrant le supplément de consommation d'énergie électrique pendant les heures de jour a fait son chemin. Ces minicentrales sont assez bon marché et bénéficient même de subventions dans certaines villes européennes. Après m'être informé sur les réglementations auxquelles elles doivent obéir et en avoir évalué quelques modèles, j'en ai installé une chez moi.

Depuis quelque temps, je regardais avec envie les panneaux photovoltaïques (PV) sur les toits de mon voisinage. À long terme, ils coquent toutes les cases pour réduire les émissions de CO₂ et répondre à notre voracité énergétique. Je ne vis dans une maison que depuis peu et je suis donc un peu en retard dans ce domaine. J'ai envisagé d'en installer un sur mon toit, mais le tarif de rachat du courant a tellement baissé que son amortissement prendrait bien trop de

temps. Et puis, cette installation causerait pas mal de tracas. Enfin, j'ai un travers : je n'aime guère confier à quelqu'un d'autre un travail (surtout en électricité) que je me sens capable de faire moi-même, ce qui me donne aussi l'occasion d'acquérir une expérience et des connaissances précieuses. En Allemagne, l'installation sur un toit d'un puissant générateur couplé au réseau implique beaucoup de paperasse et la taxation de l'électricité produite et consommée.

Cette idée me trottait dans la tête depuis tellement longtemps qu'après avoir installé une station de recharge solaire pour tondeuse robot [1] en 2020, j'ai commencé à évaluer les avantages d'une installation photovoltaïque relativement modeste couplée au réseau. Au cours de mes recherches pour savoir ce qui était disponible revenait souvent le terme « centrale solaire sur balcon » décrivant un système qui pouvait faire l'affaire (voir l'encadré « **Qu'est-ce qu'une centrale solaire sur balcon ?** »).

Est-ce que le compte y est ?

Après avoir parcouru les méandres de la réglementation, je me suis demandé si un tel mini-système solaire couplé au réseau



Qu'est-ce qu'une centrale solaire sur balcon ?

Une centrale solaire sur balcon est un petit générateur photovoltaïque. Comme son nom l'indique, elle est si petite que vous pouvez l'installer sur un balcon. Dans l'Union européenne, les installations dont la puissance de crête est inférieure à 800 Wc se situent en dessous du seuil à partir duquel la bureaucratie s'en mêle, avec sa paperasse et ses frais. C'est votre droit d'installer un tel système et personne ne peut vous en empêcher (tant que l'installation est sûre) ou l'écraser sous le poids de la bureaucratie.

Son but est également important : un balcon solaire produira juste assez d'électricité pour couvrir la consommation de base d'une maison. En d'autres termes, celle d'un réfrigérateur, d'un téléviseur, du routeur internet, de la pompe de circulation du système de chauffage ainsi que d'une pléiade de petits chargeurs et autres objets en veille. Toute l'énergie que le système produit en excédent est cadeau pour l'opérateur du réseau. Ainsi, le système reste simple et peu coûteux. C'est pourquoi il n'a pas intérêt à être surdimensionné.

pouvait être viable financièrement. Pour les installations sur balcon, on a le choix entre des systèmes à un ou deux panneaux. Avec un panneau, on peut atteindre une puissance de crête d'environ 300 Wc ('c' pour crête) et 600 Wc avec deux panneaux. Pour évaluer l'ensoleillement de ma région, j'ai étudié la carte [2] produite par le service météorologique allemand qui indique l'énergie solaire moyenne par an en Europe et plus particulièrement en Allemagne (fig. 1). Vous pouvez également trouver de nombreux calculateurs de rendement solaire sur l'internet. J'habite dans le sud du Land de Bade, une région très ensoleillée, où je peux compter sur une insolation annuelle de près de 1200 kWh par m² dans des conditions optimales. En supposant un rendement de 20% pour les cellules solaires monocristallines, je pourrais produire jusqu'à 240 kWh par m² et par an. Une installation à deux panneaux me conviendrait mieux. Un panneau moderne d'une puissance de 320 à 370 Wc a une surface utile d'environ 1,6 m². Avec deux panneaux, je pourrais donc compter sur une production annuelle de près de 770 kWh. Avec un prix de l'électricité en Allemagne de 30 centimes par kWh, ma centrale sur balcon me permettrait d'économiser jusqu'à 230 € par an. Le prix d'achat de mon

système étant presque de 600 €, il serait amorti en deux ans et sept mois !

Je suis tenté ! Pour que cela soit réaliste, il faut une orientation optimale des panneaux, sans ombre portée par les bâtiments et la végétation, ni de pertes dans l'onduleur et les câbles. Il faut aussi que toute l'énergie produite soit utilisée localement et que rien ne soit exporté vers le réseau. Dans le cas d'un appartement individuel dont l'occupant travaille à l'extérieur pendant la journée, une grande partie de l'énergie produite serait exportée vers le réseau. Un seul panneau de 300 W serait alors plus que suffisant. Chez moi, il y a deux réfrigérateurs et un congélateur ; ma mère aime regarder la télévision pendant la journée et je travaille à domicile sur un PC équipé d'un grand écran. Avec quelques autres appareils électriques, notre consommation de base excède la production d'un seul panneau de 300 W. Une installation de 600 Wc ne semble donc pas surdimensionnée.

En supposant un taux d'utilisation de 75% dans des conditions optimales, j'arrive à une économie annuelle sur les coûts d'électricité d'environ 170 €. L'emplacement que je prévois pour les panneaux n'est pas non plus idéal : il n'y aura pas d'ombre sur les panneaux, mais

l'angle d'élévation n'est que de 5° et ils seront orientés exactement au sud-sud-est à 135°. Je profiterai moins du soleil du soir, mais plus du rayonnement diffus. Si je soustrais 25% pour en tenir compte, je me retrouve avec une économie annuelle d'un peu plus de 125 €. Pour un coût de 600 €, cela donne un amortissement en quatre ans. Au-delà, c'est du bonus.

Si vous obtenez une subvention pour le système (la ville voisine de Fribourg, par exemple, donne 200 €), l'amortissement sera beaucoup plus rapide. Avec une installation par soi-même, les coûts sont minimes, mais cela ne tient pas compte des frais de remplacement du compteur que certains opérateurs peuvent exiger. Par exemple, la nécessité d'un compteur dit bidirectionnel fait actuellement l'objet d'un débat (un compteur doit au minimum être équipé d'un blocage anti-retour). Le matériel acheté doit également être fiable, les pièces défaillantes doivent être remplacées, ce qui rend le système beaucoup moins rentable.

Je pourrais augmenter l'angle des panneaux pour gagner en rendement, mais, à plat sur le toit existant, on les remarque moins et j'obtiens un meilleur rendement en été, lorsque la climatisation

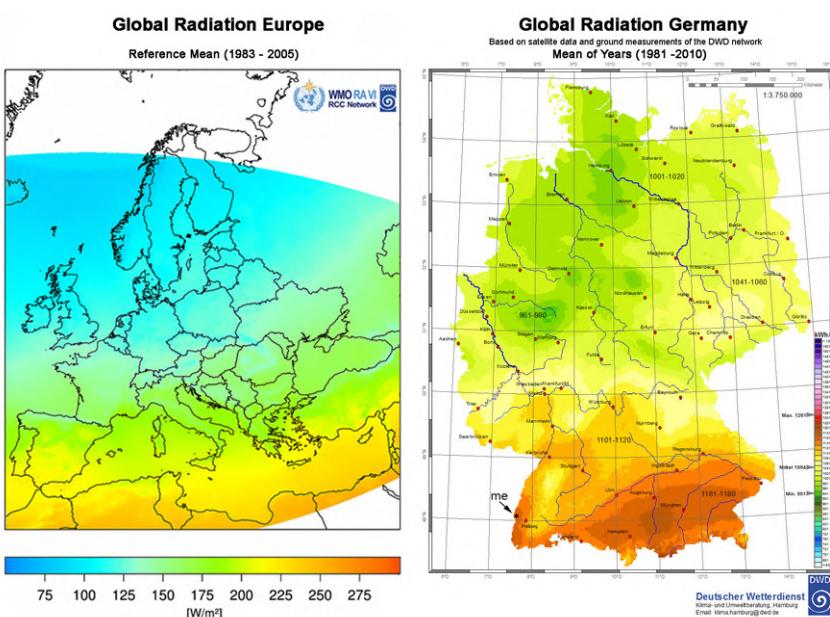


Figure 1. Carte de l'insolation solaire en Europe et en Allemagne (source : DWD [2]).

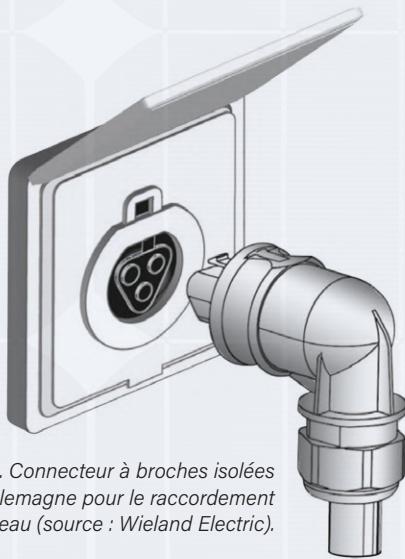


Figure 2. Connecteur à broches isolées approuvé en Allemagne pour le raccordement au réseau (source : Wieland Electric).



Figure 4. Prise secteur et capuchon.



Figure 3. L'onduleur de réseau HM-600 (un exemplaire d'Elektor donne l'échelle).



Figure 5. Onduleur HM-600, vue de dessous.



Figure 6. Un des panneaux Jinko JKM330M-60 avant l'installation.



Figure 7. L'unité consommatrice équipée d'un disjoncteur supplémentaire de 10 A (au centre à gauche) et d'un petit compteur électrique (en bas à gauche). Le compteur moderne à droite comporte un blocage anti-retour.

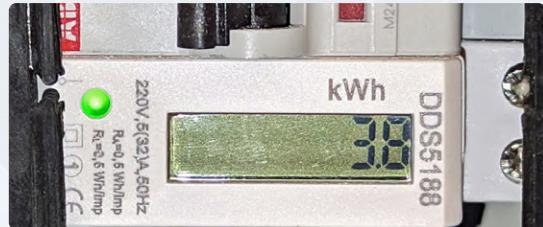


Figure 8. Ce compteur électrique à bas prix s'installe sur le rail DIN et enregistre l'énergie produite par la centrale sur balcon. L'écran affiche la production du premier jour de fonctionnement.



Produits

- Multimètre numérique avec Bluetooth OW16B DVM d'OWON
www.elektor.fr/owon-ow16b-digital-multimeter-with-bluetooth
- Pince de mesure PeakTech 4350
www.elektor.fr/peaktech-4350-clamp-meter
- Station de surveillance de l'environnement pour Raspberry Pi
www.elektor.fr/enviro-environmental-monitoring-station-for-rpi

fonctionne parfois pendant la journée (c'est moins bon en hiver cependant).

Composants du système

Une centrale électrique sur balcon se compose de cellules solaires, d'un onduleur synchrone couplé au réseau et le moyen de s'y raccorder, qui consiste souvent en une prise enfichable spéciale Wieland (**fig. 2**), configuration permise en Allemagne moyennant le respect de la sécurité et de quelques autres restrictions. L'onduleur est le composant clé du système. Il convertit la faible tension continue d'un panneau solaire en une tension alternative de 230 V conforme aux normes et compatible avec le réseau. Il gère la synchronisation de l'alimentation fournie par les panneaux. Si l'on débranche l'onduleur du réseau, sa sortie se coupe en quelques millisecondes. Lors du choix de votre système, vous devez prendre en compte la puissance produite, la puissance utile maximale, la plage de tension de fonctionnement des modules solaires et la qualité de fabrication. Parmi ceux qui m'ont paru de bonne qualité, j'ai choisi l'onduleur HM-600 de Hoymiles. Il coûte 220 € et, d'après les avis des utilisateurs, il a l'air fiable. Il y en a de moins chers, mais moins bien notés, je les ai laissés de côté.

Le HM-600 offre une puissance de sortie maximale de 600 W et peut gérer deux panneaux dans une gamme de puissance de 240 à 380 Wc. Il atteint un rendement d'au moins 96,5% et consomme moins de 50 mW la nuit. Un exemplaire d'Elektor donne l'échelle dans la **figure 3**. Il pèse 3 kg et est assez plat pour pouvoir être monté sous un panneau. Le long câble sert à le relier au réseau (une connexion appropriée est nécessaire), et le câble court avec une fiche épaisse en haut est la connexion aux onduleurs supplémentaires (elle doit être protégée par un capuchon). Vous devez aussi commander les deux pièces de la **figure 4**. La **figure 5** montre une vue de dessous.

L'onduleur et les calculs de rendement jouent un rôle dans le choix des panneaux PV. Pour obtenir 600 W à la sortie de l'onduleur, il vaut mieux que les panneaux en produisent un peu plus pour avoir une certaine réserve. J'ai choisi deux panneaux

monocristallins de 330 Wc de Jinko Solar (**fig. 6**), pour une réserve de 10%. Dans la classe supérieure à 300 Wc, presque tous les panneaux, à quelques légères déviations près, mesurent 166,5 × 100,2 cm et pèsent un peu moins de 20 kg. La garantie des panneaux Jinko indique qu'après 25 ans, leur rendement sera d'au moins 80% de leur rendement initial. Les panneaux sont vendus au prix de 150 €.

Les frais de transport des composants les plus importants du système étant élevés, il est intéressant d'aller en prendre livraison chez un distributeur local. Deux panneaux tiennent dans une voiture de taille moyenne avec les sièges arrière rabattus à plat.

Raccordement au réseau

Dans certains pays, il peut être autorisé de simplement brancher la sortie d'une centrale sur balcon dans une prise de courant murale standard à trois broches. En Allemagne, il est exigé d'utiliser une prise à broches isolées telle que le connecteur Wieland (35 €), même si le risque de choc électrique est minime, car l'onduleur se coupe dès qu'on débranche la fiche. Sur l'internet, cherchez « connecteur pour photovoltaïque ».

La prise peut être câblée en parallèle à une prise de courant du réseau électrique principal située dans un endroit pratique, ou bien être raccordée à l'unité consommatrice avec son propre disjoncteur. Dans le premier cas, il est possible d'avoir à réduire le calibre du disjoncteur principal. En effet dans le cas d'une centrale sur balcon de 600 W, le courant fourni par l'onduleur pourra atteindre 2,6 A, ce qui, dans le pire des cas, surchargerait la prise de courant. Par mesure de sécurité, son disjoncteur de 16 A devrait être remplacé par le disjoncteur de calibre immédiatement inférieur (10 A). En tenant compte de cela et en vérifiant que mon unité consommatrice avait une

marge de capacité, j'ai choisi de connecter la centrale électrique directement à cette unité, avec son propre disjoncteur de 10 A. Pour cela, j'ai posé 15 m de câble et percé un certain nombre de trous pour y accéder. Cela m'a permis d'économiser les 35 € du connecteur et me donne la possibilité d'enregistrer très facilement l'énergie produite. Il existe, bien sûr, des systèmes sophistiqués avec des applications pour afficher la production d'énergie en temps réel sur un ordiphone, mais pour moi, c'est excessif. Tout ce dont j'ai besoin, c'est du total de l'énergie produite, affiché par un simple compteur de kWh à moins de 10 € monté sur rail DIN. L'unité consommatrice avec son extension est représentée sur la **figure 7**. Le petit enregistreur de puissance se trouve en bas à gauche. Le gros plan de la **figure 8** montre le relevé de 3,8 kWh après le premier jour du 9 mai 2021, heureusement un jour sans nuages. Je viens de gagner 1,14 € ! Mon calcul initial d'une économie annuelle nette de 125 € serait-il réaliste ?

Restez du bon côté de la loi

Comme le montre la figure 7, je dispose déjà d'un compteur électrique moderne avec un blocage anti-retour. Si vous avez encore un vieux compteur électromécanique sans ce blocage, il se mettra à tourner à l'envers lorsque votre production dépassera votre consommation. Dans ce cas, vous pourriez prétendre que vous utilisez le réseau comme une pseudo-batterie que vous « déchargez » quand vous avez besoin d'énergie pour la « recharger » avec votre énergie excédentaire. Même si certains politiciens ont pu tenir des raisonnements aussi simplistes, ce n'est ni logique ni autorisé ! C'est illégal et vous devez vous en tenir aux règles. Chaque pays a son propre organisme de réglementation.

En France, pour installer des panneaux photovoltaïques, vous devez vous assurer



Figure 9. Les deux panneaux fixés sur le toit de la véranda.

que votre projet respecte la réglementation locale ou nationale d'urbanisme. L'article R. 421-9 du code de l'urbanisme stipule que les « ouvrages de production d'électricité à partir de l'énergie solaire installés sur le sol dont la puissance crête est inférieure à 3 kWc et dont la hauteur maximum au-dessus du sol ne dépasse pas 1,80 m » sont dispensés de formalités, sauf s'ils sont implantés dans un secteur sauvegardé (site classé par exemple). Avec 600 Wc, il n'y aurait donc pas besoin de permis de construire ni de déclaration préalable de travaux. Dans tous les cas, il est de votre responsabilité de vérifier si un permis est nécessaire. En cas de doute, contactez le service d'urbanisme de votre commune ou consultez le portail d'information sur le photovoltaïque [3].

En Allemagne, vous devez enregistrer votre centrale sur balcon auprès de l'opérateur de réseau et de l'Agence fédérale des réseaux le jour de la mise en service. C'est obligatoire, même pour les très petites installations comme la mienne. Vous pouvez le faire en ligne, cela prend moins de dix minutes. Si votre propre installation est inscrite dans le « registre des données de base du marché », elle sera signalée à votre opérateur, mais cela ne vous dispense pas des formalités d'enregistrement chez lui. Il remplacera au besoin votre ancien compteur par un nouveau à blocage anti-retour ou un compteur bidirectionnel.

En France, si vous produisez de l'électricité, vous pouvez choisir de n'alimenter que vos appareils domestiques (on parle d'autoconsommation) ou d'injecter le surplus sur le réseau électrique (c'est

la revente partielle). Dans le premier cas, les formalités administratives sont réduites, aucun frais de raccordement et aucun travail ne sont à prévoir. Avant de mettre en service votre installation, vous devrez établir une Convention d'Auto-Consommation Sans Injection (CACSI) [4] avec votre fournisseur d'électricité. Vous vous engagez ainsi à respecter les règles de sécurité et de non-perturbation du réseau public de distribution. Si vous voulez revendre l'énergie produite, vous devrez programmer le passage du consuel (comité national pour la sécurité des usagers de l'électricité) et souscrire un contrat d'achat avec votre fournisseur d'électricité.

Cela ne vaut peut-être pas le coup de conclure un contrat de rachat de l'électricité excédentaire. En Allemagne, le tarif de rachat est actuellement de 7,47 centimes par kWh. Si, par exemple, 25% de ma production estimée à 480 kWh par an étaient rachetés, cela me rapporterait $145 \text{ kWh} \times 7,47 \text{ centimes}$, soit près de 11 € par an ou une paire de bières. Cela couvre à peine les frais supplémentaires et les corvées administratives. Il vaut mieux donner l'énergie excédentaire au réseau : on aura la conscience tranquille en sachant que l'on a contribué, d'une manière infime, à réduire la consommation de combustibles fossiles et la production de CO₂.

Construction et installation du système

Sur la **figure 9**, vous pouvez voir que j'ai utilisé de solides supports en acier inoxydable pour fixer les deux panneaux PV sur le toit d'une véranda attenante à

la maison. Des fixations spéciales sont disponibles pour les toits en tuiles. Pour un toit plat, il existe des supports en acier prêts à l'emploi qui permettent d'obtenir l'angle de montage optimal des panneaux en Europe centrale. L'onduleur est monté sous un panneau et le câble résistant aux intempéries suit des chemins de câbles jusqu'au sous-sol où il est prolongé par un câblage NYM jusqu'à l'unité consommatrice. Il va sans dire que toute intervention sur une installation électrique domestique ne doit être effectuée que par un électricien dûment qualifié.

En tout, l'aménagement des implantations, l'installation des chemins de câbles, la pose des câbles, le perçage des trous dans les cornières en acier inoxydable, la modification de l'unité consommatrice et l'installation des panneaux m'ont pris moins de trois jours de travail. Un « technicien d'installation solaire » expérimenté aurait sans doute fait le travail en moitié moins de temps, mais j'ai la satisfaction d'avoir installé le système moi-même et de le connaître assez pour l'entretenir. La **figure 10** montre une vue latérale du système immédiatement après son installation. Ah tiens, je viens de retrouver mon échelle !

Exploitation

De nombreux fabricants allemands proposent un système complet de centrale solaire sur balcon prêt à l'emploi, à brancher sur une prise de courant. Ces systèmes peuvent être mis en service très rapidement, mais vous devriez vérifier si vous ne pouvez pas acheter moins cher les différents composants. Personnellement, j'aime faire des recherches. Ainsi, je peux choisir les composants pour réaliser le système qui répond à mes besoins.

En y repensant, maintenant que c'est terminé, est-ce que je vous inviterais à tenter l'aventure ? Absolument ! J'admettrai avoir été un peu obsédé par le suivi de la production d'énergie. J'ai remarqué que, même par temps nuageux et pluvieux, on peut obtenir quelques centaines de watts-heures. J'attends le plein été avec impatience, quand je pourrai me détendre avec une bière en sachant que les panneaux photovoltaïques qui m'ont abrité du soleil ont contribué à

la rafraîchir. Le fonctionnement en hiver devrait également être intéressant. L'année prochaine, à la même époque, je saurai ce qu'il en fut réellement de ma production d'électricité en 2021.

Vous serez peut-être tenté d'ajouter un ou deux autres systèmes photovoltaïques ; les onduleurs sont même équipés de connecteurs permettant de regrouper plusieurs systèmes. En Allemagne, cela porterait la puissance de la centrale à plus de 600 W et rendrait nécessaire le recours à un ingénieur installateur et accroîtrait inévitablement les formalités administratives. Dans tous les cas, pour toute installation de centrale solaire sur balcon, vous devez informer l'opérateur du réseau qui se rendra sur place et vérifiera que le compteur électrique existant est compatible (au moins équipé d'un blocage anti-retour). De toute façon, un système plus puissant ne vous apportera pas grand-chose, car l'excédent de production de ces installations simples qui ne gèrent pas le rachat d'énergie produite sera simplement donné au réseau. ↴

(210326-04)

Contributeurs

Idée et texte : Thomas Scherer

Rédaction : Jens Nickel

Traduction : Helmut Müller

Mise en page : Harmen Heida



Figure 10. L'installation photovoltaïque se fond dans le décor et n'est pas trop visible.

Des questions, des commentaires ?

Contactez Elektor (redaction@elektor.fr).

LIENS

- [1] Énergie solaire pour robot de tonte, Elektor, 07-08/2021 : www.elektormagazine.fr/200553-04
- [2] Carte d'insolation de l'UE : <https://ec.europa.eu/jrc/en/pvgis>
- [3] Portail gouvernemental d'information sur le photovoltaïque : www.photovoltaque.info/fr/
- [4] Produire son électricité : www.enedis.fr/les-demarches-prealables-pour-produire-son-electricite
- [5] Projets sur le thème du solaire chez Elektor : www.elektormagazine.fr/search?query=solar





La caméra Raspberry Pi HQ en action

prise de vue et diffusion vidéo en continu

avec un Raspberry Pi 4



Mathias Claußen (labo d'Elektor)

Nous avons déjà vu l'utilisation du Raspberry Pi 4 comme caméra réseau. Cette fois, nous montrons comment configurer une caméra pour diffuser en continu de la vidéo et prendre des photos. Bien sûr, ce serait moins cher avec un dispositif prêt à l'emploi, mais où serait le plaisir ? Qu'apprendriez-vous ?

Si vous cherchez une webcam bon marché, à base de RPi et prête à l'emploi, ne lisez pas cet article. Mais si vous voulez créer quelque chose et acquérir de nouvelles connaissances, poursuivez ! Ce projet de caméra est basé sur un RPi 4B, une caméra RPi HQ, un objectif, un écran TFT et un trépied récupéré sur un autre montage. Pour le boîtier, j'ai choisi un modèle d'Adafruit avec un joli style rétro ; du coup le résultat ressemble plus à un appareil photo. Le RPi 4B est le modèle disponible le plus récent. La **figure 1** montre l'installation en action.

De la lumière physique à l'image numérique

Avant de décrire l'assemblage de la caméra, intéressons-nous à la théorie du traitement des images et de la production des données. Pour comprendre ce que la caméra voit et ce que vous obtenez à la fin, quelques étapes de traitement et de conditionnement du signal sont nécessaires. La première étape est la partie optique.

La possibilité d'obtenir une image nette à une distance donnée dépendra des objectifs utilisés. Ils ont également un impact sur les images que vous capturerez ensuite, car les lentilles à l'intérieur de ces objectifs doivent guider la lumière incidente vers votre capteur d'images. Si elles sont mal agencées ou de mauvaise qualité, cela peut provoquer une distorsion de l'image. Ici, nous avons choisi un objectif de 16 mm, marqué 10 MP. Si vous demandez pourquoi un objectif peut avoir une résolution en mégapixels (MP), lisez entre autres [1]. Cela signifie également qu'un objectif de 10 MP ne pourra

jamais faire qu'un capteur d'image produise plus de pixels que ce qui est spécifié. D'ailleurs, nous effectuerons plus loin une comparaison avec l'objectif à 3 MP de 6 mm. Choisissez l'objectif en fonction de votre application.

Outre la partie optique, le capteur d'image lui-même joue un rôle important. Beaucoup de gens pensent que plus sa résolution de capture est élevée, meilleur il est. Si vous avez le choix, vous opterez sans doute pour un capteur à 18 MP plutôt qu'un modèle à 12 MP, car vous obtiendrez 6 MP de plus en résolution. Mais pour les capteurs d'images, une deuxième dimension entre en jeu : leur taille. Plus un capteur d'image est petit pour un nombre de MP donné, moins il y a de lumière disponible par pixel. En outre, plus ces capteurs sont petits, plus il est probable que les pixels voisins puissent s'influencer mutuellement, non seulement à cause de la lumière, mais aussi à cause des effets thermiques au sein de la puce. Cela signifie que la surface utilisée par un capteur peut influencer la qualité de l'image que vous obtenez, tout comme le nombre de mégapixels déclarés.

Noir, blanc et beaucoup de nuances de gris

Pourquoi parler de noir, blanc et nuances de gris alors qu'on souhaite des images en couleur ? Vous devez vous intéresser à ce que le capteur peut voir et ce qu'il va délivrer à notre système. Vous avez peut-être déjà joué avec des photodiodes qui réagissent à la lumière incidente par un changement de résistance. Sur une puce d'appareil photo, le

principe est le même, mais nous avons des millions de ces diodes, densément serrées les unes contre les autres. Chacune de ces diodes n'est capable de distinguer que la quantité de lumière qui la frappe à un endroit donné. La couleur impliquée n'est pas prise en compte, de sorte que la puce ne peut reconnaître que le noir et le blanc, avec de nombreuses nuances de gris entre les deux. Ce qui revient à dire qu'à la base la puce est achromate.

S'il faut détecter les trois principales composantes de couleur d'une image – rouge, bleu et vert – on peut ajouter un filtre optique. Ce filtre est ajouté à chaque diode ou, pour simplifier la formulation, à chaque pixel brut. Cela signifie que nous avons maintenant des millions de pixels bruts qui peuvent indiquer un niveau individuel pour la couleur à laquelle ils sont dédiés. On pourrait supposer qu'ils sont répartis sur la puce avec des motifs rouge, bleu, vert. En fait, la caméra dispose de 50% de pixels bruts verts sur son capteur, le rouge et le bleu représentant chacun 25%, selon la matrice de filtres de Bayer. Pour une explication plus détaillée, consultez l'article de Wikipédia en [2]. Le capteur doit échantillonner chacun de ces pixels bruts et produire une image RVB appropriée à partir de ces pixels. Le capteur d'image lui-même est aujourd'hui plus qu'un simple convertisseur analogique-numérique, il participe au traitement de l'image.

Transfert d'images

Lorsque le capteur d'image a préparé les informations entrantes, il faut les transmettre à un hôte qui peut poursuivre le traitement. Le

moyen de transfert le plus simple possible consiste à utiliser un bus parallèle et à émettre les données à chaque top d'horloge. Bien que simple et utilisée sur les plus petites UC embarquées, cette méthode est limitée. Déplacer de nombreux bits en parallèle à vitesses élevées peut être un défi, notamment pour la synchronisation et l'intégrité des données sur de longues distances, et le protocole peut être propre à chaque module de caméra.

L'Alliance MIPI (*Mobile Industry Processor Interface*) a défini l'interface série pour caméra (CSI, *Camera Serial Interface*) afin de permettre l'interchangeabilité des modules de caméra. La spécification CSI-2 définit une connexion de données série entre la caméra et l'hôte, constituée de plusieurs voies de données. Pour le RPi 4B, seules deux voies de données sont acheminées vers le connecteur de la caméra, ce qui donne un débit de 3 Gbits/s, chaque voie pouvant transférer 1,5 Gbits/s. La documentation MIPI CSI-2 n'étant pas librement accessible, cela explique que le pilote de l'interface de la caméra sur le RPi 4B se cache dans un micrologiciel propriétaire fonctionnant sur le GPU.

Si nous faisons quelques calculs, le transfert d'images de plus de 1920×1080 pixels avec 24 bits de couleur (huit bits par canal) se traduira par 49,77 Mbits par image. Avec une bande passante de 3 Gbits/s, cela donne un transfert de 60 images par seconde. En théorie, cela signifie que pour une résolution de 2560×1440 pixels et 24 bits de couleur, on pourrait transférer un peu plus de 30 images par



Figure 1. Configuration de la caméra Raspberry Pi HQ.

seconde. Malheureusement, cette résolution ne peut pas être utilisée pour la diffusion vidéo. S'il s'agit d'images fixes, la bande passante est moins problématique, même à la résolution maximale de 12 MP qu'offrira le module caméra Raspberry Pi HQ.

Un kit de construction de caméra

Considérons les ingrédients nécessaires à la construction d'une caméra. Au cœur de l'assemblage se trouve la caméra Raspberry Pi HQ qui présente une résolution de 12 MP (**fig. 2** et **3**). Reportez-vous à ses spécifications dans l'encadré « Spécifications de la caméra Raspberry Pi HQ ». Nous utilisons l'objectif à monture C de 16 mm (**fig. 4**) et l'objectif à monture CS de 6 mm (**fig. 5**). Les deux sont disponibles dans la boutique Elektor [3][4].

Un Raspberry Pi 4 traite les images (**fig. 6**). Toutes les pièces sont maintenues dans un boîtier imprimable en 3D d'Adafruit. Note : même s'il est imprimé en 3D, vous avez besoin de quelques vis. Pour plus de détails sur le boîtier, consultez [5]. Les pièces sont présentées dans la **figure 7**.

L'écran TFT de 3,5 pouces (**fig. 8**) et le trépied (**fig. 9**) sont issus d'anciens projets. La **figure 10** montre la caméra assemblée. Voilà un joli kit de construction pour votre propre caméra.

Nous avons un objectif à monture C et un objectif à monture CS. N'oubliez pas d'ajouter ou de retirer l'adaptateur de monture CS à

monture C selon l'objectif que vous utilisez. Si vous avez du mal à faire la mise au point, il se peut que vous ayez installé l'adaptateur alors que vous utilisez un objectif à monture CS. L'adaptateur illustré à la **figure 11** doit être retiré pour les objectifs à monture CS. Son omission ou son utilisation incorrecte peuvent entraîner des images floues.

C'est presque dans la boîte

Une fois tout monté, une version récente de Raspberry Pi OS est installée sur une carte SD. Comme un écran est connecté à des ports HDMI, aucune étape supplémentaire n'est nécessaire pour obtenir une image sur celui-ci, il suffit d'un câble approprié. Si vous utilisez un écran de 3,5 pouces moins cher avec une simple connexion SPI, vous devrez peut-être modifier certains fichiers dans le système d'exploitation du RPi. Si vous voulez acheter un nouvel écran, le fait de disposer d'une entrée HDMI vous dispense de cette modification et vous devriez recevoir directement une sortie d'image.

Une fois la configuration initiale effectuée par Raspberry Pi OS, il faut activer le module caméra. Vous pouvez utiliser *Configuration du Raspberry Pi* à cette fin. Sous l'onglet *Interfaces*, activez *Caméra*. Pour commander le système par le réseau, vous pouvez également cocher *SSH* et *VNC*. Après un redémarrage, la caméra devrait être prête à l'emploi.

Effectuez un premier test avec un terminal. Tapez `raspistill -fp -s` pour obtenir un aperçu de ce que voit la caméra. Si vous utilisez VNC

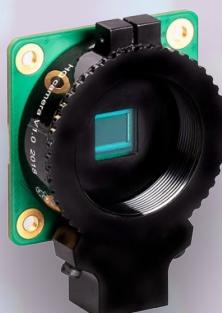


Figure 2. Module caméra Raspberry Pi HQ (source : www.raspberrypi.org).



Figure 3. Option de montage de la caméra Raspberry Pi HQ (source : www.raspberrypi.org).



Figure 4. Objectif de 16 mm pour le module caméra Raspberry Pi HQ.



Figure 5. Objectif de 6 mm pour le module caméra Raspberry Pi HQ.

pour accéder à distance à votre bureau, sachez que vous ne verrez pas la sortie de la caméra dans la visionneuse VNC. Si vous souhaitez que l'aperçu de la caméra soit visible, modifiez les paramètres du serveur VNC comme indiqué à la **figure 12**.

Pour prendre une photo et l'enregistrer sous forme d'image, tapez `raspistill -q 100 -o img.jpg`. Un fichier *img.jpg* sera créé et la qualité sera fixée à 100 pour la compression.

Première séance d'essais

Maintenant que la caméra fonctionne, prenons quelques premières images de test. Nous utiliserons d'abord l'objectif de 6 mm (fig 5). La **figure 13** montre une page de notes A5 avec des lignes droites. La **figure 14** montre un gros plan de cette feuille et vous pouvez voir que l'objectif fonctionne comme une sorte d'optique *fisheye*. Toute la lumière incidente est maintenant transformée en une image unie, mais qui a l'aspect d'une petite sphère, alors que la feuille se trouve à 13 cm de l'objectif. On trouve les spécifications techniques de cet objectif en [4].

Avec l'objectif de 16 mm, nous utilisons la même feuille de notes A5. La **figure 15** montre que l'image n'est visiblement pas déformée comme avec l'objectif de 6 mm. Les spécifications de l'objectif se trouvent en [3].

L'objectif ne participe que pour moitié à la qualité de l'image. Le module caméra Raspberry Pi HQ contribue à la deuxième moitié (fig. 3). Au cœur du module, on trouve un capteur Sony IMX477 [6].

Ceci n'est pas un appareil photo

Avec l'objectif et le RPi 4B, on a tout le matériel de base que l'on trouverait également dans un appareil photo. Ce qui est plus rare de nos jours, c'est l'absence de mise au point automatique pour l'objectif et le module caméra. C'est à vous de régler la mise au point. De même, si vous êtes habitué aux appareils photo, vous remarquerez rapidement l'absence de logiciel. Le RPi n'est pas, du point de vue logiciel, un véritable appareil photo de remplacement. La correction d'image, le filtrage et les autres paramètres que l'on trouve sur les logiciels d'appareil photo ne sont pas présents. C'est donc à vous d'écrire un logiciel de caméra pour travailler avec. Pour l'instant, je ne me concentre pas sur le développement de logiciels pour la caméra, mais je pourrais le faire dans un avenir proche.

Diffusion vidéo en continu

Bien qu'il ne s'agisse pas d'un appareil photo classique, il y ressemble. Mais qu'en est-il de la diffusion de vidéo ? Le RPi peut diffuser des vidéos, mais nous sommes limités à 1080p avec 30 images par seconde (FPS). Néanmoins, avoir une caméra qui peut être connectée via le Wi-Fi semble être une bonne chose. De plus, nous pouvons maîtriser le type de logiciel utilisé et sécuriser le RPi.

Pour la diffusion en continu, il y a plusieurs options logicielles. J'ai choisi *uStreamer* du projet *pi-kvm* [7]. Si vous voulez faire la même chose, vous devez suivre le guide pour compiler le logiciel avec le support d'OpenMAX. OpenMAX vous permet d'utiliser l'accélération matérielle

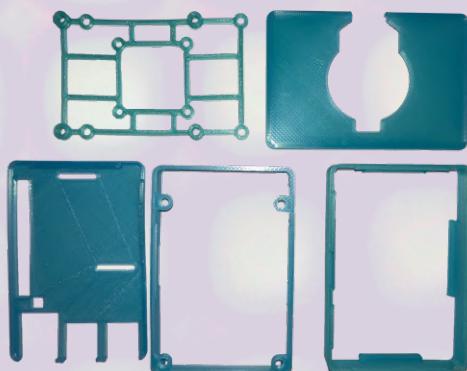


Figure 6. Pièces de base pour la caméra Raspberry Pi.



Figure 7. Pièces du kit de construction de la caméra.



Figure 8. Module TFT en option.



Figure 9. Trépied en option.



Figure 10. Caméra assemblée.

pour les tâches multimédias comme la compression d'images, de sorte qu'un flux MJPEG peut être produit sans que l'UC ait à faire tout le travail.

Clonez le dépôt avec `git clone --depth=1 https://github.com/pikvm/ustreamer` et installez les librairies requises sur le RPi avec `sudo apt install libevent-dev libjpeg8-dev libbsd-dev libraspberrypi-dev` pour mettre tous les éléments en place. Maintenant, sur un terminal, vous devez entrer dans le répertoire `ustreamer` et taper `make OMX=1` pour commencer la compilation. Cela ne prendra qu'un court instant jusqu'à ce qu'un exécutable soit prêt.

Pour lancer un flux, vous pouvez utiliser `./ustreamer --format=YUV --encoder=OMX --workers=3 --host=0.0.0.0 --port=8080 -r 1920x1088 -q 90 -f 30`. Cette commande met en place un flux MJPEG de 1920 × 1080 pixels à 30 FPS. Si vous accédez maintenant à votre RPi avec un navigateur via le port 8080, vous verrez une page web comme à la **figure 16**. Si nous accédons maintenant au flux, la fréquence d'images sera très faible, environ 3 à 4 FPS, mais vous obtiendrez une bonne qualité de vidéo.

Les pilotes et leurs modes

Le RPi fournit, pour tous les modules caméra, un pilote V4L2 qui permet d'écrire des applications pour l'entrée vidéo ou audio en utilisant une interface standard. Cela facilite l'utilisation de la caméra avec le RPi et permet de recourir à un large éventail de logiciels, mais il y a un

problème. Tant que nous demandons au pilote de fournir une résolution de 1270 × 720 pixels ou moins, il exploite le module caméra connecté en mode vidéo. Cela signifie que nous obtenons jusqu'à 60 FPS à 1270 × 720 pixels. C'est bien si vous branchez un petit capteur bon marché pour avoir une sorte de caméra de surveillance. Pour obtenir une résolution plus élevée (par ex. 1920 × 1080 pixels), le pilote passe en mode photo. Dans ce cas, il fait fonctionner le capteur comme un appareil photo et effectue tous les traitements d'image, les corrections de luminosité et le post-traitement comme s'il prenait des photos. Il en résulte un taux de rafraîchissement très faible, mais une image de bonne qualité.

Pour contourner ce problème, nous devons décharger le pilote V4L2 avec `sudo rmmod bcm2835-v4l2` et le recharger avec `sudo modprobe bcm2835-v4l2 max_video_width=4056 max_video_height=3040`. Cela force l'utilisation du mode vidéo pour toutes les résolutions que le RPi prend en charge. Si nous démarrons maintenant µStreamer avec `./ustreamer --format=YUV --encoder=OMX --workers=3 --host=0.0.0.0 --port=8080 -r 1920x1088 -q 90 -f 30`, le nombre d'images par seconde est plus élevé. Si vous ouvrez à nouveau le flux, vous pouvez également accéder aux statistiques et voir que la cadence est maintenant d'environ 20 FPS, selon votre navigateur.

Latence et plus d'images

Un inconvénient de la méthode ci-dessus est le retard introduit dans le flux. L'ensemble du processus, depuis le RPi jusqu'à votre navigateur,



Figure 11. Bague d'adaptation CS.

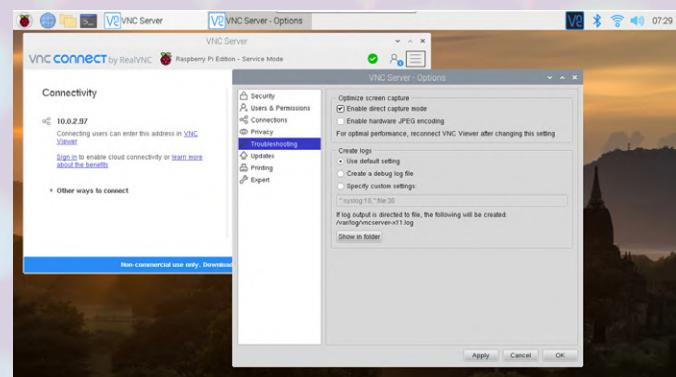


Figure 12. Paramètres VNC pour visualiser la sortie de la caméra.



Figure 13. Feuille de notes A5 avec des lignes droites.



Figure 14. Vue rapprochée avec un objectif de 6 mm.



Figure 15. Feuille A5 capturée avec un objectif de 16 mm.

dure environ une seconde. Si vous faites de la diffusion en continu ou avez l'intention d'en faire, ce délai peut être très désagréable. Le réduire dépend du RPi lui-même, du temps qu'il faut pour capturer et envoyer l'image sur le réseau, et aussi du temps que prennent le décodage et l'affichage de l'image sur le périphérique de lecture. Une solution pour augmenter la cadence est d'utiliser *raspivid* avec *raspivid -t 0 -w 1920 -h 1080 -fps 30 -l -o tcp://0.0.0.0:5000*. Cela permet à un lecteur vidéo comme VLC de se connecter à votre Raspberry Pi et de lire le flux vidéo, toujours avec une latence d'environ une seconde.

Diffusion en continu RTSP avec H.264

Outre la diffusion en continu MJPEG ou l'utilisation de *raspivid* avec TCP, nous pouvons utiliser RTSP. RTSP est un protocole de diffusion en temps réel, conçu pour contrôler la diffusion de données sur un réseau. Il orchestre l'échange de données entre les clients et le serveur de diffusion. Comme RTSP ne gère que l'échange de données, il est également nécessaire de disposer d'un outil capable de produire un flux de données compressé que nous pouvons transporter. C'est là que UV4L (*User space Video4Linux*) entre en jeu. Il permet d'utiliser l'entrée caméra du RPi 4B en combinaison avec l'encodeur H.264 inclus dans le RPi 4B pour produire un flux de données compressé H.264.

Cette configuration permet d'obtenir un flux vidéo (sans audio) avec un retard inférieur à une seconde à 6 Mbit/s et une utilisation de l'UC d'environ 10%. Pour mesurer le délai, la caméra a été dirigée vers un moniteur affichant l'heure locale et un lecteur VLC lisant la vidéo a été

placé au-dessus. Voir les **figures 17** et **18**. Nous pouvons calculer un délai de 670 ms pour le traitement complet du signal. Ce n'est toujours pas parfait, mais c'est beaucoup plus exploitable. La **figure 19** montre les détails du flux produit par cette configuration.

Comment configurer cela ? Tout d'abord, vous devez installer UV4L sur votre RPi 4B en suivant le guide [8]. En plus de l'UV4L de base, vous devez également installer les extensions avec *sudo apt-get install uv4l-raspicam-extras*. C'est la première partie pour avoir un périphérique vidéo qui peut sortir un flux H.264. La configuration sera faite dans une étape ultérieure.

La partie suivante est le v4l2rtspserver [9]. Clonez le dépôt avec *git clone https://github.com/mpromonet/v4l2rtspserver.git* sur votre RPi 4B. Installez CMake avec *sudo apt install cmake* pour pouvoir compiler le v4l2rtspserver. Utilisez un terminal et entrez dans le v4l2rtspserver qui vient d'être créé. Tapez *cmake . && make* pour compiler le code. Si tout fonctionne comme prévu, tapez *sudo make install*.

En se basant sur le post du forum [10], nous allons raccourcir un peu le guide. Dans un terminal, tapez *sudo nano /etc/uv4l/uv4l-raspicam.conf* et cherchez la ligne *encoding*. Changez-la en *encoding = h264*. Recherchez également *width* et *height*. Changez-les en *width = 1296* et *height = 972*. Ensuite, recherchez *framerate*, mettez-le à *framerate = 30*, et enregistrez vos modifications.

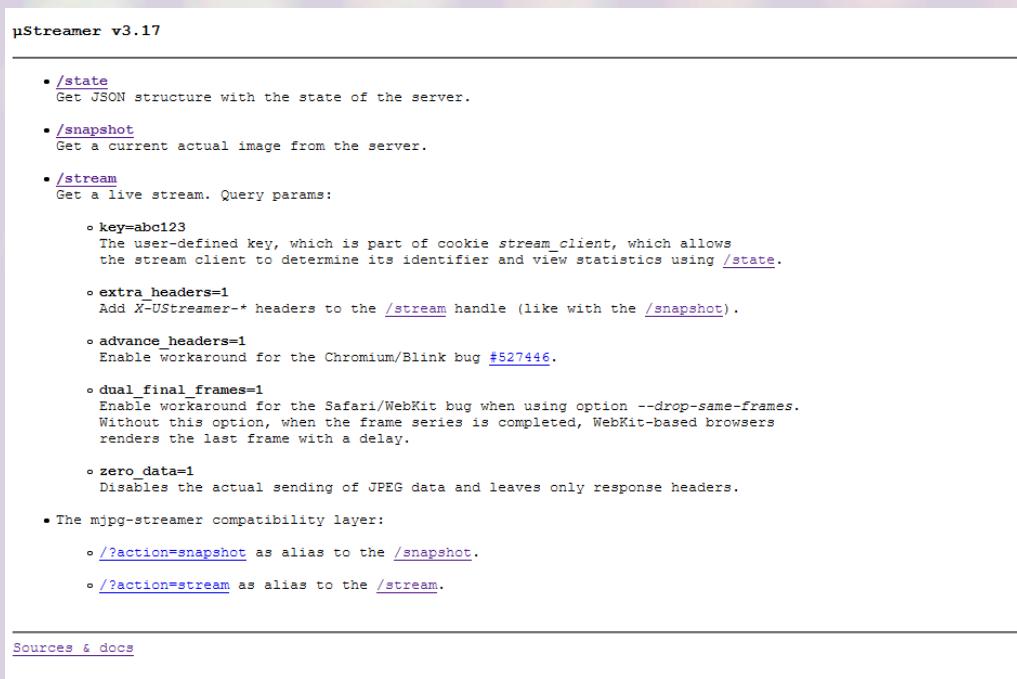


Figure 16. Interface web de μStreamer.

Vous vous demandez peut-être pourquoi nous n'optons pas pour la Full HD (1920 x 1080 pixels). Cela correspondrait à un facteur de forme de 16/9. Comme le capteur nous donne une image 4/3, nous perdrons certaines parties de l'image avec ce choix. S'il nous faut du 16/9, nous pourrons le faire plus tard.

Ensuite, il faut éditer un script pour charger l'uv4l pour la caméra Raspberry Pi HQ au démarrage. Dans un terminal, tapez `sudo nano /etc/systemd/system/uv4l_raspicam.service` et remplacez la ligne `ExecStart` par `ExecStart=/usr/bin/uv4l -f -k --sched-fifo --mem-lock --config-file=/etc/uv4l/uv4l-raspicam.conf --driver raspicam --driver-config-file=/etc/uv4l/uv4l-raspicam.conf --enable-server off`. Enregistrez le fichier et redémarrez le RPi 4B. Pour diffuser maintenant le flux de votre caméra, ouvrez un terminal et tapez `export LD_PRELOAD=/usr/lib/uv4l/uv4lext/armv6l/libuv4l.so`. Sinon, le v4lrtspserver ne se chargera pas et se terminera avec un message d'erreur. Pour démarrer un flux de test, tapez dans un terminal `v4l2rtspserver -F 30 -H 960 -W 1280 -P 8555 /dev/video0` et le v4lrtspserver devrait démarrer et afficher une fenêtre de prévisualisation sur votre bureau.

Pour afficher le flux vidéo, vous pouvez utiliser VLC [11] et ouvrir un flux réseau. Comme URL, utilisez `rtsp://<vous.raspberry.pi.ip>:8555/unicast`. Vérifiez sous *Afficher plus d'options* que la *Mise en cache* est réglée sur un maximum de 100 ms pour une meilleure performance en temps réel. Appuyez sur *Lire* et vous devriez voir le flux de votre caméra.

USB OTG

Puisque le transport de la vidéo sur une connexion réseau peut introduire des retards, pourquoi ne pas utiliser l'USB à la place ? Le RPi 4B+ offre un support USB OTG, ce qui lui permet d'agir comme un périphérique USB lorsqu'il est connecté à un PC. Actuellement, la partie USB nécessaire pour transformer le RPi 4 en webcam présente encore quelques bogues [12], c'est pourquoi nous ne l'utiliserons pas dans cet article. De plus, cette configuration nécessitera un câble en Y pour séparer l'alimentation et les données du port USB, qui doit actuellement être réalisé à la main. Donc pas de sortie vidéo stable à faible latence, du moins avec le RPi 4. Il n'en serait pas de même pour le RPi Zero W, comme le démontre la *webcam Show-me* [13].

Action !

La caméra assemblée (fig. 1) est posée sur mon bureau. Elle capture des images, les enregistre et diffuse des vidéos. Le capteur d'image, associé aux objectifs à monture C et CS, produit une excellente image en mode photo et lors de la capture de vidéos. L'association de ce matériel avec le logiciel et le pilote me laisse des sentiments mitigés. Le RPi 4B est limité en matière de codage vidéo et de mise en œuvre de l'interface de la caméra.

En écrivant cet article, j'ai eu quelques idées pour améliorer ou utiliser des logiciels existants. La première serait d'implémenter une interface utilisateur fonctionnelle pour de la photo de base avec un aperçu en direct et un enregistrement des images. Dans le meilleur des cas, celle-ci fonctionnerait également sur les réseaux avec une

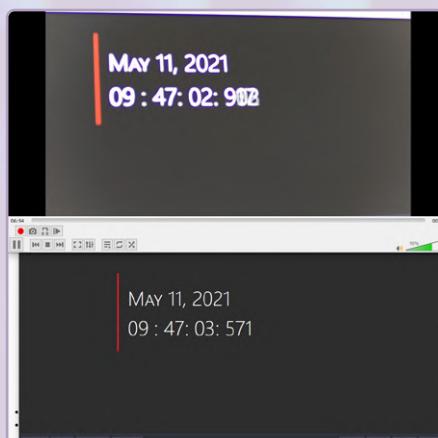


Figure 17. Première mesure du retard du flux.

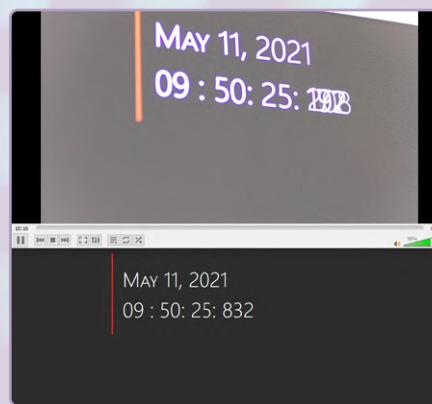


Figure 18. Deuxième mesure du retard du flux.

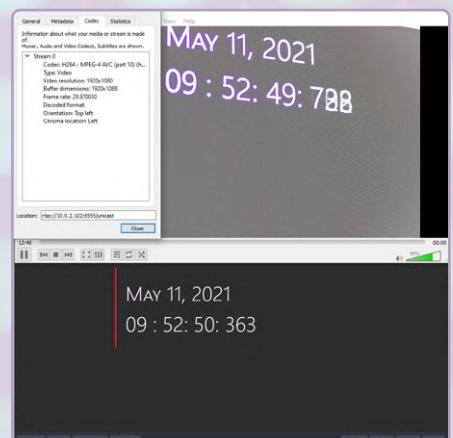


Figure 19. Paramètres du flux.

latence inférieure à une seconde. En outre, il faudrait examiner plus en profondeur l'aspect USB OTG, car le matériel du RPi 4B en est capable. Une fois encore, si cela avait fonctionné dès le déballage, où serait le plaisir ? Les meilleures leçons à tirer proviennent des défis qu'offre un projet. 

(200582-04)

Contributeurs

Texte et photos : **Mathias Claußen**
Rédaction : **Jens Nickel, C. J. Abate**
Mise en page : **Harman Heida**
Traduction : **Denis Lafourcade**

Des questions, des commentaires ?

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



PRODUITS

- **Objectif de 16 mm à monture C (10 MP) pour module caméra Raspberry Pi HQ**
www.elektor.fr/16-mm-c-mount-lens-10-mp-for-raspberry-pi-hq-camera-module
- **Objectif de 6 mm à monture CS (3 MP) pour module caméra Raspberry Pi HQ**
www.elektor.fr/6-mm-cs-mount-lens-3-mp-for-raspberry-pi-hq-camera-module
- **Module caméra Raspberry Pi HQ**
www.elektor.fr/raspberry-pi-high-quality-camera-module
- **Raspberry Pi 4B (2 Go de RAM)**
www.elektor.fr/raspberry-pi-4-b-2-gb-ram
- **Raspberry Pi 4B (4 Go de RAM)**
www.elektor.fr/raspberry-pi-4-b-4-gb-ram
- **Imprimante 3D i3 Mega-S d'Anycubic (kit)**
www.elektor.fr/anycubic-i3-mega-s-3d-printer-kit

LIENS

- [1] Mégapixels et capteur photographique : https://fr.wikipedia.org/wiki/Capteur_photographique
- [2] Matrice de filtres de Bayer : https://fr.wikipedia.org/wiki/Matrice_de_filtres_color%C3%A9s
- [3] Objectif de 16 mm, boutique Elektor :
www.elektor.fr/16-mm-c-mount-lens-10-mp-for-raspberry-pi-hq-camera-module
- [4] Objectif de 6 mm, boutique Elektor : www.elektor.fr/6-mm-cs-mount-lens-3-mp-for-raspberry-pi-hq-camera-module
- [5] Boîtier en impression 3D : <https://learn.adafruit.com/raspberry-pi-hq-camera-case>
- [6] Informations sur le capteur IMX477 : www.sony-semicon.co.jp/products/common/pdf/IMX477-AACK_Flyer.pdf
- [7] µStreamer sur GitHub : <https://github.com/pikvm/ustreamer>
- [8] Installation d'UV4L : <https://www.linux-projects.org/uv4l/installation/>
- [9] v4l2rtspserver sur GitHub : <https://github.com/mpromonet/v4l2rtspserver>
- [10] Message du forum sur la diffusion en continu H.264 à faible latence :
www.bensoftware.com/forum/discussion/3254/raspberry-pi-h264-rtsp-low-latency-camera-instructions/p1
- [11] Lecteur VLC : <https://www.videolan.org/>
- [12] Limites du « gadget » UVC pour Raspberry Pi 4B : <https://www.raspberrypi.org/forums/viewtopic.php?t=294140>
- [13] Webcam Show-me sur GitHub : <https://github.com/showmewebcam/showmewebcam>

utilisation d'écrans dans les projets Raspberry Pi

Extrait : écrans à diodes électroluminescentes organiques (OLED)

Dogan Ibrahim (Royaume-Uni)

Cet article est un extrait du livre (en anglais) de Dogan Ibrahim intitulé *Using Displays in Raspberry Pi Projects* et publié par Elektor. Depuis leur apparition, les écrans OLED ont toujours eu les faveurs des makers et des ingénieurs spécialistes des systèmes embarqués. Dans cet article, nous verrons comment incorporer des écrans OLED dans des projets avec un minimum d'efforts et de programmation – des exigences qui, disons-le, font clairement du Raspberry Pi le candidat idéal. C'est ce que nous vous proposons de découvrir !

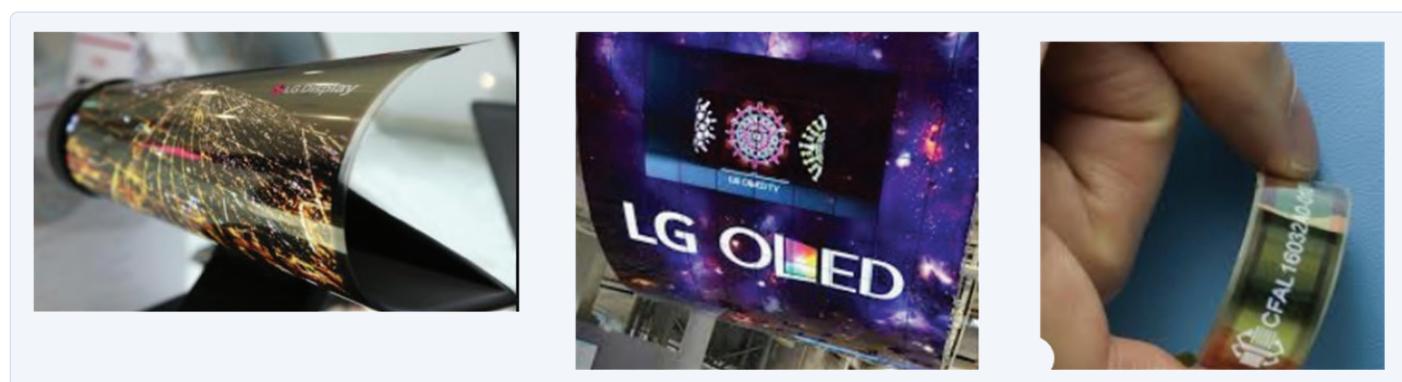


Figure 1. Quelques écrans OLED.

Note de l'éditeur : cet article est un extrait du livre « *Using Displays in Raspberry Pi Projects* » 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 ? ».

Une diode électroluminescente organique est un composant d'affichage à base de LED, dans lequel la couche électroluminescente émissive est un film de composé organique qui émet de la lumière en présence d'un courant électrique. La couche organique est située entre deux électrodes dont au moins une est transparente. Les écrans OLED sont utilisés dans de nombreuses applications commerciales, telles que les écrans de télévision, les écrans d'ordinateur, les ordiphones, les consoles de jeux et autres écrans numériques. Les OLED peuvent être pilotés par des contrôleurs

à matrice passive (PMOLED) ou à matrice active (AMOLED). Généralement, dans les écrans de type PMOLED, toutes les lignes et colonnes sont commandées séquentiellement, une par une. Dans les écrans de type AMOLED, on utilise un fond de panier de transistors à couches minces pour accéder directement à chaque pixel et l'activer ou le désactiver.

En général, les écrans OLED présentent les avantages suivants par rapport aux écrans LCD :

- La consommation d'énergie des OLED est très faible, ce qui les rend très intéressants pour les applications d'affichage mobile.
- Les écrans OLED ont des qualités d'image améliorées, un meilleur contraste, une plus grande luminosité, des angles de vision plus larges et des taux de rafraîchissement plus rapides.
- Les écrans OLED sont plus résistants, ce qui les rend aptes à une utilisation dans une plus large gamme de températures.
- Les écrans OLED peuvent être fabriqués pour être flexibles. On peut par exemple les plier, les porter au poignet, etc.

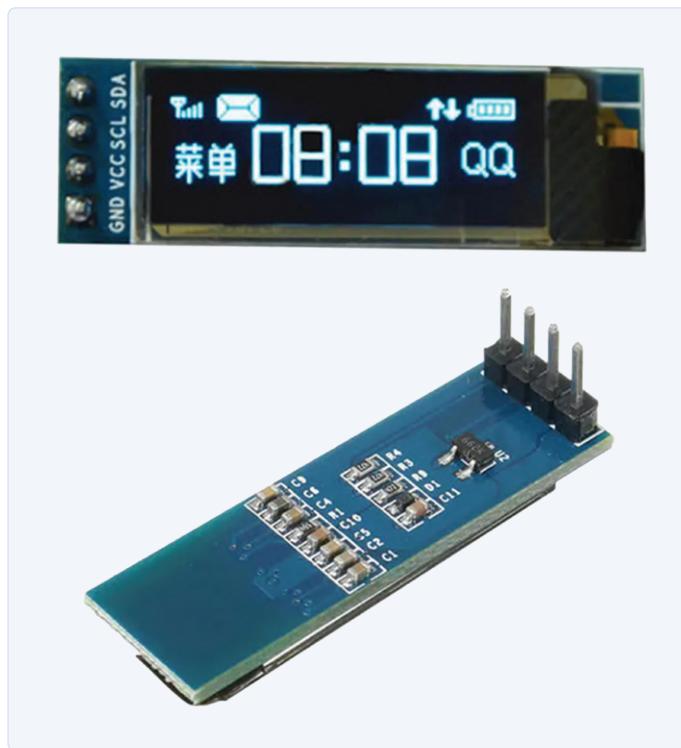


Figure 2. OLED de 128x32 pixels. Source : <https://uk.banggood.com>

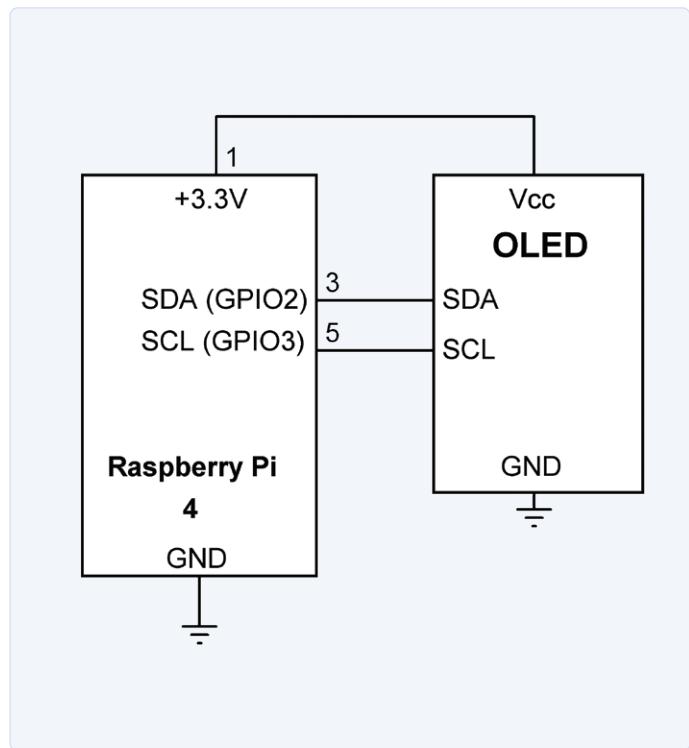


Figure 3. Le schéma du circuit.

Les écrans OLED ne sont cependant pas parfaits et présentent certains inconvénients :

- Leur coût est considérablement plus élevé.
- Leur durée de vie est limitée, bien que cette situation se soit récemment améliorée.
- Ils peuvent poser problème à la lumière directe du soleil en raison de leur nature émissive.

La **figure 1** montre quelques exemples d'écrans OLED.

Utilisation des écrans OLED

Il existe de nombreuses tailles d'écrans OLED avec différentes matrices de pixels. Il n'est évidemment pas possible de couvrir tous les types d'écrans OLED ici. Dans cet article, nous utiliserons le très populaire écran OLED de 128x32 pixels, piloté par la puce intégrée SSD1306 (voir **figure 2**). Il est peut-être opportun de passer en revue les caractéristiques de cet écran avant de l'incorporer à nos projets. Les spécifications de cet écran OLED sont les suivantes :

- Résolution : 128x32 pixels
- Couleur du pixel : blanc (le bleu est également disponible)
- Taille : 2,3 cm
- Interface : I²C
- Pilote : SSD1306
- 4 broches : GND, VCC, SCL (I²C), SDA (I²C)
- Angle de vue : plus de 160°
- Tension de fonctionnement : +3,3 V / +5 V

L'écran OLED piloté par la puce SSD1306 communique avec l'ordina-

teur hôte via l'interface I²C (c'est-à-dire que deux broches suffisent pour l'interface, en plus des broches d'alimentation et de masse). L'adresse I²C du pilote est 0x3C par défaut. La **figure 3** montre la connexion entre le Raspberry Pi et l'écran OLED. La puce fonctionne normalement en +3,3 V, mais un régulateur de tension sur la puce permet également un fonctionnement en +5 V.

Port Raspberry Pi	N° broche Raspberry Pi	Broche OLED
SDA	(GPIO2)	2
SCL	(GPIO3)	3
GND	39	GND
+3,3V	1	V _{cc}

Installation de la bibliothèque OLED

Il faut activer l'interface I²C sur le Raspberry Pi avec l'outil de configuration `sudo raspi-config`. Vous devez ensuite installer la bibliothèque OLED. Ici, nous utilisons la bibliothèque Adafruit_Python_SSD1306. Les étapes sont les suivantes (certaines de ces bibliothèques peuvent être déjà installées sur votre Raspberry Pi) :

```
pi@raspberrypi:~ $ sudo apt install -y python3-dev
python3-pil python3-pip python3-setuptools python3-rpi.gpio
pi@raspberrypi:~ $ sudo pip3 install
adafruit-circuitpython-ssd1306
```

Avant d'utiliser un dispositif connecté à l'interface I²C, il faut d'abord détecter son adresse. Pour trouver l'adresse I²C de l'écran,

```
pi@raspberrypi:~ $ i2cdetect -y 1
  0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: -----
10: -----
20: -----
30: -----          3c -----
40: -----
50: -----
60: -----
70: -----          31 -----
pi@raspberrypi:~ $
```

Figure 4. Détection de l'écran OLED.

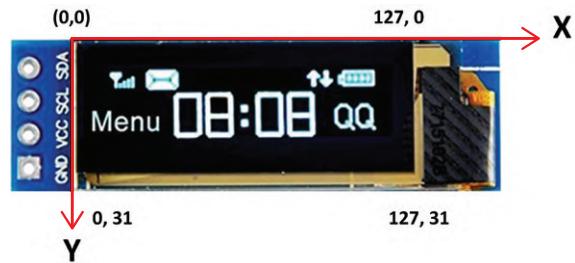


Figure 5. Coordonnées de l'écran.



Listage 1. OLED_corners.py.

```
#####
# MONTRE LES PIXELS AUX COINS DE L'ECRAN
# =====
#
# Ce programme affiche les pixels aux quatre coins de l'écran
#
# Author: Dogan Ibrahim
# File : OLED_corners.py
# Date : November 2020
#
from board import SCL, SDA
import busio
import adafruit_ssd1306

i2c = busio.I2C(SCL, SDA)
display = adafruit_ssd1306.SSD1306_I2C(128, 32, i2c)

display.fill(0)          # Efface les pixels
display.show()            # Affiche les données

display.pixel(0,0,1)      # Pixel en (0,0)
display.pixel(127,0,1)    # Pixel en (127,0)
display.pixel(0,31,1)      # Pixel en (0,31)
display.pixel(127, 31, 1)  # Pixel en (127,31)
display.show()            # Affiche les données
```



Listage 2. OLEDText.py.

```
#####
# AFFICHAGE DE TEXTE
# =====
#
# Ce programme affiche le texte E L E K T O R en (15,5)
#
# Author: Dogan Ibrahim
# File : OLEDText.py
# Date : November 2020
#
from PIL import Image, ImageDraw, ImageFont
from board import SCL, SDA
import busio
import adafruit_ssd1306

i2c = busio.I2C(SCL, SDA)
display = adafruit_ssd1306.SSD1306_I2C(128, 32, i2c)
font = ImageFont.load_default()          # police par défaut

display.fill(0)                        # efface l'écran
display.show()

width = display.width                 # Largeur
height = display.height                # Hauteur
image = Image.new('1', (width, height))
draw = ImageDraw.Draw(image)
draw.text((15, 5), "E L E K T O R", font = font, fill = 255)
display.image(image)
display.show()
```

il faut exécuter la commande suivante dans le terminal :

```
pi@raspberrypi:~ $ i2cdetect -y 1
```

Le résultat devrait ressembler à la **figure 4**.

Projet 1 : afficher des pixels aux quatre coins de l'écran

Les coordonnées de l'écran OLED sont indiquées dans la **figure 5**. Le coin supérieur gauche est en (0,0). La coordonnée X traverse l'écran de 0 à 127, tandis que la coordonnée Y le descend de 0 à 31.

Listage du programme : le **listage 1** présente le programme `OLED_corners.py`. Ce programme est inclus dans le fichier d'archive des logiciels du livre qu'on trouve à l'adresse [1], sous la rubrique Téléchargements. Au début du programme, on importe dans le programme la bibliothèque `OLED` d'Adafruit. Les pixels sont ensuite effacés en appelant la fonction `display.fill(0)`. Il est recommandé de toujours effacer les pixels au début d'un programme. Les pixels aux quatre coins de l'écran sont ensuite allumés. Notez qu'il faut appeler la fonction `display.show()` pour afficher les données sur l'écran.

La **figure 6** montre l'écran avec les quatre pixels d'angle activés.

Projet 2 : afficher du texte

Dans ce projet, on affiche le texte « ELEKTOR » à partir de la position (15,5) de l'écran.

Listage du programme : le **listage 2** montre le programme `OLEDText.py`. Il utilise les fonctions de la bibliothèque `PIL` [2]. Au début du programme, on importe les modules `image` et `font` de la bibliothèque `PIL`. La bibliothèque de polices par défaut est ensuite chargée. Enfin, le programme efface les pixels. Les variables `width` et `height` contiennent la largeur (128) et la hauteur (32) de l'écran.

Pour créer une image, il faut d'abord créer une image vide aux dimensions de l'écran, comme ceci :

```
image = Image.new('1', (width, height))
```

Avec cet objet 'image', on crée l'objet 'draw' (dessin) qui sert à dessiner des formes et du texte :

```
draw = ImageDraw.Draw(image)
```

Dans ce programme, nous voulons afficher un texte aux coordonnées (15,5), et on utilise donc la fonction `draw.text` :

```
draw.text((15, 5), "E L E K T O R", font = font, fill = 255)
```

Remarquez que `fill = 255` dessine l'image en blanc, et `0` la dessine en noir. Ensuite, nous devons afficher l'image en appelant les fonctions `display.image(image)` et `display.show()`. La [figure 7](#) montre le texte affiché.

Projet 3 : afficher de formes

On peut dessiner diverses formes sur l'écran. Pour une liste complète des fonctions, consultez la documentation de la bibliothèque PIL – il y a beaucoup plus de fonctions que celles décrites brièvement ici.

Rectangle :

`(x,y)` est le coin supérieur gauche du rectangle. `outline` est la couleur du contour de la forme (255 pour le blanc, 0 pour le noir), `fill` est l'intérieur de la forme (255 pour le blanc, 0 pour le noir).

```
draw.rectangle((x, y, width, height), outline = nn, fill = mm)
```

Ellipse :

```
draw.ellipse((x, y, width, height), outline = nn, fill = mm)
```

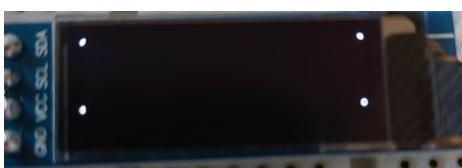


Figure 6. Nous avons des pixels aux quatre coins de l'écran OLED !



Figure 7. Le texte affiché : « E L E K T O R » – quoi d'autre ?

Ligne :

`(x1,y1)` est la première coordonnée, `(x2,y2)` est la seconde.

```
draw.line((x1, y1, x2, y2), fill = mm)
```

Arc :

```
draw.arc((x1,x2,y1,y2), s1, s2, fill = mm)
```

Trace un arc entre les angles de départ (`s1`) et d'arrivée (`s2`), à l'intérieur de la boîte (`x1,x2,y1,y2`).

Corde :

```
draw.chord((x1,x2,y1,y2), s1, s2, fill = mm)
```

Trace une ligne entre les angles de départ (`s1`) et d'arrivée (`s2`), à l'intérieur de la boîte (`x1,x2,y1,y2`).

Camembert :

```
draw.pieslice((x1,y1,x2,y2), s1, s2, fill = mm)
```

Identique à `arc`, mais dessine également des lignes droites entre les points d'extrémité et le centre de la boîte englobante.

Point :

```
draw.point((x1,x2), fill = mm)
```

Dessine des pixels aux coordonnées données. Il est possible de dessiner plus d'un pixel, comme dans l'exemple ci-dessous où trois pixels individuels sont dessinés aux coordonnées données :

```
draw.point([(x1,y1), (x2,y2), (x3,y3)], fill = mm)
```



Listage 3. LEDRect.py.

```
#
# AFFICHE UN RECTANGLE AVEC DU TEXTE A L'INTÉRIEUR
# =====
#
# Dans ce programme, on affiche un rectangle dans un coin de
# l'écran et le texte R E C T A N G L E est inscrit à l'intérieur
#
# Author: Dogan Ibrahim
# File : OLEDRect.py
# Date : November.py
#-----
from PIL import Image,ImageDraw,ImageFont
from board import SCL, SDA
import busio
import adafruit_ssd1306

i2c = busio.I2C(SCL, SDA)
display = adafruit_ssd1306.SSD1306_I2C(128, 32, i2c)
font = ImageFont.load_default()

display.fill(0)
display.show()

width=(display.width)
height=(display.height)
image=Image.new('1',(width,height))
draw=ImageDraw.Draw(image)
draw.rectangle((0,0, 127, 31),outline = 255, fill = 0)
draw.text((15, 12),"R E C T A N G L E", font = font, fill = 255)
display.image(image)
display.show()
```



Listage 4. OLEDShape1.py.

```
#-----
#      AFFICHE QUATRE RECTANGLES AVEC LES CHIFFRES 1,2,3,4
#      =====
#
# Dans ce programme, on affiche 4 rectangles. On inscrit le chiffre 1
# à l'intérieur du rectangle 1, le chiffre 2 à l'intérieur du
# rectangle 2, etc.
#
# Author: Dogan Ibrahim
# File : OLEDShape1.py
# Date : November 2020
#-----
from PIL import Image,ImageDraw,ImageFont
from board import SCL, SDA
import busio
import adafruit_ssd1306

i2c = busio.I2C(SCL, SDA)
display = adafruit_ssd1306.SSD1306_I2C(128, 32, i2c)
font = ImageFont.load_default()

display.fill(0)
display.show()

width=(display.width)
height=(display.height)
image=Image.new('1',(width,height))
draw=ImageDraw.Draw(image)
draw.rectangle((0,0, 127, 31),outline = 255, fill = 0)
draw.line((64, 0, 64, 31), fill = 255)
draw.line((0, 16, 127, 16), fill = 255)
draw.text((32, 4), "1", font = font, fill = 255)
draw.text((94, 4), "2", font = font, fill = 255)
draw.text((32, 17), "3", font = font, fill = 255)
draw.text((94, 17), "4", font = font, fill = 255)
display.image(image)
display.show()
```



Listage 5. LEDShape2.py.

```
#-----
#      AFFICHE UNE CORDE ET UN POLYgone
#      =====
#
# Dans ce programme, on dessine une corde et un polygone.
# Le polygone est rempli en blanc.
#
# Author: Dogan Ibrahim
# File : OLEDShape2.py
# Date : November 2020
#-----
from PIL import Image,ImageDraw,ImageFont
from board import SCL, SDA
import busio
import adafruit_ssd1306

i2c = busio.I2C(SCL, SDA)
display = adafruit_ssd1306.SSD1306_I2C(128, 32, i2c)
font = ImageFont.load_default()

display.fill(0)
display.show()

width=(display.width)
height=(display.height)
image=Image.new('1',(width,height))
draw=ImageDraw.Draw(image)
draw.chord((0, 0, 90, 30), 10, 180, fill = 255)
draw.polygon([(100,5), (120,9), (120,25), (100,30)],outline=255,fill=255)
display.image(image)
display.show()
```



Figure 8. L'affichage obtenu en exécutant OLEDRect.py.



Figure 9. L'affichage obtenu en exécutant OLEDShape1.py (rectangles).



Figure 10. L'affichage obtenu en exécutant OLEDShape2.py (corde et polygone).

Des questions, des commentaires ?

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

Contributeurs

Texte : **Dogan Ibrahim**
 Rédaction : **Jan Buiting**
 Mise en page : **Harmen Heida**
 Traduction : **Denis Lafourcade**

Polygone :

```
draw.polygon([(x1,y1), (x2,y2), (x3,y3), .....],  
            outline = nn, fill = mm)
```

Dessine un polygone dont le contour est constitué de lignes droites entre les coordonnées données, plus une ligne droite entre la dernière et la première coordonnée. La liste des coordonnées peut être un objet séquentiel contenant soit des couples [(x, y), ...], soit des valeurs numériques [x, y, ...]. Elle doit contenir au moins trois coordonnées.

Le programme *OLEDRect.py* (**listage 3**) montre le dessin d'un rectangle aux coins de l'écran et l'écriture du texte « RECTANGLE » à l'intérieur de ce rectangle. La **figure 8** montre le résultat.

Le programme *OLEDShape1.py* (**listage 4**) affiche quatre rectangles. Dans le rectangle 1, on inscrit le chiffre « 1 ». Dans le rectangle 2, on inscrit le chiffre « 2 », et ainsi de suite. La **figure 9** montre le résultat.

Le programme *OLEDShape2.py* (**listage 5**) affiche une corde et un quadrilatère. Le paramètre de remplissage du polygone est fixé à 255, il est donc rempli d'un fond blanc. La **figure 10** montre le résultat.



Listage 6. OLEDBitmap.py.

```
-----  
#  
#      AFFICHE UN BITMAP  
#      =====  
#  
# Ce programme affiche une image bitmap  
#  
# Author: Dogan Ibrahim  
# File : OLEDBitmap.py  
# Date : November 2020  
#-----  
from PIL import Image,ImageDraw,ImageFont  
from board import SCL, SDA  
import busio  
import adafruit_ssd1306  
  
i2c = busio.I2C(SCL, SDA)  
display = adafruit_ssd1306.SSD1306_I2C(128, 32,  
i2c)  
  
display.fill(0)  
display.show()  
  
width=(display.width)  
height=(display.height)  
image=Image.new('1',(width,height))  
image=Image.open('LetterA.png').  
resize((width,height),\br/>Image.ANTIALIAS).convert('1')  
display.image(image)  
display.show()
```

Publié



Capteurs d'humidité de la série SHT

**Mesure précise de l'humidité
basée sur la technologie CMOSens®.**

Série SHT2x solutions éprouvées pour votre application

Plage d'humidité mesurée	0-100% RH
Interface de communication	I2C, PWM, SDM

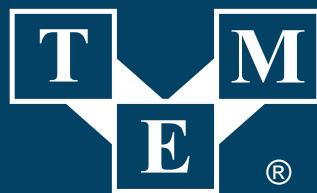
Série SHT3x mesure d'humidité fiable et précise

Plage d'humidité mesurée	0-100% RH
Interface de communication	En fonction du modèle : sortie analogique ou I2C

Série SHT40 - une nouvelle génération de mesure d'humidité encore plus précise

Plage d'humidité mesurée	0-100% RH
Interface de communication	I2C

Allez sur [na tme.eu](http://tme.eu) et découvrez-le !



Electronic Components

TRANSFER MULTISORT ELEKTRONIK

GLOBAL DISTRIBUTEUR DE COMPOSANTS ÉLECTRONIQUES

Ustronna 41, 93-350 Łódź, Pologne
+48 42 645 54 44, export@tme.eu, tme.eu



facebook.com/TME.eu
youtube.com/TMElectroniComponent
instagram.com/tme.eu

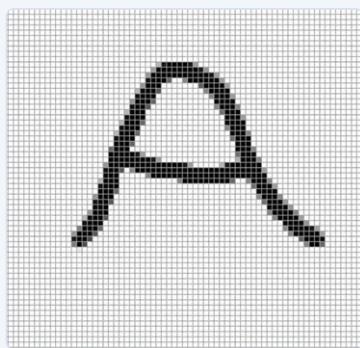
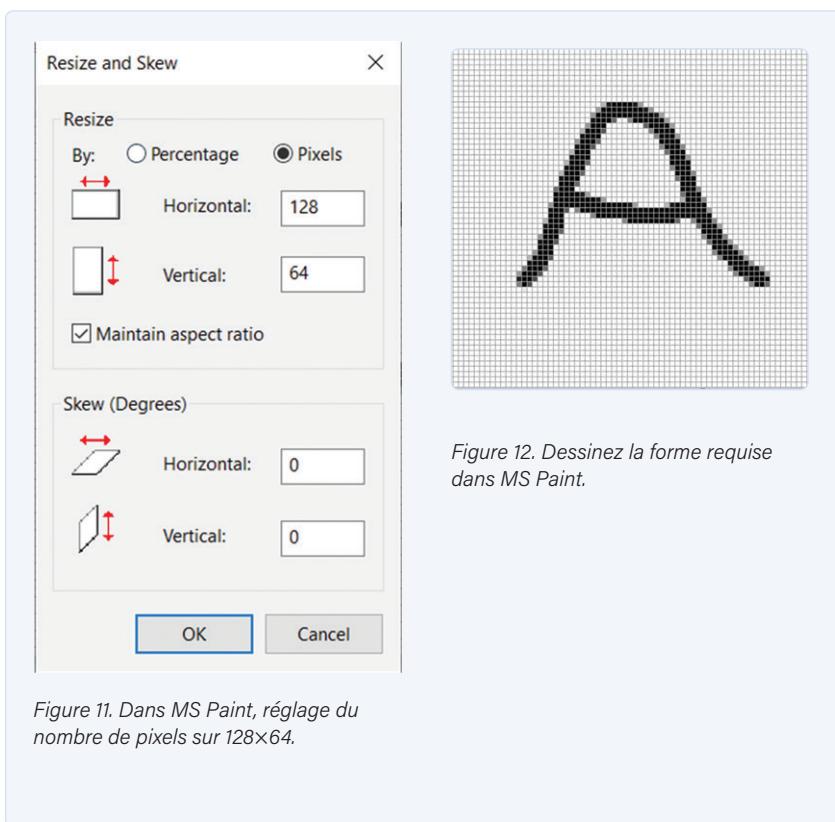


Figure 13. Gagné ! Une lettre quasi manuscrite sur l'écran OLED.

est redimensionné et converti en couleur sur 1 bit comme requis par la bibliothèque. La **figure 13** montre le résultat.

Tous les programmes abordés dans cet article sont disponibles sur la page web Elektor du livre [1]. Sur cette page, allez jusqu'à Téléchargements pour trouver le fichier qui regroupe tous les programmes. Téléchargez le fichier .zip, extrayez-le sur votre ordinateur, puis localisez et ouvrez les six programmes Python dont il est question ici. 

(210197-04)

Projet 4 : création et affichage d'un bitmap

Ici, nous créons une image bitmap de 128×32, puis l'affichons à l'écran. L'image sera la lettre « A ». Les étapes sont détaillées ci-dessous.

Création de l'image : nous utilisons le programme Paint de Microsoft pour créer notre image.

Lancez le programme MS Paint.

Cliquez sur Accueil Redimensionner, puis sur Pixels. Réglez Horizontal sur 128 pixels et Vertical sur 64 pixels comme indiqué sur la **figure 11**. Cliquez sur OK.

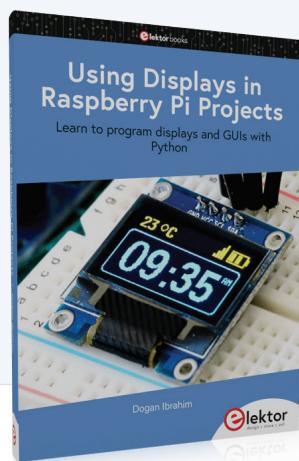
Cliquez sur Affichage, puis sur Zoom avant pour agrandir la grille. Dessinez la forme que vous souhaitez à l'aide de la souris. Dans ce projet, la lettre A est dessinée à la main, comme le montre la **figure 12**.

Enregistrez le fichier, par exemple sous le nom de *LetterA.png*. Copiez le fichier dans le répertoire personnel (/home/pi) de Raspberry Pi. Vous pouvez utiliser le programme WinSCP, disponible gratuitement, pour copier le fichier.

Le programme final, *OLEDBitmap.py*, figure dans le **listage 6**. On utilise la fonction `Image.open` pour ouvrir le fichier image, le fichier

PRODUITS

- **Livre en anglais, D. Ibrahim, *Using Displays in Raspberry Pi Projects* (Elektor 2021)**
- **Version papier :** www.elektor.fr/using-displays-in-raspberry-pi-projects
- **Version numérique :** www.elektor.fr/using-displays-in-raspberry-pi-projects-e-book



LIENS

- [1] Téléchargement des programmes : <http://www.elektor.fr/using-displays-in-raspberry-pi-projects>
- [2] Bibliothèque LIP : <http://effbot.org/imagingbook/imagedraw.htm>

Propeller 2 de Parallax (4)

Envoi de chaînes de caractères

Mathias Claussen (Elektor)

Notre série d'articles sur le Propeller 2 de Parallax se poursuit ! Cette fois, nous nous penchons sur certains problèmes que vous pourriez rencontrer lors de l'envoi de chaînes de caractères avec le langage Spin2...

L'objectif de cette série d'articles est de présenter le microcontrôleur Propeller 2 de Parallax. Dans cet article, nous élargirons les possibilités d'envoyer des chaînes de caractères, comme vous le faites avec la fonction `print` sur vos systèmes Arduino. L'idée est de pouvoir émettre de manière très classique des chaînes de caractères, ce qui peut être utile

Listage 1. Code complet.

'We run in the internal RC giving us 25 MHz clockspeed

```

PUB main()
  pinwrite(56, 1 )
  serial_start()
repeat
  waitms( 250 )
  pinwrite(56, 1 )
  tx("0")
  waitms( 250 )
  pinwrite(56, 0 )
  tx("1")

PUB serial_start()
  WRPIN( 57, %01_11110_0 )           'set async tx mode for txpin
  WXPIN( 57, ((217<<16) + (8-1) ) )  'set baud rate to sysclock/115200 and word size to 8 bits
org
  dirh    #57
end

PUB tx(val)
  WYPIN(57,val) 'load output word
org
  WAITX  #1           'wait 2+1 clocks before polling busy
  wait
  RDPIN  val,#57 WC  'get busy flag into C
  IF_C   JMP  #wait   'loop until C = 0
end

```

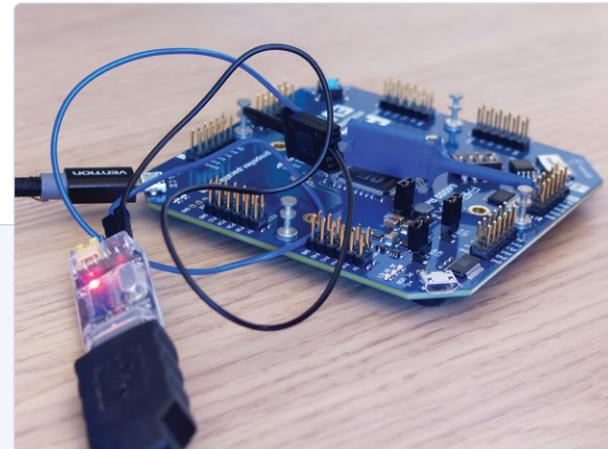
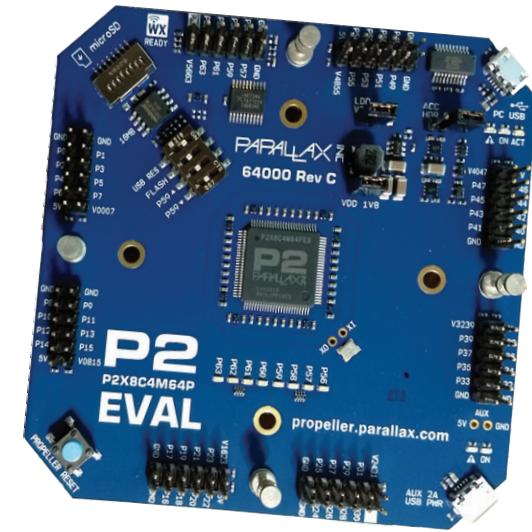


Figure 1. Propeller 2 connecté à un convertisseur USB-série.

Listage 2. Envoi d'un texte, caractère par caractère.

```
'We run in the internal RC giving us 25MHz clockspeed
PUB main()
  pinwrite(56, 1 )
  serial_start()
repeat
  waitms( 250 )
  pinwrite(56, 1 )
  tx("C")
  tx("o")
  tx("d")
  tx("e")
  tx(" ")
  tx("a")
  tx("l")
  tx("i")
  tx("v")
  tx("e")
  waitms( 250 )
  pinwrite(56, 0 )
  tx("1")

PUB serial_start()
  WRPIN( 57, %01_11110_0 )           'set async tx mode for txpin
  WXPIN( 57, ((217<<16) + (8-1)) )  'set baud rate to sysclock/115200 and word size to 8 bits

org
  dirh    #57
end

PUB tx(val)
  WYPIN(57,val) 'load output word
org
  WAITX  #1           'wait 2+1 clocks before polling busy
  wait
  RDPIN  val,#57 WC  'get busy flag into C
  IF_C   JMP  #wait   'loop until C = 0
end
```

pour le débogage. Pour récapituler où nous en sommes : notre code est capable d'envoyer un seul caractère avec les réglages suivants : 115 200 bauds, 8 bits de données, pas de parité, un bit d'arrêt. Le code complet apparaît dans le **listage 1** et il est possible de le télécharger à l'adresse suivante [1].

Si nous voulons maintenant envoyer la chaîne « Code alive » (cf. **listage 2**), il faut effectuer plusieurs appels à la fonction `tx()`, un pour chaque caractère. Si nous voyons des lignes de code qui se répètent, il va être intéressant de réfléchir à généraliser la tâche. Copier des parties de fragments de code partout dans le programme n'est jamais une bonne idée, surtout si vous devez ajouter quelque chose ou corriger le code ultérieurement. Nous construirons une fonction qui prend une chaîne de caractères comme argument, la découpe en caractères séparés et les envoie un par un à notre fonction `tx()`. Cela semble facile, et si vous avez une certaine expérience du codage, cela devrait théoriquement le rester. Si vous utilisez Spin2 ou Spin pour la première fois, il y aura sans doute un peu plus de lecture et de tâtonnements.

Passer une chaîne de caractères

Comme les chaînes de caractères occupent un espace continu de mémoire contenant tous les caractères consécutifs ainsi qu'une terminaison formée d'un zéro binaire (« \0 »), notre action consistera simplement à transmettre l'adresse de départ de cette mémoire à une fonction. Si nous utilisons le langage C/C++, l'opérateur `&` fera l'affaire, en conjonction avec le premier élément. Avec Spin2, la syntaxe est un peu différente, mais le fonctionnement est pratiquement le même. Ce qui diffère, c'est la façon dont nous déclarons la fonction et l'argument lui-même. Le type de variable passé à notre fonction n'est pas spécifié. Si nous n'indiquons aucun type, le langage Spin2 considérera qu'il s'agit d'un type long (32 bits). Il est parfois pratique d'utiliser ce réglage par défaut avec Spin2 pour les variables passées à une fonction, mais au prix d'un certain nombre d'effets secondaires indésirables. Dans notre cas, il s'agit de l'adresse mémoire de notre chaîne de caractères et cela conviendra parfaitement à notre objectif. Ce qui va être différent, c'est la manière de déclarer les variables locales. Nous voyons clairement que toutes les variables, ici `c`, doivent être déclarées en tête de la fonction.

Nous utilisons donc un caractère `|` après le crochet fermant de cette ligne. S'il nous faut plusieurs variables, nous les placerons sur cette ligne en les séparant par des virgules `(,)`. Le **listage 3** montre à quoi ressemblera l'entête de la fonction.

À l'intérieur de la fonction (voir **listage 4**), nous devons procéder à une lecture attentive des données, octet par octet. L'instruction `c := byte[s++]` nécessite quelques mots d'explication. Notre variable `c` n'a actuellement aucun type, ou plutôt, elle possède un type par défaut. En l'absence de la mention `byte[]`, nous lirions 32 bits (variable de type `uint32_t`) en une seule fois ; avec `byte[s++]`, nous lisons un octet à l'emplacement de `'s'`, à l'adresse mémoire où commence la chaîne. Le `s++` ici, combiné avec `byte[]`, garantit que l'adresse ne sera incrémentée que d'un octet. Si nous ne le faisons pas, toutes les opérations appliqueront un format de 32 bits. Comme nous savons maintenant que nous lisons octet par octet, examinons l'instruction `repeat while`. Tant que `c` n'est pas égal (`<>`) au zéro binaire, tout ce qui se trouve à l'intérieur de l'instruction `repeat while` fait partie de la boucle.

À l'intérieur de la boucle, illustrée par le **listage 5**, la fonction `tx()` émet un octet. Une autre ligne récupère l'octet suivant et le place dans notre variable `c`. Ensuite la boucle continue et commence à vérifier si `c` est encore différent du zéro binaire.

Notre fonction `prints()` est donc maintenant en place et nous sommes capables d'envoyer des chaînes de caractères, par le biais d'un UART, à un PC ou un analyseur logique. La suite consiste à lire l'état des broches d'E/S et à transmettre l'information à l'aide de l'UART. Est-il si difficile de lire l'état d'une broche d'E/S ? Eh bien, en théorie, c'est

Listage 3. Entête de la fonction.

```
PUB prints( s ) | c
```

Listage 4. Fonction `prints`.

```
PUB prints( s ) | c
c := byte[s++]
REPEAT WHILE (c <> 0)
  tx(c)
  c := byte[s++]
```

Listage 5. Fonction `tx()` appelée octet par octet.

```
REPEAT WHILE (c <> 0)
  tx(c)
  c := byte[s++]
```

très facile. Parfois, ce sont les petits extras qui font la différence et qui montrent l'étendue des fonctions qu'offrent les broches intelligentes.

(200479-D-04)

Des questions, des commentaires ?

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

Contributeurs

Conception et texte : **Mathias Claußen**

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

Mise en page : **Giel Dols**

Traduction : **Pascal Godart**



PRODUITS

➤ Convertisseur USB>TTL CH340

www.elektor.fr/ch340-usb-to-ttl-converter-uart-module-ch340g-3-3-v-5-5-v



LIEN

[1] [Page de l'article : www.elektor.fr/200479-D-04](http://www.elektor.fr/200479-D-04)



60 ans d'Elektor

Regard sur septembre

Rolf Gerstendorf (Elektor)

Après le numéro double de l'été et les vacances tant attendues, le numéro de septembre a toujours ouvert un nouveau chapitre pour l'équipe d'Elektor. Il présentait des réalisations plus conséquentes, des articles de fond pratiques et, passage obligé, quelques petits circuits intéressants étaient cachés au fil des pages. L'auteur a déniché quelques articles notables de septembre qui lui sont restés en mémoire après 35 ans de pratique assidue d'Elektor. Il va de soi que cette sélection est hautement subjective, mais nous espérons qu'elle piquera votre curiosité.

60 **lektor** **MAG**
sixty > years > young

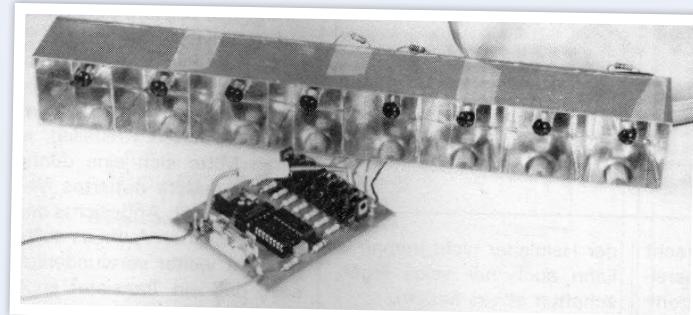
Le joueur de baseball Yogi Berra, disait volontiers « Si vous ne savez pas où vous allez, vous finirez quelque part ailleurs ». Comme tout bon aphorisme, il décline une forme de vérité. Pour une publication comme *Elektor*, la planification est essentielle. Les nombreux rouages de notre propre maison d'édition multinationale sont si imbriqués que nous devons tous suivre le plan et respecter les délais.

Avant, c'était différent. La préparation du numéro suivant tenait plus de la présentation d'une liste de souhaits et ce qui était effectivement publié dépendait de nombreux impondérables : l'auteur a-t-il terminé le circuit ? Y a-t-il des obstacles aux essais en labo ? Parmi les membres du projet qui est indispensable ? Qui est malade ou en vacances ? Pendant la période des vacances d'été, ces questions se font plus pressantes. Après mes congés annuels passés à grimper dans les montagnes, je revenais au bureau et j'y trouvais le numéro de septembre qui m'attendait. Le contenu était parfois étonnant, mais intéressant et c'était une bonne lecture.

Scanner K.I.T.T. (1985)

Dans les années 80, la coupe mulet était très tendance chez les hommes et *David Hasselhoff* était le « roi du cool ». Dans la série originale de *Knight Rider*, David conduisait une *Pontiac Firebird* assistée d'un ordinateur et un motif lumineux circulait sur la calandre quand il réfléchissait. Ici, chez *Elektor*, nous nous sommes passés de toute l'IA, des superordinateurs et de l'interface vocale de *KITT* pour montrer qu'avec une poignée de LED, un oscillateur et un registre à décalage, on pouvait obtenir le même effet lumineux.

Trente ans plus tard, on voit que les concepteurs automobiles d'aujourd'hui utilisent les mêmes effets visuels pour les clignotants, mais finis les compteurs CMOS : des MCU discutent sur le bus CAN du véhicule. En revanche, les capacités de conversation et l'intelligence de *KITT* ne sont pas dans la voiture !



www.elektormagazine.fr/magazine/elektor-198503/52469

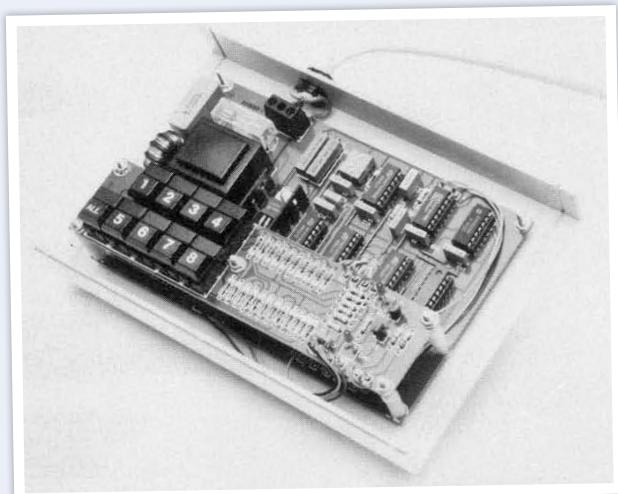
Télécommande du réseau d'éclairage (1991)

Près de la maison d'édition *Elektor*, il y avait une entreprise novatrice, malheureusement mal gérée, appelée *ELSA*. Elle développait et vendait des modems de données de haute qualité et un système de commande de l'éclairage par des signaux sur son réseau d'alimentation. Bien sûr, leurs concepts novateurs ne sont pas passés inaperçus auprès de l'équipe d'*Elektor*. Nous avons conçu notre propre version pour

contrôler l'éclairage et la partager avec vous. Nous voulions y parvenir avec des composants traversants sans un seul microcontrôleur. En feuilletant l'article pour la première fois en 30 ans, j'ai pensé : « Ouah, merci mon Dieu pour les microcontrôleurs ». À l'époque, l'émission et la réception utilisaient tout un tas de CI CMOS numériques. Voici le fonctionnement : un clavier à neuf voies de l'émetteur code la valeur de la luminosité sur 3 bits. Pour communiquer : un bit de départ, trois bits d'adresse, trois bits de luminosité et un bit de contrôle. Le message complet se répète à 40 Hz et module une porteuse à 200 kHz. La porteuse est injectée sur les fils du secteur à l'aide d'une inductance bobinée d'antiparasitage domestique. Aujourd'hui cela semble un peu risqué.

Chaque récepteur écoute le signal de commande (amplification analogique) et recherche tout message dont l'adresse correspond à sa propre adresse de récepteur prédefinie. Si c'est le cas, il récupère le réglage de puissance du message et utilise le contrôle de phase pour régler la puissance de la lampe connectée. Tout cela fut conçu à l'aide de CI numériques CMOS standard !

C'est incroyable : il a fallu trois circuits imprimés pour le récepteur, tous entassés dans le petit boîtier avec juste la place pour une prise de courant !



La société *devolo AG*, issue d'*ELSA AG* produit aujourd'hui des adaptateurs pour automates programmables de haute qualité (sans aucun CI CMOS).

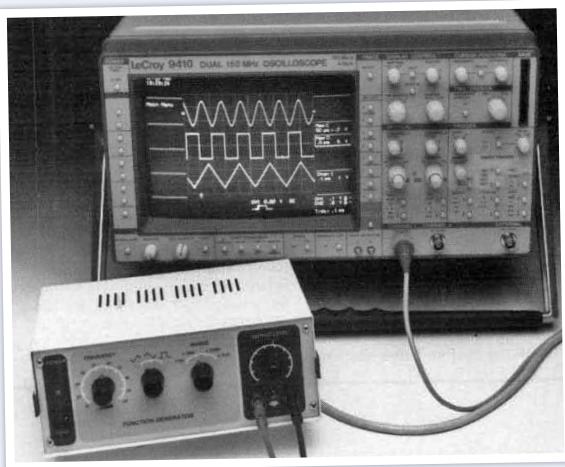
www.elektormagazine.fr/magazine/elektor-199110/35044

Générateur de fonctions simple (1992)

Ce petit générateur de fonctions a attiré mon attention surtout parce que je connaissais le composant central du circuit et cela m'a rappelé une bonne leçon. Ce CI au cœur du montage est l'omniprésent générateur de fonctions intégré XR2206 d'*Exar*. Il est un peu limité en bande passante de sortie, mais tellement facile à utiliser. Je me souviens d'une déconvenue impliquant l'une de ces puces lorsque j'étais étudiant. Au labo, en secouant imprudemment le sac en plastique pour en faire



sortir le CI, je provoquai une décharge statique ; le CI ne s'en est jamais remis. (Jetez un coup d'œil à Elektor 07-08/2021, « ESD – le destroyer fantôme ».) À l'époque, ce CI coûtait presque 50 marks (je crois), ça faisait beaucoup pour un étudiant endurci. J'aurais dû avoir plus de bon sens, mais on vit et on apprend !



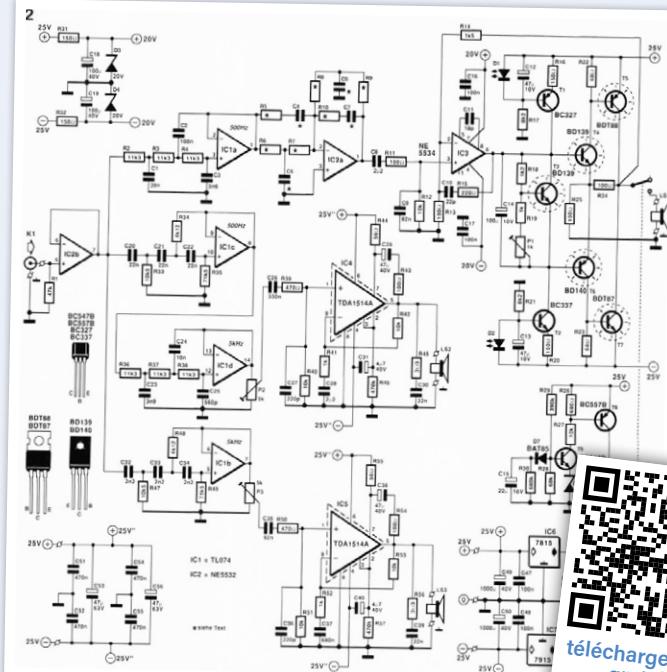
www.elektormagazine.de/magazine/elektor-199209/30004

Haut-parleur actif à 3 voies (1993)

En ces années 70, je tentais de construire un 1^{er} amplificateur audio avec en sortie une paire 2N2955/2N3055. Cela se solda par un ampli et un mois d'argent de poche partis en fumée. Depuis, j'en ai conçu et construit bien d'autres, passifs et actifs. Presque tous les amplificateurs actifs ont un point commun : ils sont encombrants et il leur faut un gros boîtier 19 pouces.

Le concept de plug-in actif à trois voies décrit dans cet article assure une intégration possible de l'amplificateur (avec réglage électronique du recouvrement des filtres) dans l'enceinte. Dans cette réalisation, les niveaux de sortie du filtre Linkwitz-Riley (!) peuvent être réglés à volonté et adaptés aux caractéristiques de l'enceinte et des haut-parleurs. Les TDA1514A (CI amplis de puissance intégrés) pour piloter les tweeters et médiums ont permis de gagner de la place, et un ampli push-pull AB classique en composants discrets fournissait 70 W de puissance dans les basses.

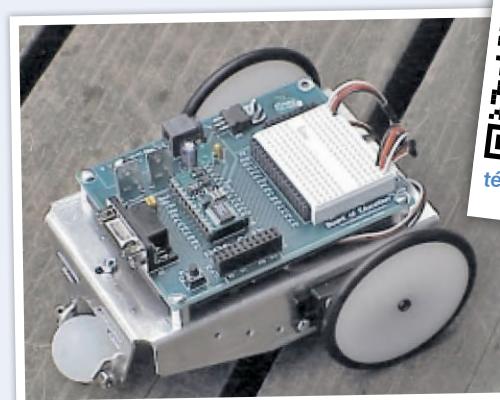
Quelques années plus tard, j'ai recréé ce système actif à trois voies et l'ai installé dans des enceintes pour que ma fille adolescente et ses amis puissent écouter leurs tubes préférés en qualité Elektor. Malgré la reproduction Hifi, les chansons sonnaient toujours assez mal... du moins à mes oreilles.



www.elektormagazine.fr/magazine/elektor-199401/35604

Cours de programmation BASIC Stamp (1999)

Un petit ordinateur monocarte abordable au format DIL, avec BASIC en ROM et de nombreuses ressources (gratuites !) pour le développement de programmes... C'est ce que l'on attend de nos jours d'un nouveau processeur, mais dans les années 90, le concept était neuf et suscita une vague d'intérêt pour ce marché émergent. Ce BoE (*Board of Education*) d'Elektor utilisait un BASIC Stamp 2 de Parallax sur une carte mère (avec zone de prototypage pour des extensions de matériel) sur un châssis simple. Avec des capteurs supplémentaires, développer une application de type robot était possible. Le matériel fut utilisé pour une série de six cours de formation couvrant le développement de programmes. Elektor a une longue tradition de cours de microinformatique. Vous pouvez également vous mettre à jour sur la version la plus récente du BASIC-Stamp en consultant notre cours pratique actuel pour le Parallax Propeller 2.

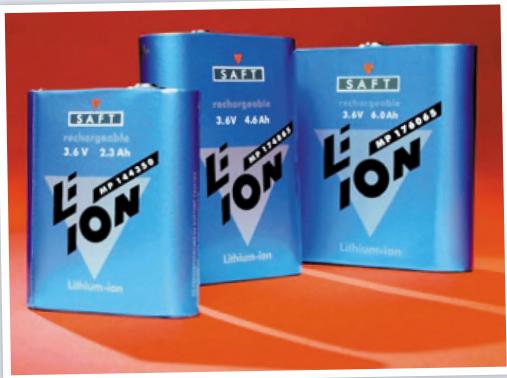


www.elektormagazine.fr/magazine/elektor-199909/37156

Cellules lithium-ion (2001)

Au moment où les accumulateurs NiCd, nuisibles à l'environnement, étaient remplacés par des NiMH sans métaux lourds, des batteries d'un genre nouveau, lithium-ion, apparaissaient pour les appareils mobiles. Au début des années 90, ces batteries lithium-ion étaient encore très chères et le processus de charge très délicat. Cet article de fond souligne bien les avantages des batteries lithium-ion : haute capacité,

petit volume, faible poids et tension élevée aux bornes ; et, comme vous pouvez l'attendre d'Elektor, il aborde aussi des circuits de charge utiles avec des CI spécialisés.

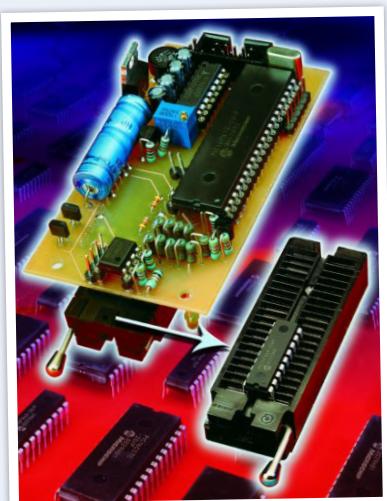


www.elektormagazine.fr/magazine/elektor-200109/9259

PICProg 2003 (2003)

À l'aube de l'ère des microprocesseurs, les outils de développement matériels et logiciels étaient d'un coût prohibitif pour les « pauvres » amateurs. Le passage à une communauté plus « ouverte », dont Elektor fait partie, œuvrant via l'internet avec ses forums, ses groupes de soutien et ses héros du codage méconnus, donnant leur temps et leur expertise gratuitement, a sans doute fait réagir les fabricants de puces. S'ils pouvaient familiariser davantage de personnes avec leurs produits et leurs outils de développement, ils finiraient par expédier davantage de puces. Côté rentabilité, il n'était pas judicieux d'imposer un investissement initial élevé. Mieux valait offrir les outils et absorber leur coût dans le prix unitaire des puces.

Ce programmeur de PIC est un bon exemple d'alternative bon marché à une unité de programmation professionnelle coûteuse. Le concept du PIC-Prog ne comprenait en fait qu'un PIC programmé, un support ZIF à 40 broches, une alimentation et un convertisseur RS-232 pour communiquer avec un PC. Le logiciel offrait de le configurer pour travailler avec plus de 85 modèles de PIC !



www.elektormagazine.fr/010202

Technique RFID (2006)

Aller au fond des nouvelles techniques et les expérimenter à l'aide d'applications pratiques a toujours été une spécialité d'Elektor. Dans le numéro de septembre 2006, la terminologie de base et la fonction des différents types de puces RFID ont été examinées à la loupe. On y trouvait aussi une carte RFID Elektor (gratuite !) qui pouvait être activée, écrite et lue en utilisant un graveur/lecteur RFID fait maison. Pour les passionnés, un circuit expérimental doté d'un microcontrôleur standard et n'utilisant pas de puce RFID spéciale fut également étudié.

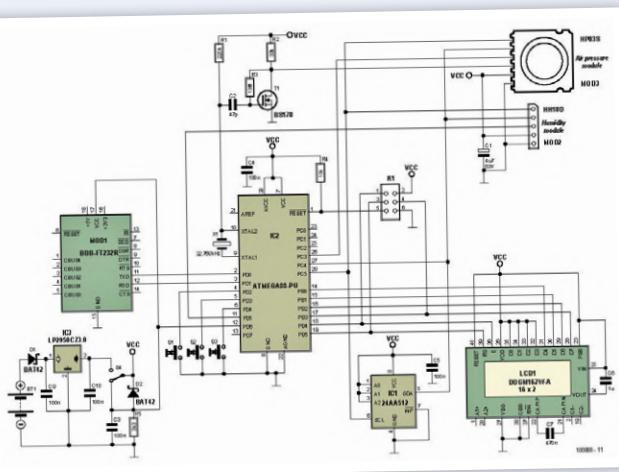


www.elektormagazine.fr/060204
www.elektormagazine.fr/060132
www.elektormagazine.fr/060132-II
www.elektormagazine.fr/060221

Enregistreur météo USB longue durée (2011)

Grâce aux microcontrôleurs à faible consommation d'énergie dotés de modes d'économie d'énergie et de veille sophistiqués, il est possible de faire fonctionner des systèmes à microprocesseur distants sur des batteries. Le matériel requis pour des applications assez complexes a été réduit à un minimum absolu (par rapport à l'ère analogique). Cet enregistreur météo est un bel exemple de ce type d'application. Le circuit contient un contrôleur AVR avec alimentation, une EEPROM, un écran et deux capteurs qui mesurent humidité, pression et température de l'air et communiquent avec le contrôleur par une interface I²C. Un pont série-USB fournit un lien de communication avec le logiciel PC afin que les valeurs enregistrées puissent être affichées et analysées. Les capteurs MEMS n'étaient pas si nouveaux à l'époque, mais ils figuraient rarement dans les projets des amateurs ou n'étaient pas connus de la communauté des makers. Un article du numéro suivant d'Elektor (10/2011) s'intéressait aux capteurs I²C.





www.elektormagazine.fr/magazine/elektor-201109/11973
www.elektormagazine.fr/magazine/elektor-201110/11981

Swiss Pi (2016)

Au début des années 2010, le *Raspberry Pi* fit irruption dans l'arène des fabricants. Le bon vieux temps du piratage matériel et logiciel à l'aide d'un Atari ou d'un C64 semblait revenu. Une ou deux faiblesses du *Raspberry Pi* sont toutefois rapidement apparues à cet égard. Cela dit, compte tenu de son prix ridiculement bas, nous pouvions lui pardonner. Parmi ces faiblesses, citons l'utilisation de ports GPIO non protégés, une prise en charge du *PWM (MLI)* rudimentaire, l'absence d'entrée analogique...

Par bonheur, le *Raspberry Pi* est extensible et on peut y brancher diverses cartes *Hardware-on-Top (HAT)* pour en compenser les lacunes. Le HAT Swiss Pi, en particulier, est doué pour ajouter quelques fonctions utiles comme une horloge en temps réel alimentée par une batterie, des interfaces RS-485 et I²C. Le logiciel d'assistance (serveur et bibliothèque) du Swiss Pi fonctionne sous Windows et Linux.

Ce module HAT particulier, compatible avec toutes les versions du *Raspberry Pi A+/B+/2B/3B/Zero/4B*, est rapidement devenu un best-seller et, cinq ans plus tard, il est toujours disponible dans la boutique Elektor. Cinq ans – c'est une éternité en électronique !



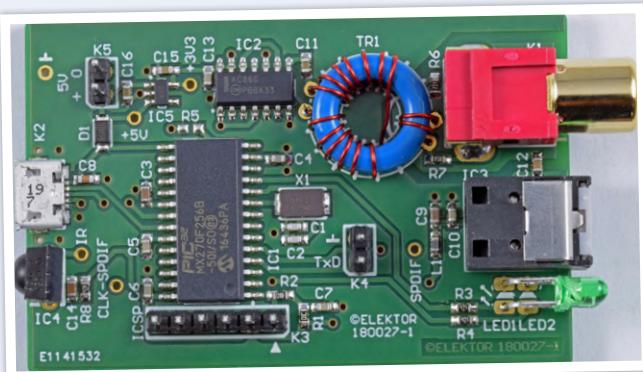
www.elektormagazine.fr/magazine/elektor-201609/39774

Interface USB-S/PDIF (2020)

Si vous souhaitez équiper votre ordinateur d'une interface S/PDIF audio de qualité, il existe des solutions prêtées à l'emploi. Qu'elles soient bon marché ou chères (beaucoup viennent d'Asie, mais certaines sont fabriquées en Europe), elles ont un point commun : ce sont des boîtes noires. On ne sait pas ce qu'il y a à l'intérieur et on ne peut pas savoir si on en a pour son argent. Chez Elektor, nous savons ce qui intéresse nos lecteurs : la réalisation de circuits qui combinent une technologie de pointe et un faible coût, mais également qui sont décrits en détail quant au fonctionnement du circuit.

Cette interface USB-S/PDIF en est un exemple presque classique. Le contrôleur central offre assez de périphériques pour décoder les signaux audio USB, les stocker sous forme de trames S/PDIF et les sortir de façon synchrone via SPI. Le signal est ensuite disponible sur un connecteur Toslink ou isolé via un connecteur Cinch. Et ce n'est pas tout ! Une télécommande IR peut également contrôler l'interface S/PDIF.

Profitant des confinements « Covid », j'en ai fabriqué une demi-douzaine et équipé le boîtier multimédia *Raspberry Pi* de mon salon, mon camping-car et aussi enrichi d'autres appareils multimédia domestiques d'une connexion audio « haut de gamme ». 



www.elektormagazine.fr/180027

(210343-04)

Contributeurs

Auteur et rédaction : Rolf Gerstendorf

Traduction : Yves Georges

Mise en page : Harmen Heida

dans l'antre de...

Menno van der Veen et de ses tubes amplificateurs

Menno van der Veen (Pays-Bas) et Eric Bogers (Elektor)

Réputé pour ses amplificateurs à tubes haut de gamme et ses transformateurs de sortie toroïdaux, Menno van der Veen est une personnalité bien connue des audiophiles. Talentueux passeur de savoir et de savoir-faire, il enseigne également la conception des amplis à tubes au sein de son académie TubeSociety.



Figure 1. L'ascension d'un homme et d'un labo : d'un étroit sous-sol...



Figure 2. ...aux vastes combles d'une maison.

Le petit monde des audiophiles se divise en deux camps : il y a d'un côté ceux qui préfèrent les amplificateurs à tubes, de l'autre ceux qui ne jurent que par les transistors. Certains partisans de ces deux camps adoptent des positions si tranchées que leurs débats virent parfois au dialogue de... sourds. Voilà une situation bien triste, car ces deux technologies ont en fait leurs propres caractéristiques, avantages et inconvénients. Et en fin de compte, la seule chose qui importe est le plaisir d'écoute qu'elles apportent. Menno van der Veen est depuis son adolescence un adepte des amplis à tubes, mais en aucun cas une personne « audio-rigide ».

J'ai commencé à construire des amplis à tubes à l'âge de quatorze ans – à vrai dire en copiant les conceptions d'autres électroniciens, car à l'époque mes connaissances sur le sujet étaient loin d'être suffisantes. Ce n'est qu'après mes études de physique appliquée à l'université de Groningue (Pays-Bas) que j'ai commencé à comprendre ce qu'étaient les tubes et que j'ai pu réaliser mes propres circuits.

J'ai publié ma première conception en 1983 : un ampli à quatre

EL84 avec étage final équilibré et transformateur de sortie toroïdal – transfo qui a par ailleurs suscité de nombreuses réactions. En 1987, j'ai construit et publié un amplificateur de 100 W doté une nouvelle fois de transformateurs toriques, et là encore ces transfos m'ont valu des critiques de la part de pratiquement tous les audiophiles des Pays-Bas : « Ce n'est pas comme ça qu'il faut faire, tout le monde utilise des noyaux EI pour les transfos de sortie ». Raison de plus à mes yeux pour poursuivre sur ma lancée. J'ai entamé une étude approfondie des transformateurs (et en ai publié le résultat), puis j'ai commencé à écrire des ouvrages sur les tubes et les transformateurs.

Comme beaucoup, j'ai débuté ma carrière d'électronicien dans un modeste laboratoire aménagé dans le sous-sol de ma maison (**fig. 1**). Plus tard je l'ai déménagé dans un grand bâtiment de la ville de Zwolle, et depuis 1984 je vis et travaille dans la Frise, une province agricole des Pays-Bas. Si mon labo est aujourd'hui si spacieux (**fig. 2**) c'est qu'il accueille aussi les étudiants de mon académie TubeSociety. Lorsque j'y suis seul, je travaille essentiellement sur la paillasse visible à l'arrière-plan.

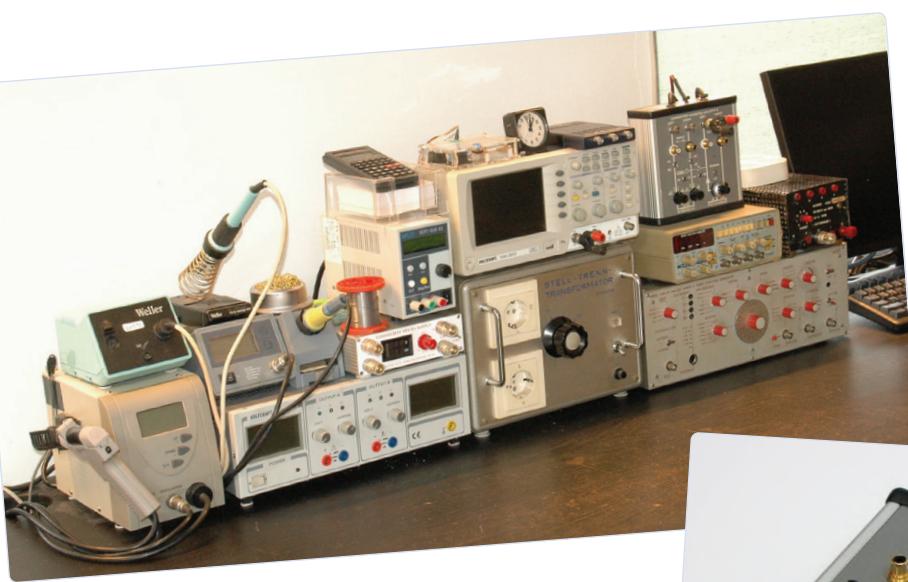


Figure 3. L'unique matériel de Menno.



Figure 4. Le système de mesure ARTA-2.



Figure 5. Un transformateur Vanderveen à noyau toroïdal.

Contrairement à ce que l'on pourrait croire, je suis assez peu outillé (**fig. 3**) : une station de soudage, des alimentations, des oscilloscopes, des oscillateurs, et le système de mesure ARTA-2 (**fig. 4**). Le boîtier ARTA-2 (que j'ai moi-même conçu et commercialisé avec l'aide de TentLabs) permet de mesurer les caractéristiques d'un signal audio avec une (bonne) carte son. Il me permet d'analyser un signal « en profondeur », jusqu'à -140 dBV ($0,1$ μ Vrms), et d'en visualiser les irrégularités les plus petites. Les schémas et dessins des circuits imprimés de l'ARTA-2 sont désormais publics, chacun peut donc assembler sa propre unité [1].

En ce moment je mène de front quatre activités. Je conçois des transformateurs de sortie toroïdaux pour amplis à tubes ; ils étaient à l'origine fabriqués par Plitron et Amplimo, mais le sont maintenant par Trafco (**fig. 5**). Je conçois bien sûr aussi des amplis à tubes, dont

le récent Trans-SE10. Je rédige des articles et des livres [2][3][4], et enfin j'enseigne dans mon académie TubeSociety [5].

On me demande souvent pourquoi je publie mes conceptions et pourquoi j'en enseigne les principes à mes étudiants. La plupart du temps je réponds que je pourrais les protéger par des brevets et les commercialiser, mais que dans le meilleur des cas le retour se résumerait à la réaction de quelques clients et à d'hypothétiques mentions dans des revues spécialisées. Partager mes connaissances avec des étudiants enthousiastes est pour moi plus gratifiant. Je les vois s'améliorer, leurs retours m'enrichissent, et j'apprécie également de les voir mettre en pratique ce que je leur transmets – et en plus je conçois mes propres produits. Je pense être plutôt un enseignant dans l'âme, heureux de permettre à tout un chacun de profiter du « filon audio » que j'ai découvert.

L'ampli Trans-SE10

J'ai récemment mis sur le marché (en collaboration avec Tentlabs) un amplificateur remarquable, le Vanderveen-Trans-SE10 (fig. 6). Il s'agit d'un amplificateur à tubes de 2×10 W à sortie asymétrique qui exploite quelques techniques inhabituelles, et s'appuie sur la rétroaction locale pour supprimer autant que possible la distorsion dans le transformateur et le tube de sortie.

La contre-réaction divise elle aussi les audiophiles de façon plus ou moins irréconciliables : il y a ceux qui la considèrent comme une aubaine pour les amplificateurs audio, et ceux qui la voient comme un fléau qui rend les micro-détails de l'image sonore quasiment imperceptibles. J'ai essayé d'adopter un juste milieu avec le SE10 : plutôt que d'utiliser la contre-réaction globale de la sortie vers l'entrée, je corrige les défauts de l'amplificateur localement, en excluant explicitement le transformateur de sortie de la boucle de contre-réaction. Cette façon de procéder combine les avantages des deux approches.

Comme le tube de sortie est attaqué par une source de courant commandée en tension (VCCS), la contre-réaction est de 100 %. Ou, dit autrement :

le tube de sortie utilise l'ensemble de son gain pour se corriger. La distorsion harmonique du tube est donc considérablement réduite et la résistance interne devient très petite. Ce point est important, car la distorsion dans le transformateur de sortie est alors minimisée. Le réglage de la source VCCS est optimisé par un circuit de contre-réaction local supplémentaire afin que l'amplificateur ait une distorsion très faible. Le résultat est un amplificateur présentant environ 70 dB de contre-réaction locale (ce qui, pour des raisons de stabilité, ne peut jamais être atteint avec une contre-réaction globale). Les haut-parleurs « libèrent » ainsi une image sonore dont les micro-détails n'ont pas disparu.

Ai-je atteint ici l'objectif ultime de toutes mes recherches ? Je ne le pense pas, car je suis en train d'adapter cette technique à des puissances de sortie élevées. L'étude est en cours, donc soyez patient !



Figure 6. Le Trans-SE10 Vanderveen.

À SUIVRE ?

Vous souhaitez en savoir plus sur le Trans-SE-10 ou d'autres amplificateurs à tubes ? Écrivez-nous à redaction@elektor.fr, si la demande est nombreuse nous y consacrerons un article complet dans un prochain numéro.

Des questions, des commentaires ?

Contactez l'auteur ou Elektor (redaction@elektor.fr).

Contributeurs

Texte et photos :

Menno van der Veen

Rédaction : **Eric Bogers**

Traduction : **Hervé Moreau**

Mise en page : **Giel Dols**

LIENS

- [1] **ARTA-2** : <https://bit.ly/38zzIPb>
- [2] **Menno van der Veen, Modern High-End Valve Amplifiers (E-book)**, Elektor : www.elektor.fr/modern-high-end-valve-amplifiers-e-book
- [3] **Menno van der Veen, High-End Valve Amplifiers 2 (E-book)**, Elektor : www.elektor.fr/high-end-valve-amplifiers-e-book
- [4] **Menno van der Veen, Designing Tube Amplifiers (E-book)**, Elektor : www.elektor.fr/designing-tube-amplifiers-ebook
- [5] **TubeSociety** : <https://bit.ly/3l9fotf>

circuits hybrides

David Ashton (Australie)

Il peut arriver qu'aucun composant disponible sur le marché ne remplisse pas la fonction dont on a besoin. Dans ce cas, on réalise de petits circuits composites utilisables comme des composants standards. Composés avec des circuits intégrés et des composants discrets ordinaires, ces hybrides peuvent être aussi fascinants à explorer qu'attrayants visuellement.

Qu'est-ce qu'un hybride ? En principe, c'est un petit circuit imprimé conçu pour prendre place sur un circuit imprimé plus grand afin de remplir une fonction spécifique requise par l'application. Les hybrides sont généralement constitués d'un substrat en céramique avec des pistes imprimées, sur lequel on monte des composants discrets (résistances, condensateurs, transistors, diodes, circuits intégrés, etc.). Comme ils combinent de nombreuses technologies sur une seule carte, ils sont parfois appelés circuits intégrés hybrides (CIH), le plus souvent abrégés en hybrides. Ils permettent de gagner beaucoup de place sur une carte en raison de leur haute densité de composants. Et, comme ils peuvent utiliser des composants ordinaires, ils sont beaucoup

plus polyvalents que les circuits intégrés. Un petit circuit imprimé installé sur une carte mère : carte fille ou hybride ? Débat intéressant !

Au départ, un hybride est un rectangle de céramique vierge, de taille pouvant atteindre quelques centimètres, sur lequel on imprime des pistes et des résistances au moyen d'un procédé à couche épaisse. Au besoin, les résistances peuvent être automatiquement ajustées à des valeurs très précises par laser. Les condensateurs et les composants actifs, généralement de type CMS, sont ensuite soudés, ainsi que les broches nécessaires à la connexion à la carte mère. Enfin, le circuit peut être mis en boîtier ou enrobé pour le protéger. L'hybride de la **figure 1** est inhabituel, car

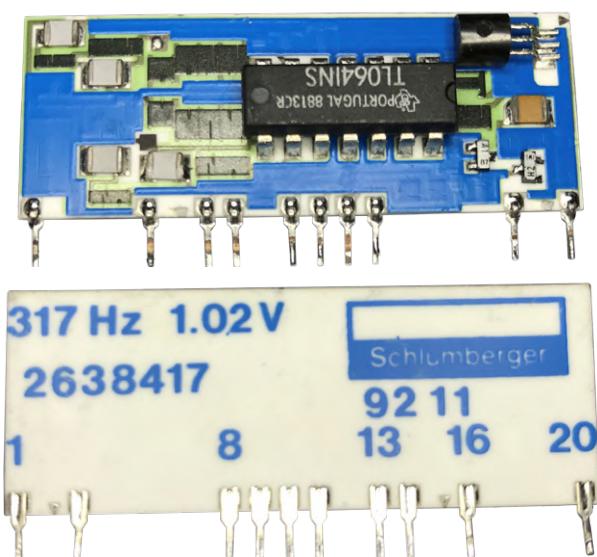


Figure 1. Cet hybride décode des commandes véhiculées par le secteur.

des composants traversants ont été adaptés pour être utilisés comme des CMS. Il sert de filtre à un relais à injection de fréquence. Il capte les commandes codées sous forme d'impulsions à une fréquence particulière et superposées au secteur. Ces commandes servent à allumer et éteindre des lampadaires ou des chauffe-eaux. Le circuit intégré est un quadruple amplificateur opérationnel DIP TLO64 aux pattes partiellement coupées, et le composant en boîtier TO-92 est un régulateur de tension 79L05. Les résistances imprimées sont ajustées au laser, dont les entailles sont clairement visibles sur les composants situés juste à gauche du TLO64. Le réglage des valeurs des composants pendant la fabrication permet de choisir la fréquence de fonctionnement du circuit, 317 Hz pour celui-ci.

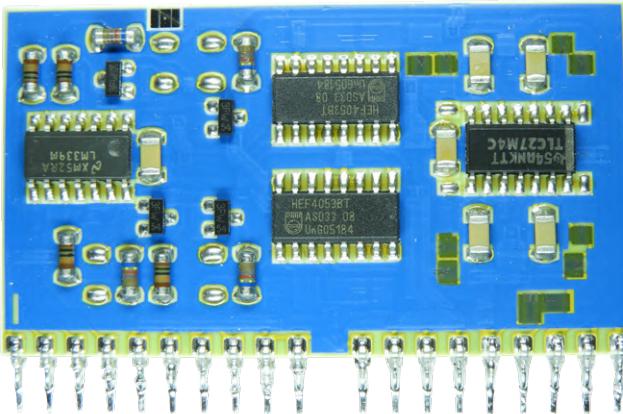


Figure 2. Combinant des circuits CMS avec des résistances discrètes et imprimées en couche épaisse, ce module fonctionne comme un multiplexeur analogique.

La **figure 2** montre un hybride utilisé dans un module d'entrée analogique d'un automate programmable. Les circuits intégrés sont des multiplexeurs analogiques 4053, un quadruple amplificateur opérationnel TLC27M4, et un quadruple comparateur LM339. Le dispositif achemine un signal sélectionné par le système vers un CA/N sur la carte mère. Dans cet exemple, les composants sont des CMS, avec quelques résistances imprimées et réglées au laser.

Notre troisième exemple peut tout juste être qualifié d'hybride (**figure 3**). Utilisé dans un dispositif de télécommunication, ce circuit simple se compose de deux résistances de précision de $600\ \Omega$. Pour la photographie de ce composant, l'éclairage a dû être soigneusement étudié afin de rendre visible l'ajustage au laser. Un segment d'un pouce d'un mètre ruban, en arrière-plan, donne l'échelle.

On trouve également des hybrides en boîtiers comme celui de la **figure 4**. Ce dispositif est un pilote utilisé dans un climatiseur. Si celui-ci est d'origine Hitachi, des sociétés telles que Sanken fabriquent depuis longtemps des amplificateurs audio et des alimentations hybrides dans un format similaire.



Figure 4. Entièrement encapsulé, ce pilote de climatiseur ne laisse aucune possibilité d'évaluer son fonctionnement.

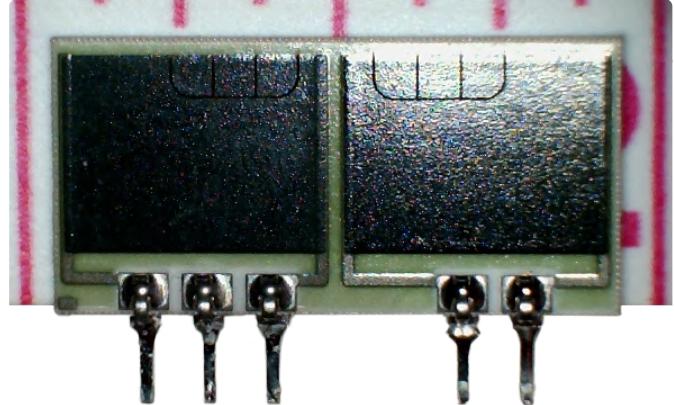


Figure 3. Utilisé en télécommunication, ce module, à la limite de l'hybride, se compose de deux résistances de précision.

Pour mieux résister aux agressions physiques et chimiques, les hybrides sont souvent enrobés, comme le circuit de la **figure 5**. Cela les rend très robustes, mais dissimule aux curieux (comme moi !) toute information sur leurs composants ou leur usage.

Il n'est malheureusement pas facile de soutirer des informations aux hybrides, car ils sont spécifiques à chaque fabricant, à chaque appareil et à chaque fonction. Vous pouvez parfois reconstituer leur schéma, ce qui peut se révéler une expérience d'apprentissage instructive. Mais il est peu probable

qu'ils aient un intérêt pour les bricoleurs : ils sont trop spécifiques dans leur conception pour être d'un usage général. En revanche, ils sont souvent très décoratifs et, à leur manière, agréables à regarder !

(210358-04 - VF : Helmut Müller)

Des questions, des commentaires ?
Contactez Elektor (redaction@elektor.fr).

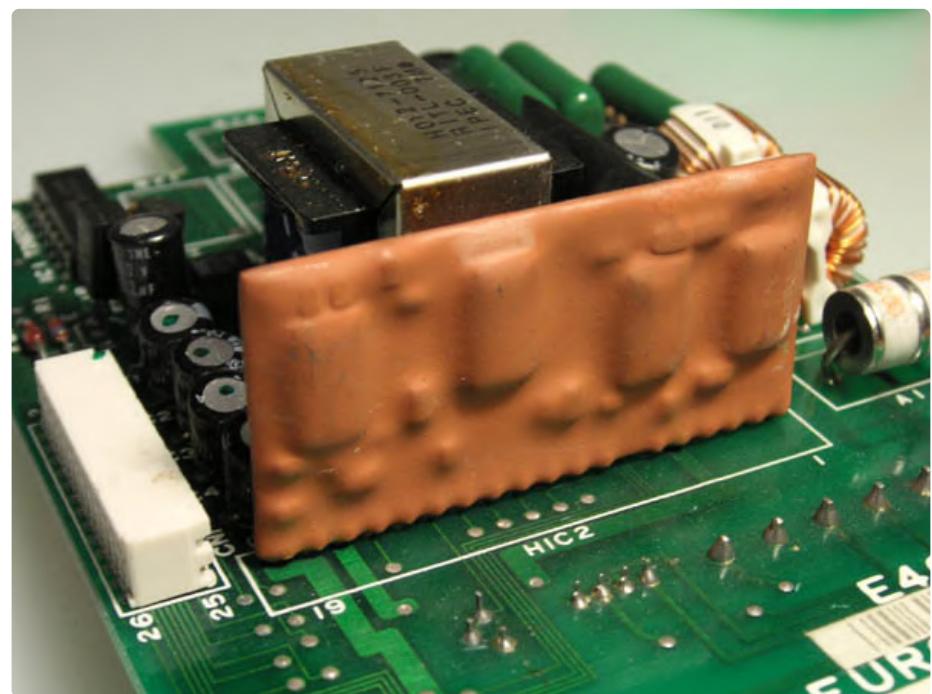
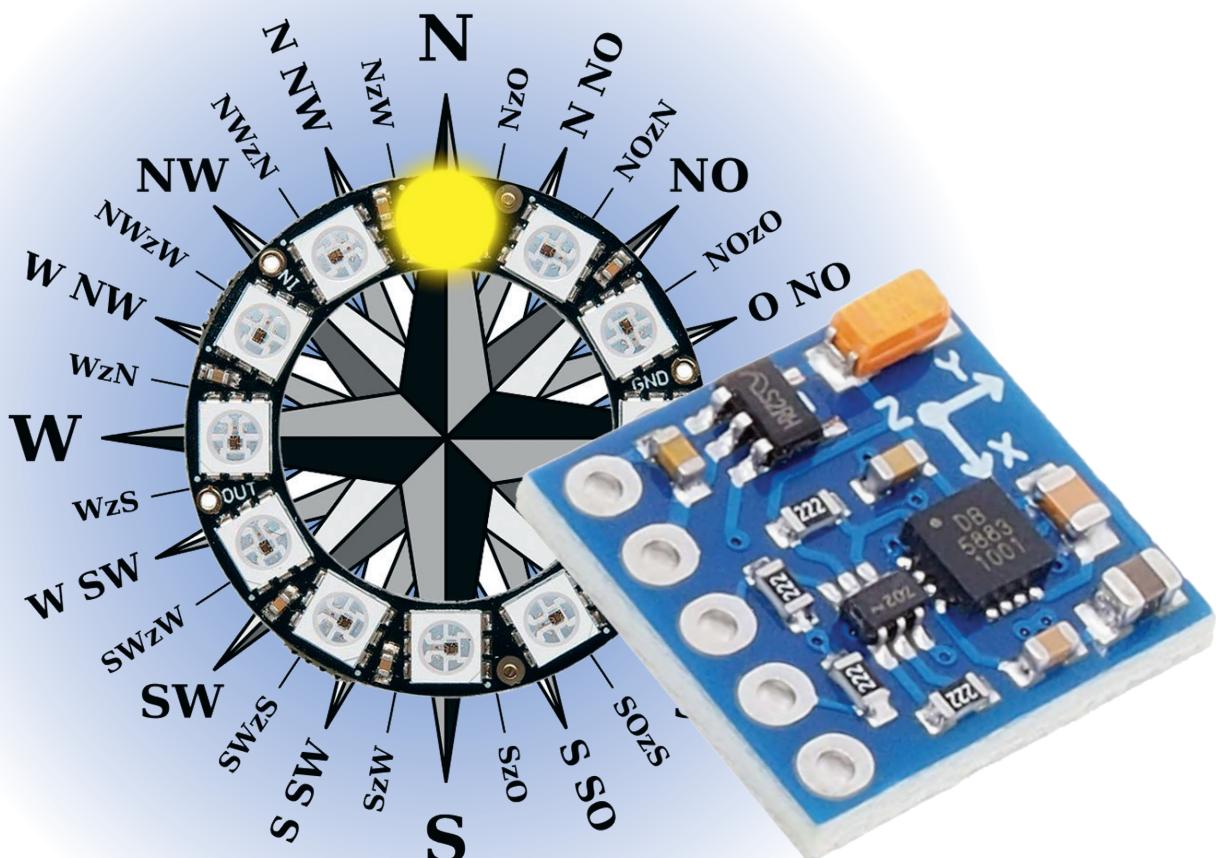


Figure 5. L'enrobage protège les hybrides des agressions physiques et chimiques, mais également de la curiosité du bricoleur. (Source : Janke / https://en.wikipedia.org/wiki/Hybrid_integrated_circuit)

rose des vents

avec le capteur GY-271

Ou pourquoi il faut faire des huit pour calibrer un capteur



Cette magnifique rose des vents est une œuvre de Soeren Gasch, mise dans le domaine public : https://commons.wikimedia.org/wiki/File:Compass_rose_german.svg.

Rolf Hase (Allemagne)

Quoi de plus simple que de réaliser une boussole électronique à partir d'un capteur de champ magnétique sur une carte d'expérimentation, d'un anneau de LED NeoPixel comme rose des vents, et d'un Arduino qui gère le tout avec un logiciel tout fait ? Mais quand ça ne marche pas comme prévu, il ne reste que le « retour aux fondamentaux » pour mettre le doigt sur un défaut surprenant et trouver une solution qui ne l'est pas moins !

Autrefois, lorsqu'on partait faire une balade à vélo en territoire inconnu, il fallait emporter une carte. Aujourd'hui, un ordiphone dans son support sur le guidon suffit, ou, si l'on

veut une image plus grande, une tablette dans le panier avant. Sur mon ordiphone, j'ai installé l'application OsmAnd, qui utilise les cartes d'OpenStreetMap. Mon appareil n'est

malheureusement pas équipé d'une boussole et je dois me contenter de la minuscule flèche de direction du parcours affichée sur l'écran d'OsmAnd qui disparaît lorsque

je m'arrête, me privant de toute possibilité d'orientation. Sur l'écran, la carte se met alors à tourner dans tous les sens, et je dois me déplacer d'une vingtaine de mètres (dans la mauvaise direction, évidemment) pour pouvoir me réorienter correctement. C'est ainsi que j'ai eu l'idée d'emporter avec moi une boussole distincte, bien visible et dépourvue d'une aiguille se mettant à danser à la moindre secousse. C'est en feuilletant d'anciens numéros d'*Elektor* qu'il me vint l'idée d'une solution électronique qui, au moins une génération d'électroniciens plus tard, devrait encore gagner en fiabilité et en précision.

Une solution utilisant un Arduino Nano, un anneau de LED NeoPixel et le module GY-271 est présentée sous [1], avec le schéma du circuit et son code source. Elle me semblait tout à fait satisfaisante, au point que j'ai fait l'acquisition du module GY-271. Comme j'aime avoir plusieurs exemplaires du même composant sous la main, j'en ai commandé trois exemplaires à peu de frais sur l'internet.

Module boussole GY-271

Le GY-271 est un module d'extension conçu pour l'Arduino, et sa rangée de broches est parfaitement compatible avec une carte d'expérimentation ordinaire. Le circuit intégré principal du module est une boussole à trois axes portant la référence HMC5883L ou QMC5883 : ce circuit est souvent présent dans les ordiphones et constitue donc un produit de masse bon marché. La **figure 1** montre le circuit interne de la puce. Les autres composants sur le module sont un régulateur de tension à faible chute de 3,3 V, quelques condensateurs de filtrage et de dérivation et deux adaptateurs de niveau pour l'interface I²C. On peut en déduire que le module peut être alimenté par une source de 3 à 5 V et qu'il communique par un bus I²C avec un microcontrôleur (sur une carte Arduino, Raspberry Pi ou n'importe quelle autre carte « intelligente »). Le protocole de communication est décrit de manière très détaillée dans la documentation d'accompagnement et dans un e-book d'applications gratuit disponible en plusieurs langues.

Comme il n'y a pas de signaux analogiques à traiter, le câblage du circuit se réduit à une connexion, sans se soucier de son alimentation. L'alimentation de l'anneau de LED passe par deux pastilles soudables, le plus et la masse, et un seul signal logique d'horloge suffit aux NeoPixels. Pour gagner de la place, un Arduino Pro Mini a été utilisé,

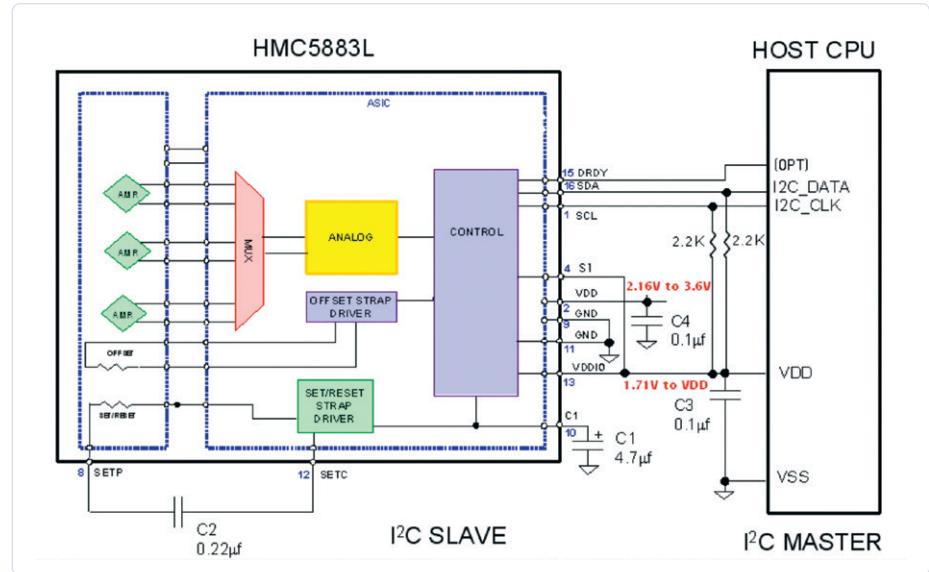


Figure 1 : Circuit interne du capteur boussole HMC5883L (source : fiche technique Honeywell).

et l'EDI Arduino doit être configuré en conséquence. Un adaptateur FTDI-USB-série est nécessaire pour programmer l'Arduino.

Jusqu'ici, tout ne va pas très bien

En théorie, donc, la construction de la boussole et son installation sur mon vélo ne devaient pas présenter de difficulté. Mais dès le premier essai, patatras ! Seul un quadrant a fonctionné, puis l'« aiguille » a fait un saut

de 180°. L'examen du code source (retiré depuis) n'a révélé aucune erreur, alors j'ai mis le problème sur le compte de la médiocre qualité des produits sans marque achetés sur eBay.

En bref, en tant que praticien, il ne me restait plus que la méthode empirique pour découvrir la cause du problème. Les premières expériences ont montré que l'appareil était très sensible aux champs externes, par exemple ceux dans le voisinage

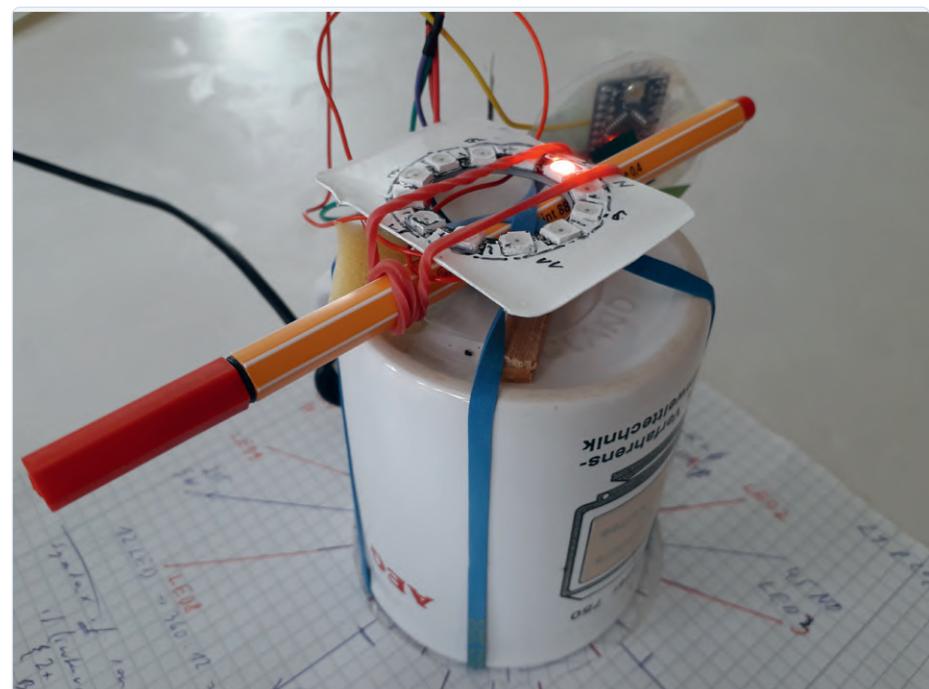


Figure 2 : Un montage expérimental astucieux pour traquer le bogue.

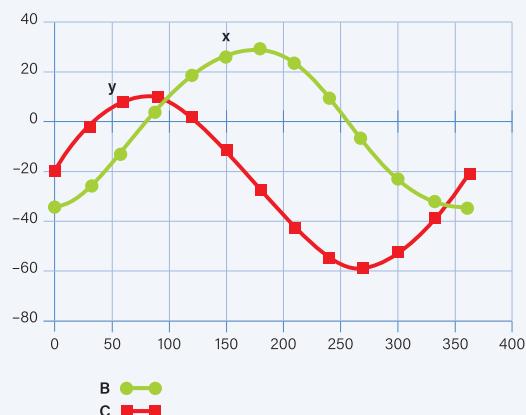


Figure 3 : Les relevés montrent un décalage du zéro, cause du comportement incorrect.



Figure 4 : Voici l'aspect idéal des deux courbes.

de mon ordinateur portable. Comme il n'est pas recommandé d'expérimenter avec un circuit d'essai qui n'est qu'un foisonnement de fils volants, j'ai construit un dispositif expérimental plus robuste, comme le montre la **figure 2**. Voici ce qu'on peut y voir : une tasse à café renversée, garantie magnétiquement saine, supporte la carte d'essai, le capteur (non visible sur cette photo) et l'anneau de LED, tous fixés à son fond à l'aide d'élastiques, en compagnie d'un stylo servant d'indicateur du « nord ». Un petit morceau de bois et une éponge servent à la fois de cale et de coussinet. Les directions des axes x et y sont indiquées sur le côté composants du module GY-271. La pointe rouge du stylo correspond à la direction de la LED 0 et à la direction x négative (!) du capteur, qui est monté avec le côté composants vers le bas. L'appareil n'utilise pas l'axe z du capteur, mais il faut le régler pour que les valeurs correspondantes soient nulles afin de minimiser les perturbations du système. (Ces valeurs, après un petit traitement mathématique, pourraient être utilisées pour mesurer des pentes

montagneuses « hors catégorie »). La tasse repose sur une feuille de papier avec le dessin d'une rose des vents rudimentaire. Un cercle central a été découpé pour pouvoir la faire tourner facilement. Et, mine de rien, ce dispositif a non seulement fourni des résultats reproductibles, mais a également produit un effet « Eurêka ! » remarquable ! J'ai donc tourné la boussole-tasse à café et saisi les valeurs des axes x et y dans la feuille de calcul Excel illustrée à la **figure 3** (valeurs de l'axe x en bleu, axe y en rouge). Si l'on compare ces valeurs avec les valeurs idéales indiquées à la **figure 4**, on constate immédiatement un écart. Les tracés sont ceux d'une courbe sinus et d'une courbe cosinus (ou d'une courbe sinus décalée de 90°). Vieux souvenirs d'école : dans un triangle rectangle, le sinus d'un angle est égal au côté opposé sur l'hypoténuse, le cosinus est égal au côté adjacent sur l'hypoténuse, et la tangente est égale au côté opposé sur le côté adjacent, ou au sinus divisé par le cosinus, l'hypoténuse se simplifiant. Pour déterminer l'angle à partir des côtés, il nous faut la fonction inverse, l'arc tangente, ou arctan.

Les bibliothèques Arduino comprennent une fonction arctan2, qui prend les valeurs x et y comme arguments. Une normalisation à 1 n'est pas nécessaire, mais les amplitudes doivent être égales. Il devient évident que le décalage du zéro est la cause du problème. Cela explique également cette danse bizarre en forme de huit à laquelle on soumet certains ordiphones pour calibrer leur boussole : il est clair que ces ordiphones utilisent des capteurs magnétiques HMC ! J'ai entendu dire que, dans certains cas, cette chorégraphie (ou devrais-je dire « octégraphie » ?) ne fonctionne pas, apparemment à cause d'un décalage du zéro trop grand pour être corrigé. Une procédure analogue permettrait de réaliser un calibrage « automatique » sur l'Arduino, en déduisant les corrections des valeurs maximales et minimales du capteur. Mais l'écriture des valeurs de calibrage vers le capteur nécessiterait une communication bidirectionnelle, donc deux broches d'interface au lieu d'une. Ayant enregistré la totalité de la courbe caractéristique du capteur, nous pouvons calculer le décalage et le reporter dans le programme plutôt que de compliquer le câblage.

Nouveau programme

Un avantage significatif du logiciel original est qu'il inclut tout le code nécessaire pour initialiser et lire la puce HMC. J'ai apporté quelques modifications au code du moniteur, rendu les valeurs de lecture plus faciles à gérer en les divisant par 100, car, comme l'anneau de LED NeoPixel n'a que 12 positions, une précision d'environ $\pm 10\%$ suffit. Si vous envisagez d'utiliser un anneau NeoPixel plus grand ou un écran graphique, vous devrez adapter ces valeurs brutes à la résolution dont vous avez besoin. Pour piloter les NeoPixels, j'ai dégraissé le code très clair disponible sous [3] et obtenu une

Listage 1. Correction du décalage codée en dur.

```
Serial.print("X Value: ");
x = x/100;
x = x+2;
Serial.print(x);
Serial.print(" Y Value: ");
y = y/100;
y = y+20;
Serial.print(y);
Serial.print(" Z Value: ");
z = z/100;
Serial.print(z);
```

version qui s'est avérée convenir pour piloter douze LED. Le **listage 1** montre le fragment de code servant à corriger le décalage du capteur. Le programme Arduino complet, avec les modifications, peut être téléchargé sous [4]. Dans ce croquis, la fonction moniteur est particulièrement intéressante et invite aux expérimentations personnelles.

Essais sur le terrain

La boussole doit être installée dans un boîtier en plastique transparent et montée sur le guidon de la bicyclette. La disposition intérieure n'est pas critique : les différentes couches peuvent être séparées par des feuilles de plastique, toutes fixées à une plaque de base à l'aide de ruban adhésif double face. L'anneau des LED NeoPixel est fixé sur une autre plaque, avec une découpe circulaire intérieure et des dentelures. Les mesures ont été faites avec la LED 0 pointant vers le nord, mais lors des premiers tests avec le logiciel, c'est la direction du sud que l'anneau a indiquée. Pour résoudre ce problème, plutôt que de modifier le logiciel, il était bien plus simple de tourner l'appareil d'un demi-tour pour faire pointer la LED 6 vers l'avant. Maintenant, lorsque la bicyclette est orientée vers le nord, la LED « droit devant » s'allume ; tournez à droite, et la LED suivante à gauche s'allume, indiquant à nouveau la direction du nord, comme il se doit. Malheureusement, il s'est avéré que l'acier du guidon, avec son magnétisme résiduel, perturbait les lectures. Cependant, j'ai découvert à cette occasion que le cadre de mon vélo est fait en aluminium dont, contrairement à l'acier, les propriétés

LA BOUSSOLE SUR UNE BICYCLETTE ÉLECTRIQUE

J'ai réalisé l'alimentation du circuit de la manière suivante. Cachée sous le guidon, j'ai monté une prise ordinaire du type de celles que l'on trouve dans les voitures, reliée par deux mini-fiches (rouge et bleue) à la prise de charge de la batterie de la bicyclette électrique. Il est essentiel de bien isoler ces connexions, car malgré sa faible tension (généralement 36 V), la batterie contient assez d'énergie pour faire pas mal de dégâts. En cas de court-circuit, un câble inadapté brûlera comme un fusible ! L'alimentation de 5 V du circuit est produite par un convertisseur abaisseur USB avec une entrée de 24 V, du monde des camions. Comme les batteries des camions sont souvent chargées à une tension nettement supérieure à leur tension nominale, ces appareils utilisent généralement un convertisseur à découpage MC34063A, qui peut supporter une tension d'entrée allant jusqu'à 40 V. Le convertisseur USB a été démonté pour souder directement sur la sortie 5 V un câble bifilaire avec des fiches, la consommation électrique de la boussole étant assez faible.

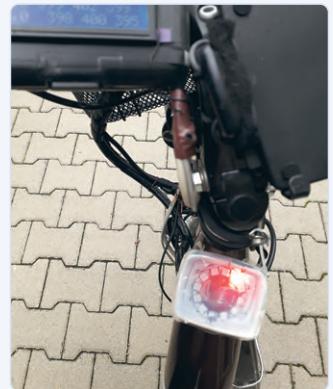


Figure 5 : La boussole assemblée et fixée au cadre de mon vélo.

paramagnétiques ne perturbent pas la boussole. Comme l'appareil est petit, il a été possible de le fixer sur le cadre tout en le gardant dans mon champ de vision (**fig. 5**). Les tests sur le terrain ont montré que l'appareil fonctionnait de manière fiable, le seul problème étant survenu près d'un pont de chemin de fer, probablement à cause du champ dû aux caténaires. Désormais, grâce à cette boussole, je peux m'orienter beaucoup plus facilement en territoire inconnu !

(200597-04)

Des questions, des commentaires ?
Contactez Elektor (redaction@elektor.fr).

Contributeurs

Idée et texte : **Rolf Hase**
Rédaction : **Rolf Gerstendorf**
Traduction : **Helmut Müller**
Mise en page : **Giel Dols**



PRODUITS

- **Coffret Arduino d'Elektor**
www.elektor.fr/elektor-arduino-electronics-bundle
- **Kit de capteurs Arduino**
www.elektor.fr/arduino-sensor-kit



LIENS

- [1] **Module GY-271** : www.az-delivery.de/en/products/gy-271-kompassmodul-kompass-magnet-sensor-fuer-arduino-und-raspberry-pi
- [2] **Guide de démarrage rapide** : GY-271 Kompassmodul Sensor (livre électronique gratuit) :
www.az-delivery.de/products/gy-271-kompassmodul-kompass-magnet-sensor-fuer-arduino-und-raspberry-pi
- [3] **Bibliothèque NeoPixel** : https://github.com/adafruit/Adafruit_NeoPixel
- [4] **Page du projet** : www.elektormagazine.fr/200597-04

Électronique et émissions de gaz à effet de serre

mesurez votre empreinte



Priscilla Haring-Kuipers (Pays-Bas)

Fût-il concepteur, fabricant ou *maker*, chaque acteur du secteur de l'électronique devrait aussi se préoccuper d'éthique. Qu'en est-il par exemple de l'empreinte carbone de nos réalisations, c'est-à-dire de notre contribution aux émissions de gaz à effet de serre ? Des outils et un soupçon de pragmatisme permettent de la calculer.

Lorsque vous avez l'audace d'ajouter au monde quelque chose qui n'existant pas auparavant, il y a forcément des conséquences. L'une d'elles est l'émission de gaz à effet de serre. L'empreinte carbone de nos trajets et de nos véhicules est calculable, mais qu'en est-il d'une activité liée à l'électronique ? Aidée de l'outil *Idemat* et de quelques hypothèses de départ, j'ai pu calculer l'empreinte carbone de ma petite entreprise de fabrication de synthétiseurs.

Empreinte carbone

Le terme « empreinte carbone » évoque l'idée d'une pression exercée par l'homme sur son environnement. Exprimée en équivalent CO₂, elle représente plus précisément la quantité des émissions de gaz à effet de serre d'origine anthropique [1]. On trouve en ligne de nombreux calculateurs conçus pour déterminer l'empreinte carbone de notre consommation d'énergie domestique ou professionnelle. On peut même calculer celle des vacances que le

virus nous a empêchés de prendre. Ce bilan carbone personnel nous renseigne sur nos choix de consommateurs, et peut aussi nous décider à « compenser » notre empreinte, c'est-à-dire à investir dans un projet de séquestration ou de réduction du CO₂. Les organisations à l'œuvre derrière ces projets fournissent généralement un autre outil en ligne déterminant le coût qu'aurait cette « compensation carbone ». Je tiens à souligner que ce mécanisme financier (calculez et payez) ne saurait se substituer aux changements politiques et industriels indispensables. Il s'agit toutefois d'une démarche volontaire que les particuliers et les acteurs économiques peuvent adopter spontanément.

Je dirige une petite entreprise de fabrication de synthétiseurs, et j'aimerais au moins compenser l'empreinte carbone des produits que nous vendons. Chiffrer cette empreinte n'est cependant pas chose aisée. Celle de l'industrie des puces est connue (et elle est énorme), mais pas celle liée à



L'auteure

Priscilla Haring-Kuipers est titulaire d'un master en psychologie des médias et membre active du site *This Is Not Rocket Science*. Convaincue des bienfaits de la recherche et de la technologie sur le devenir de l'humanité, elle apporte à ses articles et arguments l'éclairage des sciences sociales.

une puce donnée [2]. Nous savons que la production d'un iPhone « coûte » entre 65 et 95 kg de CO₂, tandis que celle d'un Fairphone 3 approche 40 kg de CO₂, mais ce que nous fabriquons ne ressemble en rien à un smartphone [3]. Il existe bien des outils d'analyse du cycle de vie (ACV) d'un produit, mais ceux-ci exigent un inventaire très fouillé de la chaîne de fabrication, ce qui prend du temps. Ces outils semblent de plus avoir été conçus pour les grandes entreprises, pas pour les PME comme la nôtre [4].

L'empreinte du Fenix

Après plusieurs recherches sur l'internet, je suis tombée sur *Idemat*, à mes yeux l'outil le plus pratique pour calculer une empreinte carbone en fonction de la nature d'un projet [5]. *Idemat* propose trois scénarios possibles de recyclage pour un produit en fin de vie. Nous avons choisi le scénario « Recyclage en circuit fermé » pour notre synthétiseur, car nos produits ont été conçus pour être réparables. Un instrument de musique électronique se doit d'être utilisable pendant des décennies, et doit pouvoir être rénové ou réparé s'il s'avère défectueux. Presque tous les



Je dirige une petite entreprise de fabrication de synthétiseurs, et j'aimerais au moins compenser l'empreinte carbone des produits que nous vendons.

vendeurs de synthétiseurs disposent d'un service de réparation, et la plupart des fabricants soutiennent cette démarche en fournissant des guides de réparation et d'entretien accompagnés de schémas. Nous le faisons aussi.

L'instrument dont nous avons calculé l'empreinte carbone est notre Fenix IV, un synthétiseur modulaire d'exception que nous produisons à cent exemplaires. Idemat commence par calculer l'empreinte des matériaux utilisés. Pour nous il s'agissait notamment d'aluminium (boîtier et panneau de commande), de plastique ABS (boutons), de carton et de papier à bulles en LLPDE (emballage). La catégorie « PCB » est décevante puisqu'elle repose sur les cartes assemblées typiques d'un ordinateur portable. Nous l'avons utilisée pour certains de nos circuits, mais avons estimé nous-mêmes l'empreinte d'une carte non assemblée en nous basant sur ses quantités relatives de cuivre et de FR-4 (1 kg de FR-4 coûte 2 kg de CO₂), et en ajoutant le coût, lui aussi estimé, de ses jacks et potentiomètres. Idemat a ensuite pris en compte nos processus de fabrication, p. ex. le moulage par injection des boutons et l'usinage du boîtier. Nous avons terminé l'analyse par les coûts en carbone du transport aérien et ferroviaire intercontinental lié à notre logistique. La liste est longue [6]. La méthode n'est pas parfaite, et loin d'être aussi détaillée qu'une analyse du cycle de vie standard, mais je pense que nous sommes parvenus à un résultat proche de la réalité. Un autre outil nous a servi à calculer l'empreinte carbone de nos déplacements :

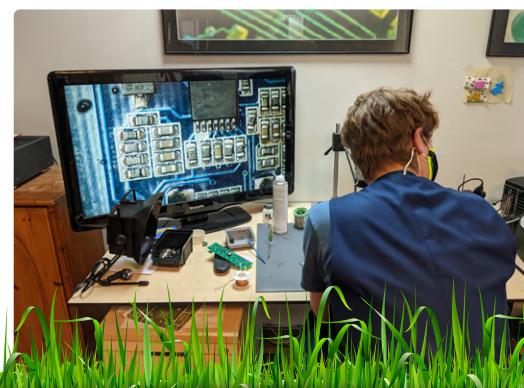
un voyage d'affaires à Shenzhen (5450 kg de CO₂) et un trajet en voiture de 2680 km pour visiter nos usines aux Pays-Bas (290 kg de CO₂). Nos déplacements personnels ont de loin la plus grande empreinte. Sans doute parce que l'humain est encore mal adapté aux dimensions de son environnement. Le bilan final est de 316 kg de CO₂ par Fenix IV. La contribution de l'aluminium est étonnamment faible (29 kg de CO₂) au regard de sa teneur dans le boîtier. Celle du

nickel semble à l'inverse élevée (3 kg de CO₂) quand on sait qu'il n'intervient que dans les jacks et les cordons de raccordement. Le bambou qui entoure notre boîtier a une contribution très faible, seulement 150 g de CO₂. Le chiffre le plus élevé vient des circuits imprimés (208 kg de CO₂). Nous avons choisi notre organisation préférée, TreesForAll [7], pour compenser les émissions de carbone de notre chaîne de production.

Et vous, quelle est votre empreinte ?

Et vous, qu'en est-il de votre projet ? En trierez-vous les composants pour les apporter à un point de collecte ou une déchetterie ? Le réparerez-vous ? Le recyclerez-vous ? J'espère que vous tentez de calculer son empreinte carbone. Si vous le faites, n'hésitez pas à partager vos chiffres ici, nous en avons vraiment besoin pour estimer l'empreinte carbone de nos créations électroniques. Et je serais curieuse de les connaître !

(210340-04 - VF : Hervé Moreau)



LIENS

- [1] Wikipédia, empreinte carbone :
https://fr.wikipedia.org/wiki/Empreinte_carbone
- [2] A. Crawford, I. King, and D. Wu, *The Chip Industry Has a Problem With Its Giant Carbon Footprint*, bloomberg.com, 8 avril 2021 :
<http://bit.ly/bloom-carbon-foot>
- [3] M. Proske, et al, *Life Cycle Assessment of the Fairphone 3*, Fraunhofer IZM, juillet 2020 : <http://bit.ly/fairphone3-lca>
- [4] Plateforme européenne d'analyse du cycle de vie :
<https://eplca.jrc.ec.europa.eu/LCDN/>
- [5] Idemat : <http://idematapp.com/>
- [6] ACV du Fenix IV : <https://bit.ly/3h0W8y0>
- [7] TreesForAll : <https://treesforall.nl/en/>

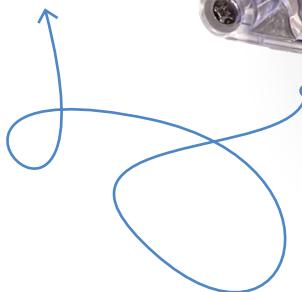


Les électroniciens professionnels et les *makers* doivent-ils suivre une ligne éthique ? L'éthique doit-elle jouer un rôle dans la façon dont les entreprises d'électronique fabriquent et vendent leurs produits ? Elektor et ses partenaires lancent le débat et vous invitent à participer à cette réflexion commune. Nous organiserons en novembre prochain le premier *Forum mondial de l'électronique éthique* (WEEF). Suivez nos questions, sondages, études et autres actualisations sur elektormagazine.fr.



thermostat connecté à ESP32

Conservez votre vin à la bonne température !



Yves Bourdon (France)

Pour un bon vieillissement à long terme, le vin doit être conservé dans un endroit sombre et frais. Même si les spécialistes ne sont pas d'accord sur la meilleure température pour cela, ils recommandent de maintenir une valeur de température aussi constante que possible. Le thermostat présenté ici tente non seulement de le faire, mais il envoie également des alertes par courrier électronique lorsque la température devient trop élevée.

L'idée de ce projet m'est venue alors que je construisais l'excellent thermostat Wi-Fi d'Elektor [1], conçu autour de l'ESP8266 d'Espressif. J'ai essayé de porter le logiciel de ce projet sur un ESP32, mais j'ai abandonné à cause du travail que cela impliquait. J'ai donc décidé de repartir de zéro en m'inspirant de ce qui avait déjà été fait (interface web,

horloge NTP, régulation de température avec hystérésis programmable, etc.) et d'améliorer l'interface web en corrigéant ses défauts (en particulier, le fait que l'interface utilisateur ne soit pas rafraîchie en temps réel). J'ai également ajouté un petit écran OLED très pratique, qui permet d'utiliser ce thermostat même si la connexion Wi-Fi est interrompue (mode dégradé).

Ce projet est facilement adaptable à d'autres applications. J'ai par ex. utilisé le même matériel avec un détecteur d'humidité DHT11 (ou BME280) pour commander le système de ventilation mécanique de ma salle de bains. L'entrée opto-isolée détecte si la lumière est allumée (spots de 12 V) afin de pouvoir éteindre temporairement l'extracteur d'air bruyant lorsqu'on occupe la salle de bains.

Le circuit

La **figure 1** donne le schéma du thermostat à ESP32. Pour optimiser le coût, la fiabilité et la taille, j'ai décidé d'utiliser le module WiFi Kit 32 de Heltec Automation (**fig. 2**) au cœur du thermostat. On trouve facilement sur l'internet ce module basé sur l'ESP32, doté d'un bel écran OLED bleu de 2,4 cm et 128 × 64 pixels ainsi que d'une LED blanche programmable par l'utilisateur. Il se branche sur K5 et K6.

Notez qu'il est possible de remplacer le module WiFi Kit 32 par un module ESP32



Spécifications

- Sonde de température DS18B20, précision de $\pm 0,5$ °C, affichage à 0,1 °C près.
- Relais avec contacts amortis permettant de commuter des charges CA jusqu'à 2.200 W.
- Alimentation 100-240 VCA.
- Bus d'extension I2C compatible avec modules Grove et Qwiic.
- Le circuit imprimé est compatible avec le WiFi Kit 32 de Heltec (avec écran OLED intégré) ou le DevKit ESP32 avec écran OLED externe.
- Buzzer pour alarme sonore.
- Entrée pour interrupteur externe.
- Entrée isolée optiquement.
- Trois seuils programmables : minimum, maximum et alarme.
- L'alarme peut envoyer un courriel.
- Maintien de l'heure basé sur NTP.
- Minuterie hebdomadaire.
- Peut commander un refroidisseur ou un réchauffeur.



DevKitC (connecteurs K7 et K8) avec un écran séparé OLED I₂C de 2,4 cm et 128 × 64 pixels, tous deux assez courants (**fig. 3**). L'écran OLED existe en deux versions de brochage (avec VCC et GND inversés) et donc deux connecteurs sont prévus pour lui (K9 et K10). Le logiciel est identique pour les deux modules.

Alimentation électrique

Le bloc d'alimentation est un petit convertisseur CA/CC de 3 W. Il accepte une tension d'entrée comprise entre 100 et 240 VCC pour une sortie en 5 V jusqu'à 600 mA. La LED3 indique si l'appareil est sous tension. Un fusible protège l'appareil et le secteur, tandis que la varistance VAR1 assure une protection contre les surtensions (très appréciée par les normes de conformité). Le raccordement au secteur s'effectue via le bornier K1.

Le module ESP32 est alimenté à partir du 5 V, car il dispose de son propre régulateur de 3,3 V. Le module peut également être alimenté directement via le bus micro-USB.

Relais

La charge du thermostat est commutée par le relais RE1, qui, à son tour, est commandé par un petit transistor NPN (T1) connecté au port IO18. La LED2 indique l'état du relais. La diode D1 protège l'électronique des courants de commutation induits par le relais, tandis que le réseau de filtrage C4/R9 ou C4/R10 (ne mettre qu'une résistance) protège les contacts du relais en amortissant toute étincelle de commutation en cas de forte charge.

Le relais utilisé peut piloter des charges de 250 VCC jusqu'à 10 A. Les contacts normalement ouverts (NO) et normalement fermés (NC) sont tous deux disponibles sur K2. Notez que K1 et K2 peuvent être combinés en un seul bornier à cinq voies.

Buzzer

Un buzzer avec oscillateur intégré (Buz1), piloté par T2 via le port IO19, est disponible pour un retour auditif. Comme il dispose d'un oscillateur intégré, il suffit de l'alimenter pour qu'il émette un bip. On peut aussi utiliser un buzzer à courant alternatif, mais l'ESP32 devra alors produire lui-même le son.

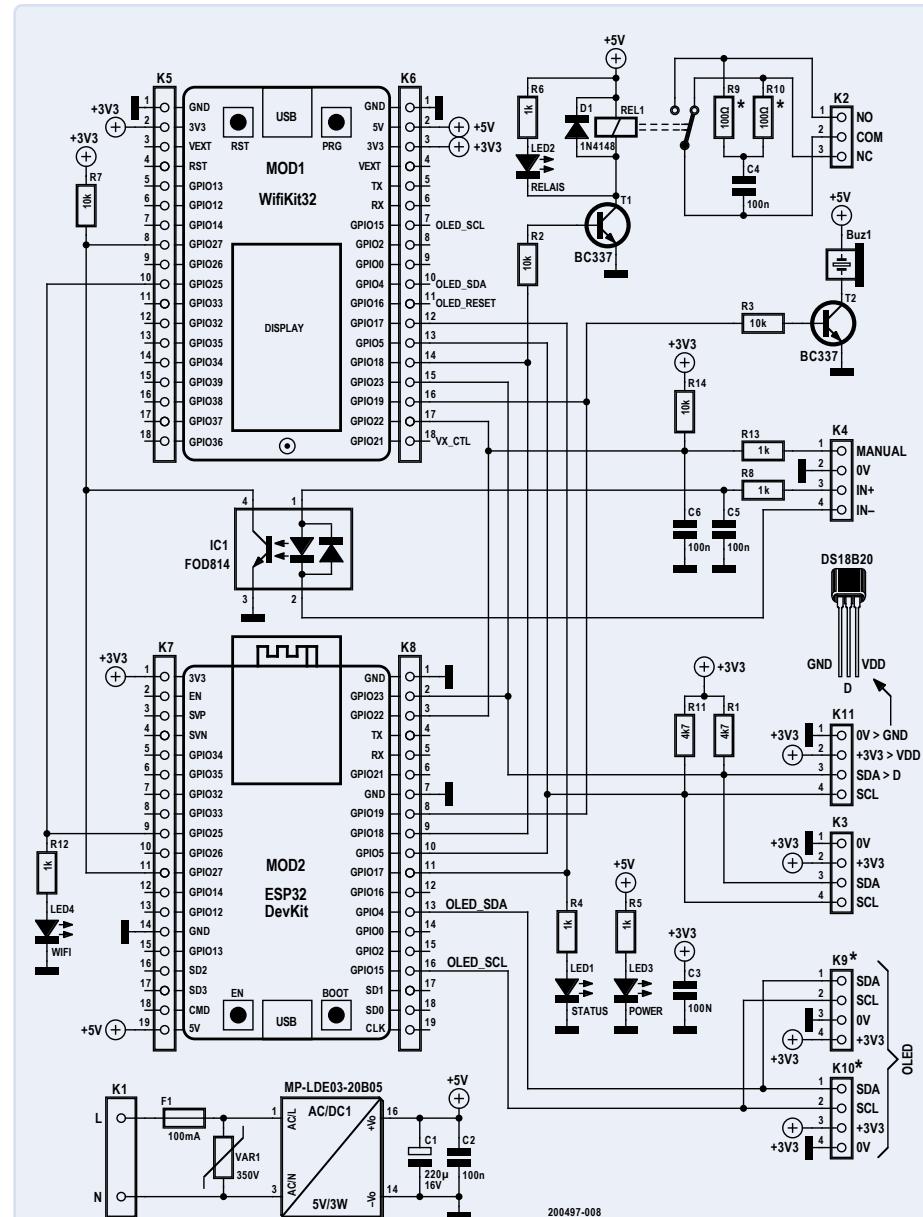


Figure 1. On trouve de nombreux connecteurs sur le schéma du thermostat à ESP32, car il y a plusieurs options de configuration.



Figure 2. Le WiFi Kit 32 de Heltec Automation.



Figure 3. Le thermostat construit par le labo d'Elektor utilise un module DevKitC ESP32 avec un module d'affichage OLED séparé.



Figure 4. Le thermostat reprogrammé en contrôleur d'humidité, ici équipé d'un module Heltec WiFi Kit 32 avec écran OLED intégré.



Figure 5. La page principale affiche la température mesurée ainsi que les seuils de commutation et d'alarme.

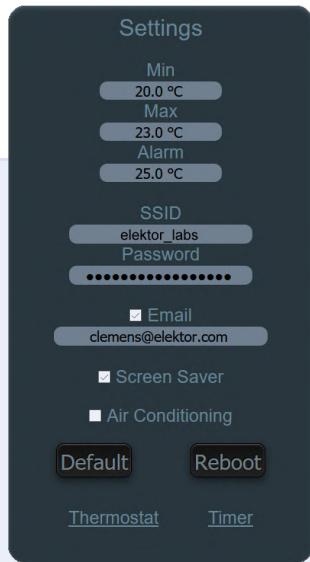


Figure 6. Les températures minimale et maximale ainsi que le seuil d'alarme sont définis ici. C'est également sur cette page que vous définissez le SSID et le mot de passe du réseau Wi-Fi que le thermostat doit utiliser.

LED d'état

La LED1, connectée au port IO17, est un indicateur d'état. Un clignotement rapide signale une erreur due à des données EEPROM mauvaises ou invalides, ou encore à des problèmes de communication du capteur DS18B20. Ces erreurs sont bloquantes, et le programme s'arrête jusqu'à ce que le problème soit résolu. Des clignotements courts en mode de fonctionnement normal indiquent que l'on accède au serveur web intégré.

Contrôles externes

Le bornier K4 permet la connexion d'un interrupteur externe (*manuel*) entre le port IO22 et GND. Avec le logiciel par défaut, cet interrupteur sera prioritaire sur le thermostat et activera le relais. Un réseau RC de filtrage/antirebonds (1,5 kHz avec les valeurs de composants données) peut être mis sur cette entrée. Lorsqu'on utilise un interrupteur bistable, on peut omettre C6 et remplacer R13 par un bout de fil.

Une entrée numérique IN isolée optiquement (optocoupleur IC1) connectée au port IO2 est disponible pour une extension future et n'est pas exploitée par le logiciel actuel. Elle me sert pour détecter l'allumage de l'éclairage de la salle de bains mentionné précédemment (fig. 4).

Connexion du capteur et port I²C

Le capteur de température, un DS18B20 de type One-Wire, est destiné à être

connecté au bornier K3. Il faut relier sa broche 1 à GND, sa broche DQ (broche 2) à la broche 3 de K3 et enfin sa broche VDD au 3,3 V. Cela signifie que ses pattes 2 et 3 se croisent. Néanmoins, pour des applications comme un thermostat, il ne faut pas mettre le capteur de température à l'intérieur du boîtier, car l'alimentation et le module ESP32 y chauffent l'air, ce qui fausse la régulation de température.

La résistance R1 est nécessaire pour le bon fonctionnement du DS18B20. Si on remplace ce capteur de température par quelque chose comme un capteur d'humidité (tel qu'un BME280, SHT11 ou DHT22), on peut ne pas avoir besoin de R1.

En réalité, K3 est câblé de manière à être compatible avec les standards de facto des fabricants de composants à bus I²C comme Grove (Seeed Studio) et Qwiic (SparkFun). De fait, K11 est un connecteur Grove (au pas de 2 mm). De nombreuses cartes d'extension

compatibles avec ce format de connecteur sont disponibles, ce qui fait du circuit du thermostat une excellente base pour d'autres applications.

Circuit imprimé

Elektor a mis au point un circuit imprimé qui s'insère dans un boîtier bon marché doté d'un couvercle transparent. Les pistes du relais qui se connectent à la charge peuvent et doivent être renforcées avec de la soudure.

Logiciels et programmation

Le programme du thermostat est écrit en C/C++ dans l'EDI Arduino (avec la version 1.8.12). Il est assez gros – plus de 1.500 lignes de code – mais il est abondamment commenté (bien que dans un mélange d'anglais et de français). Il est assez facile de l'adapter à vos besoins et je n'ai utilisé que des bibliothèques *open source* gratuites et éprouvées.

Avant de pouvoir utiliser l'ESP32 avec l'EDI Arduino, vous devez d'abord installer le paquetage de cartes correspondant. De nombreuses pages web ont été écrites pour expliquer cette procédure, référez-vous à [2] pour plus de détails (avec vidéo). Le paquetage de cartes officiel ESP32 inclut la prise en charge du Heltec WiFi Kit 32 et du DevKitC (« ESP32 Dev Module »). Quel que soit le module choisi, vous pouvez laisser tous les autres paramètres (vitesse de l'UC, taille de la mémoire flash, etc.) à leurs valeurs par défaut. Veillez toutefois à sélectionner le bon port COM.



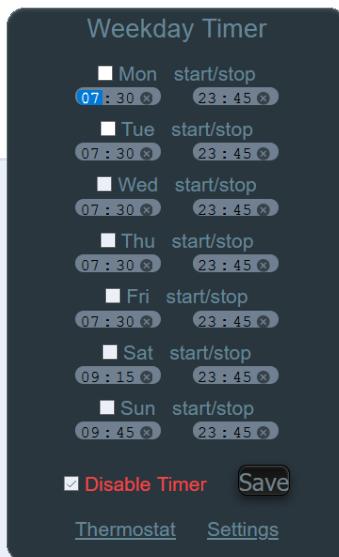


Figure 7. La page « Timer » vous permet de définir pour chaque jour de la semaine une période pendant laquelle le thermostat doit réguler la température.

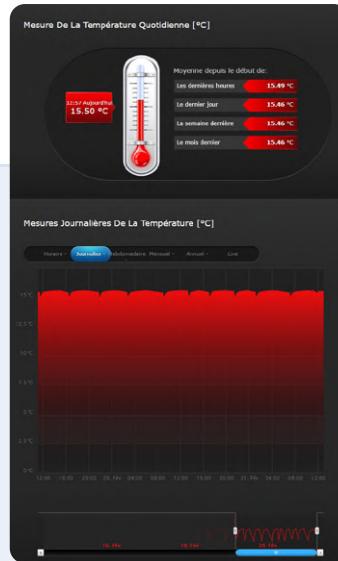


Figure 8. La régulation est plutôt bonne comme le montre cette capture d'écran de mon système de domotique. Il faut savoir qu'une variation de 1 °C dans le sous-sol correspond en fait à une variation de seulement 0,1 °C de la température du vin dans la bouteille.

Outre l'installation du paquetage de cartes ESP32, il est également nécessaire d'installer l'outil SPIFFS *ESP32 Sketch Data Upload*. Là encore, reportez-vous à [2] pour savoir comment procéder.

Téléchargez le croquis du thermostat depuis [3] et décompressez-le dans le carnet de croquis de l'EDI Arduino. Le fichier *Thermostat.ino*, les fichiers qui l'accompagnent et le dossier *data* doivent se trouver dans un dossier nommé *Thermostat*. Le sous-dossier *data* contient les fichiers HTML et CSS ainsi que les programmes JavaScript nécessaires au serveur Web. Le fichier *D7MR.woff2* contient la police à 7 segments que j'ai utilisée. Un grand « Merci ! » à son auteur.

Pour votre confort, j'ai inclus dans le téléchargement toutes les bibliothèques nécessaires au projet [3]. Elles doivent être installées correctement, c'est-à-dire en les copiant dans le sous-dossier *libraries* du carnet de croquis de votre EDI. L'emplacement de ce dossier sur votre ordinateur dépend de la façon dont vous avez configuré l'EDI Arduino. Reportez-vous à [4] pour plus de détails sur l'EDI Arduino.

Une fois que tout le logiciel est téléchargé et installé, vous devez régler certains paramètres par défaut du programme pour les adapter à votre environnement domestique ou votre application. Ils sont situés en haut du programme. Comme le thermostat utilise une adresse IP fixe, vous devez en définir une avec la passerelle et d'autres éléments typiques du réseau.

Configuration du courriel

Comme le thermostat peut envoyer des courriels, il est nécessaire d'entrer les caractéristiques du serveur de messagerie. Je vous recommande de créer un compte Gmail pour cela. Dans ce cas, le serveur de messagerie est *smtp.gmail.com* et le port à utiliser est 465. Pour des raisons de sécurité, vous devez configurer le compte Gmail. Si votre compte utilise une authentification à un facteur (c'est-à-dire qu'il ne nécessite qu'un mot de passe), configurez le compte en suivant le lien [5]. En cas d'authentification à deux facteurs (c.-à-d. un mot de passe et un code par SMS), référez-vous à [6].

Il est maintenant temps de compiler et de télécharger le croquis comme tout autre croquis Arduino. Vous ne devriez pas rencontrer de problèmes sérieux, voire aucun. Sinon, vérifiez votre installation.

Après avoir programmé le croquis, téléchargez les pages web et les autres fichiers de données avec la commande *ESP32 Sketch Data Upload* disponible dans le menu *Outils* de l'EDI. Cette commande permet de transférer tout ce qui se trouve dans le dossier *data* vers la mémoire flash (SPIFFS) de l'ESP32.

Premier démarrage

Lorsque vous démarrez l'appareil pour la première fois, connectez-vous à son propre point d'accès Wi-Fi (AP) avec le SSID « Thermostat ESP32 » ; il n'y a pas de mot de passe. Ouvrez un navigateur et allez à l'adresse IP 192.168.4.1. L'interface web



PRODUITS

- **ESP32 DevKitC**
www.elektor.fr/esp32-devkitc-32d
- **Écran OLED I²C, 128x64 pixels, de 2,3 cm**
www.elektor.fr/blue-0-96-oled-display-i2c-4-pin
- **Combiné Elektor : kit ESP32 & livre en anglais « The Complete ESP32 Projects Guide »**
www.elektor.fr/19033

devrait maintenant apparaître. Cliquez sur le lien *Settings*, puis sur le bouton *Default*. Cela permettra d'initialiser tous les paramètres – y compris les réglages de la minuterie et de la température – avec leurs valeurs par défaut et d'activer le SSID et le mot de passe afin que le thermostat puisse se connecter à votre réseau Wi-Fi. Redémarrez l'appareil pour que les paramètres soient actifs. Le bouton *Reboot* permet de réinitialiser le thermostat. Notez que lorsque le SSID et/ou le mot de passe sont modifiés, le système redémarre automatiquement.

Pour les paramètres de la minuterie et de la température, il n'est pas nécessaire de valider un champ après avoir modifié sa valeur. Il suffit de déplacer la souris hors du champ. De plus, il y a un contrôle de validité sur les seuils de température minimum et maximum. Si la valeur maximale est inférieure à la valeur minimale, les deux sont permutées.

Interface utilisateur chatoyante

Une grande attention a été portée à la conception de l'interface utilisateur. L'objectif était de créer une interface web soignée qui s'affiche correctement sur les navigateurs les plus courants et récents ainsi que sur les ordiphones iPhone et Android. L'interface est très réactive, sans latence notable, et les valeurs sont rafraîchies toutes les secondes. L'interface utilisateur basée sur le web comporte trois pages : *Main* (fig. 5), *Settings* (fig. 6) et *Timer* (fig. 7).

Sur la page principale, des codes de couleur

servent à indiquer l'état des paramètres. Par exemple, l'intensité du signal Wi-Fi (RSSI) est affichée en vert, orange ou rouge selon la qualité du signal. L'heure, obtenue à partir d'un serveur NTP (*Network Time Protocol*) situé quelque part sur l'internet, est affichée en rouge lorsque le minuteur commande le relais ; sinon, elle est en vert. De même, la température est affichée en bleu clair si la température est inférieure à la valeur minimale ; elle est en vert si la température est comprise entre le minimum et le maximum. Lorsque la température est supérieure à la valeur maximale, elle est en rouge. Lorsqu'elle commence à clignoter en orange, le système a rencontré un problème. Une petite LED est symbolisée sur la page principale pour montrer en temps réel l'état du relais (rouge s'il est activé, vert sinon). La feuille de style CSS a demandé beaucoup de travail pour rendre possible cet affichage original.

L'écran OLED affiche également l'état du système. Comme on a constaté que ces écrans ont tendance à se dégrader assez rapidement lorsqu'ils restent allumés en permanence, on a ajouté une option d'économiseur d'écran. En appuyant sur un bouton-poussoir connecté entre GPIO05 (K11, broche 4) et GND (K11, broche 1), l'écran s'allume pendant quelques minutes. Cette fonction peut être activée ou désactivée sur la page des paramètres.

Page de la minuterie

Les horaires hebdomadiers d'activation et de désactivation, ainsi que la case à cocher d'activation correspondante, doivent être enregistrés avec le bouton *Store*. En cliquant sur ce bouton, la page entière sera sauvegardée. La case à cocher *Timer Disabled* ne nécessite pas de validation avec le bouton *Store*. Lorsque la minuterie est désactivée, le thermostat régule la température en continu.

Mode débogage

En surveillant le port série via l'interface USB intégrée (115.200 bauds), on peut observer le processus d'initialisation du thermostat après sa mise sous tension. Très utile pour identifier des choses telles que des problèmes de connexion Wi-Fi.

Un système fiable

Le thermostat décrit dans cet article régule la température de mon sous-sol depuis

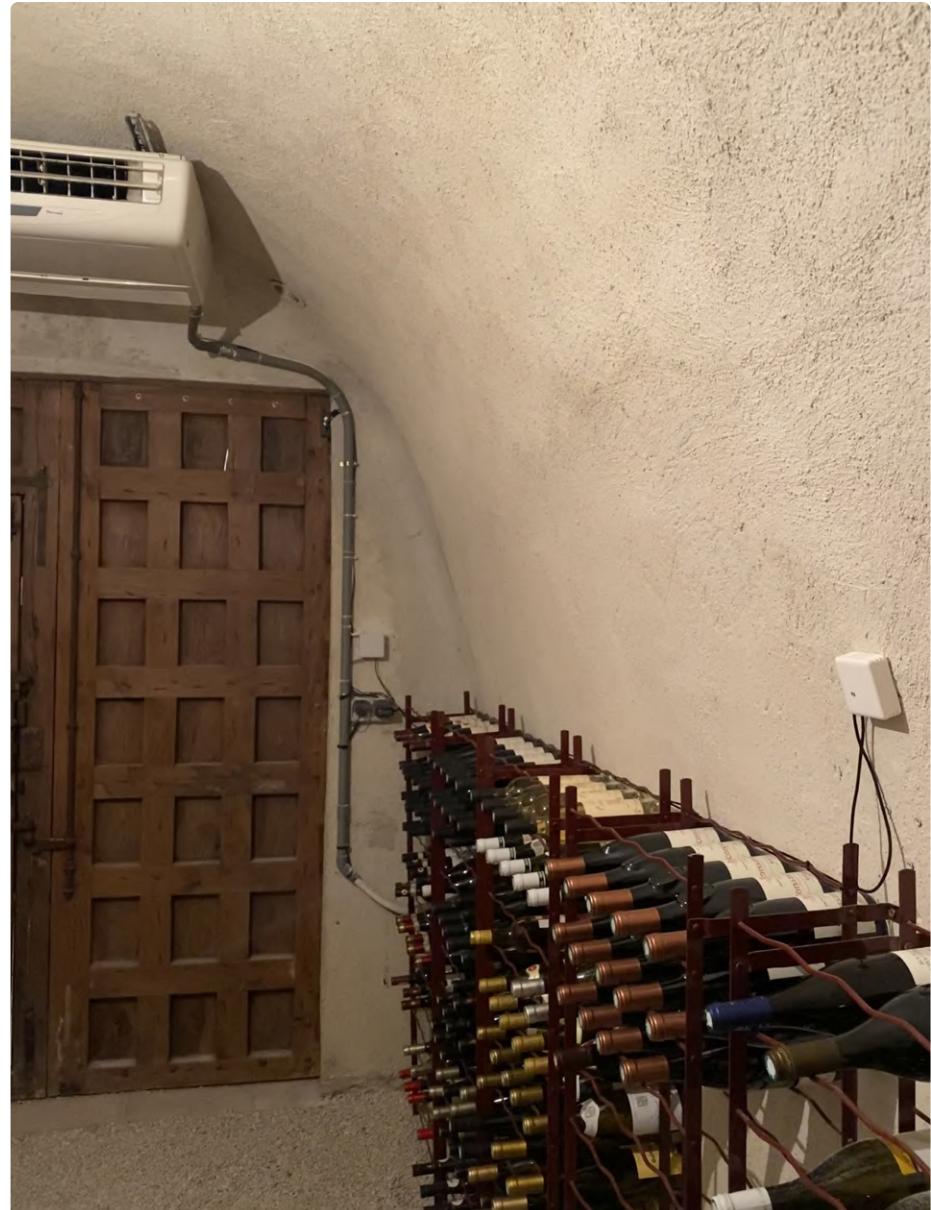


Figure 9. Le capteur DS18B20 a été monté dans une petite boîte en plastique séparée, fixée à un endroit stratégique de la cave. Un deuxième capteur, identique, transmet la température à mon système de domotique via un module spécifique. Cela ajoute une redondance de sécurité au cas où le thermostat tomberait en panne. Si cela se produit, mon système de domotique me prévient.

plus d'un an maintenant et je n'ai constaté aucun problème majeur. Le système est fiable et la régulation est très bonne (fig. 8). J'ai pu tester l'interface web sur la majorité des navigateurs internet (Chrome, Edge, Safari, etc.) ainsi que sur iPhone et certains téléphones Android et je n'ai observé aucun problème de compatibilité.

Je tiens à remercier Stéphane Calderoni (docteur en informatique) qui vit à La Réunion (une île d'Afrique de l'Est) qui m'a beaucoup aidé dans le développement de l'interface HTML et du JavaScript qui va avec. J'ai profité de la période d'enfermement de mars 2020 en France pour améliorer mes connaissances sur cette partie que je ne maîtrisais pas du tout. 

(200497-04)

Contributeurs

Idée, conception et texte : Yves Bourdon

Conception des circuits imprimés et

rédaction : Clemens Valens

Mise en page : Harmen Heida

Traduction : Denis Lafourcade

Des questions, des commentaires ?

Envoyez un courriel au rédacteur

(clemens.valens@elektor.com) ou

contactez Elektor (redaction@elektor.fr).

LISTE DES COMPOSANTS

Résistances (5%, 0,25 W)

R9, R10 = 100 Ω , 1 W (voir texte)

R4, R5, R6, R8, R12, R13 = 1 k Ω

R1, R11 = 4,7 k Ω

R2, R3, R7, R14 = 10 k Ω

Condensateurs

C2, C3, C5, C6 = 100 nF, pas de 2,5 ou 5 mm

C4 = 100 nF, X2, pas de 15 mm

C1 = 220 μ F 16 V, pas de 3,5 mm

Semi-conducteurs

D1 = 1N4148

LED1, LED2 = LED, rouge, 3 mm

LED3, LED4 = LED, verte, 3 mm

IC1 = FOD814, DIP4

T1, T2 = BC337

Divers

BUZ1 = buzzer avec oscillateur, pas de 6,5

ou 7,6 mm

F1 = porte-fusible, 5x20 mm + fusible 250 V, 100 mA, lent

K1 = bornier à 2 voies, pas de 5 mm

K2 = bornier à 3 voies, pas de 5 mm

K3, K4 = bornier à 4 voies, pas de 3,5 mm

K5, K6 = barrette femelle à 18 positions, pas de 2,54 mm (uniquement MOD1)

K7, K8 = barrette femelle à 19 positions, pas de 2,54 mm (uniquement MOD2)

K9, K10 = barrette femelle à 4 positions, pas de 2,54 mm, hauteur de 20 mm (uniquement MOD2)

K11 = barrette mâle à 4 broches, pas de 2 mm (Grove)

REL1 = relais, 10 A, SPDT, 5 VDC

VAR1 = varistance 350 V

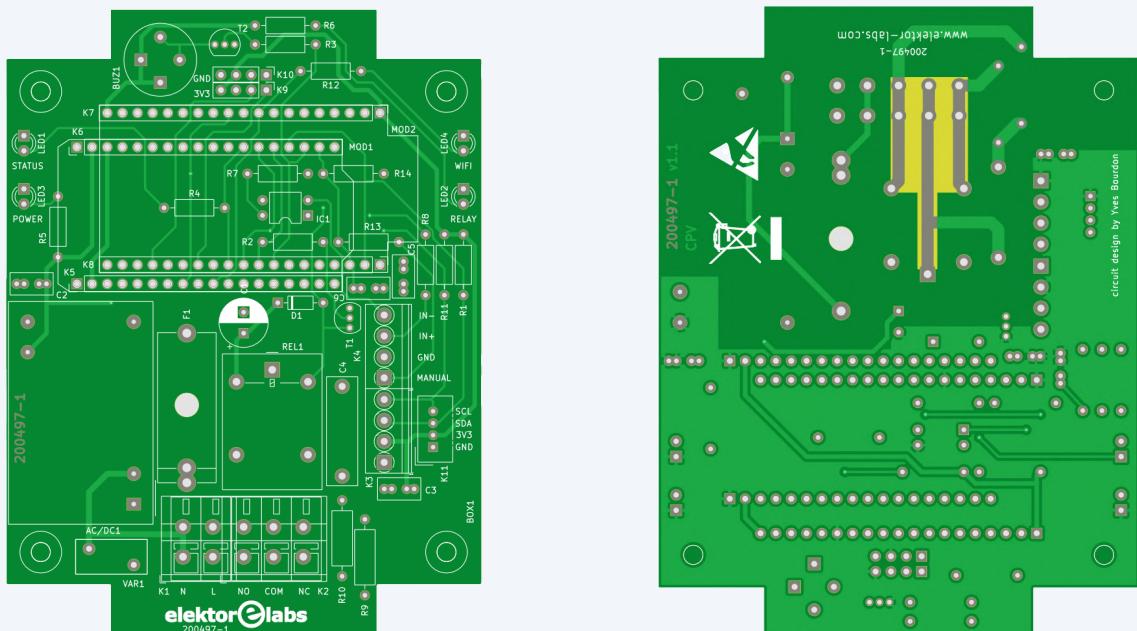
AC/DC1 = 5 V, 3 W, convertisseur CA-CC (par ex. Multicomp MP-LDE03-20B05)

BOX1 = boîtier, IP65, couvercle transparent (Multicomp MC001106)

MOD1 = WiFi Kit 32 de Heltec

ou

MOD2 = ESP32 DevKitC + écran OLED, I²C, 128x64 pixels



LIENS

- [1] R. Aarts & C. Valens, « thermostat de bureau wifi », Elektor, 01-02/2018 : www.elektormagazine.fr/160269
- [2] Vidéo « ESP32 – Getting started » : www.elektormagazine.fr/labs/1874
- [3] Thermostat ESP32, Elektor Labs : www.elektormagazine.fr/labs/4216
- [4] FAQ Arduino : www.elektormagazine.fr/labs/1876
- [5] Authentification à un facteur de Gmail : <https://myaccount.google.com/lesssecureapps>
- [6] Authentification à deux facteurs de Gmail : <https://security.google.com/settings/security/apppasswords>



lévitation magnétique, version numérique

comparateur analogique remplacé par un ESP32 Pico

Peter Neufeld (Allemagne) et
Luc Lemmens (Elektor)

Dans l'article « lévitation magnétique sans peine », nous avons vu que la lévitation magnétique n'est pas trop difficile à réaliser avec un aimant permanent, la bobine et le noyau d'un relais 5 V standard, un capteur à effet Hall et un bon vieux comparateur analogique. Dans ce deuxième projet, c'est un SoC ESP32 Pico qui est aux commandes, intégré dans le très pratique module ATOM de M5Stack.

Le titre du premier article, « Lévitation magnétique sans peine » [1][2], pouvait laisser penser que nous allions cette fois-ci envisager une méthode plus complexe pour réaliser la lévitation, mais ce n'est pas le cas. Dans ce 2^e volet du projet de lévitation [3], un petit module basé sur l'ESP32 (ATOM Lite ou Matrix de M5Stack, voir l'encadré « ATOM... ») reprend le rôle du comparateur analogique

du 1^{er} projet : il mesure la tension en sortie du capteur Hall et commande la bobine. En fait, pour construire cette version numérique, si on ne compte pas les composants internes du module M5Stack, il faut bien moins de pièces qu'avec le circuit analogique. C'est amusant : pour certains de nos lecteurs, l'utilisation d'un microcontrôleur est la solution de facilité, tandis que d'autres demanderont pourquoi utiliser un module ESP32 alors qu'un simple LM311 fait l'affaire. Alors pour satisfaire tout le monde, nous avons décidé de présenter les deux options.

Pour ce projet, tout module ESP32 conviendrait à la place de l'ATOM de M5Stack. Pour l'une des versions du prototype (voir ci-dessous), le bouton-poussoir (les modules ATOM en ont un) de réglage de la boucle d'asservissement de la lévitation sera très pratique.

Nous avons préféré les modules de M5Stack pour leurs qualités : bas prix, très faible encombrement et boîtier plastique protégeant bien contre les courts-circuits. Toutes les E/S nécessaires à ce projet sont facilement accessibles sur les prises SIL situées au bas des ATOM. Les LED Neopixel du modèle M5Stack ATOM Matrix ne sont utilisées qu'accessoirement et ne contribuent pas au processus de lévitation. Pour ce projet, le modèle Lite, à une seule LED Neopixel, convient aussi.

Au début de l'écriture de cet article, j'en discutais avec Peter qui a eu l'idée de remplacer le bouton-poussoir de réglage de la boucle d'asservissement par un trimmer ; il a finalement construit et testé cette solution également. Cette



« version supplémentaire » est traitée sur ses pages web [4] et dans cet article.

Le matériel, version originale

Comme le montre le schéma de la **figure 1**, le matériel utilisé est à peu près le même que dans la version analogique. La modification de la carte à relais est quasi identique, la différence la plus importante est que le capteur à effet Hall est désormais alimenté en 3,3 V, à partir du régulateur de tension interne du module ATOM. La **figure 2** montre l'electroaimant résultant avec le capteur Hall monté sur le noyau. Ici, la diode de roue libre (*flyback*) du relais est remplacée par deux LED blanches qui sont utilisées comme « spots » pour éclairer l'objet en lévitation. Comme dans la solution analogique, une seule LED conviendrait. Cependant, il semble que la stabilité de l'objet en lévitation soit meilleure dans la configuration à deux LED. Il est important de maintenir un courant de retour faible dans les LED, car un courant plus élevé maintient le champ électromagnétique trop longtemps, même lorsque le transistor est coupé [5]. Ce champ supplémentaire nuit à la boucle d'asservissement et déstabilise le vol stationnaire de la charge. Les LED et la résistance sont cependant nécessaires à la place de la diode d'origine, car elles limitent la tension d'induction appliquée au transistor du relais à une valeur inoffensive (< 50 V) et garantissent des impulsions de commutation assez courtes. Deux LED blanches en série avec une résistance de 220 à 330 Ω font très bien le travail.

À propos de capteurs à effet Hall...

Au début, sur les pages web de Peter pour les deux versions du projet de lévitation, il était mentionné que seuls les A1302 ou A1308 d'*Allegro Microsystems* convenaient, et que le SS49E (*Honeywell*), plus abordable, ne donnait pas satisfaction. La raison en est restée un peu obscure, jusqu'à ce qu'il se rende compte que pendant le développement du tout 1^{er} prototype, à un moment donné, il avait changé plus d'une pièce à la fois (dont le SS49E remplacé par un A1302) et ensuite, le montage avait fonctionné. On soupçonnait que le capteur Hall était le coupable, mais des essais plus récents prouvent que le SS49E fonctionne dans les deux versions du projet. Bonne nouvelle, à cause du prix, mais aussi parce que les capteurs Honeywell sont beaucoup plus faciles à trouver.

Bien entendu, la tension d'alimentation de 3,3 V permet de maintenir la sortie du capteur entre 0 V et 3,3 V, sans excéder la tension d'entrée max. du CA/N du M5Stack ATOM. Nota : cette tension n'est pas conforme aux spécifications de la plupart des capteurs Hall (voir l'encadré « À propos de... »). Selon les fiches techniques, la tension d'alimentation minimale des A1302/A1308 devrait être de 4,5 V, mais ils semblent fonctionner à une

tension inférieure (en tout cas pour ce projet) et cela évite un diviseur de tension pour rester dans les limites d'entrée du CA/N. Cependant, il y a un inconvénient : apparemment sous 3,3 V, la tension de sortie du capteur commence à plafonner à une intensité de champ magnétique plus faible, et cela limite le poids de la charge qui peut léviter. Et pour le SS49E c'est aussi un facteur limitant, bien que ce capteur Hall soit spécifié pour une tension d'alimentation minimale de 2,6 V.

Le circuit est alimenté en 5 V soit par le connecteur USB-C, soit par son connecteur Grove ou encore par la prise SIL située au-dessous du M5Stack ATOM.

Le logiciel

Le croquis Arduino pour piloter la lévitation est très simple : le CA/N de l'ESP32 mesure la tension de sortie du capteur Hall, la compare à un seuil de déclenchement (avec une certaine hystérésis) et active ou coupe l'alimentation de la bobine, le logiciel imite donc le comparateur LM311 de la 1^{re} réalisation. Dans le 1^{er} exemple sur la page web de Peter (*MagLev_1.ino*), le seuil de déclenchement et l'hystérésis sont définis comme constantes dans le croquis et doivent être ajustés à la charge utilisée.

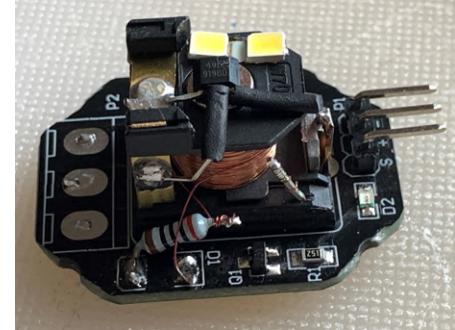


Figure 2. Carte relais modifiée et capteur à effet Hall sur le noyau de l'électroaimant.

Trouver le seuil de déclenchement correct peut prendre un certain temps, car il dépend de la charge et bien sûr de la taille et de la force de l'aimant permanent utilisé.

Le code de ce 1^{er} croquis s'apparente plutôt à une démo de l'algorithme d'asservissement, car il faut le recompiler et le télécharger à chaque fois que l'on change une de ces constantes, ce qui n'est pas très pratique. Le 2^e croquis utilise le bouton-poussoir intégré à l'ATOM pour modifier le seuil de la tension de sortie du capteur Hall pendant l'exécution. Au démarrage de ce programme, le niveau est à une valeur prédéfinie ; ensuite il suffit d'appuyer sur le bouton-poussoir pour la diminuer jusqu'à ce que la charge plane sous l'électroaimant. Avec la version actuelle de ce croquis (*MagLev_2*).

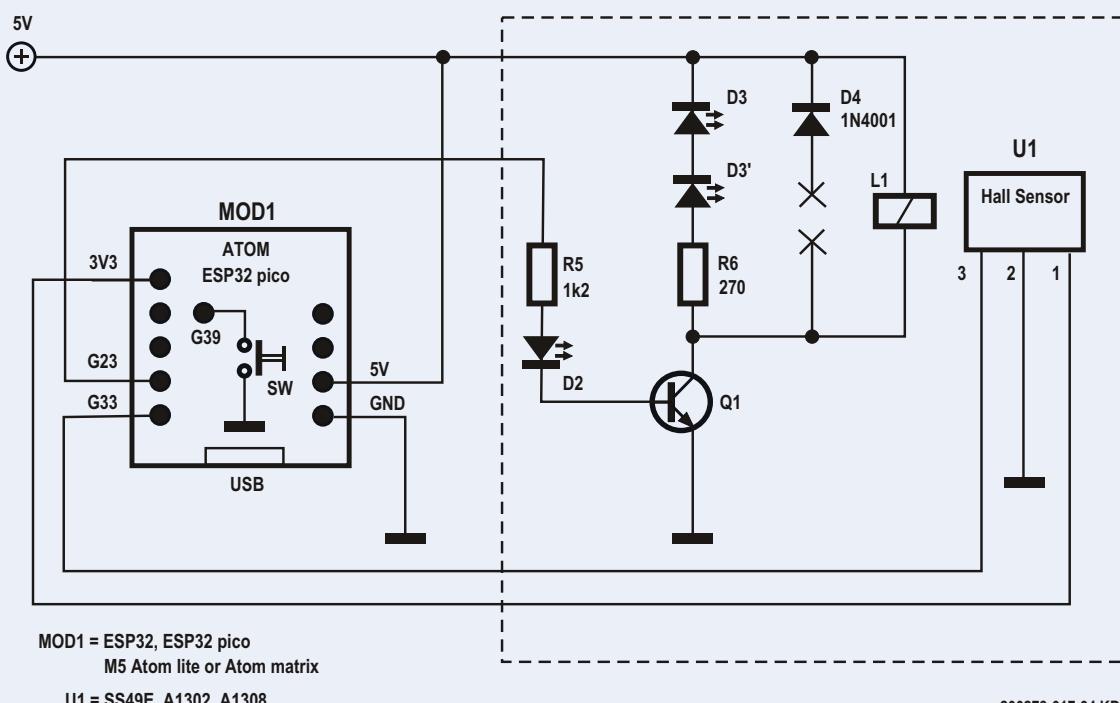


Figure 1. Le schéma ressemble à la version analogique, mais nécessite moins de composants.

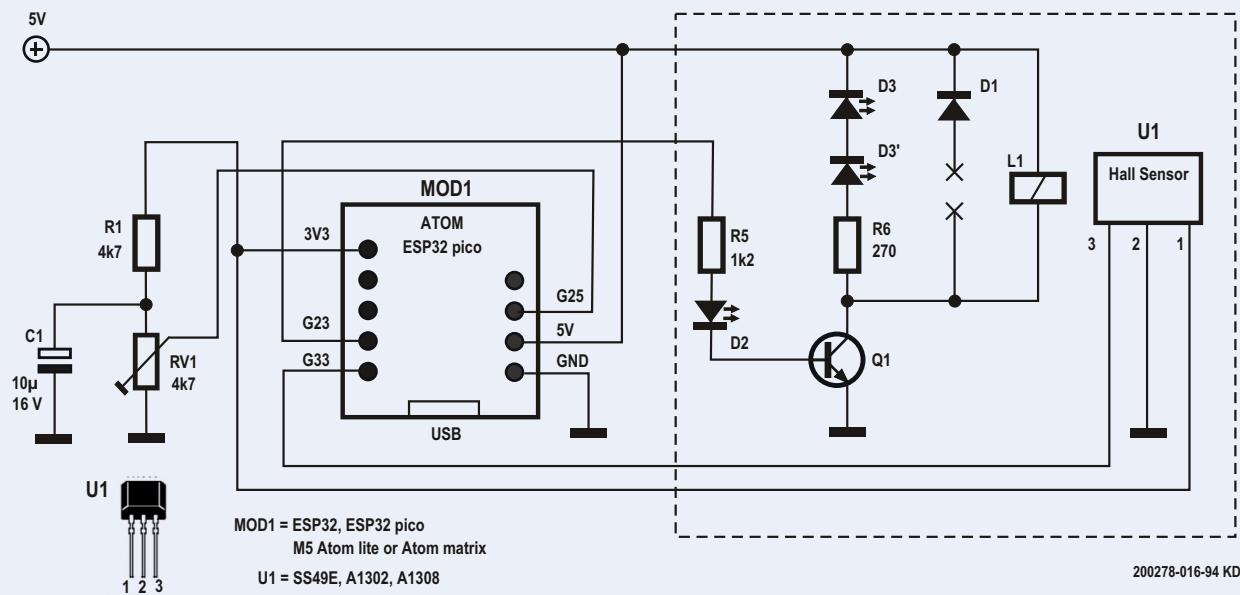


Figure 3. Trois pièces supplémentaires simplifient le réglage.

INO) et avec un seul bouton-poussoir défini par l'utilisateur pour commander l'étalonnage, le seuil ne peut être que décrémenté. Si la valeur réglée est trop basse, il faut réinitialiser l'ATOM et recommencer le processus. Le moniteur série de l'EDI Arduino permet de suivre la progression et une fois la valeur correcte du seuil déterminée, elle peut être entrée dans le code source du croquis qui peut ensuite être

recompilé et téléchargé dans le module ATOM. Bien sûr, tant que la charge n'est pas modifiée, aucun autre réglage n'est nécessaire lorsque le matériel est réinitialisé ou remis sous tension. En regardant le code de ce croquis, on remarque qu'il y a plusieurs lignes commentées avec différents paramètres par défaut pour le seuil. Bien sûr, chaque charge a son seuil et ce dernier peut varier selon le type de capteur

à effet Hall. Mais il y a des différences même lorsque le module ATOM Matrix est remplacé par la version ATOM Lite. Apparemment, les caractéristiques des CA/N (ou leur tension de référence ?) diffèrent d'un module à l'autre. L'hystérosis semble être un peu moins critique, mais c'est utile d'expérimenter avec différentes valeurs pour stabiliser l'objet en lévitation. Ce paramètre est codé en dur pour toutes les versions du logiciel, il ne peut donc être modifié que dans le code source qui doit être recompilé et téléchargé comme micropogramme.

À la croisée numérique/ analogique...

Le bouton-poussoir n'est pas la méthode la plus confortable pour régler ce projet de lévitation. Une fois que le programme a été chargé, il faut une connexion à un ordinateur et une reprogrammation de l'ESP32 pour adapter la commande par logiciel à la charge, car on ne veut pas régler les deux paramètres à chaque remise sous tension. Si la charge doit être modifiée ultérieurement, il faut à nouveau se connecter à un ordinateur pour adapter le seuil au nouvel objet. À cet égard, la solution analogique de la première partie était vraiment « la solution sans peine » : un tournevis suffisait pour régler le potentiomètre sur la bonne position pour que la nouvelle charge plane. Mais comme mentionné plus haut dans cet article, Peter a eu l'idée d'ajouter un potentiomètre pour ajuster le seuil pendant

ATOM Lite et ATOM Matrix de M5Stack

Le boîtier Lite (un peu) moins cher est légèrement moins haut que le boîtier Matrix (quelques mm), la différence est probablement due à la matrice Neopixel de ce dernier. En outre, le Matrix contient une LED IR et une unité de mesure inertielle (IMU) MPU8668. La différence de prix entre les deux versions est minime, et si vous souhaitez utiliser les modules pour de futurs projets, vous devriez peut-être considérer les extras du Matrix (un peu) plus cher.



Configuration de l'EDI Arduino

Pour utiliser le module ATOM Lite ou Matrix dans l'EDI Arduino, il faut une certaine préparation. En supposant que l'EDI lui-même soit déjà installé, la 1^{re} étape consiste à ajouter le module complémentaire *ESP32 Arduino core* dans *Fichiers* → *Préférences de l'EDI* et à saisir https://m5stack.oss-cn-shenzhen.aliyuncs.com/resource/arduino/package_m5stack_index.json dans le champ *URL de gestionnaire de cartes supplémentaires*. Fermez les *Préférences* en cliquant sur *OK*, puis ouvrez *Outils* → *Type de carte* → *Gestionnaire de cartes...*, entrez « M5Stack » dans le champ de recherche en haut de cette fenêtre et appuyez sur le bouton d'installation « M5Stack by M5Stack official » ; au moment de la rédaction de cet article, la version 1.0.7 était la plus récente. Ensuite, vous pouvez sélectionner le module M5Stack ATOM dans *Outils* → *Type de carte* → *M5Stack Arduino* → *M5Stack-ATOM* ; n'oubliez pas de sélectionner le port série correct dans *Outils* → *Port*. En outre, si vous voulez utiliser les LED Neopixel, vous devez installer la bibliothèque Neopixel d'Adafruit via *Croquis* → *Inclure une bibliothèque* → *Gérer les bibliothèques*

l'exécution, comme on peut le voir sur le schéma modifié de la **figure 3**. Le curseur du potentiomètre prégréglé est connecté à un deuxième canal analogique de l'ESP32. Cela peut être une solution temporaire pour régler facilement la lévitation magnétique au seuil correct, au lieu d'utiliser le bouton-poussoir

LISTE DES COMPOSANTS

Résistances

R5* = 1,2 kΩ
R6 = 270 Ω

Semi-conducteurs

D2* = LED rouge
D3, D3' = LED blanche (voir texte)
D4* = non montée (ou à retirer de la carte à relais)
Q1* = BC550
U1 = capteur à effet Hall A1302 ou A1308 (Allegro) ou SS49E (Honeywell)

Divers

Carte à relais 5 V*

MOD1 = module ATOM Lite ou ATOM Matrix de M5Stack

Disques magnétiques en néodyme

*= voir [1] pour les détails sur la carte de l'électroaimant/relais.

Composants supplémentaires pour le réglage analogique

Résistances

R1 = 4,7 kΩ
RV1 = trimmer 5 kΩ

Condensateurs

C1 = 10 µF, 10 V radial

comme dans le croquis précédent. Ainsi, la 3^e version du logiciel (*MagLev_3.I/NO*) peut aider à déterminer le seuil correct qui peut ensuite être codé en dur dans *MagLev_2.I/NO*. Le code source est reproduit dans cet article. Pour simplifier, la partie concernant les LED Neopixel du module ATOM Matrix est omise du croquis. Si on souhaite réduire le matériel au strict minimum, le trimmer, la résistance et le condensateur électrolytique supplémentaires (R1, RV1 et C1) peuvent être omis ultérieurement, le seuil correct étant fixé dans la deuxième version du croquis. La photo de la **figure 4** montre que ces composants ne prennent que peu de place et peuvent effectivement être cachés. À mon avis, ce

réglage analogique de la version numérique est un ajout précieux au projet de lévitation. Je ne le supprimerais pas !

Pour télécharger le code source des trois croquis, voir [6].

Construction

Les photos de la **figure 5** montrent comment réaliser la construction mécanique du projet de lévitation numérique. L'électroaimant (la carte à relais modifiée) avec le capteur Hall est monté en haut d'un cadre en bois de 10 cm de Ø, le module M5Stack ATOM en bas. Si l'option « trimmer » est utilisée, le meilleur emplacement pour les trois composants supplémentaires est dans ou à la base de



Figure 4. Trimmer à la base de la construction mécanique.



Figure 5. Construction mécanique, vue d'ensemble.



PRODUIT

➤ Sphère magnétique en kit

www.elektor.fr/magnetsphere-kit

la construction : c'est mécaniquement plus stable pendant le réglage et il est plus facile d'y cacher les composants.

Analogique ou numérique ?

Être facile ou ne pas l'être, telle est la question. Nous vous laissons le soin de préférer la version « vieille école » avec comparateur analogique ou la version plus moderne à microcontrôleur de cette seconde partie. Comme nous l'avons mentionné plus haut, la tension d'alimentation de 3,3 V pour le capteur Hall limite le poids de la charge qui peut léviter avec cette solution numérique. Les modules ESP32 ont été utilisés en raison de leur prix et de leur petite taille, et non parce qu'il fallait une connexion « sans fil ». Au bout du compte, dans cette réalisation, l'objet en lévitation est la seule chose sans fil !

(200278-04)

Des questions, des commentaires ?

Envoyez un courriel au rédacteur (luc.lemmens@elektor.com) ou contactez Elektor (redaction@elektor.fr).

Contributeurs

Conception : Peter Neufeld
Texte et rédaction : Luc Lemmens
Illustrations : Peter Neufeld, Patrick Wielders, Luc Lemmens
Mise en page : Harmen Heida
Traduction : Yves Georges



Le croquis : la méthode numérique

```
int HALL_PIN = 33; //Entrée analogique du capteur Hall
int HALL_VAL = 0; //Sortie du capteur Hall
int ADJ_PIN = 25; //Entrée analogique du trimmer (0 à 2,5 V)
int ADJ_VAL = 0; //Seuil de déclenchement pour la charge utile
réelle
int HYSTERESIS = 50; //Hystérésis absolue pour le seuil de
déclenchement
int HYST = 0; //Ajoute ou soustrait l'hystérésis
int RELAIS_PIN = 23; //Sortie vers relais/électro-aimant 5 V
int i = 0;

//-----
//Sortie vers l'électroaimant
void setup(){

//Par défaut : CAN 12 bits avec atténuation de 11 dB.
//-----
void loop(){
    HALL_VAL = 0;
    ADJ_VAL = 0;
    for (i=1; i <= 2; i++){ //Lire 2 fois pour réduire le bruit
        HALL_VAL = (HALL_VAL + analogRead(HALL_PIN));
        ADJ_VAL = ADJ_VAL + analogRead (ADJ_PIN);
    }
    HALL_VAL = HALL_VAL / (i-1); //Calculer la moyenne
    ADJ_VAL = ADJ_VAL / (i-1); //Calculer la moyenne
    ADJ_VAL = (ADJ_VAL/3)+2450; //Décaler vers une gamme adéquate
    if (HALL_VAL < (ADJ_VAL + HYST)){ //Charge trop faible
        digitalWrite(RELAIS_PIN, HIGH); //Soulever la charge utile
        HYST = HYSTERESIS; //Définir le niveau supérieur d'hystérésis
    }
    else { //Charge trop élevée :
        digitalWrite(RELAIS_PIN, LOW); //Relâcher la charge utile
        HYST = 0 - HYSTERESIS //Définir le niveau inférieur d'hystérésis
    }
}
//-----
```

LIENS

- [1] « lévitation magnétique sans peine », elektor 07-08/2021 : www.elektormagazine.fr/200311-04
- [2] P. Neufeld, « Magnetic Levitation - The Easy Way » :
<https://peterneufeld.wordpress.com/2020/06/20/magnetic-levitation-the-easy-way/>
- [3] P. Neufeld, « Magnetic Levitation - The Digital Way » :
<https://peterneufeld.wordpress.com/2020/07/04/magnetic-levitation-the-digital-way/>
- [4] P. Neufeld, « Magnetic Levitation - Reloaded » :
<https://peterneufeld.wordpress.com/2021/05/01/magnetic-levitation-reloaded/>
- [5] Diode de roue libre ou d'amortissement (flyback ou snubber) : https://en.wikipedia.org/wiki/Snubber#Diode_snubbers
- [6] Téléchargement du logiciel : www.elektormagazine.fr/200278-04

ULTIMATE Arduino Uno Hardware Manual

Extrait : chargeur d'amorçage du microcontrôleur principal



Warwick A. Smith (Afrique du Sud)

Cet article contient environ un quart de chapitre du livre intitulé *Ultimate Arduino Uno Hardware Manual* de Warwick A. Smith, publié par Elektor en juin 2021. Au fil du temps, l'Arduino Uno est devenu un « produit de base ». Comme il est bon marché, l'utilisateur est peu regardant sur les caractéristiques matérielles et il peut se retrouver en difficulté pour faire fonctionner son application. Le chargeur d'amorçage de l'Uno et les réglages des fusibles sont obscurs, la plupart des questions techniques reçues chez Elektor portent sur ces sujets. Les réponses « définitives » données par ce nouveau livre d'Elektor seront donc appréciées.

Note de l'éditeur : cet article est un extrait du livre *Ultimate Arduino Uno Hardware Manual* 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 ? ».



Chargeur de démarrage du microcontrôleur principal

Sur une carte Arduino Uno, si le microcontrôleur principal (ATmega328P) est remplacé par une puce vierge, il faudra y planter le chargeur. C'est faisable avec Microchip Studio et un programmeur USB externe connecté au connecteur ICSP. La programmation a lieu de la même manière que pour l'ATmega16U2, mais il faut utiliser le connecteur ICSP à l'extrémité de la carte, côté opposé au connecteur USB. De cette façon, il est aussi possible de sauvegarder le chargeur déjà implanté dans un ATmega328P.

Le détail de ces actions figure dans les sections qui suivent et incluent les réglages des fusibles pour l'ATmega328P utilisé sur une carte Arduino Uno. Le chargeur peut aussi être restauré rapidement à partir de l'EDI Arduino, mais avec Microchip Studio installé, cela ne fonctionne pas à cause d'un conflit de pilotes.

Sauvegarde du logiciel de l'ATmega328P avec Microchip Studio

Par précaution, le logiciel de l'ATmega328P peut être lu et copié sur la carte Arduino Uno et sauvegardé dans des fichiers HEX et BIN avec l'utilitaire de programmation des puces de Microchip Studio. C'est très pertinent si on utilise une carte clone qui peut avoir un chargeur de démarrage différent de celui des cartes Arduino Uno authentiques.

Branchez un programmeur USB adéquat à l'ordinateur, puis le câble avec le connecteur femelle à 6 broches du programmeur dans le connecteur ICSP à l'extrémité de la carte Arduino Uno. Consultez le brochage de l'embase ICSP pour ATmega328P qui indique où se trouve la broche 1. Alimentez l'Arduino Uno soit par l'USB, soit avec une alimentation externe.

Démarrez Microchip Studio et ouvrez l'utilitaire *Device Programming* en cliquant sur l'icône *Device Programming* dans la barre d'outils du haut, ou en sélectionnant *Tools Device Programming* sur le menu d'accueil de Microchip Studio. Dans la boîte *Device Programming*, sélectionnez le programmeur USB correct sous *Tools*, l'ATmega328P sous *Device*, vérifiez qu'ISP (*In-System Programmer*) est sélectionné sous *Interface*, et cliquez sur le bouton *Apply*. Pour vérifier que tout est bien configuré et que Microchip Studio peut se connecter à la cible, cliquez sur le bouton *Read* sous *Device Signature*. La signature de la cible et la consigne de tension seront lues et affichées dans la boîte de dialogue *Device Programming*.

Dans la colonne de gauche de la boîte *Device Programming*, cliquez sur *Memories*. Dans la section *Flash* de *Memories* de la boîte de dialogue, cliquez sur *Read...* pour naviguer jusqu'à un dossier où stocker le fichier HEX récupéré de la mémoire Flash de l'ATmega328P. Au bas de la boîte *Save as*, entrez un nom valide au choix pour le fichier HEX. Cliquez sur *Save* quand c'est terminé. L'utilitaire *Device Programming* lit la mémoire Flash de l'ATmega328P et la sauvegarde dans un fichier HEX du nom choisi. Ce fichier HEX peut être utilisé pour restaurer le logiciel original de l'ATmega328P à tout moment. Le fichier HEX comprend toute la mémoire Flash et donc le croquis utilisateur programmé dans la mémoire Flash au moment de la création du fichier HEX.

Micrologiciel du chargeur de démarrage Optiboot

Les cartes Arduino Uno utilisent désormais le chargeur **Optiboot** dans le microcontrôleur ATmega328P. Il remplace un ancien chargeur. Optiboot occupe un segment de 512 octets de mémoire Flash, l'ancien chargeur avait besoin de 2 Ko. Il y a donc 1,5 Ko de mémoire Flash de plus disponibles pour les croquis de l'utilisateur. Le code source C et les fichiers HEX d'Optiboot se trouvent dans le dossier Arduino EDI, sous le chemin :

```
arduino-1.8.13\hardware\arduino\avr\bootloaders\optiboot
```

où le dossier racine aura un nom différent pour les différentes versions de l'EDI Arduino. Les mêmes fichiers figurent sur les pages GitHub Arduino [1].

Le projet Optiboot est sur GitHub à l'adresse github.com/Optiboot/optiboot et pour en savoir plus, le wiki Optiboot est à l'adresse github.com/Optiboot/optiboot/wiki.

Optiboot fait clignoter la LED L de l'Arduino au démarrage. Il commence alors à exécuter le croquis présent, ou attend qu'un nouveau croquis soit chargé si l'EDI de l'Arduino le réinitialise. Pour plus de détails, voir le Wiki Optiboot [2].

Restauration du chargeur de démarrage

Microchip Studio et un programmeur USB externe permettent de restaurer/charger le chargeur d'amorçage Optiboot sur l'ATmega328P d'origine ou un vierge, comme cela a déjà été décrit dans ce chapitre pour l'ATmega16U2. Pour se connecter à l'ATmega328P depuis Microchip Studio, utilisez la même méthode que pour sauvegarder le chargeur d'amorçage de la section « *Backing up the ATmega328P Firmware with Microchip Studio* ». Pour charger Optiboot sur l'ATmega328P de l'Arduino Uno, connectez le programmeur USB au connecteur ICSP à l'extrémité de la carte comme décrit ci-dessus. Ouvrez l'utilitaire *Device Programming* dans Microchip Studio, et de même, connectez le programmeur et l'ATmega328P. Cliquez sur l'élément *Memories* dans la colonne de gauche de la boîte *Device Programming*. Cliquez sur [...] dans la section *Flash* de la boîte de dialogue pour en ouvrir une autre qui sert à naviguer vers le fichier HEX à charger dans l'ATmega328P. Le fichier HEX de restauration du chargeur de démarrage peut être soit celui créé par la sauvegarde du chargeur, comme décrit à la section 5.4.1, soit le fichier HEX correct provenant de :

```
arduino-1.8.13\hardware\arduino\avr\bootloaders\optiboot
```

qui pour l'Arduino Uno est *optiboot_atmega328.hex*. Sélectionnez le bon fichier dans la boîte de dialogue, puis cliquez sur *Open*. De retour dans la boîte *Device Programming*, cliquez sur *Program* dans la section *Flash* de celle-ci. Si tout est bien configuré, la mémoire Flash sera programmée avec le fichier HEX sélectionné.

Si le chargeur de démarrage a été installé sur un ATmega328P vierge, les fusibles de celui-ci doivent être réglés correctement avant de

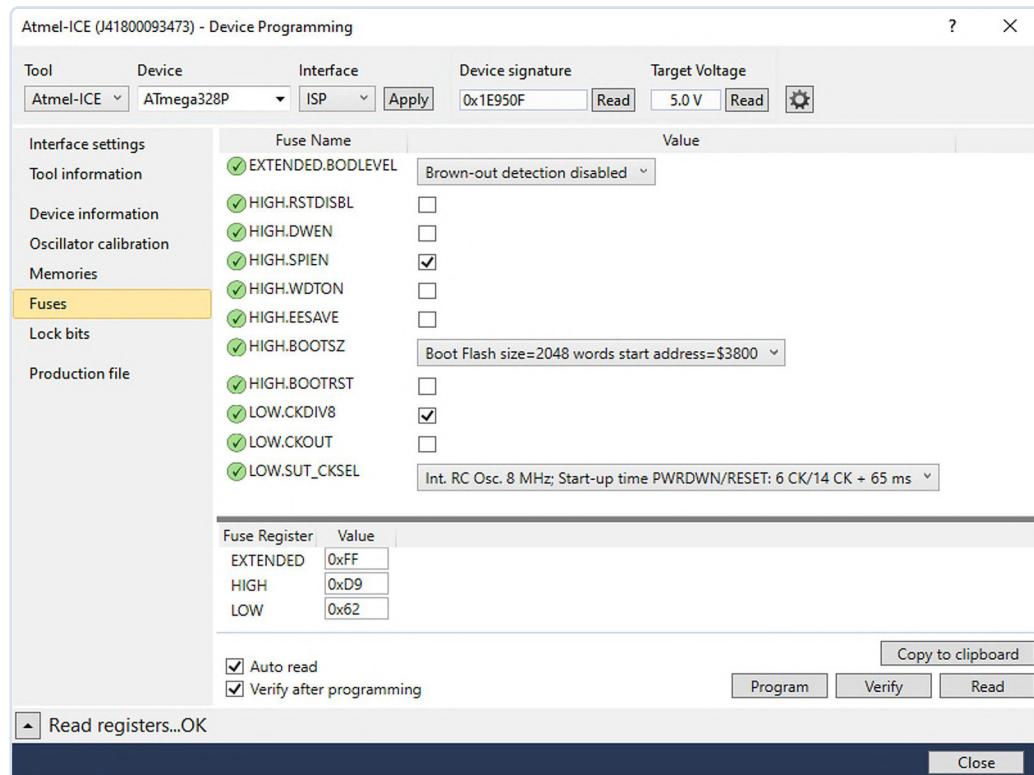


Figure 1. Paramètres par défaut des fusibles pour un ATmega328P vierge.

continuer, comme décrit plus loin. Si les fusibles sont déjà réglés correctement, fermez la boîte *Device Programming* de Microchip Studio et débranchez l'alimentation de l'Arduino Uno. Déconnectez le programmeur USB de l'Arduino, puis remettez l'Arduino sous tension. Si les fusibles sont déjà bien réglés, testez le chargeur de démarrage tout juste implanté en ouvrant l'EDI Arduino et en chargeant un croquis de test avec la méthode normale du câble USB.

Réglage des fusibles de l'ATmega328P

Avec un ATmega328P vierge, il faut d'abord charger le bootloader Optiboot comme décrit ci-dessus. Les fusibles de l'ATmega328P doivent ensuite être réglés pour être utilisés dans l'Arduino Uno. Les valeurs des fusibles peuvent être lues et réglées avec l'utilitaire *Device Programming* de Microchip Studio. Pour lire et régler les fusibles, d'abord branchez un programmeur USB externe sur le connecteur ICSP à l'extrémité de l'Arduino Uno comme décrit plus haut. Dans Microchip Studio, ouvrez l'utilitaire *Device Programming*, choisissez les paramètres corrects pour l'outil, la puce et l'interface comme déjà décrit, puis cliquez sur *Apply* pour vous connecter. Dans la colonne de gauche de la boîte *Device Programming*, cliquez sur *Fuses*.

La **figure 1** montre ce à quoi il faut s'attendre pour les valeurs des fusibles d'un microcontrôleur ATmega328P vierge dans la boîte *Device Programming*. Les valeurs par défaut des registres de fusibles sont :

EXTENDED	0xFF
HIGH	0xD9
LOW	0x62

Ces fusibles doivent être programmés selon les valeurs de la **figure 2** et du **tableau 1**. Les valeurs des fusibles suivants doivent être modifiées :

Fusible	Remplacer par :
EXTENDED.BODLEVEL	Brown-out detection at VCC=2.7V
HIGH.EESAVE	(optional, see next page for an explanation)
HIGH.BOOTSRZ	Boot Flash size=256 words; start address=\$3F00
HIGH.BOOTRST	checked
LOW.CKDIV8	unchecked
LOW.SUT_CKSEL	Ext. Crystal Osc. 8.0- MHz; Start up time PWRDWN/RESET: 16K/14 CK + 65ms

Une fois ces réglages effectués, les valeurs des registres de fusibles du bas de la boîte de dialogue sont :

EXTENDED	0xFD
HIGH	0xDE (EESAVE unchecked) or 0xD6 (EESAVE checked)
LOW	0xFF

En général, EESAVE est coché sur les anciennes cartes Arduino Uno R3, mais décoché sur les nouvelles. Si la case EESAVE (fuse HIGH.EESAVE) est cochée dans la boîte de dialogue, elle protège le contenu de l'EEPROM si la puce est effacée.

Après avoir effectué les changements, cliquez sur *Program* près du bas de la boîte de dialogue. Les nouvelles valeurs des fusibles seront alors appliquées à l'ATmega328P. Pour continuer, cliquez sur *OK* dans la boîte d'avertissement. Fermez la boîte *Device Programming*,

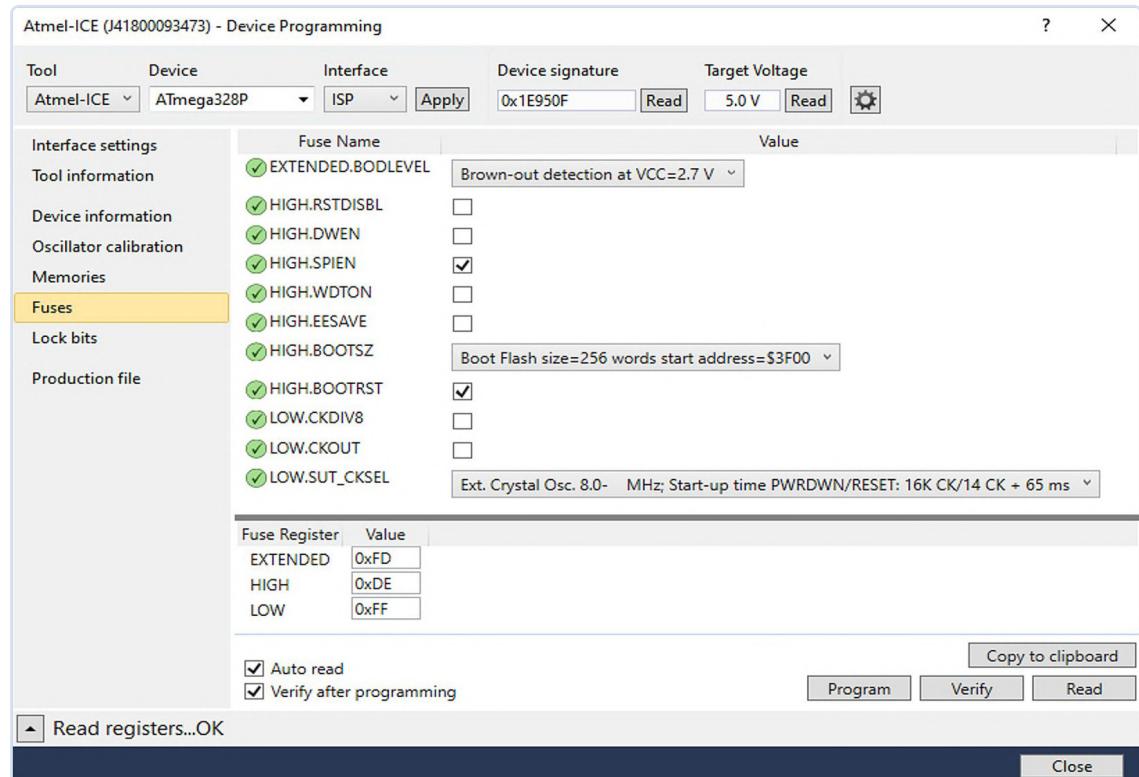


Figure 2. Réglage des fusibles de l'ATmega328P pour l'Arduino Uno.

Tableau 1. Réglage des fusibles de l'Arduino Uno ATmega328P.

Nom du fusible	Valeur
EXTENDED.BODLEVEL	Brown-out detection level at VCC=2.7 V
HIGH.RSTDISBL	unchecked
HIGH.DWEN	unchecked
HIGH.SPIEN	<input checked="" type="checkbox"/>
HIGH.WDTON	unchecked
HIGH.EESAVE	unchecked on new boards
HIGH.BOOTSZ	Boot Flash size=256 words, start address=\$3F00
HIGH.BOOTRST	<input checked="" type="checkbox"/>
LOW.CKDIV8	unchecked
LOW.CKOUT	unchecked
LOW.SUT_CKSEL	Ext.Crystal Osc. 8.0 MHz; Start-up PWRDWN/RESET: 16K CK/14 CK + 65 ms

débranchez l'alimentation de l'Arduino Uno, puis le programmeur USB. Branchez l'Arduino Uno au port USB du PC, puis chargez-y un croquis à partir de l'EDI Arduino pour vérifier que le chargeur fonctionne et que les paramètres des fusibles sont corrects.

Le pont de soudure RESET-EN

Certains des programmeurs AVR USB ont un débogueur, notamment l'AVR Dragon (Microchip ne le vend plus) et l'Atmel-ICE (il remplace les anciens programmeurs USB). Notez que

l'AVRISP mkII (Microchip ne le vend plus) n'a pas de capacité de débogage, c'est un programmeur au sens strict. C'est aussi le cas des programmeurs proposés aux amateurs, tels que l'USBasp et l'USBtinyISP, qui sont des appareils limités à la programmation. Le débogage fait référence à la capacité du programmeur/débogueur USB à fonctionner conjointement avec un logiciel tel que Microchip Studio, et permettre à l'utilisateur, d'accéder au microcontrôleur principal de l'Arduino pour visualiser le contenu de sa mémoire et de ses registres internes, ainsi que de parcourir le code source en une seule étape et d'y insérer des points d'arrêt. La version 1.8.13 de l'EDI Arduino, la version en cours au moment de la rédaction, ne dispose pas de cette capacité.

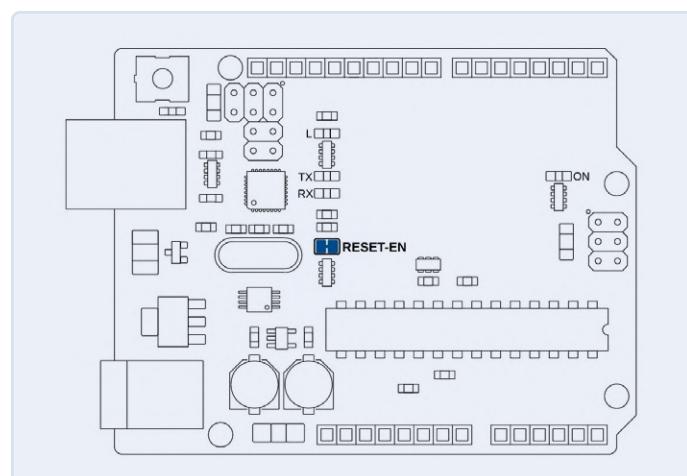


Figure 3. Emplacement du pont de soudure RESET-EN sur l'Arduino Uno.



Pour exploiter les capacités de débogage du programmeur/débogueur USB, on le branche au connecteur ICSP comme on le ferait normalement pour programmer la mémoire Flash du dispositif AVR cible, ou pour lire les valeurs des fusibles. Un programme en C ou C++ peut ensuite être chargé sur l'AVR à l'aide de Microchip Studio. Microchip Studio peut alors être utilisé pour déboguer l'AVR cible.

Le système de débogage du microcontrôleur ATmega328P s'appelle **debugWIRE** et permet au logiciel de débogage de se connecter au microcontrôleur cible à travers le connecteur ICSP au moyen d'un programmeur USB capable de débogage. Pour que debugWIRE fonctionne sur l'ATmega328P des cartes Arduino Uno, il faut déconnecter le circuit de réinitialisation du signal RESET. Les cartes Arduino Uno ont un pont appelé RESET-EN formé de deux pastilles soudées reliées par une piste de circuit imprimé. Il faut couper cette piste afin de déconnecter une partie du circuit de réinitialisation du signal RESET lors de l'utilisation de debugWIRE. Pour rétablir la connexion, ressoudez les deux pastilles ensemble. La **figure 3** montre où sont les pastilles de soudure RESET-EN sur l'Arduino Uno. Répétons que cette fonction ne sera utilisée que par les utilisateurs avancés qui programmrent la carte en C avec Microchip Studio. Les lecteurs désireux d'apprendre le langage C et programmer les cartes Arduino seront intéressés par le livre *C Programming with Arduino* (ISBN 978-1-907920-46-2) du même auteur, aux éditions Elektor International Media. Ce livre comprend un chapitre sur le débogage des cartes Arduino Uno et Mega 2560 à l'aide d'Atmel Studio (depuis remplacé par Microchip Studio).

Autres méthodes de programmation du micrologiciel

Des méthodes alternatives pour programmer le chargeur d'amorçage d'un Arduino Uno sont présentées sur le site *playground Arduino*, section *Bootloader* [3]. On y trouve des infos sur l'utilisation d'un second Arduino comme programmeur d'un Arduino cible



PRODUITS



- **Livre en anglais**
« **Ultimate Arduino Uno Hardware Manual** »
de W. A. Smith, Elektor 2021

Version papier : www.elektor.fr/19678

Version numérique : www.elektor.fr/19679



- **Livre en anglais**
« **C Programming with Arduino** »
de W. A. Smith, Elektor 2018

Version papier : www.elektor.fr/17574

Version numérique : www.elektor.fr/18209

ainsi que sur l'utilisation d'autres programmeurs.

L'exploitation d'un Arduino comme ISP est aussi décrite sur le site [4]. Le chapitre 2, section 2.8, du livre contient plus d'informations sur le chargement des croquis sur l'Arduino Uno avec un programmeur externe et l'EDI Arduino. Consultez la même section pour des infos sur la restauration du chargeur d'amorçage sur le microcontrôleur principal avec un programmeur externe et l'EDI Arduino.

Enfin, voir [5] pour des informations supplémentaires sur la programmation DFU (Device Firmware Update) de l'ATmega16U2. ▶

(210345-04)

Des questions, des commentaires ?

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

Contributeurs

Texte et illustrations : Warwick A. Smith

Rédaction : Jan Buiting

Traduction : Yves Georges

Mise en page : Sylvia Sopamena

LIENS

- [1] Optiboot sur Github : www.github.com/arduino/ArduinoCore-avr/tree/master/bootloaders/optiboot
- [2] Wiki Optiboot : www.github.com/Optiboot/optiboot/wiki/HowOptibootWorks
- [3] Chargeur de démarrage Arduino : <https://playground.Arduino.cc/Main/ArduinoCoreHardware/#Bootloader>
- [4] Arduino comme ISP : <https://www.arduino.cc/en/Tutorial/BuiltInExamples/ArduinoISP>
- [5] Programmation DFU (Device Firmware Update) de l'ATmega16U2 : <https://www.arduino.cc/en/Hacking/DFUProgramming8U2>

microPython pour ESP32 et ses copains

Partie 2 : piloter facilement les écrans matriciels

Günter Spanner (Allemagne)

Le premier article de cette série portait sur l'interface de développement et les commandes simples. Passons maintenant aux applications pratiques. Par exemple, avec des bibliothèques de programmes appropriées, fabriquer et programmer des écrans matriciels à LED de grande taille devient un jeu d'enfant. Outre le défilement de textes, vous pouvez également afficher l'heure ou la température locale.

Le mini-écran du premier article [1] était plutôt destiné aux plus petits appareils. Dans cet article, nous montrons comment réaliser un afficheur de grande surface avec l'aide de Python. Les dispositifs choisis ici sont des matrices de LED de $8 \times 8 = 64$ LED individuelles chacune. En joignant plusieurs unités, il est possible de construire des afficheurs de taille presque illimitée. Ils peuvent servir comme télescripteurs pour les cours de bourse ou panneaux d'affichage de résultats sportifs à la maison. Avec quelques instructions en Python, on peut créer de grands afficheurs pour l'heure ou la température, qui peuvent parfaitement servir à des fins publicitaires à de multiples occasions.

Afficheurs à matrice de points

Les afficheurs matriciels permettent de produire n'importe quels symboles et graphiques simples en plus de l'affichage de chiffres et de lettres. Ils sont donc plus intéressants que les afficheurs classiques à sept segments, qui ne permettent que l'affichage des chiffres de 0 à 9. En outre, une fois spécifiée, la taille des afficheurs à sept segments est fixe. Les matrices de points, en revanche, peuvent être mises à l'échelle comme on le souhaite.

Il est aussi possible d'obtenir une luminosité bien supérieure à celle des écrans OLED. De plus, des surfaces d'affichage de plusieurs mètres de long et de large peuvent être réalisées avec des matrices de LED. On utilise souvent cette catégorie d'afficheurs pour des panneaux d'affichage ou de publicité de grande surface dans des endroits très fréquentés, gares ou aéroports, bus et trains, etc. Dans les salles des marchés boursiers du monde entier, ils affichent les cours des actions sous forme de télescripteurs. Par ailleurs, les matrices de points sont particulièrement répandues en Asie,



car elles peuvent afficher sans problème les caractères extrême-orientaux, comme le montre la **figure 1**.

Pilotage

Les matrices de LED avec $5 \times 7 = 35$ LED sont très utilisées. Cependant, une grande variété d'autres modèles sont également disponibles. On trouve beaucoup d'éléments avec 8×8 LED. La commande directe de 8×8 points nécessiterait 65 lignes, car il faudrait raccorder 64 anodes et une cathode commune. Avec une matrice, on peut s'en sortir avec beaucoup moins de connexions. Ici, 8 LED sont connectées ensemble en lignes et en colonnes. Ainsi, seules $8 + 8 = 16$ connexions sont nécessaires. La **figure 2** montre le principe pour une matrice de 3×4 . Au lieu de 13 connexions pour la commande individuelle des LED, 7 connexions suffisent ici. Par exemple, pour allumer la 2^e LED de la 2^e ligne de la matrice, toutes les lignes, sauf la seconde, doivent être soumises au potentiel HIGH. La deuxième ligne, en revanche, doit être au potentiel GND. Pour les colonnes, seule la seconde doit être au potentiel HIGH.

Le pilotage direct des matrices de points mobilise une grande partie des ressources du contrôleur. S'il faut également enregistrer les valeurs de capteurs ou commander des actionneurs, même un contrôleur ESP32 puissant atteint rapidement ses limites. La commande d'écrans plus grands, comportant une centaine de diodes électroluminescentes ou plus, deviendrait également rapidement un problème – d'une part en raison de la puissance de calcul requise, d'autre part en raison du nombre limité de broches disponibles sur un contrôleur. En outre, la vitesse d'exécution relativement faible du code Python aurait également ici un effet négatif. Il est donc conseillé d'utiliser des puces pilotes d'écran bon marché, telles que le MAX7219. Ces composants ont une interface compatible SPI et peuvent piloter avec seulement trois broches des écrans qui comportent jusqu'à $8 \times 8 = 64$ éléments de matrice. L'interface périphérique série (SPI) est très répandue et fréquemment utilisée, notamment dans l'électronique grand public. Le système de bus et surtout son utilisation sous MicroPython sont décrits en détail dans un livre Elektor de l'auteur [2].

On trouve des modules qui regroupent le pilote et la matrice. La **figure 3** en montre un exemple.

La connexion des modules pilotes à l'ESP32 est très simple. Seules les trois broches SPI doivent être reliées à la carte à contrôleur :

- MAX7219_data (DIN) Port D02
- MAX7219_load (CS) Port D05
- MAX7219_clock (CLK) Port D04
- MAX7219_GND ESP32 GND

À cause de la consommation électrique relativement élevée, il est recommandé d'avoir une alimentation externe. Notez que la tension d'alimentation des modules doit être d'environ 3,3 à 4 V pour garantir la compatibilité avec l'ESP32. Ainsi, d'une part, les modules reçoivent toujours la tension d'alimentation minimale admissible selon la fiche technique. D'autre part on évite une surtension sur les entrées du contrôleur. Il est également possible d'utiliser un convertisseur de niveau bidirectionnel 3,3/5 V.

Il faut aussi prévoir des condensateurs de blocage suffisants. Des lignes VCC et de masse à faible résistance sont également essentielles. Si vous prenez ces précautions, vous pouvez construire des écrans de presque toutes les tailles. Le contrôleur n'est



Figure 1. Les matrices de points à LED sont très populaires en Asie.

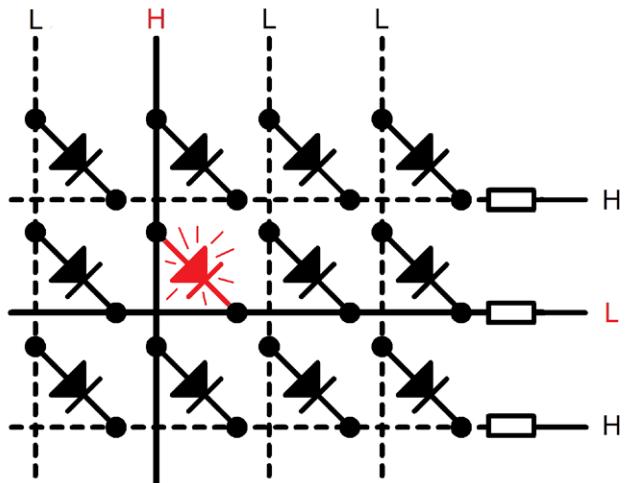


Figure 2. Principe de commande des matrices de points.

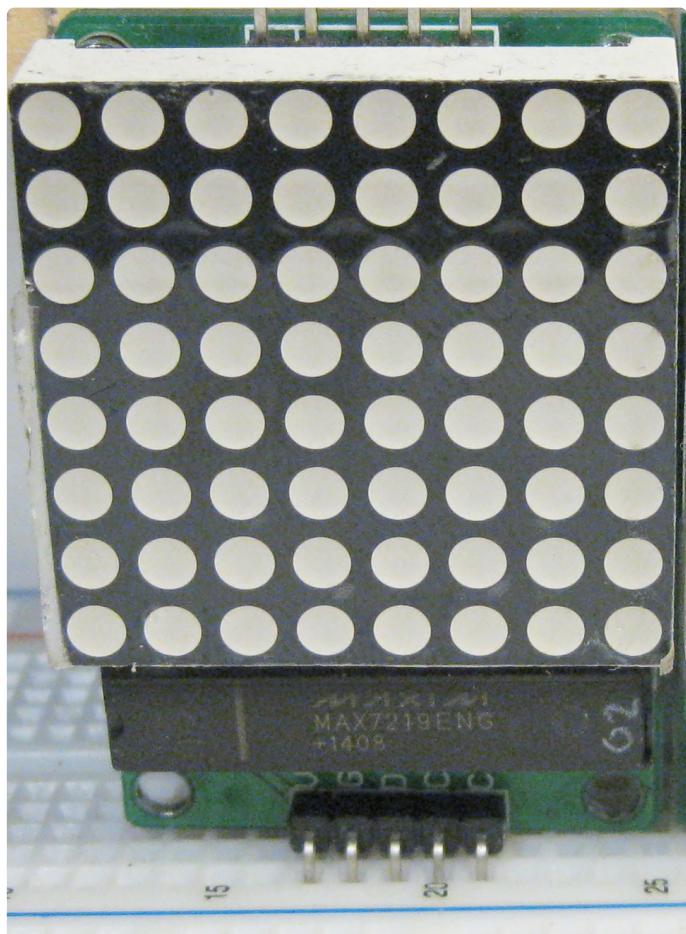


Figure 3. Module matriciel à LED.

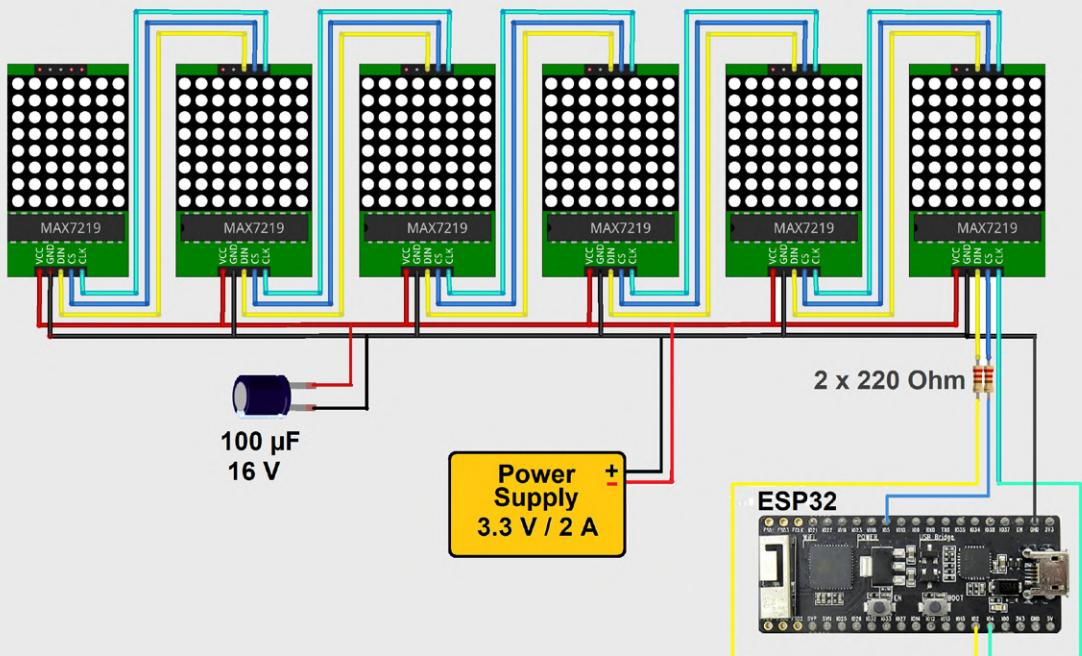


Figure 4. Regroupement de modules matriciels à LED.

pratiquement plus chargé, car il ne fait qu'envoyer des commandes simples via le bus SPI. En outre, suffisamment de broches restent libres pour gérer des capteurs externes ou d'autres périphériques. La réalisation d'écrans de grande surface, par ex. à des fins publicitaires ou pour des événements sportifs, n'est donc plus un obstacle. La **figure 4** montre le schéma du circuit complet pour réaliser un afficheur matriciel à six chiffres avec un module à contrôleur ESP32.

Réussir avec la bonne bibliothèque

Vous trouverez dans le paquet à télécharger en [4] un lien pour une bibliothèque Python spécialisée dans la commande des circuits intégrés Maxim. Après avoir chargé le fichier pilote *Max7219.py* sur le contrôleur, les instructions suivantes sont disponibles :

```
spi = SPI(1, baudrate=10000000, polarity=1, phase=0,  
          sck=Pin(CLK), mosi=Pin(DIN))  
ss = Pin(CS, Pin.OUT)
```

Pour cela il faut affecter les broches ainsi :

CIK = 4, DIN = 2 et CS = 5

Ensuite, un objet d'affichage peut être créé :

```
display = max7219.Matrix8x8(spi, ss, MN)
```

où MN est le nombre d'éléments de matrice utilisés. Ensuite, les textes et les graphiques peuvent être dessinés à l'aide des commandes du **listage 1**. Il est ainsi possible de créer des publicités efficaces, faciles à lire même à plusieurs mètres de distance.

Matrice de LED en action

Le code suivant montre un exemple d'application pour un affichage avec six éléments de matrice 8 x 8 :

```
# LED_matrix_test.py

import max7219
from machine import Pin, SPI

spi = SPI(1, baudrate=10000000, polarity=1, phase=0,
          sck=pin(4), mosi=pin(2))
ss = Pin(5, Pin.OUT)
```

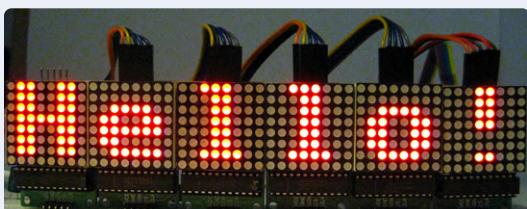


Figure 5. Affichage matriciel par points avec six éléments de matrice 8×8



Figure 6. Affichage matriciel par points avec douze éléments de matrice



Listage 1. Commandes d'affichage.

```
display.pixel(x,y,1)          # définit un pixel aux coordonnées x, y
display.pixel(x,y,0)          # supprime un pixel aux coordonnées x, y
display.hline(x,y,l,1)        # ligne horizontale de x, y - longueur l
display.vline(x,y,l,1)        # ligne verticale de x, y - longueur l
display.line(a,b,c,d,1)       # ligne de a, b à c, d
display.rect(a,b,c,d,1)       # rectangle avec coins a, b, c, d
display.text('Text',x,y,1)     # texte à la position x, y
display.scroll(x,0)           # défilement par x pixels
display.show()                # rafraîchit l'affichage
```

```
display = max7219.Matrix8x8(spi, ss, 6)
display.text('Python',0,0,1)
display.show()
```

Après avoir chargé le programme dans le contrôleur ESP, le texte s'affiche sur l'écran (**fig. 5**).

L'afficheur peut être facilement agrandi avec des modules de matrice supplémentaires. La **figure 6** montre une matrice de douze éléments. L'affichage n'est pas limité aux symboles et textes statiques puisque Python permet de mettre en oeuvre sans problème des graphiques en mouvement. La section suivante en présente un exemple.

Téléscripteurs boursiers et autres

La fonction de défilement peut servir à créer des téléscripteurs. Il est ainsi possible d'afficher des textes plus longs même sur des écrans petits ou courts. Le programme du **listage 2** fait défiler le texte « Python » de droite à gauche sur un écran matriciel.

Les lettres individuelles sont ainsi transmises avec

```
display.text('...',40,0,1)
```

et créées sur le bord droit de l'écran. Elles sont ensuite déplacées vers la gauche de la valeur **pixelDistance** avec la fonction **moveLeft**:

```
for i in range(8):
    display.scroll(-1,0)
    sleep(speedFactor)
```

La valeur **speedFactor** permet de modifier la vitesse d'exécution. Admirez le résultat sur YouTube [3].

Affichage de la température en grand format

De nombreuses pharmacies, banques et autres boutiques tentent d'attirer l'attention de clients potentiels en faisant de la publicité avec de grands afficheurs de température ou d'heure. Ces afficheurs attirent également l'attention lors de manifestations sportives, de foires commerciales, d'expositions et dans les FabLabs. Pour réaliser un afficheur de température, par exemple, il suffit d'ajouter le capteur correspondant à la configuration présentée ci-dessus. Un modèle particulièrement adapté est le DS18x20 de Maxim Integrated (anciennement Dallas). Avec la bibliothèque Python appropriée, le capteur peut être lu sans problème.

Ces capteurs communiquent via le bus 1-Wire et n'occupent donc

Listage 2. Téléscripteur.

LED_matrix_ticker.py

```
import max7219
from machine import Pin, SPI
from time import sleep

spi = SPI(1,baudrate=10000000, polarity=1, phase=0,
          sck=Pin(4), mosi=Pin(2))
ss = Pin(5,Pin.OUT)
speedFactor=0.05
pixelDistance=7
display = max7219.Matrix8x8(spi, ss, 6)

def moveLeft(Pixel):
    for i in range(Pixel):
        display.scroll(-1,0)
        sleep(speedFactor)
        display.show()

while True:
    display.text('P',40,0,1)
    moveLeft(pixelDistance)
    display.text('y',40,0,1)
    moveLeft(pixelDistance)
    display.text('t',40,0,1)
    moveLeft(pixelDistance)
    display.text('h',40,0,1)
    moveLeft(pixelDistance)
    display.text('o',40,0,1)
    moveLeft(pixelDistance)
    display.text('n',40,0,1)
    moveLeft(pixelDistance)
    display.text(' ',40,0,1)
    moveLeft(pixelDistance)
```

qu'une seule broche d'E/S de l'ESP32. En outre, une bibliothèque Python prête à l'emploi est disponible pour cette série de capteurs. En plus du capteur lui-même, il faut juste une résistance de rappel de 4,7 kΩ. En utilisant le rappel interne de l'ESP32, il est même possible de l'omettre. Le capteur présente les caractéristiques suivantes :

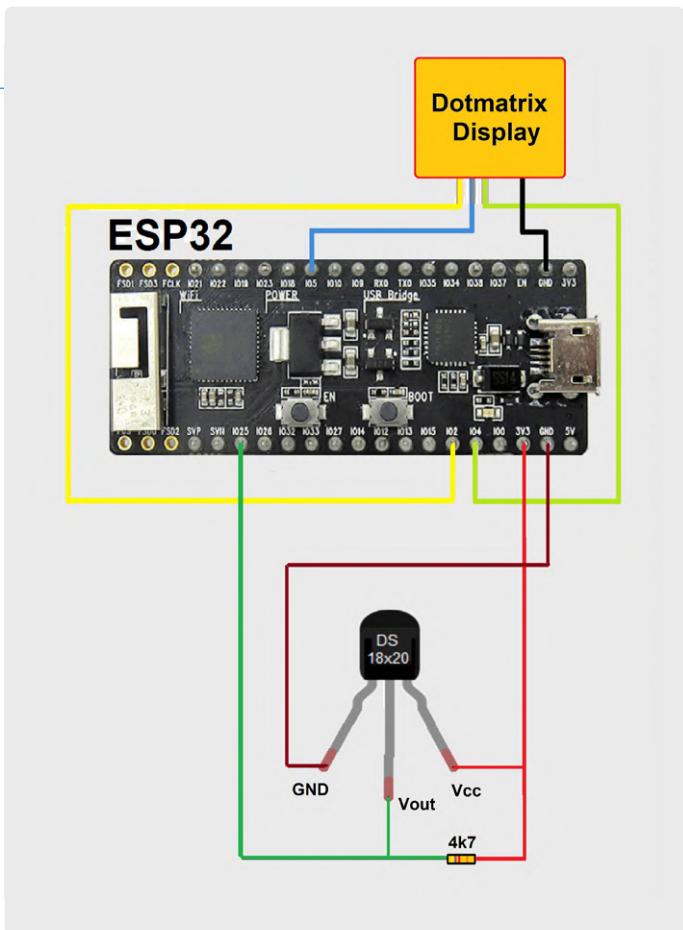


Figure 7. Capteur thermique DS18x20 relié au contrôleur ESP32.

- Tension d'alimentation : 3,0 V à 5,5 V
- Plage de température : -55 °C à +125 °C
- Précision de mesure : $\pm 0,5$ °C (-10 °C à +85 °C)
- Résolution : 9 bits, correspondant à environ 1/10 °C
- Durée de la période de mesure : 750 ms (max.)

Une broche connectée au bus 1-Wire permet de récupérer les mesures de plusieurs capteurs. En effet le protocole spécial du bus 1-Wire permet d'interroger en parallèle presque n'importe quel nombre de capteurs de température via une seule broche du contrôleur. Cependant, nous nous contentons ici d'un seul capteur. La **figure 7** montre la connexion du DS18x20 à l'ESP32. Un programme d'évaluation pour le DS18x20 affiche les valeurs de température enregistrées sur la console :

DS18x20_TempSens_demo.py

```
from machine import pin
import onewire
import ds18x20
import time

ow = onewire.OneWire(Pin(25)) # initialise le bus one
wire
ow.scan()
ds=ds18x20.DS18X20(ow) # crée l'objet ds18x20

while True:
    units=ds.scan() # recherche les unités ds18x20
    ds.convert_temp() # convertit la température
    for unit in units:
        print(ds.read_temp(unit)) # affichage
        time.sleep(1)
        print()
```

Le micrologiciel MicroPython contient par défaut les modules de lecture du capteur.

L'extension du programme à l'écran matriciel LED se fait en quelques lignes, voir le **listage 3**. La **figure 8** montre le résultat. On peut trouver dans le livre [2] d'autres détails sur la connexion de divers capteurs d'intensité lumineuse, d'humidité ou de champs magnétiques, entre autres.

Graphiques animés

Outre les lettres et les données numériques, des graphiques et même des animations peuvent être présentés sur la matrice de LED. Le programme du **listage 4** fait apparaître n'importe quel graphique en pixels sur l'écran.

Le graphique est défini dans l'image matricielle **icon**. Les points à allumer sont marqués d'un « 1 », les points sombres d'un « 0 » :

```
[0, 0, 0, 0, 0, 0, 0, 0, 0],
[0, 1, 1, 0, 0, 1, 1, 0],
[0, 1, 1, 0, 0, 1, 1, 0],
[0, 0, 0, 0, 0, 0, 0, 0],
[0, 0, 0, 1, 1, 0, 0, 0],
[0, 0, 0, 0, 0, 0, 0, 0],
[0, 1, 0, 0, 0, 0, 1, 0],
[0, 0, 1, 1, 1, 1, 0, 0],
```

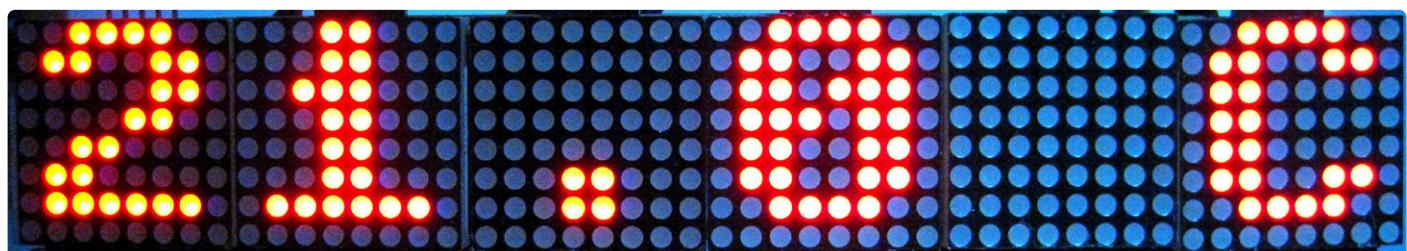


Figure 8. Un méga-afficheur de température.



Listage 3. Affichage de la température.

```
# DS18x20_to_LED_matrix.py

import max7219
from machine import Pin, SPI
import onewire
import ds18x20
import time

# pin assignment for LED matrix
spi = SPI(1, baudrate=10000000, polarity=1, phase=0, sck=Pin(4), mosi=Pin(2))
ss = Pin(5, Pin.OUT)

ow = onewire.OneWire(Pin(25)) # initialise le bus 1-Wire
ow.scan() # recherche les unités ds18x20

ds=ds18x20.DS18X20(ow)

while True:
    units=ds.scan() # recherche les unités ds18x20
    ds.convert_temp() # convertit la température
    for unit in units:
        T=ds.read_temp(unit) # crée la chaîne formatée
        output="T = {:.1f} °C" # crée la chaîne formatée pour la matrice de LED
        print(output.format(T)) # écrit la température sur la matrice de LED
        output=str(T)
        output="{:4.1f} °C"
        display = max7219.Matrix8x8(spi, ss, 6)
        display.text(output.format(T),0,0,1)
        display.show()
    time.sleep(1)
```

De cette façon, n'importe quelle icône peut être créée. La fonction `enumerate` convertit le bitmap en un graphique affichable :

```
for y, row in enumerate(icon):
    for x, c in enumerate(row):
        display.pixel(x, y, c)
```

On obtient ainsi la **figure 9** sur la matrice de LED. On peut alors utiliser la fonction de défilement de la boucle principale pour déplacer sur l'afficheur l'image créée ainsi (voir également la vidéo YouTube [3]).

Résumé et perspectives

Après avoir présenté les commandes élémentaires dans le premier article de la série [1], cette deuxième partie permet de les mettre en œuvre dans diverses applications pratiques. De puissantes bibliothèques permettent de réaliser des projets impressionnantes avec seulement quelques lignes de code. Cela confirme la potentiel remarquable de MicroPython également pour les applications de contrôleurs. Vous trouverez de plus amples informations et de nombreux projets pratiques dans le livre *MicroPython for Microcontrollers* [2]. La commande de servomoteurs, la transmission de données sans fil par RFID, le protocole MQTT et la transmission de mesures de capteurs sur l'internet y sont également traités en détail.

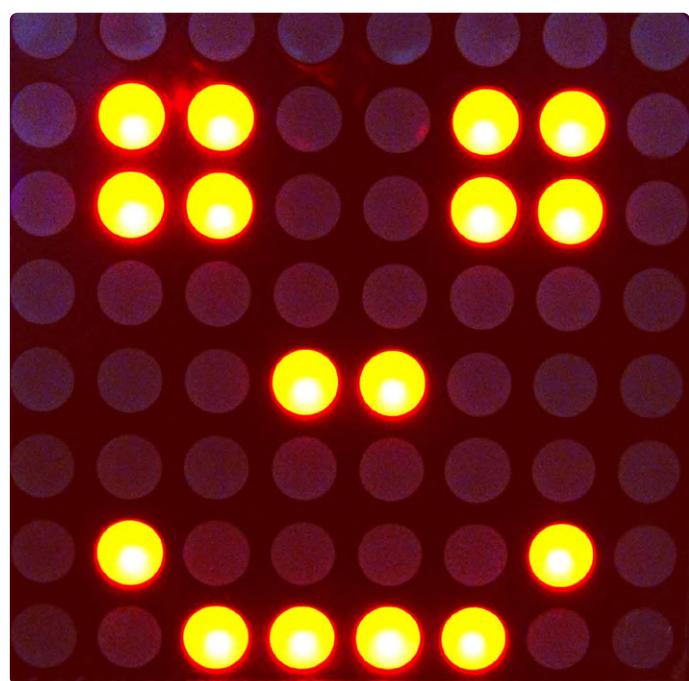


Figure 9. Graphique avec des pixels, fait main, sur l'écran matriciel.



Listage 4. Graphique en pixels.

```
#LED_matrix_icon.py

import max7219
from machine import Pin, SPI
from time import sleep

# sck -> serial clock - CLK; mosi -> master out slave in - DIN
# ss -> chip select - CS
spi=SPI(1,baudrate=10000000,polarity=1,phase=0,sck=Pin(4),mosi=Pin(2))
ss=Pin(5,Pin.OUT)

display=max7219.Matrix8x8(spi,ss,6)
display.brightness(0)

icon = [
    [0, 0, 0, 0, 0, 0, 0, 0],
    [0, 1, 1, 0, 0, 1, 1, 0],
    [0, 1, 1, 0, 0, 1, 1, 0],
    [0, 0, 0, 0, 0, 0, 0, 0],
    [0, 0, 0, 1, 1, 0, 0, 0],
    [0, 0, 0, 0, 0, 0, 0, 0],
    [0, 1, 0, 0, 0, 0, 1, 0],
    [0, 0, 1, 1, 1, 1, 0, 0],
]

for y, row in enumerate(icon):
    for x, c in enumerate(row):
        display.pixel(x, y, c)

display.show()

while True:
    for i in range(40):      # déplace le carré de gauche à droite
        display.show()
        display.scroll(1, 0)
        sleep(0.05)

    for i in range(40):      # déplace le carré de droite à gauche
        display.show()
        display.scroll(-1, 0)
        sleep(0.05)
```

Contributeurs

Texte et illustrations : **Günter Spanner**
Rédaction : **Jens Nickel, C.J. Abate**

Mise en page : **Harmen Heida**
Traduction : **Denis Lafourcade**

En outre, la combinaison de Python avec le domaine en pleine expansion de l'apprentissage automatique et de l'intelligence artificielle permettra, dans un avenir proche, des applications qui étaient auparavant impensables. Les nouvelles générations de contrôleurs, comme le Kendryte K210, permettent déjà l'évaluation directe des signaux audio et vidéo. Avec les algorithmes de traitement d'images basés sur Python, un large éventail d'options tournées vers l'avenir s'ouvre aux développeurs. 

(210179-B-04)

Des questions, des commentaires ?
Contactez Elektor
(redaction@elektor.fr).



PRODUITS

- **Livre en anglais « MicroPython for microcontrollers »**
www.elektor.fr/micropython-for-microcontrollers
- **ESP32-PICO-KIT V4**
www.elektor.fr/esp32-pico-kit-v4
- **Platine d'expérimentation**
www.elektor.fr/breadboard-830-tie-points

LIENS

- [1] **Günter Spanner, « MicroPython pour l'ESP32 et ses copains », Elektor 07-08/2021 :** www.elektormagazine.fr/200179-04
- [2] **Livre en anglais de G. Spanner, *MicroPython for Microcontrollers*, Elektor 2020 :** www.elektor.fr/micropython-for-microcontrollers
- [3] **Vidéo de démonstration sur YouTube :** <http://www.youtube.com/watch?v=5uligjyKMEk>
- [4] **Téléchargement du logiciel :** <http://www.elektormagazine.fr/210179-B-04>

carte MadMachine SwiftIO

un langage récent pour
du matériel tout nouveau

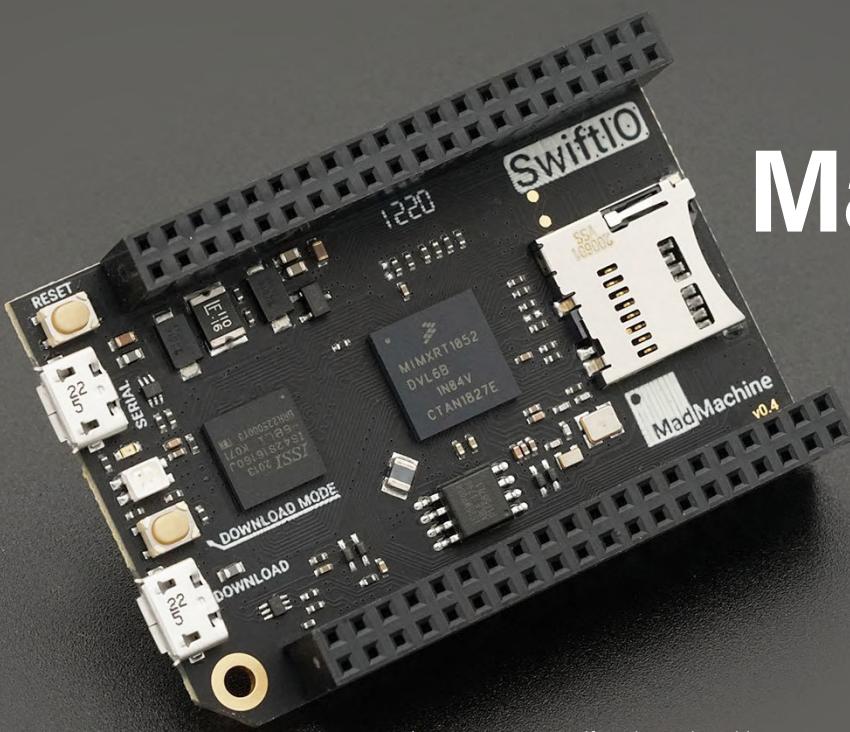


Figure 1. La carte SwiftIO de MadMachine.



Mathias Claußen (Elektor)

Le langage de programmation Swift qui est bien établi sur les appareils Apple peut désormais être utilisé sur un microcontrôleur. Jetons un œil à la carte SwiftIO qui intègre un processeur ARM Cortex-M7 et 32 Mo de RAM.

MadMachine SwiftIO est une carte de développement au format BeagleBone Black. Ce qui la rend attrayante pour moi est le fait que nous avons un ensemble de matériel intéressant combiné à un cadre logiciel peu commun. Le code peut être écrit en Swift, un langage de programmation bien établi sur les appareils Apple.

Le matériel

Alors que le langage Swift est destiné au matériel Apple, la carte SwiftIO (fig. 1) de MadMachine introduit Swift dans le monde de l'embarqué.

Les principaux composants utilisés sur la carte sont :

- Processeur NXP i.MX RT1052 avec un cœur ARM Cortex-M7 cadencé à 600 MHz
- Flash QSPI externe de 8 Mo
- 32 Mo de SDRAM
- Connecteur pour carte micro-SD, supportant les cartes standard et haute capacité

Avec le i.MX RT1052, la carte est animée par un microcontrôleur robuste, dont les caractéristiques sont détaillées sur la figure 2. Elle offre en plus :

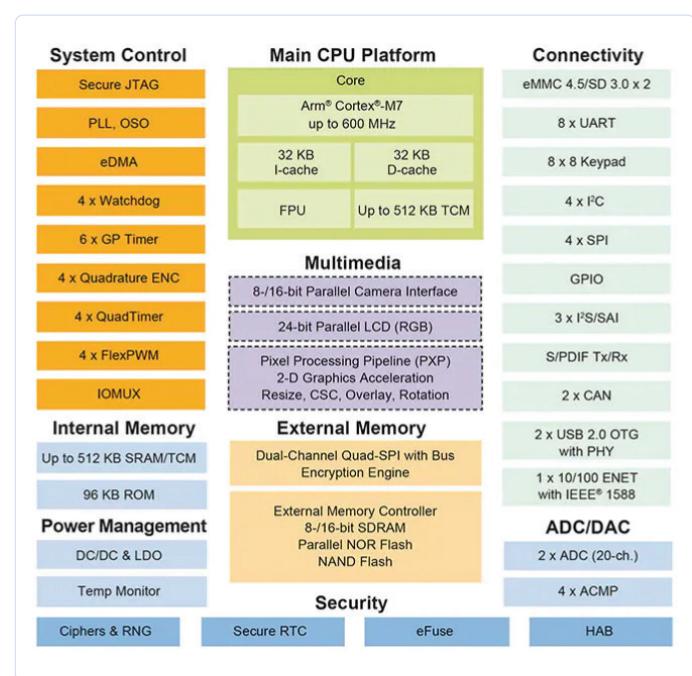


Figure 2. Vue d'ensemble du i.MX RT1052 (source : NXP).

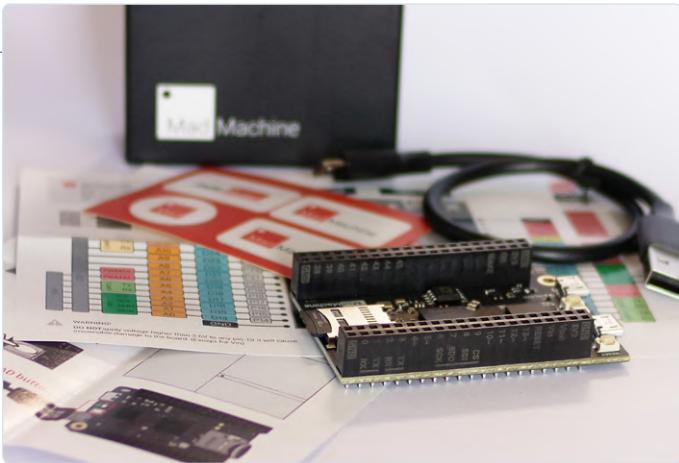


Figure 3. Contenu de la boîte.

```

1 // Change the LED state every second by setting the interrupt.
2
3 // Import the library to enable the relevant classes and functions.
4 import SwiftIO
5
6 // Initialize the red onboard LED and a timer to set interrupt.
7 let red = DigitalOut(Id.RED)
8 let timer = Timer()
9
10 // Raise the interrupt to turn on or off the LED every second.
11 timer.setInterrupt(ms: 1000) {
12     red.toggle()
13 }
14

```

Memory region	Used Size	Region size	Page Count
CODEM	0 GB	128 KB	0.00%
DTCM	0 GB	128 KB	0.00%
ITCM	0 GB	128 KB	0.00%
FLASH	0 GB	8 MB	0.00%
SRAM	305408 B	32752 KB	9.10%
IDT List	0 GB	2 KB	0.00%

Figure 4. L'EDI de MadMachine.

- 46 GPIO (E/S à usage général)
- 12 canaux A/N (12 bits)
- 14 canaux MLI
- 4 UART
- 2 bus CAN 2.0B
- 2 bus I²C (3,4 MHz max)
- Moteur graphique 2D
- Sortie vidéo RVB (1366 × 768 WXGA max)
- Entrée caméra (CSI 8, 16, 24 bits)

Il ne s'agit pas d'une liste exhaustive des caractéristiques, mais elle donne une idée du matériel. Même l'emballage est compact. Vous recevez une carte, un câble USB, un jeu d'autocollants et la carte SwiftIO avec une carte SD insérée (fig. 3). Tout ce dont vous avez besoin pour commencer.

L'EDI

En ce qui concerne le développement de logiciels, MadMachine propose un EDI pour SwiftIO, téléchargeable sur son site Web (voir à quoi il ressemble sur la figure 4). Il est disponible pour Windows et macOS, ce qui est une bonne nouvelle pour les utilisateurs Apple. Si vous n'aimez pas l'EDI, vous pouvez utiliser les outils ILC (interface en ligne de commande) fournis pour cette carte ou les associer à XCode ou Visual Studio Code pour votre prochain projet. MadMachines fournit un tutoriel sur la façon de les utiliser.

Il suffit de copier

La carte comprend dans sa mémoire flash SPI un chargeur d'amorçage qui copiera votre application dans la SDRAM au démarrage puis l'exécutera. Cela facilite la reprogrammation de la carte. Le programmeur intégré permet d'écrire votre code sur la carte micro-SD en l'affichant comme périphérique de stockage de masse. De plus, vous pouvez simplement retirer la carte et écrire votre nouveau micrologiciel directement sur celle-ci.

SwiftIO

Ce qui a retenu mon attention, c'est le logiciel choisi. Le cadriel SwiftIO (fig. 5) utilise le système d'exploitation en temps réel (RTOS) Zephyr. L'accès au matériel de bas niveau, comme les GPIO, les fonctions analogiques, la MLI, l'I²C et ainsi de suite, passe par une couche d'abstraction, il devrait donc être facile d'écrire son propre logiciel sans avoir à s'occuper de tous les détails du microcontrôleur

i.MX RT1052. De plus, cela concerne également l'accès à la SDRAM et son initialisation ainsi que la configuration de la flash SPI. Quant à Swift, le langage de programmation retenu pour cette carte, il est couramment utilisé sur les systèmes Apple et on le retrouve principalement dans l'écosystème Apple. Il n'est pas limité à ce monde et convient pour développer des applications pour Linux et, depuis peu, pour Windows. Comme Swift offre des éléments et des concepts modernes, le fait qu'il soit disponible sur un microcontrôleur permet d'approfondir ses connaissances Swift. Si vous vous demandez à quoi ressemble un programme SwiftIO, le **listage 1** montre comment faire clignoter une LED. Le **listage 2** montre comment utiliser la MLI pour commander sa luminosité. Des exemples plus avancés sont déjà inclus, afin que vous puissiez commencer à explorer SwiftIO.

Assistance et informations supplémentaires

Pour cette carte, vous trouverez facilement de l'aide sur le serveur Discord de MadMachine. Posez vos questions (même les plus techniques) dans les chats appropriés dans un langage amical et poli ; personne ne veut de trolls. De plus, s'il vous manque des documents, n'hésitez pas à les demander.

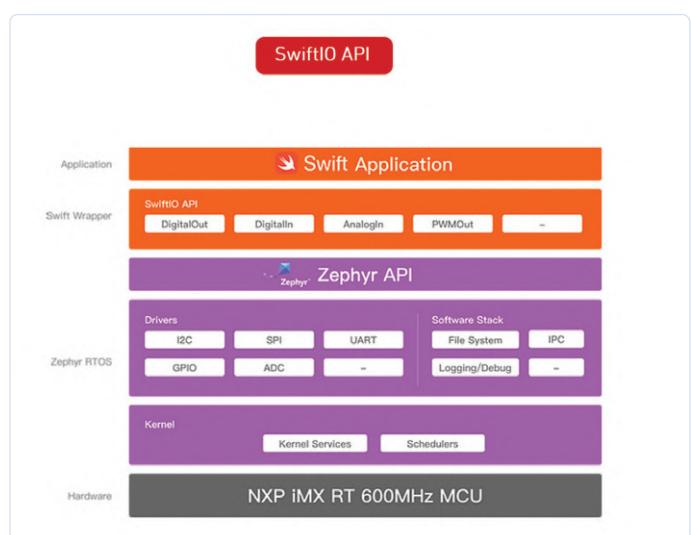


Figure 5. Pile logicielle (source : MadMachine.io).

Listage 1. Faire clignoter une LED.

```
/// Changer l'état d'une LED toutes les secondes avec une interruption.  
/// Importer la bibliothèque pour accéder aux classes et fonctions nécessaires.  
import SwiftIO  
/// Initialiser la LED rouge et la minuterie pour déclencher l'interruption.  
let red = DigitalOut(Id.RED)  
let timer = Timer()  
/// Déclenchez l'interruption pour allumer ou éteindre la LED toutes les secondes.  
timer.setInterrupt(ms: 1000) {  
    red.toggle()  
}  
  
while true {
```

Dernières réflexions

Si vous cherchez une carte compatible Arduino ou qui peut être programmée en C/C++, oubliez SwiftIO. En outre, son prix est plus élevé que celui de la plupart des autres cartes de développement. Toutefois, si vous considérez que vous disposez d'une assistance technique et d'un langage de programmation moderne, et que le développement est possible sur les principaux systèmes d'exploitation, c'est peut-être une option. ↗

(210297-04)

Listage 2. Avec MLI.

```
/// Faire varier l'intensité d'une LED en modifiant le rapport cyclique du signal MLI.  
/// Importer la bibliothèque pour accéder aux classes et fonctions nécessaires.  
import SwiftIO  
/// Initialiser la sortie MLI connectée à la LED.  
let led = PWMOut(Id.PWM0A)  
/// Initialiser une variable pour stocker le rapport cyclique.  
var value: Float = 0.0  
/// Faire varier la luminosité du min au max, et inversement.  
while true {  
    // Allumer la LED sur une durée de deux secondes.  
    while value <= 1.0 {  
        led.setDutyCycle(value)  
        sleep(ms: 20)  
        value += 0.01  
    }  
    // Garder le rapport cyclique entre 0,0 et 1,0.  
    value = 1.0  
    // Eteindre la LED sur une durée de deux secondes.  
    while value >= 0 {  
        led.setDutyCycle(value)  
        sleep(ms: 20)  
        value -= 0.01  
    }  
    // Garder le rapport cyclique entre 0,0 et 1,0.  
    value = 0.0
```

Contributeurs

Conception et texte :
Mathias Claußen

Rédaction : **Jens Nickel**

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

- **SwiftIO – carte à microcontrôleur, compatible avec le langage Swift**
www.elektor.fr/swiftio-swift-based-microcontroller-board

LIENS

- [1] **MadMachine** : <https://www.madmachine.io/>
- [2] **Guide pour l'interface en ligne de commande, MadMachine** :
<https://resources.madmachine.io/about-our-project#how-does-the-building-procedure-work>
- [3] **Serveur Discord MadMachine** : <https://discord.gg/zZ9bFHK>
- [4] **Ressources de MadMachine** : <https://resources.madmachine.io/>

sur le vif

Passions incertaines

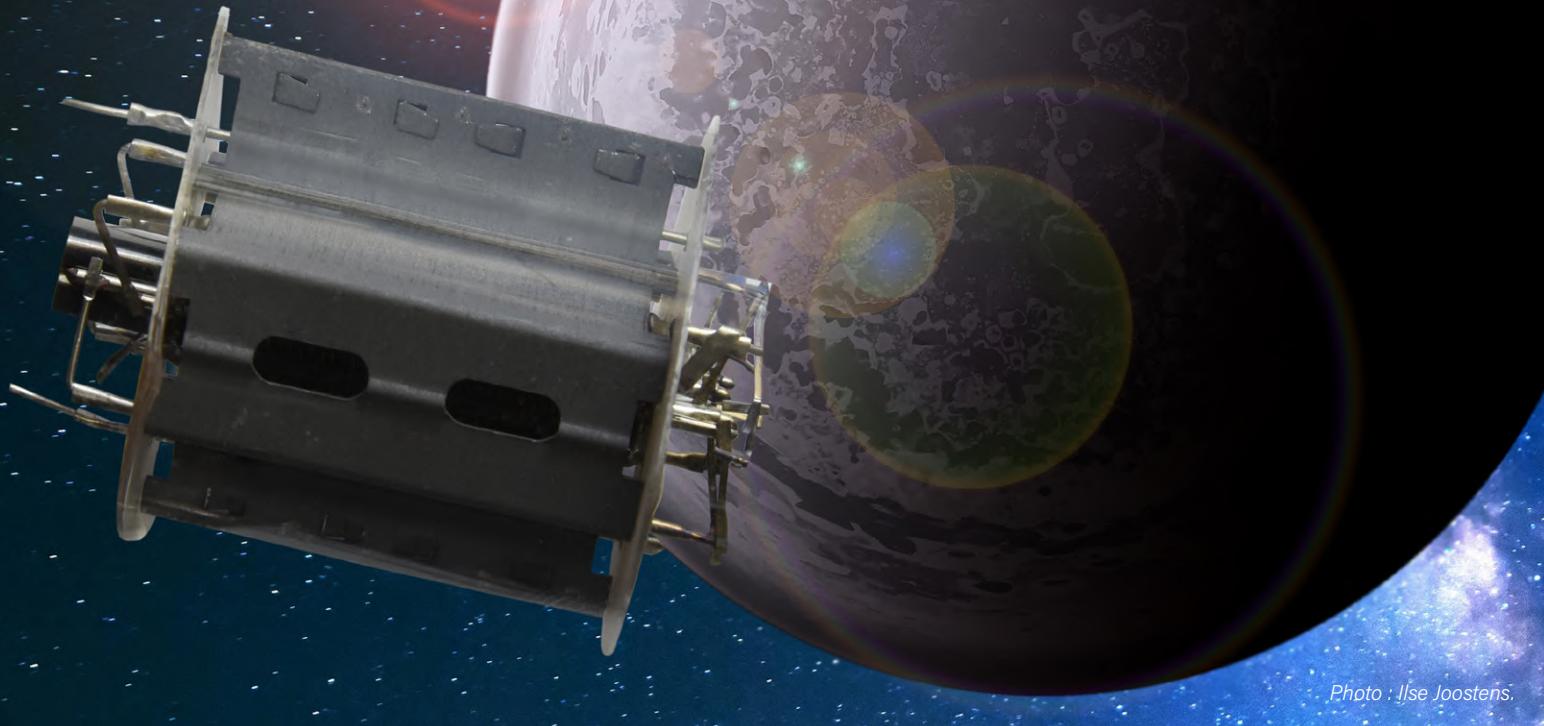


Photo : Ilse Joostens.

Ilse Joostens (Belgique)

Dans les années 1970, mes parents possédaient un téléviseur noir et blanc à tube cathodique de la marque ACEC [1]. En rentrant de l'école, j'ai un jour trouvé mon père penché derrière le poste, visiblement aussi absorbé qu'un mécanicien sur un moteur. L'arrière de l'appareil avait été retiré, et plusieurs tubes reposaient sur une petite table. Mon père s'est redressé, a rallumé le poste, et je me souviens encore de la mystérieuse lueur dont furent soudain baignés les tubes. Je me suis alors sentie pleine d'admiration pour ce père si intelligent. Ce que j'ignorais, c'est qu'en fait il n'y connaissait rien du tout en téléviseur, qu'il s'était contenté de suivre les instructions laissées par mon oncle, un monteur d'ascenseurs. Les tubes de rechange aussi m'intriguaient. À mes yeux d'enfant, leur verre semblait dissimuler des petits satellites ou quelque vaisseau spatial.

Fureurs et crépitements

Mon enfance à la campagne a été traversée par diverses passions, dont le point noué au crochet, l'identification des plantes sauvages et les expériences avec des lampes et des piles. La science me passionnait, j'ai d'ailleurs dévoré plus d'un titre de la série *How and Why Wonder Books*. Un jour, mon oncle, le monteur d'ascenseurs, m'a offert un carton rempli d'objets fascinants, dont des récepteurs téléphoniques à l'ancienne, ceux avec un microphone à charbon. Comme tout enfant espiègle, je leur ai aussitôt trouvé un usage bien plus intéressant que l'original : terroriser les braves gens. J'avais découvert que relier le haut-parleur d'un combiné à une pile produisait un crépitement terrible, ce qui me permettait, aidée d'une bonne longueur de fil, de faire sursauter et d'inquiéter plus d'une oreille. En un lieu et une époque où les gens de la campagne étaient surtout habitués aux bruits de la nature, c'était évidemment plus facile.

Derrière notre jardin s'étalaient des prairies entourées de clôtures électriques. L'appel de cette électricité était irrésistible. J'adorais y relier toutes sortes de choses, comme ce VU-mètre récupéré dans un vieux magnétophone et dont j'observais avec intérêt l'aiguille sursauter au rythme des impulsions électriques parcourant le fil. Hélas, visiblement peu soucieux de l'éveil des enfants, un paysan furieux ne manquait jamais d'interrompre ma formation précoce par quelques coups de sonnette impérieux à notre domicile.

La cornue prend le relais

À la fin de ma petite scolarité, nous avons déménagé dans une petite ville de province, paisible mais plutôt morne. Dans une des rues que j'empruntais pour me rendre au collège se succédaient un magasin de jouets et un magasin de composants électroniques. Je connaissais le magasin de jouets pour y être déjà rentrée, mais un jour c'est celui d'électronique qui m'attira : les kits qui étaient exposés en vitrine me semblaient soudain aussi séduisants que des bonbons. Je suis entrée et ai acheté un kit d'interrupteur crépusculaire. Il comprenait une photorésistance, quelques transistors et composants passifs, ainsi qu'un relais. Je l'ai assemblé et testé avec une lampe torche. Le relais se déclenchaient comme attendu, ce qui sur le moment me rendit fière comme un paon. Mission accomplie ! D'autres kits ont suivi, assemblés avec plus ou de moins de succès, mais il faut dire que je me contentais de suivre les instructions de montage sans vraiment comprendre ce que je faisais. Un jour j'ai réussi à me procurer une boîte de Gakken EX-System, un kit éducatif japonais permettant toutes sortes d'expérimentations par combinaison de « blocs denshi », ou « blocs électroniques » [2][3]. De cette boîte sont sortis d'innombrables bruits étranges, et même des émissions radio insouciantement diffusées sur des ondes moyennes « interdites ».



Le Gakken EX-System et ses blocs denshi.

Durant mes années de lycée, mes intérêts se sont déplacés vers la chimie. J'avais même installé un petit laboratoire chez nous, une initiative a priori louable, mais sans doute quelque peu irresponsable si l'on considère qu'à un moment donné il contenait assez de substances toxiques pour décimer notre bourgade et ses alentours. Cela dit je n'ai jamais provoqué d'explosion. La pire de mes mésaventures fut une fontaine de mauvais vin rouge bouillonnant qu'un bouchon de distillateur mal fermé fit soudain jaillir jusqu'au plafond. J'avais besoin d'éthanol pur, mais comme il s'agissait d'un produit cher, j'eus l'idée, enivrante selon moi, de distiller du vin pour en récupérer l'éthanol.



Photo : Shutterstock

d'autre choix que d'intégrer une formation en électromécanique. Mon diplôme en poche, je ne trouvais aucun poste stimulant qui eut pu combler mes aspirations. J'en décrochai finalement un dans le secteur des TIC, autrement dit reléguais une fois de plus l'électronique à l'arrière-plan.

Je commençai comme analyste-programmeuse, puis devins administrateur système. Semée d'embûches, trop stressante pour mon tempérament, la vie d'administrateur système finit par ne plus me convenir. Et c'est ainsi qu'il y a maintenant dix ans de cela, je devins conceptrice indépendante en électronique. Ce renouveau m'a permis de publier plusieurs projets sympas dans ces pages, mais avec la concurrence accrue de l'Asie et l'engouement généralisé pour tout ce qui n'est pas cher, mon amour pour l'électronique n'est plus du tout ce qu'il était. La chimie a de nouveau mes faveurs, mais par le biais d'anciens procédés photographiques [4][5] et de l'ébonisation du bois à l'acétate de fer. Au lycée, je détestais l'art et l'histoire, et voilà que depuis quelques années ces deux disciplines me passionnent. Je suis plus vieille maintenant, je vieillirai encore, et les ans sur leur passage charrient bien des changements. Alors peut-être un jour retrouverai-je la passion de l'électronique. Qui sait. ↗

Contributeurs

Texte : Ilse Joostens

Rédaction : Eric Bogers

Traduction : Hervé Moreau

Maquette : Harmen Heida

Des questions, des commentaires ?

Contactez Elektor (redaction@elektor.fr).

(210341-04)

LIENS

- [1] ACEC : https://www.radiocollection.be/fr/acec1968_fr.html
- [2] Gakken EX-System : <https://www.petervis.com/electronics-lab/gakken-ex-system/gakken-ex-system.html>
- [3] Gakken EX-System : https://en.wikipedia.org/wiki/Gakken_EX-System
- [4] Cyanotype : <https://fr.wikipedia.org/wiki/Cyanotype>
- [5] Anciens procédés photographiques : <http://www.alternativephotography.com/a-dash-of-salt/>

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.com.
Seule exigence :
jamais cher, toujours surprenant !

Velleman VTSS230 – 2 en 1 station à air chaud pour CMS

Prix : 82,50 €

 www.elektor.fr/19833



Envox EEZ Bench Box 3 (BB3) 4 voies – Solutions de Mesures et de Test Modulaire



Prix : 899,00 €

Prix (membres) : 809,10 €

 www.elektor.fr/19825



M5Stack UnitV2 Caméra IA
pour l'Edge Computing

Prix : 79,95 €

Prix (membres) : 71,96 €

www.elektor.fr/19782

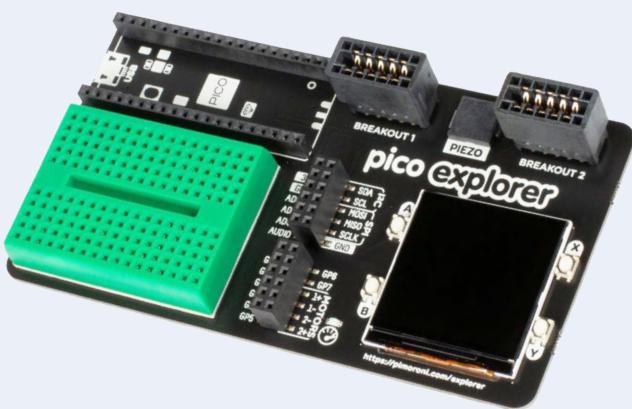


Phiz Scanner Laser 3D
pour Smartphone

Prix : 349,00 €

Prix (membres) : 314,10 €

www.elektor.fr/19708



Pimoroni Raspberry Pi Pico
Explorer Base

Prix : 29,95 €

Prix (membres) : 26,96 €

www.elektor.fr/19767



Prix : 24,95 €

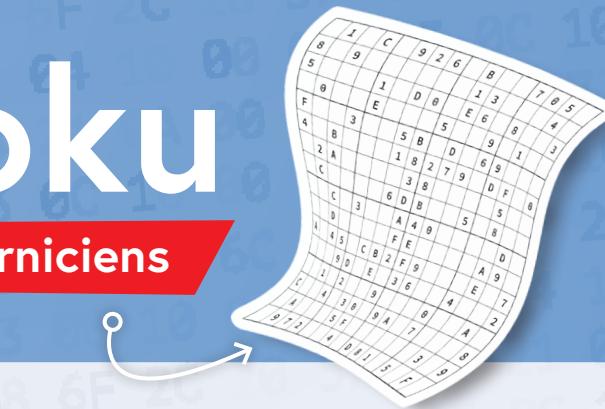
Prix (membres) : 22,46 €

www.elektor.fr/19709

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 octobre 2021** à l'adresse **hexadoku@elektor.fr**

LES GAGNANTS

La solution de la grille du numéro de juillet/août 2021 est **ADFB3**.

La liste des gagnants est publiée ici : www.elektormagazine.fr/hexadoku

Bravo à tous les participants et félicitations aux gagnants !

A		5			6	4			1			8
D		7	4		2	3		5	6			B
	C	B					F	A		E		
E	0	C		8	1	6		4		D		
2	0	E	3	B	C		A	1	6	D	4	7
5	4			F		0			3	1		
6	9	B	2				C	A	8		0	
		9	4				7	8				
		C	9				8	E				
E	5	F	D				A	8	B		4	
1	B			8		6			9	A		
7	6	8	C	1	E		9	D	3	F	2	5
B	2		E	F	C	5	9		8			
0	D	6				4	3		2			
8	A	1		C	D		2	4		6		
5		7		4	A		C		3			

C	4	8	F	2	E	0	3	D	6	B	1	7	A	5	9
5	3	9	D	F	1	7	8	0	C	A	E	4	6	2	B
6	A	0	1	4	D	9	B	5	7	2	8	C	E	3	F
B	2	E	7	A	5	6	C	3	4	9	F	8	0	D	1
D	B	2	5	0	9	1	6	4	F	3	7	E	C	8	A
E	F	6	3	B	8	2	4	C	A	0	5	9	D	1	7
8	9	4	C	3	7	A	E	1	B	D	6	2	F	0	5
7	0	1	A	5	C	F	D	E	2	8	9	B	3	4	6
F	D	3	9	C	6	B	5	2	8	1	A	0	4	7	E
2	6	7	0	8	A	D	F	B	3	E	4	5	1	9	C
A	5	B	4	7	2	E	1	F	9	C	0	D	8	6	3
1	8	C	E	9	3	4	0	6	5	7	D	A	B	F	2
0	C	F	6	E	B	5	9	7	D	4	3	1	2	A	8
9	E	A	2	1	4	3	7	8	0	6	C	F	5	B	D
3	1	5	B	D	0	8	A	9	E	F	2	6	7	C	4
4	7	D	8	6	F	C	2	A	1	5	B	3	9	E	0

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.

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

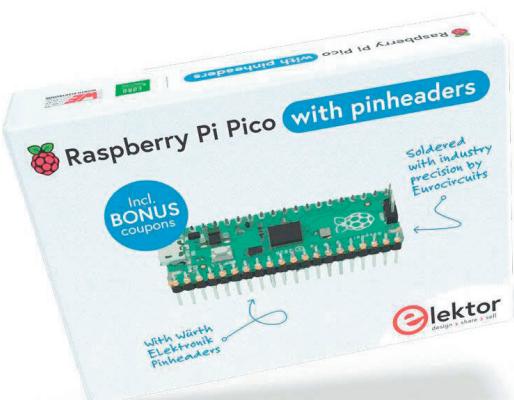
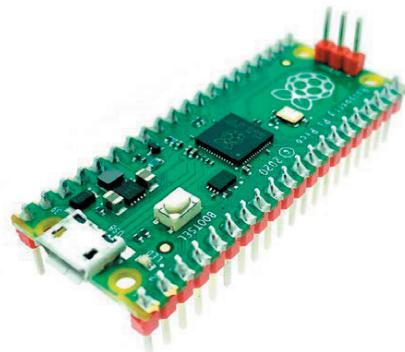


www.elektormagazine.fr/membres



ABONNEZ-VOUS ET RECEVEZ

Raspberry Pi + Headers GRATUIT



**TOUS LES 2 MOIS, LES
DERNIÈRES NOUVELLES DU
RASPBERRY PI ET LES
MEILLEURS PROJETS !**



**Souscrivez dès maintenant
un abonnement d'un an
au magazine MagPi, nous
vous offrons :**

- Six numéros du magazine MagPi
- Une carte Raspberry avec headers

Vos avantages :

- 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