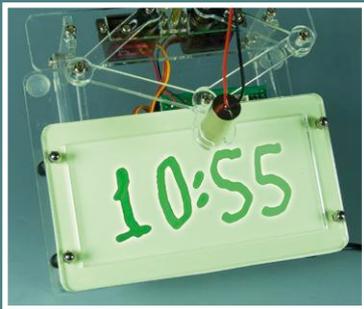
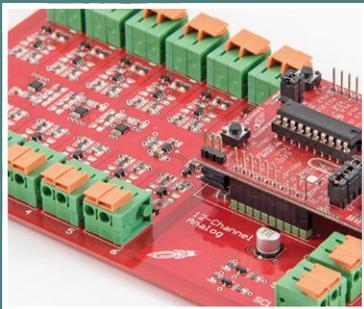


# elektor

DÉCOUVRIR > CRÉER > PARTAGER



**Laser Time Writer**  
horloge à laser



**acquisition de mesures par USB**  
en quête du chauffage optimal



**lecteur Card Sound**  
l'audio à la carte SD

## Bio-lampe

les « émotions » d'une plante en couleur



**Dans cette édition :**  
8 projets de labo  
6 projets de lecteur  
EEBUS, ESP8266, laser,  
OLED, RPi, USB,  
Wi-Fi...

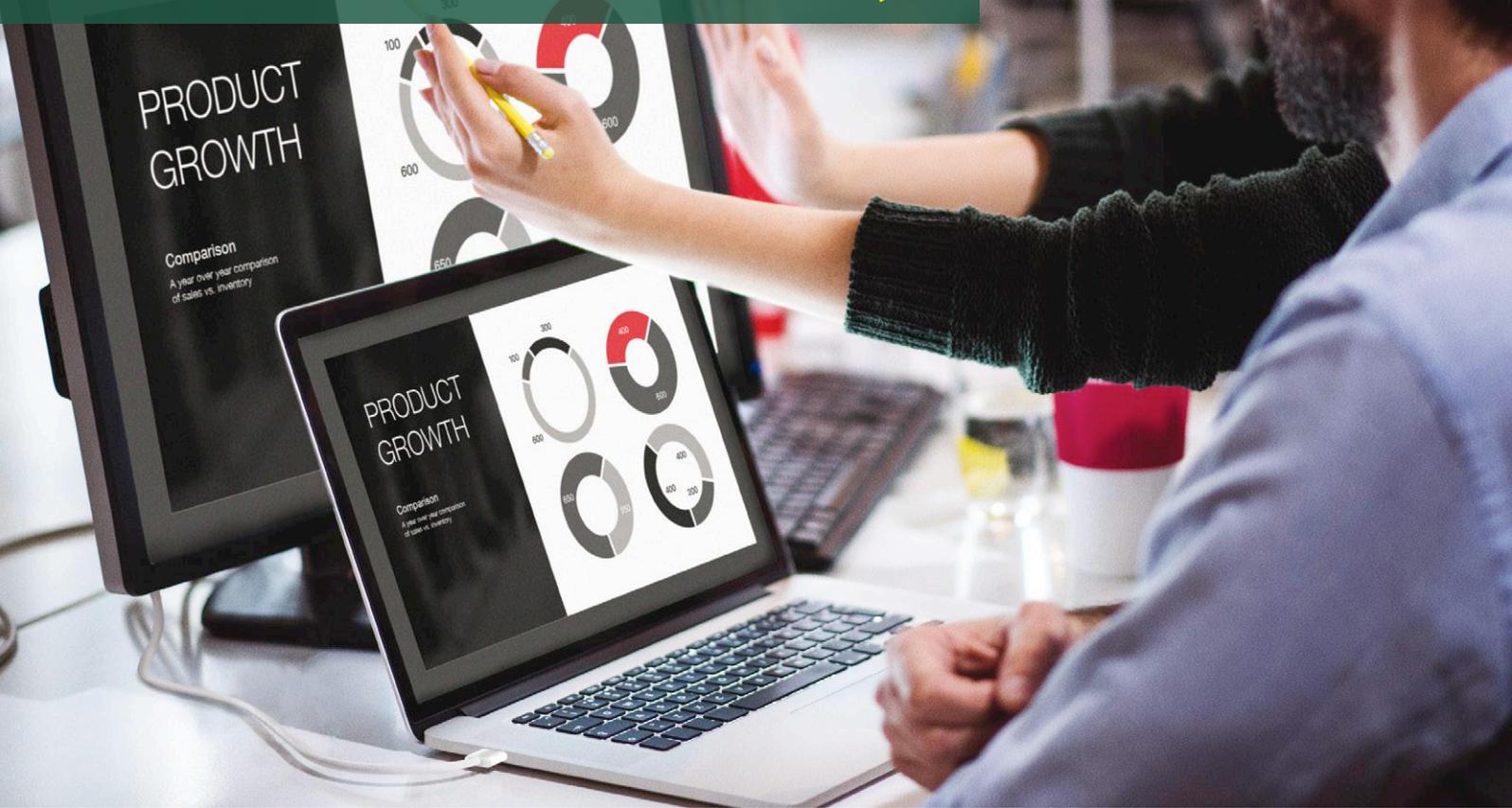
**adaptateur de programmation USB pour ESP8266** ●

Q & R : les afficheurs ● **alimentation solaire pour répéteur Wi-Fi** ● Rétronique  
- UHER 4000 Report-L ● **micro-bobine Tesla & synthétiseur MIDI** ● libérez le  
spectre radio ! ● **station météo RPi Zero W** ● afficheurs OLED ● **thermostat de  
bureau wifi** ● banc d'essai : compteur d'énergie SmartPi ● **shield d'expérimentation  
Arduino version 2.0** ● couteau suisse pour µC : PlatformIO ● **DoubleSPIDER** ●  
R. Lacoste : comment consommer moins ● **Easy Parking System** ● l'EEBUS arrive



# Optimisez vos solutions USB Power Delivery

## UPD360 – Contrôleur USB Power Delivery



Le contrôleur UPD360 Power Delivery (PD) de Microchip est un contrôleur USB Type-C™/PD certifié USB-IF. Le contrôleur UPD360 intègre les blocs de fonction requis pour l'USB Type-C et les communications PD, qui incluent des FET  $V_{CONN}$  et des contrôleurs d'alimentation de ports. Le contrôleur UPD360 peut fonctionner de manière autonome ou sous forme de complément d'un microcontrôleur, d'un contrôleur embarqué ou d'un hub USB, connectés via une interface I<sup>2</sup>C/SPI. Le UPD360 peut être intégré à des applications requérant la connectivité USB, alterne plusieurs protocoles (par ex. Display Port) et gère l'alimentation (en tant que source ou sink) jusqu'à 100 W via des connecteurs USB Type-C.

### Fonctions clés

- ▶ USB Type-C et fonction de distribution de l'alimentation
- ▶ Commutateur d'alimentation intégré
- ▶ FET  $V_{CONN}$  intégrés
- ▶ Compatible avec les batteries déchargées
- ▶ Interface I<sup>2</sup>C/SPI



**microchip**  
**DIRECT**  
[www.microchipdirect.com](http://www.microchipdirect.com)

**MICROCHIP**

[www.microchip.com/UPD360](http://www.microchip.com/UPD360)

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

@ : [service@elektor.fr](mailto:service@elektor.fr)

Téléphone : (+33) 01 70 61 27 41  
du lundi au vendredi de 10h à 13h

Télécopie : (+33) 01 70 61 27 42

[www.elektor.fr](http://www.elektor.fr) | [www.elektormagazine.fr](http://www.elektormagazine.fr)

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

#### Publicité :

Margriet Debeij  
Tél. : +49 (0)241 955 09 174  
[margriet.debeij@eimworld.com](mailto:margriet.debeij@eimworld.com)

#### DROITS D'AUTEUR :

© 2018 Elektor International Media B.V.

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

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

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



## Neurobiologie végétale ?

Au début du XX<sup>e</sup> siècle, le physicien bengali Jagadish Chandra Bose, pionnier de la radio, a étudié la croissance des plantes et leur réaction aux ondes électromagnétiques (la physiologie végétale). Beaucoup plus tard, Cleve Backster, inspiré par les travaux de Bose, s'est fait connaître par ses expériences sur des végétaux. Dans les années 60, ce spécialiste des interrogatoires à la CIA reliait à la feuille d'une plante les électrodes d'un polygraphe. Cet appareil utilisé pour la détection de mensonges faisait sans doute partie de sa panoplie. Les résultats des expériences de Backster, largement critiqués et rejetés par la communauté scientifique, font état de changements de résistance électrique lorsque la plante était endommagée ou risquait de l'être.

Plus récemment, le physicien français Joël Sternheimer a breveté la *généodique* : l'art de soigner les plantes par la musique. Les plantes seraient sensibles à leur environnement sonore. Des « mélodies de protéines » ou « protéodies », diffusées à proximité des plantes, permettraient de réguler le taux de synthèse des protéines, donc de favoriser la croissance et de renforcer les défenses naturelles. Que vous y croyiez ou qu'il s'agisse pour vous de fumisterie, vous vous demandez sans doute pourquoi je vous raconte tout ça...

C'est à cause du pot de fleurs en couverture du magazine. Elektor se lance-t-il dans l'horticulture ? Non, mais nous publions le schéma d'une « bio-lampe » qui convertit en couleurs les « émotions » d'une plante ! Ce projet a suscité bien des commentaires chez les rédacteurs et les traducteurs (l'un d'eux a même refusé de traduire cet article). Maintenant ce sont vos remarques que nous attendons. Si vous avez la main verte, envoyez-nous le résultat de vos observations.

*Bonne et heureuse année 2018 !*

Mariline Thiebaut-Brodier

### Notre équipe

Rédactrice en chef :	Mariline Thiebaut-Brodier ( <a href="mailto:redaction@elektor.fr">redaction@elektor.fr</a> )
Rédaction internationale :	Thijs Beckers, Jan Buiting, Jens Nickel
Laboratoire :	Ton Giesberts, Luc Lemmens, Clemens Valens (responsable), Jan Visser
Coordination :	Hedwig Hennekens
Ont coopéré à ce numéro :	Thierry Destinobles, Pascal Duchesnes, Robert Grignard, Denis Lafourcade, Jean-Louis Mehren, Denis Meyer, Hervé Moreau, Helmut Müller, Kévin Petit, Xavier Pfaff, Guy Raedersdorf
Service de la clientèle :	Cindy Tijssen
Graphistes :	Giel Dols, Mart Schroijsen, Patrick Wielders
Elektor en ligne :	Daniëlle Mertens

# Laser Time Writer

## horloge à laser



- 5 bientôt dans Elektor
- 10 agenda : janvier 2018
- 11 Elektor : guide de connexion(s)
- 28 agenda : février 2018
- 42 ICM7216 d'Intersil  
drôle de composant n°34
- 61 vol tous azimuts  
piloté par Clemens Valens
- 64 Q & R  
(presque) tout ce que vous avez toujours  
voulu savoir sur... les afficheurs
- 121 Rétronique  
UHER 4000 Report-L (1964) - fabuleux  
magnétophone à bande portatif
- 126 questions d'éthique  
libérez le spectre radio !
- 128 l'e-choppe d'Elektor
- 130 hexadoku  
casse-tête pour elektorniciens

## lecteur *Card Sound* l'audio à la carte SD

Avec un ordiphone ou un lecteur MP3, vous pouvez écouter de la musique enregistrée en numérique, mais pas au volume voulu dans un salon, ou alors de piètre qualité. Avec le projet Card Sound, la puissance y est et la qualité aussi ! Ce lecteur est léger, petit, restitue l'enregistrement avec très peu de bruit de fond et dispose d'atouts intéressants comme des pilotes de bus I<sup>2</sup>C et RS485.

# 44



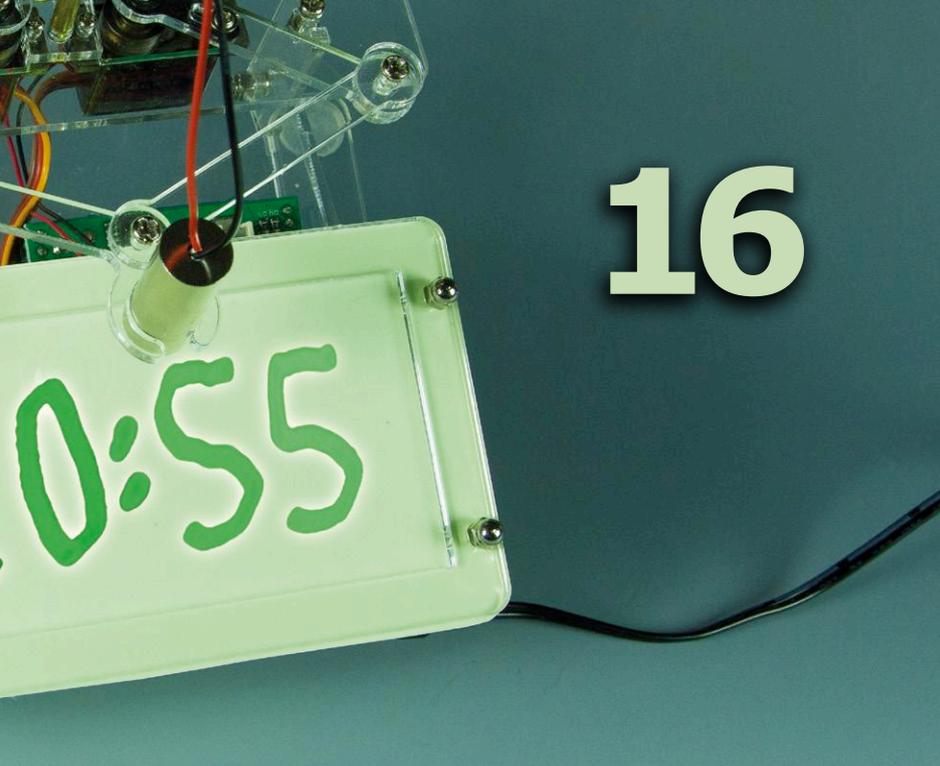
## en coulisse

- 6 afficheurs OLED  
petits afficheurs peu énergivores  
pour vos projets faits maison
- 15 nous avons déménagé !
- 24 l'EEBUS arrive  
fin de la tour de Babel  
pour l'Internet des Objets ?
- 39 on compte sur SmartPi  
le compteur d'énergie intelligent  
pour Raspberry Pi
- 79 couteau suisse pour microcontrôleurs  
PlatformIO,  
un outil universel de programmation
- 84 trucs et astuces  
condensateurs neufs pour écrans fatigués

- 100 bruits de labo...
- 104 hors-circuits de R. Lacoste  
comment consommer moins  
avec du calcul et de la méthode
- 110 sur la toile  
son chipamp à soi, un meilleur son  
avec moins de composants
- 112 projet 2.0  
corrections, mises à jour  
et courrier des lecteurs

## comment consommer moins

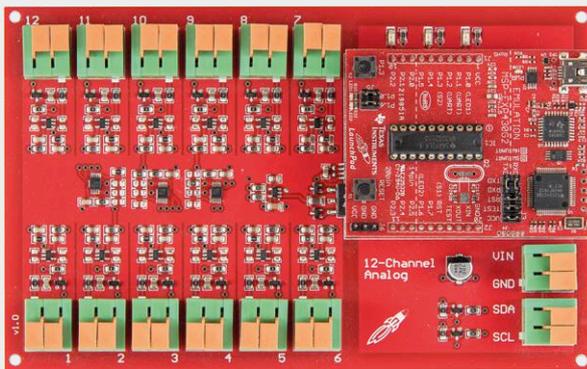




- 32 de la micro-bobine Tesla au synthétiseur MIDI**  
Zeus se la joue Travolta
- 36 EPS - Easy Parking System**  
ou comment éviter de rayer sa voiture
- 44 lecteur Card Sound**  
l'audio à la carte SD
- 54 station météo RPi Zero W**  
sans pièces mécaniques mobiles
- 66 bricolons avec le bus CAN**  
aux commandes des 4000 lampes d'un immeuble
- 72 bio-lampe**  
les « émotions » d'une plante en couleur
- 86 thermostat de bureau wifi**  
surveillance de température flexible et programmable
- 93 transi-testeur\* optique**  
vert = NPN, rouge = PNP
- 94 acquisition de mesures par USB**  
en quête du chauffage optimal
- 101 shield d'expérimentation Arduino version 2.0**  
les mêmes fonctions, mais écran au choix
- 114 DoubleSPiDer**  
adaptateur universel d'interfaces pour projets à microcontrôleurs

## acquisition de mesures par USB en quête du chauffage optimal

94



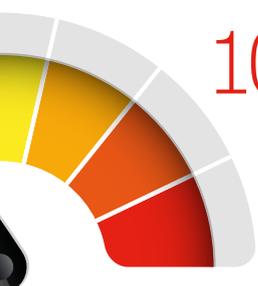
L'optimisation d'une installation de chauffage requiert le suivi continu et précis de différents paramètres. Seules ces données permettent d'introduire rapidement des changements même minimes. L'enregistreur de données décrit ici emmagasine une grande quantité de paramètres analogiques et numériques sur un port de PC à travers une séparation galvanique sûre.

 **bientôt sur ces pages**

 **sur la scène : les projets**

avec du calcul et  
de la méthode

104



- 12 adaptateur de programmation USB pour ESP8266**  
dans la famille Espressif, je voudrais l'ESP-01 et l'ESP-012
- 16 Laser Time Writer**  
horloge à laser
- 29 alimentation solaire pour répéteur Wi-Fi**  
au choix : version analogique ou numérique

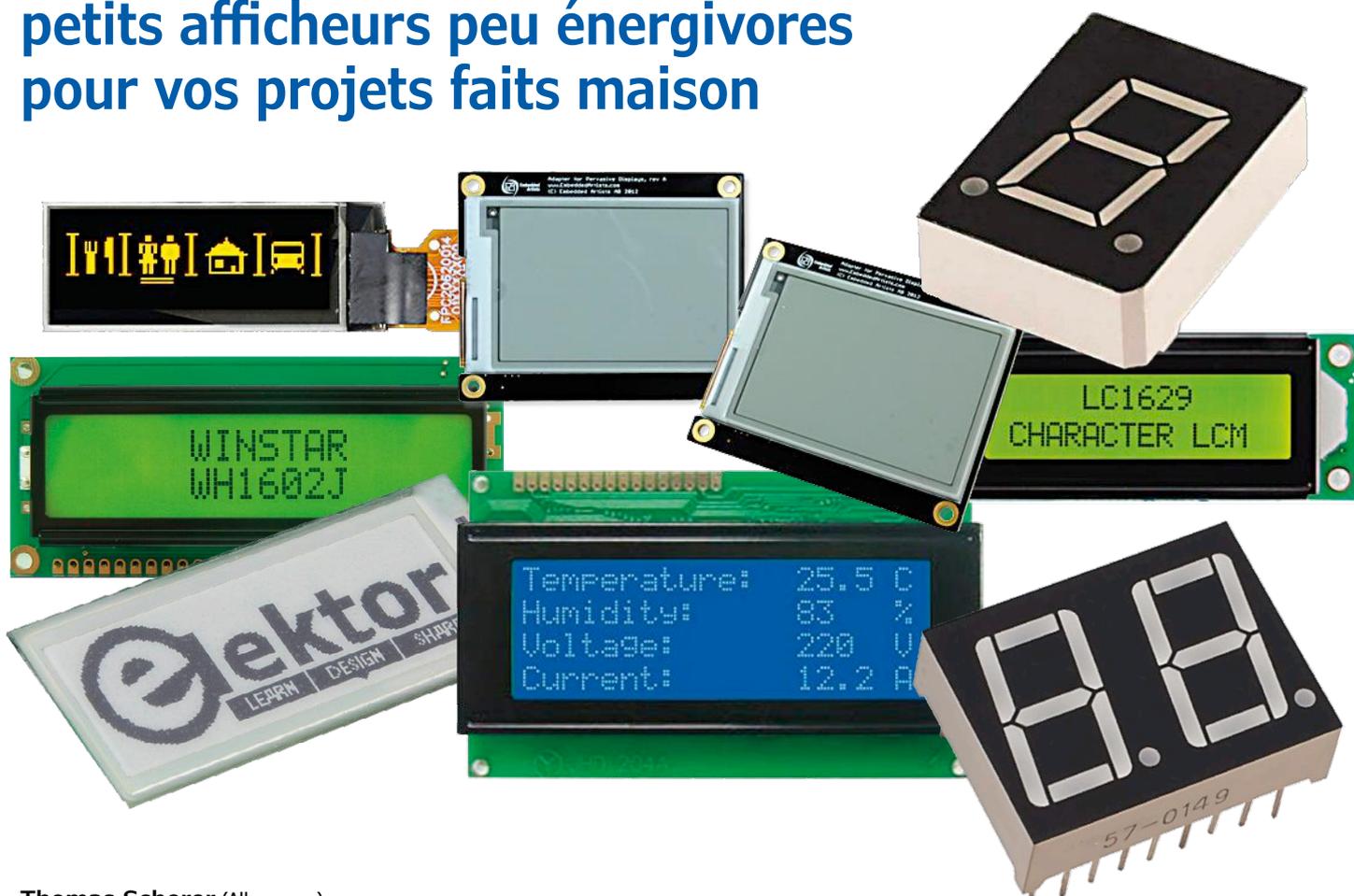
**Extrait du sommaire du prochain numéro :**

- Alimentation Raspberry Pi
- Carillon pour horloge de sable
- Émulateur DCF77 avec ESP8266
- Mesure de courant à potentiel flottant
- Mesure de niveau d'eau à distance
- Mesure de puissance HF de 1 MHz à 10 GHz
- Outil de débogage du bus CAN
- Testeur de port série Platino
- Etc.

Sous réserve de modification.  
Le numéro de mars-avril 2018 paraîtra le 1<sup>er</sup> mars 2018.

# afficheurs OLED

petits afficheurs peu énergivores  
pour vos projets faits maison



Thomas Scherer (Allemagne)

Bien que les écrans d'ordinateur restent les traditionnels modèles à cristaux liquides (LCD), les téléviseurs à écran OLED deviennent peu à peu abordables et bon nombre d'ordiphones de haut de gamme sont déjà équipés en standard d'écrans OLED ; même Apple a suivi le mouvement avec l'iPhone X. Il existe désormais de petits écrans OLED très bon marché qui peuvent être connectés par liaison série à des microcontrôleurs et ainsi être utilisés très simplement dans des projets électroniques personnels.

Les temps sont bien révolus, où téléviseurs (et moniteurs) ont été équipés pendant des siècles de tubes browniens. Le changement de millénaire a vu le passage aux « écrans plats » à cristaux liquides qui certes consomment moins, mais réclament pourtant pas mal d'énergie. Ce dernier point est dû au fait que les cristaux liquides ne s'illuminent pas eux-mêmes, mais modifient simplement la polarisation de la lumière qui les traverse. À l'aide d'un film à polarisation fixe, un pixel LCD ne peut donc que changer la luminosité ou l'intensité de la lumière qui le traverse. La lumière elle-même a initialement toujours été produite par des lampes fluorescentes. Il y a une dizaine d'années, le rétroéclairage à diodes électroluminescentes blanches s'est imposé (même à LED RVB dans la version haut de gamme). Dans un temps historiquement assez court, la technique LCD d'abord affectée par divers inconvénients, s'est transformée en vastes écrans aujourd'hui difficiles à battre. C'est bien connu, le mieux est l'ennemi du bien, et désormais

on trouve les précurseurs de la révolution OLED sous diverses formes d'écrans.

Les premiers écrans à OLED (*Organic Light-Emitting Diode*) produits industriellement à faible coût sont apparus sous forme de modèles simples, souvent monochromes, à petites diagonales. Une nouvelle technologie est généralement plus coûteuse qu'une technologie établie, et avec de petites diagonales, les coûts réels de l'écran jouent un rôle moindre. Il n'est donc pas étonnant qu'on a pu admirer les OLED d'abord sur les ordiphones, puis plus tard sur les grands téléviseurs. Des écrans OLED encore plus petits sont produits en Extrême-Orient depuis plusieurs années, maintenant pour des appareils tels que les moniteurs de fitness. L'extension des capacités de production des petits écrans profite désormais aussi à l'électronicien qui désire équiper de petites séries ou des prototypes avec un affichage raisonnable. Là aussi, le temps est révolu où il fallait se battre avec des écrans LCD ou même des afficheurs à sept

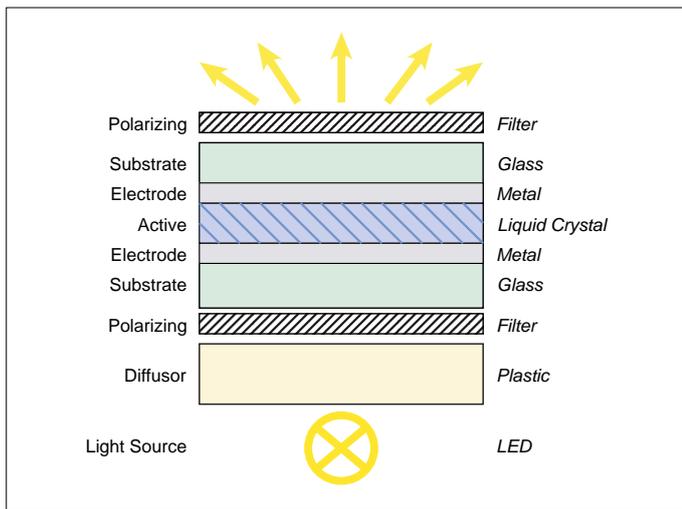


Figure 1. Principe d'un LCD.

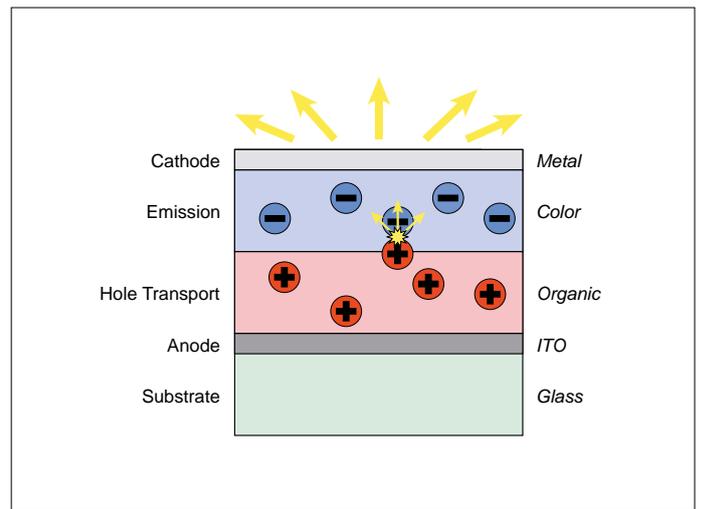


Figure 2. Principe d'un écran OLED. (ITO : couche transparente formée d'oxyde d'indium-étain)

segments. On peut trouver sur eBay des centaines d'écrans OLED modernes à des prix raisonnables, un exemplaire typique est disponible aussi dans l'e-choppe d'Elektor [1].

### Les avantages des OLED

Les écrans OLED doivent bien avoir un plus, sans quoi ils n'auraient pas le potentiel à moyen terme de remplacer complètement les écrans LCD. Les avantages des OLED dépendent directement des faiblesses technologiques des LCD. Cela devient évident lorsque la structure de base des LCD (**fig. 1**) est comparée à celle des écrans OLED (**fig. 2**).

**Rendement** : le rendement lumineux des LED modernes à semi-conducteurs est très bon et reste bien supérieur à celui des OLED (LED à couches semi-conductrices organiques) ; mais la majorité de la lumière du rétroéclairage d'un écran LCD est tout simplement perdue. D'abord, au moins 20% de la lumière est égarée lors de la répartition de la lumière à travers le film diffuseur. Puis par son principe, le filtre de polarisation détruit au moins 50% de la lumière restante. En outre, la transmission de la lumière par le matériau CL (cristal liquide) n'atteint pas 100% avec des pixels entièrement allumés (lumineux). Toutefois un pixel éteint bloque la lumière presque à 100%. Viennent ensuite les couches d'électrodes certes très minces, mais pas complètement transparentes. Enfin et surtout, les filtres de couleur (non représentés) de chaque pixel ne laissent passer qu'une bande spectrale étroite de lumière. Résultat : la transmission totale s'exprime en pourcentage à un chiffre. Pour les écrans OLED cependant, le filtre polarisant et le diffuseur sont absents ; la lumière doit traverser une seule couche d'électrodes. Notez aussi qu'un pixel OLED ne consomme du courant que lorsqu'il est allumé. Le rendement global d'un afficheur OLED est ainsi nettement plus élevé, et la consommation en courant n'est donc qu'une fraction de celle d'un LCD pour une même luminosité.

**Contraste** : avec un LCD, la couche active fait tourner la polarisation de la lumière en fonction de la tension sur les électrodes. Il est très difficile de rendre ceci assez précis pour que toute la lumière soit vraiment absorbée. En outre, les différentes couches de l'écran LCD reflètent également une par-

tie de la lumière ambiante. C'est pourquoi il n'est guère possible d'obtenir des contrastes extrêmement élevés (= rapports lumière-obscurité) pour des pixels voisins. Sur un affichage « à éclairage direct », certaines zones peuvent être obscurcies en assombrissant différentes LED du rétroéclairage, mais cette astuce ne résout pas complètement le problème. En revanche, un pixel OLED est complètement éteint lorsqu'aucun courant ne circule. En outre, l'arrière d'un écran OLED peut être noirci de sorte que moins de lumière ambiante soit réfléchi.

**Influence de l'angle** : bien que les LCD modernes soient annoncés avec un angle de vision de 178°, il est facile de vérifier que pour les angles de vue extrêmes, la luminosité baisse, et ce parfois de manière inégale selon les couleurs, de sorte que des couleurs peuvent changer selon l'angle de vue. Si l'on observe la structure de la figure 1, on se rend vite compte que ces problèmes sont inhérents à cette architecture et qu'il est impossible de les résoudre totalement, même avec des couches très minces. C'est beaucoup plus simple avec un pixel OLED. La lumière qui arrive dans la couche supérieure est en soi indépendante de la direction, et donc un angle de près de 180° peut être atteint sans effort particulier avec une baisse minimale de luminosité et pratiquement sans changement de couleur en cas de vue latérale.

**Vitesse de commutation** : avec les cristaux liquides, pour changer la polarisation, il faut réellement et physiquement déplacer de la matière. Cela prend du temps ! La rapidité de commutation est limitée compte tenu de l'énergie disponible, et malgré bon nombre d'astuces techniques dans la commande. Un écran LCD avec un taux de rafraîchissement (réel) de plusieurs centaines de Hz est à peine réalisable. Cependant, les temps de commutation des OLED sont inférieurs d'au moins un ordre de grandeur, c'est pourquoi, en principe, il est possible de réaliser un afficheur avec une fréquence de rafraîchissement de 1 kHz ; mais qui en aurait vraiment besoin ?

**Inconvénients, etc.** : là où il y a de la lumière, il y a de l'ombre. En l'occurrence dans le cas des OLED, il n'y en a vraiment pas beaucoup. La principale difficulté est la stabilité

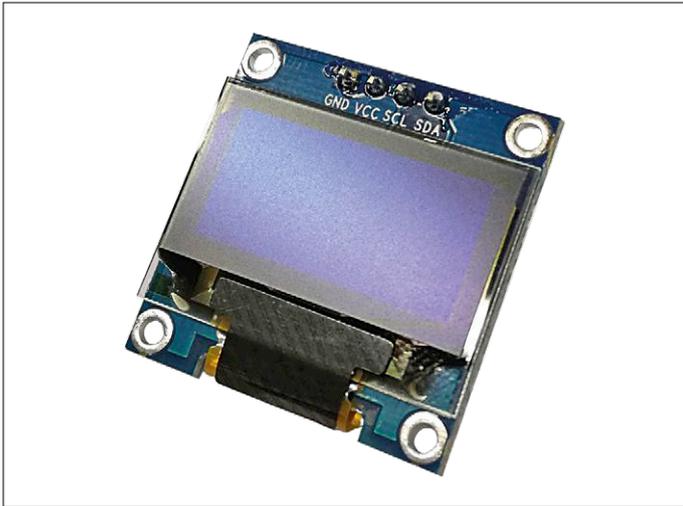


Figure 3. Écran OLED de l'e-choppe Elektor avec une diagonale de 0,96'', 128x64 pixels et interface I<sup>2</sup>C.

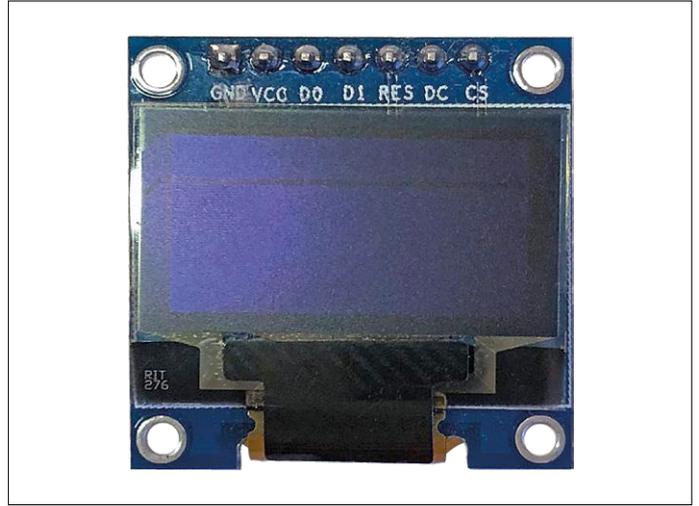


Figure 4. Importé directement de Chine : écran OLED avec interfaces I<sup>2</sup>C et SPI.

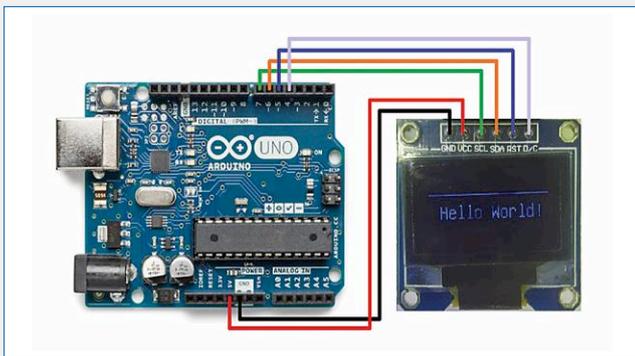
des substances organiques. Les semi-conducteurs vieillissent de fait plus lentement et sont plus robustes. Cependant, les OLED ont été considérablement améliorées durant ces dernières

années, notamment quant à leur durabilité et leur utilisation à long terme. L'industrie de l'électronique a déjà accumulé bien plus d'expérience avec les semi-conducteurs qu'avec la chimie organique. En revanche grâce aux progrès croissants dans la fabrication, à moyen terme les écrans OLED seront encore moins chers que les écrans LCD, car les OLED ne nécessitent pas de production en salle blanche. À partir de ce moment, les écrans LCD seront voués à l'extinction. Cette prévision ne s'applique toutefois que si les derniers écrans LED matriciels basés sur de minuscules LED à semi-conducteurs ne font pas des progrès rapides dans les années à venir, et commencent à remplacer aussi bien les afficheurs LC que les OLED.

## Écran OLED & Arduino

Un écran OLED standard peut être facilement connecté à n'importe quel microcontrôleur, une interface série I<sup>2</sup>C ou SPI étant toujours disponible. La **figure** montre comment l'écran OLED de 0,96» de l'e-choppe d'Elektor est connecté à une carte Arduino Uno sur l'I<sup>2</sup>C. Il suffit de réaliser les connexions suivantes :

n°	écran	Arduino
1	GND	GND
2	VCC	5V
3	SCL	7
4	SDA	6
5	RST	5
6	D/C	4



Connexion de l'écran OLED standard d'Elektor à l'Arduino Uno.

Ensuite, récupérez la bibliothèque *U8glib.h* de [3] et ajoutez-la à l'EDI. Avec le manuel de référence pour l'utilisateur disponible en [4], vous saurez comment utiliser les différentes fonctions, initialiser l'écran et afficher rapidement et facilement lignes, courbes, texte ou images. Vous pouvez télécharger le croquis de démonstration *Hello World* dans la page web de cet article [5].

### Écran standard

Nous allons nous concentrer dans ce qui suit sur de simples écrans OLED équipés d'une interface série pour une connexion à un microcontrôleur ou à d'autres cartes de développement. Comme déjà mentionné, des centaines d'écrans OLED monochromes de diagonales comprises entre 0,6'' et 3'' sont proposés sur eBay, à des prix allant de moins de 5 € à 50 €. Cherchez avec les mots clés *oled display*. Presque tous les modèles sont des afficheurs graphiques à matrice de LED avec des résolutions typiques de 64x48 et 128x32 pixels jusqu'à environ 256x64 pixels. Ils sont pilotés via des interfaces série établies telles que SPI ou I<sup>2</sup>C. Certains modules offrent même les deux interfaces. La connexion électrique est donc triviale avec quatre à six broches (alimentation comprise).

La **figure 3** montre un « écran standard ». Il a une diagonale de 0,96'' (2,43 cm) et une résolution de 128x64 pixels ainsi qu'une interface I<sup>2</sup>C. De tels écrans sont extrêmement courants. Vous les trouverez facilement à moins de 10 € en import direct depuis la Chine (via eBay ou Alibaba, etc.). Il vous faudra pour cela de la patience, et prendre le risque que la douane veuille inspecter le colis, et qu'un fonctionnaire nonobstant amical ne décide un jour pluvieux que la valeur indiquée par l'expéditeur chinois a été « embellie ». On reçoit alors un courrier avec une mise en demeure de régler en personne les formalités de douane au bureau le plus proche et de justifier le prix bas. En effet, la TVA et le cas échéant d'autres taxes fixes s'appliquent à partir d'une valeur de 20 €. Ceci serait

généralement supportable, mais si, comme pour moi, le poste de douane le plus proche est à 120 km du domicile, alors un tel écran ne vaut ni le voyage ni le temps perdu, et la douane gardera la marchandise.

Si vous optez pour la sécurité, commandez chez un fournisseur national ou au moins dans un pays voisin (de l'UE), alors tout ira plus vite et sans problème, même si cela a un coût. Par exemple, dans l'e-choppe d'Elektor [1], un tel écran coûtera 14,95 € ou 13,46 € pour les membres.

J'ai néanmoins aussi essayé la méthode chinoise de « Shopping » et ai commandé pour 7 € port compris un tel module avec interface I<sup>2</sup>C et SPI combinée (fig. 4). J'ai eu de la chance : après un peu plus de trois semaines, un sachet en plastique gris au contenu intact se trouvait dans ma boîte aux lettres. Les modules de 0,96" sont disponibles comme montré sur la figure 5 avec une couleur d'affichage dans la partie supérieure et une autre couleur dans la partie inférieure. Il existe également des versions monochromes en bleu-vert ou blanc. Pratiquement tous ces écrans sont équipés d'un contrôleur de type SSD1306 [2] du fabricant Solomon. Ces modules peuvent fonctionner à des tensions comprises entre 3,3 V et 5 V, et ne nécessitent qu'un courant de 10 mA. Si l'on recherche « SSD1306 » avec Google, on obtient une avalanche de liens sur le brochage, vers des conseils d'utilisation et des bibliothèques (par exemple pour Arduino ou même Bascom). Ces modules sont donc idéaux pour l'affichage d'informations pour des projets personnels, des prototypes ou des petites séries.

## ▶ Écrans OLED : précis et économes !

L'encadré **Écran OLED & Arduino** décrit comment connecter un module à une carte Arduino Uno et quel logiciel est nécessaire. Grâce à un large éventail de logiciels à code source ouvert, il n'est pas nécessaire de gérer chaque pixel et de réinventer la roue, car on a accès à toutes sortes de fonctions graphiques déjà testées.

### Autres écrans OLED

Outre le quasi standard de 0,96", il existe aussi la demi-version avec 128×32 pixels pour une diagonale de 0,91" (2,31 cm). La figure 6 en montre un exemple : le prix est identique à celui du 0,96" et ce type s'appuie également sur la puce SSD1306. En raison du nombre de pixels horizontaux moindre, on fera attention lors de l'utilisation de bibliothèques pour l'écran standard. Ce standard existe également en un peu plus grand, à savoir 1,3" (3,30 cm), voire 1,54" (3,91 cm). La police très nette, mais plutôt petite est alors un peu plus lisible, toutefois la consommation augmente légèrement. La figure 7 montre la version miniature avec seulement 0,66" (1,67 cm) et 64×48 pixels. Vous l'aurez deviné : il est lui aussi équipé du SSD1306.

Bien entendu, outre ces types standards, on trouve en Chine à peu près tout ce qui est sorti pour les appareils électroniques existants. Néanmoins pour ces écrans spéciaux, il est souvent difficile d'identifier le contrôleur et d'obtenir suffisamment de données techniques. De plus, on dispose assurément de beaucoup moins de travaux préalables de la communauté que pour le contrôleur standard SSD1306. Ces écrans sont parfois

équipés de câble plat au lieu de broches ou points de soudure, pour lesquels il faudra d'abord trouver le bon connecteur. En revanche, avec les modules SSD1306, on est en sécurité et il devient difficile de faire des erreurs.

Exception à la règle du SSD1306 : il existe des écrans OLED alphanumériques (fig. 8) à deux ou quatre lignes de 12, 16 ou 20 caractères à écriture blanche, bleue, jaune ou rouge sur fond noir, et avec des diagonales d'écran de 2,5" à 6". Ils conviennent pour remplacer les écrans à cristaux liquides alphanumériques classiques et offrent l'avantage d'une meilleure lisibilité pour une consommation d'énergie nettement moindre, le rétroéclairage des LCD étant particulièrement gourmand.



Figure 5. Écran OLED bicolore avec contenu de démonstration.



Figure 6. Écran OLED avec diagonale de 0,91", 128×32 pixels et interface I<sup>2</sup>C

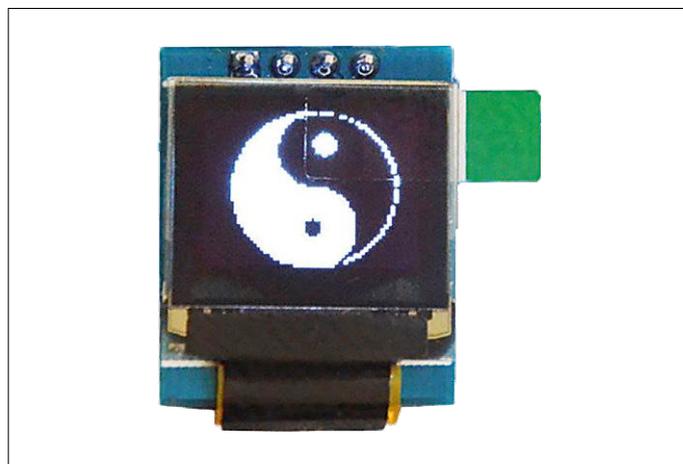


Figure 7. Écran OLED avec diagonale de 0,66", 64×48 pixels et interface I<sup>2</sup>C.



Figure 8. Écran OLED alphanumérique à deux lignes de 16 caractères chacune en remplacement des LCD habituels.

Ce luxe n'est cependant pas particulièrement bon marché : on démarre à plus de 20 € jusqu'à bien au-delà de 50 € pour les grands modèles. En tout cas, on est toujours au-delà des limites de franchise douanière, et on doit gérer les inconvénients qui en découlent. Par ailleurs, ces écrans s'appuient également sur un contrôleur Solomon : la puce SSD1311 avec générateur de caractères intégré.

### Conclusion

Il n'a jamais été aussi facile d'équiper à bon marché un projet à microcontrôleur d'un écran graphique monochrome. Les écrans OLED sont très précis et conviennent parfaitement à une alimentation par batterie grâce à leur consommation relativement modeste. Toutefois, un tel projet ne reste simple que si l'on s'en tient aux écrans équipés de la puce quasi standard SSD1306. Sur ce sujet, on dispose sur l'internet de bien plus d'informations que nécessaire. Les modules d'affichage sont extrêmement bon marché lorsqu'ils sont importés directement de Chine. Mais si vous ne voulez pas attendre des semaines ou prendre de risques avec les taxes douanières, vous trouverez en Europe un choix suffisant à des prix raisonnables. Si vous possédez une carte Arduino ou toute autre plateforme de développement, vous devriez en tout cas vous offrir un tel écran. Un affichage élégant est toujours utile. ◀

(160550 - version française : Xavier Pfaff)

### Liens

- [1] Afficheur OLED dans l'e-choppe d'Elektor : [www.elektor.fr/3d-printer-head-and-mat-temperature-controllerboard](http://www.elektor.fr/3d-printer-head-and-mat-temperature-controllerboard)
- [2] SSD1306 : [www.solomon-systech.com/en/product/display-ic/oled-driver-controller/ssd1306](http://www.solomon-systech.com/en/product/display-ic/oled-driver-controller/ssd1306)
- [3] Bibliothèque Arduino : <https://github.com/olikraus/u8glib>
- [4] Manuel de référence des fonctions de la bibliothèque : <https://github.com/olikraus/u8glib/wiki/userreference>
- [5] Croquis de démonstration : [www.elektormagazine.fr/160550](http://www.elektormagazine.fr/160550)

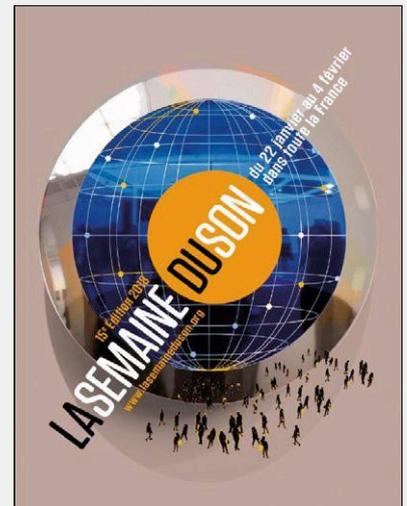


### Un événement oublié ?

Vous organisez une conférence, un salon... ou bien vous participez à un séminaire ou tout autre événement qui aurait sa place ici, partagez cette information avec tous les lecteurs. Envoyez-nous tous les détails à [redaction@elektor.fr](mailto:redaction@elektor.fr).

## janvier 2018

- ◇ **Cartexpo**  
12 au 13/01 – Paris  
[www.cartexpo.fr](http://www.cartexpo.fr)
- ◇ **Salon de la plongée sous-marine**  
12 au 15/01 – Paris  
[www.salon-de-la-plongee.com/fr/accueil.html](http://www.salon-de-la-plongee.com/fr/accueil.html)
- ◇ **Rencontres de la mobilité intelligente**  
14 au 25/01 – Paris  
[www.congres-atecitsfrance.fr](http://www.congres-atecitsfrance.fr)
- ◇ **Convention FFDomotique 2018**  
17 au 18/01 – Bonnemain  
[www.ffdomotique.org](http://www.ffdomotique.org)
- ◇ **Salon du travail & mobilité professionnelle**  
19 au 20/01 – Paris  
[www.salondutravail.fr](http://www.salondutravail.fr)
- ◇ **Semaine du son**  
22/01 au 04/02 – Paris et partout en France  
[www.lasemaineudson.org](http://www.lasemaineudson.org)
- ◇ **10<sup>e</sup> Forum international de la cybersécurité**  
23 au 24/01 – Lille  
[www.forum-fic.com/](http://www.forum-fic.com/)
- ◇ **Salon de la Radio**  
25 au 27/01 – Paris  
[www.salondelaradio.com](http://www.salondelaradio.com)
- ◇ **ANGOULÊME BD**  
25 au 28/01 – Angoulême  
[www.bdangouleme.com/](http://www.bdangouleme.com/)
- ◇ **Cidisc** - convention Internationale du Disque de Collection  
27 au 28/01 – Paris  
[www.jukeboxmag.com/](http://www.jukeboxmag.com/)
- ◇ **SEPEM Industries Nord**  
29 au 31/01 – Douai  
[douai.sepem-industries.com/](http://douai.sepem-industries.com/)
- ◇ **SEPEM Industries Nord-Ouest**  
30/01 au 01/02 – Rouens  
[rouen.sepem-industries.com/](http://rouen.sepem-industries.com/)
- ◇ **Congrès Embedded Real-Time Software and Systems**  
31/01 au 02/02 – Toulouse  
[www.erts2018.org](http://www.erts2018.org)



# la communauté d'Elektor

DÉCOUVRIR → CRÉER → PARTAGER

82

pays

248153

membres actifs

1040

experts & auteurs

489

publications

235332

visiteurs (par mois)

Elektor, c'est bien plus qu'un simple magazine. C'est une communauté d'électroniciens, du débutant au professionnel, désireux d'apprendre, de concevoir et de partager une électronique qui étonne.



**Boutique en ligne d'Elektor :** la caverne d'Ali Baba des électroniciens ! 10 % de remise permanente pour les membres GREEN et GOLD.  
[www.elektor.fr](http://www.elektor.fr)



**Magazine Elektor :** six numéros par an d'électronique inédite (projets, bancs d'essais, cours, trucs et astuces, actualité...)  
[www.elektormagazine.fr](http://www.elektormagazine.fr)



**Elektor PCB Service :** gravure à la demande de vos circuits imprimés, en un ou plusieurs exemplaires. [www.elektorpcbservice.com](http://www.elektorpcbservice.com)



**Elektor GRATUIT, version numérique :** lettre d'information à l'affût des nouveautés. C'est gratuit : [www.elektor.fr/inscription](http://www.elektor.fr/inscription)



**Elektor Academy :** l'apprentissage par la pratique au travers de webinaires, séminaires, ateliers, DVD... [www.elektor.fr](http://www.elektor.fr)



**Livres & DVD Elektor :** Arduino, Raspberry Pi, microcontrôleurs, Linux et bien d'autres. 10 % de remise pour les membres !  
[www.elektor.fr](http://www.elektor.fr)



**Elektor TV :** présentations, comptes-rendus, déballage et évaluation de produits. Pour y voir plus clair.  
[www.youtube.com/user/ElektorIM](http://www.youtube.com/user/ElektorIM)



**Elektor Labs :** l'incubateur où partager ses projets et participer à ceux des autres. Nous faisons grandir vos idées !  
[www.elektormagazine.fr/labs](http://www.elektormagazine.fr/labs)

## connectez-vous à notre communauté

### formule

**GREEN** 92,50 € par an

- ✓ accès à l'archive d'Elektor
- ✓ 10% de remise dans l'e-shoppe
- ✗ 6x magazine imprimé
- ✓ 6x magazine numérique
- ✓ des offres exclusives
- ✓ accès à plus de 1 000 fichiers Gerber
- ✗ le DVD annuel d'Elektor

connectez-vous

[www.elektor.fr/formule-greencard](http://www.elektor.fr/formule-greencard)

### formule

**GOLD** 127,50 € par an

- ✓ accès à l'archive d'Elektor
- ✓ 10% de remise dans l'e-shoppe
- ✓ 6x magazine imprimé
- ✓ 6x magazine numérique
- ✓ des offres exclusives
- ✓ accès à plus de 1 000 fichiers Gerber
- ✓ le DVD annuel d'Elektor

connectez-vous

[www.elektor.fr/formule-goldcard](http://www.elektor.fr/formule-goldcard)

### formule

**GRATUITE**

- ✗ accès à l'archive d'Elektor
- ✗ 10% de remise dans l'e-shoppe
- ✗ 6x magazine imprimé
- ✗ 6x magazine numérique
- ✓ des offres exclusives
- ✗ accès à plus de 1 000 fichiers Gerber
- ✗ le DVD annuel d'Elektor

connectez-vous

[www.elektor.fr/inscription](http://www.elektor.fr/inscription)



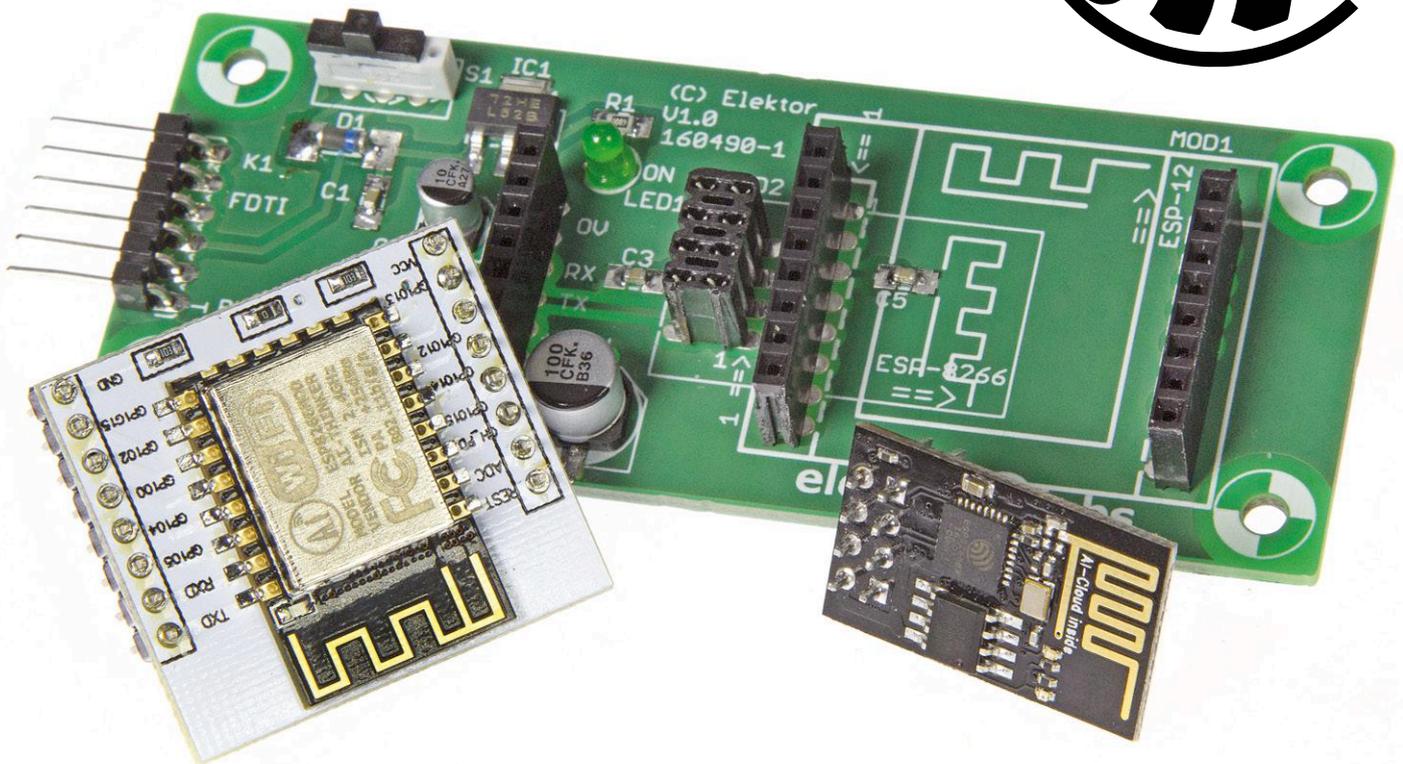
[facebook.com/ElektorFR](https://facebook.com/ElektorFR)



[twitter.com/Elektor](https://twitter.com/Elektor)

# adaptateur de programmation USB pour ESP8266

dans la famille Espressif,  
je voudrais l'ESP-01 et l'ESP-012



Pascal Rondane (Tours/France)

Ce montage permet de programmer les modules ESP8266 de types ESP-01 et ESP-012 via une interface USB FDTI. Un connecteur additionnel permet de s'adresser à tout autre module de la famille ESP8266, à l'aide de fils de câblage.

Ce montage est très pratique, notamment lors de la mise au point sur des plaques d'essai ou pour du prototypage rapide, car il permet de se passer de commutateurs ou de ponts pour activer le mode programmation. Un simple câble avec convertisseur USB-TTL FTDI

permet d'alimenter le montage et de transmettre le code.

## Électronique

Le schéma (fig. 1) montre que la tension d'alimentation de 5 V qui arrive sur le connecteur à six points K1 est abais-

sée à 3,3 V par le régulateur IC1. Cette tension réduite est nécessaire pour alimenter les modules ESP8266. Un interrupteur permet de couper la tension sans débrancher le câble FDTI ; la LED verte indique si le montage est sous tension ou pas. Les condensateurs C3 et C4

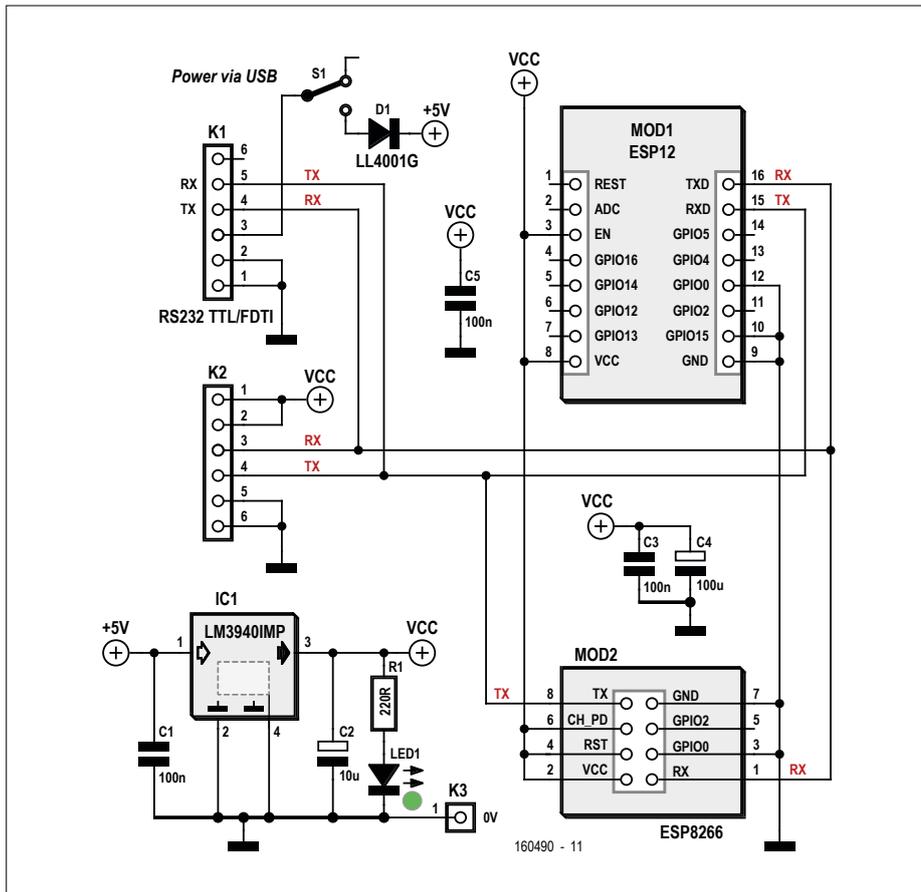


Figure 1. Schéma de l'adaptateur.

assurent un découpage efficace de la tension d'alimentation de IC1 d'une part et des modules ESP d'autre part. Il est inutile de configurer les modules pour les passer en mode programmation, tout est réalisé sur le circuit imprimé. Pour l'ESP-01, la broche GPIO0 est reliée au 0 V, les broches RST et CH\_PD au 3,3 V. Pour l'ESP-012, la broche EN est reliée à VCC, les broches GPIO0 et GPIO15 à la masse.

Le connecteur femelle K2 met à disposition les bornes d'alimentation 0 V et 3,3 V ainsi que les signaux TX et RX. Ainsi il est possible avec quelques fils volants de programmer un module externe au brochage atypique. Le câblage n'offre pas de difficulté particulière, on commencera par souder les composants CMS et on finira par les connecteurs.

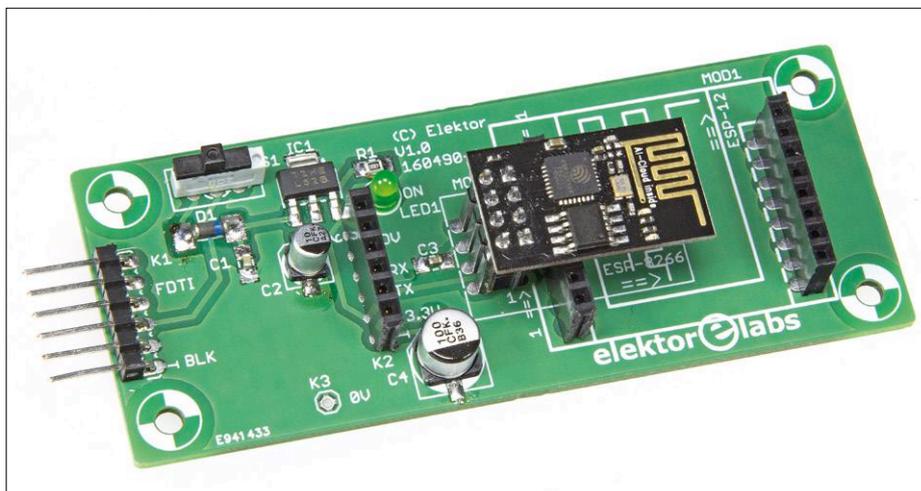


Figure 2. Le module ESP-01 est prêt à être programmé.

### INFOS SUR LE PROJET

#

ESP8266
Wi-Fi

▬

débutant

connaisseur

expert

⌚

env. 20 mn

🔧

Câble avec convertisseur  
USB-TTL FTDI

€

env. 15 €

Cette carte est polyvalente, je m'en sers pour tester des prototypes. Elle offre l'avantage de ne pas être soumise à des mauvais contacts ou des problèmes de CEM liés aux plaques d'essai...

### Logiciel

Commencez par brancher le câble avec convertisseur USB-TTL FTDI. Attention : le fil vert de l'interface est côté interrupteur, le noir est repéré par la sérigraphie sur le circuit imprimé (BLK). L'idéal est de coller une gommette de couleur sur la partie visible du connecteur comme repère. Sur l'ordinateur, installez le pilote FTDI, ensuite sélectionnez le port COM de l'interface.

Pour programmer les modules, j'utilise l'EDI Arduino. Il faut au préalable ajouter les cartes ESP8266. Ouvrez [Fichiers](#) → [Préférences](#) et en bas de la fenêtre, copiez le lien suivant dans le champ URL de [gestionnaire de cartes supplémentaires](#) :

[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)

Ensuite, ouvrez [Outils](#) → [Type de carte](#) → [Gestionnaire de carte](#). Enfin, parcourez la liste jusqu'à [esp8266](#) et cliquez sur [Installer](#). Dans le menu [Outils](#) → [Type de carte](#) devrait apparaître [ESP8266 Modules](#) avec différentes variantes. Il

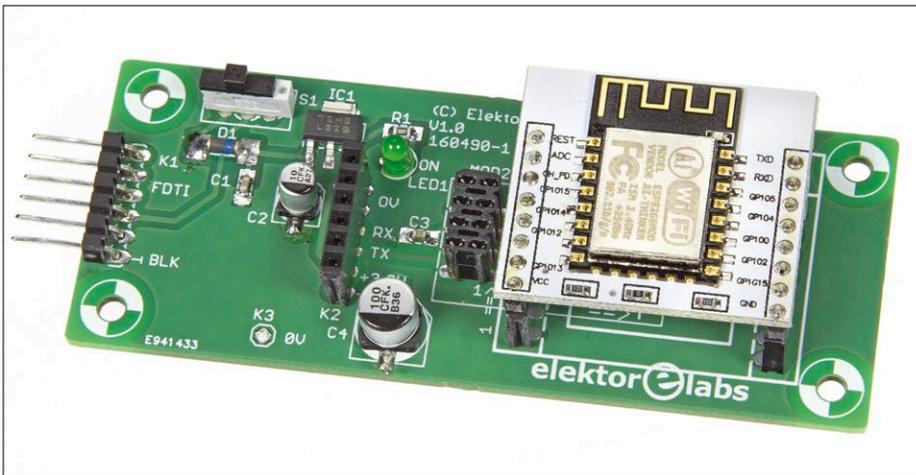


Figure 3. Le module ESP-012 repose sur les deux rangées d'embases femelles.

reste à sélectionner le module à programmer : *Generic ESP8266 module* pour l'ESP-01, *NodeMCU09 (ESP-12 Module)* pour l'ESP-012.

Une fois que le module est correctement enfilé sur son support (voir **figures 2 et 3**), la LED verte doit être allumée. Vous n'avez plus qu'à téléverser votre programme depuis l'EDI Arduino. ◀

(160490)



## LISTE DES COMPOSANTS

### Résistance :

R1 = 220 Ω, à couche épaisse, 5%, 0,1 W, 150 V, CMS 0805

### Condensateurs :

C1, C3, C5 = 100 nF, 50 V, X7R, CMS 0805

C2 = 10 µF/16 V, radial, CMS, Panasonic, EEEFK1C100R

C4 = 100 µF/16 V, radial, CMS, Panasonic, EEEFK1C101P

### Semi-conducteurs :

D1 = diode LL4001G, 50 V, 1 A

IC1 = LM3940, 3,3 V, à faible tension de déchet

LED1 = LED verte, 3 mm

### Divers :

K1 = barrette sécable, 1×6 contacts, horizontale, au pas de 2,54 mm

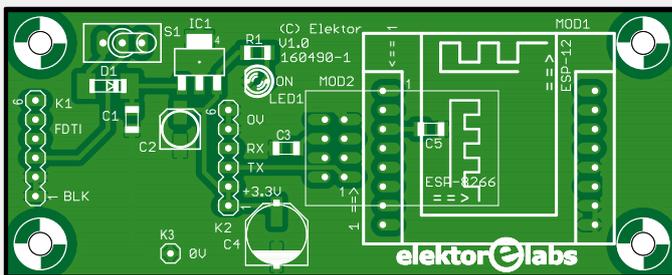
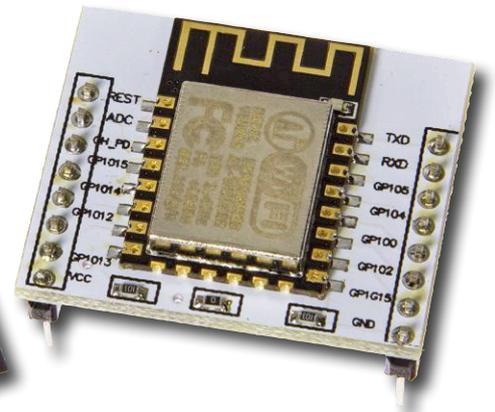
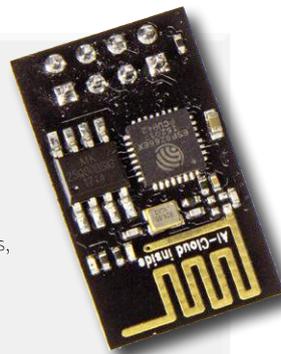
K2 = barrette femelle sécable, 1×6 contacts, verticale, au pas de 2,54 mm

S1 = commutateur à glissière SPDT, traversant, 500 mA (série EOZ 1K2)

MOD1 = 2× barrette femelle pour l'ESP-012, sécable, 1×8 contacts, verticale

MOD2 = 1× barrette femelle pour l'ESP-01, sécable, 1×8 contacts, verticale  
1 câble avec convertisseur TTL-232R-5V FTDI, 1,8 m

Circuit imprimé 160490-1



## Liens

- [1] Pilote FTDI : [www.ftdichip.com/FTDrivers.htm](http://www.ftdichip.com/FTDrivers.htm)
- [2] Fiche technique du câble avec convertisseur USB-TTL FTDI : [www.ftdichip.com/Support/Documents/DataSheets/Cables/DS\\_TTL-232R\\_CABLES.pdf](http://www.ftdichip.com/Support/Documents/DataSheets/Cables/DS_TTL-232R_CABLES.pdf)
- [3] Page de l'article : [www.elektormagazine.fr/160490](http://www.elektormagazine.fr/160490)



DANS L'E-CHOPPE

→ 160490-1  
circuit imprimé



# nous avons déménagé !

**Thijs Beckers** (Elektor)

Finis les douves, le puits, la cave, le grenier...  
Elektor a quitté son petit château de Limbricht (Pays-Bas) pour une résidence plus moderne à Aix-la-Chapelle (Allemagne). Le labo et la rédaction sont maintenant réunis au dernier étage (très lumineux) d'un immeuble de bureaux, situé près de l'université

Elektor est désormais dans une région en plein essor technologique, et cela porte déjà ses fruits : nous collaborons à un projet intéressant avec l'université de Rhénanie-Westphalie ; nous en parlerons bientôt dans le magazine !  
À quoi ressemble désormais le labo ? Voyez par vous-même ! Ces photos en disent plus que mille mots.  
Enfin, sachez que nous n'avons pas tous quitté le pays qui a



technique de Rhénanie-Westphalie. Nos lecteurs – surtout les néerlandais – pourraient croire que nous nous sommes expatriés, mais nous sommes à un jet de pierre de notre siège précédent. Le service commercial a suivi et occupe un autre étage du bâtiment.

vu naître votre magazine préféré : le service à la clientèle et la logistique restent à Susteren, dans la région du Limbourg néerlandais, à une cinquantaine de kilomètres d'Aix-la-Chapelle. ◀

(160534 – version française : Jean-Louis Mehren)



# Laser Time Writer

## horloge à laser

Ilse Joostens & Peter S'heeren (Belgique)

D'après une idée de Nicholas Stock (San Diego, Californie)

### INFOS SUR LE PROJET



Arduino

laser

horloge



débutant

→ connaisseur

expert



env. 1 h ½



Fer à souder à pointe fine,  
outils divers,  
ordinateur



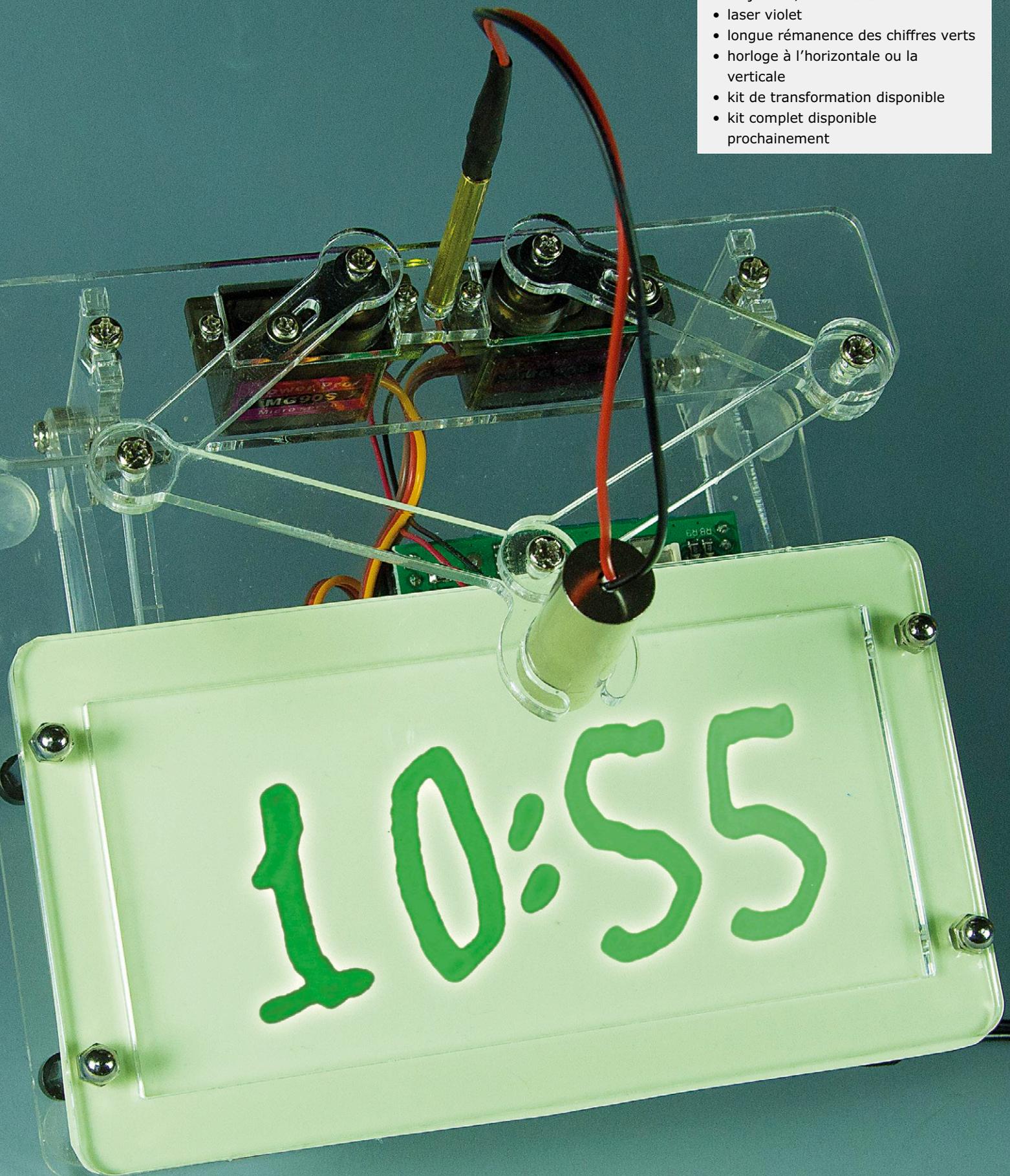
env. 45 € (kit de  
transformation)

env. 120 € (horloge de sable)

La très populaire horloge de sable a été décrite dans ces pages il y a un an (Elektor janvier/février 2017) ; elle était basée sur la Plotclock du FabLab de Nuremberg, qui écrivait l'heure avec un feutre. Nous avons depuis réfléchi à améliorer l'horloge, et nous sommes fiers de vous présenter l'horloge *Laser Time Writer*, qui inscrit l'heure sur un morceau d'autocollant phosphorescent.

## Caractéristiques

- basé sur l'horloge de sable d'Elektor de janvier/février 2017
- laser violet
- longue rémanence des chiffres verts
- horloge à l'horizontale ou la verticale
- kit de transformation disponible
- kit complet disponible prochainement



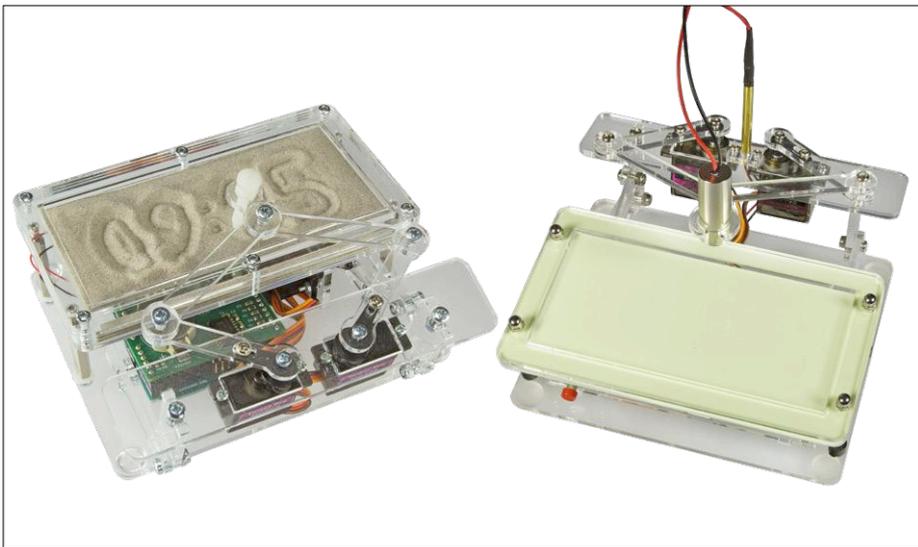


Figure 1. L'horloge de sable (à gauche) et l'horloge *Laser Time Writer* (à droite).

## Entracte 1 : ce qui luit dans le noir...

L'autocollant que nous utilisons est recouvert d'une substance qui absorbe l'énergie de la lumière incidente et la restitue ensuite lentement sous forme de... lumière. La lumière restituée est généralement verte, mais il y a d'autres matériaux et d'autres couleurs. Quoi qu'il en soit, la théorie de ce mécanisme n'est pas si simple que ça.

### Fluorescence et phosphorescence

Lorsqu'un matériau irradie des photons suite à un apport d'énergie, on parle de luminescence ; si c'est suite à un apport de chaleur, on parle plutôt d'incandescence.

La photoluminescence est une forme de luminescence où un matériau absorbe et réémet des photons : lors de l'absorption, l'énergie excite les électrons du matériau, et cette énergie est restituée lorsque les électrons retombent dans leur état initial (relaxation). La longueur d'onde est souvent plus longue lors de la réémission ; il s'agit du déplacement de Stokes. On distingue trois types de mécanismes différents : la fluorescence, la phosphorescence et l'effet Raman (ne vous tracassez pas, il n'y a pas d'interro à la fin de l'article !).

Le terme fluorescence vient de la désignation d'un minéral, la fluorine (ou fluorite), composée de fluorure de calcium, sur laquelle on a pour la première fois étudié le phénomène. Le délai entre absorption et réémission est relativement court, de l'ordre de la nanose-

conde ; la lumière incidente est donc immédiatement réémise avec une longueur d'onde différente. Exemples d'application : feutres fluos et lumière noire (substances et pigments qui réémettent des UV sous forme de lumière visible). Les tubes fluorescents, les ampoules à économie d'énergie ou à LED, et les LED blanches font également appel à des matériaux fluorescents (souvent qualifiés, à tort, de phosphorescent !) pour réémettre des UV ou une lumière bleue sous forme d'un spectre que nous considérons comme blanc.

Les substances phosphorescentes sont connues depuis le Moyen Âge (le mot grec *phosphoros* signifie porteur de lumière). C'est l'alchimiste allemand Hennig Brand qui a découvert le phosphore par hasard en 1669. Le nom attribué à cet élément vient du fait que le phosphore blanc émet de la lumière visible dans l'obscurité quand il est exposé à l'air ; cette réaction chimique n'est pas de la phosphorescence, mais bien de la chimiluminescence. On parle de phosphorescence lorsque les électrons excités par une onde lumineuse retournent à des niveaux d'énergie plus bas lentement (le délai va de quelques millisecondes à plusieurs heures) en réémettant de la lumière ; il s'agit d'un phénomène relevant de la mécanique quantique. Les films ou peintures phosphorescentes en sont un bel exemple, ils continuent à luire des heures durant après avoir été illuminés.

L'horloge de sable était vraiment fascinante [1], mais elle avait quelques inconvénients pour un usage régulier. Le lit de sable doit rester horizontal, et la bonne lisibilité de l'heure dépend de l'éclairage. Le lissage de la couche de sable est bien entendu indispensable, mais les vibrations sont très énervantes. Ce n'est donc pas une horloge à placer sur sa table de chevet...

L'horloge *Laser Time Writer* « inscrit » quant à elle l'heure sur un morceau d'autocollant phosphorescent (c'est-à-dire qui continue à émettre de la lumière après avoir été éclairé). Après une pause, qui dépend des conditions d'éclairage, les chiffres s'estompent et un nouveau cycle recommence. Le phénomène est expliqué dans l'encadré **entracte 1**. Nous aurions pu utiliser une LED bleue ou blanche puissante au lieu d'un laser, mais avec ce dernier la distance entre la source lumineuse et la couche phosphorescente n'a pas beaucoup d'importance ; en plus, ça fait *high-tech*.

L'horloge de sable peut être transformée en horloge *Laser Time Writer*, un kit est disponible dans l'e-choppe ; la **figure 1** montre les deux horloges côte à côte. Un kit complet de l'horloge en version laser sera bientôt disponible, nous vous tiendrons au courant.

### La phosphorescence en pratique

Dans le temps, on utilisait du sulfure de zinc dopé avec du cuivre, ou parfois de l'argent, comme matériau phosphorescent. De nos jours, on préfère des peintures à base d'aluminate de strontium, dopées avec de l'europlum, du dysprosium ou d'autres terres rares. On a aussi récemment utilisé des silicates de métaux alcalino-terreux dopés, pour obtenir d'autres couleurs, comme le bleu. La plupart de ces substances sont inertes chimiquement et biologiquement ; en d'autres termes, elles ne sont pas toxiques.

Le matériau autocollant que nous avons choisi n'existe à notre connaissance qu'en vert, mais on peut acheter des poudres phosphorescentes pour fabriquer ses propres peintures. Des recettes sont disponibles sur l'internet.

### La mécanique

Il n'y a pas grand-chose à dire pour la partie mécanique, l'assemblage est en gros celui de l'horloge de sable. Une poignée de composants suffit pour la transformation en *Laser Time Writer*, et une

## Entracte 2 : classification des lasers

### Classe 1

Les lasers de cette classe sont intrinsèquement sans danger pour les yeux, soit parce que la puissance est très faible, soit parce qu'un dispositif empêche les yeux d'être touchés par la lumière.

### Classe 1M

A priori sans danger si on regarde directement le faisceau, mais l'utilisation d'une loupe ou d'un télescope peut les rendre dangereux. Le faisceau est divergent ou d'un diamètre relativement grand. La lumière peut être visible ou non, tout comme pour la classe 1.

### Classe 2

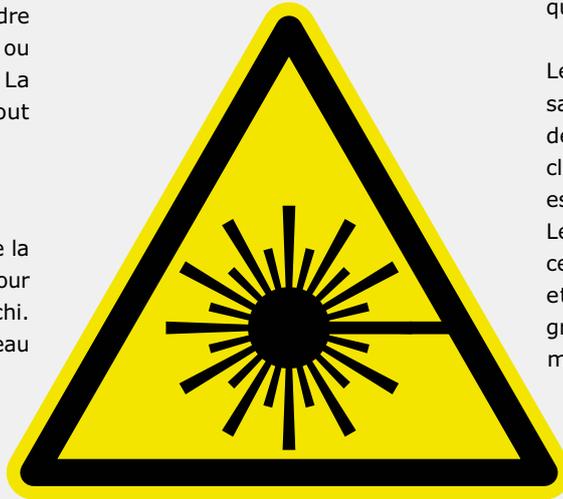
Les lasers de cette classe émettent de la lumière visible ; ils sont sans danger pour les yeux lorsque le faisceau est réfléchi. La vision directe et prolongée du faisceau peut causer des lésions oculaires.

### Classe 2M

La lumière émise est visible, et il n'y a en principe pas de danger pour les yeux lorsque les réflexes de défense (réflexe palpébral) ne sont pas inhibés. Le faisceau, même réfléchi, peut être dangereux lors de l'utilisation de dispositifs optiques tels que ceux décrits plus haut pour la classe 1M. Ce faisceau est très divergent ou large.

### Classe 3R

Les risques pour l'œil sont faibles, sauf si on regarde directement dans le faisceau. Le niveau d'émission est limité à cinq fois la limite d'émission accessible (LEA) des lasers des classes 1 (lumière invisible) et 2 (lumière visible). Un laser de 1 à 5 mW, qui peut émettre de la lumière visible en continu, sera en principe de classe 3R.



### Classe 3B

La vision directe du faisceau est toujours dangereuse et occasionne donc des lésions sérieuses ; le faisceau peut aussi occasionner des dommages à la peau, mais dans une moindre mesure.

Les réflexions diffuses sont en principe sans danger. En émission continue, la puissance est limitée à 500 mW.

### Classe 4

C'est la classe la plus élevée. Le faisceau est dangereux pour les yeux, même en cas de réflexions diffuses, et pour la peau. L'illumination d'un matériau inflammable avec un laser de cette classe peut provoquer un incendie.

Le laser que nous utilisons a une puissance de 5 à 10 mW, il devrait donc être de la classe 3B (limite inférieure de la classe) ; l'utilisation que nous en faisons est cependant sans danger.

Le diamètre relativement grand du faisceau (de l'ordre de quelques millimètres) et la fixation du laser sur le pantographe limitent encore le risque. Avec le mouvement continu du mécanisme, la durée d'exposition de l'œil au faisceau réfléchi ne serait que d'une fraction de seconde, ce qui est sans danger.

**En tout cas, n'exposez jamais directement vos yeux au faisceau, et ne dirigez pas celui-ci sur des surfaces réfléchissantes !**

notice (en anglais) richement illustrée est disponible sur la page du projet [2] ou dans l'e-choppe.

Le sable est bien entendu remplacé par un morceau d'autocollant phosphorescent, et le stylet par un module laser. La plaque de base a aussi été modifiée : quatre trous permettent de fixer l'horloge sur un mur, et deux positions sont possibles pour l'Arduino, de telle manière que les connecteurs de l'alimentation et USB soient à l'arrière ou sur le devant. Notez cependant que cette nouvelle plaque de base ne fait pas partie du kit de transformation proposé dans l'e-choppe !

Les bras du pantographe ont été allongés, de manière à pouvoir écrire jusque dans les coins. On peut même inscrire l'heure à l'envers (rotation de 180°). Il y a maintenant un bouton-poussoir pour démarrer un cycle d'écriture. Cette

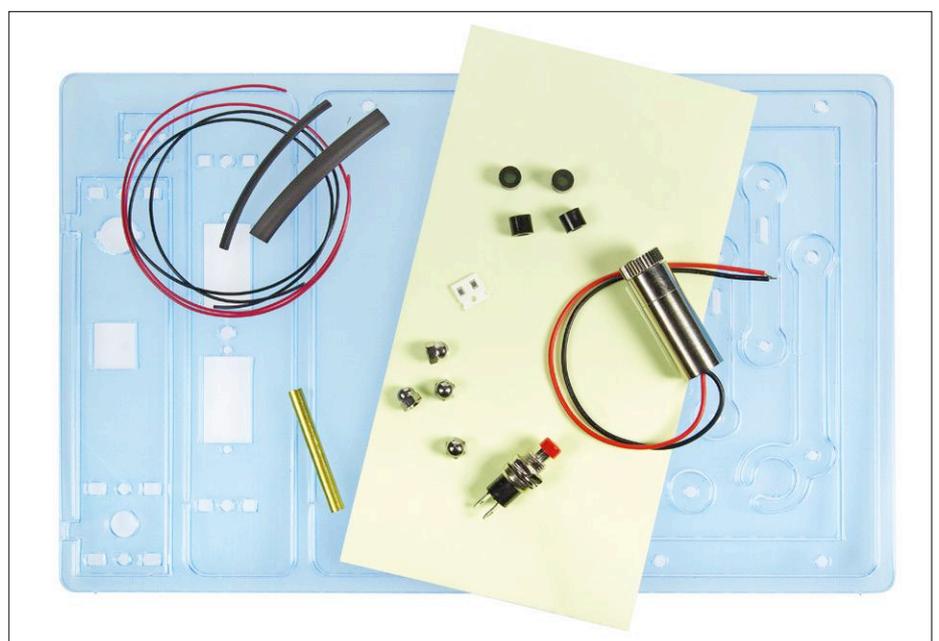


Figure 2. Le contenu du kit de transformation.



option avait déjà été ajoutée à la dernière version du logiciel de l'horloge de sable. Et pour finir, un support en Plexiglas est prévu pour maintenir l'horloge droite et la poser sur une armoire ou un bureau. Le contenu du kit de transformation est visible en **figure 2**.

### L'électronique

L'électronique est essentiellement la même que celle de l'horloge de sable. Le transistor T1 qui commandait les moteurs vibrants commande maintenant le module laser. Ce dernier est alimenté en 5 V directement à partir de l'Arduino ; cette solution a été choisie non seulement parce que le courant nécessaire n'est que de 60 mA sous 5 V, mais aussi pour éviter tout risque d'interférence avec les servos.

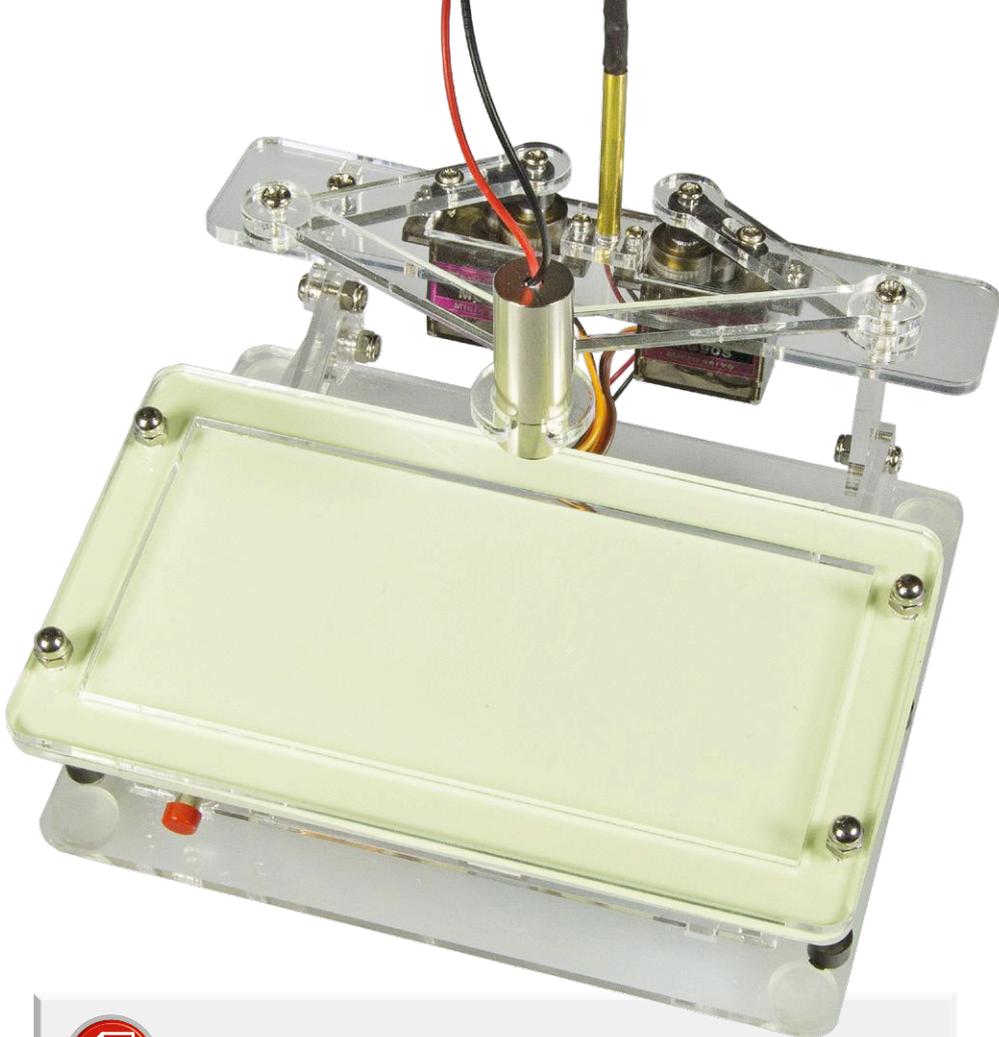
Le transistor T1, commandé par l'Arduino, se comporte comme une source de courant. Sur l'horloge de sable originale, cela permettait de limiter la tension des moteurs vibrants, mais ce schéma ne fonctionne pas avec le module laser. Lorsque la tension d'alimentation de celui-ci diminue, le courant augmente, ce qui mène à un blocage : au démarrage, le laser demande momentanément plus de courant que T1 ne peut en fournir, ce qui fait chuter la tension d'alimentation, ce qui augmente la demande en courant, jusqu'à ce que la tension chute complètement et que le courant à travers T1 soit trop élevé. La solution est de diminuer la valeur de R4, résistance de base de T1 : 1 k $\Omega$  au lieu de 1,8 k $\Omega$ . Lors de la transformation, il faut donc souder une résistance de 2,2 k $\Omega$  en parallèle sur la résistance de 1,8 k $\Omega$  ; la résistance ainsi obtenue sera d'environ 990  $\Omega$ .

Sur la dernière version du circuit imprimé, la valeur de R4 peut être sélectionnée à l'aide d'un pont de soudure. Des ilots sont aussi prévus pour faciliter la connexion du module laser et du bouton-poussoir, ce dernier entre la broche 6 de l'Arduino et la masse.

Le schéma complet est en **figure 3**, avec les modifications clairement indiquées.

### Le module laser

Le module laser du kit de transformation est bâti autour d'une diode laser de couleur violette, longueur d'onde de 405 nm (voir **entractes 2 et 3**). Les recherches de l'ingénieur japonais Shuji Nakamura, de la *Nichia Corporation*, ont permis, en 1993, le développement des LED bleues ; il a d'ailleurs reçu le prix Nobel



## LISTE DES COMPOSANTS

### Résistances (CMS 0805)

R1, R2 R3, R11 = 220  $\Omega$   
 R4 = 1,8 k $\Omega$  \*  
 R5 à R9 = 4,7 k $\Omega$   
 R10 = 100 k $\Omega$   
 R12 = 2,2 k $\Omega$  \*

### Condensateurs

C1 à C4 = 100 nF, CMS 0805 MLCC  
 C5 = 10  $\mu$ F/10 V, CMS 1206 MLCC  
 C6 = 22  $\mu$ F/16 V

### Semi-conducteurs

D1 à D3 = S2J-E3  
 T1 = BCX56  
 IC1 = convertisseur CC/CC 5 V/1 A, SIP3  
 (Würth Elektronik 173 010 578)  
 IC2 = PCF2129A

### Divers

K1 = jeu de barrettes sécables, mâle, pas de 2,54 mm (1 $\times$ 10, 2 $\times$ 8, 1 $\times$ 6)  
 SERVO\_Z, \_L, \_R = barrette sécable coudée, mâle, 3 contacts, pas de 2,54 mm  
 B1 = pile bouton CR2032, avec support pour circuit imprimé  
 Arduino UNO R3, ou équivalent  
 1 bouton-poussoir SPST NO, avec trous de montage de 6 mm  
 1 jeu de fils (rouge/noir) + gaine thermorétractable  
 1 module laser, 12 $\times$ 35 mm, 405 nm, 5 à 10 mW (eBay)

### Composants mécaniques

6 vis M2 $\times$ 10, tête Phillips ou Pozidriv  
 6 vis M2, 5 $\times$ 8, acier zingué, Pozidriv DIN 7985A  
 2 vis M3 $\times$ 8, acier zingué, Pozidriv DIN 7985A  
 15 vis M3 $\times$ 10, acier zingué, Pozidriv DIN 7985A  
 6 écrous hexagonaux M2, acier zingué, DIN 934  
 7 écrous hexagonaux M3, acier zingué, DIN 934  
 4 écrous de blocage M3, hauts, chromés, DIN 1587  
 4 entretoises pour M3, 3 mm, polyamide  
 4 entretoises pour M3, 5 mm,  $\varnothing$  ext. 6 mm, polyamide  
 4 entretoises hexagonales M/F pour M3, 25 mm, laiton nickelé (hauteur totale minimale de 33 mm, p. ex. TME TFM-M3X25/DR213)  
 1 tube en laiton de 35 mm de long,  $\varnothing$  4 mm  $\times$  0,5 mm  
 4 patins adhésifs en caoutchouc  
 3 servos, Tower Pro MG90S ou MG90, avec engrenages métalliques  
 Plexiglas extrudé, 3 mm, transparent, découpé au laser  
 Autocollant phosphorescent, 80 $\times$ 150 mm<sup>2</sup> (cherchez sur eBay avec les mots clés *glow in the dark sticker*)

\* voir texte

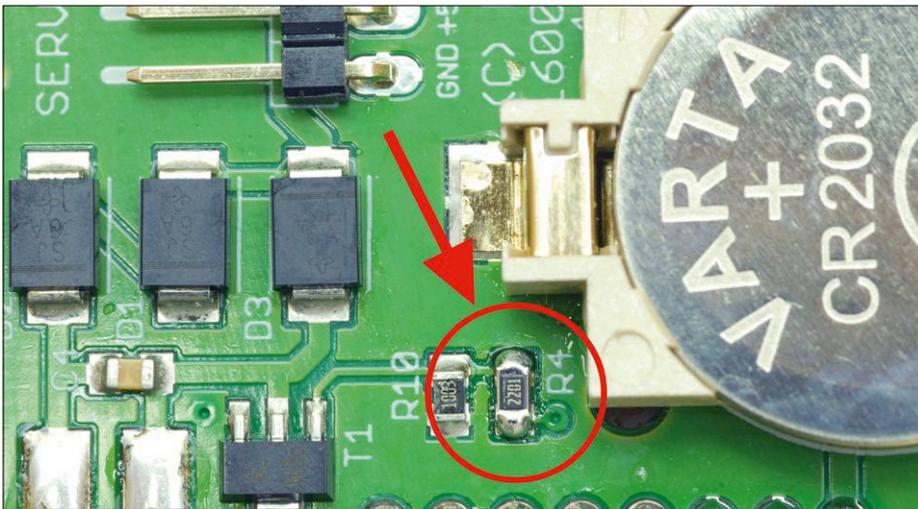


Figure 4. Le circuit imprimé du *shield* Arduino ; l'endroit où la résistance de 2,2 kΩ doit être soudée sur la résistance de 1,8 kΩ est indiqué.

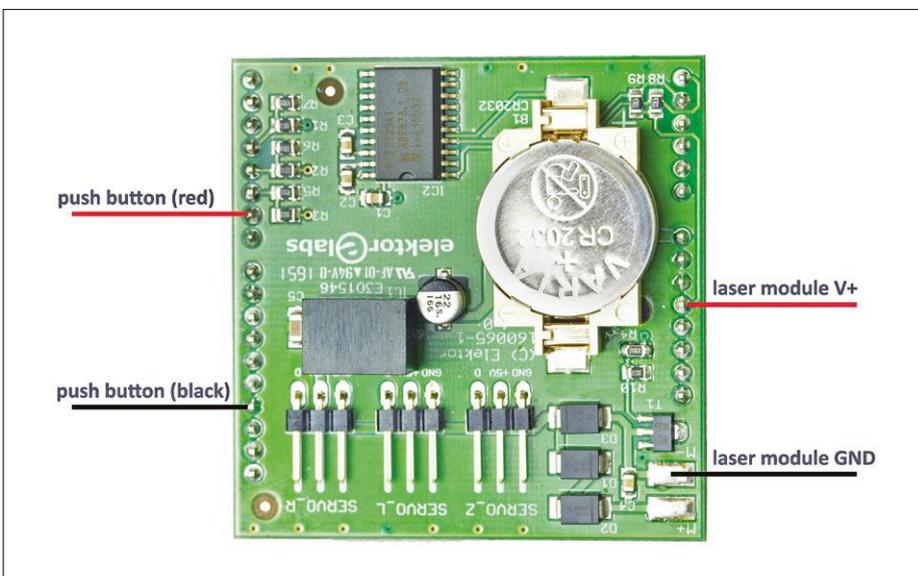
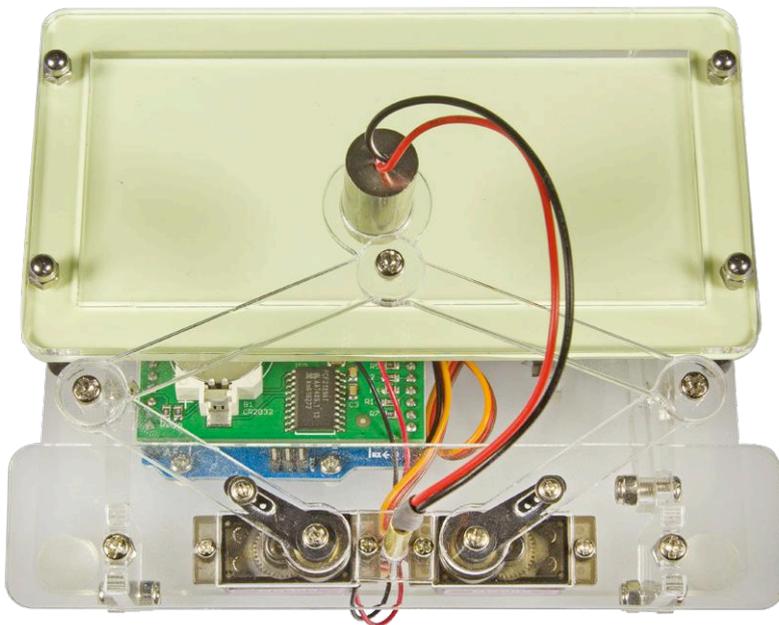


Figure 5. Connexions du bouton-poussoir et du module laser sur le *shield*.

de physique en 2014, avec deux autres chercheurs, Isamu Akasaki et Hiroshi Amano. Quatre ans plus tard, il a mis au point la diode laser « bleue » (la couleur apparaît comme violette) ; elle put être produite en série à partir de 1999, grâce à l'obtention de cristaux ultra purs de nitrure de gallium par le Dr Sylwester Porowski de l'Académie polonaise des sciences. L'application commerciale la plus connue du laser violet est dans les lecteurs/enregistreurs *Blu-Ray*. Les techniques de production s'améliorant sans cesse, on trouve désormais à côté du laser violet (405 nm) un laser bleu violet (450 nm), dont la puissance peut atteindre plusieurs watts. Le prix des diodes laser a aussi chuté de manière spectaculaire, vu leur emploi dans des projecteurs bon marché.

### Le logiciel

Le croquis de l'Arduino est basé sur celui de l'horloge de sable, mais il n'est pas identique : en plus de la commande spécifique du laser, on peut aussi écrire à l'envers. Le servo permettant de relever ou d'abaisser le laser n'a toujours que deux positions possibles. L'ensemble des commandes est donc quelque peu différent, mais la procédure de calibrage est sensiblement la même.

Les mouvements du module laser entre l'écriture de l'heure et celle des minutes sont plus lents, car il est plus lourd, et les bras du pantographe peuvent vibrer. Le nouveau croquis de l'horloge *Laser Time Writer* est disponible sur la page du projet [2] ; la notice d'assemblage reprend aussi l'installation du logiciel et le calibrage.

### L'assemblage

L'assemblage de l'horloge *Laser Time Writer* est semblable à celui de l'horloge de sable, et il est décrit pas à pas dans la notice illustrée ; celle-ci a été conçue non seulement pour la transformation, mais aussi pour le kit complet qui sera bientôt disponible.

Avant d'installer l'autocollant phos-

**DANS L'E-CHOPPE**

→ 160065-71  
kit de l'horloge de sable

→ 160538-71  
kit de transformation en  
*Laser Time Writer*

## ▶ l'heure reste visible longtemps

phorescent, il faut humidifier la plaque de Plexiglas avec de l'eau additionnée d'un peu de liquide vaisselle. Après avoir positionné l'autocollant sur la plaque, il faut faire sortir les éventuelles bulles d'air à l'aide d'un petit morceau de carton rigide. N'oubliez pas non plus d'éponger l'eau avec de l'essuie-tout.

Les fils de connexion entre l'Arduino et les servos doivent être assez longs pour ne pas gêner les déplacements du pantographe.

Le circuit imprimé est en **figure 4**, avec la modification de R4 à nouveau indiquée : soudure d'une résistance de 2,2 k $\Omega$  sur la résistance de 1,8 k $\Omega$ . Les connexions sont indiquées à la **figure 5**, et la **figure 6** montre quelques étapes de l'assemblage.

Vous avez maintenant une bonne idée de ce qui vous attend lors de la transformation de votre horloge de sable. Nous vous souhaitons beaucoup de plaisir ! ◀

(160569 – version française : Jean-Louis Mehren)

### Liens

[1] [www.elektormagazine.fr/160065](http://www.elektormagazine.fr/160065)

[2] [www.elektormagazine.fr/160569](http://www.elektormagazine.fr/160569)

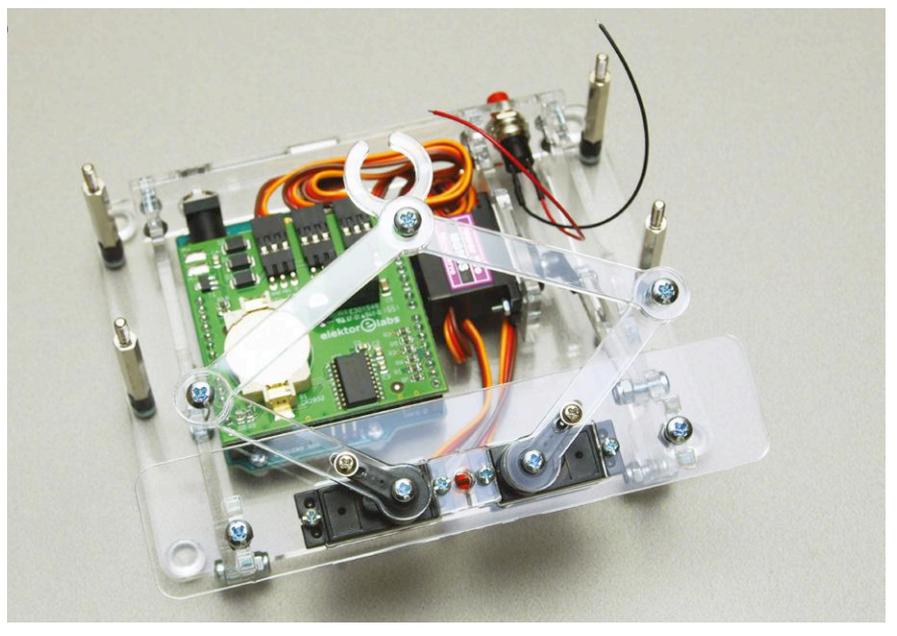
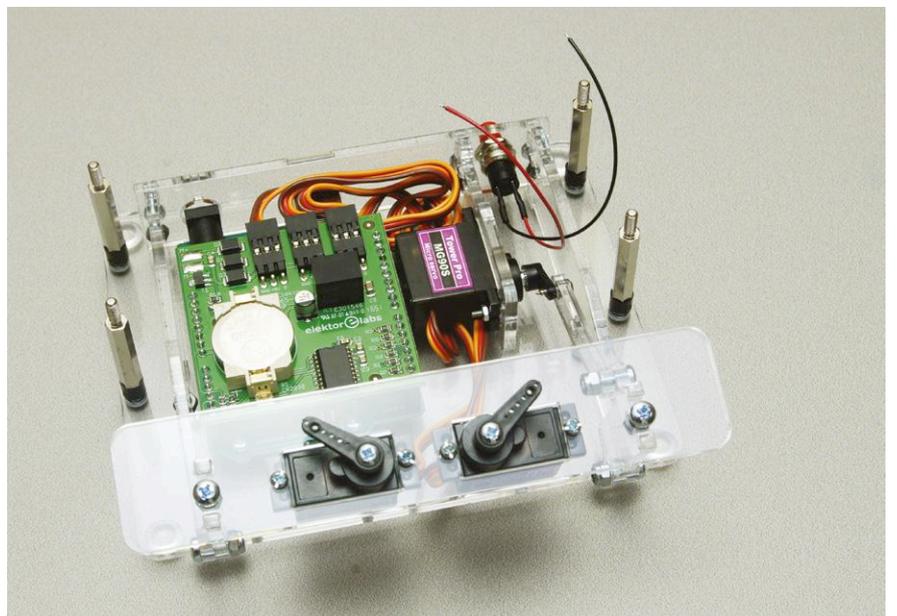
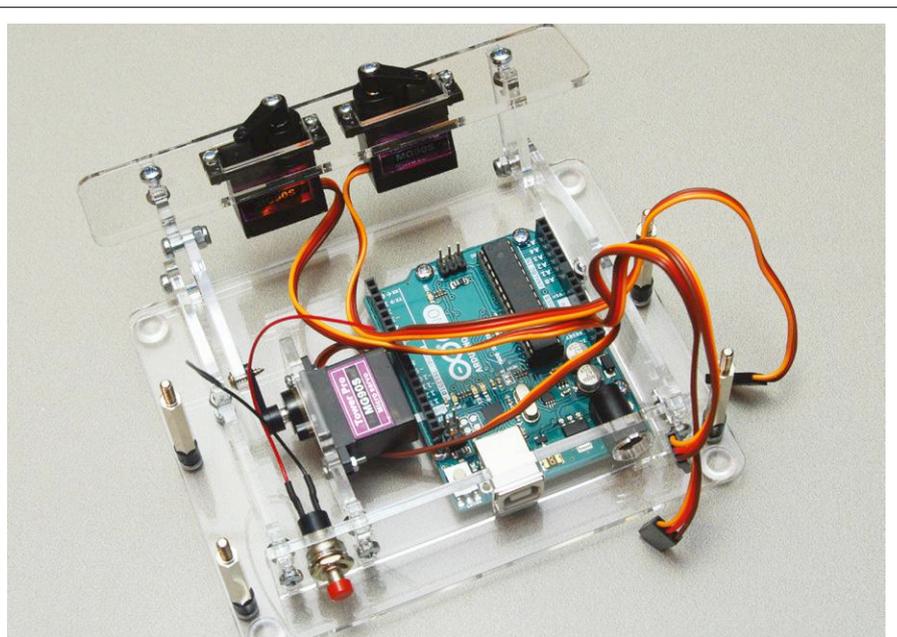


Figure 6. Trois étapes de la transformation de l'horloge de sable en horloge laser

# L'EEBUS arrive

## fin de la tour de Babel pour l'Internet des Objets ?

Tam Hanna

Les maisons connectées sont généralement un salmigondis d'appareils de différents fabricants. Si ces appareils veulent communiquer entre eux, la maîtrise de la variété babylonienne de protocoles et de langages est un travail de Sisyphe. L'Initiative EEBUS fondée en Allemagne est un organe de normalisation qui souhaite apporter une solution à ce problème.



Commençons nos considérations par un aperçu du marché. À part Samsung, LG et Haier, peu d'entreprises peuvent offrir à leurs clients une solution globale allant du sèche-linge à l'ordiphone en passant par le réfrigérateur. Des entreprises plus petites, spécialisées sur un secteur particulier, sont à la traîne face à leurs concurrents mieux capitalisés. Cela crée un déséquilibre sur le marché et enlève aux utilisateurs différentes possibilités lors de la composition de leur « parc d'appareils ». La solution à ces problèmes est la raison d'être de l'Initiative EEBUS e.V., instance de normalisation domiciliée à Cologne (voir l'encadré « Qui est l'EEBus ? »).

### Architecture

L'EEBUS a connu un développement similaire au Bluetooth LE par ex. : il fallait créer une architecture de communication qui soit aussi générique que possible et qui couvre tous les cas d'application (y compris futurs). Une telle norme a également la responsabilité de rassembler « à la même table » des appareils de plusieurs fabricants.

Les bases de données KV (KV = *key/value* ou clé/valeur) sont une approche possible. D'une part, elles sont suffisamment simples techniquement, ce qui signifie qu'elles ne constituent pas un obstacle pour le développeur. D'autre part, la standardisation des clés et/ou des contenus des clés permet de créer une norme assez stricte.

Lorsqu'on parle d'EEBUS, cela on pense immédiatement à SPINE - l'acronyme de *Smart Premises Interoperable Neutral-message Exchange*. Cette norme est fortement orientée vers l'application : le développeur prend en compte les exigences de communication interdispositifs dans l'application en question et décide ensuite de la présentation des messages correspondants.

Observons l'architecture représentée sur la **figure 1**. Au niveau hiérarchique supérieur se trouve l'appareil physique, par exemple une machine à laver, un ordiphone ou une lampe. Dans le domaine de l'électroménager de cuisine, les appareils effectuent plusieurs tâches dans une même « boîte » – comme ces boîtiers sont en général blancs, on parle de « produits blancs ». SPINE a créé pour ces appareils le concept d'entité (*entity*). Une entité décrit une fonction de l'appareil. Dans le cas d'un refroidisseur à vin avec distributeur de glaçons intégré, il s'agit d'une part du refroidisseur lui-même et d'autre part du distributeur de glaçons.

Les différentes caractéristiques qui décrivent chaque fonction d'un appareil se trouvent au niveau le plus bas.

### Une question de format de données

Le Bluetooth LE se différencie du modèle SPINE, dans la mesure où la norme de radio spécifie le format de communication aussi bien physique que logique. SPINE se limite lui au septième niveau du modèle de couches OSI : le processus par lequel les données se déplacent d'un hôte à un autre est du ressort exclusif du développeur. L'EEBUS propose uniquement comme option d'utiliser SHIP, c'est-à-dire *Smart Home IP*, un protocole maison qui s'appuie sur TCP/IP.

L'EEBUS est bien plus strict en ce qui concerne le choix du codage des messages – les données sont au format texte, la syntaxe XML a été retenue (voir encadré « XML »). Cela simplifie le travail d'implémentation, car chaque langage comporte un ou même plusieurs analyseurs XML. Comme la puissance de calcul disponible dans les composants embarqués est toujours plus grande, le surcoût (*overhead*) évidemment énorme, dû à l'utilisation du XML, va fortement diminuer.

## Qui est l'EEBUS ?

L'histoire de l'initiative EEBUS se réfère à l'Initiative E-Energie lancée par le gouvernement fédéral allemand : son objectif était la préparation de l'infrastructure énergétique de l'Allemagne pour relever les défis dus au changement énergétique et à l'Internet des Objets.

L'association a réellement été fondée en 2012 lors de la Foire de Hanovre, et compte au moment de l'impression 65 membres. Son étendue est particulièrement intéressante : en plus des fournisseurs d'énergie, la VDA (association de l'industrie automobile allemande), la VDIK (fédération des constructeurs auto-

mobiles internationaux) pour la branche automobile et quelques fabricants d'appareils électroménagers comme Miele font partie des partenaires.

Le comité EEBUS travaille en suivant un cycle PDCA (*plan, do, check, adjust*) selon Demings : d'abord les documents de spécification créés suite à l'identification de « problèmes » sont convertis en cas d'application qui ouvrent la voie à la spécification technique. Les expériences issues de la pratique ont un impact sur la révision suivante de la norme.

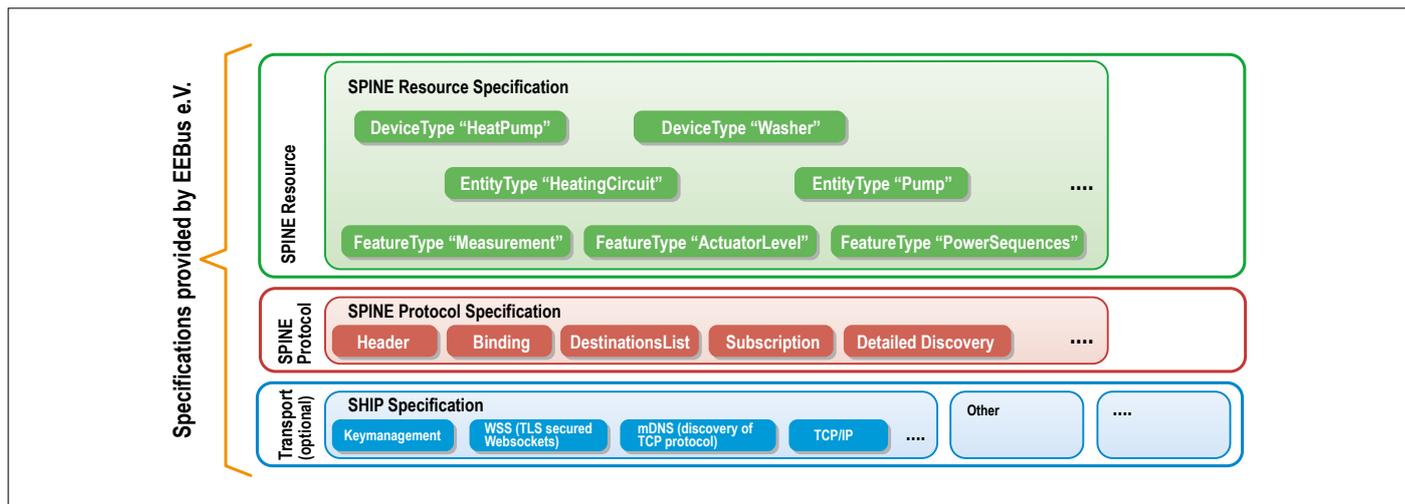


Figure 1. Architecture EEBUS avec trois « niveaux d'adresse ».

### Un exemple

Prenons un exemple simple pour illustrer la construction des messages SPINE : un appareil envoie une valeur de mesure acquise par un capteur, sur demande d'un autre appareil.

Du point de vue technique, tous les messages EEBUS sont en deux parties. Commençons par l'entête représenté à la **figure 2**, composé d'une bonne douzaine de champs différents. Les champs `addressSource` et `addressDestination` forment le chemin qui mène de l'expéditeur au destinataire. On passe normalement à une hiérarchie à trois niveaux : appareil, entité et caractéristique. Dans quelques cas particuliers, on peut

renoncer à indiquer l'adresse pour économiser de la place.

Le champ `CmdClassifier` fixe le rôle du message en transfert. Actuellement, la norme prévoit six types qui sont énumérés dans le tableau.

Les champs restants de l'entête sont plus ou moins « standard » : ils donnent des informations sur la version, le besoin d'une confirmation explicite de la réception et autres fonctions semblables. Comme petit exemple, considérons ici l'entête d'une trame quelconque, mise à disposition par l'organisation de normalisation EEBUS :

## XML

Le modèle SPINE transfère des informations au format XML. Vous pouvez vous représenter le « eXtensible Markup Language » comme une sorte d'HTML pour les ressources : le format XML décrit toutes sortes d'éléments du monde réel, plutôt que le contenu d'une page web. Ce format à 8 bits, peu économique en bande passante et peu maniable, présente un avantage parce qu'il est traité de manière semblable sur toutes les plateformes.

Les fichiers `.xsd` décrivent quelles balises XML peuvent être utilisées (ici dans le cas de SPINE). Ces fichiers (*XML Schema Definition*) utilisent eux-mêmes XML pour décrire la structure de messages XML : ils fixent quels champs sont disponibles

et comment les informations contenues sont structurées, par exemple le type de données « double », etc.

Il est recommandé pour les premiers essais de simplement ouvrir les fichiers XML et XSD dans un éditeur de texte, ces fichiers sont assez faciles à comprendre. De nombreux éditeurs XML sont disponibles pour les utilisateurs avancés ; leur utilisation dépasse le cadre de cet article.

Programmer un analyseur XML n'est pas un travail intéressant pour un électronicien, mieux vaut laisser cette tâche aux développeurs spécialisés. Cherchez XML Parser accompagné du langage de programmation de votre choix, vous trouverez plusieurs analyseurs.

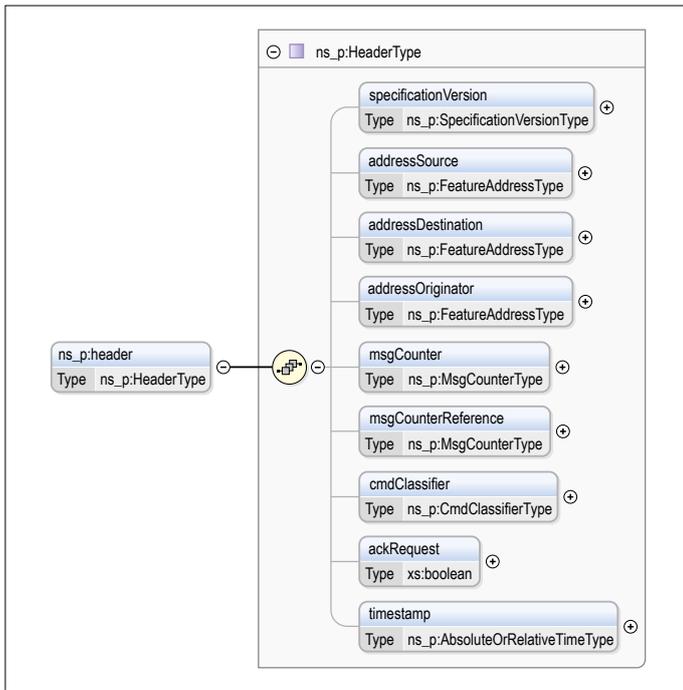


Figure 2. Entête SPINE de plus près.

```

<header>
  <specificationVersion>1.0.0</specificationVersion>
  <addressSource>
    <device>d:_i: 46925_TestTempSensor_1</device>
    <entity>2</entity>
    <entity>1</entity>
    <entity>1</entity>
    <feature>1</feature>
  </addressSource>
  <addressDestination>
    <device>d:_i:46925_TestSHM_1</device>
    <entity>1< /entity>
    <feature>1</feature>
  </addressDestination>
  <msgCounter>15</msgCounter>
  <msgCounterReference>8</msgCounterReference>
  <cmdClassifier>reply</cmdClassifier>
</header>

```

Le rôle des deux paramètres `MessageCounter` est intéressant. Le paramètre `msgCounter` est incrémenté à chaque envoi de paquets – il s’agit d’un ID de message qui identifie clairement chaque « message envoyé au cours d’une courte période ». Le paramètre `MsgCounterRef` permet la corrélation entre message et déclencheur – un message de réponse (*reply*) peut ainsi se référer à la commande traitée par celui-ci.

Les protocoles de communication spécifient souvent leur propre format pour les adresses : dans le cas de SPINE, le deux-points sert de séparateur des différents éléments. La chaîne de caractères `d:` sert d’entête de protocole, le `_i` suivant permet de stocker d’autres informations spécifiques au fabricant. Le but de la structure d’adressage est de produire une chaîne globalement unique qui identifie un appareil en particulier.

### Fonctions et filtres

La seconde partie de la trame est le contenu à proprement parler du message : elle est normalement disponible sous la forme d’un élément `payload`. Sa déclaration commence par un piège : même si le protocole permet la présence de plusieurs balises `cmd` dans un message, les implémentations n’évaluent normalement que la couche supérieure.

```

<payload>
  <cmd>
    <function>measurementListData</function>
    <filter>
      <cmdControl>
        <partial/>
      </cmdControl>
    </filter>

```

Tableau : CmdClassifier

CmdClassifier	
fixe le rôle du message en transfert.	
Désignation	Rôle
<code>read</code>	Commande qui pousse le destinataire à livrer des données placées sous sa gestion, p. ex. « Combien de glaçons avons-nous ? ».
<code>write</code>	Commande qui contraint le destinataire à traiter une valeur.
<code>call</code>	Commande qui pousse le destinataire à traiter la commande ou l’exécution d’une action, p. ex. « Jeter cinq glaçons en forme d’étoile ».
<code>reply</code>	Message qui représente la réponse à une commande <code>read</code> , p. ex. « J’ai six glaçons en forme d’étoile ».
<code>notify</code>	Message envoyé par le propriétaire des données à un client intéressé, p. ex. « Attention, quatre glaçons ont été retirés ».
<code>result</code>	Message qui confirme la réception réussie ou non réussie ou le traitement d’une commande, p. ex. « Non, je ne peux pas traiter le message précédent ».

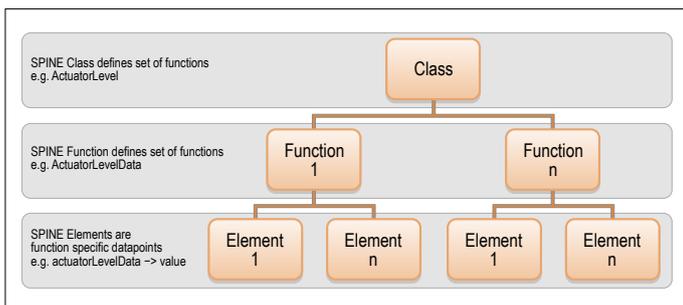


Figure 3. Fonctions et éléments sont rassemblés en classes.

En premier se trouve la déclaration de la tâche à exécuter : la présence d'une balise `function` suppose en outre un filtre qui limite la zone du fichier associée à la commande en question. L'EEBUS fournit un document intitulé *Resource Specification* qui spécifie la plupart des fonctions ; nous y reviendrons plus en détail. SPINE prend les développeurs par la main avec une sorte de construction de classes qui décrit comment les éléments sont groupés (**fig. 3**). Normalement la classe elle-même n'est pas citée dans le message, mais elle sert plutôt à décrire et grouper des fonctions et leurs éléments. Selon la fonction, il y a alors une ou plusieurs familles d'éléments qui décrivent les données utiles.

### Les données

Maintenant suivent les éléments du message. Ils mettent à disposition les informations réelles. Dans notre cas, cela serait une valeur mesurée par une sonde :

```
<measurementListData>
<measurementData>
  <measurementId>1</measurementId>
  <valueType>value</valueType>
  <timestamp>2015-1105T10:14:00.0Z</timestamp>
  <value>
    <number>22</number>
  </value>
  <valueSource>measuredValue</valueSource>
</measurementData>
</measurementListData>
</cmd>
</payload>
```

### Requête

La requête pour l'appareil qui lit le capteur pourrait ressembler à ce qui comme suit :

```
<cmd>
<function>measurementListData</function>
<filter>

  <cmdControl>
    <partial/>
  </cmdControl>
  <measurementListDataSelectors>
    <measurementId>5</measurementId>
    <valueType>minValue</valueType>
  </measurementListDataSelectors>

</filter>

<measurementListData/>

</cmd>
```

Avec la requête, l'information elle-même est placée dans le filtre. Il n'y a pas d'éléments de données.

### De la normalisation...

La structure décrite ici est jusqu'ici techniquement fondée, mais



### Un événement oublié ?

Vous organisez une conférence, un salon... ou bien vous participez à un séminaire ou tout autre événement qui aurait sa place ici, partagez cette information avec tous les lecteurs. Envoyez-nous tous les détails à [redaction@elektor.fr](mailto:redaction@elektor.fr).

## février 2018

### ◆ Mondial des métiers

01 au 04/02 – Lyon  
[www.mondial-metiers.com](http://www.mondial-metiers.com)

### ◆ PARIS MANGA & SCI-FI SHOW

03 au 04/02 – Paris  
[www.parismanga.fr/](http://www.parismanga.fr/)

### ◆ Salon de l'analyse industrielle

06 au 07/02 – Paris  
[www.analyse-industrielle.fr/](http://www.analyse-industrielle.fr/)

### ◆ OPTRO 2018

06 au 08/02 – Paris  
[www.optro2018.com/](http://www.optro2018.com/)

### ◆ Biogaz Europe

07 au 08/02 – Nantes  
[www.biogaz-europe.com](http://www.biogaz-europe.com)

### ◆ C!PRINT

07 au 08/02 – Lyon  
[salon-cprint.com](http://salon-cprint.com)

### ◆ ENOVA LYON

07 au 08/02 – Lyon  
[www.enova-event.com/](http://www.enova-event.com/)

**Elektor  
partenaire presse  
du salon !**

### ◆ Salons des entrepreneurs

07 au 08/02 – Paris  
[www.salondesentrepreneurs.com/](http://www.salondesentrepreneurs.com/)

### ◆ Retromobile

07 au 11/02 – Paris  
[www.retromobile.fr](http://www.retromobile.fr)

### ◆ Colloque du Syndicat des Énergies Renouvelables

08/02 – Paris  
[ser-evenements.com](http://ser-evenements.com)

### ◆ Techinnov

08/02 – Paris  
[techinnov.events](http://techinnov.events)

### ◆ Maker Faire Lille

09 au 11/02 – Lille  
[lille.makerfaire.com/](http://lille.makerfaire.com/)



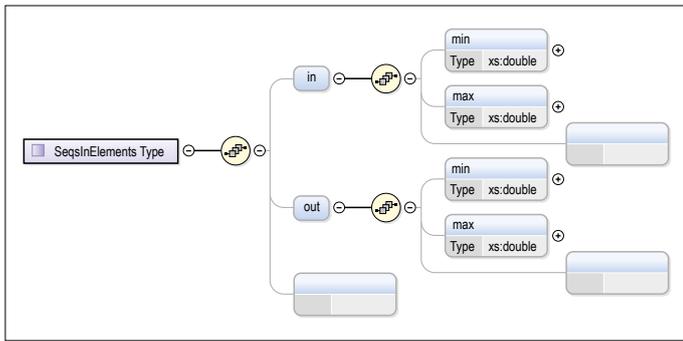


Figure 4. Les analyseurs doivent compter à certains endroits sur des extensions.

ne résout toutefois pas le problème fondamental décrit dans l'introduction : chaque fabricant pourrait réaliser sa propre combinaison d'entités et de caractéristiques, ce qui réduirait tous ces efforts à néant.

Chez EEBUS, on s'oriente ici vers les modèles de Bluetooth SIG. L'entreprise force dès le départ les développeurs d'analyseurs et d'autres systèmes de traitement de données à préparer leurs produits pour des versions de protocole ultérieures. Le **figure 4** montre bien que des extensions de la norme doivent avoir lieu en ajoutant des enfants supplémentaires à certains points dans le modèle de données. Si un analyseur rencontre un élément inconnu, l'activité d'analyse poursuit avec le prochain élément connu.

L'idée sous-jacente est simple : si tous les fabricants d'un certain bien transposent les parties spécifiques de la norme, une interopérabilité « de base » est possible. Les méthodes de reconnaissance décrites ci-dessus permettent alors la reconnaissance d'extensions particulières à un fabricant qui autorisent d'autres fonctions (s'il y en a).

### ... à la reconnaissance

Mais comment des appareils peuvent-ils analyser et se connecter à des interlocuteurs inconnus ? Découvrons-le à présent. Ce processus est intégré dans le protocole sous la notion de *Functional Commissioning* : la procédure nécessaire pour annoncer une *Notification* se trouve également dans la norme.

La spécification impose que la caractéristique zéro de l'entité zéro – donc adresse de caractéristique 0 et adresse d'entité 0 – permette la mise en œuvre d'un *NodeManagement*. Une analyse complète d'un quelconque appareil éloigné s'effectue en envoyant une commande **READ** qui s'adresse à la fonction `nodeManagementDetailedDiscoveryData`. Le destinataire réagit à l'envoi d'une structure de classe relativement complexe qui représente une sorte d'image complète des fonctions et entités disponibles.

Du côté de l'organe de normalisation, il n'y a pas de directives sur la manière dont l'identification d'un appareil à connecter doit avoir lieu – il y a dans la norme une mention larmoyante à « a rather intelligent implementation » (*une mise en œuvre plutôt intelligente*) (page 57).

### Comprendre la documentation

L'organisation derrière l'EEBUS propose en téléchargement libre les spécifications après saisie d'une adresse de courriel : une

adhésion à l'instance de normalisation n'est pas nécessaire. Dans le fichier *EEBUS\_SPINE\_TR\_Introduction.pdf* se trouve un aperçu de la norme EEBUS. Si vous souhaitez vous attaquer sérieusement à l'EEBUS, vous devrez imprimer ce document. D'autres informations techniques sont à disposition dans les fichiers *EEBUS\_SPINE\_TS\_ProtocolSpecification.pdf* et *EEBUS\_SPINE\_TS\_ResourceSpecification.pdf*. Vous aurez besoin du second document pour analyser un concept décrit dans le premier.

Enfin il y a un groupe de fichiers `.xsd`, rassemblés dans une archive Zip. Ces fichiers désignés comme *définition de schéma XML* contiennent une représentation graphique des différents messages ; différents outils permettent de les visualiser. Quelques EDI basés sur l'XML peuvent aussi utiliser ces fichiers pour vérifier le code créé – la recherche d'erreurs est bien plus simple, si on sait que les balises sont conformes à la norme. N'oubliez pas toutefois que les fichiers XSD sont souvent plus tolérants que des mises en œuvre dans la vie réelle.

### Résumé

Du point de vue technique, EEBUS et SPINE sont intéressants en raison de leur immense flexibilité : il y a peu d'appareils qui ne puissent être représentés dans une base de données KV. Grâce à la similitude avec le Bluetooth LE, les développeurs connaissent les grandes lignes des concepts sur lesquels repose ce système.

Toutefois, répétons-le : SPINE n'offre qu'un cadre pour une communication entre appareils de différents fabricants, mais il ne fournit pas de directives d'implémentation spécifiques pour des types d'appareils particuliers.

Pour garantir une vraie interopérabilité, les fabricants doivent encore négocier pour chaque classe d'appareils quels messages SPINE sont utilisés et sous quelle forme. Ces efforts viennent de commencer ; les premières normes pour des lave-linges et des lave-vaisselles sont apparues.

L'avenir nous dira si chaque appareil EEBUS travaillera réellement en collaboration avec les autres (comme dans le cas du GPIB). ◀

(150583 – version française : Thierry Destinobles)

### Liens

- [1] Téléchargement des spécifications : [www.eebus.org/download-standard/](http://www.eebus.org/download-standard/)
- [2] Liste des membres : [www.eebus.org/initiative/mitglieder/](http://www.eebus.org/initiative/mitglieder/)
- [3] Modèles de données et détails : [www.eebus.org/technologie/datenmodell/](http://www.eebus.org/technologie/datenmodell/)

## Il faut toujours une connexion

Considérez que la procédure décrite ici suppose la présence d'une connexion de transport entre les appareils – elle ne remplace pas DHCP, *broadcasts* et autres processus de reconnaissance de niveau inférieur.

# alimentation solaire pour répéteur Wi-Fi

au choix : version analogique ou numérique



J'ai développé il y a quelques années une caméra de surveillance à alimentation solaire et transmission des données par Wi-Fi ou carte SIM pour réseau GSM. Cependant, certains utilisateurs désiraient un modèle Wi-Fi avec plus de portée. J'ai donc réfléchi à un répéteur Wi-Fi alimenté par un panneau solaire.

**Laurent Labbe** (Shenzhen, Chine)

## Les grandes lignes

Pour la fonction de répéteur, j'utilise un tout petit router de marque *tp link* (comme le TL-WR802L) qui présente deux avantages : une faible consommation (env. 200 mA ; 140 mA mesurés en mode répéteur) et l'absence de bouton marche/arrêt. Ce boîtier riquiqui a pour

fonction principale le routage d'un réseau Ethernet vers le Wi-Fi, mais il peut être configuré en répéteur. J'ai testé plusieurs versions de ce modèle, cela fonctionne parfaitement – sauf avec les modèles achetés en Chine (dont la ROM est exclusivement en chinois, et non flashable en anglais). Mon alimentation solaire dispose de deux modes de fonctionnement : le premier mode est purement analogique, il s'agit de deux comparateurs ; le second mode

est enrichi par un zeste de numérique, grâce à un petit microcontrôleur à huit broches de Microchip, le 12F1572. La batterie utilisée est un assemblage de deux cellules cylindriques 18650 (Li-Ion), montées en parallèle (elles doivent donc être identiques !). Durant ma quête (chinoise) de cellules de grande capacité, j'ai eu pas mal de surprises. Il y a énormément de « fausses » cellules avec des capacités attractives. Finale-

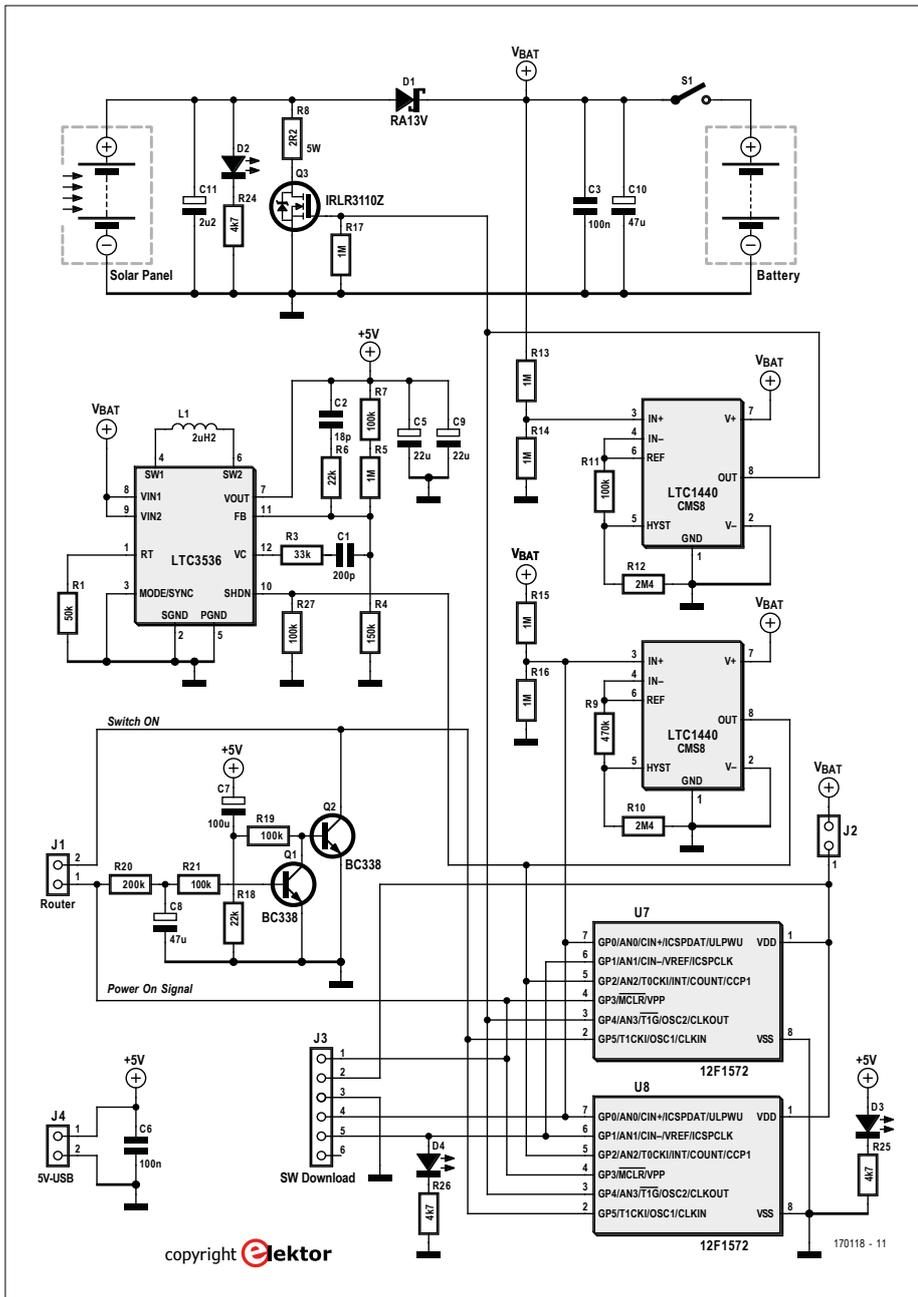


Figure 1. Schéma de l'alimentation solaire du répéteur Wi-Fi.

ment, j'ai trouvé (et mesuré) des cellules de 3000 mAh. Un truc pour repérer les fausses, il faut les peser. Choisissez toujours les plus lourdes... Maintenant, avant d'acheter, je demande toujours le poids des cellules. Pour monter les cellules en parallèle, j'ai modifié un adaptateur de batteries, initialement prévu pour une mise en série. Pour charger la batterie, j'ai opté pour

une solution simple, mais peu orthodoxe. Le panneau solaire charge directement la batterie via la diode Schottky D1. Cette diode Schottky est une RA13V (Sanken Electric) qui accepte 2 A. Vous pouvez la remplacer en tenant compte du courant maximal de votre panneau solaire. Enfin, le routeur est alimenté par un convertisseur DC-DC. J'ai beaucoup cherché et j'ai testé plusieurs circuits

intégrés, mon choix s'est porté sur le LTC3536. Il peut délivrer 1 A, et surtout, il a un véritable mode « power OFF ». En effet, beaucoup de convertisseurs DC-DC à l'arrêt coupent bien la MLI, mais un courant circule toujours au travers de la bobine. Il n'y a donc jamais 0 V en sortie. La LED D3 s'allume quand la tension de 5 V est présente en sortie, sur la prise USB à raccorder au répéteur.

Le panneau solaire possède les caractéristiques suivantes : tension de sortie = 5 V ; courant de sortie = 1 A ; puissance de sortie = 5 W. La LED D2 s'allume quand le panneau solaire débite la tension de 5 V et charge la batterie. Bien entendu le montage ne fonctionnera correctement que si le panneau bénéficie du maximum d'ensoleillement. Le répéteur tp link consomme environ 200 mA et donc l'autonomie avec la batterie chargée est de 30 h.

Passons aux détails du fonctionnement.

### Mode purement analogique avec les comparateurs

Les deux comparateurs sont des LTC1440 (fig. 1), une version améliorée, mais compatible broche à broche du MAX931. Avant j'utilisais toujours le MAX931, mais j'ai eu des soucis (certains ont grillé), j'ai donc basculé sur le LTC qui en plus est moins cher et semble moins sensible.

Le premier LTC1440 compare la tension de la batterie à la valeur de 4,1 V. Dès que la tension dépasse cette valeur (la batterie est alors en fin de charge), le comparateur provoque la fermeture de Q3 qui shunte le panneau solaire avec la résistance de puissance R8. Bien entendu les caractéristiques de R8 dépendent du courant maximal débité par le panneau ( $P = R8 \times I_{max} \times I_{max}$  ;  $R8 = 4,6 \text{ V}/I_{max}$ ). Ce système permet d'éviter l'ajout d'un transistor (donc d'une chute de tension et d'un circuit de commande). À cause de l'hystérésis, la charge redémarre à 3,9 V.

Le second comparateur examine la tension de la batterie. Il met en route le convertisseur DC-DC (broche ShutDown) quand la tension est supérieure à 3,8 V et l'arrête quand elle est inférieure à 2,9 V. Ces seuils sont déterminés par les résistances R13-R14-R15-R16 (voir tableau 1). Les hystérésis sont déterminées par les valeurs de R9 (470 kΩ) et R11 (100 kΩ).

Pourquoi utiliser deux comparateurs séparés et pas un modèle double ?

Tableau 1. Résistances R13-R4-R15-R16 en fonction du mode de fonctionnement.				
Variante	R13	R14	R15	R16
Mode analogique	2,4 MΩ	1 MΩ	1,8 MΩ	1 MΩ
Mode numérique	non montée	non montée	1 MΩ 1%	1 MΩ 1%

Parce que la tension de référence doit être connectée à la même entrée (IN-) alors que sur un comparateur double, en interne, la référence est sur IN- et IN+.

### Mode numérique avec le $\mu\text{C}$

Dans ce mode, le logiciel reprend les tâches des deux comparateurs : marche/arrêt de la charge et marche/arrêt du convertisseur DC-DC. Toutefois la LED D4 indique l'état de la charge par le nombre de flashes.

J'ai choisi le 12F1572 pour plusieurs raisons : faible consommation, mais surtout une tension de référence interne qui permet de calibrer le convertisseur analogique-numérique. D'habitude, on utilise la tension d'alimentation, mais je ne voulais pas ajouter de régulateur pour le  $\mu\text{C}$ . La tension de la batterie est amenée sur le pont R15-R16 (1 M $\Omega$  chacune, de préférence à 1%). Attention : la tension à mesurer ne doit jamais dépasser la tension interne de référence, fixée ici à 2,048 V ! Avec le pont diviseur par deux, la tension de batterie maximale mesurable est donc de 4,096 V, ce sera donc la tension maximale de charge. Dans le logiciel, il est possible de définir un rapport 1/3 au lieu de 1/2, mais il faut recompiler et aussi redéfinir R15-R16, en remplaçant R15 par deux résistances de 1 M $\Omega$  par exemple.

Le programme est écrit en C et compilé avec la version gratuite de MikroC qui est limitée en taille de code. Le circuit imprimé permet l'utilisation du 12F1572 soit en boîtier DIP, soit en boîtier SO.

### Option routeur & point d'accès Wi-Fi

Sur le schéma, il y a tout un groupe de composants autour des transistors Q1 et Q2. Cet aréopage est nécessaire à un mode de fonctionnement particulier : le *MiFi*, l'appareil est à la fois routeur et point d'accès Wi-Fi. Dans ce cas, il faut appuyer sur un bouton pour activer le routeur. Sur les deux types de routeurs MiFi que j'ai testés (Huawei et Alcatel), la mise en marche est déclenchée par la mise à la masse d'un signal. En démontant le routeur MiFi et en soudant un fil sur le bouton marche/arrêt, on peut faire démarrer le routeur par une impulsion à la masse. Un second fil est nécessaire pour détecter la tension d'alimentation du routeur (3 ou 5 V quand le routeur a démarré). Cela permet, via le circuit « retard » C8/R20 et Q1, d'empêcher la mise à la masse qui simule une pression

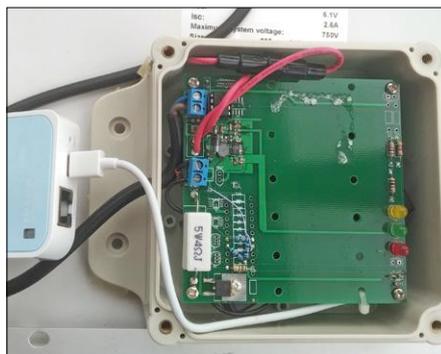


Fig. 2. Carte câblée.

du bouton Marche. Il y a deux raisons à cette fonction. Certains routeurs n'apprécient pas une pression trop longue sur le bouton (définie ici par le couple R18/C7) et donc le signal va remonter au démarrage du routeur. La seconde raison est plus importante. Le routeur MiFi a une batterie intégrée. Il peut donc arriver que le chargeur solaire est déchargé. Si le routeur est toujours en marche lorsque le chargeur solaire met en service le 5 V, la simulation du bouton Marche sera prise pour un arrêt par le routeur. Avec Q1 bloqué, il n'y aura pas d'impulsion sur le signal du bouton Marche et le routeur continuera à fonctionner normalement. En mode purement analogique, la mise en route du convertisseur DC-DC provoque la conduction de Q2 et donc la mise à la masse du signal de démarrage pendant une durée définie par C7-R18. Après le démarrage du routeur, le signal est relâché pendant une durée déterminée par R20-C8.

En mode numérique, le logiciel s'occupe de tout. Il faut compiler le programme soit en variante répéteur (pas besoin de signal de démarrage), soit en routeur (avec signal de démarrage).

### Le logiciel

Après les initialisations de rigueur, le code fixe le chien de garde à 2 s. C'est encore un avantage de ce  $\mu\text{C}$  qui permet un ajustement dynamique du chien de garde.

La boucle principale commence par une lecture de la tension de la batterie. Comme je n'ai pas réussi à utiliser la fonction `adc_read` de MikroC, j'ai écrit la routine (`adc_read_ll`). Le FVR (*Fix Voltage Regulator*) est réglé sur ON uniquement pendant la lecture, pour économiser du courant.



Fig. 3. Montage de l'ensemble au dos du panneau solaire.

En fonction de la tension de la batterie, relevée toutes les 2 s (chien de garde), la LED va clignoter pour indiquer l'état de charge :

- **1 flash** : batterie basse. Attente de recharge. Répéteur arrêté.
- **2 flashes** : batterie en dessous du seuil de redémarrage du convertisseur DC-DC.
- **3 flashes** : batterie au-dessus du seuil de démarrage du convertisseur DC-DC. La tension de 5 V est présente sur la prise USB.
- **4 flashes** : batterie complètement chargée. Activation de la dérivation du courant du panneau solaire dans la résistance de puissance.

Dans le cas de l'utilisation en mode routeur avec simulation de la pression du bouton marche-arrêt, il y a une vérification régulière que le routeur est bien démarré. Dans le cas contraire, une nouvelle impulsion de démarrage sera renvoyée. Cela peut être utile dans le cas d'un arrêt automatique du routeur (cela s'est produit avec le Huawei).

Je programme le PIC avec un PCKit 3.5 et un petit circuit imprimé d'adaptation pour enficher le composant. Puis je le place sur le circuit final. On peut aussi programmer le PIC directement en le laissant sur la carte. J2 et J3 servent à la programmation du  $\mu\text{C}$  sur la carte. Il faut retirer la connexion entre les deux broches de J2 pour programmer. En mode normal, la connexion doit être permanente. J3 est une extension PickIt 3.5. Le logiciel est disponible en [1].

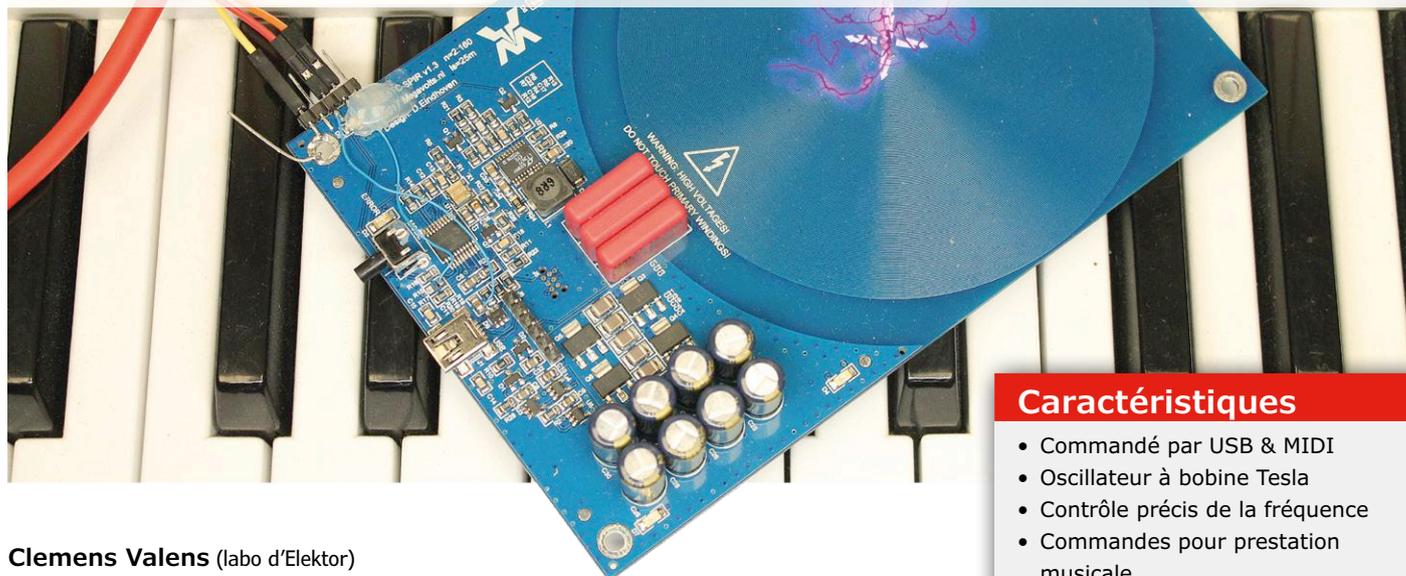
(160570)

### Lien

- [1] Page de l'article : [www.elektormagazine.fr/160570](http://www.elektormagazine.fr/160570)

# de la micro-bobine Tesla au synthétiseur MIDI

## Zeus se la joue Travolta



### Caractéristiques

- Commandé par USB & MIDI
- Oscillateur à bobine Tesla
- Contrôle précis de la fréquence
- Commandes pour prestation musicale

Clemens Valens (labo d'Elektor)

Telle un mini-Zeus en discothèque, la micro-bobine imprimée Tesla du numéro de nov.-déc. 2017 [1] darde des étincelles crépitantes au-dessus des pistes. Le projet est formidable, mais les fréquences d'impulsion limitées à 5, 10 et 20 Hz en restreignent l'utilité. En transformant le module en synthétiseur à interface MIDI, notre mini-Zeus va réellement pouvoir se la jouer Travolta.

En 2010, le groupe américain ArcAttack fit une prestation scénique ébouriffante sur le plateau de l'émission *America's Got Talent* en utilisant des bobines Tesla musicales. ArcAttack se sert de bobines de plusieurs mètres de hauteur (fig. 1), mais le concept peut être appliqué à notre bobine imprimée et à son « aiguille » de 3 cm.

Ce projet transforme la micro-bobine Tesla en synthétiseur à commandes MIDI, mais seule la hauteur du son sera réglée puisque nous n'avons ici aucun moyen de numériser le son, autrement dit aucun moyen de l'envoyer au micro-contrôleur pour le traiter. Pas de filtres élaborés ici donc, mais des paramètres de fréquence ajustables en temps réel au moyen de commandes MIDI :

- profondeur et vitesse de vibrato (modulation de fréquence) ;
- modulation de la hauteur (décalage de fréquence) ;
- portamento (glissement d'une hauteur de note vers une autre).

Les données MIDI peuvent être transmises au synthétiseur via USB ou, après ajout d'un adaptateur MIDI approprié (j'y reviendrai), via un bus MIDI classique. Commençons par l'indispensable, le fonctionnement du logiciel de la micro-bobine Tesla.

### Les petites tâches de Smalltask

La version originale du micrologiciel de la micro-bobine Tesla repose sur un système d'exploitation en temps réel baptisé

Smalltask. Grosso modo, Smalltask est un SE multitâche coopératif dans lequel un ordonnanceur parcourt la liste des tâches chaque fois qu'il le peut, et lance une par une celles devant être exécutées à un instant donné. Une tâche s'exécute aussi longtemps qu'elle le souhaite, et lorsqu'elle est terminée l'ordonnanceur s'occupe de lancer la tâche planifiée suivante. Si une tâche ne se termine jamais ou pas assez rapidement, elle bloque le reste du programme, tout comme elle bloquerait un programme monotâche. Trois tâches tournent en permanence dans le micrologiciel de la micro-bobine :

1. `Led_task` pour faire clignoter les LED de façon plaisante ;
2. `Check_switch` pour détecter la

pression sur le bouton de sélection du mode de fonctionnement et mesurer la durée de cette pression ;

3. **Control\_task**, un automate fini supervisant la séquence de mise sous tension, les modes d'impulsion de 5, 10 ou 20 Hz, ainsi que la gamme.

Deux autres tâches de la liste, **melody\_task** et **error\_blink**, ne sont exécutées que sur demande. La première est lancée lorsque la gamme doit être jouée, la seconde lorsque la LED rouge doit être éteinte à la sortie du mode obscurité ou haute tension.

L'automate fini est lancé à la mise sous tension. S'il « sent » que tout va bien, il lance les tâches liées aux LED et au poussoir, puis entre dans une sorte de veille. Une pression sur le poussoir le remet en mode actif, puis il se remet en mode inactif.

### Faire des étincelles, c'est simple comme le silex

Comment le micrologiciel déclenche-t-il la production des étincelles ? Très simplement : par une courte impulsion sur la broche PWREN (IC5, port RC4, broche 6). En mode 5 Hz, la durée des impulsions est de 1 500 µs, en mode 10 Hz de 750 µs, et en mode 20 Hz elle vaut 330 µs. Le mode haute tension double ces durées et met la broche MODE au niveau bas (IC5, port RC2, broche 14). Dans le cas des notes de musique, la durée des impulsions est obtenue avec un rapport cyclique de 0,7 %, ce qui signifie que pour la note C2 (Do1) de 65,41 Hz, la durée de l'impulsion devrait être de 107 µs. Le code, quelque peu conservateur, utilise en fait 100 µs. Ici aussi le mode haute tension double la durée des impulsions et met au niveau bas la broche MODE.

Étonnamment, c'est tout. Ni fonctions de commande compliquées ici, ni consignes de synchronisation strictes. En fait, un temporisateur 555 et quelques commutateurs pourraient remplacer le PIC.

### Deux timers pour le générateur d'impulsions

L'objectif est de créer de façon logicielle un générateur d'impulsions de fréquence réglable, commandé par MIDI et dont la durée des impulsions est ajustable. À vrai dire le système d'exploitation Smalltask n'était pas indispensable à ce projet, mais le garder laisse à disposition les fonctions de la micro-bobine. Si vous

n'avez pas de données MIDI, vous pourrez ainsi toujours épater la galerie avec des bouquets d'étincelles crépitantes et zébrantes.

J'ai utilisé deux temporisateurs/compteurs (*timers*) pour l'oscillateur : Timer1 pour la fréquence, Timer2 pour la durée des impulsions. Chaque fois que Timer1 est déclenché, c.-à-d. au terme d'une période, Timer2 est lancé pour synchroniser la durée des impulsions. Smalltask utilise aussi un *timer* (Timer0) déclenché toutes les 125 µs (8 kHz), mais celui-ci n'est pas assez précis pour la production de notes : à 110 Hz (A2, ou La1), l'erreur dépasserait déjà 1 %. Timer1 est cadencé par une source de 1,5 MHz, soit une résolution de 667 ns qui maintient l'erreur sur la hauteur en deçà de 0,5 %, et ce sur une plage de 0 à 8 kHz (même si nous n'irons pas si haut). Timer2 a la même résolution, la limite supérieure de fréquence étant fixée à plus de 10 kHz pour respecter le rapport cyclique de 0,7 %.

### Réception des données MIDI

Les données MIDI sont asynchrones et voyagent en groupe. Parfois elles débarquent en masse, la plupart du temps le quai reste vide. Le débit est de 31 250 bits/s et le traitement d'un octet prend 320 µs. La réception des données MIDI aurait donc pu se faire par scrutation de l'UART du PIC dans un *thread* Smalltask, mais comme cet UART peut produire une interruption à chaque réception d'un octet, une réception par interruption s'avère préférable, car beau-

## INFOS SUR LE PROJET

Bobine Tesla
synthétiseur

musique
MIDI
système

d'exploitation en temps réel

▮

débutant

➔ **connaisseur**

expert

🕒

env. 2 h

🔧

micro-bobine imprimée  
Tesla, MPLAB X, program-  
mateur PIC, clavier MIDI

€

env. 10 €

coup moins gourmande en temps de traitement.

Ceci dit je ne me suis pas entièrement passé de Smalltask, j'ai ajouté une tâche *worker* de fréquence 1 kHz pour synchroniser les commandes MIDI sur l'oscillateur d'impulsions. Cette tâche reçoit un message lorsqu'un paquet MIDI a été reçu (jusqu'à 3 octets s'il y a l'octet d'état). La commande MIDI est ensuite analysée et exécutée. L'état « exécution » (*running*) – lorsque les données MIDI sont envoyées sans octet d'état – est pris en charge ; le sens actif – lorsque les octets

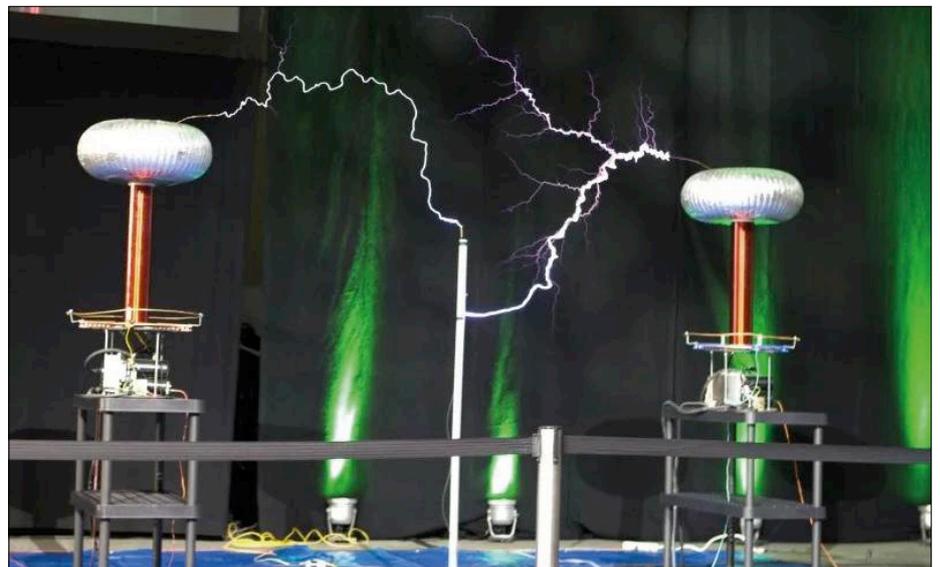


Figure 1. Nous avons eu la chance de rencontrer ArcAttack en 2010 lors du salon NIWeek à Austin, Texas.

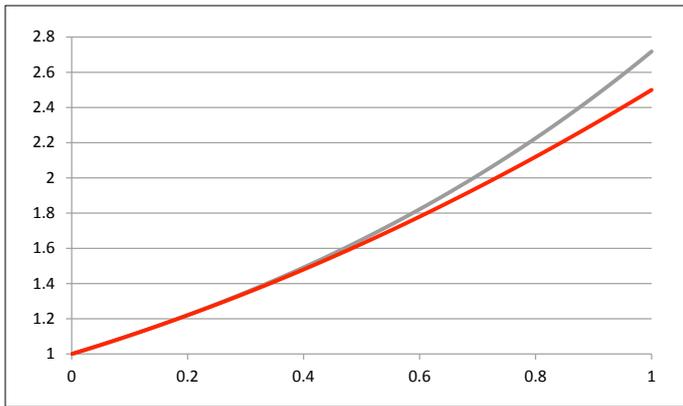


Figure 2. La fonction  $e^x$  (en gris) et notre approximation naïve (en rouge) divergent rapidement pour les grandes valeurs de  $x$ .

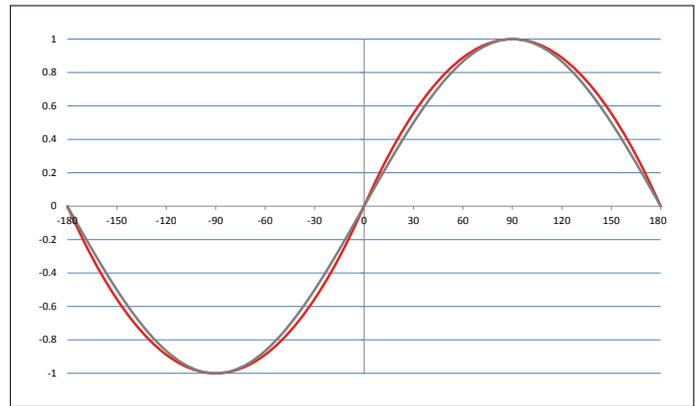


Figure 3. Une approximation quadratique (en rouge) de la fonction sinus (en gris) suffit amplement à la représentation d'un oscillateur « vibrato ».



## LISTE DES COMPOSANTS

### Résistances (0,25 W)

R1=1 k $\Omega$   
R2=220  $\Omega$   
R3=100  $\Omega$  (voir texte)

### Condensateurs

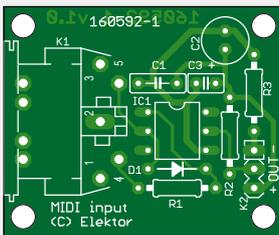
C1 = 100 nF, pas de 5 mm (voir texte)  
C2 = 100  $\mu$ F 6,3 V, 2,5 mm (voir texte)  
C3 = 1 nF, pas de 2,5 mm (voir texte)

### Semi-conducteurs

D1 = 1N4148  
IC1 = CNY17-3

### Divers

K1 = prise DIN à 5 contacts, 180°  
K2 = barrette de picots à 3 contacts, pas de 2,54 mm  
Circuit imprimé, réf. 160592-1



de maintien (*keep-alive*) sont transmis périodiquement – est ignoré. Puisque le synthé est monophonique, une seule note peut être jouée à la fois. Une touche pressée pendant la lecture d'une note joue la nouvelle note ; relâcher la nouvelle touche lorsque la précédente est encore pressée ne relance pas la note précédente. J'ai implanté des octets *Control Codes* (CC) standard chaque fois que possible (**tableau 1**). Le code CC 1 p. ex. correspond d'ordinaire à la molette de modulation d'un clavier. Le modulateur de hauteur d'un clavier MIDI n'utilise pas de code CC, mais sa propre commande MIDI, à savoir l'octet d'état 0x0b. Le temps de relâchement (*release time*) associé au code 72 définit le temps d'extinction d'une note après relâchement de la touche. Le synthé Tesla l'utilise comme « fondu de sortie » (*fade-out*) sur la largeur d'impulsion, mais le résultat est différent.

### Du portamento réservé au legato

La tâche *worker* lance l'oscillateur à basse fréquence (LFO) utilisé pour les effets de vibrato. Le signal LFO est donc échantillonné à 1 kHz, une valeur plus que suffisante pour la fréquence LFO maximale de 20 Hz.

Véritable abeille affairée, la tâche *worker* veille aussi à ce que la modulation de la hauteur soit correctement appliquée ( $\pm 2$  demi-tons). Elle surveille par ailleurs le portamento, implanté ici comme glissement linéaire d'une note A vers une note B, la durée du glissement étant indépendante de la distance entre A et B. Le portamento n'est en outre appliqué que si les notes sont jouées legato, c.-à-d. sans silence entre elles. Cela per-

met de mélanger des notes avec et sans portamento, sans utiliser de bouton. Le code CC 84 de commande du portamento ne fait rien (pour l'instant).

### Calculs simplifiés

Pour être euphonique, la modulation de la hauteur exige des calculs compliqués. Celui de la valeur de la modulation passe notamment par une élévation de 2 à la puissance 1/12, fonction qui n'existe pas en C. On peut heureusement écrire :

$$2^{\left(\frac{x}{12}\right)} = e^{\ln\left(2^{\left(\frac{x}{12}\right)}\right)}$$

puis simplifier le logarithme népérien :

$$\ln 2^{\left(\frac{x}{12}\right)} = \left(\frac{x}{12}\right) \cdot \ln(2) =$$

$$x \cdot \frac{\ln(2)}{12} = x \cdot 0,057762265$$

L'élévation à la puissance 1/12 se réduit ainsi à un calcul d'exponentielle :

$$e^{(x \cdot 0,057762265)}$$

Le langage C a bien une fonction `exp()`, mais sa précision ralentit le calcul. Pour nos besoins, on peut la remplacer par un développement limité d'ordre 2 (**fig. 2**) :

$$y = x \cdot 0,057762265$$

$$e^y \approx 1 + y + 0,5 \cdot y^2$$

Le calcul du sinus d'un angle de phase pour le LFO se heurte à un problème similaire, puisque là encore la fonction `sin()` s'avère bien trop lente. Un tableau de valeurs aurait fait l'affaire, mais la mémoire du PIC est déjà encombrée par le code du Smalltask. J'ai trouvé sur l'internet une solution futée qui utilise l'équation d'une parabole pour approcher les valeurs de la fonction sinus (**fig. 3**).

### Tableau 1. Correspondance entre les messages MIDI et les actions de la micro-bobine Tesla.

code CC	action
1 (0x01)	profondeur du vibrato
5 (0x05)	vitesse du portamento
12 (0x0c)	vitesse du vibrato
65 (0x41)	portamento on/off
72 (0x48)	temps de relâchement
84 (0x54)	commande de portamento



# EPS - Easy Parking System

## ou comment éviter de rayer sa voiture



**Olivier Croiset** (Reyrieux, France)

L'an dernier, mon fils me rejoint en voiture, les portières éraflées. Je sais qu'il n'est pas très habile pour manœuvrer (il a heureusement bien d'autres qualités !). Il m'explique que le parking de son immeuble est étroit et que le couloir en sous-sol qui mène à son parking est exigü. Garer sa voiture dans ces conditions n'est pas aisé. J'ai rapidement eu une idée pour pallier cette situation : estimer la distance de chaque côté de la voiture lorsqu'elle avance dans la place du garage. Un boîtier central signale de façon simple (voyants lumineux) si les distances limites sont atteintes, voire dépassées.

Le garage n'est pas alimenté en électricité, le couloir peu éclairé. L'EPS (*Easy Parking System*) doit donc être autonome et consommer très peu, mais apporter une aide fiable à la difficulté de se garer ; il sera donc alimenté par des cellules AA de 1400 mAh. Cette batterie sera à recharger régulièrement, une fois tous les deux mois environ. La consommation en veille profonde (*power down*) est de 400  $\mu$ A.

Dans le cas où le l'EPS est installé dans un garage alimenté par le secteur, il est bien sûr possible de se passer de la batterie en l'alimentant avec un bloc sec-

teur ; la fonction de veille sera tout de même active.

Je pensais pouvoir alimenter l'EPS au moyen d'une batterie portable USB. Malheureusement il semble que ces batteries (*power banks*) doivent débiter un courant d'une certaine intensité pour délivrer leur charge. Ce courant minimal ainsi que la durée pendant laquelle ce courant est disponible varient d'un fabricant à l'autre. En outre ce genre de dispositif se met en veille quand le courant est trop/très faible. Or mon montage consomme le moins possible : moins de 500  $\mu$ A en veille, quelques dizaines de mA pendant 10 min par jour (le temps de se

garer). J'ai donc abandonné cette source d'énergie pour une batterie constituée de quatre cellules de 1,4 V, car il est difficile de travailler avec une source dont on ne connaît pas les caractéristiques.

### Schéma

Le microcontrôleur ATmega328P est au centre de cette réalisation (**fig. 1**). Les capteurs à ultrasons sont placés de chaque côté de l'entrée du garage. Ils sont alimentés par le boîtier central qui envoie les commandes et réceptionne les distances mesurées.

Afin d'économiser la batterie d'alimentation, le boîtier central passe en veille au

bout de 10 min de fonctionnement. Pour le réveiller, il suffit de faire un appel de phares lorsque l'on arrive devant la place de parking ! Pas besoin de descendre de voiture pour mettre l'EPS en marche ! Notez que je n'ai pas testé le montage dans un environnement plus lumineux qu'un sous-sol dans la semi-obscurité. En plein jour, l'allumage de phares ne permettra sans doute pas de réactiver le circuit.

Il doit être possible d'utiliser un capteur infrarouge (comme le LHI 878 trouvé dans un détecteur de présence). Malheureusement, je ne suis pas parvenu à en obtenir quelque réaction (il était peut-être grillé). Il me semble que cela serait une solution à étudier ; de plus, il n'y aurait même plus à donner un coup d'appel de phares, la chaleur du moteur – à l'avant – suffirait pour stimuler le circuit. L'inconvénient de ce capteur est que lorsqu'on quitte la place de parking – moteur froid, l'EPS reste assoupi. Il faut bien que les Porsche 911 (moteur à l'arrière) aient un inconvénient !

J'ai étudié mon montage avec une carte Arduino, idéale pour ce type de projet ; or, le quartz et les deux condensateurs ne sont pas nécessaires, car la pré-

sion au millimètre près est inutile. En effet, l'important n'est pas de connaître les distances par rapport aux murs, mais de savoir si la voiture ne s'approche pas dangereusement des murs à gauche et à droite. Si on utilise l'oscillateur interne, cela libère les ports du quartz, et permet de mettre un troisième couple « capteur + LED » à l'avant par exemple.

L'utilisation des ports du  $\mu\text{C}$  est définie dans le programme par les directives `#define`. Ceci permet de modifier aisément l'affectation des ports et ainsi de faciliter le travail du projeteur CAO qui réalise le routage du PCB. Le dessin des pistes en sera plus simple.

Dans la mesure du possible, je laisse libres les ports RX et TX pour pouvoir diagnostiquer le fonctionnement des montages, en cas de difficulté. Quelques messages sur ce port série, un convertisseur série/USB et HTerm permettent de savoir ce qui se passe au cœur de l'Atmega328 et du programme.

Les trois touches pourraient être câblées en parallèle sur le port ISP ; cela impose que l'on n'appuie pas sur les touches pendant la programmation, mais tout bon électronicien le comprendra. Ceci permet de libérer des ports.

### Voyant de batterie

La LED rouge/verte indique l'état de l'EPS (marche/arrêt) ainsi que l'état de charge de la batterie. Le pont diviseur ( $R6/R7$ ) allié à un des ports analogiques (port PC4 – ADC4) mesure la tension de sortie du régulateur. Cette valeur est alors comparée à la référence interne de 1,1 V du  $\mu\text{C}$ . Si la tension de la batterie est suffisante ( $> \text{env. } 4 \text{ V}$ ), c'est la LED verte qui s'éclaire ; dans le cas contraire, la LED rouge s'allume. La tension de 4,5 V est la tension minimale d'alimentation des capteurs à ultrasons selon leur feuille de caractéristiques, mais ils fonctionnent correctement jusqu'à 4,0 V.

Ce pont doit être adapté à la batterie d'alimentation, la valeur limite est déterminée par le programme, mais aussi par les deux capteurs à ultrasons HC-SR04 qui ont besoin de 4 V !

### Capteurs à ultrasons

Lorsque l'EPS est sorti de l'état de veille, la voiture peut avancer dans le garage ; la distance entre chaque capteur et la carrosserie est alors mesurée en permanence. Le boîtier central alimente successivement le capteur gauche puis le droit, et leur envoi une impulsion de déclen-

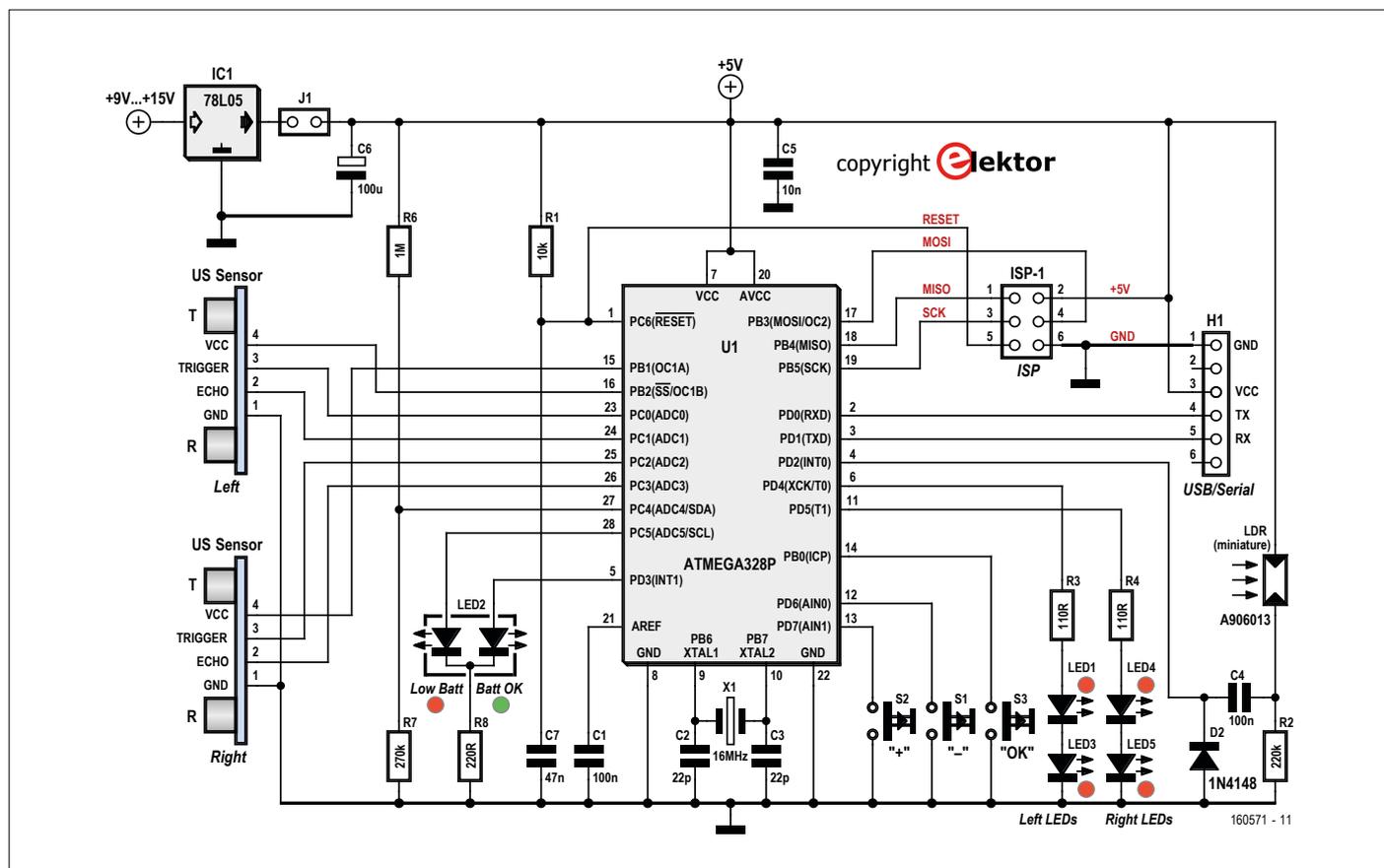


Figure 1. Schéma du montage animé par un  $\mu\text{C}$  ATmega328P.

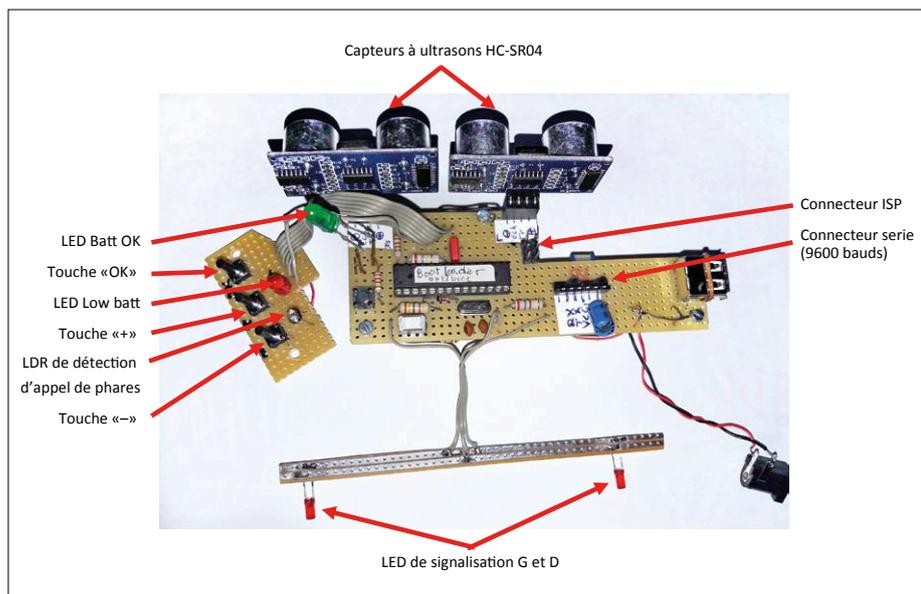


Figure 2. Prototype de l'EPS.

chement. Selon la distance mesurée par les capteurs à ultrasons gauche/droit (et le paramétrage que l'on verra plus loin), le voyant gauche/droit (chacun constitué d'une paire de LED : LED1-LED3 et LED4-LED5) s'éclairera, ou pas.

Pendant cette période active, le système consomme environ 30 mA, selon l'allumage des LED. Les plus habiles feront des économies !

L'EPS repasse en veille profonde (mode *power down*) au bout de 10 min de fonctionnement environ. Il est bien sûr possible de le réactiver pour une nouvelle période de 10 min en donnant un nouvel appel de phares. Mais je suis sûr que même les moins adroits se seront garés dès la première fois !

### Mise en marche de l'EPS

L'EPS est mis en fonction par un appel de phares. La LDR (réf. A 9060-13) et les composants associés transforment cette variation d'intensité lumineuse en une impulsion transmise à la broche INT0. Cette impulsion déclenche une interruption qui sort le µC de la veille profonde. La diode D2 est une diode de roue libre, elle absorbe la tension négative lors de cette impulsion. Les capteurs à ultrasons sont alors activés, comme indiqué précédemment.

### Mise en service : première mise sous tension et paramétrage

Lors de la première mise sous tension (et donc lors de chaque remplacement de la batterie d'alimentation), il est pos-

sible – mais pas obligatoire – de régler la distance pour laquelle les LED gauches et droites vont s'allumer pour alerter le conducteur. Voici la procédure, très simple :

- Placez un obstacle en face du capteur à ultrasons gauche, à la distance voulue pour qu'aucune partie de la voiture ne puisse toucher les montants du garage. Tenez compte des rétroviseurs !
- Mettez en marche l'EPS (si la batterie est correctement chargée, la LED verte s'allume), le voyant gauche va clignoter deux fois lentement.
- Appuyez sur les boutons « + » et « - » pour que les LED de gauche s'allument, puis appuyez sur « OK ». Les deux LED gauches ainsi que les deux droites clignotent deux fois pour indiquer que le réglage de la voie gauche est terminé. Nota : la touche « - » réduit la distance de détection jusqu'à 0, la touche « + » l'augmente jusqu'à un mètre environ.
- Comme pour la voie gauche, placez un obstacle en face du capteur à ultrasons droit.

- Appuyez sur les boutons « + » et « - » pour que les LED de droite s'allument, puis appuyez sur « OK ».
- Le voyant de gauche LED1-LED3 et celui de droite LED4-LED5 vont clignoter deux fois lentement, ce qui indique que la procédure d'adaptation à vos besoins est terminée.

Ce réglage est sauvegardé dans l'EEPROM de l'ATMega, il n'est donc pas nécessaire de refaire ce réglage à chaque mise en service.

Lorsque vous remplacez la batterie, l'EPS passe par la procédure de réglage ; si le réglage convient, il suffit d'appuyer sur le bouton « OK » pendant que les voyants gauche et droit s'éteignent (la fonction de surveillance se poursuit pendant 10 min). J'ai installé les modules à ultrasons dans des coffrets KS28 G026 de KEMO-Electronic. Le plus gênant, ce sont les deux câbles qui relient le boîtier central aux deux modules. On ne peut pas avoir que des avantages !

Voici un petit montage qu'apprécieront les moins adroits au volant ainsi que les plus anciens qui ont parfois des difficultés à se contorsionner pour apprécier toutes les distances, dans toutes les directions. Une amélioration possible du montage serait de compléter les avertissements lumineux avec un signal sonore, bien sûr assez fort pour être audible lorsque le moteur de la voiture tourne. ◀

(160571)

### Liens

- [1] Page de l'article : [www.elektormagazine.fr/160571](http://www.elektormagazine.fr/160571)
- [2] Page du projet dans le labo d'Elektor : [www.elektormagazine.fr/labs/easy-parking-system-1](http://www.elektormagazine.fr/labs/easy-parking-system-1)
- [3] Vidéo du prototype en fonctionnement : [www.youtube.com/watch?v=79tF6HEACns&feature=youtu.be](http://www.youtube.com/watch?v=79tF6HEACns&feature=youtu.be)

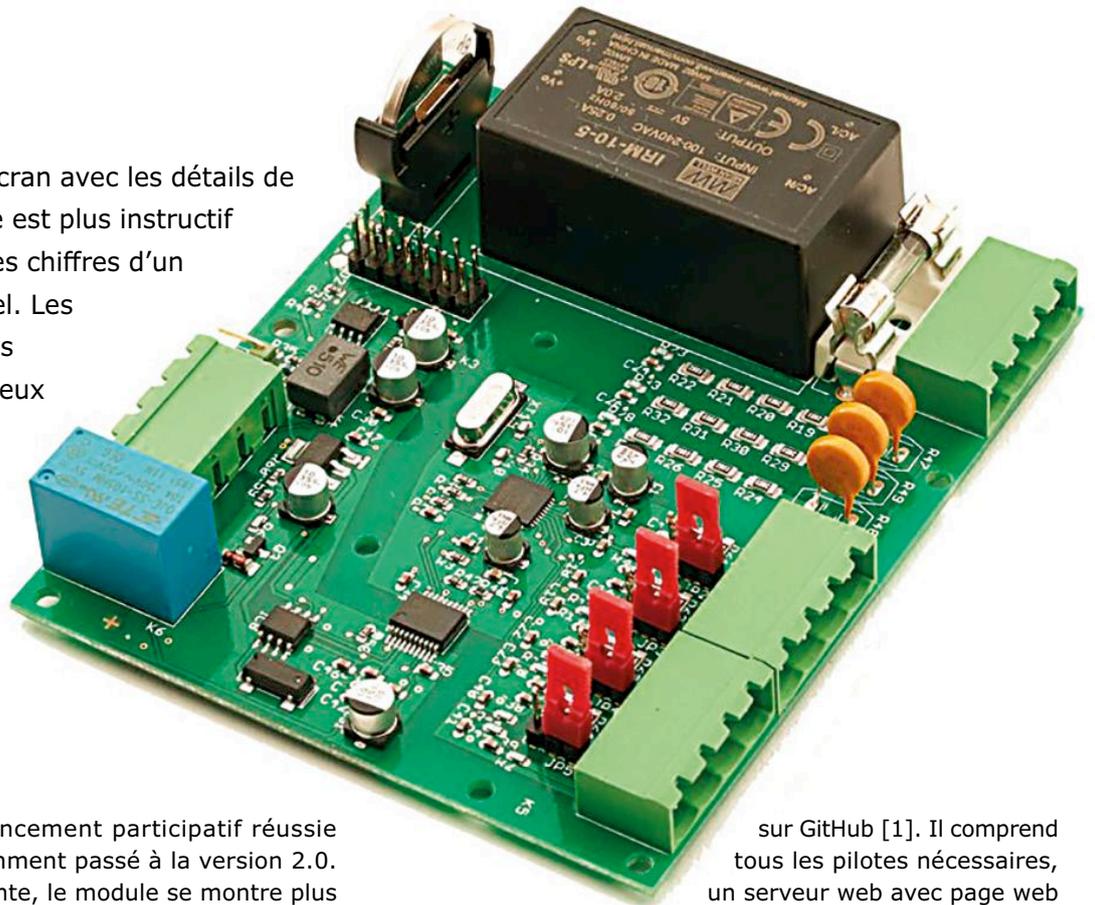
banc d'essai

# on compte sur SmartPi

## le compteur d'énergie intelligent pour Raspberry Pi

Clemens Valens (labo d'Elektor)

Pour l'utilisateur lambda, un écran avec les détails de sa consommation d'énergie est plus instructif que le morne défilement des chiffres d'un wattheuremètre traditionnel. Les fournisseurs d'énergie et les opérateurs de réseaux ont eux aussi besoin de connaître en détail nos « habitudes de consommation ». Le compteur électrique « bidirectionnel » SmartPi se révèle à cet égard intéressant et bon marché.



Après une campagne de financement participatif réussie en 2016, le SmartPi est récemment passé à la version 2.0. Comparé à la version précédente, le module se montre plus pratique grâce à une alimentation intégrée et au placement sur un même côté des connecteurs les plus utiles. La carte gagne également en flexibilité puisque tous les connecteurs disposent maintenant de borniers à vis et qu'une interface RS-485 permet la connexion de dispositifs Modbus. Un port pour relais a été ajouté, et l'horloge en temps réel (RTC) a une pile de secours. Enfin, le nouveau boîtier possède un clip compatible rail DIN.

### Spécifications

Le SmartPi mesure des intensités jusqu'à 100 A sur quatre canaux (trois phases + neutre), des tensions (jusqu'à 390 V<sub>eff</sub>) sur trois canaux ainsi que des fréquences. Il calcule différentes puissances (active, réactive et apparente) et la consommation. Et puisqu'il mesure aussi la production d'énergie, le fabricant l'a à juste titre nommé « compteur intelligent bidirectionnel ». Le SmartPi peut être utilisé avec des capteurs de courant inductifs à sortie secondaire de 50 mA ou 1 A, ce dernier type de sonde permettant, après modification d'un cavalier, la mesure d'intensités jusqu'à 700 A.

La carte d'extension SmartPi peut être achetée séparément, sous forme de kit configurable, ou entièrement assemblée. La version assemblée (celle de ce banc d'essai) est construite autour d'un Raspberry Pi 3 modèle B.

Le logiciel est écrit en Go, son code source ouvert est disponible

sur GitHub [1]. Il comprend tous les pilotes nécessaires, un serveur web avec page web ainsi qu'une interface REST permettant l'intégration du SmartPi dans des applications maison.

### Plus succinct, c'est un télégramme

Comme trop souvent de nos jours, le manuel livré avec le multimètre SmartPi est pour le moins succinct. Une visite pleine d'espoir sur le site du fabricant s'avère tout aussi décevante (à propos de ce site, notez que s'il est en allemand les pages consacrées au SmartPi sont disponibles en anglais). Entrez l'URL [2] pour vous y rendre, ou cliquez sur le lien vidéo (!) de l'e-choppe. La page Kickstarter [3] est sans doute la meilleure des sources d'information, sans oublier le forum [4].

La page *Installation* du site n'explique pas l'installation matérielle du module, mais celle du logiciel sur un Raspberry Pi. Quant au raccordement du module au réseau électrique d'une maison, il a carrément été laissé à titre de travaux pratiques. La page *Use* montre bien un schéma de connexion, mais ici non plus rien n'est dit sur le câblage du module au secteur. Il est même suggéré de confier l'installation à un électricien professionnel (fig. 1). C'est évidemment de bon conseil, mais nous aurions aimé davantage de détails ici.

Autre frustration, la page n'a pas été mise à jour et se réfère toujours au SmartPi 1.0 et à son alimentation externe.



Figure 1. L'installation du SmartPi par un technicien qualifié (je le suis. Vraiment).



Figure 2. Le brochage des connecteurs de tension et de courant.

### Connexion du SmartPi à un réseau de données

Le SmartPi arbore tous les connecteurs du RPi, dont le connecteur Ethernet. Une fois le SmartPi relié à mon routeur, il est apparu dans la liste des clients DHCP. Faites suivre l'adresse IP indiquée par votre routeur du numéro de port 1080, et l'interface utilisateur du SmartPi s'affichera aussitôt.

Une liaison Wi-Fi m'aurait été bien utile, car mon routeur est loin de mon tableau électrique, mais nulle part il n'est fait mention de cette possibilité. Comme le RPi 3 embarque un module Wi-Fi, j'ai tenté sa configuration via SSH. Le nom d'utilisateur par défaut est *pi*, avec comme mot de passe *smart4pi* ou *raspberry*.

### SmartPi et Wi-Fi

Connectez le SmartPi via SSH avec PuTTY ou tout autre client SSH, et ouvrez le fichier `wpa_supplicant.conf` avec la commande :

```
sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
```

Ajoutez les lignes suivantes à la fin du fichier en remplaçant `mySSID` et `myPassPhrase` par votre SSID et votre mot de passe :

```
network={
    ssid="mySSID"
    psk="myPassPhrase"
    key_mgmt=WPA-PSK
}
```

Enregistrez le fichier avec *Ctrl-O* suivi de *Entrée*, et quittez nano avec *Ctrl-X*. Débranchez le câble Ethernet, redémarrez le SmartPi, et voilà, la connexion Wi-Fi fonctionne. Pourquoi diable n'ont-ils pas explicité tout ça ?

### Modification du mot de passe du SmartPi

Profitez-en pour changer le mot de passe, une opération recommandée par la documentation officielle même si, là encore, rien n'est expliqué sur le comment du truc. Le truc, le voici. (Re)lancez le client SSH, connectez le SmartPi, puis lancez l'outil de configuration du Raspberry Pi avec :

```
sudo raspi-config
```

Sélectionnez **Change User Password** et entrez un nouveau mot de passe. Voilà, c'est fait.

### Raccordement du secteur au SmartPi

Là non plus la documentation n'aide pas vraiment. Par chance les entrées du SmartPi correspondent aux indications phases (L) et neutre (N) étiquetées sur la face arrière du module (**fig. 2** ; cela dit il n'y a rien sur le mien).

Le SmartPi d'évaluation m'a été livré avec trois sondes de courant précâblées. J'habite en France, et le réseau monophasé de ma maison est séparé en deux groupes. J'ai placé dans chacun une sonde et fixé la troisième sur le fil allant au chauffe-eau (second groupe). La documentation précise que les sondes ont une flèche qui doit pointer vers le consommateur d'énergie (moteur, bâtiment, etc.), mais il m'a fallu inverser ces flèches pour lire des valeurs positives.

J'ai relié les trois entrées de tension et les ai connectées à un fusible indépendant.

Lorsque vous avez réussi à obtenir des valeurs positives de tension et de puissance sur les (trois) canaux, ne reste plus qu'à interpréter les données.

### Tableau de bord

Ce qui suit s'applique à la version 0.6.4 du logiciel.

L'interface web (par défaut sur le port 1080, mais vous pouvez en choisir un autre) affiche trois intensités, trois tensions et trois fréquences (**fig. 3**).

L'onglet *Measurement* de la section *Settings* permet d'inverser le sens du courant, de sélectionner un type de sonde ainsi que

**DANS L'E-CHOPPE**

- 18165 : carte d'extension SmartPi 2.0 avec 3 sondes de courant 100 A
- 18166 : boîtier pour SmartPi 2.0
- 18167 : sonde de courant, 100 A

les canaux de tension à surveiller. On peut aussi paramétrer la tension et la fréquence du secteur. J'ai dû redémarrer le SmartPi à chaque changement de paramètre. Pour une raison que j'ignore, la désactivation du canal du neutre est restée sans effet.

Le SmartPi a pour ainsi dire les fils qui se touchent lorsque les intensités sont faibles : les mesures deviennent incohérentes et les valeurs de la fréquence et du cos phi font du trampoline (de 0 à plus de 50 GHz, **fig. 4**). Il ne devrait pas être difficile de corriger ça dans le code.

Attendez quelques jours de collecte des données pour mieux comprendre les courbes affichées. Les données peuvent être exportées dans un fichier CSV et exploitées dans un tableur (on peut définir le caractère servant de virgule décimale depuis l'onglet *Expert* de la section *Settings*, mais pas le séparateur de champs...)

Les courbes sont agréables à l'œil et automatiquement mises à l'échelle ; parfois c'est bien, parfois ça l'est moins, en particulier lorsqu'il y a eu un large pic dans un passé récent. Il suffit de cliquer sur le bouton correspondant pour voir ou masquer une courbe d'intensité, tension, puissance, consommation ou production d'énergie. Passez la souris sur une courbe pour visualiser les valeurs et l'horodatage à la position du curseur. La production et la consommation d'énergie de la semaine passée sont indiquées au moyen d'histogrammes.

### Options et interfaces

L'API REST mentionnée plus haut permet de faire communiquer le SmartPi avec d'autres applications. La section *Settings* du tableau de bord dévoile à cet égard les intentions des développeurs puisqu'on y trouve différentes options de communication et d'interfaçage. On peut ainsi configurer une connexion UMTS, un serveur FTP et un client MQTT. Une recherche dans le dépôt GitHub nous apprend que MQTT est supposé fonctionner, mais rien n'est dit sur FTP. On devine juste que le protocole a été ajouté pour charger automatiquement des fichiers CSV via un serveur FTP.

La prochaine version devrait inclure le port RS-485 (Modbus) et le contact commandé par relais.

### Une plateforme pour amateurs avertis

Les heures que j'ai passées à bidouiller le matériel et le logiciel du SmartPi me laissent une impression mitigée. D'un côté c'est un produit au potentiel indéniable, d'un autre côté l'immaturation du code et la pauvreté de la documentation (du moins à ce jour) m'ont vraiment gêné. Autrement dit il est à réserver à des amateurs avertis.

Le logiciel est à l'évidence plein de bonnes intentions, mais il manque de précision et de finesse. Cela dit il est régulièrement mis à jour – vérifiez la version en cours sur le forum [4]. Les utilisateurs sans expérience de Linux ou du RPi seront vite dépassés. C'est dommage, car il ne devrait pas être difficile de compléter le menu de configuration (qui aurait alors une gestion du mot de passe, de la configuration Wi-Fi, etc.), d'ajouter des pages d'aide et d'enrichir le manuel de quelques paragraphes bien illustrés. Le SmartPi pourrait alors intéresser un public bien plus vaste. Ceci dit, si vous recherchez une plateforme sur laquelle construire un système de mesure intelligent de la consommation et de la production d'énergie, le SmartPi est assurément un bon point de départ [5].

(160485 – version française : Hervé Moreau)

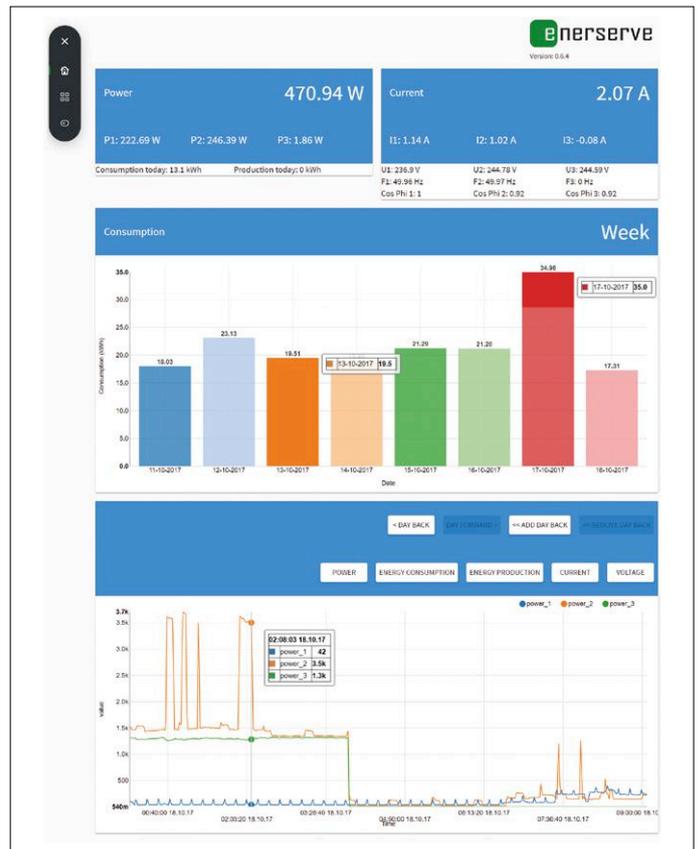


Figure 3. Le tableau de bord affiche les consommations actuelles et passées.

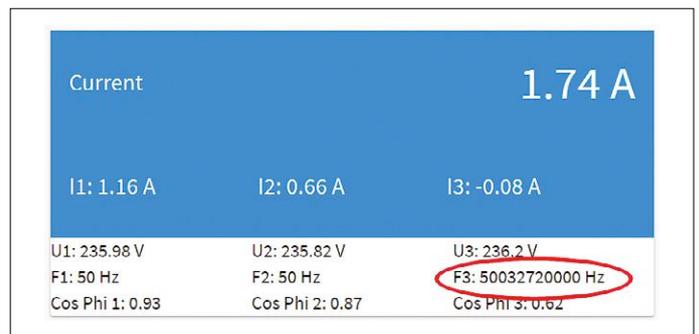


Figure 4. Une fréquence secteur de 50 GHz ? Qui peut y croire ?

### Liens

- [1] SmartPi sur GitHub : <https://github.com/nDenerserve/SmartPi>
- [2] Page d'accueil : <http://www.emanager.eu/en/products/smartpi>
- [3] Kickstarter : <https://www.kickstarter.com/projects/1240982104/smartpi-turn-your-raspberry-pi-into-a-smartmeter>
- [4] Forum : <https://forum.enerserve.eu/>
- [5] [www.elektormagazine.fr/160485](http://www.elektormagazine.fr/160485)

# ICM7216 d'Intersil

## drôle de composant n°34

David Ashton (Australie)

Dans la rubrique Rétronique de novembre 2016 d'Elektor, Jan Buiting évoquait avec nostalgie la vénérable puce d'horloge MM5314 [1]. Cela m'a rappelé d'autres puces tout aussi rétro, mais plus originales : la série unique de compteurs-fréquence-mètres intégrés ICM7216.



Figure 1. Un ICM7216. Source : Eric de Pauw / CircuitsOnline.net.

Ces puces sont parmi les premières à mériter le qualificatif de circuit à haute densité (LSI ou *Large Scale Integration*) apparu au milieu des années 70. Vous trouverez des feuilles de caractéristiques datant de 1976 à 2004, chez Intersil et Harris (acheté par Intersil en 1999). Une feuille de caractéristiques couvrant toutes les versions décrites ici peut être trouvée sur [2].

Quiconque a conçu un compteur-fréquence-mètre avec des puces TTL sait que la tâche n'est pas simple. Il faut une base de temps précise et stable et un générateur pour les signaux de déclenchement du comptage, verrouillage et remise à zéro qui permettent au compteur de compter les impulsions en entrée sur une période précise, d'arrêter le comptage puis de verrouiller et décoder les signaux pour un afficheur. Et cela juste pour un simple compteur. Les puces de la série ICM7216 font cela et plus en un seul circuit intégré à 28 broches (fig. 1).

### L'ICM7216

Il existe quatre versions de l'ICM7216 : A, B, C et D. Les versions A et B sont des compteurs-fréquence-mètres complets à huit digits capables également de mesurer période, rapport de fréquence, durée ou de compter des événements. Les versions C et D sont des compteurs-fréquence-mètres de base. Les versions A et C sont prévues pour les afficheurs à 7 seg-

ments à anode commune, les versions B et D pour ceux à cathode commune. Leur gamme de fréquences s'étend du continu à 10 MHz. Le circuit fondamental d'un compteur à ICM7216B est visible en figure 2.

En plus du circuit intégré, il faut huit afficheurs, un résonateur à quartz et divers autres petits composants. Vous avez le choix entre un résonateur à 1 MHz ou 10 MHz pour la base de temps. La résolution du compteur est de 1 pour 10<sup>8</sup>,

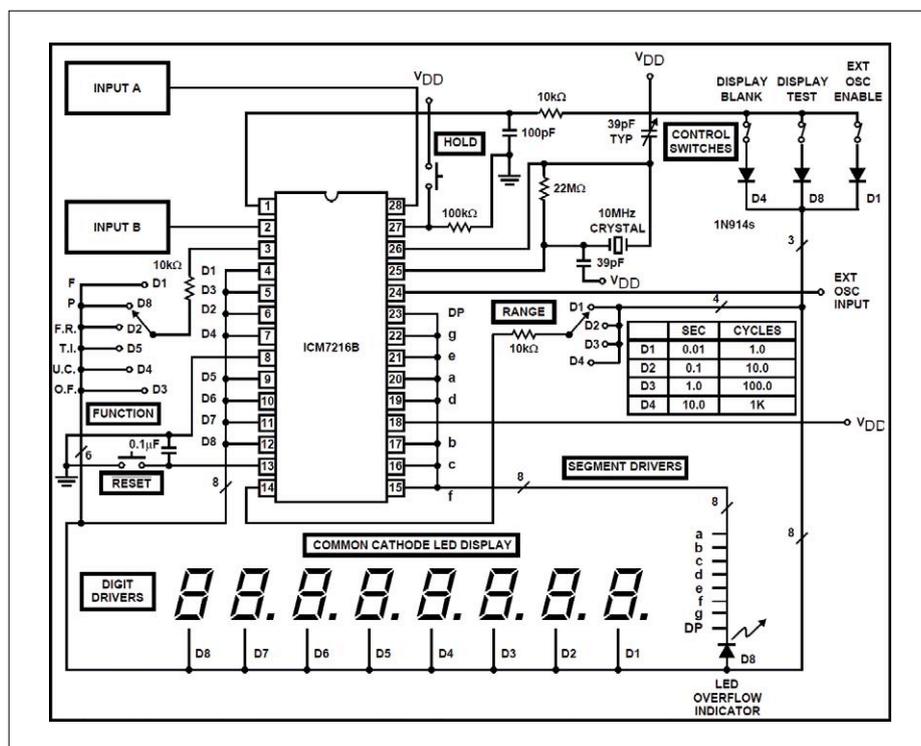


Figure 2. Schéma d'un compteur/chronomètre 10 MHz à ICM7216. Source : feuille de caractéristiques d'Intersil.

mais un quartz de base ne permettra pas cette précision : il dérivera avec le temps et la température ambiante. Vous pouvez aussi utiliser une horloge externe ; un TCXO (*Temperature Compensated Crystal Oscillator*) sera meilleur et un OCXO (*Oven Controlled Crystal Oscillator*) aura la stabilité requise pour cette résolution. La stabilité d'une horloge au rubidium ou césium (n'ayez crainte, une simple horloge GPS possède cette précision) dépasserait la résolution. La puce pourra commander directement la plupart des petits afficheurs à 7 segments et placer automatiquement le point décimal. Les modes et réglages sont sélectionnés à l'aide d'interrupteurs et cavaliers astucieusement multiplexés. Vous pourrez mettre en place un interrupteur pour tester les afficheurs (tous les segments illuminés) ou les mettre hors circuit (commande des afficheurs désactivée afin de pouvoir les utiliser pour autre chose). Vous pourrez aussi ajouter des boutons de maintien ou remise à zéro de la mesure. Tout cela dans une petite puce polyvalente.

#### Au-dessus de 10 MHz

Les ICM7216C/D sont parfaits pour les fréquencemètres intégrés. Les feuilles de caractéristiques décrivent une astuce utile : si vous utilisez un quartz à 2,5 MHz, au lieu de 10 MHz, le compteur sera multiplié par un facteur 4. Utilisez un 74S112 pour diviser la fréquence d'entrée par 4, et ces deux changements insignifiants auront étendu la gamme jusqu'à 40 MHz. Très pratique pour tester quartz et oscillateurs : beaucoup de modèles courants dépassent les 10 MHz, mais la plupart restent en dessous de 40 MHz. Avec les ICM7216A/B, mieux vaut ajouter un prédiviseur pour atteindre 100 MHz ou même 1 GHz. À l'époque de ces puces, le 11C90 ECL (*Emitter Coupled Logic*) était un bon prédiviseur 100 MHz ; de nos jours vous aurez peut-être la chance de trouver un 74S196 ou un circuit diviseur par dix similaire. Au-delà, des diviseurs par dix qui vont jusqu'à 600 MHz ou même 1 GHz se trouvent sans trop de peine. Suivant vos besoins, il vous faudra probablement aussi amplifier le signal.



Figure 3. Carte du fréquencemètre Sinclair PFM200. Source : Eric de Paauw / CircuitsOnline.net.

#### Et vous ?

L'ICM7216 a été utilisé dans des produits du commerce ; le fréquencemètre Sinclair (Royaume-Uni) PFM200 en est un exemple notable. Il s'agit d'un modèle miniature qui utilisait les mêmes afficheurs et boîtiers que les calculatrices de poche de la marque. La **figure 3** montre une carte à PFM200. La densité de composants élevée (pour l'époque) est typique de Sinclair. Le SP8660 à huit broches est le prédiviseur de choix pour la gamme 100 MHz. L'interrupteur à glissière du bas sélectionne la gamme. L'ICM7216, et sans doute aussi le 7226, étaient au centre de plusieurs projets Elektor des années 80 et début 90.

#### Alternatives d'hier et d'aujourd'hui

Aujourd'hui, vous utiliseriez sans doute un petit microcontrôleur ou configureriez un FPGA pour mesurer votre fréquence. Mais, pour créer rapidement et facilement un compteur-frequencemètre, ces circuits vieux de 40 ans sont toujours une bonne solution. Malheureusement ils ne sont pas donnés : attendez-vous à payer 25 \$ ou plus pour la version A ou quand même 10 \$ pour la version D sur

eBay. Mais pour une solution complète, ce n'est pas tant que ça, hein ?

Très peu d'autres puces font aussi bien. Intersil avait aussi à son catalogue une puce au fonctionnement presque identique, l'ICM7226, dans un boîtier à 40 broches, dotée de sorties BCD (parfait pour envoyer ou enregistrer les mesures). Sinon, il existe le 4553 à 3 digits ou la série 74C925 à 4 digits ; mais ce ne sont que de simples compteurs : vous devrez gérer le démarrage et l'arrêt de la mesure vous-même. De plus, leur technologie CMOS primaire n'est pas très rapide. Pour réaliser une fonction équivalente avec des TTL standard, il faudra en remplir une carte ! Au final, si vous souhaitez réaliser un compteur polyvalent, sans programmation ou conception matérielle, ces puces feront le boulot admirablement et vous ne ferez guère mieux que le circuit de référence de la feuille de caractéristiques. Je suis peut-être vieux jeu, mais j'avoue sans complexes mon faible pour ces circuits d'antan. ◀

(160398 – version française : Kévin Petit)

#### Liens

- [1] « MOS Clock 5314 (1974) », *Rétronique*, Elektor 11/2016 p. 74 : [www.elektormagazine.fr/160102](http://www.elektormagazine.fr/160102)
- [2] Feuille de caractéristiques des ICM7216A/B/C/D : [www.datasheetarchive.com/dl/255c6395adcb0ebce85a5719d13d509555f929/O/ICM7216](http://www.datasheetarchive.com/dl/255c6395adcb0ebce85a5719d13d509555f929/O/ICM7216)



Avec un ordiphone ou un lecteur MP3, vous pouvez écouter de la musique enregistrée en numérique, mais pas au volume voulu dans un salon, ou alors de piètre qualité. Avec le projet *Card Sound*, la puissance y est et la qualité aussi ! Ce lecteur est léger, petit, restitue l'enregistrement avec très peu de bruit de fond et dispose d'atouts intéressants comme des pilotes de bus I<sup>2</sup>C et RS485.

### Caractéristiques techniques

- Tensions d'alimentation analogique/numérique : 3 à 18 V<sub>cc</sub> / 4,5 à 12 V<sub>cc</sub>
- Consommations : cf. encadré Mesures
- Puissance de sortie sinus P<sub>sortie</sub> sur 4 et 8 Ω : 12,5 W et 6,2 W (mesuré : 10,0 V<sub>crête-crête</sub>)
- Sortie : en court-circuit max 2 A
- Distorsion : cf. encadré Mesures
- Bruit sans signal : env. 1,4 mW
- Formats de données : PCM 16, 24, 32 bits à 32, 44,1, 48, 96, 128 kHz stéréo
- Dimensions (L x l x h) : 89 × 33 × 30 mm
- Poids sans carte SD : 26 g

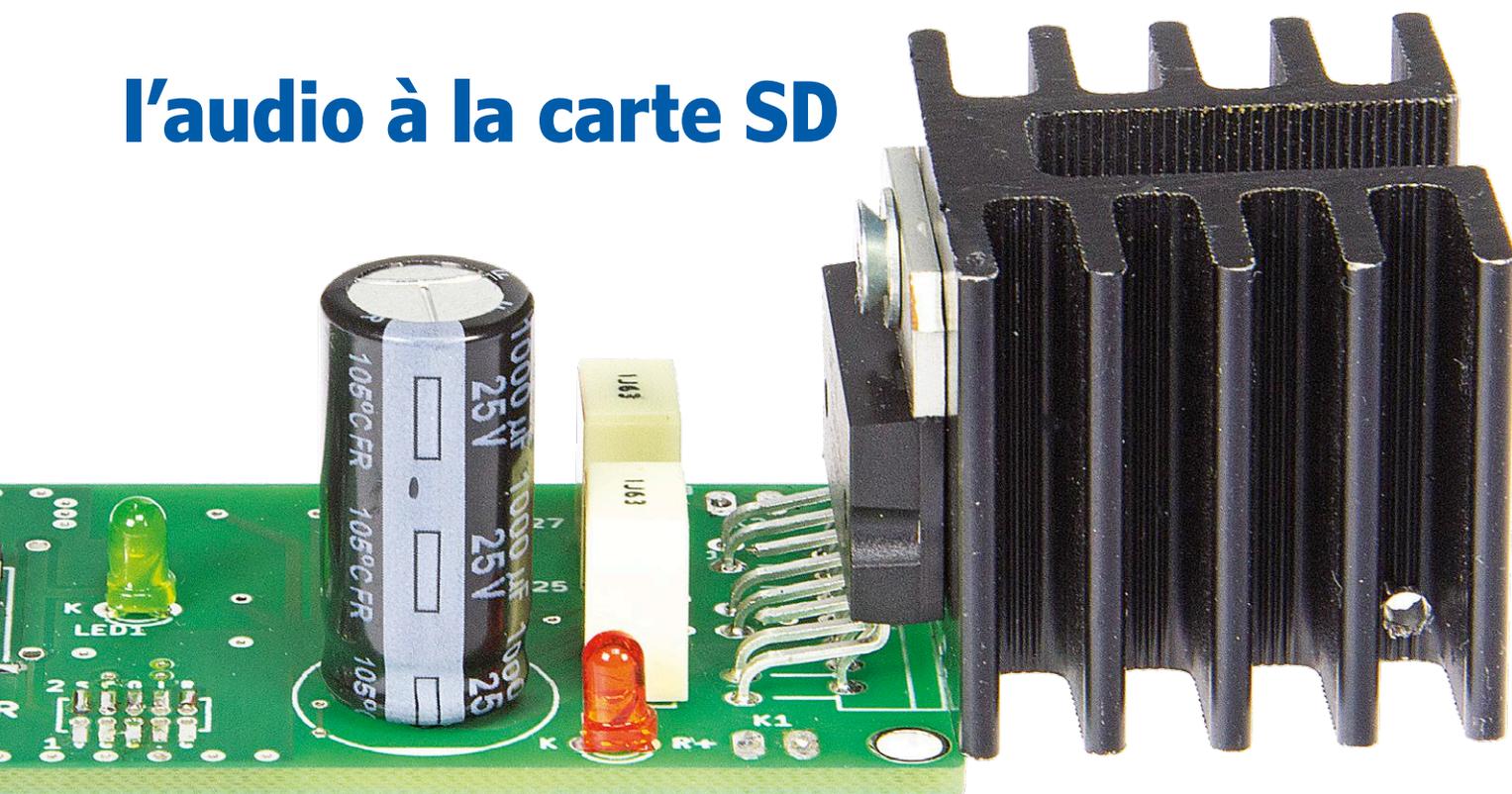
Valeurs valables sous alimentation de 5 V<sub>cc</sub> (numérique) et 12 V<sub>cc</sub> (analogique) avec signal sinus 1 kHz pour 16 bits et taux d'échantillonnage de 44,1 kHz.

**Falko Bilz** (cardsound@gmx.net)

*Card Sound* est un lecteur audio petit, mais de qualité avec de multiples capacités. Comme support pour les données audio, il utilise une carte micro-SD qui offre une grande capacité à faible prix. N'importe quel PC ou portable doté du matériel audio actuel permet de remplir une carte de ce type. Le projet ne se limite pas à jouer de la musique, il peut

# lecteur *Card Sound*

## l'audio à la carte SD



vous servir de sonnette de porte haut de gamme, de réveil matin mélodique, de module son à deux canaux (!) en modélisme pour des haut-parleurs indépendants ou d'auxiliaire sonore dans une installation domotique, par exemple, et encore dans bien d'autres applications. Il n'y a pratiquement aucune limite au nombre de fichiers son à lire. Ce projet constitue aussi un bon exemple de la manière de se servir d'une carte SD et d'un convertisseur N/A par bus I<sup>2</sup>S (*Inter-IC Sound*). Survol des caractéristiques principales :

- son de qualité avec étage de sortie en classe AB
- 2x7 W sinus, en stéréo ou double ampli grave/aigu séparés
- plusieurs interfaces disponibles : I<sup>2</sup>C, UART, RS485, GPIO
- matériel et logiciel 100 % ouverts

- une seule tension d'alimentation, séparée pour l'audio en option
- le nombre de fichiers enregistrables et leur longueur ne dépendent que de la capacité de la carte SD
- format compact, poids plume, faible consommation.

### Le circuit

Dans la **figure 1**, suivons le trajet des données audio stockées dans la carte SD enfilée dans le socle K5, lui-même relié au microcontrôleur IC2 par bus SDIO. Ce bus accepte le transfert bidirectionnel des données sur quatre bits de largeur (D0 à D3), avec horloge (CK) et ligne de commande (CMD). Pour les détails, suivez le lien [1]. On combine chaque fois deux bits sur les quatre lignes de données pour faire un octet, ce qui accélère sensiblement la lecture par rapport au bus à un bit habituel. Dans le contrôleur,

### INFOS SUR LE PROJET



Sortie audio

CN/A

carte SD



débutant

→ **connaisseur**

expert



env. 4 h



poste à souder CMS et expérience, TINA-TI, compilateur OpenSTM32 et programmeur STVP STM32, PC



env. 50 €

les résistances de rappel haut sont commutées par logiciel si nécessaire. Les données audio de la carte SD, le contrôleur STM32F401 (avec cœur Cortex-M4 de ST Microelectronics) les convertit en signaux en PCM (modulation d'impulsion codée) selon les besoins de l'utilisateur pour les envoyer sur le bus I<sup>2</sup>S, une interface spéciale développée par Philips Semiconductors pour l'audio numérique sériel. Ce bus de données comporte donc une ligne

d'horloge de bit CK, une ligne de données SD et une de commutation entre canaux gauche et droit LRCK. Il y a en outre une ligne non spécifiée dans la norme pour l'horloge maître MCK, fixée par IC2 à 256 fois LRCK. Les signaux du bus I<sup>2</sup>S commandent le convertisseur N/A CS4344 (IC5) fabriqué par le spécialiste audio Cirrus Logic. En raison du rapport fixe entre MCK et LPCK, IC5 ne peut travailler qu'aux fréquences d'échantillonnage de 32 kHz,

44,1 kHz, 88,2 kHz, 96 kHz et 128 kHz, un compromis facilement acceptable puisque la célèbre fréquence du CD à 44,1 kHz et celle de qualité supérieure de 128 kHz y sont représentées (voir la formule).

Le paquet logiciel [2] contient le fichier [calculations.ods](#), dans lequel il y a la valeur appropriée du registre que l'on retrouvera dans le fichier Setup [bsp\\_card\\_sound.c](#). À remarquer : on ne peut

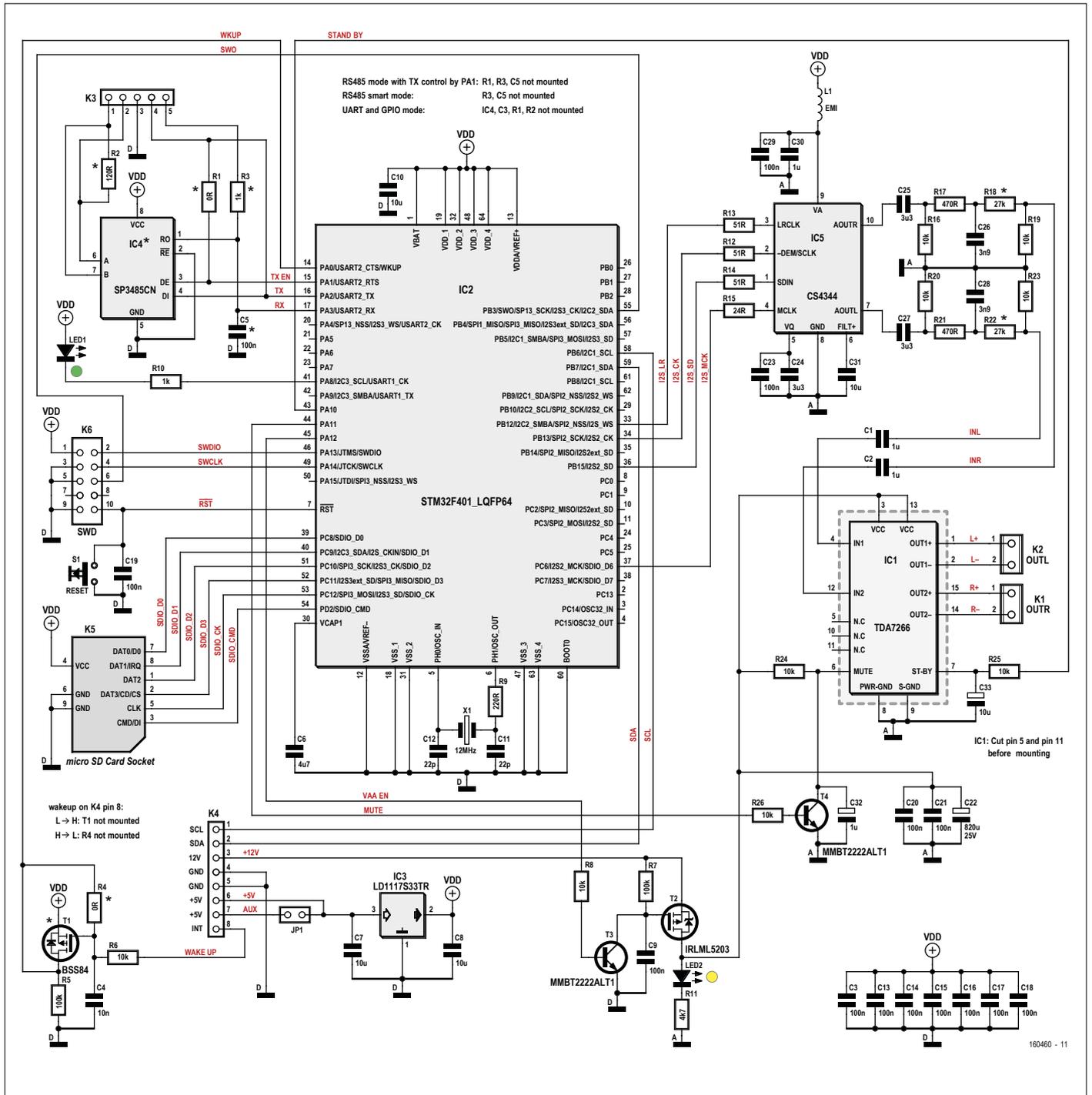


Figure 1. Schéma du lecteur Card Sound.

pas produire exactement ces fréquences avec le quartz de 12 MHz utilisé. Mais la déviation absolue maximale de fréquence reste inférieure à 0,06 %.

Les résistances R12 à R15 limitent les pointes de courant qui se produisent surtout lors des flancs des signaux numériques. Les résistances forment avec les capacités d'entrée de 8 pF de IC5 des filtres passe-bas. Le constructeur recommande [3] des valeurs de 51 Ω et 22 Ω. Il en va de même pour le condensateur tampon C31 de la tension de référence interne et pour C23 et C24 qui servent au lissage de la tension médiane.

Comme IC5, ainsi que d'autres parties du circuit, ne disposent pas de tensions d'alimentation symétriques, pour former une tension sinusoïdale d'amplitude maximale, il faut une tension neutre située à la moitié de la tension d'alimentation. Cette tension médiane est commodément produite en interne. Les signaux analogiques de sortie pour les deux canaux aux broches 7 et 10 présentent ainsi une tension continue de décalage d'environ 1,6 V.

Les parasites provoqués par l'effet de repliement au-dessus de la moitié de la fréquence d'échantillonnage nécessitent toujours un filtrage à la sortie d'un convertisseur N/A. Pour les fréquences PCM de 44,1 kHz et plus, ils se trouvent bien dans le domaine de l'in audible, mais peuvent entraîner des oscillations et de la distorsion dans les amplificateurs qui suivent. IC5 est doté d'un filtre de sortie à flancs raides qui, par exemple pour le 44,1 kHz, atténue les parasites supérieurs à 24 kHz d'au moins 50 dB [4]. Avec une puissance de sortie de 5 W, cela représente 50 mW.

Les condensateurs C25 et C27 découplent la partie en continu évoquée. Le filtre passe-bas du 1<sup>er</sup> ordre qui suit, composé de R17/C26 avec l'impédance d'entrée R16 d'un côté, de R21/C28 et R20 de l'autre, réduit les résidus de HF à partir de 100 kHz d'au moins 5 dB dans le signal PCM, comme le suggère la fiche technique du CN/A [4].

Enfin, les diviseurs de tension R18/R19 et R22/R23 atténuent le signal de sortie, parasites inclus, d'à peu près 15 dB. En tout, les fréquences supérieures à 1 MHz subissent ainsi une réduction d'au moins 70 dB. Les valeurs des résistances dépendent de la tension d'alimentation analogique, elles doivent veiller à ce que l'ampli analogique IC1 ne soit pas saturé, mais aussi à garder une plage de dyna-

$$LRCK = \frac{12MHz \cdot PLLI2SN}{Bits \cdot Channels \cdot PLLM \cdot PLLI2SR \cdot (2 \cdot I2SDIV + ODD) \cdot 4 \cdot (2 - CHLEN)}$$

Cette formule permet de calculer la fréquence d'échantillonnage résultante pour la valeur du registre IC2.

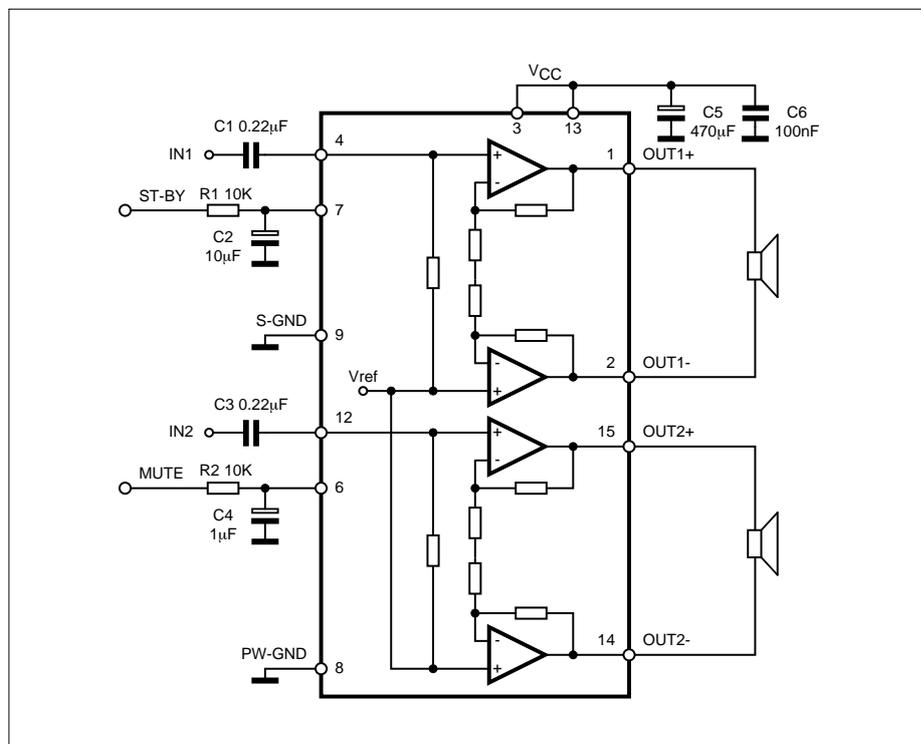


Figure 2. Circuit interne du TDA7266.

mique suffisante à IC5. Le **tableau 1** en indique les meilleures valeurs.

### L'étage de sortie

L'amplificateur de puissance intégré IC1, un TDA7266 de STMicroelectronics [5], se tire d'affaire presque sans composants externes, comme le montre le schéma fonctionnel de la **figure 2**. Son gain est fixé à 26 dB. Les tensions alternatives à l'entrée sont ainsi multipliées par 20. La puce limite le courant de sortie à 2 A. Elle se présente sous boîtier dit *multiwatt* à 15 broches et est alimentée également

en tension asymétrique ( $V_{AA}$ ), elle produit en interne la tension médiane qui sert de masse virtuelle, mais qui n'est pas accessible de l'extérieur, bien que trois des broches soient inutilisées. On est ainsi obligé de prévoir deux condensateurs de découplage supplémentaires (C1 et C2).

À vrai dire, la masse virtuelle n'est établie que si la broche 7 (ST-BY) atteint au moins +1,8 V. Sur la broche 6 (MUTE), on peut avoir besoin d'au moins 4,1 V, donc plus que la tension de la partie numérique, et donc c'est T4 qui doit enclencher

**Tableau 1. Valeurs optimales d'amortissement.**

$V_{analogique} [V_{CC}]$	R18, R22 [kΩ]	R19, R23 [kΩ]
5	27	3,3
12	27	10
18	15	10

la sourdine. R24 limite le courant dans le transistor. Comme dans nombre de chaînes hi-fi, les haut-parleurs sont attaqués directement, sans condensateurs de liaison, par K1 et K2. Théoriquement, chaque ampli a une puissance de sortie de 7 W, mais tout dépend de la tension d'alimentation, cf. le **tableau 2**.

Avec le régulateur approprié IC3 de 3,3 V dans la partie numérique pour alimenter le microcontrôleur et la carte SD, le circuit est prêt à fonctionner. Alors, à quoi servent les autres composants ? La tension d'alimentation de l'étage de sortie audio IC1 est enclenchée par T2 quand il le faut. Cela réduit la consommation au repos. La LED2 s'allume quand l'ampli est sous tension. Les broches FT de IC2, parmi lesquelles on compte PA12,

résistent au 5 V, avec 5,2 V comme maximum. T3 sert d'adaptateur de niveau. Sans lui, si PA2 devait commander directement T2 à travers R8, particulièrement en mode de veille alors qu'un faible courant traverse PA12, la chute de potentiel sur R8 serait très faible, la tension sur la broche du contrôleur serait tellement élevée qu'il risquerait d'être bousillé.

Le contrôleur dispose de plusieurs niveaux de consommation. Le mode le plus économique est le sommeil profond appelé *Standby* dans lequel, selon le constructeur, il ne prend que 4  $\mu$ A à température ambiante. Cependant, on ne peut l'extraire de ce mode que par un flanc positif sur PA0, une impulsion de RàZ sur la broche 7 ou une RàZ complète

(*hardreset*). Petit truc : reliez la ligne SDA à l'entrée de réveil INT sur K4, cela lance une séquence de démarrage sur le bus I<sup>2</sup>C qui tire le contrôleur de son sommeil (*Standby*). On peut encore activer d'autres modes de puissance par le logiciel, par exemple pour la commande par RS485. On trouve le bus I<sup>2</sup>C aux broches 1 et 2 de K4, tandis que le bus RS485 est disponible sur les broches 1 et 2 de K3.

Le courant de repos du régulateur de tension IC3, avec environ 10 mA, est relativement fort. S'il est trop élevé pour certaines applications, il convient d'éteindre l'alimentation numérique de l'extérieur. Le cavalier JP1 n'est là que pour les tests. Le contrôleur reçoit son signal d'horloge à 12 MHz du circuit oscillant externe X1, C1 et C2. La résistance R9 en élève le facteur de mérite et minimise la gigue (l'instabilité temporaire de la cadence de MLCK), ce qui améliore la qualité du son. Nous voudrions que la carte du module son devienne aussi universelle que possible. Mais on ne peut pas tout résoudre dans le logiciel. Pour ajouter des fonctions de commande déterminées, il faudra faire chauffer le fer à souder pour apporter des modifications au matériel (**tableau 3**). Quelques petites modifications de valeurs permettent de vouer un canal à l'aigu, l'autre au grave. À titre d'exemple, les valeurs des composants sont rassemblées dans le **tableau 4** avec les domaines de fréquence simulés.

**Tableau 2. Puissance de sortie pour différentes tensions d'alimentation de l'étage final.**

Impédance H-P [ $\Omega$ ]	P <sub>sortie</sub> par canal [W]	V <sub>analogique</sub> [V <sub>CC</sub> ]
4	7	7,5 à 18
8	7	11 à 18
16	7	15 à 18
4	3	5
8	1,5	5

**Tableau 3. Fonctions qui demandent des travaux de soudure.**

Fonction	composants nécessaires	ne pas souder
RS485 transmission commandée par PA1	IC4, C3	R1, R3, C5
RS485 commande par matériel	IC4, C3, R1	R3, C5
RS485 terminaison de bus	R2	./.
UART en mode 3,3 V sur br. 4 et 5 de K3	R3	IC4, C3, R1, R2
wake-up par flanc L→H et commutation sur K4 broche 8	R4	T1
wake-up par flanc H→L et commutation sur K4 broche 8	T1	R4

**Tableau 4. Réglage stéréo ou grave/aigu pour 12 V<sub>CC</sub> d'alimentation.**

Nature du travail :	stéréo	grave	aigu
domaine de fréq. à -3 dB	20 Hz à 65 kHz	19 Hz à 1,7 kHz	560 Hz à 65 kHz
C25, C27	3,3 $\mu$ F	4,7 $\mu$ F	33 nF
C26, C28	3,9 nF	22 nF	4,7 nF
C1, C2	1 $\mu$ F	3,3 $\mu$ F	33 nF
R17, R21	470 $\Omega$	4,7 k $\Omega$	470 $\Omega$
R18, R22	27 k $\Omega$	18 k $\Omega$	18 k $\Omega$

### Le logiciel et son développement

Le logiciel est en C, il utilise les bibliothèques CMSIS et FatFS que l'on peut télécharger de la page du projet [2]. La plupart des fichiers ont été créés avec le programme de configuration STM32CubeF4 [6], ce qui rend l'étude du manuel de référence du STM32F401 largement superflue. L'application proprement dite a aussi trouvé sa source dans le fichier `main.c`. La reproduction sonore est implémentée dans le fichier `wave-player.c` : on envoie un fichier tampon au convertisseur N/A IC5 en DMA par le bus I<sup>2</sup>S. Après transmission de la moitié du tampon, on écrit l'autre moitié avec les données PCM suivantes du fichier WAV. Une adaptation du volume intervient par simple multiplication dans l'unité à virgule flottante (FPU). On y fait aussi appel pour la fonction *Fade out* pour le fondu final du son. Cela donne au total un flux de données I<sup>2</sup>S

sans intervalle, ce qui épargne encore des ressources de calcul pour d'autres extensions du programme, par exemple pour une fonction *Loudness*. Les fichiers restants concernent principalement des données auxiliaires avec fonctions particulières spéciales.

On l'a dit, il s'agit ici d'un projet, matériel et logiciel, purement ouvert. En outre, les outils de développement sont bien sûr, eux aussi, libres et gratuits. Il faut malgré tout s'inscrire gratuitement sur le site avant le téléchargement. Les outils sont les suivants :

- KiCad [7] pour le schéma et le dessin du matériel
- TINA-TI [8] pour des simulations spéciales du matériel (elles fonctionnent aussi sous Linux et en environnement Wine)
- l'environnement compilateur et éditeur de liens SW4STM32 [9], connu aussi comme OpenSTM32
- STVP-STM32 [10] pour la programmation de IC2 avec le fichier hexa fait pour lui.

Depuis les fichiers sources, on fait un appel Make dans le dossier `software/cube/SW4STM32` qui produit le fichier `CardSound.hex`. Comme matériel de programmation, on peut prendre par exemple une carte d'évaluation Nucleo, à laquelle on adjoint une interface SWD à usage externe. K6 reproduit le brochage à dix contacts de l'habituel débogueur Cortex [11].

Les protocoles de bus sont (presque) indénombrables. Dans l'exemple de logiciel, on a implémenté l'ElektorBus [12] en mode dit A0 sans contrôle CRC. Ce bus se distingue par sa simplicité et son intelligibilité. Mais ce projet s'intéresse davantage à la reproduction sonore qu'aux protocoles de bus. Selon l'application, on peut adapter cette partie du logiciel à des besoins spécifiques. Les fragments de programme qui utilisent l'union des données `U_ELB` sont faciles à trouver, à modifier ou à supprimer, voire à remplacer par son logiciel de protocole favori.

### Configuration de la reproduction sonore

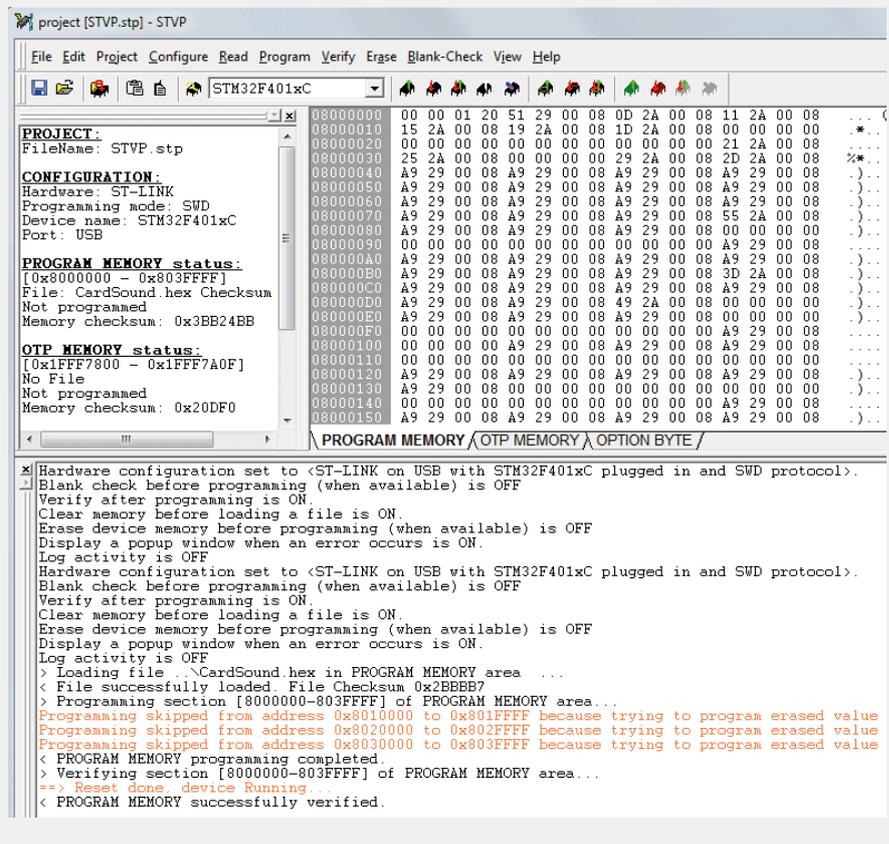
Si le transfert dans la commande I<sup>2</sup>C ou RS485 du nom de fichier son à lire est évident, le logiciel actuel suit un autre chemin. Il lit le fichier d'affectation `map`.

## Programmation du microcontrôleur

La carte STM32F401 Nucleo 64 est un matériel avantageux pour la programmation du microcontrôleur. Il faut lui ôter les deux cavaliers appelés ST-LINK, puis on relie les trois contacts de CN4 SWCLK, GND et SWDIO avec les broches de même nom de K6 sur le lecteur Card Sound.

broche de CN4 sur Nucleo 64	broche sur K6	fonction
2	4	SWCLK
3	3, 5 bzw. 9	GND(D)
4	2	SWDIO

On relie ensuite la tension d'alimentation (5 à 10 V) à la broche 6 de K4. Après installation du programmateur ST Visual [10], on branche CN1 de Nucleo 64 sur l'USB du PC. Le pilote s'installe automatiquement. Par *Project* → *Open...*, on ouvre le fichier `STVP.stp` du paquet logiciel [2] et on choisit *Program* → *All tabs*. La réussite de la programmation est indiquée par le message *PROGRAM MEMORY successfully verified*.



```
csv. Chacune de ses lignes se présente
comme suit :
mot de commande, nom de fichier son, volume
# Index, file name, volume (1000 = 100%)
1,audio1.wav,1000
5,audio2.wav,500
10,empty.wav,1000
```

Dans l'exemple suivant, une commande avec la valeur 1 fait jouer le fichier `audio1.wav` à pleine puissance et avec la commande 5 le fichier `audio2.wav` à demi-puissance (le signe # marque le

début d'un commentaire). Il n'y a pas dans le logiciel d'exemple de commande d'arrêt, on donne pour cela

une commande de sélection du fichier son [empty.wav](#). Il est bien présent sur la carte SD, mais ne contient aucun échantillon. L'avantage de cette méthode de configuration plutôt que de programmation réduit manifestement le volume du

code de programme. Card Sound peut décoder des données audio aux formats 16 bits, 24 bits et 32 bits (2 canaux ou stéréo) avec des taux d'échantillonnage de 32 kHz, 44,1 kHz, 48 kHz, 96 kHz et 128 kHz. Le maté-

riel audio actuel convertit facilement l'un de ces formats avec les programmes Audacity [13] et ocenaudio [14]. Avec Audacity, on choisit Fichier → Audio exporter... → Type de fichier ; on choisit par exemple : WAV (Microsoft) signed



## LISTE DES COMPOSANTS

### Résistances :

(\* : cf. tableau 2)

R1\*, R4\* = 0 Ω, film épais, 5 %, 0,1 W, 150 V, 0805

R2\* = 120 Ω, film épais, 1 %, 0,25 W, 200 V, 1206

R3\*, R10 = 1 kΩ, film épais, 5 %, 0,1 W, 150 V, 0805

R5, R7 = 100 kΩ, film épais, 5 %, 0,1 W, 150 V, 0805

R6, R16, R19, R20, R23, R25 = 10 kΩ, film épais, 1 %, 0,25 W, 200 V, 1206

R8, R24, R26 = 10 kΩ, film épais, 5 %, 0,1 W, 150 V, 0805

R9 = 220 Ω, film épais, 5 %, 0,1 W, 150 V, 0805

R11 = 4,7 kΩ, film épais, 5 %, 0,1 W, 150 V, 0805

R12, R13, R14 = 51 Ω, film épais, 5 %, 0,1 W, 150 V, 0805

R15 = 24 Ω, film épais, 1 %, 0,1 W, 150 V, 0805

R17, R21 = 470 Ω, film épais, 1 %, 0,25 W, 200 V, 1206

R18, R22 = 27 kΩ, film épais, 1 %, 0,25 W, 200 V, 1206

### Condensateurs :

C1, C2 = 1 μF, 63 V PET, au pas de 5 mm  
C3\*, C5\*, C13, C14, C15, C20, C21, C23 = 100 nF, X7R, 10 %, 100 V, 1206

C4 = 10 nF, X7R, 10 %, 50 V, 1206

C6 = 4,7 μF, Y5V, 16 V, 1206

C7, C8, C31, C33 = 10 μF, X7R, 10 V, 1206

C9, C16, C17, C18, C19, C29 = 100 nF, 50 V, X7R, 0805

C10 = 10 μF, X5R, 10 V, 0805

C11, C12 = 22 pF, 50 V, COG/NPO, 0805

C22 = 820 μF, 25 V, radial, 20 %, 10 mm, au pas de 5 mm

C24 = 3,3 μF, 25 V, X7R, 1206

C25, C27 = 3,3 μF, 63 V PET, au pas de 5 mm

C26, C28 = 3,9 nF, 250 V, 5 %, PPS

C30, C32 = 1 μF, 16 V, 10 %, X7R, 1206

### Inductance :

L1 = perle de ferrite 0,4 Ω, 200 mA, Murata BLM21BD102SN1D

### Semi-conducteurs :

LED1 = LED, vert, 3 mm

LED2 = LED, jaune, 3 mm

T1\* = BSS84 MOS canal P, -170 mA, -60 V, 8 Ω @ -10 V, -1,5 V

T2 = IRLML5203 MOS canal P, -3 A, -30 V, 98 mΩ @ -10 V, -2,5 V

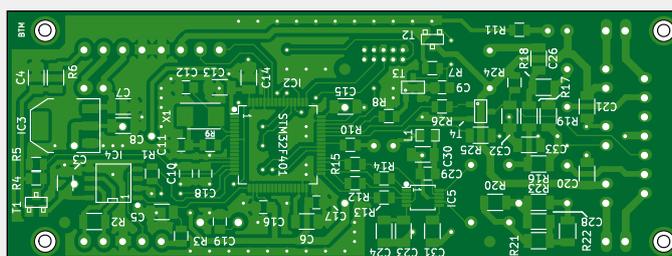
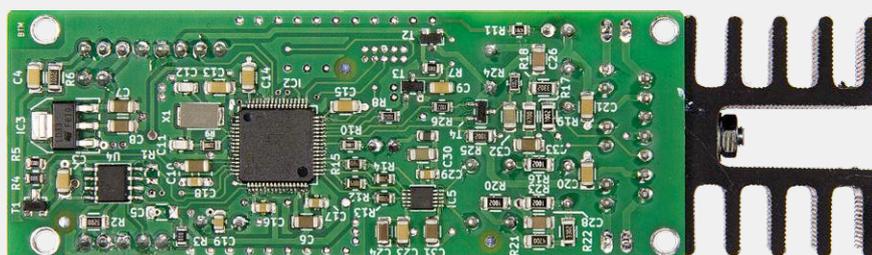
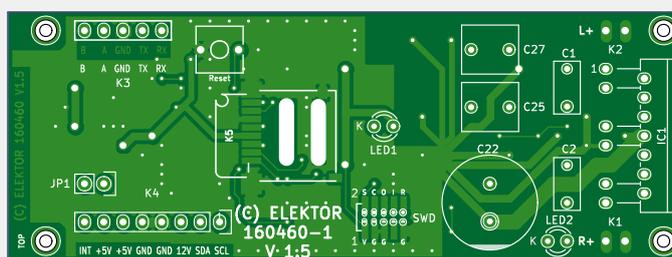
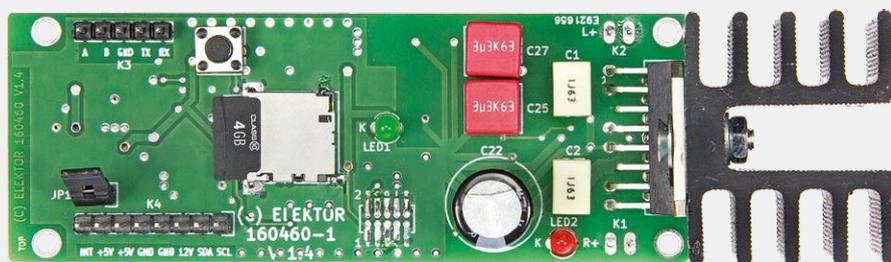


Figure 3. Les deux circuits imprimés (version 1.50) du lecteur Card Sound. Photos de la version 1.40.

T3, T4 = MMBT2222, NPN, 40 V, 600 mA

IC1 = TDA7266 (ampli audio)

IC2 = STM32F401RCT6 ARM Cortex-M4

IC3 = LD1117S33TR (3,3-V-LDO, 800mA)

IC4\* = SP3485CN (trancepteur RS422/RS485)

IC5 = CS4344-DZZ (CN/A audio 24 bits 192 kHz)

### Divers :

X1 = quartz 12 MHz, CMS, 5 mm × 3,2 mm

K1, K2 = embase à 1×2 picots droits,

au pas de 2,54 mm

K3 = embase à 1×5 picots droits,

au pas de 2,54 mm

K4 = embase à 1×8 picots droits,

au pas de 2,54 mm

K5 = socle micro-SD encartable, Hirose DM3CS-SF

K6 = embase à 2×5 picots droits,

au pas de 1,27 mm

S1 = bouton-poussoir 6 mm, bipolaire,

p.ex. FSM4JRT (TE Connectivity)

circuit imprimé, réf. 160460-1 V1.5

16-bit PCM. Avec ocenaudio on choisit : Fichier→Exporter...→WAV le format Linear PCM.

Avant la conversion, il faudrait vérifier la dynamique du fichier son à écouter et, le cas échéant, pousser le volume au maximum. On sait que les données au format de 16 bits peuvent aller de -32768 à 32767.

Le logiciel accepte les cartes de type SDSC et SDHC. Celles-ci, si ce n'est déjà fait d'origine, doivent être formatées en FAT pour pouvoir y copier les fichiers CSV et WAV.

### Construction et mise en service

Si vous avez un minimum d'expérience du travail sur CMS, vous pouvez assembler vous-même le lecteur. Sur le circuit imprimé (**figure 3**) de l'e-choppe, le plus petit écartement entre les pistes pour IC2 et IC5 est d'environ 0,5 mm. Les résistances et condensateurs les plus petits sont au format 0805. On réussit leur mise en place au moyen d'une lampe loupe, de fil de soudure mince riche en flux utilisé avec parcimonie, d'une pincette et d'un fer à souder à panne fine de 0,5 mm ou 1 mm. Utilisez les moyens de protection contre les décharges électrostatiques (ESD), gants et bracelet de mise à la terre, pour éviter des dégâts inopinés. Selon le but de l'application, il y a des composants qu'il ne faut pas installer. Commencez par les CMS, à l'exception des condensateurs C28 et C28 qui craignent la chaleur. Ensuite les composants à fils, pour terminer par les deux condensateurs CMS cités.

Avant de souder IC1, il faut lui couper deux pattes, la 5 et la 11 (**figure 4**). La numérotation des broches est effectuée de gauche à droite, vue du côté de l'impression. Si vous présentez la puce en face des trous, vous verrez directement quelles pattes sont concernées. Le but est en fait d'améliorer le parcours des pistes adjacentes.

Pour la première mise en service, on ne relie que l'alimentation de la partie numérique aux broches 5 (GND) et 6 (+5V<sub>CC</sub>). Si tout est correctement câblé, les puces dans le bon sens, la consommation devrait rester sous 100 mA. Sinon, éteignez vite l'alimentation et entreprenez les recherches d'anomalie.

L'étape suivante est la programmation de IC2 avec le fichier [CardSound.hex](#) par

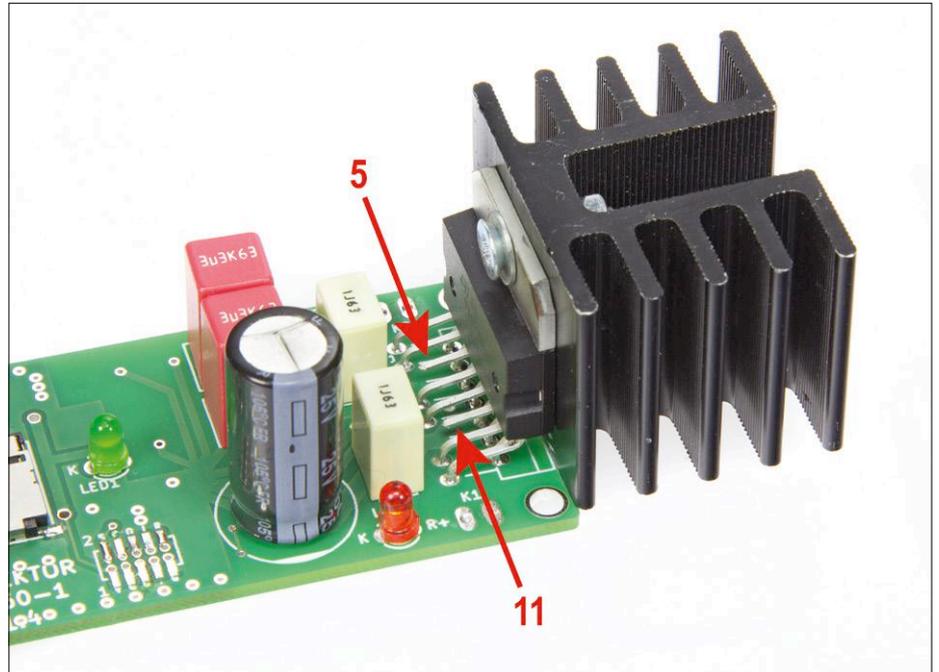


Figure 4. La puce amputée : deux pattes en moins à l'amplificateur pour faire une carte plus simple et plus esthétique.

le connecteur K6 de programmation et débogage. L'encadré sur la programmation vous en dira plus. Après une action sur le bouton de RàZ S1, la LED1 devrait s'allumer. Insérez d'abord la carte SD sur laquelle vous avez copié le fichier [map.csv](#) et le fichier d'exemple WAV depuis le paquet logiciel. Puis appliquez aussi la tension d'alimentation sur la broche 3 de K4 et branchez un haut-parleur d'essai

est la valeur réservée à la synchronisation du flux de données. Les octets restants n'ont aucune fonction définie, par souci de simplicité. Pour le bus I<sup>2</sup>C, c'est le fichier [i2c.c](#) qui indique l'adresse d'appareil 0x40 au format de 7 bits. Dans le logiciel, l'adresse 0x40 est décalée d'un bit vers la gauche, ce qui donne 0x80. De cette manière, l'adresse d'appareil est compatible avec la bibliothèque

▶ avec un peu d'expérience des CMS, vous pouvez construire ce lecteur

à K1 ou K2. Poussez de nouveau sur le bouton S1 de RàZ en espérant entendre le son de test sur le haut-parleur. Alors, le matériel est bon pour le service.

Le module reçoit ses commandes indifféremment par bus I<sup>2</sup>C ou RS485. La tension du signal peut se situer entre 3 et 5 V sur le bus I<sup>2</sup>C comme sur l'interface UART. Le protocole de l'ElektorBus spécifie juste 16 octets. Le premier, pour la synchronisation, a toujours la valeur 0xAA. Le deuxième octet détermine le mode, dans notre cas, c'est 0xA0. Le troisième octet définit l'indice dans le [map.csv](#), il peut prendre n'importe quelle valeur de 0 à 255, sauf 170 = 0xAA qui

de logiciel de ST Micro.

Bien entendu, on peut toujours adapter le logiciel à ses propres besoins, par exemple pour un simple contrôle de la tension aux broches 4 et 5 de K3 (fonction GPIO) ou l'envoi d'un télégramme de réponse au début et à la fin de la reproduction sonore.

Pour jouer des fichiers son courts, de moins d'une demi-minute, entrecoupés de longues pauses, il ne faut pas vraiment de radiateur, mais il est préférable de visser directement sur IC1 un radiateur de moins de 3 K/W, garant de ce que rien ne grillera ! La partie métallique du boîtier est reliée intérieurement

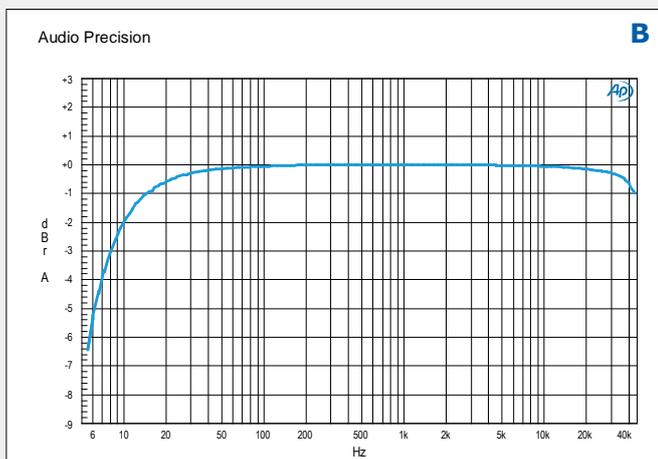
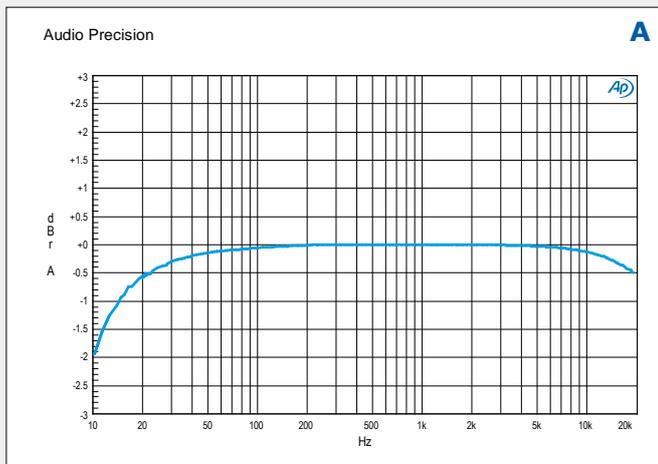
## Mesures

Le laboratoire d'Elektor a effectué des mesures sur le lecteur Card Sound (encore avec une ancienne version du logiciel) au moyen d'un analyseur audio de marque AudioPrecision. Il est apparu alors que les différents formats de fichiers ne s'harmonisaient pas bien avec toutes les possibilités de taux d'échantillonnage, mais dans la nouvelle version le défaut a été corrigé. Sur le lecteur Card Sound à l'essai, C26 et C28 étaient encore du type céramique. Avec des condensateurs PPS à leur place, le taux de distorsion a encore été abaissé.

- Fréquence d'échantillonnage : 48 kHz
- Format : PCM 24 bits avec signe et non comprimé
- R18, R22 = 33 k $\Omega$  (modifié plus tard en 27 k $\Omega$  pour augmenter le gain et en même temps réduire un peu la consommation)
- Pleine puissance = 1 000 (correspond à 100 %)

Nous avons également enregistré quelques courbes caractéristiques :

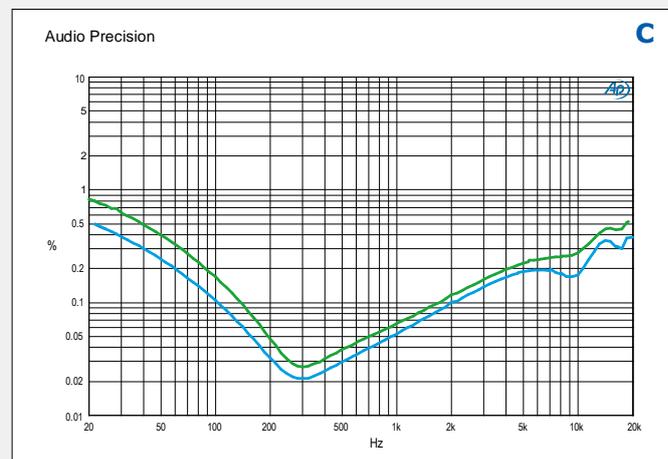
**Courbe A** : l'amplitude en fonction de la fréquence. Le signal mesuré est une onde sinusoïdale dont la fréquence se modifie 100 fois à la puissance de sortie de 1 W sur une charge de 8  $\Omega$ , la fréquence d'échantillonnage est de 48 kHz (24 bits avec signe en PCM). Le résultat est une belle courbe plate avec un affaiblissement de 0,6 dB à 20 Hz comme à 20 kHz.



<b>DHT+b, 1,1 W/8 <math>\Omega</math></b>	100 Hz	0,12 % (B = 22 kHz)
		0,12 % (B = 80 kHz)
	1 kHz	0,023 % (B = 22 kHz)
		0,049 % (B = 80 kHz)
	7 kHz	0,041 % (B = 22 kHz)
		0,18 % (B = 80 kHz)
<b>DHT+b, 2,2 W/4 <math>\Omega</math></b>	100 Hz	0,18 % (B = 22 kHz)
		0,18 % (B = 80 kHz)
	1 kHz	0,045 % (B = 22 kHz)
		0,063 % (B = 80 kHz)
	7 kHz	0,16 % (B = 22 kHz)
		0,26 % (B = 80 kHz)
<b>S/b (pour 1,1 W/8 <math>\Omega</math>)</b>		92 dB (B= 22 kHz)
		94 dB (pondération A)
<b>Consommation</b>	5 V	0,026 A (sans signal)
		0,07 A (avec signal)
	12 V	0 A (éteint)
		0,64 A (2 x 1,1 W/8 $\Omega$ )
		1,1 A (2 x 2,2 W/4 $\Omega$ )

**Courbe B** : l'amplitude en fonction de la fréquence. Le signal mesuré est une onde sinusoïdale dont la fréquence se modifie 100 fois à la puissance de sortie de 1 W sur une charge de 8  $\Omega$ , la fréquence d'échantillonnage est de 96 kHz (16 bits avec signe en PCM). L'atténuation à 20 Hz se situe aussi à près de 0,6 dB, tandis qu'à 20 kHz, la valeur est d'environ 0,15 dB plus faible que dans la courbe A. Pour 45 kHz, l'atténuation frôle le 1 dB.

**Courbe C** : on voit la distorsion harmonique totale plus le bruit (DHT+b) sur une largeur de bande de 80 kHz en fonction de la fréquence. Le signal mesuré est une onde sinusoïdale dont la fréquence se modifie 100 fois entre 20 Hz et 20 kHz (24 bits avec signe en PCM). La courbe se situe bien à l'intérieur des spécifications du TDA7266.

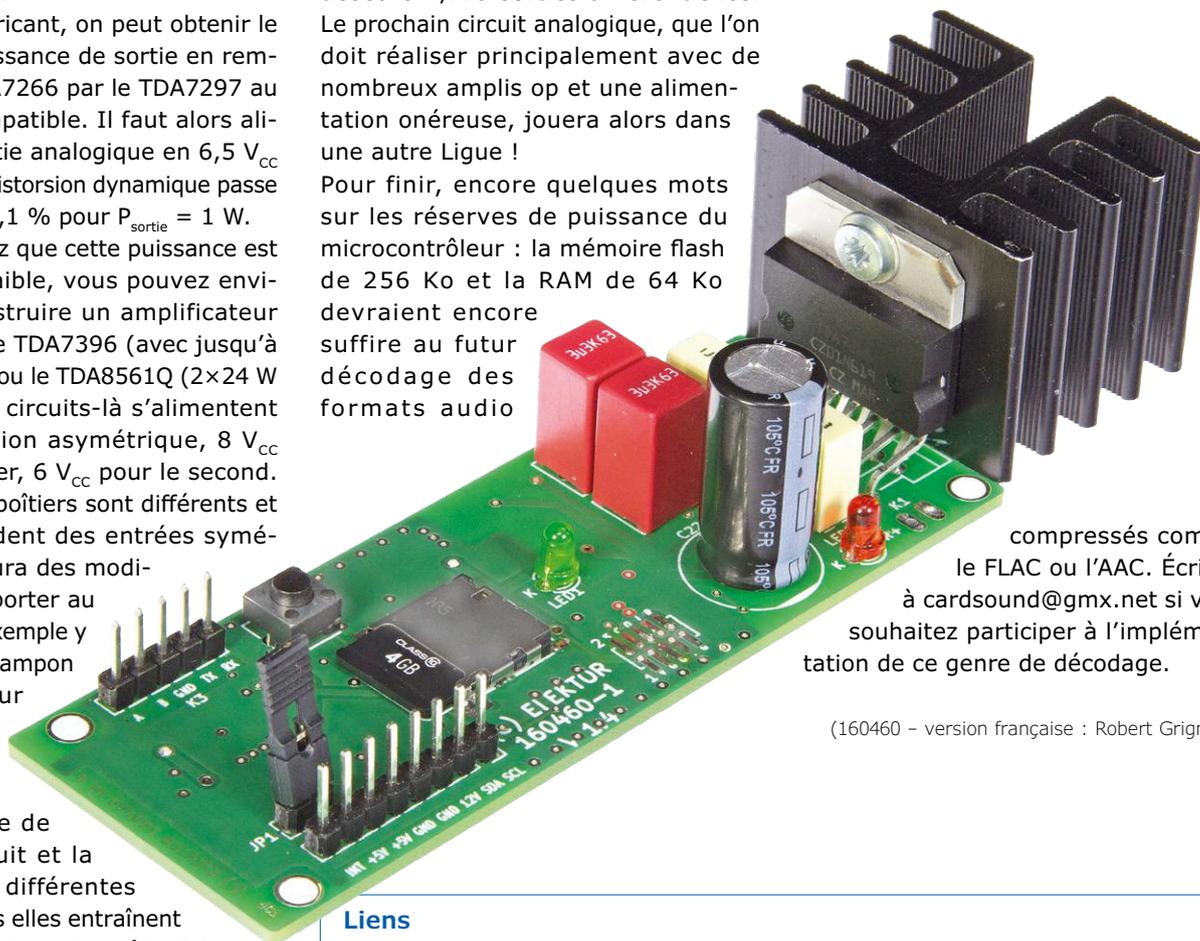


à la broche 8 (Power GND), ce qui peut nécessiter l'isolation électrique du radiateur, selon le mode de construction de l'ensemble.

### Variantes de circuit et optimisation

D'après le fabricant, on peut obtenir le double de puissance de sortie en remplaçant le TDA7266 par le TDA7297 au brochage compatible. Il faut alors alimenter la partie analogique en  $6,5 V_{CC}$  minimum. La distorsion dynamique passe de 0,05 % à 0,1 % pour  $P_{\text{sortie}} = 1 \text{ W}$ . Si vous trouvez que cette puissance est encore trop faible, vous pouvez envisager de construire un amplificateur monophonique TDA7396 (avec jusqu'à 60 W sur  $2 \Omega$ ) ou le TDA8561Q ( $2 \times 24 \text{ W}$  sur  $4 \Omega$ ). Ces circuits-là s'alimentent aussi en tension asymétrique,  $8 V_{CC}$  pour le premier,  $6 V_{CC}$  pour le second. Comme leurs boîtiers sont différents et qu'ils demandent des entrées symétriques, il y aura des modifications à apporter au circuit et par exemple y incorporer un tampon à amplificateur opérationnel avec sorties différentielles. Il est possible de réduire le bruit et la distorsion de différentes manières, mais elles entraînent une augmentation du coût. L'alimentation relativement simple à LD1117 donne  $100 \mu V_{\text{eff}}$  de bruit sur  $V_{CC}$ . Prendre une plus grande capacité pour C8 peut rendre moins audible le bruit de grésillement. On peut aussi alimenter la partie numérique par une source stabilisée et filtrée entre 3,0 et  $3,45 V_{CC}$ , et remplacer IC3 par un pont de câblage ou une résistance de  $0 \Omega$  entre les broches 2 et 3. La gigue aussi sur la ligne d'horloge maître (MCK), qui conduit inévitablement à des distorsions dans IC5, mérite un examen plus approfondi. Le fabricant du microcontrôleur IC2 donne  $25 \text{ ps}_{\text{eff}}$  pour la boucle à phase asservie (PLL) à la fréquence d'horloge maximale. Ce n'est déjà pas si mal, mais avec un bon générateur d'horloge externe, comme un AW-11.2896MBE-T et l'adaptation correspondante du circuit, on peut atteindre une valeur inférieure à 2 ps. Avec l'horloge fixe, la fréquence d'échantillonnage du matériel audio utilisé serait alors de 44,1 kHz.

De meilleures marges au bruit que celle du CS4344 donnée à  $-90 \text{ dB}$  à pleine amplitude sont offertes par le WM8716 avec  $-97 \text{ dB}$  ainsi que le CS4361 avec  $-94 \text{ dB}$ . Des valeurs encore supérieures ne sont atteintes que par des convertisseurs N/A à sorties différentielles. Le prochain circuit analogique, que l'on doit réaliser principalement avec de nombreux amplis op et une alimentation onéreuse, jouera alors dans une autre Ligue ! Pour finir, encore quelques mots sur les réserves de puissance du microcontrôleur : la mémoire flash de 256 Ko et la RAM de 64 Ko devraient encore suffire au futur décodage des formats audio

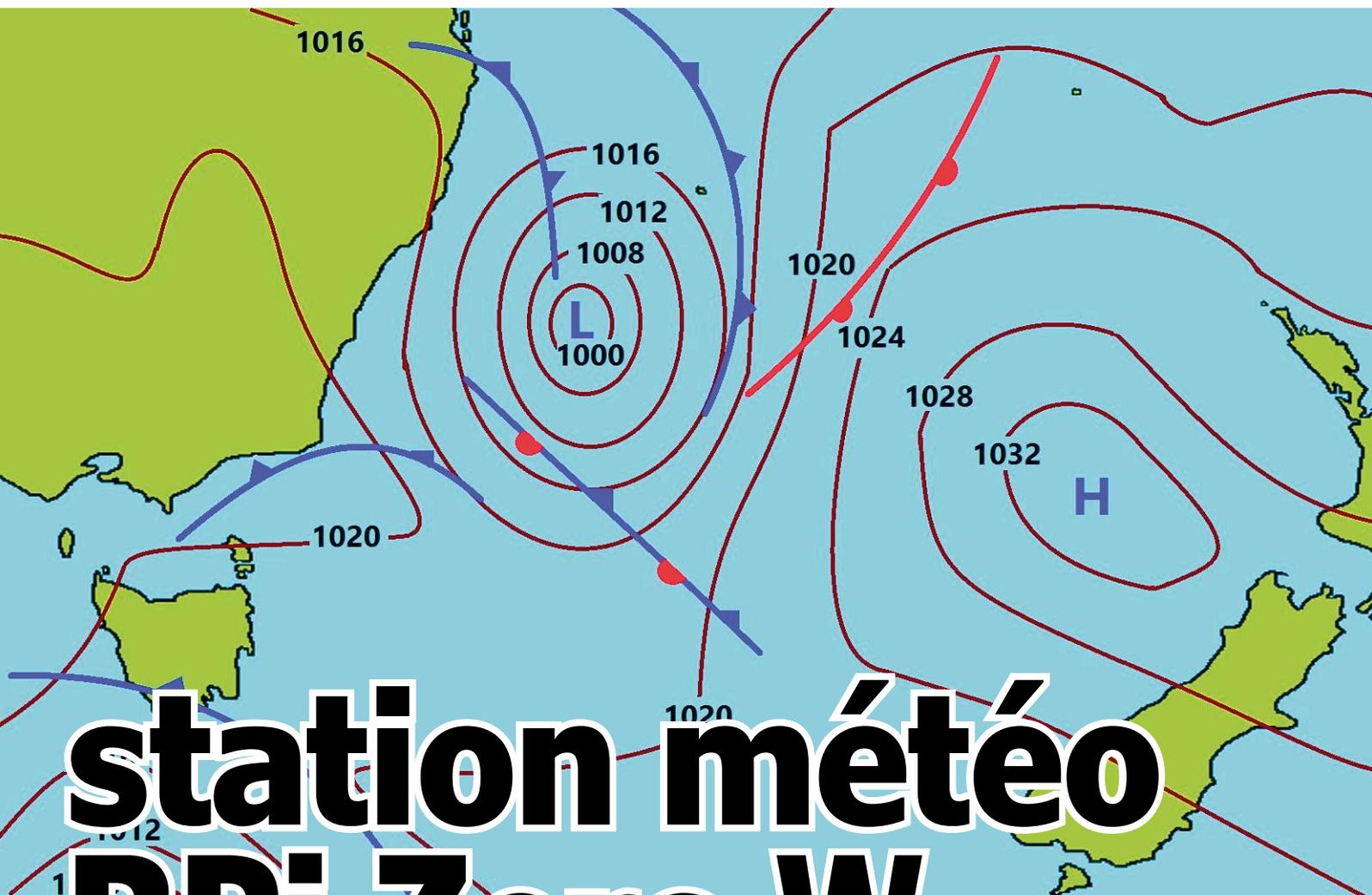


compressés comme le FLAC ou l'AAC. Écrivez à [cardsound@gmx.net](mailto:cardsound@gmx.net) si vous souhaitez participer à l'implémentation de ce genre de décodage. ◀

(160460 – version française : Robert Grignard)

### Liens

- [1] Physical Layer Specification Version 5.00 : [www.sdcard.org](http://www.sdcard.org)
- [2] Page du projet dans le labo : [www.elektormagazine.fr/labs/card-sound](http://www.elektormagazine.fr/labs/card-sound)
- [3] Carte d'évaluation pour CS4344 : <https://d3uzseaevmutz1.cloudfront.net/pubs/rdDatasheet/CDB4344-EB.pdf>
- [4] Fiche technique du CS4344/5/8 : [https://d3uzseaevmutz1.cloudfront.net/pubs/proDatasheet/CS4344-45-48\\_F2.pdf](https://d3uzseaevmutz1.cloudfront.net/pubs/proDatasheet/CS4344-45-48_F2.pdf)
- [5] Fiche technique du TDA7266 : [www.st.com/resource/en/datasheet/tda7266.pdf](http://www.st.com/resource/en/datasheet/tda7266.pdf)
- [6] STM32CubeF4 : [www.st.com/content/st\\_com/en/products/embedded-software/mcus-embedded-software/stm32-embedded-software/stm32cube-embedded-software/stm32cubef4.html](http://www.st.com/content/st_com/en/products/embedded-software/mcus-embedded-software/stm32-embedded-software/stm32cube-embedded-software/stm32cubef4.html)
- [7] KiCad : <http://kicad-pcb.org/>
- [8] TINA-TI : [www.ti.com/tool/TINA-TI](http://www.ti.com/tool/TINA-TI)
- [9] SW4STM32 : [www.st.com/en/development-tools/sw4stm32.html](http://www.st.com/en/development-tools/sw4stm32.html)
- [10] STVP-STM32 : [www.st.com/content/st\\_com/en/products/development-tools/software-development-tools/stm32-software-development-tools/stm32-programmers/stvp-stm32.html](http://www.st.com/content/st_com/en/products/development-tools/software-development-tools/stm32-software-development-tools/stm32-programmers/stvp-stm32.html)
- [11] Cortex Debug Connector : [http://infocenter.arm.com/help/topic/com.arm.doc.faqs/attached/13634/cortex\\_debug\\_connectors.pdf](http://infocenter.arm.com/help/topic/com.arm.doc.faqs/attached/13634/cortex_debug_connectors.pdf)
- [12] ElektorBus : [www.elektormagazine.fr/100817](http://www.elektormagazine.fr/100817) et articles suivants
- [13] Audacity : [www.audacityteam.org](http://www.audacityteam.org)
- [14] Ocenaudio : [www.ocenaudio.com](http://www.ocenaudio.com)



# station météo

# RPi Zero W

## sans pièces mécaniques mobiles

Zeno Otten (Pays-Bas)

De nombreux aspects de notre vie quotidienne sont régis par le temps qu'il fait : prendre son parapluie ou pas, aller à pied ou en voiture, ouvrir la fenêtre ou la fermer, faire la grasse matinée ou se lever, etc. Un simple coup d'œil par la fenêtre suffit pour avoir un aperçu des conditions actuelles, mais seule la mesure permet de faire des prévisions, et notre station météo est parfaite pour ça.

La réalisation d'une station météo n'est pas très rentable de nos jours, du moins si on ne considère que les aspects financiers. On trouve dans le commerce des appareils *made in Asia* bon marché, mais

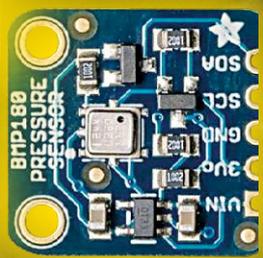
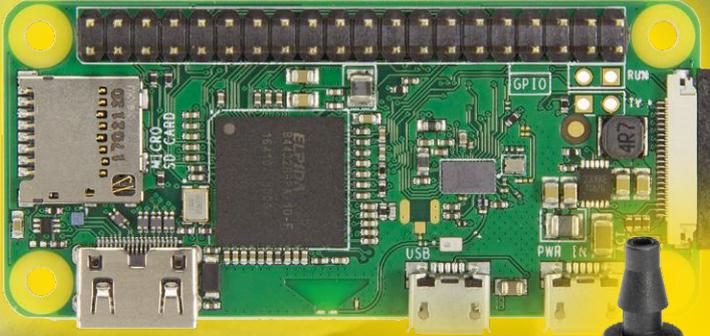
qui n'en sont pas moins performants, et un électronicien amateur ne peut pas les concurrencer. Ces appareils ont cependant quelques inconvénients, ce qui justifie l'approche *DIY* (*do it yourself*), en

dehors de la satisfaction que l'on éprouve à fabriquer quelque chose soi-même. Pour mesurer la vitesse du vent, on utilise généralement une espèce de rotor avec des coupelles, et pour détecter sa



## Caractéristiques

- mesure de la température, de l'humidité et de la luminosité
- mesure de la vitesse et de la direction du vent, sans pièce mobile
- calcul du point de rosée, de la température ressentie et de la hauteur des nuages
- plateforme Raspberry Pi Zero



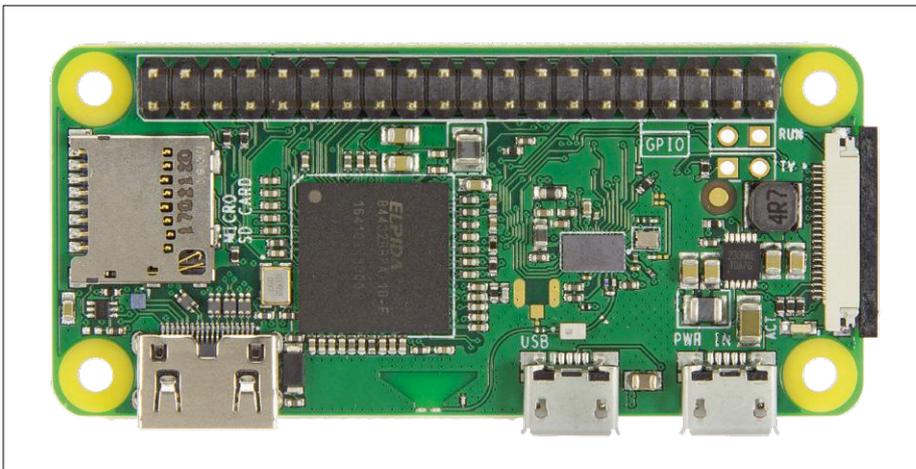


Figure 1. Le Raspberry Pi Zero W.

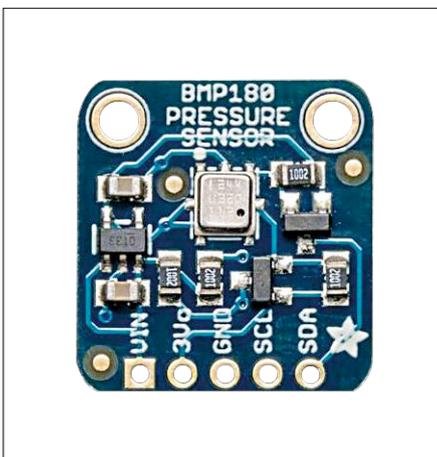


Figure 2. Le capteur de pression et de température BMP180.



Figure 3. Le capteur de température et d'humidité HYT221.

- possibilité d'extension (par ex. pour un pluviomètre) ;
- logiciel compréhensible et évolutif.

### Le projet

Au vu de notre cahier des charges, le choix du cœur de notre circuit a été vite fait. Le Raspberry Pi Zero W est parfait pour la tâche : ce microcontrôleur a suffisamment de broches d'E/S, une bonne puissance de calcul, et tout ce qu'il faut pour réaliser facilement une liaison sans fil avec un réseau local ou internet. Si on ajoute que sa taille est celle d'une carte de crédit (**figure 1**) et qu'il ne coûte que 10 €, pas besoin de chercher plus loin. Elektor nous a présenté la « bête » dans le numéro de septembre/octobre [1], et un kit de démarrage complet [12] est disponible dans l'e-choppe pour ceux qui n'ont pas encore d'expérience du RPi (Zero).

Les capteurs communiquent avec le RPi Zero via le bus I<sup>2</sup>C ; deux fils sont aussi nécessaires pour leur alimentation. Le bus I<sup>2</sup>C, développé initialement par Philips, est très populaire ; il permet la communication entre microcontrôleurs, systèmes complets ou divers circuits intégrés via deux fils : *SDA* (*Serial Data*) pour les données, et *SCL* (*Serial Clock*) pour l'horloge. Avec l'alimentation, nous aurons donc des liaisons à quatre fils avec les capteurs.

Le protocole de communication est un système maître-esclave : le maître, dans notre cas le RPi, s'adresse à un esclave (un des capteurs) et lui demande la transmission des données disponibles (par ex. la température). Chaque esclave dispose d'une adresse unique, que l'utilisateur peut en général – et au moins partiellement – choisir lui-même.

Le bus utilise des résistances de rappel dont la valeur est de 4,7 à 10 k $\Omega$  ; les lignes *SDA* et *SCL* sont à l'état haut au repos. Ces résistances sont parfois intégrées au capteur, il faut donc veiller à ce que la valeur totale reste dans les limites acceptables.

### Les capteurs

La mesure de la pression atmosphérique, de la température, de l'humidité et de la luminosité ne pose pas de problème particulier ; c'est un peu plus compliqué pour la vitesse et la direction du vent, mais chaque chose en son temps...

### Pression atmosphérique

Pour la pression atmosphérique, nous uti-

direction un simple fanion. Ce sont des pièces mécaniques, qui vont s'user et qu'il faudra remplacer tôt ou tard (en fonction de leur qualité). Ces pièces mécaniques ne sont pas toujours proposées sous forme de pièces détachées, et les problèmes commencent. La plupart du temps, la communication entre les capteurs et la station de base est effectuée à l'aide d'émetteurs-récepteurs qui travaillent par ex. dans la bande 433 MHz, et avec un protocole « maison ». Il est dès lors très difficile de modifier ou d'étendre l'installation.

Enfin, certaines grandeurs dérivées, comme le point de rosée ou la température ressentie (ou indice de refroidissement éolien), ne sont pas disponibles.

### Le défi

Les raisons exposées ci-dessus étaient suffisantes pour relever le défi et concevoir sa propre station météo. Le cahier des charges est le suivant :

- pas de pièces mobiles ;
- mesure de la pression atmosphérique, de l'humidité, de la température, de la luminosité ambiante, de la vitesse et de la direction du vent ;
- calcul du point de rosée, de la température ressentie et de la hauteur des nuages ;
- utilisation du protocole I<sup>2</sup>C pour les capteurs ;
- tension d'alimentation de 3,3 V ;
- stockage des résultats des mesures dans une base de données ;
- affichage des résultats via une interface réseau ;
- communication Wi-Fi par réseau local ou internet ;



Figure 4. Le capteur de luminosité ambiante BH1750.

lisons un BMP180 [2], un capteur barométrique qui mesure la pression absolue entre 300 et 1 100 hPa (mbar). Comme la pression atmosphérique est inversement proportionnelle à l'altitude – ceux qui ont déjà escaladé l'Everest le savent bien, le capteur peut aussi être utilisé comme altimètre.

Le capteur est monté sur une petite carte de liaison (*Breakout Board – BoB*), **figure 2**. La résolution est de 12 bits, soit 0,03 hPa. La tension d'alimentation est de 3,3 V et les résistances de rappel du bus I<sup>2</sup>C sont déjà montées sur la BoB. La communication avec le RPi se fait via les lignes SDA et SCL.

Il est possible que le BMP180 soit difficile à trouver. On peut utiliser le BME280, qui est disponible dans l'e-choppe [3]. Une adaptation du microgiciel sera peut-être nécessaire.

### Température et humidité

Nous avons choisi un capteur robuste, le HYT221 [4], **figure 3** ; il est idéal pour un usage à l'extérieur. Il mesure l'humidité relative de 0 à 100% avec une précision de ±1,8%, et la température de –40 à +120 °C avec une précision

Tableau 1 : vitesse du vent	
vent violent à tempête	> 17 m/s
fortes bourrasques	14 à 17 m/s
forte brise	10 à 14 m/s
brise fraîche	8 à 10 m/s
brise légère à modérée	1,5 à 8 m/s
très légère brise	0,3 à 1,5 m/s
calme	0 à 0,2 m/s

de ±0,2 °C.

Les résistances de rappel pour le bus I<sup>2</sup>C sont intégrées au capteur, et la tension d'alimentation couvre la plage de 2,7 à 5,5 V.

### Luminosité ambiante

La mesure de la luminosité ambiante est utilisée pour détecter la nuit, la couverture nuageuse, ou l'ensoleillement ; ces caractéristiques sont converties en valeurs concrètes. Le capteur est un BH1750 [5], dont la plage de mesure s'étend de 1 à 65 535 lx. Le capteur est monté sur un BoB (**figure 4**), de même que les résistances de rappel pour le bus I<sup>2</sup>C. La tension d'alimentation varie

de 3,3 à 5 V.

Valeurs indicatives : 0,001 à 0,02 lx pour la nuit, 500 à 25 000 lx pour un ciel couvert, et 50 000 à 100 000 lx pour une journée ensoleillée.

### Direction et vitesse du vent

Nous expliquons brièvement dans un cadre séparé comment ces deux valeurs sont obtenues à l'aide de deux capteurs de pression différentielle.

La plage de mesure est l'élément essentiel pour le choix du capteur. Le **tableau 1** reprend les vitesses de vent courantes dans nos régions. Si nous considérons une vitesse de 20 m/s pour une tempête moyenne et une masse

## Intermédiaire : « Éole, ce fils de Poséidon... »

La plupart des anémomètres pour la maison ou le jardin utilisent une espèce de rotor avec des coupelles (par ex. des balles de tennis de table coupées en deux) pour mesurer la vitesse, et un fanion ou une girouette pour la direction ; bref, des pièces mobiles. À notre époque, où microcontrôleur et logiciel règnent en maître absolu, l'électronicien amateur a horreur de la mécanique ; il nous fallait donc trouver une solution à ce problème.

Cette solution saute aux yeux quand on réfléchit à ce qui maintient un avion en l'air. L'aile d'un avion est plus courbée sur le dessus que sur le dessous. Dans un même laps de temps, l'air sur le dessus de l'aile doit parcourir un trajet plus long, et il en résulte une différence de pression (elle est plus faible sur le dessus de l'aile). Cette différence de pression induit une force vers le haut, qui maintient l'avion en l'air. Ce principe a été formulé au 18<sup>e</sup> siècle par Daniel Bernoulli. Le tube de Pitot [11], utilisé en aéronautique pour mesurer la vitesse de l'air, repose sur un principe identique. La pression d'un fluide en mouvement, l'air par exemple, est mesurée face au flux et perpendiculairement à l'écoulement du fluide. La différence de pression permet de calculer la vitesse d'écoulement du fluide, en m/s, à l'aide de la formule établie par Bernoulli :

$$v = \sqrt{\frac{2\Delta p}{\rho}}$$

$\Delta p$  est la différence de pression, et  $\rho$  la masse volumique du fluide (1,3 kg/m<sup>3</sup> pour l'air).

Pour un anémomètre statique, c'est-à-dire sans pièces mobiles, nous utilisons un cylindre vertical autour duquel l'air s'écoule, comme représenté en **figure 5**.

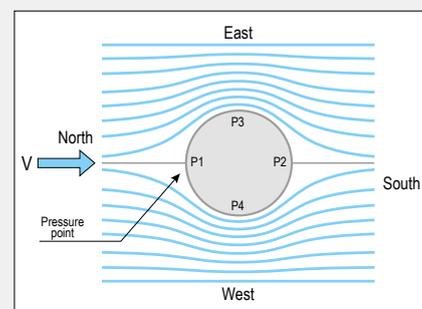


Figure 5. Écoulement de l'air autour d'un cylindre.

Deux capteurs de pression différentielle sont montés à l'intérieur du cylindre. Nous avons donc quatre points de mesure, P1 à P4, qui permettent la mesure de la différence de pression entre P1 et P2 d'une part, P3 et P4 d'autre part. Pour obtenir la direction du vent, on compte pendant un intervalle de temps constant combien de fois la pression maximale apparaît à chaque point. Le calcul de la moyenne permet de déterminer la direction.



Figure 6. Le capteur de pression différentielle SDP600/610-125Pa.

volumique de l'air ( $\rho$ ) de  $1,3 \text{ kg/m}^3$ , la formule de l'encadré nous donne une différence de pression de 260 Pa. En théorie, le capteur devrait donc au moins pouvoir mesurer cette différence de pression. Comme les tempêtes ne sont heureusement pas trop fréquentes sur la majeure partie de notre territoire, nous avons choisi le SDP600/610-125Pa de Sensirion [6], qui peut mesurer une différence maximale de 125 Pa. La précision de ce capteur est cependant meilleure. La tension d'alimentation est de 3,3 V, et les données sont transmises via le bus I<sup>2</sup>C, mais il y a un hic : le fabricant a fixé l'adresse du capteur à  $0x40$  (valeur

hexadécimale). Or nous avons besoin de deux capteurs ; nous devrions alors utiliser deux bus I<sup>2</sup>C, ce qui n'est guère pratique. Heureusement pour nous, l'adresse se trouve dans une mémoire EEPROM interne, et le fabricant nous a aimablement expliqué comment la modifier (ce qui annule la garantie...). Nous avons fixé l'adresse d'un des deux capteurs à  $0x21$ . Vous pouvez vous aussi demander au fabricant la procédure à suivre. Il suffit d'exécuter une routine avec un seul des deux capteurs connecté ; celle-ci est contenue dans le programme *weerstation.py* qui est disponible sur la page du projet [7].

Ce capteur ne possède pas les résistances de rappel pour le bus I<sup>2</sup>C, il faut les ajouter.

### Les grandeurs dérivées

Jusqu'à présent, notre station météo n'est en rien différente de ce que l'on peut trouver dans le commerce ; nous ne mesurons que les grandeurs habituelles : température, humidité, pression atmosphérique, luminosité, vitesse et direction du vent. Ce qui rend notre projet intéressant, c'est le calcul de quelques valeurs dérivées moins connues.

### Point de rosée

Le point de rosée est la température à laquelle l'air est saturé d'humidité. C'est un paramètre appréciable en météorologie ; il permet d'estimer la couverture nuageuse, le risque de brouillard, et le risque de formation de glace, ce qui est important pour le trafic aérien. Une valeur élevée du point de rosée indique également un plus grand risque d'orage. Le point de rosée peut être calculé à partir des valeurs de l'humidité relative et de la température de l'air, comme expliqué sur Wikipédia [8].

### Température ressentie ou indice de refroidissement éolien (ou facteur vent)

Il y a température et température : il y a la valeur mesurée avec un thermomètre, et puis il y a celle que nous ressentons. La différence entre ces deux températures peut être significative ; avec un vent fort et une humidité relative élevée, une température de  $-3 \text{ }^\circ\text{C}$  peut être ressentie comme  $-10 \text{ }^\circ\text{C}$ , voire encore moins !

L'indice de refroidissement éolien, aussi appelé facteur vent, est calculé à partir des valeurs de l'humidité relative et

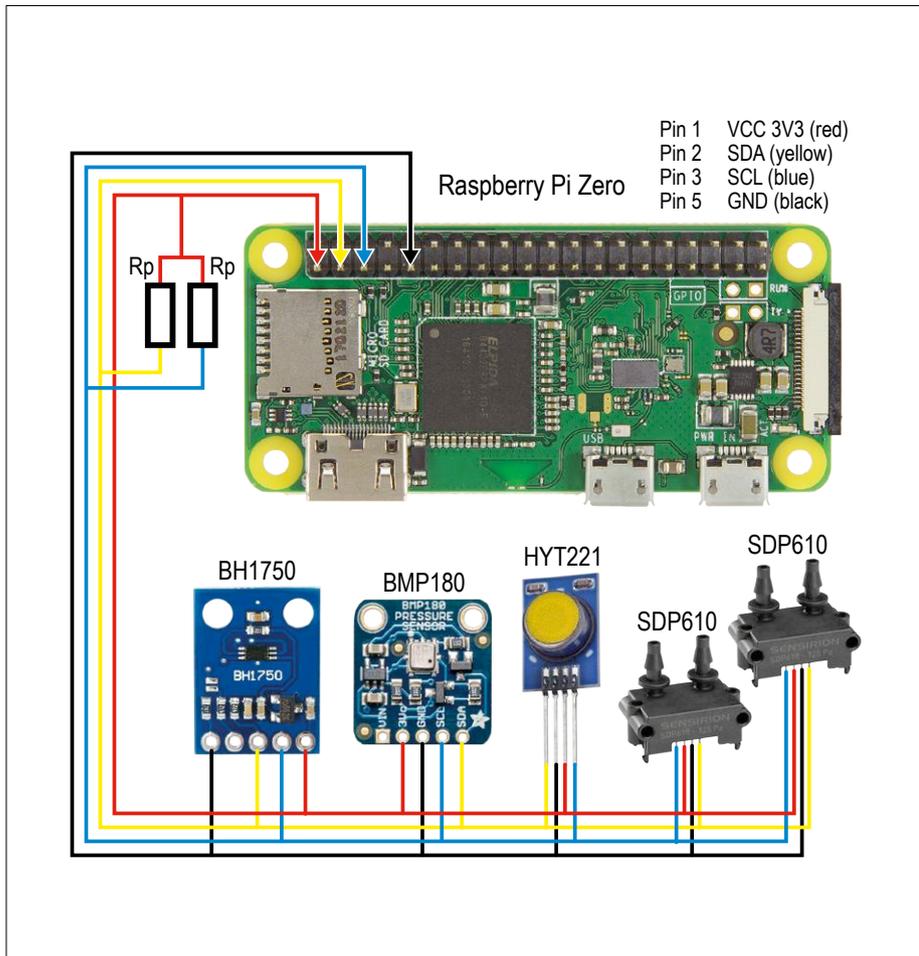


Figure 7. Connexion des capteurs au RPi Zero W via un bus à quatre fils.

Tableau 2 : composants I <sup>2</sup> C et leur adresse		
composant	adresse (hexa)	fonction
BMP180	0x77	capteur de pression
HYT221	0x28	capteur de température et d'humidité
SDP01	0x40	capteur de pression différentielle 1
SDP02	0x21	capteur de pression différentielle 2
BH1750	0x23	capteur d'éclairage lumineux

de la vitesse du vent. Il existe plusieurs modèles pour ce calcul, nous avons choisi celui du service météo australien [9].

### Hauteur des nuages

La hauteur des nuages est aussi une donnée importante : plus ils sont bas, plus le risque de précipitations est élevé ; les agriculteurs l'ont d'ailleurs toujours su... Météo France explique comment calculer cette hauteur à partir des valeurs de la température de l'air et du point de rosée [10].

### L'électronique

Si nous considérons le RPi et les capteurs comme des boîtes noires, il n'y a pas grand-chose à dire sur l'électronique. La **figure 7** reprend schématiquement l'ensemble du montage : un RPi Zero W, les capteurs, et un bus à quatre fils (deux pour l'I<sup>2</sup>C et deux pour l'alimentation). Nous n'avons pas repris l'alimentation du RPi sur ce schéma, on peut utiliser un chargeur/adaptateur USB relié au connecteur d'alimentation de la carte (en bas à droite sur la photo de la figure 1). Les deux résistances  $R_p$  sont les résistances de rappel vers le haut des capteurs de pression différentielle. La **figure 8** montre le prototype de l'auteur, installé dans un petit boîtier en plastique. Le câble d'alimentation passe par un serre-câble à étrier. Il en va de même pour le câble UTP à quatre fils de l'anémomètre, qui sur ce prototype a une longueur de 5 m ; cela ne pose pas de problème pour la transmission des données, et permet l'installation des capteurs sur le toit.

Le **tableau 2** reprend les adresses I<sup>2</sup>C des différents capteurs.

### Le montage

Pour le montage de l'anémomètre, il faut faire preuve d'un peu d'ingéniosité. L'auteur a utilisé un tube en PVC de 10 cm de diamètre, dans lequel il a percé quatre trous, aux quatre points cardinaux. Des morceaux de tuyau flexible, par ex. celui utilisé pour une pompe d'aquarium, sont collés au tube en PVC et connectés aux ouvertures des capteurs de pression différentielle. La **figure 9** montre le résultat final. Les dessus et dessous du tube doivent bien entendu être étanches, afin de protéger l'électronique ; les extrémités des morceaux de tuyau doivent aussi être protégées des insectes, qui pourraient s'y installer, par exemple avec de la gaze. En tout cas, les bricoleurs peuvent s'en donner à cœur joie.

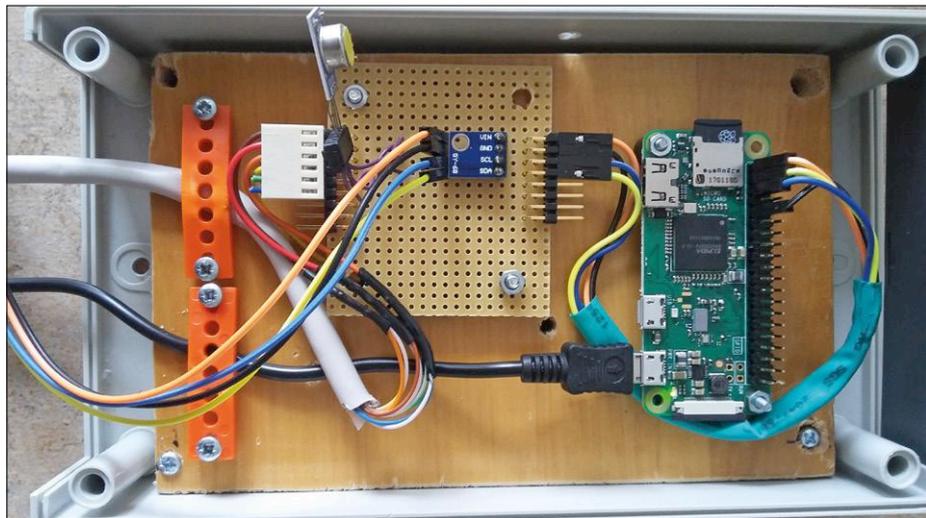


Figure 8. Le prototype de l'auteur.



Figure 9. Le capteur éolien dans son élément, au-dessus du toit.

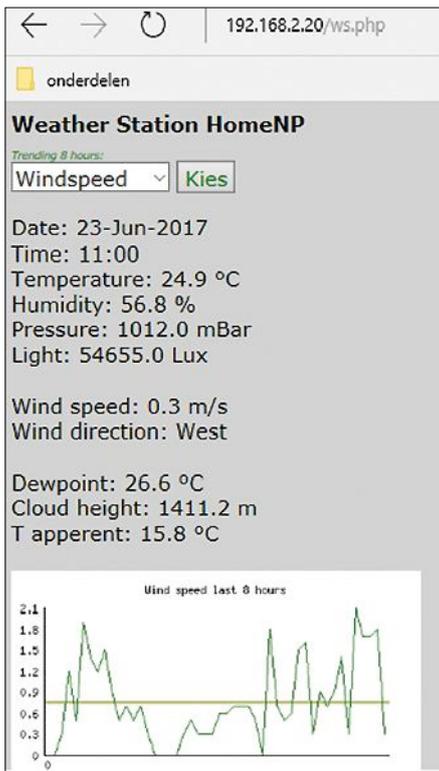


Figure 10. Page web avec les résultats des mesures.

### Le micrologiciel

Le micrologiciel de la station météo comporte deux parties. Tout d'abord le programme *weerstation.py*. Celui-ci est écrit en Python, ce qui est pratique, puisque *Python 2.6* est partie intégrante de la distribution Linux *Jessie* du RPi Zero. Le programme collecte les données des capteurs toutes les dix minutes, exécute les calculs nécessaires, et entrepose les résultats dans une base de données MySQL.

La deuxième partie du micrologiciel est un script combiné HTML et PHP. Celui-ci tourne sur le serveur Apache du RPi Zero, lit les données de la base de données et les présente sur une page web (exemple en **figure 10**).

Cela n'aurait aucun sens de reproduire le code du micrologiciel ici, vous pouvez le télécharger en quelques clics de souris. Le programme est convenablement structuré et se passe de commentaires.



**DANS L'É-CHOPPE**

→ 18328  
kit de démarrage Raspberry Pi  
Zero W

Nous ne citerons que les bibliothèques utilisées par *weerstation.py* :

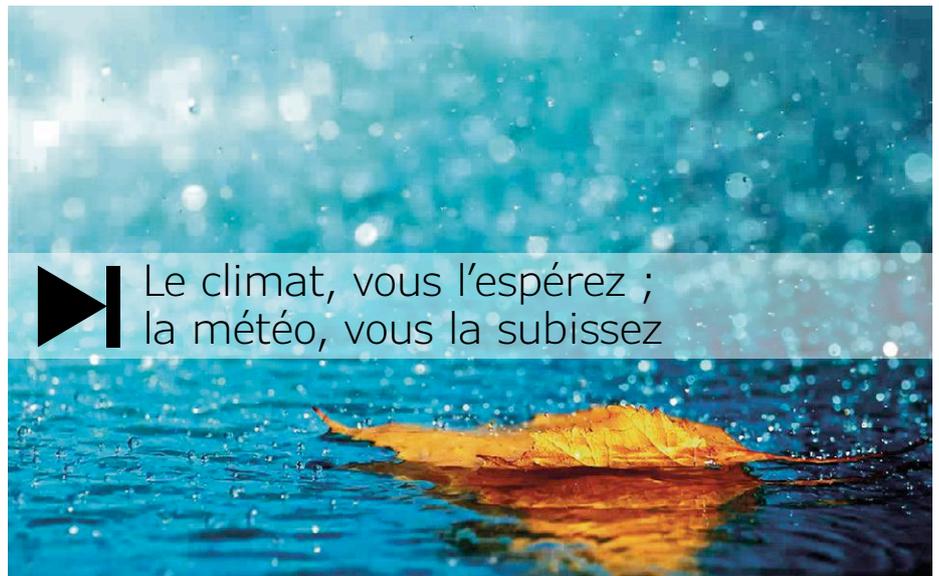
```
import MySQLdb as mdb
import sys
import time
import os
import smbus
from ctypes import c_short
import math
```

La page web *ws.php* (voir dossier de téléchargement) est publiée par le serveur Apache installé sur le RPi Zero. Chez l'auteur, le RPi Zero est relié au réseau local de la maison ; ainsi les données de la station météo sont consultables sur tous les appareils connectés au réseau. La page peut aussi être publiée sur internet via le routeur.

### Conclusion

Ce projet ne sera pas proposé sous forme d'un kit prêt à l'emploi, c'est plutôt une source d'inspiration pour les lecteurs qui souhaitent réaliser un projet semblable. Nous serions heureux de recevoir vos remarques et suggestions. N'hésitez pas à nous présenter vos réalisations de station météo. ◀

(160566 - version française : Jean-Louis Mehren)



▶ Le climat, vous l'espérez ;  
la météo, vous la subissez

### Liens

- [1] [www.elektormagazine.fr/160451](http://www.elektormagazine.fr/160451)
- [2] [learn.sparkfun.com/tutorials/bmp180-barometric-pressure-sensor-hookup](http://learn.sparkfun.com/tutorials/bmp180-barometric-pressure-sensor-hookup)
- [3] [www.elektor.fr/bme280-mouser-intel-i2c-version-160109-91](http://www.elektor.fr/bme280-mouser-intel-i2c-version-160109-91)
- [4] [www.farnell.com/datasheets/1643979.pdf](http://www.farnell.com/datasheets/1643979.pdf)
- [5] [domotix.com/esp8266-wifi-lichtintensiteit-sensor-bh1750-gy-302-nodemcu](http://domotix.com/esp8266-wifi-lichtintensiteit-sensor-bh1750-gy-302-nodemcu)
- [6] [www.sensirion.com/en/environmental-sensors/humidity-sensors/digital-differential-pressure-sensors-without-zero-point-drift](http://www.sensirion.com/en/environmental-sensors/humidity-sensors/digital-differential-pressure-sensors-without-zero-point-drift)
- [7] [www.elektormagazine.fr/160566](http://www.elektormagazine.fr/160566)
- [8] [fr.wikipedia.org/wiki/Point\\_de\\_rosée](http://fr.wikipedia.org/wiki/Point_de_rosée)
- [9] [www.bom.gov.au/info/thermal\\_stress/?cid=003bl08](http://www.bom.gov.au/info/thermal_stress/?cid=003bl08)
- [10] [www.meteofrance.fr](http://www.meteofrance.fr)
- [11] [fr.wikipedia.org/wiki/Tube\\_de\\_Pitot](http://fr.wikipedia.org/wiki/Tube_de_Pitot)
- [12] [www.elektor.fr/raspberry-pi-zero-w-starter-kit](http://www.elektor.fr/raspberry-pi-zero-w-starter-kit)



# vol tous azimuts

piloté par **Clemens Valens** (labo d'Elektor)  
version française : **Hervé Moreau**



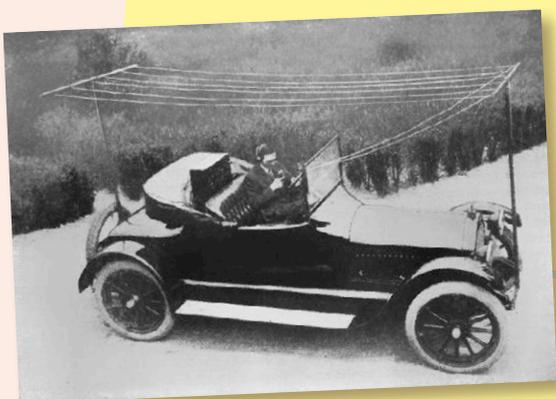
En 1997, Opel (filiale allemande de General Motors) présente l'Opel Corsa Moon, une voiture concept basée sur la populaire Corsa B. Ces antennes étaient-elles intelligentes, ou aussi bêtes que des tapettes à mouches ? (Photo : Wikimedia, Lexicar)

## Des antennes intelligentes dressées contre les cybermenaces

Si j'ai pu rouler en Opel Corsa B durant des années, c'est que l'électronique la plus élaborée de ma guimbarde était probablement celle de ses ampoules (je ne compte pas l'autoradio, il était en option). Que de changements depuis ! Les voitures d'aujourd'hui reposent tellement sur l'électronique que le moindre fil desserré les immobilise. Avec toute cette électronique à ras le capot, ce qui était autrefois un problème inconnu est devenu un péril, tant et si bien que l'on commence à lire des annonces comme « STMicroelectronics mène la course à la protection des véhicules connectés contre les cybermenaces. » Pour contrer lesdites menaces, ST a créé la famille des processeurs STA1385 Telemaco3P « de télématique et connectabilité ». Ces puces à ARM Cortex-A7 double cœur intègrent un module de sécurité matériel (HSM) à ARM Cortex-M3 chargé de vérifier l'authenticité des messages reçus et celle des tentatives de connexion extérieure. Ces nouvelles puces très robustes opèrent jusqu'à 105 °C, donc en des endroits qui pourraient s'avérer de la braise pour elles, p. ex. dans une antenne intelligente placée à fleur de toit d'une voiture. Une antenne *intelligente* ? Allons bon, voilà que même un fil peut devenir « intelligent » et capable d'exécuter un système d'exploitation compatible POSIX au moyen de trois (!) processeurs à 32 bits. Autrefois simple fil de cuivre, l'antenne a désormais des interfaces CAN FD, Gigabit Ethernet et Secure Digital IO à 100 Mbits/s pour créer sa propre interférence. Nous sommes tous avides de progrès et de technologie, mais bon sang, pourquoi compliquer autant les choses ?



Peut-on rester *smart* (intelligente) coiffée d'une antenne stupide ?



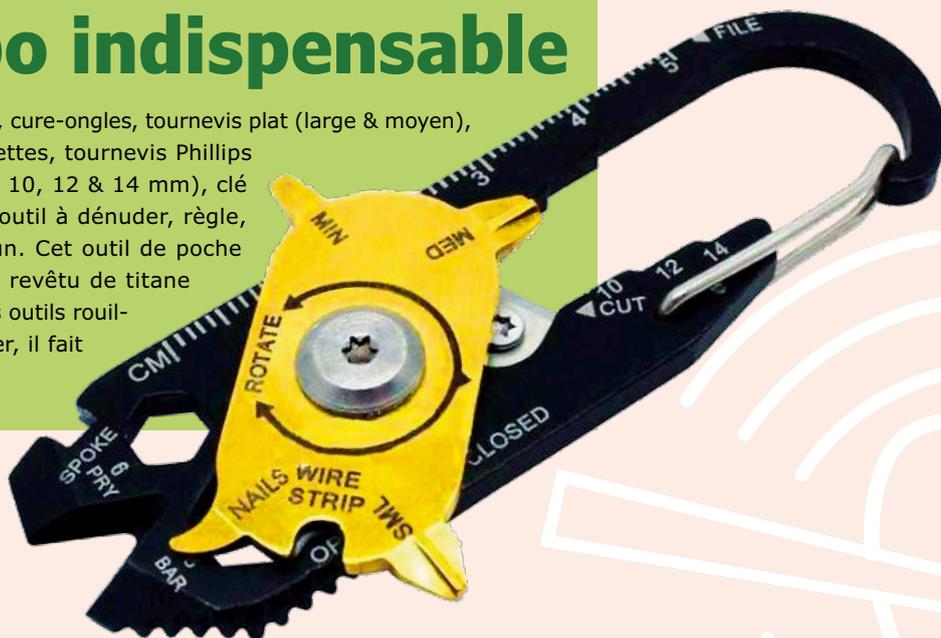
Cette voiture connectée à antenne en L renversé n'avait pas vraiment l'air intelligente, même en 1919. (Photo : Wikimedia, Pacific and Atlantic)

Les microprocesseurs sécurisés Telemaco3P de ST ont pour mission de protéger nos voitures connectées. (Photo : STMicroelectronics)



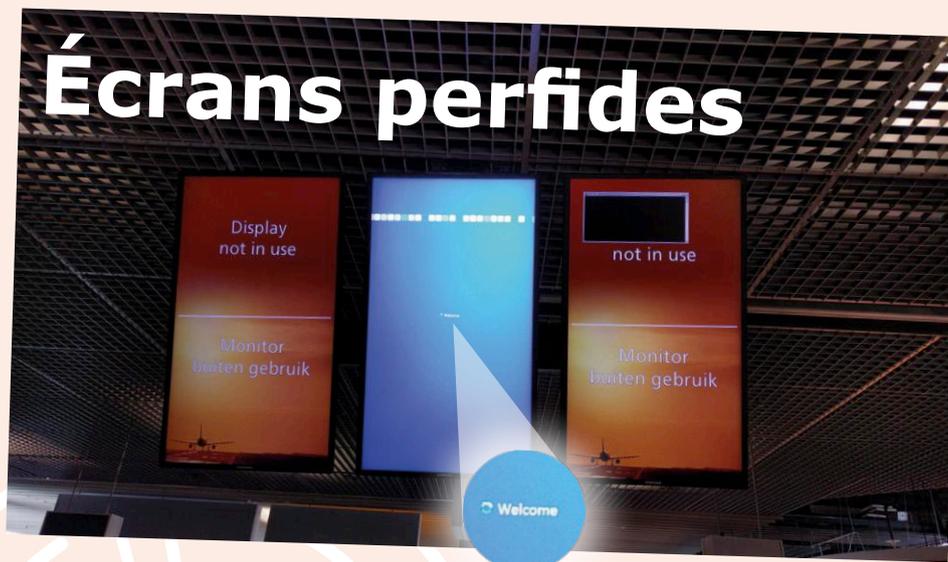
# Outil de labo indispensable

Fixr, de True Utility, est un décapsuleur, cure-ongles, tournevis plat (large & moyen), tournevis plat pour montures de lunettes, tournevis Phillips (moyen & petit), clé à molette (6, 8, 10, 12 & 14 mm), clé à rayons pour vélo, lame de coupe, outil à dénuder, règle, ouvre-boîte, levier et lime tout-en-un. Cet outil de poche de 80 x 25 mm en acier inoxydable revêtu de titane remplacera avantageusement tous vos outils rouillés et encombrants. Oh, j'allais oublier, il fait aussi porte-clés.

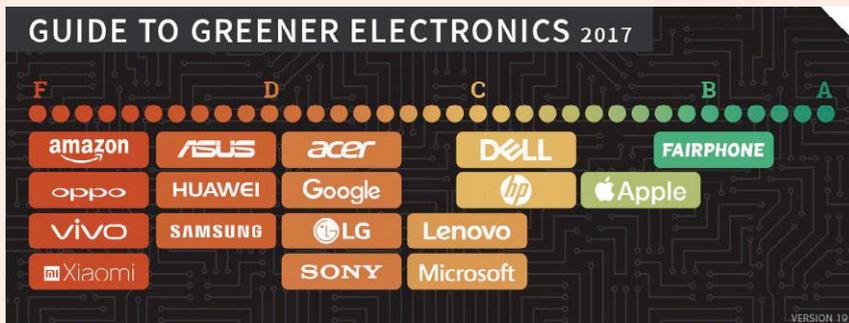


**Les voitures électriques Polestar de Volvo ne seront vendues qu'en ligne et sous condition d'abonnement de deux ou trois ans.**

## Écrans perfides



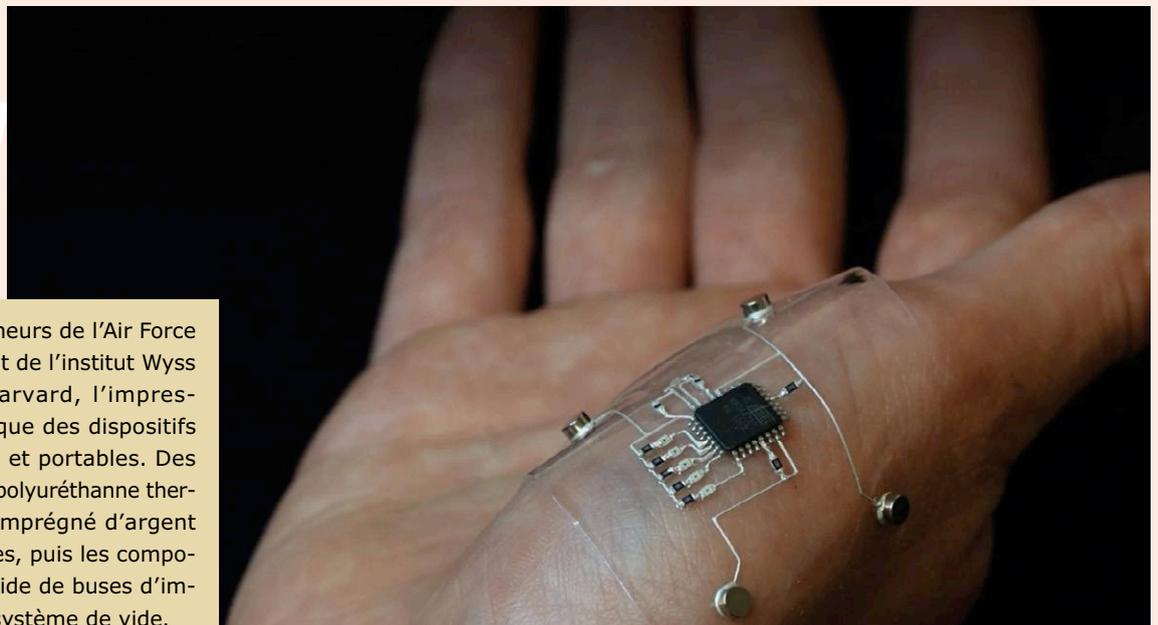
Pour déstresser les voyageurs cherchant désespérément une information lorsque les écrans n'affichent rien, les concepteurs du système d'annonce de vol de l'aéroport de Schiphol (Pays-Bas) ont trouvé une drôle de solution. Je mets concepteurs au pluriel, car il est peu probable qu'une seule personne soit responsable du stress potentiel de millions de voyageurs. Une équipe, donc, de concepteurs et d'experts en satisfaction de la clientèle s'est creusé les méninges afin de trouver le message à délivrer, tandis que d'opéreaux consultants ont cherché dans de gigantesques banques d'images celle qui serait la mieux adaptée à un écran non utilisé. Mais lequel ne l'est pas ? Les trois, ou juste celui du milieu ? Dans ce cas, a-t-on besoin de deux écrans pour annoncer qu'un troisième n'est pas utilisé ? Des flèches pointant vers cet écran auraient-elles rendu le message plus clair ? La solution retenue (photo) a été d'afficher *Welcome* sur l'écran du milieu, qui est donc utilisé, pour signifier que les deux autres ne le sont pas. Pourtant ils affichent bien un message informatif. Je suis resté si longtemps devant ce paradoxe que j'en ai presque raté mon vol. Foutus écrans inutiles, ils feraient aussi bien de les éteindre.



**Le guide Greenpeace de l'électronique verte, version 19**

Votre ordiphone, tablette ou ordinateur portable est-il écoresponsable ? Vert c'est bien, rouge c'est bouh.

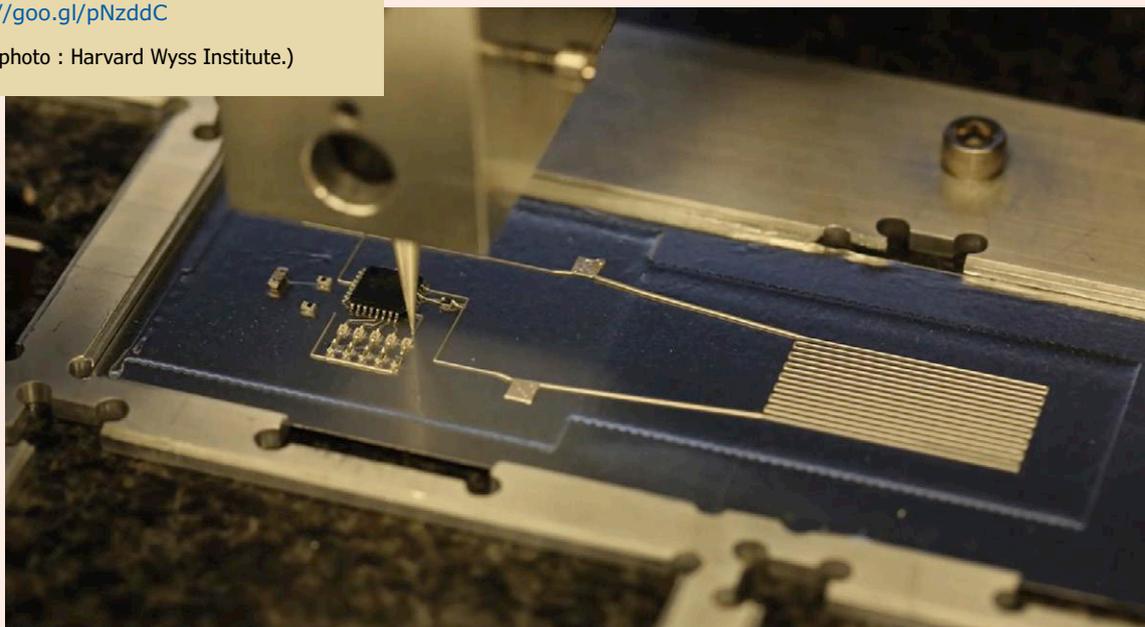
Le rapport complet est sur <http://greenpeace.org/greenerguide>.



Conçue par des chercheurs de l'Air Force Research Laboratory et de l'institut Wyss de l'université de Harvard, l'impression 3D hybride fabrique des dispositifs électroniques souples et portables. Des pistes conductrices en polyuréthane thermoplastique flexible imprégné d'argent sont d'abord imprimées, puis les composants sont placés à l'aide de buses d'impression reliées à un système de vide.

<https://goo.gl/pNzddC>

(Crédit photo : Harvard Wyss Institute.)

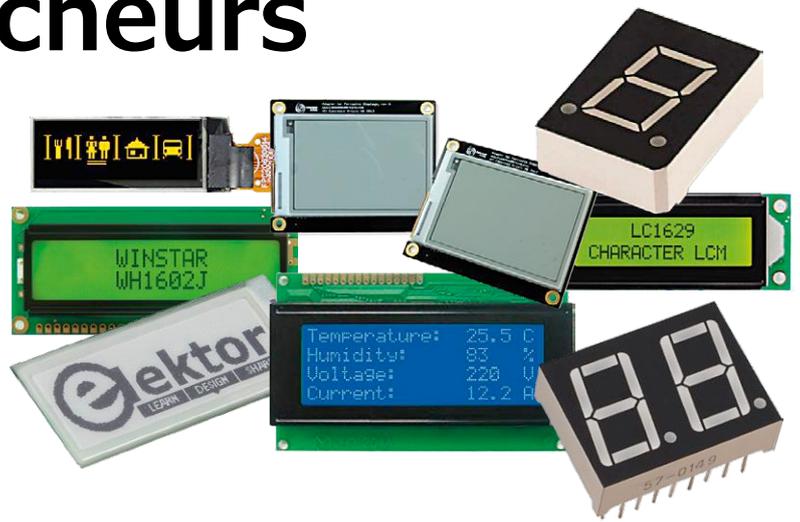


**Vous voulez participer ? Envoyez vos commentaires, suggestions, trucs et astuces à [labs@elektor.com](mailto:labs@elektor.com)**



# (presque) tout ce que vous avez toujours voulu savoir sur... les afficheurs

Pour de nombreux projets électroniques, les afficheurs restent indispensables. Autrefois, on se servait d'afficheurs à sept segments pour signaler des états, aujourd'hui, il existe toute sorte d'afficheurs pour présenter les données d'un microcontrôleur. La question est : qu'est-ce qui les distingue ?



Réponses de **Thomas Scherer**

**Q** *Quel est l'afficheur qui convient le mieux ?*

**R** En électronique, comme souvent dans la vie, la réponse à cette question dépend des critères imposés. L'afficheur qui convient le mieux est celui qui permet, en premier lieu, de répondre précisément aux besoins du projet. À partir de cela, il convient de se poser toute une série de sous-questions pour pouvoir prendre une décision éclairée :

- Qu'est-ce qu'on veut afficher ? Du texte ou des graphiques ?
- Quelle doit être la taille de l'afficheur ? La gamme va du mini-afficheur (< 2,54 cm) à l'écran d'ordinateur.
- Combien de couleurs doit-on afficher ? Le monochrome suffit-il ? Ou bien les couleurs sont-elles indispensables ?
- De quelle interface a-t-on besoin ? Une interface parallèle à 4/8 bits ou plutôt un protocole série ?
- Quelle est la limite à ne pas dépasser quant à la consommation ? L'afficheur doit-il fonctionner sur accu ou sur secteur ?

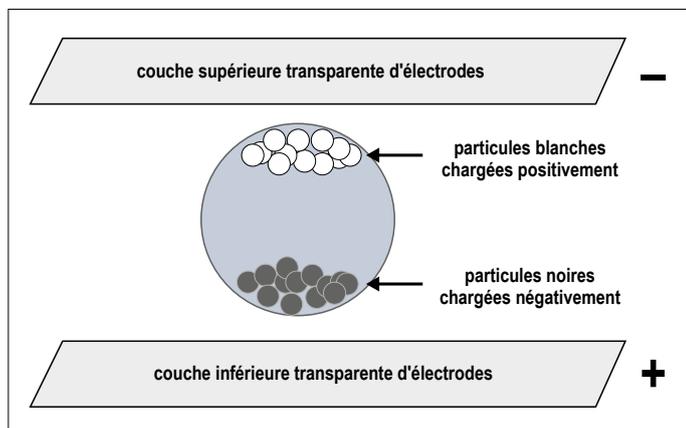


Figure 1. Structure d'un pixel de papier électronique.

- Quelle est la qualité d'affichage recherchée ? Cet aspect sera déterminant dans le choix de la technique d'affichage.
- Quel doit être son coût ? Le prix dépend de la technique d'affichage retenue et de la taille de l'afficheur.

Ces sept critères jouent un rôle au moment de choisir le type d'afficheur utilisé pour un projet concret.

**Q** *Quels sont les avantages et inconvénients du papier électronique ?*

**R** La notoriété du papier électronique résulte de l'apparition de la liseuse Kindle de la société Amazon il y a une dizaine d'années. Dotée d'un écran de type papier électronique, cette liseuse assure des heures de lecture grâce à sa gestion de l'énergie. Et c'est bien là l'avantage le plus important de ce type d'écran, sa faible consommation d'énergie. Cet avantage résulte principalement de la structure du pixel (**fig. 1**). Lorsqu'on applique un champ électrique positif à la surface de la feuille, les particules noires migrent vers la surface et deviennent visibles, tandis que les particules blanches se déplacent lorsqu'un champ négatif est appliqué. Les particules n'ont besoin d'aucune énergie pour rester dans leur position respective.

Le papier électronique est passif, il n'émet pas de lumière. Le contraste et la résolution ne sont pas aussi forts qu'avec un autre type d'afficheur et le temps de réponse est beaucoup plus lent. Le bât blesse lorsque l'image doit bouger. Certes, les couleurs sont possibles avec le papier électronique, mais c'est plutôt chose rare. En revanche, ce type d'afficheur est idéal pour l'alimentation par accu.

Il existe également des écrans plus petits que celui de la liseuse Kindle qui travaillent selon ce principe. Ces écrans nécessitent des puces de contrôleur spéciales papier électronique. Pour ceux qui s'intéressent à cette technique, il existe des kits de développement « e-paper » dont le prix varie entre 20 et 50 €. De plus, Elektor a consacré tout un article sur cette technologie d'affichage dans son numéro d'octobre 2016 [1].

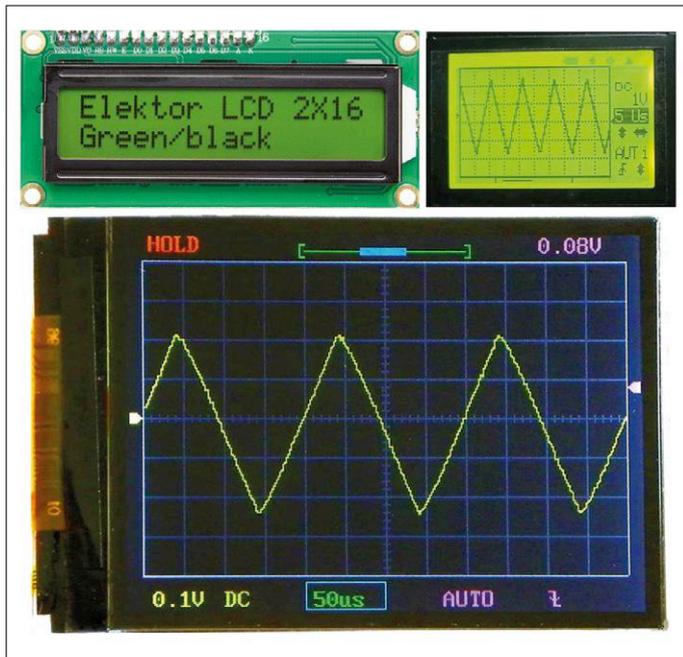


Figure 2. Afficheur LCD alphanumérique, monochrome et polychrome.



Figure 3. Afficheur OLED classique, 2,43 cm, affichage en blanc.

**Q** Quels sont les avantages et les inconvénients des afficheurs LCD ?

**R** Les afficheurs à cristaux liquides sont utilisés dans les projets à microcontrôleur depuis des décennies. C'est la raison pour laquelle, il existe toute une variété de technologies et de domaines d'application (fig. 2). Elektor a consacré beaucoup d'articles sur l'affichage LCD [2]. Chaque afficheur LCD utilise son propre contrôleur. Les modèles les plus simples sont ceux qui sont dotés de deux ou quatre lignes à raison de 12 à 20 caractères par ligne. Ils ne coûtent que quelques euros seulement et en général ils sont pilotés avec un bus parallèle de 4 ou 8 bits de large. Les inconvénients sont entre autres que le pilotage occupe beaucoup de broches du  $\mu$ C, qu'il est impossible de présenter des graphiques, que l'angle de vue et que le contraste sont plus faibles qu'avec d'autres afficheurs. En revanche, presque tous les environnements de développement proposent des bibliothèques qui conviennent aux afficheurs LCD.

Bien entendu, il existe une quantité énorme de LCD qui permettent d'afficher des graphiques monochromes ou en couleurs. En règle générale, ces afficheurs sont pilotés par un bus série, p. ex. via I<sup>2</sup>C ou SPI, ce qui facilite leur mise en œuvre électronique. Vu la multitude de modèles, il est conseillé de vérifier que l'EDI de votre  $\mu$ C préféré offre un soutien logiciel pour le contrôleur du LCD que vous envisagez d'acheter, à moins que vous vouliez réinventer la roue. Les écrans tactiles sont constitués d'un afficheur LCD doublé d'une dalle tactile, à quelques exceptions près.

L'inconvénient majeur de pratiquement tous les afficheurs LCD est leur rétroéclairage (relativement) énergivore.

**Q** Quels sont les avantages et les inconvénients des afficheurs à diodes électroluminescentes organiques (OLED) ?

**R** Les afficheurs OLED pour les projets à microcontrôleur font l'objet d'un article détaillé dans ce numéro d'Elek-

tor. Nous nous limiterons ici aux avantages et inconvénients liés à ce type d'afficheur.

Pour : haute résolution, riche en contrastes, faible consommation d'énergie, système monopuce (ou presque) et coût modéré (en cas de petites diagonales).

Contre : pas très stable à long terme, coûts plus élevés dès qu'on passe à des diagonales plus grandes. Pour l'instant, il n'y a pratiquement pas d'afficheurs OLED polychromes abordables.

**Q** Et qu'en est-il des autres types d'afficheurs ?

**R** Il existe effectivement des afficheurs à LED et des afficheurs à sept segments à LED, mais nous ne voulons pas trop nous attarder sur eux. Certes, cette technique d'affichage anachronique est parfaitement lisible, en revanche les afficheurs à sept segments sont difficiles à piloter et leur affichage se limite aux chiffres. En revanche, si vous aimez le style « steampunk » ou rétrofuturisme, pourquoi ne pas utiliser un tube Nixie ? Elektor regorge d'articles et de projets sur le sujet. Pour finir, parlons des afficheurs graphiques à  $\mu$ LED (micro-LED). En principe, cette technique combine les avantages des afficheurs OLED avec une durée de vie plus longue et des temps de commutation très courts. Ils permettraient d'avoir des fréquences d'image de l'ordre de la milliseconde. Malheureusement, ces afficheurs ne sont pas encore commercialisés. Attendons de voir ce que l'avenir nous réserve... ◀

(160563 - version française : Pascal Duchesnes)

## Liens

[1] Article Elektor sur le papier électronique : [www.elektormagazine.fr/160196](http://www.elektormagazine.fr/160196)

[2] Articles Elektor sur les afficheurs LCD : <https://goo.gl/H96YEK>

# bricolons avec le bus CAN

## aux commandes des 4000 lampes d'un immeuble

Grâce à la prolifération d'Arduino et à la vague du matériel et du code source libres, utiliser un bus CAN pour expérimenter est désormais un jeu d'enfant. Il est possible d'acheter en ligne du matériel d'interfaçage avec le bus CAN, compatible avec Arduino et bon marché, et de télécharger gratuitement les bibliothèques logicielles de prise en charge. Il ne reste plus qu'à bâtir l'application. Nous vous montrons ici combien c'est devenu facile.

**Clemens Valens** (labo Elektor)  
Sur une idée de **Bera Somnath** (Inde)

Le bus CAN — officiellement *Controller Area Network* — permet d'échanger des messages courts entre les dispositifs connectés. À l'origine c'est un bus de communication fiable pour l'automobile qui n'utilise qu'un minimum de câbles (paire torsadée) ; aujourd'hui il a trouvé sa voie dans toutes sortes d'autres applications.

CAN est un réseau basé sur les messages, pas sur une connexion maître-esclave, pair-à-pair (P2P) ou autre. De ce fait les dispositifs connectés — les nœuds — envoient des messages quand bon leur semble, sans se soucier des récepteurs. La détection des collisions et erreurs ainsi

qu'un système de priorisation garantissent que tout fonctionne tant que l'état du trafic sur le réseau demeure normal.

Les nœuds n'ont ni adresses ni identifiants (ID), mais les messages sont typés. Les nœuds n'écoutent ou n'envoient que des messages d'un type donné ; ils ne peuvent pas se parler directement sans ajouter un niveau de plus au protocole. Toutefois, comme un nœud peut demander un type de message donné, et que la spécification de CAN exige que les nœuds utilisent des ID de message uniques, une communication pair-à-pair élémentaire est possible (voir l'encart sur les trames de requête). Le CAN standard autorise jusqu'à 2048 types de messages (ID de message sur 11 bits), le CAN étendu plus de 500 millions ( $2^{29}$  exactement, avec des ID de message sur 29 bits). Pour les

### Caractéristiques

- Pas de soudure
- Capacité de plus de 4000 nœuds
- Logiciel facile à comprendre
- Basé sur des modules bon marché

deux variantes de CAN, la charge utile peut aller jusqu'à huit octets.

Comme sur le bus CAN, un « 0 » est plus fort (dominant) qu'un « 1 » (récessif), plus l'ID du message est bas, plus haute est sa priorité ; ID=0 surpasse tout. L'expérience a montré que pour obtenir un haut niveau d'utilisation du bus, on doit attribuer les ID selon l'importance du message, pas de son contenu.

Ce qui précède est tout ce que vous devez connaître pour les expériences décrites dans la suite de cet article. Pour

plus d'informations sur le mode de transit des bits sur les fils et autres détails techniques, nous recommandons vivement l'internet.

### Matériel nécessaire

Nos expériences reposeront sur l'AVR Playground, mais au début n'importe quelle carte compatible Arduino Uno fera l'affaire. Comme le « N » de CAN signifie *Network*, nous aurons besoin d'au moins deux nœuds, concrètement deux cartes AVR Playground. Il faut une interface CAN pour les deux nœuds. Plusieurs solutions existent ; nous en avons choisi une basée sur un contrôleur CAN MCP2515 de Microchip en raison de sa notoriété. Comme l'AVR Playground a un emplacement mikroBUS, nous avons opté pour les cartes CAN SPI click de MikroElektronika (en version 5 V, la version 3,3 V devrait aussi fonctionner) et avons laissé libres pour d'autres usages les connecteurs pour les *shields* Arduino. Vous trouverez en ligne, pour moins de 5 €, des modules presque identiques, mais de forme différente et qui nécessitent des fils de liaison pour leur connexion.

Pour nos expériences plus sérieuses (voir ci-dessous), nous recommandons un de ces *shields* relais à quatre canaux bon marché et très répandus sur l'internet. Une fois les modules CAN connectés aux cartes compatibles Arduino Uno, il est temps de câbler le réseau. Normalement il faut un câble à paire torsadée, mais pour une faible longueur de simples fils suffisent. On peut trouver des paires torsadées dans des câbles Ethernet par exemple (fig. 1).

On doit mettre des terminaisons de bus à chaque extrémité du câble ; les modules CAN ont un cavalier pour cela. Les nœuds intermédiaires *ne* doivent pas terminer le câble.

### Logiciel nécessaire

D'abord et avant tout vous aurez besoin de votre EDI Arduino.

Pour accéder facilement aux périphériques de l'AVR Playground, installez son paquetage de cartes dans l'EDI Arduino. Reportez-vous à [2] pour les détails. Après l'installation, sélectionnez l'AVR Playground comme carte Arduino et choisissez le port COM correspondant. Installez aussi la bibliothèque du bus CAN pour Arduino — *mcp\_can* — aimablement mise à notre disposition par SeedStudio [3].

C'est tout, on y va !

### Premiers pas

Nous commençons simplement par vérifier que l'interface CAN peut être initialisée correctement. Les instructions se rapportent au menu principal de l'EDI Arduino.

1. Si besoin, ouvrez un nouveau croquis : Fichier → Nouveau
2. Incluez la bibliothèque CAN : Croquis → Inclure une bibliothèque → *mcp\_can*
3. Ajoutez ce qui suit après les directives `include` mais avant la fonction `setup` :

```
const int SPI_CS_PIN = 10;
MCP_CAN CAN(SPI_CS_PIN);
```

Ceci suppose que la broche CS de votre interface CAN soit connectée à la broche 10 de l'Arduino.

4. Ajouter ce qui suit à la fonction `setup` (entre les {}):

```
Serial.begin(115200);
while (CAN_OK!=CAN.
begin(CAN_200KBPS))
// 125 Kbps avec quartz 10 MHz
{
  Serial.print("*");
  delay(100);
}
Serial.println();
Serial.print("CAN init OK.");
```

5. Compilez le croquis et chargez-le dans l'AVR Playground (on vous demandera de le sauvegarder d'abord) : Croquis → Téléverser
6. Ouvrez le moniteur série : Outils → Moniteur série
7. Sélectionnez un débit de 115200 bauds (coin inférieur droit).

Si le texte « CAN init OK. » n'apparaît

### INFOS SUR LE PROJET

CAN AVR Playground  
Arduino réseau  
prototypage rapide

débutant → connaisseur expert

Env. 4 h

AVR Playground, carte compatible Arduino Uno, module contrôleur CAN MCP2515, ordinateur

Env. 100 €  
(en fonction du matériel choisi)

pas, vérifiez les connexions de votre interface CAN et examinez votre code avec soin.

Si vous pouvez connecter les deux nœuds en même temps à votre PC, répétez les étapes ci-dessus dans une seconde instance de l'EDI Arduino connectée au second nœud. Si vous ne pouvez connecter qu'un nœud à la fois, répétez seulement les étapes 5 et 6. Les deux nœuds devraient s'initialiser correctement (ouf).

### Envoyer et recevoir des messages

Pour les expériences suivantes, nous aurons besoin d'au moins deux nœuds connectés au bus CAN. N'oubliez pas de terminer le bus.

Le **listage 1** montre le programme, identique pour tous les nœuds, sauf les valeurs de `my_message_id` et `my_message` qui doivent être uniques pour chaque nœud.

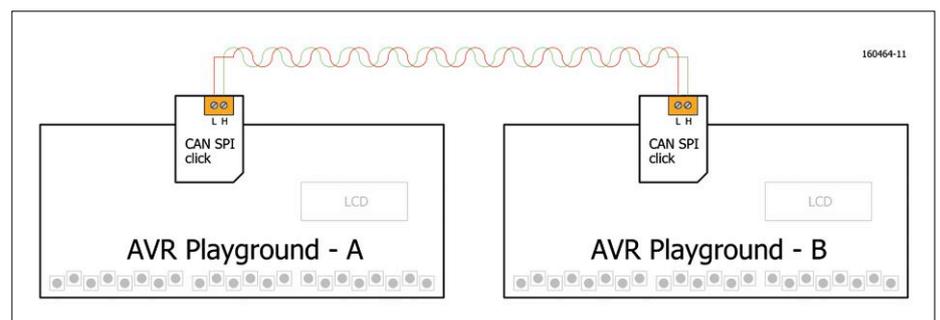


Figure 1. Un CAN élémentaire pour nos premières expériences.

**Listage 1. Envoie et reçoit des messages de type texte courts. Utilisez des valeurs différentes pour my\_message\_id et my\_message pour chaque noeud.**

```
/*
 * Elektor article 160464 Bricolons avec le bus CAN
 * Envoi et réception de messages
 */

#include <mcp_can.h>
#include <mcp_can_dfs.h>

const int SPI_CS_PIN = 10;
MCP_CAN CAN(SPI_CS_PIN);
const int my_message_id = 100;
const char *my_message = "Node 1"; // 8 caractères max !
uint8_t can_buffer[MAX_CHAR_IN_MESSAGE];

void setup(void)
{
  Serial.begin(115200);
  while (CAN_OK!=CAN.begin(CAN_200KBPS)) // 125 Kbps avec quartz 10 MHz
  {
    Serial.print("*");
    delay(100);
  }
  Serial.println();
  Serial.print("CAN init OK.");
}

void loop(void)
{
  if (CAN_MSGAVAIL==CAN.checkReceive())
  {
    uint8_t len;
    CAN.readMsgBuf(&len,can_buffer);
    uint8_t message_id = CAN.getCanId();
    Serial.println("---");
    Serial.print("Message ID: ");
    Serial.print(message_id);
    Serial.print(", received ");
    Serial.print((int)len);
    Serial.println(" bytes.");
    for (int i=0; i<len; i++)
    {
      Serial.print((char)can_buffer[i]);
    }
    Serial.println();

    CAN.sendMsgBuf(my_message_id,0,strlen(my_message),(uint8_t*)
my_message);
    delay(1000);
  }
}
```

La fonction `setup` est la même que dans l'exemple précédent, mais `loop` a été remplacée. Elle commence par vérifier si un message a été reçu. Si oui, il est lu et son contenu envoyé sur le port série, accompagné de l'ID du message. Rappelez-vous qu'un message doit être lu avant de pouvoir vérifier son ID avec `getCanId`. On peut utiliser la fonction `readMsgBufID` pour lire en même temps un message et son ID. Qu'un message soit reçu ou non, la fonction se termine en envoyant le message propre au nœud, puis attend une seconde.

Seules les lignes contenant « CAN. » (notez ce point) ont à voir avec l'interface CAN, toutes les autres sont là pour du support et de la (re)présentation de données.

### Filtrage de message

Dans l'exemple précédent, la fonction `loop` de tous les nœuds recevait tous les messages circulant sur le bus CAN. C'est bien pour le débogage ou quand l'hôte n'a rien d'autre à faire. En revanche, quand il est sensé réaliser aussi d'autres tâches, devoir inspecter chaque message pour voir s'il contient de l'information utile n'est pas efficace. Il est possible d'améliorer nettement cette situation en exploitant les capacités de filtrage du MCP2515.

Le MCP2515 n'est pas qu'une simple interface de bus CAN, mais un contrôleur CAN avec de nombreuses fonctions. Ainsi il propose des filtres et masques qui peuvent être appliqués aux messages entrants. Seuls les messages qui passent ces filtres seront transmis à l'hôte. Les masques et filtres sont appliqués à l'ID du message, et comme le CAN étendu utilise des ID à 29 bits, les masques et filtres font 29 bits de long. Toutefois, le CAN standard travaille avec des ID à 11 bits, ce qui laisserait la majorité (18) des bits de filtres et masques inutilisés. Pour cette raison, lorsqu'on utilise le CAN standard, 16 des bits de filtres et masques restants sont appliqués aux deux premiers octets de la charge utile. Quand un bit du masque est à 0, le bit correspondant dans l'ID du message est toujours accepté, quel que soit le bit de filtre correspondant. Quand un bit du masque est à 1, le bit correspondant de l'ID du message sera accepté seulement s'il a la même valeur que le bit de filtre correspondant (**fig. 2**). Au démarrage, les masques sont effacés et donc tous les messages seront acceptés.

Le MCP2515 utilise trois tampons pour la réception des messages. Le premier, le *Message Assembly Buffer* ou MAB reçoit tous les messages. Quand un message est complet, il est transféré du MAB vers un des tampons de réception RXB0 ou RXB1. RXB0 a un masque et deux filtres (1 & 2) ; RXB1 a un masque et quatre filtres (3 à 6). Les masques et filtres de RXB0 sont appliqués en premier, donnant à ce tampon une priorité supérieure à RXB1. L'idée est que les messages à haute priorité traversent RXB0 tandis que les messages moins importants transitent par RXB1.

La bibliothèque `mcp_can` que nous utilisons n'autorise pas l'utilisateur à choisir le tampon à lire, elle vérifie simplement RXB0 en premier puis RXB1. Ceci implique que les deux masques doivent être réglés pour éviter de recevoir quand même tous les messages. De plus, la bibliothèque ne prend pas en charge le filtrage des bits de données dans le mode CAN standard.

Pour activer le filtrage de message, ajoutez les lignes suivantes à la fin de la fonction `setup` du précédent exemple :

```
// Activer les masques & filtres
  en mode standard (0).
CAN.init_Mask(0,0,0x3ff); // RXB0
CAN.init_Mask(1,0,0x3ff); // RXB1
// Accepter seulement les ID 100
  & 103
CAN.init_Filt(0,0,100); // RXB0
CAN.init_Filt(1,0,100); // RXB0
CAN.init_Filt(2,0,103); // RXB1
CAN.init_Filt(3,0,103); // RXB1
CAN.init_Filt(4,0,103); // RXB1
CAN.init_Filt(5,0,103); // RXB1
```

Ces lignes configurent les masques pour que seuls les filtres déterminent quels messages passent. Ici nous avons dirigé les messages avec ID 100 vers RXB0 et ceux avec ID 103 vers RXB1 (bien que vous ne le verrez pas). Notez que le filtrage cesse dès qu'une correspondance est trouvée, donc seuls les filtres 0 et 2 produiront des correspondances. Pour voir le filtrage fonctionner, vous devez, bien sûr, programmer un nœud pour envoyer ces ID de message et quelques autres. Vous pouvez télécharger depuis [1] un croquis qui vous permet de le faire. Le croquis envoie six messages avec six ID différents et écoute les messages que vous avez spécifiés pour les filtres. Il peut être utilisé par tous

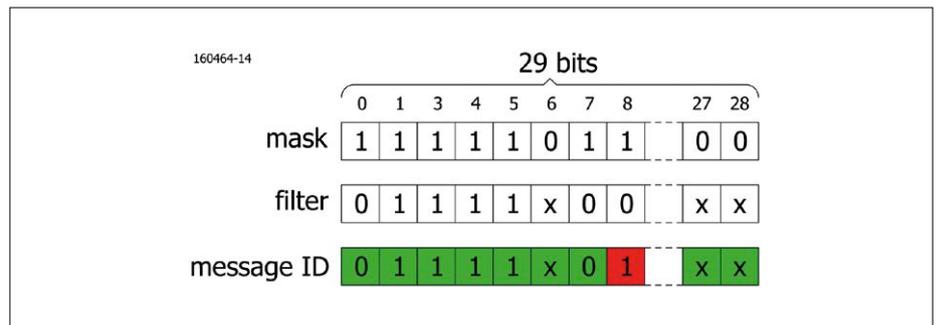


Figure 2. Pour limiter le nombre de messages à lire par l'hôte, le MCP2515 applique d'abord un masque puis jusqu'à six filtres. Si le bit du masque est à 0, peu importe les bits du filtre et du message ; si le bit du masque est à 1, alors le bit du filtre doit être identique à celui du message. Cet ID de message est rejeté à cause du bit 8.

les nœuds sur le bus sous réserve que chaque nœud envoie des messages avec des ID uniques. La variable `tx_id_base` permet d'assurer cela. Une `tx_id_base` de nœud doit être choisie pour que les six ID de message qu'il transmet (`tx_id_base` à `tx_id_base+5`) soit uniques sur le bus.

### Commande d'éclairage

Parvenus jusqu'ici, nous savons presque tout ce qu'il faut savoir pour créer une application intéressante, il reste à ajouter du sens aux messages et à implémenter des fonctions pour faire des choses utiles, comme activer un relais ou lire un bouton-poussoir.

Pourquoi ne pas mettre en place un CAN dans un immeuble pour commander les lampes dans chaque pièce et à tous les étages ? Tous les interrupteurs de l'immeuble deviendront des nœuds CAN et des nœuds contrôleurs peuvent être placés à quelques endroits stratégiques pour allumer ou éteindre ces lampes. Le concept peut, bien sûr, être étendu aux radiateurs, à l'air conditionné...

Nous simulerons le système avec nos cartes AVR Playground, l'une d'entre elles équipée d'un *shield* relais à 4 canaux

(disponible en ligne, env. 10 €) pour commuter les lampes. Les cartes AVR Playground disposent de LCD, buzzers, boutons-poussoirs, LED et potentiomètres utiles pour commander les relais et fournir un retour de ce qui se passe (**fig. 3**).

### Mappage des ID de message

Avant de commencer à programmer, nous devons d'abord réfléchir au plan du réseau et au mappage des ID de message. Prenons un immeuble de dix étages, avec dix pièces par étage et dix nœuds (ou prises) par pièce, 1000 ID seront nécessaires pour adresser chaque nœud individuellement. Le CAN standard dispose de 2048 ID, nous avons de la marge. Il faut des ID de messages uniques pour les nœuds autorisés à envoyer des messages (comme des informations d'état). Sinon, un seul ID de message est suffisant pour commuter toutes les lampes individuellement en stockant simplement son adresse et son état dans la charge utile de 8 octets. Supposons que notre système exige que chaque nœud soit capable d'émettre des messages (d'état). Associer 1000 ID de 11 bits à 10 lampes dans 10 pièces

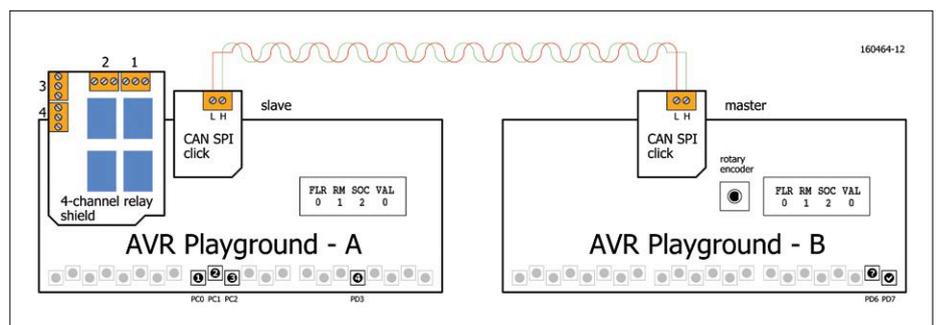


Figure 3. Avec ce système nous pouvons simuler une centrale de commande d'éclairage pour un immeuble à étages.

## À propos du débit binaire et des fréquences de quartz

La bibliothèque *mcp\_can* considère que le MCP2515 est rythmé par un quartz à 16 MHz. Toutefois, les modules trouvés en ligne peuvent comporter d'autres quartz (par ex. 8 MHz) ; les cartes CAN SPI click que nous avons utilisées fonctionnent à partir de 10 MHz. Mixer ces cartes sur un bus sans ajuster le débit binaire de chaque module dans le micrologiciel conduira à des débits binaires incompatibles sur le réseau. Malheureusement, la bibliothèque *mcp\_can* n'offre pas de fonction pour fixer le débit binaire, seules quelques constantes traitées ultérieurement en interne sont accessibles. Mais il y a une bonne nouvelle : sans modifier la bibliothèque, trois débits binaires prédéfinis fonctionnent avec les modules CAN MCP2515 à 8, 10 et 16 MHz (voir le tableau ci-dessous). Utilisez ces constantes à l'appel de la fonction *begin()* de la bibliothèque (voir par ex. listage 1). Si vous tenez à un autre débit binaire, alors vous devrez vous plonger dans la bibliothèque et l'adapter à vos besoins.

	8 MHz	10 MHz	16 MHz
<b>25 Kbps</b>	CAN_50KBPS	CAN_40KBPS	CAN_25KBPS
<b>50 Kbps</b>	CAN_100KBPS	CAN_80KBPS	CAN_50KBPS
<b>125 Kbps</b>	CAN_250KBPS	CAN_200KBPS	CAN_125KBPS

sur 10 étages nécessiterait une table de correspondance, car il n'est pas facile de parvenir à une association intuitive des ID en les numérotant simplement de 1 à 1000. Un tel schéma introduirait aussi des différences de priorité importantes dans un système où tous les nœuds sont supposés équivalents.

Pour simplifier l'installation, il serait préférable que chaque nœud produise lui-même un ID unique basé sur son emplacement. L'installateur aurait à préciser

les numéros d'étage, de pièce et de prise et le nœud ferait le reste. Le CAN étendu avec ses ID à 29 bits peut traiter cela très facilement : réservez simplement des champs de quatre bits pour l'étage, la pièce et la prise et ajoutez un grand décalage. Ceci les répartira tous dans un espace d'adressage restreint (12 bits) avec des priorités équivalentes dans l'énorme réservoir d'ID possibles et il serait facile pour un nœud contrôleur de déterminer où est un nœud. Utilisons

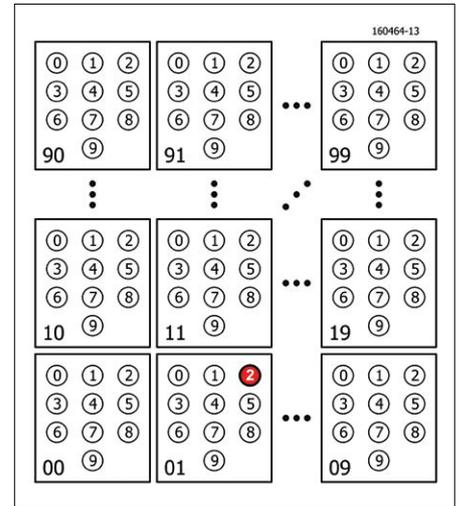


Figure 4. La charge utile <2><1><0><1> du message avec l'ID 0xffff allumera le nœud indiqué.

donc le CAN étendu avec les adresses de nœud formatées ainsi <n3n2n1n0><p3p2p1p0><e3e2e1e0> et ajoutons un décalage de, disons 0x10000. Le nœud (prise) 0 dans la pièce 0 de l'étage 0 aurait l'adresse 0x10000 et le nœud (prise) 9 dans la pièce 9 de l'étage 9 aurait l'adresse 0x10999. (Où serait le nœud 0x10123 ? Exactement : prise 1 dans la pièce 2 de l'étage 3). Ceci laisse de l'espace pour quelque 3000 nœuds supplémentaires...

Pour utiliser la bibliothèque *mcp\_can* en mode CAN étendu, il suffit de mettre l'argument « ext » (des fonctions qui l'utilisent) à 1 (au lieu de 0).

Un ID supplémentaire (arbitrairement choisi à 0xffff) serait nécessaire pour commander les nœuds. Une commande de commutation préciserait alors l'étage, la pièce et la prise dans sa charge utile, avec l'action de commutation comme ON ou OFF, ou, puisque les LED RVB dominent le monde aujourd'hui, une valeur RVB (ou une vitesse de ventilateur, un niveau de chauffage, etc.). Pour prendre en charge toutes ces options, nous pourrions coder l'action sur trois octets et garder deux octets pour de futures extensions. Ici, par souci de simplicité, nous avons codé l'action sur un octet avec OFF=0 et ON=1 (**fig. 4**).

Bien sûr, commander une lampe à distance c'est bien, mais le vrai pouvoir serait de commander des groupes de lampes simultanément. Pour cela, nous utiliserons la valeur 255 (0xff). Quand l'étage 255 est spécifié, tous les nœuds avec les numéros de pièce et de prise

Tableau 1. On peut commander des groupes en utilisant l'adresse de groupe ou « toutes », 255.

Étage	Pièce	Prise	Action
255	2	9	La prise 9 des pièces 2 de tous les étages répondra
5	255	2	La prise 2 de toutes les pièces de l'étage 5 répondra
0	3	255	Toutes les prises de la pièce 3 de l'étage 0 répondront
255	6	255	Toutes les prises de la pièce 6 de tous les étages répondront
1	255	255	Toutes les prises de toutes les pièces de l'étage 1 répondront.
255	255	255	Toutes les prises de l'immeuble répondront

Tableau 2. Configurations des commutateurs DIP du maître CAN AVR Playground (« H » pour Haut, « B » pour Bas, « M » pour Milieu).

Commutateur DIP	Position (gauche à droite)
S24	MMMMMMMM
S33	HBBHHHHMM
S15	HBBBHHH
S25	BHHHBBH
S27	MMMHMMMM

Tableau 3. Configuration des commutateurs DIP de l'esclave CAN (multinœuds) AVR Playground (« H » pour Haut, « B » pour Bas, « M » pour Milieu).

Commutateur DIP	Position (gauche à droite)
S24	MMMMMMMM
S33	HBBHHHHMM
S15	HBBBHHH
S25	BHHHBBH
S27	MMMHMMMM

spécifiés de tous les étages seront commutés. De même, quand la pièce 255 est spécifiée, tous les nœuds avec le numéro de prise spécifié de l'étage en question répondront. Quand l'étage, la pièce et la prise sont tous mis à 255, alors tous les nœuds de l'immeuble écouteront et ils peuvent être commutés (ON ou OFF) tous ensemble. Le **tableau 1** montre quelques exemples pour clarifier le concept.

### Programmation

Nous avons implémenté la description ci-dessus dans trois croquis Arduino. Le premier croquis, le maître, à lancer sur l'AVR Playground (voir **tableau 2** pour la configuration de ses commutateurs DIP), permet à l'utilisateur de sélectionner un étage, une pièce et une prise en tournant le codeur rotatif (changement de valeur) et en le poussant (passer au paramètre suivant, noter le curseur « > »), et de changer l'état du nœud en pressant le bouton PD7. On peut changer l'état d'un nœud parce qu'au préalable on a demandé son état actuel. Ceci est réalisé automatiquement dès que le codeur est immobile pendant un moment. Si un nœud ne répond pas à cette demande au bout d'une seconde, parce qu'il n'existe pas par exemple, l'état affiché est « --- » et on entend un son à basse fréquence (si le buzzer est activé). Presser le bouton PD6 enverra aussi une demande d'état. Presser PD7 ne fera rien pour les nœuds « morts ». Lorsqu'un nœud répond, son état est affiché et une tonalité haut-perchée retentit. Si un groupe d'adresses est sélectionné pour un ou plusieurs paramètres (indiqué par « All »), il faut choisir l'état avec le codeur rotatif et presser PD7 pour envoyer la commande.

### Esclave multimessages

Le croquis maître est complété par un croquis esclave multinœuds, aussi prévu pour l'AVR Playground (voir **tableau 3** pour la configuration de ses commutateurs DIP) parce qu'il utilise le LCD pour afficher les informations d'état. L'esclave dispose d'un *shield* relais à 4 canaux où

chaque relais correspond à un nœud avec ses propres numéros d'étage, de pièce et de prise (à définir au début du croquis dans la structure de données des nœuds). Ces valeurs sont utilisées pour créer l'ID de message d'un nœud.

Chaque nœud a aussi sa propre tonalité, ce qui permet d'utiliser l'esclave multinœuds sans la carte relais.

Les boutons-poussoirs PC0, PC1, PC2 et PD3 (pas PC3) commutent les nœuds et leur état (le nœud le plus récemment mis à jour) est affiché sur le LCD. Chaque fois qu'un nœud change d'état, soit localement par pression d'un bouton soit à distance par le maître, un message de mise à jour d'état est transmis sur le bus CAN. Ceci fournit non seulement un retour au maître, mais permet aussi un balayage total ou partiel du réseau en une seule fois simplement en utilisant la valeur « All » pour un ou plusieurs des paramètres étage, pièce et prise. Comme tous les nœuds vivants du réseau vont répondre à ce message avec une mise à jour d'état, le maître peut maintenant facilement en établir une liste. Nota : cette toute dernière étape sera un exercice pour le lecteur.

### Esclave simple

Un second croquis d'esclave a aussi été écrit, beaucoup plus simple et prenant en charge un seul nœud (ou ID de mes-



## LISTE DES COMPOSANTS

### Divers

2× carte AVR Playground  
2× carte CAN SPI click  
1× *shield* relais à 4 canaux  
Câble à paire torsadée

sage). Des constantes de configuration figurent en haut de ce croquis, le rendant utilisable avec une carte Arduino Uno (ou compatible) et d'autres modules MCP2515 qui peuvent comporter des quartz différents (c.à.d. fréquences).

### Conclusion

Nous avons montré ici comment configurer sans trop d'effort un réseau CAN grâce aux matériel et code source libres et aux modules contrôleurs CAN compatibles MCP2515 largement diffusés. Avec cette approche, nous avons pu nous concentrer sur l'application, un système de commande de l'éclairage d'un immeuble.

Le code présenté dans cet article peut être librement téléchargé en [1]. ◀

(160464 – version française : Denis Lafourcade)

### Les trames de requête

Le croquis maître envoie des demandes d'état directement à un nœud, avec l'ID (message) du nœud, ce qui signifie qu'il y a deux nœuds utilisant le même ID de message sur le bus, le maître et le nœud lui-même, une situation que la spécification CAN essaye d'éviter. La Trame de Requête (*Remote Frame*), spécifiée par la norme CAN, permet cela. Dans une trame de requête, le bit RTR est récessif alors que dans une trame normale il est dominant. Ainsi, si une trame de données et une trame de requête avec le même ID de message sont transmises en même temps, les données gagneront l'arbitrage, car elles ont une priorité plus haute, le demandeur recevant alors ce qu'il a demandé, tout est pour le mieux.

La principale raison de l'existence des trames de requête est d'éviter que les nœuds inondent le réseau avec des messages qui n'intéressent personne. Cela semble en contradiction avec ce que nous expliquions en début d'article, mais en autorisant les nœuds à ne demander que les données dont ils ont réellement besoin, on réduit le trafic sur le réseau.

### Liens

- [1] Page de l'article : [www.elektormagazine.fr/160464](http://www.elektormagazine.fr/160464)
- [2] Carte AVR Playground : [www.elektormagazine.fr/labs/avr-playground-129009-2](http://www.elektormagazine.fr/labs/avr-playground-129009-2)
- [3] Bibliothèque CAN : [https://github.com/Seeed-Studio/CAN\\_BUS\\_Shield](https://github.com/Seeed-Studio/CAN_BUS_Shield)

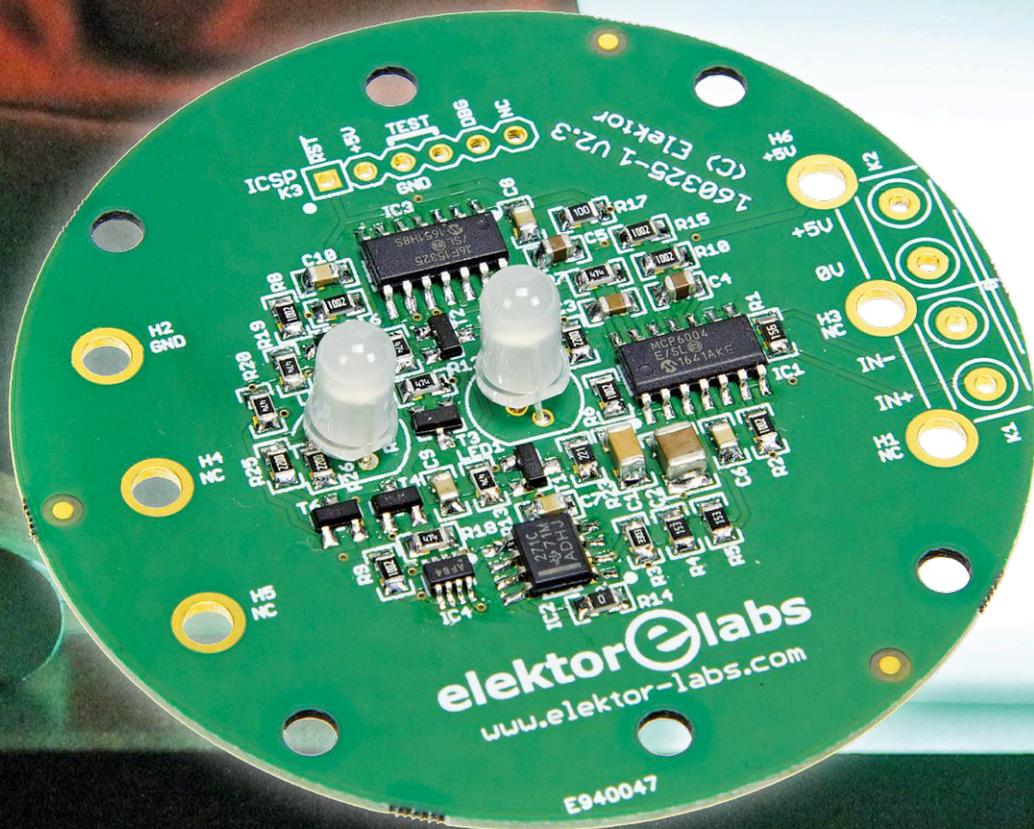


## DANS L'E-CHOPPE

→ 129009-2

AVR Playground

Carte CAN SPI click (5 V)





# bio-lampe

**les « émotions »  
d'une plante en couleurs**

Walter Polleros (Autriche) et Clemens Valens (labo d'Elektor)

Proches depuis toujours des plantes, arbres, fleurs, arbustes, herbes bonnes ou mauvaises, nous n'avons pourtant jamais trouvé le moyen de communiquer avec elles. Elles sont vivantes, nous le savons, mais à quoi pensent-elles et que ressentent-elles ? Comme nous n'en avons pas la moindre idée, nous avons réalisé la bio-lampe pour visualiser les émotions de nos vertes amies.

## Caractéristiques

- Traduit les bio-s signaux en lumière colorée
- Design discret et élégant
- Alimentation à piles
- Fonctionne très bien avec les plantes d'intérieur communes

Soyons clairs dès le début : nous ne prévoyons pas tout savoir sur la vie émotionnelle secrète des plantes. En fait nous ne savons même pas si elles en ont une. Tout ce que nous savons c'est que les plantes produisent des signaux électriques et – en tant qu'électroniciens – nous connaissons bien ces signaux et la façon de les traiter. Sans pouvoir les interpréter, nous devrions néanmoins pouvoir les rendre visibles avec des LED multicolores. Voilà ce qu'est la bio-lampe, un dispositif qui rend visibles les signaux électriques (bio-s signaux) produits (pensons-nous) par les plantes.

C'est connu depuis des années, des signaux bioélectriques existent dans les plantes (dorénavant nous laisserons tomber « électriques »), notre savoir sur leurs significations et fonctions est à vrai dire minimal. L'être humain produit aussi des bio-s signaux ; l'électroencéphalogramme (EEG) et l'électrocardiogramme (ECG) sont deux exemples bien connus. Dans cet article, un bio-s signal est une différence de potentiel variable entre deux parties d'une plante.

Pour mesurer un tel signal, il suffit de coller deux électrodes quelque part dans ou sur la plante et d'amplifier la différence de potentiel entre elles. Le microcontrôleur de la bio-lampe numérise le signal et l'associe à un spectre de couleurs qui peut être reproduit avec une LED RVB.

### Deux signaux

Au cours de l'expérimentation, on a constaté que le signal de la plante est en réalité composé de deux signaux : une porteuse lente sur laquelle vient se superposer un signal faible plus rapide (fig. 1). Lente signifie ici plusieurs secondes ou minutes, voire plus – les signaux vraiment rapides se situent dans la gamme de 2 à 5 Hz. La bio-lampe rend visibles ces signaux (les plus) rapides, considérés comme plus intéressants. Toutefois, nous devons reconnaître que nous ne sommes pas sûrs que ce signal soit réellement produit par la plante ou bien causé par l'environnement, car il est quelquefois possible d'obtenir les mêmes effets sans que les électrodes soient connectées à la plante.

### Compensation de décalage

Le signal intéressant est très faible, de l'ordre de quelques millivolts, et doit être amplifié. Mais le faire sans précautions amplifierait aussi la porteuse qui a un niveau bien plus haut (dizaines à centaines de millivolts) avec pour résultat un écrêtage. Pour obtenir des résultats utiles, la porteuse doit être rejetée avant amplification, c.-à-d. qu'il faut un circuit de réjection de mode commun. La bio-lampe résout ce problème élégamment. Au lieu d'ajouter ou soustraire une tension de décalage à l'entrée, elle traite la porteuse elle-même comme une tension de décalage. Comme la plage de signal de la porteuse est similaire à la plage de compensation de décalage d'un amplificateur opérationnel (AOP), la porteuse peut être éliminée par la fonction d'annulation de tension de décalage externe d'un AOP. Ceci nécessite, bien sûr, que l'AOP utilisé ait une telle fonction, ce qui est heureusement le cas de beaucoup de circuits intégrés mono-AOP.

La suppression du décalage nécessite généralement l'utilisation d'un poten-

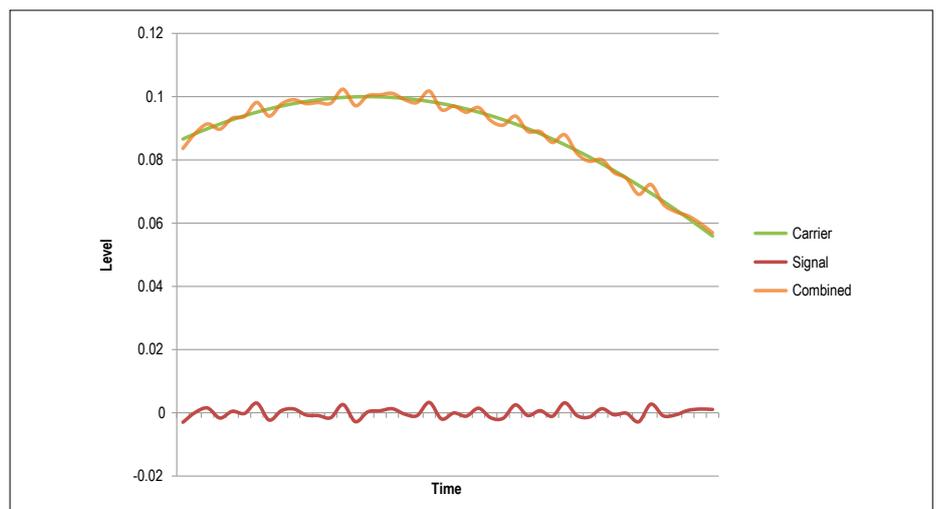


Figure 1. Le signal utile de la plante surfe sur une vague à évolution lente.

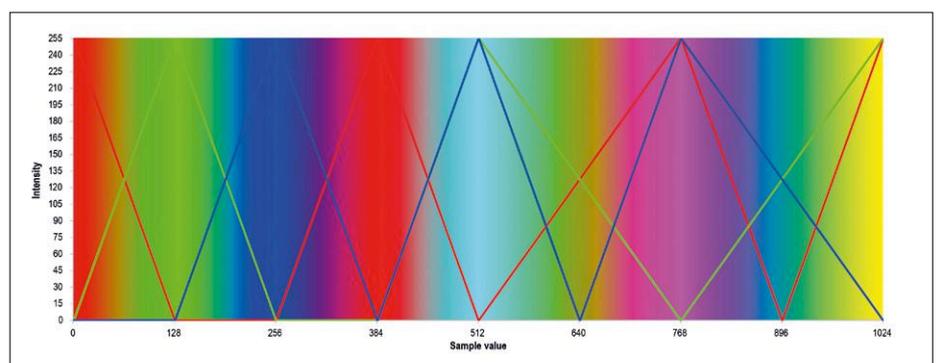


Figure 2. Cette charte de couleurs montre comment un échantillon est associé à une couleur. Lorsque la porteuse a été complètement éliminée, la couleur de la plante évolue autour du cyan.

### INFOS SUR LE PROJET

- LED RVB
- bio-signal
- PIC16F15325

débutant → **connaissseur** expert

env. 4 h

Programmeur PIC, poste de soudage CMS, perceuse

env. 40 €

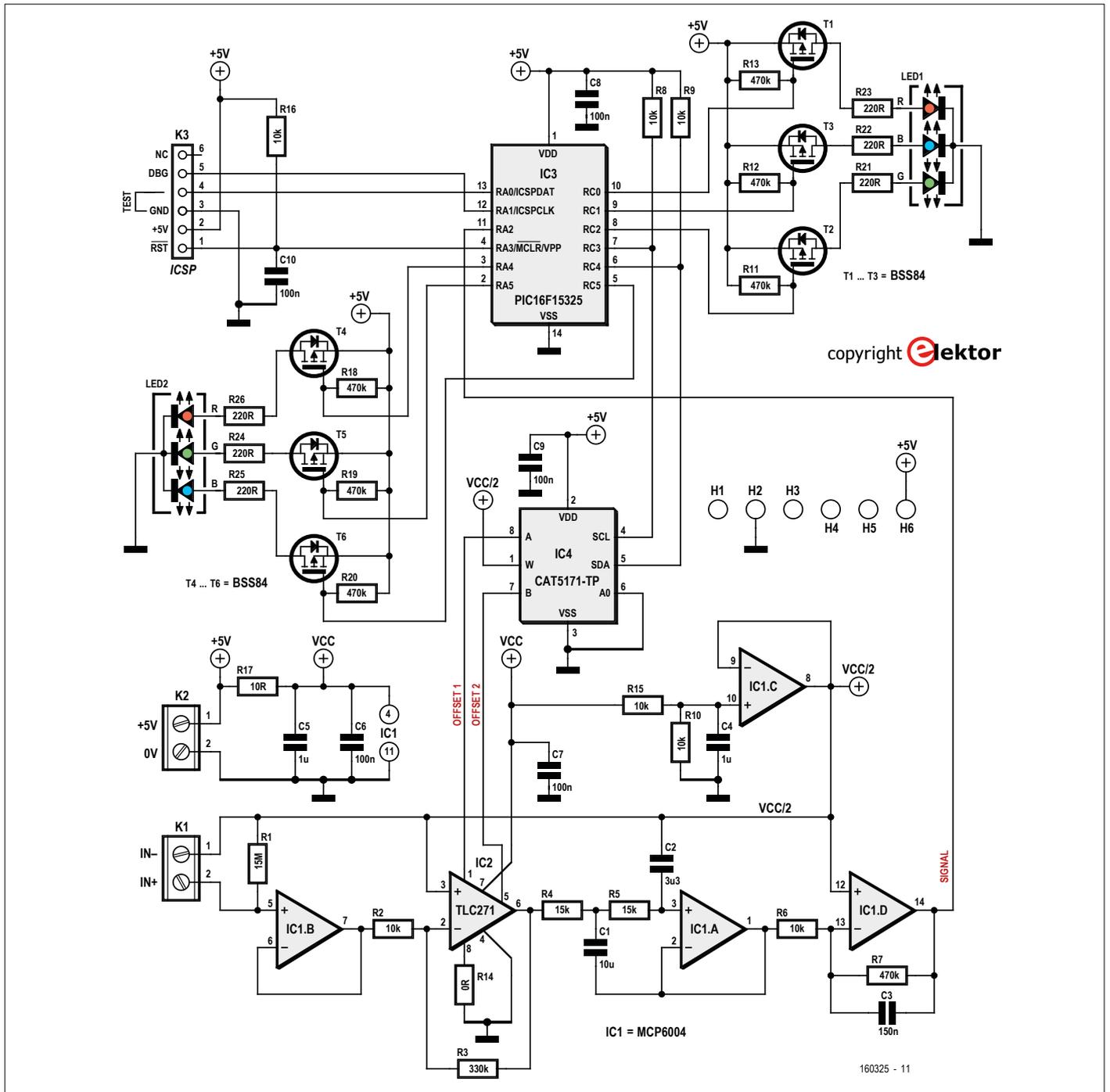


Figure 3. Après amplification et filtrage du signal, le  $\mu\text{C}$  (IC3) le transpose dans le domaine numérique où il est associé à une couleur. En fonction de la valeur de l'échantillon, il peut alors régler le potentiomètre numérique IC4 pour recentrer le signal.

tiomètre de réglage, d'où le recours à un potentiomètre numérique.

**Le logiciel**

Après suppression de la porteuse, le signal utile est amplifié et filtré avant numérisation par un microcontrôleur. Le programme à l'intérieur du  $\mu\text{C}$  échantillonne le signal à environ 27 Hz, une valeur choisie pour éviter autant que possible tout bruit de ligne CA (50 ou 60 Hz) qui aurait pu traverser les filtres d'entrée. La valeur échantillonnée est alors associée à une

couleur visuellement agréable (**fig. 2**). Lorsque la valeur du signal s'approche du niveau minimum ou maximum, la compensation de décalage agit pour déplacer le signal dans la direction opposée. Pour éviter les oscillations dues à la latence du filtre, la vitesse d'ajustement de la compensation de décalage doit être inférieure à la fréquence de coupure du filtre passe-bas. Si le décalage est compensé trop rapidement, le signal peut déborder car le filtre met du temps à le rattraper. Il sera alors envoyé dans l'autre sens

où il pourra aussi déborder avant que le filtre ne le rattrape. Le résultat est une compensation de décalage oscillante, un effet hautement indésirable.

**Le circuit**

Un seul petit pas sépare la description de haut niveau ci-dessus du circuit représenté sur la **figure 3**. Le signal d'entrée, c.-à-d. la plante, est connecté à K1 où il est tamponné par IC1.B. Il est alors amplifié par IC2 qui fournit aussi la compensation de déca-

## Évolution de la bio-lampe

Walter Polleros a conçu la bio-lampe et l'a présentée au concours **Fast Forward Award** organisé par Elektor au salon *electronica 2016* à Munich. Le prototype consistait en une boîte surmontée d'une balle translucide, éclairée par en-dessous par une puissante LED RVB. Elle était connectée à une plante qui influençait la couleur de la balle. Il était très intéressant de voir comment la plante semblait réagir aux badauds tandis qu'elle était presque statique par moments, comme si elle se reposait. Après le salon Walter publia son projet sur Elektor Labs et nous autorisa à développer notre propre bio-lampe tandis qu'il continuait à travailler sur la sienne.

Le projet original utilise une LED de haute puissance, alimentée en 24 V. Notre bio-lampe devait être alimentée en 5 V par un chargeur de téléphone ou trois piles AA(A), aussi avons-nous ajouté un convertisseur-élévateur. Ensuite nous avons découvert des cache-pots translucides avec une LED RVB à l'intérieur pour les faire changer de couleur. Cette découverte nous a incités à abandonner la LED de puissance et à développer une carte qui s'intégrerait dans un tel cache-pot.



Bio-lampe de luxe issue d'une campagne de financement participatif, une balle translucide sur un piédestal en noyer.

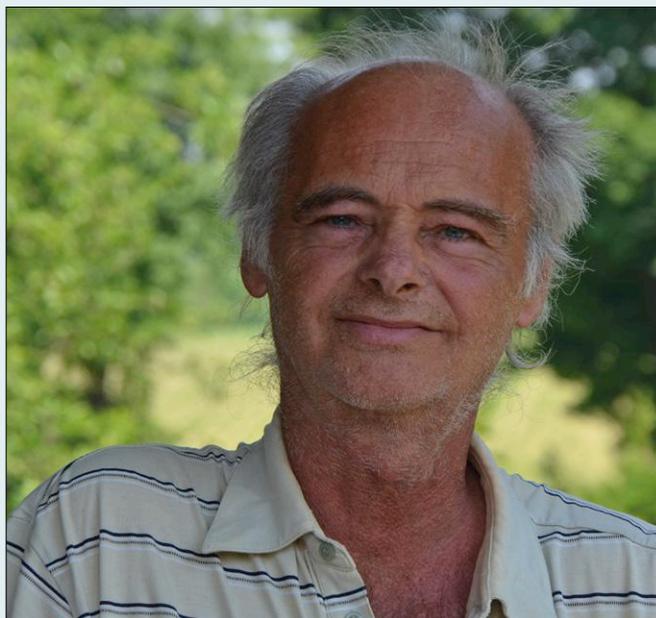
Alors que le projet de Walter utilisait une alimentation symétrique de  $\pm 12$  V pour le conditionnement du signal et du 5 V pour le  $\mu$ C, nous voulions du 5 V pour tout. Il fallait adapter la partie analogique à un fonctionnement en simple alimentation et donc remplacer les AOP TL061 & TL082 de Walter par des modèles rail à rail MCP6004. Le TL061 initialement utilisé pour la suppression de la porteuse a été remplacé par un TLC271. La compensation de décalage originale était réglée par un potentiomètre numérique AD5220. Comme ce composant est cher et difficile à trouver, nous l'avons remplacé par un CAT5171. Bien que beaucoup plus petit et moitié moins cher, il a une interface I<sup>2</sup>C au lieu d'un compteur à incrément/décément qui nécessite

un pilote logiciel différent.

Le projet d'origine était basé sur un ancien  $\mu$ C à 28 broches PIC16F882. Comme nous pensions que ce  $\mu$ C était surdimensionné, nous avons interrogé Microchip pour une solution plus adaptée. Ils nous ont recommandé le PIC16F15325, un modèle moderne en boîtier à 14 broches de la série milieu de gamme améliorée.

Remplacer le  $\mu$ C impliquait aussi le portage du micrologiciel. Celui de Walter était écrit en assembleur mais nous l'avons complètement réécrit en C (plus populaire). Un pilote I<sup>2</sup>C a été ajouté et les signaux MLI de commande de la LED RVB sont produits par le matériel plutôt que par le logiciel.

Bien que le nouveau  $\mu$ C n'ait que 14 broches, il y en a encore trois de trop (sans compter les deux dédiées au connecteur ICSP), aussi avons-nous ajouté une deuxième LED RVB. Nous avons d'abord montré avec elle le niveau du signal porteur, mais à la fin nous avons décidé de lui faire faire la même chose que la LED principale pour augmenter l'intensité lumineuse. Elle a son propre pilote MLI dans le micrologiciel (nous avons empilé six signaux MLI indépendants dans ce contrôleur) et elle est



Walter Polleros, créateur de la bio-lampe.

disponible pour ceux qui voudraient essayer d'autres fonctions. Finalement, même si notre bio-lampe repose exactement sur le même principe que l'originale, presque tous les composants ont été remplacés.

Nous terminons notre projet lorsque Walter nous a appris qu'il avait lancé une campagne de financement participatif pour le sien. Malheureusement cette campagne sera terminée quand vous lirez ces lignes, mais allez la voir quand-même.

<http://igg.me/at/Bio-Light>

lage. R14 définit le courant de polarisation de l'AOP. Avec le montage indiqué, le courant de polarisation est élevé. Ceci fournit la plus large plage de compensation de décalage mais augmente aussi la puissance consommée. Pour utiliser ce circuit en mode « basse puissance », on doit connecter la broche 8 d'IC2 à VCC. La résistance R1 est une énorme 15 M $\Omega$ , valeur choisie pour obtenir un signal correct de la plante. Toutefois, c'est une valeur qui peut être difficile à trouver. On peut diminuer cette valeur jusqu'à environ 4,7 M $\Omega$  au détriment de l'amplitude du signal. On peut ajuster le gain de l'étage d'amplification pour compenser cela.

L'étape suivante dans la chaîne de traitement du signal est le filtre passe-bas construit autour d'IC1.A. Il a une fréquence de coupure d'à peu près 2 Hz. IC1.D rajoute de l'amplification tout en supprimant les fréquences supérieures à environ 2 Hz (**fig. 4**). L'étage d'entrée a un gain total autour de 1500.

IC1.C fournit une masse virtuelle pour l'étage d'entrée à alimentation simple. Le potentiomètre numérique IC4 (50 k $\Omega$ ) règle la compensation de décalage. Le circuit intégré lui-même est commandé par un bus I<sup>2</sup>C connecté au  $\mu$ C, un PIC16F15325 en boîtier à 14 broches (voir **encadré**). En plus de commander le potentiomètre de réglage, il délivre aussi les signaux MLI qui règlent les couleurs des deux LED RVB. LED1 est la LED principale placée au centre du circuit imprimé circulaire, LED2 sert pour la couleur de fond d'accompagnement. Les transistors T1 à T6 débitent le courant nécessaire pour que les LED brillent avec éclat. Le connecteur K3 sert à la programmation in situ du  $\mu$ C.

### Construire une bio-lampe

Nous avons conçu un circuit imprimé d'une épaisseur de 1 mm et d'un diamètre extérieur de 66 mm, adapté à notre cache-pot translucide. Bien que très dense avec de minuscules composants CMS – tous concentrés dans un cercle intérieur de 36 mm – il n'est pas trop difficile à assembler pour le guerrier du fer à souder le plus expérimenté. Le composant le plus délicat est sans doute IC4, non parce qu'il est minuscule, mais parce qu'il faut de bons yeux pour voir le repère de la broche 1.

Il faut fixer les LED près de la carte et couper leurs broches à ras de l'autre côté. Tâchez de conserver ce côté de la carte aussi lisse que possible.

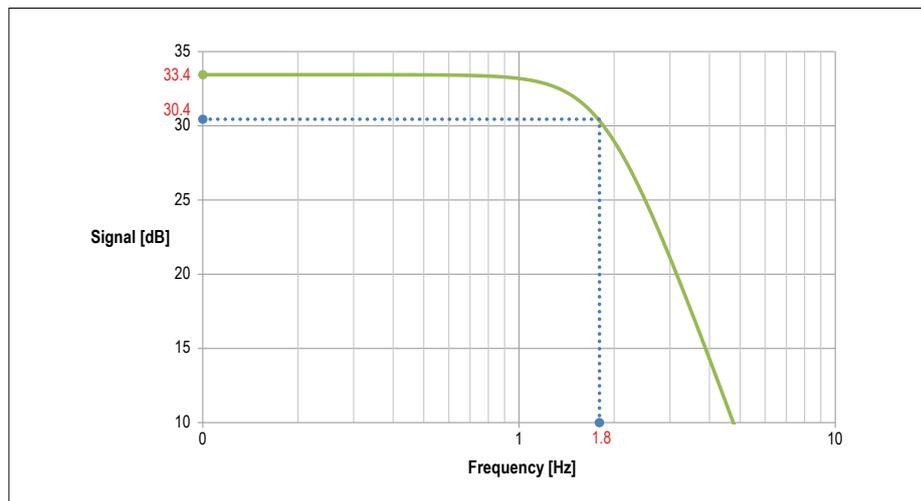


Figure 4. Fonction de transfert de l'étage de filtrage et d'amplification constitué d'IC1.A et IC1.D. Le gain est légèrement supérieur à 33 dB et la fréquence de coupure se situe à 1,8 Hz.

Ne placez aucun des connecteurs. Téléchargez le logiciel depuis [1]. Pour programmer le  $\mu$ C, enfoncez une barrette mâle avec des broches assez longues dans les trous de K3, connectez-la à une prise de programmation en pressant la barrette sur le côté pour assurer un bon contact pendant que le programme est téléchargé. Retirez-la lorsque c'est fini.

### Autotest

Il est temps de tester la carte. Court-circuitez les broches 3 et 4 de K3 avec un bout de fil, ainsi que les broches 1 et 2 de K1, puis branchez une alimentation 5 V sur K2 (ou aux trous H2 et H6). Lorsque l'alimentation est activée, la carte exécute un autotest. D'abord LED1 balaye ses couleurs rouge, vert et bleu, puis LED2 fera de même. On a alors un éclair blanc des deux LED.

Ensuite vient le potentiomètre numérique ; si vous connectez un voltmètre sur la broche 6 d'IC2, il doit indiquer une tension qui varie lentement. Pendant ce temps LED2 devrait s'affaiblir, ce qui indique que la communication I<sup>2</sup>C est active. Enfin les deux LED clignotent encore une fois, et une seconde plus tard le fonctionnement normal commence. Une fois les tests passés avec succès, ôtez le pont de K1. Vous pouvez aussi enlever celui de K3, mais cela ne gêne pas de le laisser.

### Montage du cache-pot

Notre cache-pot translucide est en deux parties. Dans la moitié inférieure se trouve un support circulaire pour trois piles AAA avec interrupteur intégré. La

plante vient dans la moitié supérieure. Sortez le support de piles. Il y a une LED RVB dessus ; enlevez-la mais gardez les fils de l'interrupteur.

Il y a des trous sur le circuit imprimé à l'endroit des connexions au support de piles. Placez la carte de façon à ce que le pôle négatif du support de piles traverse H2 (GND). Vous devrez peut-être appuyer dessus et le déformer un peu. De même le pôle positif doit traverser H1 (NC). La carte doit reposer aussi à plat que possible sur le support de piles. Soudez la carte au support de piles. Soudez le fil volant de l'interrupteur à H6 (+5 V), voir le **figure 5**. Si vous mettez des piles neuves dans le support de piles et allumez le montage, les LED doivent s'éclairer.

Percez un trou d'environ 3 mm sur le bord du compartiment à piles, près du fond (**fig. 6**). Soudez deux fils longs sur K1 et passez-les au travers du trou précédemment percé. Ils doivent maintenant être à



Figure 5. Le circuit de la bio-lampe monté sur le support de piles. Notez comment l'interrupteur est connecté entre les trous H1 et H6.

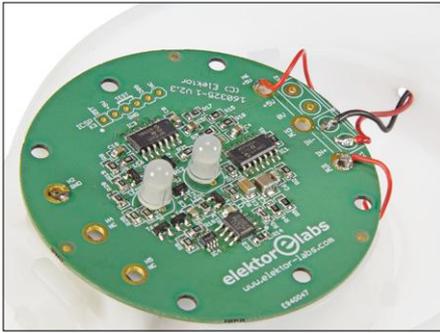


Figure 6. Percez un trou pour les fils d'électrodes dans le compartiment des piles près du fond du cache-pot de façon à éviter autant que possible que de l'eau pénètre.

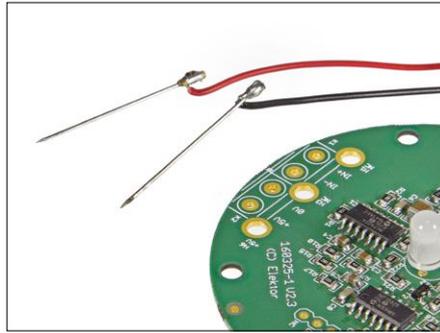


Figure 7. Des épingles de couture forment des électrodes pratiques, faciles à fixer sur une plante. Il y a sans doute de meilleures solutions, moins douloureuses, comme des épingles à cheveux.



Figure 8. Voici comment nous avons branché notre spathyphillum. Les électrodes sont peut-être trop envahissantes ?

l'intérieur de la base du cache-pot. Tirez sur les fils jusqu'à ce que vous puissiez reboîter le support de piles dans son compartiment. Ceci fait, vérifiez que le montage fonctionne toujours.

### Branchement de la plante

Vous pouvez percer un trou dans le rebord de la moitié supérieure du cache-pot et passer les fils au travers pour les cacher autant que possible, ou vous pouvez les laisser sortis entre les deux moitiés (ce que nous avons fait). Soudez une épingle de couture en métal ou une

aiguille à coudre à l'extrémité de chaque fil (**fig. 7**). Si vous trouvez cette méthode barbare, vous pouvez essayer avec des épingles à cheveux ou des trombones. Assemblez les deux moitiés du cache-pot et insérez la plante. Plantez les aiguilles dans la plante éloignées l'une de l'autre, par ex. dans deux branches différentes, ou bien l'une dans une branche et l'autre dans une feuille d'une autre branche (**fig. 8**). Allumez le cache-pot. Au début les couleurs devraient changer de façon erratique. C'est normal, il faut un moment avant que le signal porteur soit supprimé

et que la plante retrouve sa sérénité. Quelquefois, sans raison apparente, la couleur va changer de façon chaotique pendant environ une demi-minute (pourquoi ?). Si rien ne se passe et que la couleur ne change pas du tout, essayez une autre plante. Nous avons obtenu de bons résultats avec une fleur de lune (spathyphillum), plante reconnue par la NASA pour améliorer la qualité de l'air intérieur.

### Faisons équipe !

Comme nous ne savons pas quoi faire des couleurs produites par une seule plante, il pourrait être intéressant de collecter et comparer les données de plusieurs plantes. Peut-être que si un grand nombre de personnes installaient une bio-lampe avec la même espèce de plante (un spathyphillum), connectée de la même façon, et qu'elles partageaient leurs observations, il serait possible de détecter des modèles et de trouver une signification à ces signaux ? Faisons équipe et perçons ensemble le secret des bio-signaux de nos végétaux. ◀

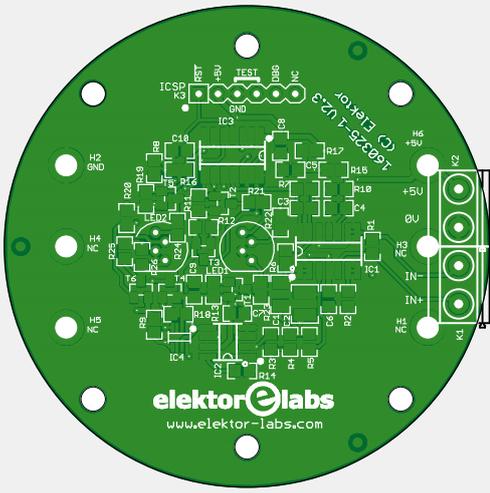
(160325 - version française : Denis Lafourcade)

### Liens

- [1] [www.elektormagazine.fr/160325](http://www.elektormagazine.fr/160325)
- [2] [www.elektormagazine.fr/labs/Bio-Light](http://www.elektormagazine.fr/labs/Bio-Light)



## COMPONENT LIST



**Résistances :**  
Toutes CMS 0805

R14 = 0 Ω  
R17 = 10 Ω  
R21, R22, R23, R24, R25, R26 = 220 Ω  
R2, R6, R8, R9, R10, R15, R16 = 10 kΩ  
R4, R5 = 15 kΩ  
R3 = 330 kΩ  
R7, R11, R12, R13, R18, R19, R20 = 470 kΩ  
R1 = 15 MΩ

**Condensateurs :**  
Sauf indication contraire : 0805

C6, C7, C8, C9, C10 = 100 nF  
C3 = 150 nF  
C4, C5 = 1 μF  
C2 = 3,3 μF, 1210  
C1 = 10 μF, 1206

**Semi-conducteurs :**  
IC1 = MCP6004-E/SL  
IC2 = TLC271ID  
IC3 = PIC16F15325-I/SL, programmé  
IC4 = CAT5171TBI-50GT3  
T1, T2, T3, T4, T5, T6 = BSS84P  
LED1, LED2 = RVB, CC, 5 mm

**Divers :**  
K1, K2 = bornier à vis pour circuit imprimé, à 2 voies, au pas de 5,08 mm  
K3 = barrette mâle à 6 picots, au pas de 2,54 mm  
Circuit imprimé réf. 160325-1 (épaisseur 1 mm)  
2 épingles de couture ou épingles à cheveux  
Cache-pot translucide à LED (H 19 cm, D 17 cm) avec compartiment à piles de 66 mm de diamètre



## DANS L'E-CHOPPE

→ 160325-1  
circuit imprimé nu

→ 160325-71  
lot de composants

→ 160325-91  
module prêt à l'emploi

# coûteau suisse pour microcontrôleurs

## PlatformIO, un outil universel de programmation



Markus Ulsaß (Allemagne)

Les programmeurs de systèmes embarqués qui travaillent fréquemment avec diverses cartes connaissent bien le problème : l'environnement de développement intégré d'un fabricant ne tourne souvent que sous un seul système d'exploitation et ne concerne que des microcontrôleurs d'un type bien déterminé. D'autres EDI sont plus universels, mais n'ont que des fonctions très limitées ou sont (très) coûteux. L'environnement de programmation universel PlatformIO se propose d'en finir avec ces inconvénients. Il supporte un grand nombre de systèmes d'exploitation, de cartes et d'environnements de développement et offre une quantité de fonctions importantes. Nous nous proposons d'illustrer sa prise en main avec des exemples simples.

PlatformIO est un projet à code source ouvert, né il y a environ trois ans de l'initiative du programmeur ukrainien Ivan Kravets. Ce projet doit offrir une base commune de programmation pour des microcontrôleurs de différents fabricants, sous des systèmes d'exploitation et des environnements de développement divers. Cette base est fournie par *PlatformIO Core*, un outil de programmation à code source ouvert, gratuit, orienté ligne de commande et écrit en Python 2.7. Il peut être intégré à divers environnements au moyen de modules d'extension (*plugins*). Actuellement, le projet supporte plus de 400 cartes de développement (dont Atmel à 8/32 bits, ESP8266/ESP32, Freescale, Microchip, Nordic Semiconductor, Silicon Labs, STMicro), 20 éditeurs de texte/platformes de développement (Atom, VS Code, Eclipse, VIM, Visual Studio etc.) et 10 cadres (*frameworks*, dont Arduino, CMSIS, ESP-IDF, mbed). De plus, des centaines de bibliothèques logicielles sont disponibles pour cet environnement, et le projet est en croissance permanente. On trouvera le dépôt *github* de PlatformIO sous [1] et son site web avec une documentation complète sous [2].

Deux des environnements de programmation/éditeurs de texte les plus populaires pour l'intégration de *PlatformIO* sont *Atom* et *VisualStudio Code*. Pour chacun d'entre eux, il existe un plugin *PlatformIO IDE* qui fournit une interface utilisateur graphique (GUI) pour *PlatformIO Core*.

Dans la suite, nous examinerons l'installation, quelques fonctions et exemples de programmation à l'aide du *plugin* pour l'éditeur de texte Atom [3]. C'est celui dont l'intégration est la plus avancée et qui offre un grand nombre de possibilités d'utilisation.

### PlatformIO IDE pour Atom

Pour commencer, il faut le fichier d'installation d'Atom, disponible sous [3] pour le système d'exploitation choisi. Après l'installation, le programme démarre automatiquement.

L'installation de paquets complémentaires permet d'ajouter à l'éditeur de texte Atom de puissantes fonctions. Pour intégrer le paquet PlatformIO IDE dans Atom, on sélectionne *Install a*

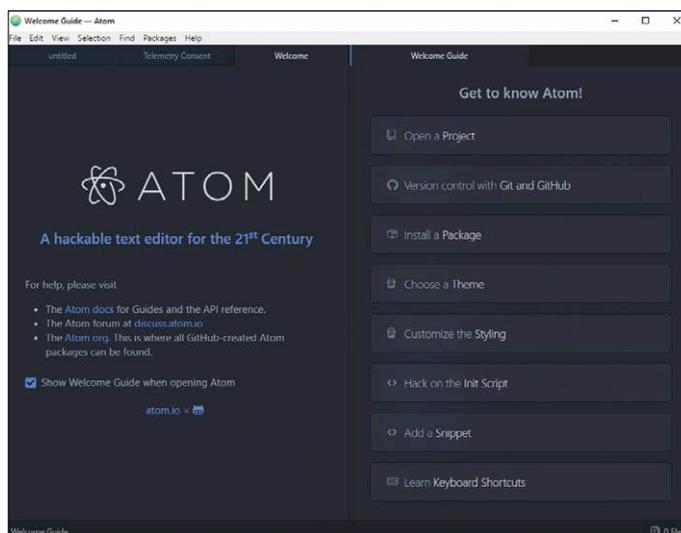


Fig. 1. On peut intégrer des modules supplémentaires (*Packages*) dans l'éditeur de texte Atom.

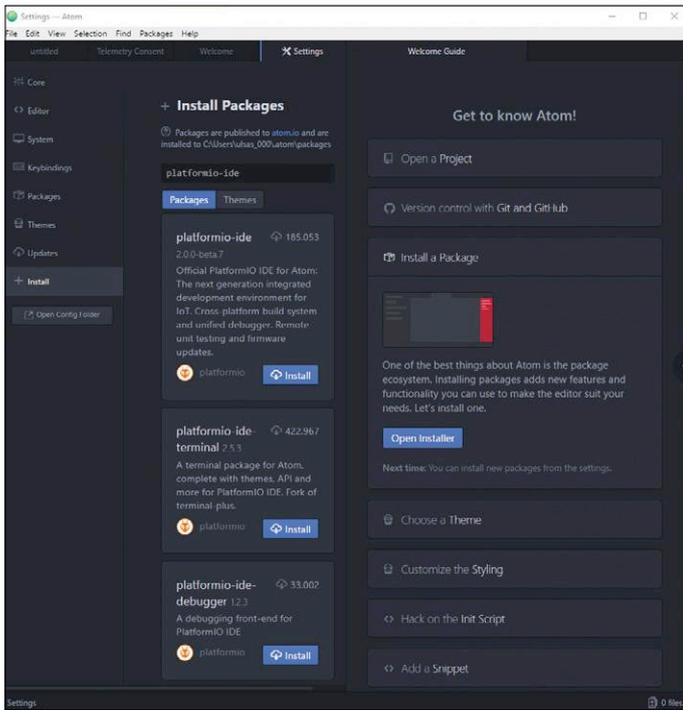


Fig. 2. L'EDI PlatformIO se présente sous forme de paquet (Package).

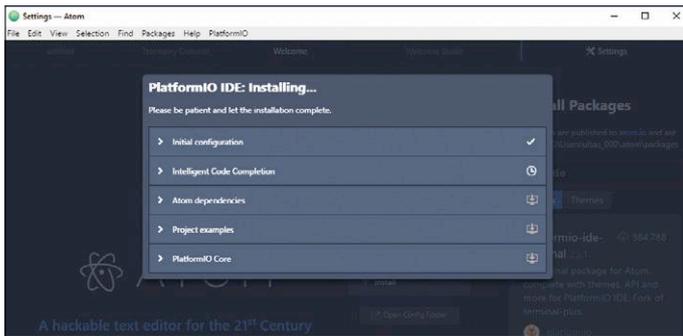


Fig. 3. Soyez patient !

*Package* sous l'onglet de droite (fig. 1). Puis on clique sur le bouton bleu *Open Installer* et l'on saisit *platformio-ide* dans le champ de recherche de l'onglet qui vient de s'ouvrir. On confirme avec *Entrée*. Dans la liste de sélection apparaissent diverses extensions PlatformIO ; nous choisissons *platformio-*

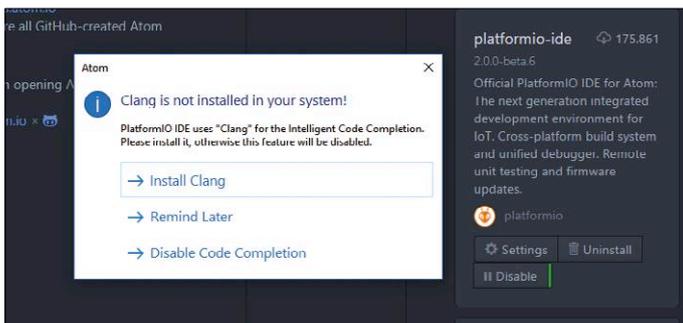


Fig. 4. Clang et Python doivent être installés manuellement.

*ide* et cliquons sur le bouton *Install* (fig. 2).

Alors commence le processus d'installation du paquet PlatformIO-IDE (fig. 3) qui exige, si ce n'est déjà fait, l'installation de Clang (compilateur pour C/C++ avec bibliothèques LLVM « Low Level Virtual Machine ») et de Python 2.7.

À certains moments du processus d'installation, quand on clique sur *Install Clang* (fig. 4) et *Install Python*, on est orienté vers les pages web des fichiers et des directives d'installation. Le téléchargement et l'installation de ces composants doivent être effectués manuellement.

Dans le cas de Clang, sur la page web [4], il faut faire défiler la page jusqu'à l'option *II. Clang for intelligent Code Completion* qui présente les directives d'installation pour le système d'exploitation utilisé.

Au cours de l'installation de Clang pour Windows, il est important de choisir dans la page de paramétrage de LLVM l'option *Add LLVM to the system PATH for all users*, sinon on risque l'apparition ultérieure de messages d'erreur (fig. 5). Pour toutes les autres options, on peut conserver les valeurs par défaut.

De même, pour l'installation de Python, sous *Customize Python 2.7.12* et pour *Add python.exe to Path*, il faut choisir l'option *Will be installed on local hard drive* (fig. 6).

Une fois l'installation du paquet et des logiciels additionnels requis terminée, il faut redémarrer Atom (fig. 7). Après le redémarrage, tous les composants sont installés et nous pouvons procéder au test du code d'un programme, qu'on peut télécharger, gratuitement comme toujours, sur le site web d'Elektor [5].

### Prêt à coder

Pour un premier exemple, sous l'onglet *PlatformIO Home* qui vient de s'afficher, nous actionnons le bouton *Project Examples* (fig. 8). Dans la liste d'exemples proposés, nous choisissons *espressif/esp8266-wifiscan* et validons avec le bouton bleu *Prepare* (fig. 9).

Comme il s'agit à l'origine d'un projet réalisé avec l'EDI Arduino, une fenêtre s'ouvre qui avertit que le vérificateur de syntaxe C/C++ *Smart Code Linter* est désactivé pour les croquis Arduino. Pour l'instant, nous l'activerons manuellement (*Enable*) ; pour un projet ultérieur, nous utiliserons un fichier .cpp. La page qui s'ouvre automatiquement, nous renseigne sur les particularités d'utilisation de code Arduino et du Smart Code Linter.

Dans la colonne de droite apparaît sous l'onglet *WiFiScan.ino* le croquis Arduino (fig. 10). Si le fichier ne s'ouvre pas automatiquement, nous procédons manuellement en cliquant sur son nom dans la colonne *Project* à gauche.

Pour une meilleure lisibilité, nous pouvons fermer les onglets relatifs à Atom, ainsi que l'onglet d'accueil de PlatformIO IDE.

Cette copie d'écran montre l'une des forces de PlatformIO IDE. Rangés de façon claire, on a, à gauche, une colonne menu, puis une vue d'ensemble des fichiers du projet et du code proprement dit, enfin, à droite, un repérage rapide de la position dans le code. Le bandeau inférieur contient des informations d'état et d'autres menus. Pour des projets plus gros, avec plusieurs fichiers de code ou bibliothèques, cette vue d'ensemble peut constituer une aide précieuse.

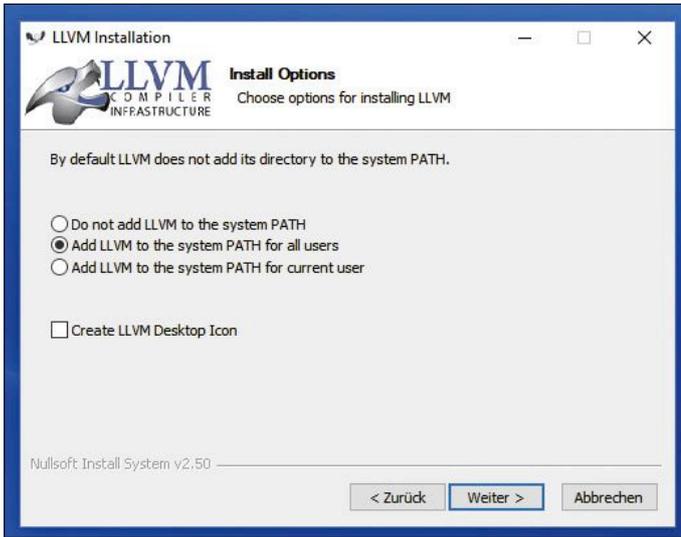


Fig. 5. Il faut choisir l'option *Add LLVM to the system PATH for all users*.

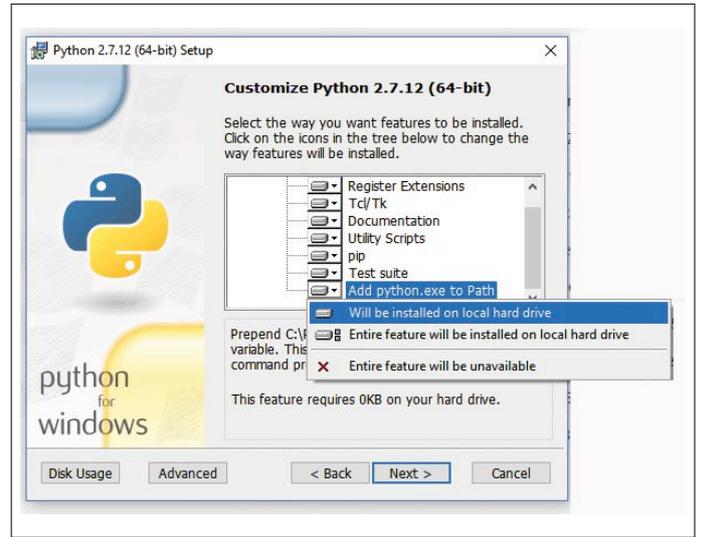


Fig. 6. L'option *Will be installed on local hard drive* est obligatoire pour Python.

Le fichier de configuration *platformio.ini* est une autre particularité de PlatformIO IDE. Il sert à sélectionner des options diverses, dont des cartes différentes pour le même code, des indications particulières pour l'utilisation de bibliothèques ou d'options de téléchargement.

Dans le code d'exemple, on voit, dans le fichier *platformio.ini*, sous *board*, que la carte *esp01* est mentionnée. Nous allons modifier le projet en conséquence, en ajoutant la carte « Wemos D1 mini ».

### Un code – plusieurs cartes

On ajoute au menu de gauche une nouvelle carte par l'option du menu *PlatformIO -> Initialize or Update PlatformIO Project* ou par l'icône dédiée (le rectangle avec les signes <>). Dans la rubrique *PlatformESPRESSIF8266*, nous choisissons *Wemos*

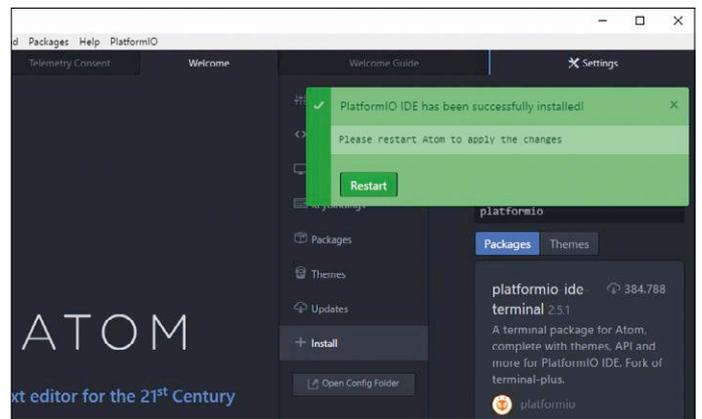


Fig. 7. Redémarrage d'Atom.

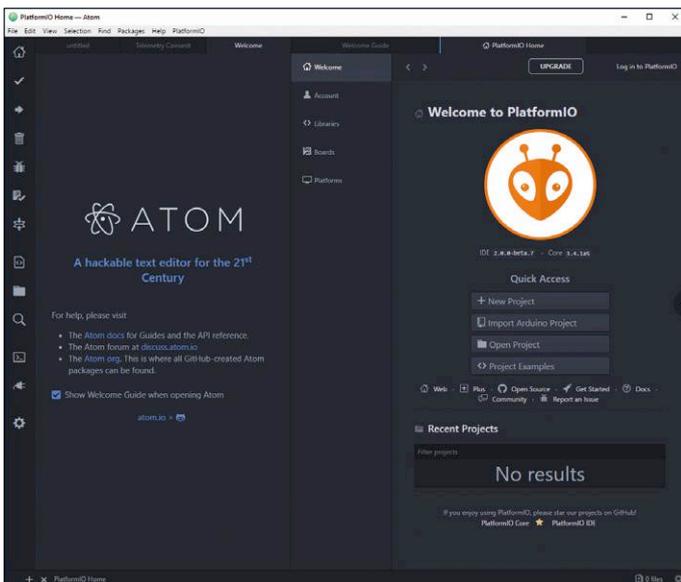


Fig. 8. Pour un premier exemple, nous cliquons sous *PlatformIO Home* sur le bouton *Project Examples*.

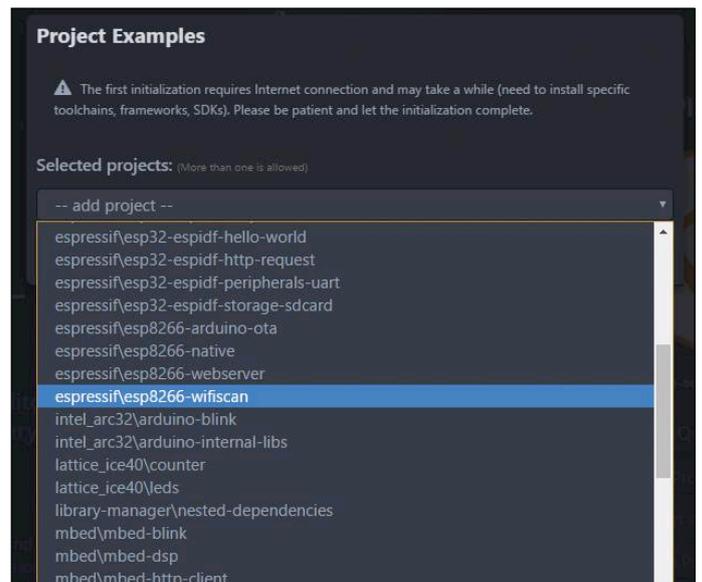


Fig. 9. C'est ici que nous choisissons notre exemple de programme.

D1 mini et cliquons sur le bouton bleu *Process*, ce qui a pour effet d'installer tous les éléments nécessaires pour la carte sélectionnée.

Dans le fichier *platformio.ini* est apparue la référence de la carte *Wemos D1 mini* (fig. 11). Nous supprimons la référence *esp01*, car seule la première carte dans le fichier de configuration est activée pour l'achèvement du code (Clang) et la vérification de syntaxe (Smart Code Linter), et enregistrons le fichier avec *Alt+S*.

Nous testons ensuite une autre fonction intéressante de PlatformIO IDE : la proposition automatique d'achèvement du code. Dans le code de *WiFiScan.ino*, juste sous *Serial.begin(115200);* nous tapons *Serial.pr*. Une fenêtre s'ouvre automatiquement avec les options possibles. Avec la souris, nous choisissons *size\_t print(char)*, validé par *Entrée*, ce qui complète automatiquement l'instruction. Nous remplaçons *char* par *Setup start -*, et notre ligne devient :

```
Serial.print("Setup start - ");
```

Comme toujours, n'oubliez pas le point-virgule ! Nous enregistrons le fichier avec *Alt+S*.

Pour créer le projet, nous cliquons sur l'icône « coche » dans le menu de gauche ou bien choisissons l'option *Build* avec *F7*. Nous connectons ensuite notre carte Wemos D1 mini à un port USB et, pour le téléversement, nous cliquons dans le menu de gauche sur l'icône « flèche droite ». La reconnaissance automatique recherche le port COM adéquat et envoie le code vers la carte.

PlatformIO IDE identifie automatiquement les cartes connectées au port USB, mais on peut aussi lui indiquer ses propres paramètres en modifiant le fichier de projet *platformio.ini* (port, vitesse, etc. voir ci-dessous).

Dans le menu de gauche, nous cliquons maintenant sur l'icône « prise de courant » et saisissons le port utilisé (reconnu automatiquement) et la vitesse (115200 bauds) et confirmons avec *Start*. Une fenêtre de terminal s'ouvre ; le résultat d'une exploration du réseau Wi-Fi devrait y apparaître.

## Un projet personnel

Après ce passage en revue des fonctions de base de PlatformIO IDE sous Atom, lançons-nous dans la réalisation d'un projet personnel.

Créons un nouveau projet en choisissant l'option du menu *PlatformIO -> Initialize or Update PlatformIO Project* ou en cliquant sur l'icône de la page dans la colonne de gauche. Nous choisissons la carte *Wemos D1 mini*, puis *l'Arduino Uno*. Ensuite, dans un dossier de notre choix sur le disque dur, nous créons un nouveau répertoire que nous désignons comme répertoire de projet pour notre code en renseignant l'option *Choose the Directory*. Un clic sur *Process* lance le processus de production des chaînes d'outils nécessaires pour les deux cartes.

Une fois ceci terminé, nous voyons que quelques nouveaux répertoires et fichiers sont apparus dans l'espace *Project* sur la fenêtre. Les deux cartes choisies sont référencées dans *platformio.ini*. Nous connectons les deux cartes sur des ports USB pour pouvoir observer de plus près le fonctionnement de PlatformIO Core.

Dans le cas du paquet PlatformIO IDE pour Atom, on se sert essentiellement de l'interface graphique. On peut aussi envoyer des commandes à la base PlatformIO Core au moyen de la console. Pour cela, ouvrons sous l'option *PlatformIO -> Terminal -> New Terminal* (raccourci : *Alt + Shift + T* ou bien l'icône « écran » dans la colonne de gauche) un nouveau terminal qui apparaît dans la partie inférieure de la fenêtre du code.

En saisisant la commande *pio --help*, on obtient la liste des différentes options proposées par PlatformIO Core. Comme nous avons connecté deux cartes, nous voulons savoir, pour commencer, quel port est utilisé par chacune des cartes. Pour cela, saisissons la commande *pio device list*. Le système affiche les informations (fig. 12) qu'on peut également obtenir par le menu *PlatformIO -> List Serial Ports*.

Pour que les ports soient correctement utilisés au moment du téléversement, nous les spécifions dans le fichier *platform.ini*. Dans Windows, on ajoute

```
upload_port = COMX
```

pour chaque carte (en remplaçant *X* par le numéro de port effectif). Pour notre Wemos D1 mini, l'insertion complète pourrait prendre l'aspect suivant :

```
[env:d1_mini]
platform = espressif8266
board = d1_mini
framework = arduino
upload_port = COM3
```

Il faut encore créer un fichier *main.cpp* dans le répertoire *src* (clic droit sur le répertoire puis sur *New File*) avec le contenu suivant :

```
#include <Arduino.h>
// Blink Sketch
void setup() {
  pinMode(13, OUTPUT);
}

void loop() {
  delay(1000);
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
}
```

C'est le croquis *Blink* bien connu d'Arduino, avec la seule différence que nous devons le lier à la bibliothèque de base *Arduino.h*. Nous enregistrons le fichier.

Le contrôle de syntaxe est une autre fonction très utile, que nous pouvons tester en introduisant une erreur dans le code, par ex. en modifiant le nom de la fonction *pinMode* en *pinModa*. Immédiatement après l'enregistrement, une erreur de syntaxe est indiquée par un point rouge dans la ligne fautive. Si l'on clique alors dans le menu inférieur sur le point d'exclamation cerclé de rouge, on obtient un complément d'information sur l'erreur (description, ligne). Comparé à l'EDI Arduino ou à d'autres environnements de développement, on a là une méthode de signalement des erreurs très efficace.

Même le téléversement du même code vers des cartes différentes est possible de manière simple. Si nous voulons charger le code sur nos deux cartes, il suffit d'effectuer un *Build* et un *Upload* du projet. Pour sélectionner une carte particulière, nous choisissons *Run other target...* dans le menu PlatformIO (ou l'icône dans la colonne de gauche) et choisissons dans le processus *Build* ou *Upload* la carte en question (faire défiler). C'est ainsi qu'on peut tester simultanément un code identique sur des cartes de développement différentes.

### Allez plus loin !

Pour l'instant, malgré l'examen de nombre de ses fonctions, nous n'avons fait qu'effleurer la surface de l'impressionnant projet PlatformIO. Utiliser différentes bibliothèques dans le cadre d'un même projet, importer des projets Arduino ou rechercher avec le gestionnaire de bibliothèques intégré font encore partie des fonctions simples.

Avec le téléchargement (automatique) de micrologiciel, qui est aussi contrôlable à distance (*PIO remote*), un débogueur (*PIO Unified Debugger*) ainsi qu'un outil de test local et d'intégration continue (CI), PlatformIO inclut d'autres composants très puissants, mais qui ne sont pas tous gratuits. On trouvera une vue d'ensemble des fonctions et des prix sous [6].

La description de toutes les options dépasserait le cadre de cette introduction – leur intérêt n'est toutefois pas limité aux utilisateurs avancés.

Outre les services mentionnés ci-dessus, il est probable que le service (payant) *PIO Plus PlatformIO Core* jouera à l'avenir un rôle plus central. Ce noyau devrait être encore plus facile à intégrer dans d'autres environnements de développement. Sa base à code source ouvert devrait rester gratuite, sa maintenance et son développement seraient financés par les revenus des services associés.

Les développeurs et les programmeurs professionnels, qui se heurtent aux limites de l'EDI Arduino et d'autres environnements de développement (simples) ou qui apprécient l'indépendance par rapport aux plateformes, devraient tester sans tarder cet outil aux possibilités aussi multiples. Il y a fort à parier qu'ils ne voudront alors plus le lâcher. ◀

(160061 – version française : Helmut Müller)

### Liens

- [1] <https://github.com/platformio>
- [2] <https://platformio.org>
- [3] <https://atom.io>
- [4] <http://docs.platformio.org/en/latest/ide/atom.html#clang-for-intelligent-code-completion>
- [5] [www.elektormagazine.fr/160061](http://www.elektormagazine.fr/160061)
- [6] <http://platformio.org/pricing>

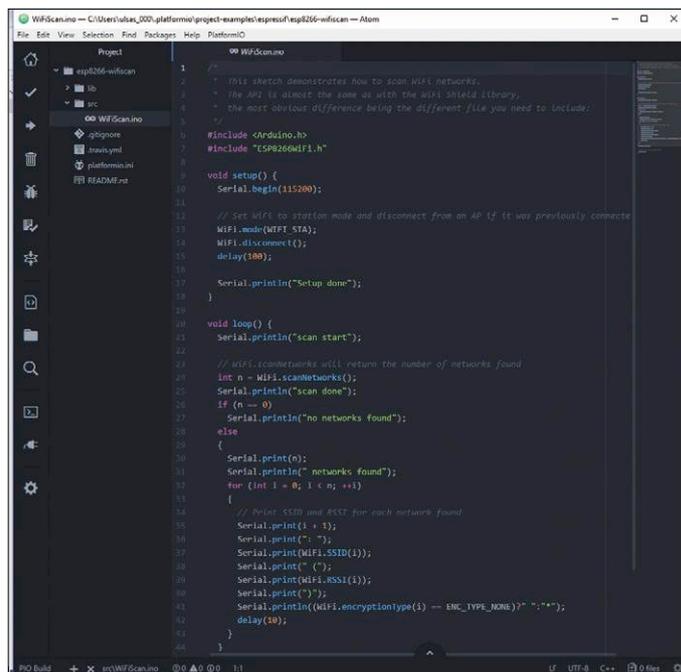


Fig. 10. Le croquis Arduino.

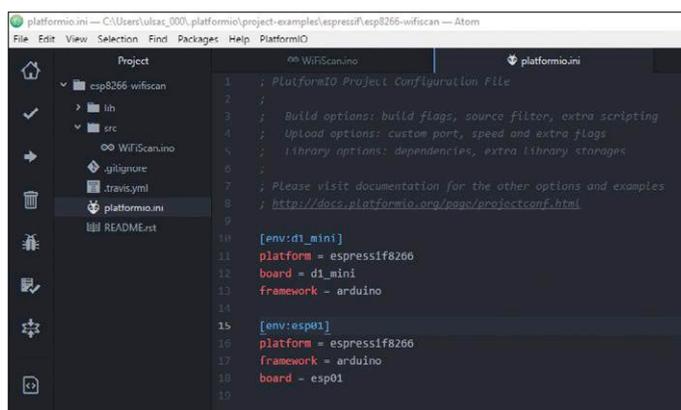


Fig. 11. La déclaration des cartes dans platformio.ini.

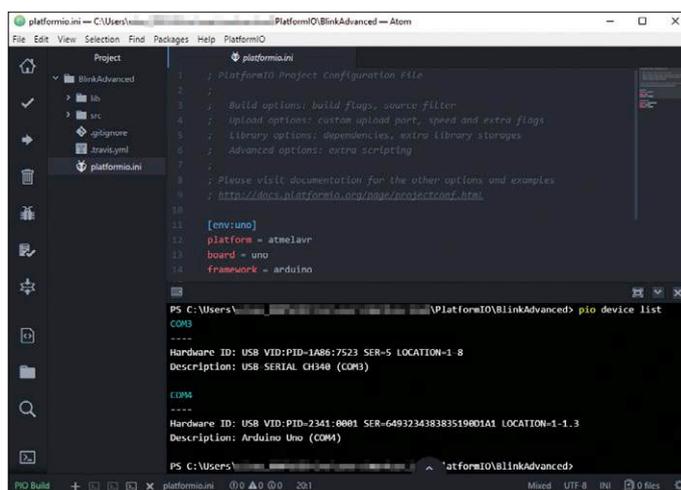
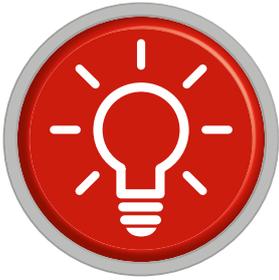


Fig. 12. Une commande simple permet d'afficher les ports utilisés par les cartes.



# trucs et astuces

## par les lecteurs pour les lecteurs

Encore une solution astucieuse qui facilite la vie des électroniciens.

### Condensateurs neufs pour écrans fatigués

Rolf Gerstendorf

Il y a bien des années, j'avais acquis à un prix très intéressant un écran avec une bonne dalle S-IPS (super in-plane switching) et un large spectre de couleurs. Alors même que l'usage de ce vieux moniteur 4:3 se limitait désormais au rôle d'écran secondaire, je fus consterné lorsque s'annonça le décès imminent de

Je découvris rapidement que ce défaut est apparemment typique et fréquent sur les écrans TFT (thin film transistor, à couche mince) quand les condensateurs électrolytiques low ESR (à faible résistance série équivalente) de l'alimentation rendent l'âme (fig. 1). À côté de nombreuses indications bien intentionnées, mais obscures, je tombai sur un site web baptisé Elko-Verkauf [1], exclu-



ce cher objet. Après la mise sous tension, l'écran s'éteignait par intermittence et même le voyant du bouton mécanique (!) de démarrage suivait ce rythme jusqu'à ce que l'électronique ait atteint sa température normale de fonctionnement. Au début, cela durait une minute, puis plusieurs minutes, puis de plus en plus longtemps, jour après jour.

On a malheureusement l'habitude de se débarrasser immédiatement d'un appareil électronique en panne et de chercher son remplaçant sur l'internet. Mais après une courte recherche, je dus me rendre à l'évidence : le prix de ce genre de moniteur graphique n'était pas piqué des hannetons. Avant d'investir dans une nouvelle acquisition un argent durement gagné, peut-être qu'une nouvelle recherche sur l'internet me fournirait de l'aide pour une réparation.

sivement consacré à ce thème. Le propriétaire, Andreas Grupe, de Hanovre, explique comment lui est venue l'idée de ce site :

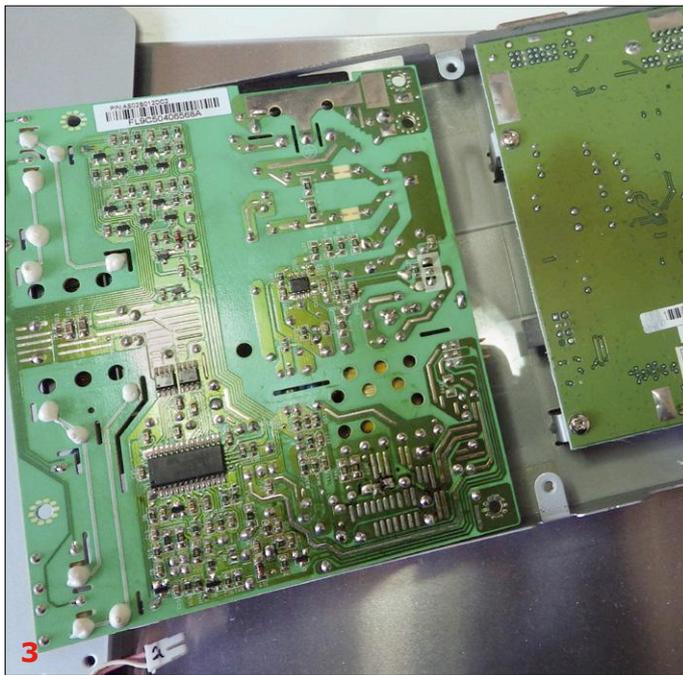
« Je m'efforce de faire durer les appareils électroniques aussi longtemps que possible afin de sauvegarder les ressources naturelles pour mes enfants (et ceux des autres, bien sûr)... Mon écran TFT de 15" de récupération avait une panne de condensateur. Pour la réparation, il me fallait des condensateurs low ESR et j'ai pensé que d'autres pourraient bien avoir besoin de ces condensateurs spéciaux pour sauver leurs appareils de la casse. C'est ainsi qu'est née, il y a quelques années, l'idée de Elko-Verkauf (boutique de vente de condensateurs électrolytiques). »

Une idée de commerce était née ! Le site web offre, classés de manière claire, des lots de condensateurs adaptés à des cen-

taines d'appareils, de la tireuse à bière au téléviseur plasma, en passant par la console de mixage de studio d'enregistrement, et à des prix raisonnables. De plus, il existe, pour la plupart des appareils, une notice de démontage illustrée, ce qui s'avère très utile, car il n'est pas si évident d'ouvrir un écran TFT sans casse. Et Andreas Grupe de poursuivre :

« Ce qui me tient le plus à cœur, c'est l'aide réciproque. Ce n'est qu'avec une collaboration assidue avec mes clients qu'il m'a été possible de constituer tous ces kits convoités, qui facilitent la vie à beaucoup. Ce n'est qu'ensemble qu'on peut constituer, à l'usage des nombreux « handicapés du condensateur », une base de connaissances assez riche pour une infinité d'appareils. »

Cela paraît assez digne de confiance et « politiquement correct ». Après une courte recherche, je découvris mon écran TFT et téléchargai sa notice de démontage moyennant deux euros. Même s'il me fallut bien constater que mon modèle d'écran différait quelque peu de celui de cette notice, grâce à elle, je pus le démonter d'une main sûre, sans casser d'ergot ou abîmer quelque autre pièce.



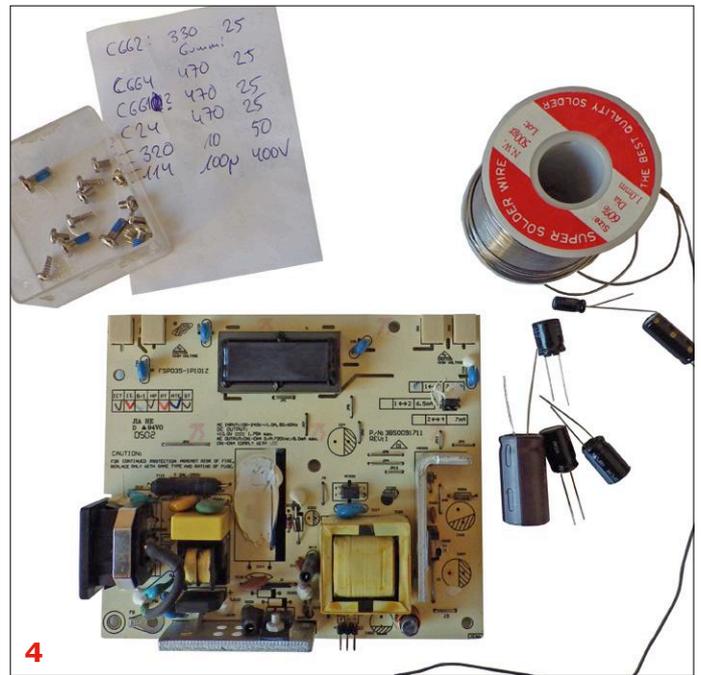
L'électronique était cachée derrière une plaque de tôle (fig. 2). Il fallait débrancher avec précaution un faisceau de câbles, opération pour laquelle les conseils de la notice de démontage de soigneusement repérer les connecteurs et les emplacements de connexion s'avérèrent particulièrement pertinents. On pouvait ensuite retourner la plaque de tôle. La carte d'alimentation (à gauche sur la figure 3, à côté de la « carte inverseuse ») fut facile à démonter et, débarrassée de toutes les vis, feuilles d'isolant et bouts d'adhésif, elle se retrouva bientôt devant moi, sur

ma paillasse. Comme les références des condensateurs suspects correspondaient à celles du kit, la commande partit illico et, deux jours plus tard, l'enveloppe matelassée contenant les six condensateurs se retrouva dans ma boîte aux lettres.

Je pouvais enfin m'occuper sérieusement de cette alimentation en panne. Pour commencer, il faut identifier les condensateurs et noter leurs numéros de référence et leurs valeurs (fig. 4). Quant à la polarité, la sérigraphie sur la carte me renseigna. S'il n'en était pas ainsi, on peut toujours la repérer sur la carte avec un marqueur pour transparents. Pas de problème pour retirer les vieux condensateurs, j'eus juste à faire très attention à deux CMS très proches de l'une des soudures.

Au bout d'un quart d'heure, les nouveaux condensateurs étaient en place, je pouvais procéder au remontage. Trois quarts d'heure plus tard, l'écran était réassemblé et je le rebranchai sur l'ordinateur. Fonctionnement impeccable ! Moralité : pour une dépense de 15 € et une heure de travail environ, j'avais récupéré un écran qui m'aurait coûté plusieurs centaines d'euros à remplacer. ◀

(160507 - version française : Helmut Müller)



[1] [www.elko-verkauf.de](http://www.elko-verkauf.de)

Vous avez trouvé une solution astucieuse pour un cas tordu ? Vous utilisez un composant ou un outil d'une manière inhabituelle ? Vous avez une idée pour résoudre un problème d'une manière plus simple ou meilleure que précédemment ? Écrivez-nous. Pour chaque astuce que nous publions, il y a 40 € à gagner.



# thermostat de bureau wifi

## surveillance de température flexible et programmable

Roy Aarts & Clemens Valens (labo d'Elektor)

Les thermostats sont partout – dans le frigo, le congélateur, la voiture, sur les murs – partout, sauf sur mon bureau. J'en voudrais un pour commander un ventilateur quand il fait trop chaud dans mon antre. Je pourrais aussi en avoir un pour couper la plaque chauffante quand le liquide que je réchauffe a atteint une certaine température, ou encore un pour démarrer le circulateur du chauffage quand la température chute.



### Caractéristiques

- 6 pré-réglages
- 3 programmes utilisateur
- connexion wifi
- contacts normalement ouvert (NO) et normalement fermé (NF)
- commute jusqu'à 2500 VA

Il me faut donc un thermostat autonome, portable et modulable, capable de commuter une charge. Si vous cherchez un peu sur l'internet, vous trouverez ce qu'on pourrait appeler des « thermostats de bureau » qui se rapprochent de cette description. Ces dispositifs avec des températures de démarrage et d'arrêt programmables sont capables de commuter un ventilateur ou une pompe. Malheureusement ils ne font que ça, sans la flexibilité que je recherche. Je décidais donc d'en concevoir un. Pour être honnête, j'ai sous-traité la

conception du thermostat à mon stagiaire préféré et l'ai laissé faire le plus dur. Pour rendre le projet plus intéressant, j'ai ajouté le wifi à la liste des spécifications. Pendant que j'étais occupé à d'autres tâches, notre stagiaire Roy travaillait tranquillement dans un coin du labo. Au bout de quelques semaines, il a levé les yeux et annoncé que le thermostat était prêt.

La vérification du travail de Roy a montré que son thermostat fonctionnait comme prévu – il avait vraiment fait un excellent

travail. Toutefois le thermostat n'était toujours pas pratique pour mon bureau et le logiciel n'était pas aussi souple que souhaité, il fallait revenir sur le projet. Comme Roy était entretemps parti vers de nouvelles aventures, j'ai dû m'y atteler. Le dispositif décrit ici est le résultat de ce processus en deux étapes.

### Spécifications et fonctions

Avant de plonger dans les détails techniques, voyons rapidement les spécifications du thermostat :

**Tableau 1. Les six pré-réglages du thermostat. La colonne « n° » se rapporte à la table d'index du logiciel, ce qui explique le nombre 0.**

n°	Démarrage	Arrêt
0	$T < T_{\text{BAS}}$	$T \geq T_{\text{BAS}}$
1	$T > T_{\text{HAUT}}$	$T \leq T_{\text{HAUT}}$
2	$T > T_{\text{BAS}}$ et $T < T_{\text{HAUT}}$	$T \leq T_{\text{BAS}}$ ou $T \geq T_{\text{HAUT}}$
3	$T < T_{\text{BAS}}$ ou $T > T_{\text{HAUT}}$	$T \geq T_{\text{BAS}}$ et $T \leq T_{\text{HAUT}}$
4	$T > T_{\text{HAUT}}$	$T < T_{\text{BAS}}$
5	$T < T_{\text{BAS}}$	$T > T_{\text{HAUT}}$

- Relais 250 V CA, 10 A (2500 VA) avec contacts normalement ouvert (NO) et normalement fermé (NF)
- 6 pré-réglages de commutation
- 3 programmes utilisateur
- Commande manuelle
- Interface utilisateur sur ordiphone, tablette ou ordinateur
- Connectivité au réseau wifi
- Connecteur micro-USB pour alimentation de 5 V

### Pré-réglages de commutation

Le micrologiciel comporte six « programmes » de commutation qui couvrent les scénarios les plus communs ainsi que certains plus inhabituels. Deux niveaux de température programmables  $T_{BAS}$  et  $T_{HAUT}$  déterminent les points de commutation de ces programmes. Voir les détails dans le **tableau 1**.

### Programmes personnalisés

Trois programmes personnalisés ont été prévus pour les situations non couvertes par les six pré-réglages principaux. Ils sont laissés vides dans le micrologiciel, charge à l'utilisateur de les remplir. Pour rendre cet exercice le moins pénible possible, l'interface entre la page web et le

micrologiciel a déjà été implémentée en réservant trois entrées dans la liste de pré-réglages de la page web : « Custom 1 », « Custom 2 » et « Custom 3 ». Vous pouvez bien sûr les changer, mais si vous les conservez vous n'aurez pas à réécrire d'HTML ni de Javascript et n'aurez pas à reprogrammer le système de fichiers de l'ESP-12F, travail qui prend du temps. Tout ce qu'il vous reste à faire est, dans l'EDI Arduino, d'ajouter un bout de code dans la boucle principale du programme de l'ESP.

### Le circuit

Au cœur du thermostat se trouve un module ESP-12F (**fig. 1**). Il est presque identique à l'ESP-12E, mais avec les broches 17 à 22 en plus (dont nous n'avons pas besoin). À l'intérieur de l'ESP-12F se trouve un ESP8266EX, une des rares puces industrielles sans titre ronflant (quelque chose comme *Système-sur-Puce à haut niveau d'intégration et capacité wifi* aurait été approprié), associée à 4 Mo de mémoire flash sérielle.

Le module ESP s'occupe de tout. Il lit les quatre boutons-poussoirs, commande les trois LED, communique avec le capteur de température et active ou relâche le relais. L'alimentation est bâtie autour du régulateur de tension IC1. D2 protège contre les inversions de polarité. Le module ESP consomme beaucoup de courant quand le wifi est actif, ce qui peut provoquer des interférences sur la ligne d'alimen-

### INFOS SUR LE PROJET

wifi

thermostat ESP826

Arduino DS18B20

débutant

➔ **connaisseur**

expert

env. 4 h

EDI Arduino, convertisseur série-USB, soudage CMS, perceuse

env. 50 €

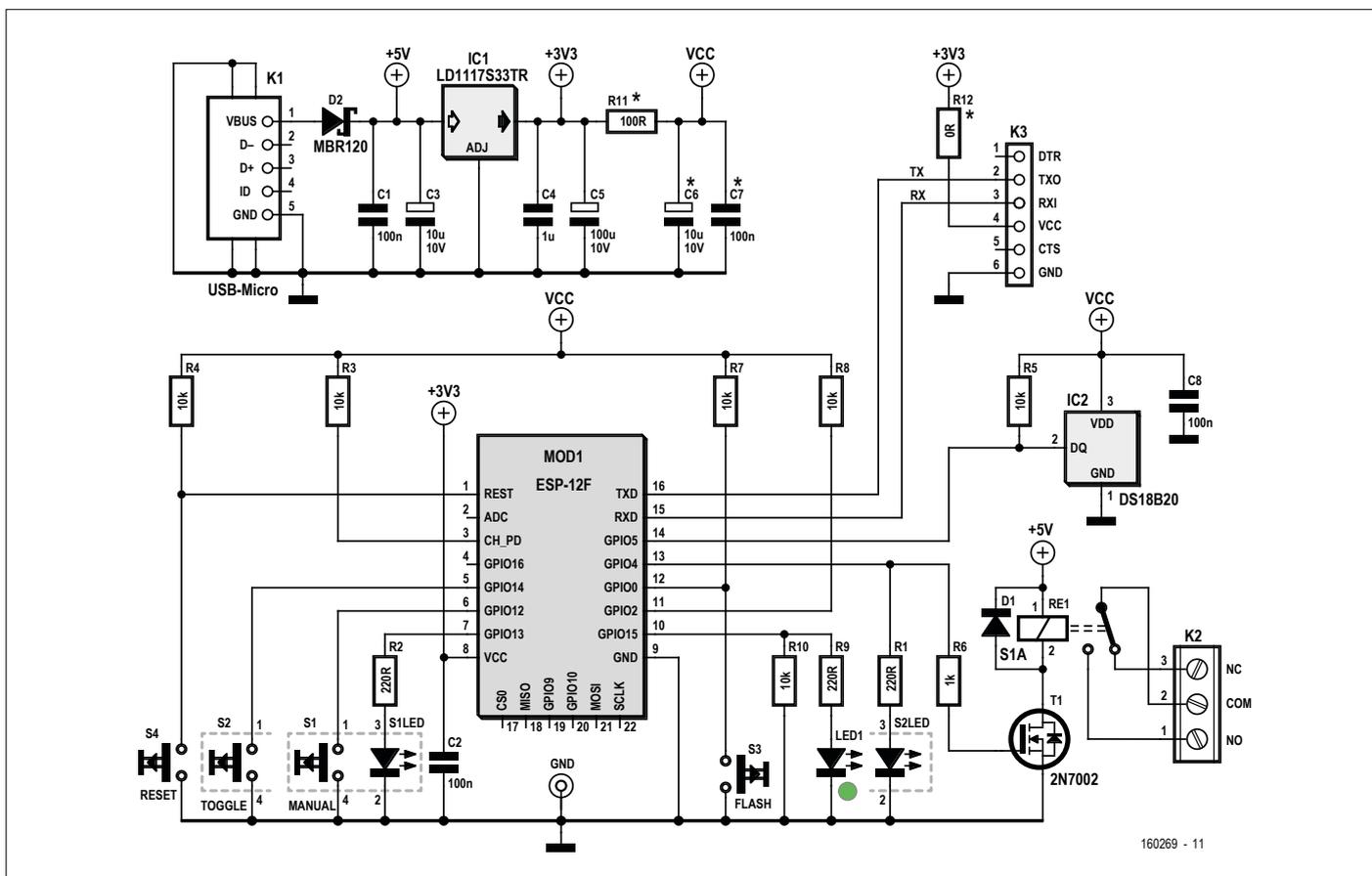


Figure 1. Le cœur du thermostat de bureau n'est pas un Arduino, mais MOD1, un ESP-12F.



Figure 2. Le circuit imprimé a été conçu pour être facile à monter dans un boîtier bon marché.

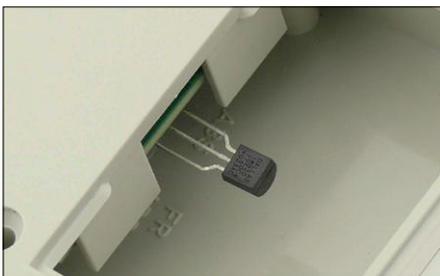


Figure 3. Le compartiment à piles est largement assez grand pour accueillir le capteur de température, même s'il est connecté à la carte avec de longs fils.

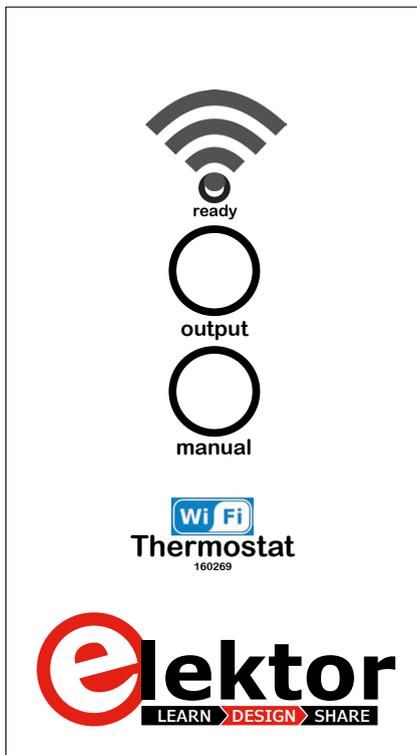


Figure 4. Suggestion de présentation pour la face avant.

## Boutons-poussoirs

Seuls deux des quatre boutons-poussoirs de la carte, S1 et S2, sont disponibles pour l'utilisateur. S1 permet de basculer entre les modes Manuel et Thermostat. En mode Manuel, le dispositif agit comme un simple interrupteur, utilisé par ex. au démarrage ou en situation d'urgence. Dans ce mode, l'état du relais est basculé avec S2.

Les boutons S3 et S4 servent à mettre l'ESP en mode de programmation au moyen d'un protocole secret :

1. Presser S4 (*Reset*) ;
2. Maintenir S4 et presser S3 (*Flash*) ;
3. Relâcher S4 (*Reset*) ;
4. Relâcher S3 (*Flash*).

Si la séquence est exécutée correctement, la petite LED bleue du module ESP va clignoter brièvement, indiquant que le module peut maintenant être programmé par son port série.

## LED

La LED1 indique l'état de la connexion wifi. Lorsqu'elle clignote, le thermostat n'est pas connecté ; lorsqu'elle est fixe, le thermostat est connecté. Deux possibilités dans ce cas :

1. Le thermostat est lui-même un point d'accès (AP) et un client (ordiphone, ordinateur, etc.) lui est connecté ;
2. Le thermostat est connecté à un point d'accès (AP).

Les boutons-poussoirs lumineux S1 et S2 ont des LED intégrées. Quand la LED de S1 est allumée, le thermostat est en mode Manuel ; quand elle est éteinte, il est en mode Thermostat. La LED de S2 indique l'état du relais. Allumée, le relais est activé (le contact NO est fermé) ; éteinte, le relais est relâché (le contact NF est fermé).

## Boîtier et assemblage

La carte a été conçue pour s'intégrer dans un boîtier en ABS bon marché avec compartiment à piles, un modèle 1593Qx de Hammond Manufacturing (**fig. 2**). Ce boîtier de 112 × 66 × 28 mm est disponible en gris (x = GY), noir (x = BK) et bleu translucide (x = TBU). Quand le capteur de température est monté sur le CI, il peut être disposé de façon à ce que sa tête pende dans le compartiment à piles. Avec le couvercle du compartiment retiré, l'air peut circuler autour (**fig. 3**).

tation. Le filtre constitué par R11, C6 et C7 garantit qu'IC2 en particulier voit une tension d'alimentation relativement propre.

K3 est le connecteur du port série, compatible avec un câble convertisseur série-USB « FTDI » en 3,3 V. Notez que la ligne VCC de ces câbles est en 5 V, même si les signaux sont en 3,3 V, raison pour laquelle R12 n'est pas montée par défaut. Elle est toutefois disponible, au cas où le port série utiliserait un autre circuit tirant son alimentation du thermostat (ou l'inverse).

Nous avons retenu comme capteur de température le fameux thermomètre numérique 1-Wire DS18B20 en boîtier TO-92. On peut le monter sur le circuit imprimé ou le connecter avec des fils pour plus de liberté.

Le relais a une empreinte standard disponible chez plusieurs fournisseurs. Toutefois avant d'en acheter un adapté à la carte, vérifiez que ses contacts supportent la charge qu'ils doivent commuter. La bobine doit être en 5 V et, pour commuter les 2500 VA spécifiés, il faut des contacts prévus pour un courant continu (et non maximum) d'au moins 10 A à 250 V CA. Les photos de cet article montrent une version antérieure de la carte, avec un relais 5 A bien plus petit. Pour commuter une puissance telle que 2500 VA, les pistes du CI entre les contacts du relais et le connecteur K2 font 2,54 mm de large. Avec une épaisseur de cuivre standard de 35 µm cela autorise environ 5 A tout en restant raisonnablement froid. Comme nous en voulons plus, ces traces n'ont pas de masque de soudure et doivent être renforcées d'une couche de soudure. Pas besoin de beaucoup – 0,5 mm suffiront.

Si le capteur est relié par des fils, le compartiment à piles est suffisamment spacieux pour stocker le capteur quand le thermostat n'est pas utilisé.

Un gabarit de perçage est disponible [1] avec un modèle de face avant (fig. 4) pour vous aider à réaliser un projet bien fini. Il n'y a que trois trous à percer.

La profondeur spécifiée du boîtier est de 21,7 mm, les entretoises font 4,5 mm. La hauteur du relais est de 15,7 mm. Avec un CI standard de 1,6 mm d'épaisseur, cela fait 0,1 mm de trop. On peut régler ça en raccourcissant un peu les entretoises ou avec un CI plus fin (1 mm par exemple). De plus, il y a quatre entretoises sur le couvercle supérieur, dont une gêne le relais et doit donc être retirée. C'est possible avec un foret de 10 mm ou une lame tranchante.

Après le montage du relais, n'oubliez pas de recouvrir d'une couche de soudure de 0,5 mm d'épaisseur (ou plus) les pistes qui relient le relais à K2. Il faudra peut-être raccourcir l'entretoise sous le relais pour compenser ce surplus de soudure.

Il existe des extensions (5 et 8 mm) pour les boutons-poussoirs. Avec celles de 5 mm, les boutons dépassent juste du couvercle du boîtier.

Le montage de la LED demande quelques ajustements manuels. Pour faire ça proprement, soudez d'abord l'anode – la plus facile à souder/dessouder – et ajustez la hauteur à votre convenance. Soudez alors l'autre broche à la masse.

Le boîtier a un panneau latéral amovible qui donne accès aux connecteurs. Comme nous travaillons avec des tensions de ligne CA, il est **clairement et fortement** déconseillé de le supprimer. Le gabarit de fraisage a aussi un plan des découpes nécessaires pour ce petit panneau. Certes c'est petit, mais faisable avec un minimum d'effort pour assurer la sécurité du thermostat.

Pour l'embase USB, nous avons trouvé qu'une découpe de la taille de l'extrémité de la fiche était suffisante pour procurer un bon contact. Si ce n'était pas le cas, sachez qu'il existe dans le commerce des câbles de chargeur micro-USB avec embouts extralongs (8 mm au lieu de 5).

## Logiciel

Le programme pour le thermostat est écrit comme un croquis Arduino avec les fonctions `setup` et `loop`. C'est un programme volumineux, car il intègre

un serveur web et un point d'accès wifi, mais presque tout ceci peut être assimilé à des boîtes noires. L'application « thermostat » est implémentée dans la fonction `loop` où vous pouvez ajouter des préreglages personnalisés. C'est vraiment sans difficulté. Tout est téléchargeable depuis la page web de l'article [1].

## Les requêtes GET

La communication entre le programme principal et la page web qui sert d'interface utilisateur au périphérique distant est plus difficile à comprendre. Le **figure 5** illustre graphiquement le mécanisme. Il est important de comprendre ici que le thermostat est un serveur web qui répond à ce que l'on nomme des requêtes HTTP



## LISTE DES COMPOSANTS

### Résistances

CMS 0805, 0,1 W  
 R12 = 0 Ω  
 R11 = 100 Ω  
 R1, R2, R9 = 220 Ω  
 R6 = 1 kΩ  
 R3, R4, R5, R7, R8, R10 = 10 kΩ

### Condensateurs

CMS 0805  
 C1, C2, C7, C8 = 100 nF  
 C4 = 1 μF  
 C3, C6 = 10 μF, boîtier A  
 C5 = 100 μF, 16 V, boîtier C

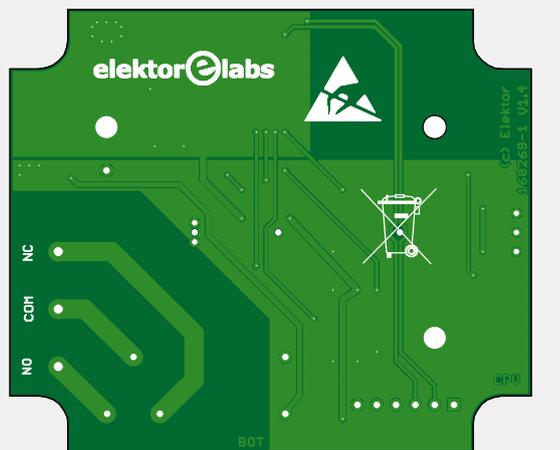
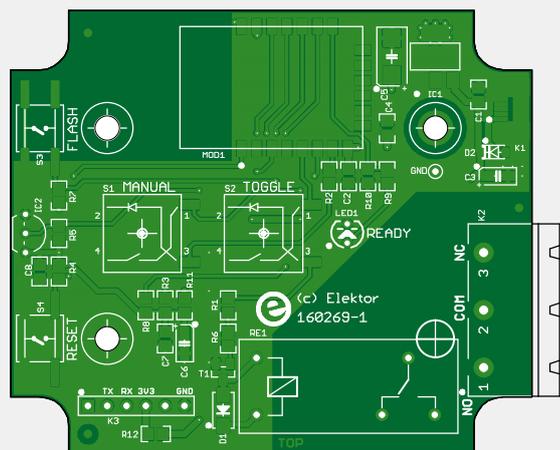
### Semi-conducteurs

D1 = S1A  
 D2 = MBR120LSFT1G  
 IC1 = LD1117S33TR  
 IC2 = DS18B20  
 LED1 = LED verte, 3 mm  
 T1 = 2N7002

### Divers

K1 = connecteur micro-USB de type B  
 K2 = bornier à vis à 3 contacts au pas de 7,6 mm  
 K3 = barrette mâle à 6 contacts au pas de 2,54 mm  
 MOD1 = ESP-12F  
 RE1 = relais SPDT, bobine de 5 V, contacts 10 A/250 V CA (Omron G2RL-1 par ex.)  
 S3, S4 = commutateur tactile, 6×6 mm, bornes en aile de mouette (*gullwing*)  
 S1, S2 = commutateur tactile, avec LED intégrée, Multimec 5GSH93542SMD  
 Capuchon pour S1 & S2, Multimec 1US16  
 Rehausseur pour S1 & S2, 5 mm, Multimec 2SS09-5.0  
 Boîtier : Hammond de type 1593Qx (x = GY/BK/TBU)

Circuit imprimé, réf. 160269-1



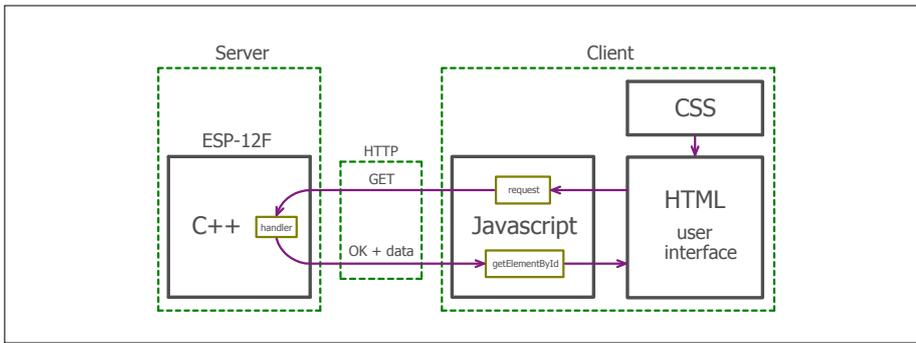


Figure 5. Le thermostat est le serveur tandis que le client est le téléphone, la tablette ou l'ordinateur qui affiche la page web. Quatre langages de programmation et de script sont utilisés pour les faire tous communiquer.

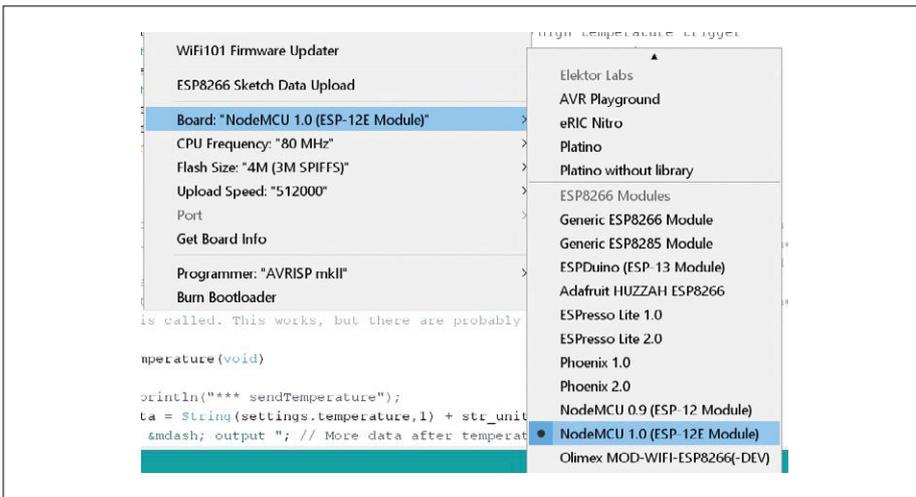


Figure 6. Après installation du paquet de cartes adéquat et de l'outil de programmation des données, l'EDI Arduino devrait reconnaître le module ESP-12E/F et afficher « ESP8266 Sketch Data Upload » dans son menu « Outils ».

GET. De telles requêtes attendent une réponse, un fichier HTML ou une image par exemple, mais ce peut être aussi des données. Au lancement, la fonction C++ `setupWebserver` associe les requêtes GET prises en charge à des fonctions (*handlers*) au sein du programme C++. Par exemple lorsqu'une requête GET pour « curtemp » – la température courante – est reçue, la fonction C++ `sendTemperature` est appelée. Pour savoir comment ça se passe exactement, il faudrait fouiller les bibliothèques de l'ESP8266 utilisées par le code du serveur web. La fonction C++ `sendTemperature` collecte et formate les données requises et les envoie au serveur web qui à son tour les envoie au client d'où les requêtes GET sont issues.

### Couche Javascript

Comme le langage HTML est un peu limité et n'aurait pas permis à lui tout seul tout ce que nous avons en tête pour

l'interface utilisateur, nous avons ajouté une couche Javascript. Cette couche crée les requêtes GET voulues, décompacte et distribue les données reçues en réponse. Pour continuer notre exemple, quand la page web est chargée, un temporisateur est lancé pour demander toutes les 5 s la température mesurée par le thermostat. Ce temporisateur exécuté par le navigateur qui affiche la page, appelle la fonction Javascript `request` pour mettre en forme et envoyer la requête GET « curtemp ». Si une réponse est reçue dans un intervalle donné, la fonction Javascript `showTemperature` est appelée avec comme arguments les données reçues. Cette fonction décompacte les données et met à jour la page HTML, et l'utilisateur voit s'afficher la température.

Un modèle CSS ou feuille de style est utilisé pour rendre l'interface utilisateur agréable.

### Configurer l'environnement de développement du logiciel

Le programme du thermostat a été écrit avec l'EDI Arduino. Comme l'ESP-12F n'est pas un Arduino, un « paquet » de cartes doit être ajouté à l'EDI pour la prise en charge de l'ESP8266. Pour cela la procédure est assez simple, mais requiert une connexion à l'internet. Elle part du menu « Fichiers → Préférences » de l'EDI → Arduino. Copiez l'URL [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json) (une ligne, sans espace, attention aux erreurs de saisie) dans la boîte « URL de gestionnaire de cartes supplémentaires » de la fenêtre « Préférences ». Refermez la fenêtre. Ouvrez le Gestionnaire de carte (Outils → Type de carte). En haut à gauche de la fenêtre qui s'ouvre sélectionnez « Contribué ». Recherchez « esp8266 by ESP8266 Community » dans la liste qui apparaît, cliquez dessus puis cliquez sur le bouton « Installer ». L'EDI va télécharger les fichiers requis et les copier au bon endroit. Fermez alors la fenêtre. Maintenant, quelque part dans le menu « Type de carte », sous le titre « ESP8266 modules » on va trouver « NodeMCU 1.0 (ESP-12E Module) » ; sélectionnez-le. Choisissez 80 MHz pour « CPU Frequency » et « 4M (3M SPIFFS) » pour « Flash Size ». Choisissez « Upload Speed » aussi haute que possible pour réduire le temps de programmation. La vitesse maximale utilisable dépend du convertisseur série-USB utilisé.

### Outil de chargement du système de fichiers

Après installation du paquet de cartes, il faut installer une seconde extension de l'EDI, nécessaire pour charger la page web dans la mémoire flash série du module ESP. La procédure est aussi simple que stocker un fichier dans un nouveau dossier. Le fichier en question (`esp8266fs.jar`) ainsi que les instructions détaillées pour le mettre à son emplacement (`<...>\sketchbook\tools\ESP8266FS\tool\`) se trouvent à [3]. Ceci fait (redémarrez l'EDI si vous l'aviez laissé ouvert pendant l'installation), vous devriez avoir une entrée « ESP8266 Sketch Data Upload » dans le menu « Outils » de l'EDI (fig. 6).

### Bibliothèque supplémentaire

Enfin il faut installer une bibliothèque 1-Wire pour communiquer avec le capteur DS18B20. On la trouve en [4]

et on l'installe avec le gestionnaire de bibliothèque de l'EDI (Croquis → Inclure une bibliothèque → Gérer les bibliothèques...).

### Programmer le micrologiciel

Connectez le thermostat à l'ordinateur avec un convertisseur série-USB. Branchez aussi le connecteur micro-USB du thermostat à une alimentation 5 V. Téléchargez le croquis et les données du thermostat [1] et ouvrez-les dans l'EDI Arduino. Utilisez le protocole d'activation de la programmation (cf. ci-dessus : appuyez sur S4, puis S3, relâchez S4 puis S3). Sélectionnez la carte et le port série, et lancez le téléchargement du croquis. Le croquis devrait être compilé et téléchargé sans erreurs ni alertes. L'étape suivante consiste à télécharger la page web et tout ce qui se trouve dans le sous-dossier « data » du croquis : utilisez de nouveau le protocole d'activation de la programmation puis sélectionnez « ESP8266 Sketch Data Upload » depuis le menu « Outils » de l'EDI. Le téléchargement devrait démarrer et prendre un moment (plusieurs minutes), car l'ensemble pèse 3 Mo.

### Utilisation

Maintenant que le micrologiciel et la page web sont installés, il est temps d'essayer. Le premier essai consiste à allumer le thermostat et à tester le bouton « Manuel » (S1). Appuyer dessus devrait basculer sa LED intégrée. Appuyez dessus pour allumer sa LED. Puis appuyez sur le bouton « Sortie » (S2) pour basculer sa LED intégrée. Ceci devrait aussi basculer le relais, ce qui s'entend. Pendant ce temps, LED1 devrait clignoter à une fréquence de 0,5 Hz.

### Le wifi

Ouvrez les réglages wifi d'un ordiphone, une tablette ou un ordinateur et cherchez le réseau « *Wi-Fi Thermostat* ». Avec un téléphone, il est recommandé de couper les données mobiles d'abord.

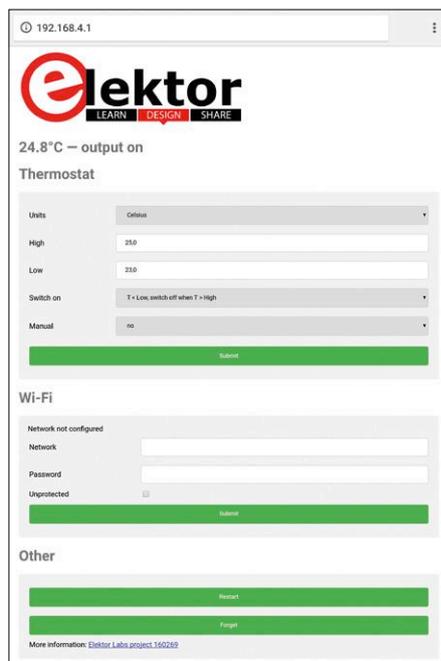


Figure 7. La page web servie par le thermostat (à la fois en modes point d'accès et client) montre si le dispositif est connecté à un réseau ou non. S'il l'est, son adresse IP figure dans la section wifi à la place de la mention « *Network not configured* ».

Se connecter (pas de mot de passe, LED1 devrait s'arrêter de clignoter et rester allumée), puis ouvrez un navigateur et allez à l'adresse 192.168.4.1. La page web du thermostat devrait apparaître (fig. 7). Attendez cinq secondes pour lui permettre de recevoir et mettre à jour la température correcte. S'il le fait, tout va bien et vous pouvez changer les réglages. Cliquez (ou tapotez) sur « *Submit* » pour envoyer les nouveaux réglages au thermostat. Cela peut prendre jusqu'à cinq secondes pour que la page web affiche les changements, alors soyez patient.

### Connexion à un réseau existant

Une section de la page web permet de configurer le thermostat pour un réseau wifi existant. Entrez son SSID et mot de passe et appuyez sur le bouton « *Submit* » correspondant. Vous devrez alors

appuyer sur le bouton « *Restart* » pour reprendre les changements effectifs. Notez que le premier redémarrage avec ce bouton après la programmation du module ESP provoquera un plantage. Ce n'est pas de notre faute, mais une anomalie connue quelque part dans le module ESP et/ou les bibliothèques. Si cela arrivait, redémarrez simplement le dispositif et tout devrait bien se passer. Si, après un redémarrage, LED1 reste allumée en permanence, le thermostat s'est connecté avec succès au réseau (pour en être sûr, déconnectez d'abord tous les dispositifs qui peuvent s'être connectés au point d'accès du thermostat). Pour trouver l'adresse IP du thermostat, connectez-vous à son point d'accès. La page web l'affichera. Vous pouvez aussi consulter la liste des dispositifs connectés à votre réseau.

### Effacement

Pour supprimer le thermostat du réseau, appuyez sur le bouton « *Forget* ». Ceci effacera la mémoire persistante du dispositif dans laquelle sont stockés les réglages. Tous ces réglages seront réinitialisés à leurs valeurs par défaut d'usine. Appuyez sur le bouton « *Restart* » pour redémarrer le thermostat.

### Que faire ensuite ?

Bien que cet article traite d'un thermostat, rien ne vous empêche d'en faire autre chose. Ce que nous avons ici est simplement un relais commandé par wifi appliqué à une surveillance de température. Il suffit, évidemment, de remplacer le capteur de température par un autre capteur pour transformer le thermostat en quelque chose de complètement différent. Le logiciel est facile à modifier – toutes les parties difficiles sont prémâchées – et il peut être adapté à plein d'autres applications. Laissez libre cours à votre imagination, pour reprendre le cliché, et prenez cette étude comme point de départ d'un nouveau projet. ◀

(160269 – version française : Denis Lafourcade)

#### Liens

- [1] Site web du projet : [www.elektormagazine.fr/160269](http://www.elektormagazine.fr/160269)
- [2] Paquet des cartes ESP8266 : [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)
- [3] Outil de chargement du système de fichiers ESP8266FS : <https://github.com/esp8266/Arduino/blob/d7044eceab4d4453e4d73ac49dcbbc8b8d0c9eb1/doc/filesystem.rst>
- [4] Bibliothèque Arduino 1-Wire : <https://github.com/PaulStoffregen/OneWire>





# enova

Salon de l'innovation en électronique,  
mesure, vision et optique

**LYON**  
Cité Centre  
de Congrès

**7-8**  
**FÉVRIER**  
**2018**

**PRENEZ DE L'AVANCE**  
**SUR LE FUTUR**

Thèmes à l'honneur

◆ **AUTOMOBILE**  
◆ **MÉDICAL**

Votre badge gratuit sur [www.enova-event.com](http://www.enova-event.com)



**enova**  
2018 >

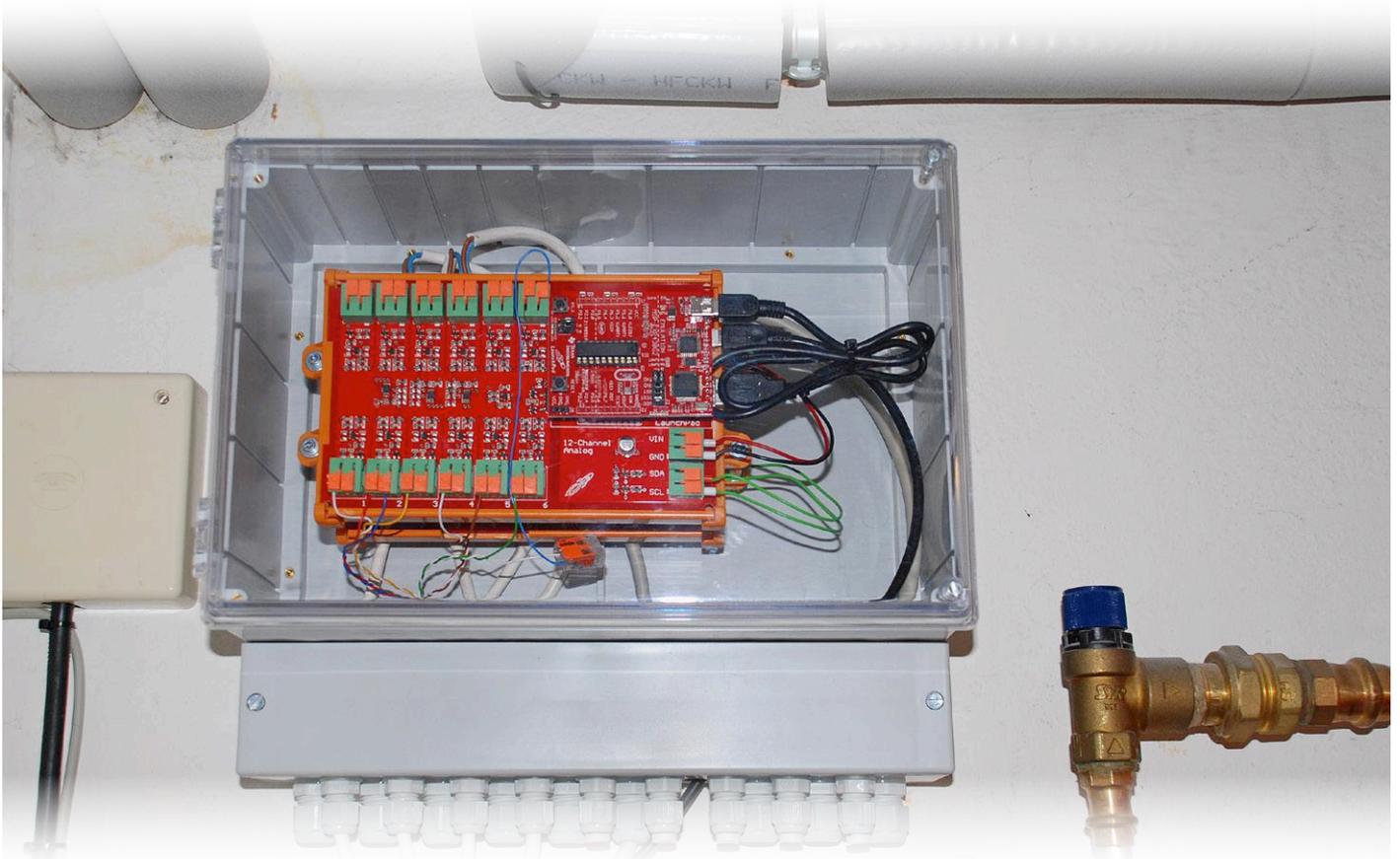
**LYON**  
7-8 FÉVRIER

**TOULOUSE**  
30-31 MAI

**PARIS**  
23-24 OCTOBRE



# acquisition de mesures par USB en quête du chauffage optimal



**Matthias Ulmann**, Texas Instruments

L'optimisation d'une installation de chauffage requiert le suivi continu et précis de différents paramètres. Seules ces données permettent d'introduire rapidement des changements même minimes. L'enregistreur de données décrit ici emmagasine une grande quantité de paramètres analogiques et numériques sur un port de PC à travers une séparation galvanique sûre.

Les chauffages modernes sont dotés de nombreux réglages pour d'innombrables paramètres. Mais il n'est pas évident ni directement vérifiable de trouver qui fait quoi dans le système. En outre, le chauffage réagit très lentement, le test de différents réglages pour trouver les valeurs optimales des paramètres prend donc du temps. Pour améliorer vraiment le fonctionnement, il n'est pas humainement possible de surveiller sans arrêt les différents paramètres comme les variations de température dans le ballon d'eau chaude, l'état actuel des pompes, des vannes et éventuellement du brûleur de fioul. Si l'on recueillait ces données, on verrait tout de suite les effets

de changements, même infimes.

Il est vrai que des appareils de chauffe déjà anciens sont équipés d'une commande par interface de bus, mais rares sont ceux dont le protocole et le logiciel correspondant sont accessibles à l'utilisateur. Le raccordement d'un PC à l'électronique de la chaudière est encore plus rare.

C'est ce qui m'a conduit à prélever directement les données brutes des capteurs analogiques de température du chauffage et l'état des périphériques qui en dépendent. Il existe bien dans le commerce des modules de mesure et des cartes enfichables pour ces tâches, mais leurs canaux sont trop peu

nombreux ou bien l'ensemble coûte beaucoup trop cher. Mieux vaut développer son propre système de mesure pour acquérir des signaux analogiques et des paramètres numériques, les traiter et les envoyer sur un PC par USB.

L'ampleur du projet dépend forcément de la complexité de votre installation et de la nécessité d'une optimisation. Pour que le circuit soit assez généraliste et que la dépense soit proportionnée au besoin, pas de système tentaculaire, l'approche est ici totalement modulaire. Tout y est pourvu d'un isolement galvanique en amont du transfert vers un système à micro-contrôleur ou un PC, voire les deux.

### Comment ça marche ?

Le système de mesure nécessite deux (peut-être trois) interfaces différentes entre les paramètres d'entrée et l'ordinateur.

- On peut utiliser comme capteurs de température différents types standard analogiques (KVT20, KTY10-6, Ni1000), linéarisés par résistance série et alimentés sous 5 V. Ces capteurs demandent des entrées de mesure dans le domaine de 0 à 5 V.
- Côté numérique, pour détecter l'état du brûleur de fioul, du circulateur ou de la pompe solaire, il faut savoir s'ils sont alimentés en électricité ou pas en mesurant la tension sur les bornes correspondantes. Auquel cas, il faut bien sûr prendre les dispositions de sécurité appropriées.
- Il convient aussi de prévoir des canaux supplémentaires pour des usages ultérieurs avec des tensions d'entrée entre 5 et 24 V=, question d'éviter tout embarras. Il s'agit alors de canaux pour signaux purement numériques, donc pas pour la mesure de signaux analogiques.

Un microcontrôleur sera chargé de relever systématiquement tous les canaux et de transmettre leur valeur, analogique ou numérique, au PC par USB. C'est là que s'effectueront la conversion et la présentation graphique des mesures.

Ce procédé offre plusieurs avantages sur d'autres solutions. L'enregistrement reste universel et applicable à d'autres systèmes et à d'autres capteurs sans demander de modification du micrologiciel. Lors de la conception du circuit, il y a lieu de mettre l'accent sur la robustesse et la sécurité, puisqu'il est installé à demeure et doit pouvoir fonctionner longtemps sans souci.

### Entrées numériques

J'ai prévu deux sortes de canaux numériques, l'une pour la tension du secteur (230 V~), l'autre pour la plage de tension 5 à 24 V=. Dans l'une comme l'autre, c'est un photocoupleur du type CNY17F-3 qui assure la séparation électrique. Il y a aussi huit entrées numériques qui sont reliées par une puce d'extension numérique à un bus I<sup>2</sup>C.

Pour éviter l'usage d'autres composants ou de transformateurs, par exemple, les entrées numériques pour la tension du secteur sont conçues de manière à pouvoir être aisément branchées en parallèle sur la charge. La **figure 1** montre que quatre résistances de 100 kΩ au format 1206 limitent à 1 mA le courant dans le photocoupleur. Elles sont en série pour diviser la tension à leurs bornes, ce qui évite en tout cas un dépassement de la limite de rigidité diélectrique de chacune d'elles. La diode tête-bêche sur la LED du photocoupleur permet le passage du courant lors de l'alternance négative de la tension

## Caractéristiques techniques

- jusqu'à 64 entrées numériques
- 12 entrées analogiques
- isolement galvanique des entrées de capteurs
- isolement galvanique de l'alimentation
- traitement des données par carte LaunchPad à prix modique
- liaison USB avec le PC
- représentation graphique des mesures sur PC

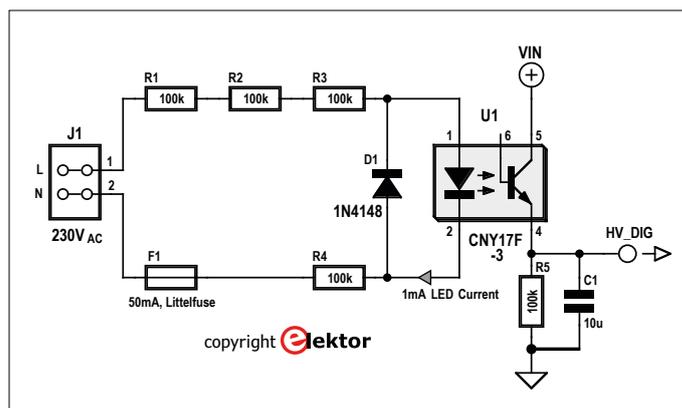


Figure 1. Schéma de raccordement des entrées numériques pour la tension du secteur à 230 V~.

du secteur pour protéger la LED. On trouve encore un fusible de 50 mA qui supporte 250 V pour protéger l'entrée. Du côté de la sortie, le photocoupleur transmet la tension VIN (+5 V) lorsqu'on applique un signal sur l'entrée. Comme le photocoupleur ne commute que pendant l'alternance positive, on a prévu un condensateur de 10 μF comme réservoir pour maintenir une tension suffisante pendant l'alternance négative. En l'absence de tension à l'entrée, le condensateur se décharge dans la résistance de 100 kΩ. Le retard que cette combinaison entraîne, des dizaines de millisecondes à la mise sous tension et des centaines au déclenchement sont tout à fait acceptables dans une application lente comme celle-ci. Le photocoupleur garantit une séparation fiable, 5 000 V<sub>eff</sub> entre l'entrée et la sortie. Toutefois, lorsqu'on dessine le circuit imprimé, il faut respecter les lignes de fuite et les distances d'isolement nécessaires et prescrites. Cela s'applique également entre les différents canaux.

Le circuit des canaux pour les entrées dans la gamme de 5 à 24 V= est un rien plus compliqué à cause de l'étendue de sa plage. Une réduction du courant de LED par résistance série ne suffit plus, il faut inclure une limitation active à courant constant de 1 mA. À la **figure 2**, on voit la source de courant d'un canal, insensible à la grandeur de la tension appliquée. Comme protection, on voit aussi un barrage contre l'inversion de polarité et une diode de suppression des transitoires (Transil). La sortie est comparable à celle de la version à 230 V~, mais avec un retardement nettement moindre. Comme il n'y a pas de tension alternative à l'entrée, un condensateur réservoir de grande capacité est inutile.

L'extenseur d'E/S PCA9539 (**figure 3**) sert à relier les seize canaux numériques au bus I<sup>2</sup>C. Comme on peut lui conférer

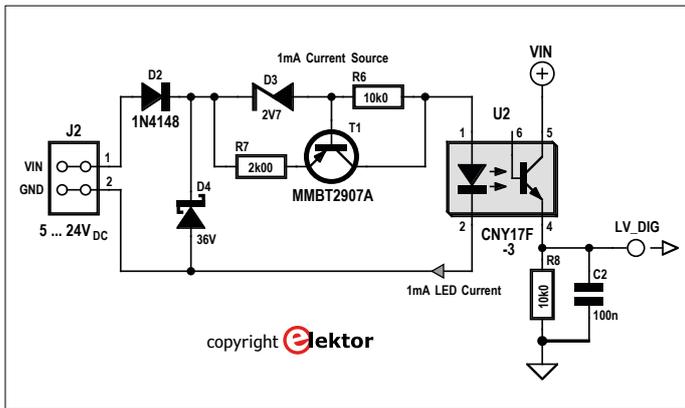


Figure 2. Schéma de raccordement des entrées numériques pour des tensions continues entre 5 et 24 V=.

quatre adresses différentes au moyen des lignes A0 et A1, il peut relever jusqu'à 64 entrées numériques. Elles devraient suffire aux installations de chauffage les plus développées !

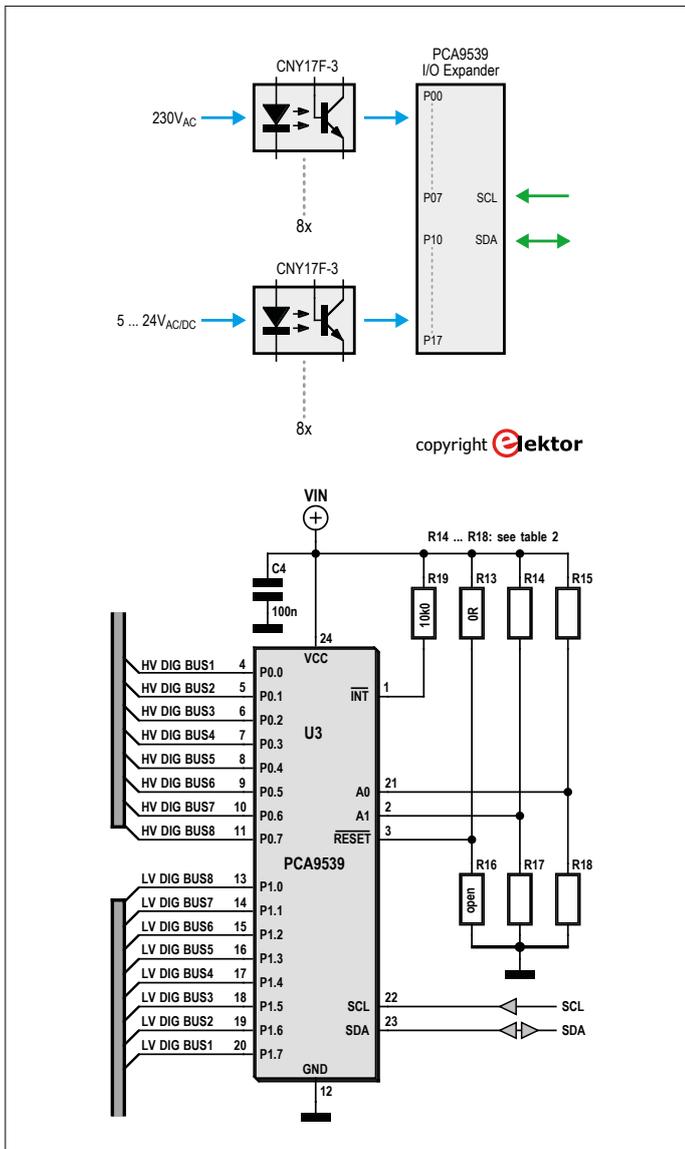


Figure 3. Vue d'ensemble des entrées numériques et raccordement au bus I<sup>2</sup>C.

## Entrées analogiques

Les entrées analogiques ont une masse commune et sont de construction identique. Le domaine d'entrée du convertisseur A/N à 12 bits ADS1015 avec interface I<sup>2</sup>C s'étend de 0 à 4,096 V et peut mesurer à tour de rôle quatre entrées asymétriques. Chaque canal dispose à l'entrée d'un amplificateur opérationnel pour amplifier le signal si nécessaire (figure 4).

Le signal analogique est d'abord débarrassé des parasites HF par le filtre passe-bas R9/C4. Ensuite on trouve la double diode D1 qui « verrouille » la tension d'entrée à +5 V ou la masse, ce qui protège l'ampli opérationnel d'une éventuelle surtension. Il importe que ce composant se trouve *derrière* le filtre et la résistance d'entrée pour que le courant soit limité par les diodes. Si les diodes de verrouillage étaient montées directement à l'entrée, le courant ne serait pas écrêté et les diodes risqueraient le claquage thermique.

Comme amplificateur, c'est un amplificateur opérationnel du type OPA335 qui a été choisi. La haute impédance de son entrée CMOS ne charge que très peu les capteurs. Il se contente d'une seule tension d'alimentation (donc asymétrique) et se distingue par une dérive en température extrêmement faible de 0,05 μV/K et stable à long terme, il convient parfaitement pour des mesures précises de toutes sortes.

Le circuit et le tracé des pistes permettent de configurer l'ampli op comme tampon ou comme amplificateur non-inverseur. Dans le premier cas, on n'installe pas la résistance R11 et on remplace R10 par un pont ou une résistance nulle. Dans le second cas, les résistances R10 et R11 déterminent le gain  $v = 1 + (R10/R11)$ . Quant à R6, on ne l'installe dans aucun de ces deux cas.

Si aucun amplificateur ni étage tampon n'est nécessaire, on peut aussi renoncer à l'ampli opérationnel et appliquer le signal analogique via R6, comme pont de 0 Ω, directement à l'entrée du convertisseur A/N.

## Circuits imprimés

Nous sommes en présence de trois sortes de circuits d'entrée à reproduire en plusieurs exemplaires, je les ai rassemblés sur deux circuits imprimés.

Sur la carte analogique, il y a trois convertisseurs A/N adressables. Le convertisseur ADS1015 (figure 5) dispose de quatre adresses différentes. Et pourtant, à y regarder de plus près, il n'a qu'une seule ligne d'adresse, ADDR ! Comment sélectionner jusqu'à quatre convertisseurs ? On peut comme d'habitude relier cette ligne à GND ou VDD, mais la puce peut prendre deux autres adresses en reliant cette ligne à SCL ou SDA. Cependant, pour pouvoir utiliser SDA à cette fin, il faudrait une certaine entorse dans le logiciel, je préfère l'éviter. On se contentera donc des lignes GND, VDD et SCL. En fin de compte, douze possibilités de sélection d'entrées analogiques avec trois adresses (cf. tableau 1) sont amplement suffisantes ! Outre les sections analogiques, se trouve aussi sur la carte (figure 6), mais à bonne distance, le microcontrôleur. Nous y reviendrons dans les deux paragraphes suivants.

Sur la carte numérique (figure 7), il n'y a qu'une seule puce d'extension à quatre adresses. Les huit entrées pour la tension du secteur (HU) sont à droite sur la photo, les huit pour la basse tension (LU) sont à gauche. Toutes les distances nécessaires entre composants et pistes respectent les prescriptions de sécurité. Les canaux sont reliés à la puce d'extension d'E/S en bas de la carte sur la photo, à côté des résistances R14

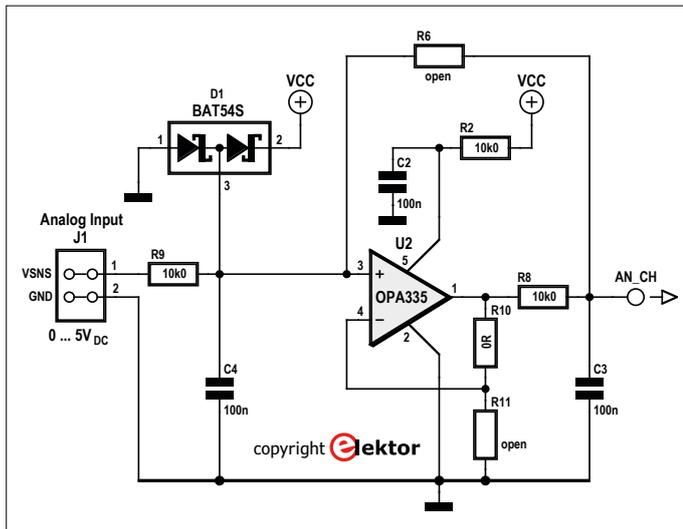


Figure 4. Schéma des entrées analogiques protégées, avec tampon ou amplificateur en option.

à R18 qui opèrent la sélection (**tableau 2**). Pour atteindre l'ensemble des 64 entrées numériques, il faut réaliser quatre circuits imprimés.

Toutes les cartes sont au format européen de 100 × 160 mm. Comme tous les trous de fixation se trouvent au même endroit, il est facile de superposer les cartes pour en faire un bloc stable et l'installer dans un coffret de répartition, comme le montre la photo du chapeau de l'article, ce qui garantit aussi l'isolement et donc la sécurité des personnes.

### Isolement analogique

Dans la partie numérique, ce sont des photocoupleurs qui assurent la séparation galvanique avec l'interface USB et le

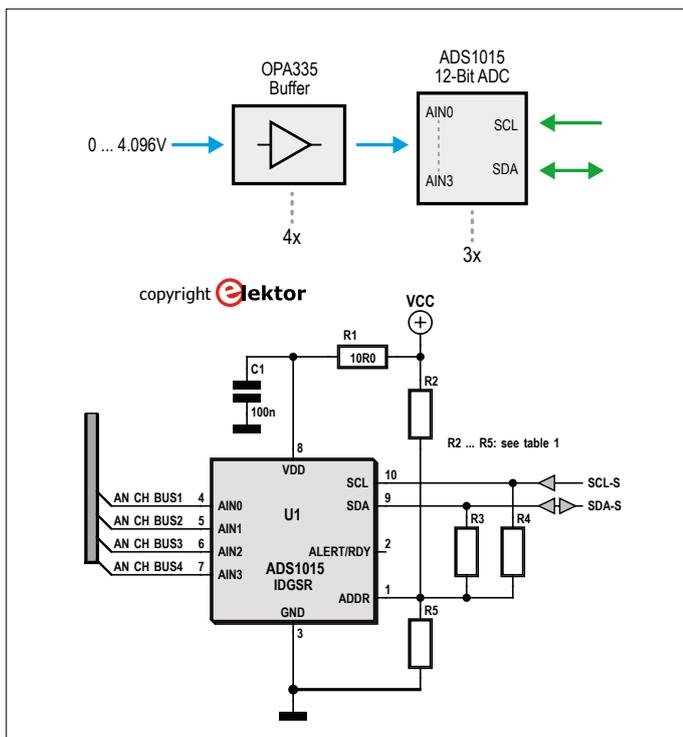


Figure 5. 4 × 3 font douze entrées analogiques à relever par bus I<sup>2</sup>C.

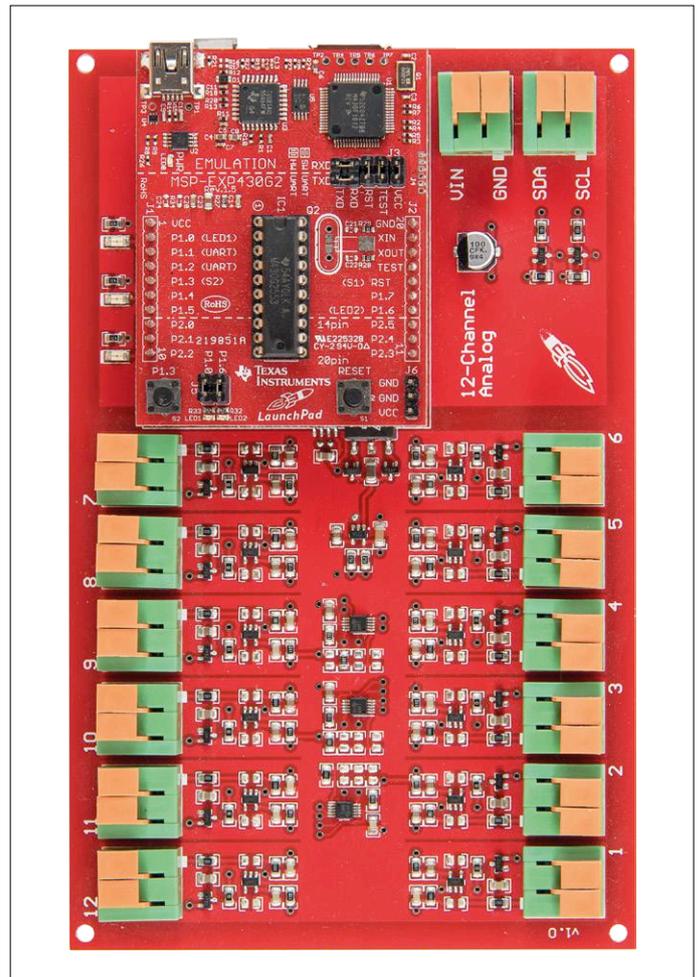


Figure 6. La carte analogique avec le module du contrôleur isolé en impériale.

microcontrôleur ; en analogique c'est un peu plus compliqué. Comme on le voit à la **figure 8**, il faut isoler le bus I<sup>2</sup>C (en vert) mais aussi l'alimentation (en rouge) entre le convertisseur A/N et le microcontrôleur. Pour le bus, on passe par un

**Tableau 1. Adressage du convertisseur A/N ADS1015 (-= ouvert; •= 10 kΩ).**

ADDR (broche 1)	adresse hexa (décimal)	R2	R3	R4	R5
GND	0x48 (72)	-	-	-	•
VDD	0x49 (73)	•	-	-	-
SDA	0x4A (74)	-	•	-	-
SCL	0x4B (75)	-	-	•	-

**Tableau 2. Adressage des extensions d'E/S (-= ouvert ; •= 0 Ω).**

A1 (broche 2)	A0 (broche 21)	adresse hexa (décimal)	R14	R15	R17	R18
L	L	0x74 (116)	-	-	•	•
L	H	0x75 (117)	-	•	•	-
H	L	0x76 (118)	•	-	-	•
H	H	0x77 (119)	•	•	-	-

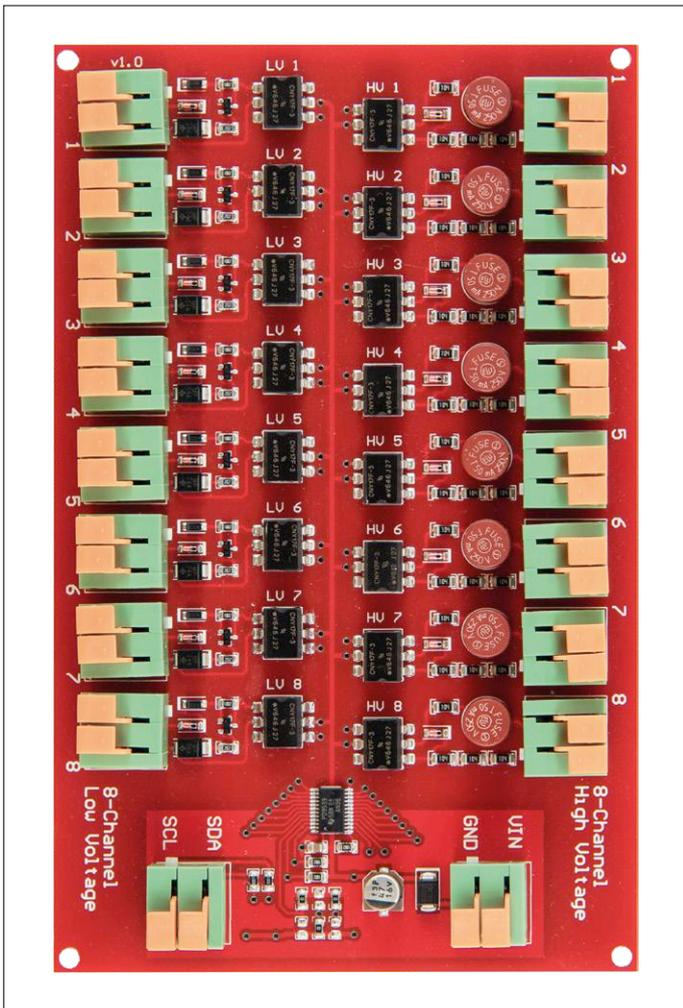


Figure 7. La carte numérique au format européen.

ISO1541, un isolateur bidirectionnel pour I<sup>2</sup>C, tandis que pour la tension d'alimentation, c'est un tampon *push-pull* SN6501 qui transfère l'énergie vers la sortie en passant par un transformateur. La **figure 9** montre le schéma en détail, un régulateur fixe LP2985-5.0 stabilise la tension à 5,0 V pour la partie analogique. C'est la tension de +5 V de l'interface USB qui sert à alimenter le reste du circuit du côté « froid ». Il ne faut donc pas d'autre source et la liaison entre le PC et la carte de mesure se résume au câble USB.

De cette manière, il y a une séparation fiable entre toutes les entrées et l'interface USB. En cas d'anomalie éventuelle du côté des entrées, l'utilisateur et son PC sont protégés à coup sûr.

### Le contrôleur et son micrologiciel

Au vu du circuit imprimé, il est clair que la partie qui concerne le contrôleur (en haut sur la figure 6) est parfaitement isolée. Comme contrôleur, j'ai choisi un module de développement tout fait et très bon marché, le LaunchPad MSP-EXP430G2 [1] de Texas Instruments, à enficher simplement sur la carte analogique. Ce module est conçu pour le développement avec des contrôleurs à faible consommation et bas coût de la série MSP430G2x, et en boîtier DIP, s'il vous plaît ! Il y a autour du connecteur du module plusieurs composants de protection contre les surtensions ainsi que des composants de découplage et des voyants qui ne seront pas évoqués ici. Vous trouverez

les circuits partiels correspondants dans les schémas compris dans le paquet à télécharger [2].

Au kit de développement LaunchPad (**figure 10**) sont joints deux contrôleurs MSP430 avec jusqu'à 16 Ko de flash, 512 octets de RAM, une horloge à 16 MHz et des périphériques intégrés tels qu'un CA/N à 8 canaux de 10 bits, un temporisateur, des interfaces sérielles (UART, I2C, SPI) et plus encore. Sur la carte, on trouve en outre une prise DIP à 14/20 contacts, des boutons, des LED et des possibilités de connecter un large éventail de modules aux fonctions supplémentaires (sans-fil, afficheur), ainsi qu'un câble USB de programmation. Si votre enregistreur du chauffage n'est pas en service, vous pourrez encore réaliser de belles expériences avec le LaunchPad !

Le micrologiciel a été réalisé avec Energia [3], un EDI Arduino basé sur la plateforme de prototypage rapide pour le LaunchPad de Texas Instruments. On y trouve de nombreuses bibliothèques intégrées toutes prêtes pour simplifier la rédaction du logiciel. Le code source en C est bien ordonné, on peut facilement le modifier et l'étendre, par exemple pour éditer ou formater les résultats de mesure et les adapter à d'autres logiciels que le programme de visualisation utilisé. Dans le dossier téléchargé, vous trouverez deux versions du micrologiciel, l'une pour Energia 16 dans lequel le projet a été initié, l'autre pour la plateforme actuelle Energia 18. Le code C pour Energia 18 est actualisé, mais il reste tout de même lieu d'apporter quelques modifications pour que la plateforme puisse reconnaître le microcontrôleur MSP430G2553. Tout cela est décrit dans les commentaires du programme.

Le logiciel lui-même a une structure très simple. Après la définition et l'initialisation des E/S, du bus et de l'interface USB, le contrôleur teste le cavalier du port 2 (il est sous le LaunchPad, donc pas visible sur la photo). Ce cavalier matérialise le choix de la durée entre deux prises de mesure, par exemple toutes les cinq ou dix secondes sur un système lent comme une chaudière. Ensuite, dans la boucle principale, le microcontrôleur relève en séquence les données brutes du CA/N et de l'extension d'E/S. On utilise la variable `channel` dans les branches `switch-case` pour différencier les différents registres et adresses I<sup>2</sup>C. Le logiciel prend en considération les trois CA/N, mais un seul extenseur d'E/S à l'adresse 0x74. Si vous voulez mettre en jeu d'autres extenseurs d'E/S, vous devrez dupliquer le fragment équivalent et y incorporer les adresses de chacun d'eux. Les valeurs sont alors converties et délivrées sur le port sériel, qui apparaît sous forme d'un port COM sur le PC. Les valeurs analogiques sont présentées en nombres entiers de millivolts, les entrées numériques sous forme de 1 ou de 0. On envoie d'abord les douze valeurs analogiques, puis les seize numériques. Les valeurs sont chaque fois séparées par une barre oblique et à la fin de la collection, on va à la ligne.

La représentation des données sur PC s'effectue avec le logiciel traceur de courbe *Rea/View* de la société Abacom [4]. Ce logiciel est très souple et facile à associer à des montages personnels. C'est lui aussi qui s'occupe de convertir les millivolts en températures, grâce aux formules qu'il contient.

### Conclusion

Le système d'enregistrement de données présenté ici remplit son office depuis de nombreux mois sans difficulté. En outre il a été d'une aide décisive pour optimiser les réglages de la

chaudière ainsi que son rendement, et augmenter sa durée de vie. Comme il est possible d'observer les valeurs déterminantes de température et de suivre les commutations, on peut se prononcer assez vite sur la validité des changements des paramètres de commande du chauffage, voir s'ils vont dans la bonne direction et s'ils sont conformes sur le long terme dans des conditions variées.

Ce projet ne se restreint évidemment pas au chauffage. D'une manière générale, il traite de l'acquisition de données tant numériques qu'analogiques, et de leur analyse sur PC. Autrement dit, il ouvre la voie à d'autres projets dans le vaste domaine de la mesure. ◀

(160358 - version française : Robert Grignard)

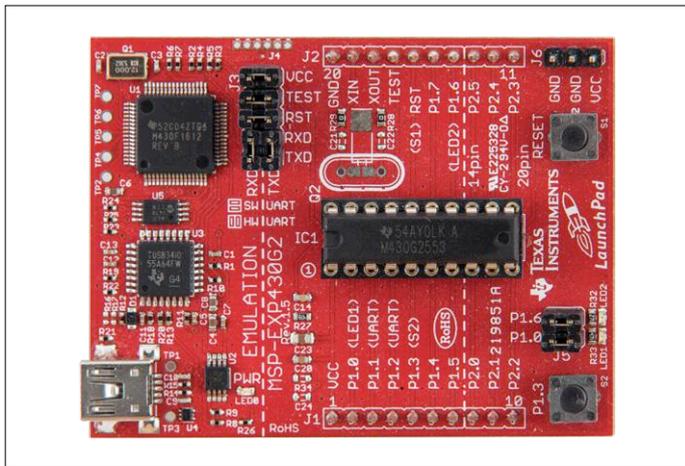


Figure 10. Le microcontrôleur choisi est sur le module LaunchPad de Texas Instruments, solution simple et bon marché.

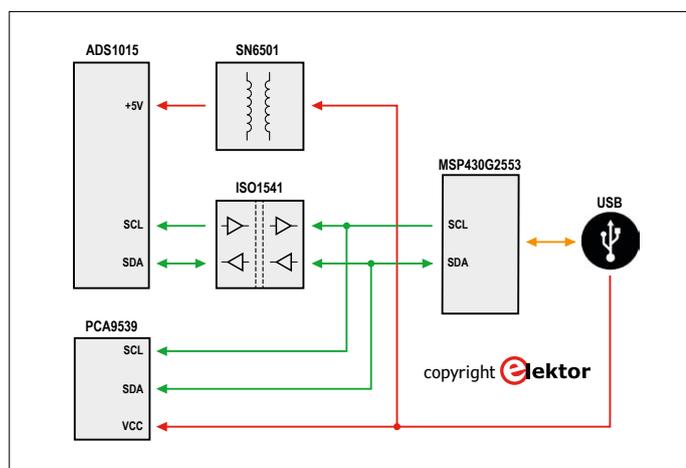


Figure 8. Voici comment isoler le contrôleur et son interface USB des entrées de mesure.

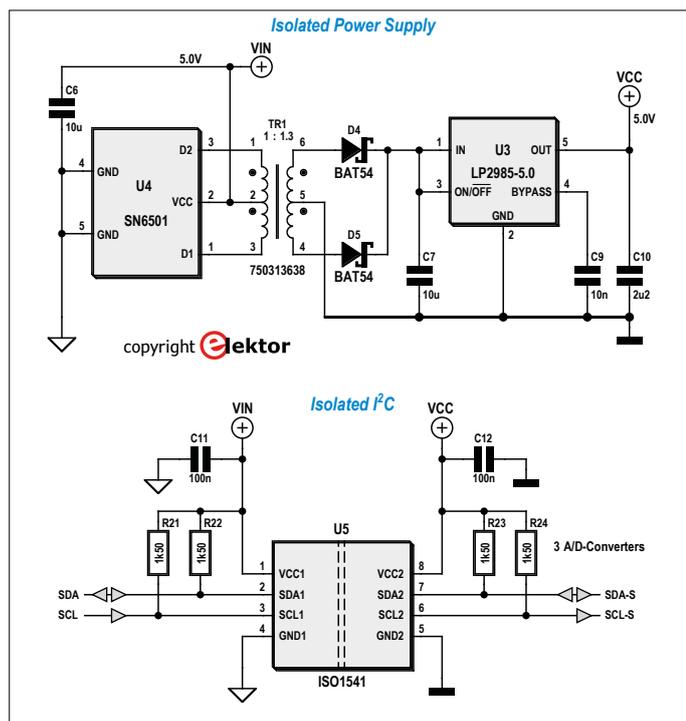


Figure 9. Isolement du bus I<sup>2</sup>C et de la tension d'alimentation.

## Liens

- [1] <http://www.ti.com/tool/MSP-EXP430G2#0>
- [2] [www.elektormagazine.fr/160358](http://www.elektormagazine.fr/160358)
- [3] <http://energia.nu/>
- [4] [www.abacom-online.de/uk/html/realview.html](http://www.abacom-online.de/uk/html/realview.html)

## Dans le paquet à télécharger [2]

### Vous trouverez :

- le micrologiciel pour Energia 16 et Energia 18
- la liste des composants
- tous les schémas en PDF
- les tracés des pistes et les sérigraphies en PDF
- l'épure des pistes réalisée sous Altium (compatible Eagle)
- les données de forage (Gerber)

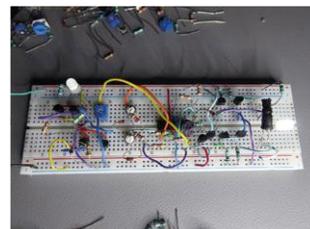
# bruits de labo



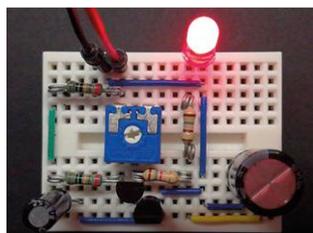
Il y a plusieurs semaines déjà, le labo d'Elektor a lancé un modeste concours : il s'agissait de concevoir un circuit de fondu lumineux à LED, mais sans faire appel à un microcontrôleur. Et nous avons reçu un nombre surprenant de propositions. Bravo à tous !

## La clôture du concours n'a pas arrêté le gagnant

Par ses nombreuses propositions de circuits et l'éventail des techniques utilisées, Roel Arits nous est apparu comme le roi incontesté du fondu lumineux à LED. Ses premiers circuits comportaient une photorésistance et des miroirs de courant. Mais il ne s'en est pas tenu là et a fait d'autres propositions, y compris après la clôture du concours, en explorant les effets de fondu logarithmique et sinusoïdal, mais aussi les oscillateurs tangentiels. Aucune piste n'a été négligée, comme le montre son ultime circuit, basé sur un potentiomètre motorisé.



<https://goo.gl/rDWTLz>



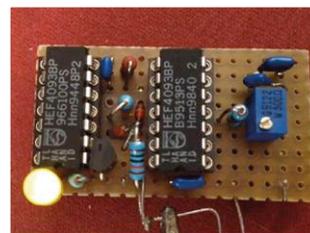
<https://goo.gl/cPajhh>

## Et pourquoi pas un transistor unijonction programmable ?

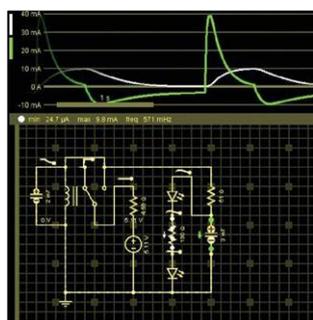
Autre concurrent, Anto a remporté le Prix spécial pour un oscillateur à transistor UJT programmable aussi simple qu'efficace. Nous apprécions particulièrement les transistors unijonction et nous serions ravis de les voir plus souvent utilisés (d'autant que le 2N2646 semble être immédiatement disponible). Pour contourner la question de l'achat, Anto a fabriqué son UJT à partir de deux transistors des plus classiques.

## Une fréquence de battement, c'est bien aussi...

La plupart des participants ont tous plus ou moins utilisé un générateur avec une fréquence d'oscillation égale à celle du rythme de l'effet de fondu lumineux. Mais Arnoldus, autre concurrent, a opté pour une solution très différente. Décidé à sortir des sentiers battus, il a préféré deux oscillateurs à signal carré, légèrement désaccordés. Il a ainsi créé une différence de fréquences de 0,5 Hz qui permet de commander la LED. Prisée des spécialistes des ondes radio, cette technique est baptisée hétérodynage. Le signal de battement obtenu à partir des deux signaux carrés est similaire à un signal de modulation de largeur d'impulsions. Du fait de la modulation linéaire, le fondu est lui-même linéaire et semble moins naturel qu'avec la méthode logarithmique.



<https://goo.gl/awWMCi>



<https://goo.gl/1tMejd>

## Et même un relais...

Nous sommes tellement habitués aux composants transistorisés que nous oublions souvent qu'avant l'apparition des semi-conducteurs, les opérations de commutation étaient souvent réalisées à l'aide de composants électromécaniques. D'où l'idée d'Heribert Houben de revenir aux fondamentaux et d'élaborer un circuit de fondu lumineux électromécanique (!) ne comportant que deux semi-conducteurs : les LED. Il s'agit d'un oscillateur à relais, version mécanique des circuits de sonnettes ou de carillons d'entrée, mais équipé ici d'un gros condensateur pour éviter les fronts raides et obtenir une certaine pente de montée.

## Quelques repères sur les concours du labo d'Elektor

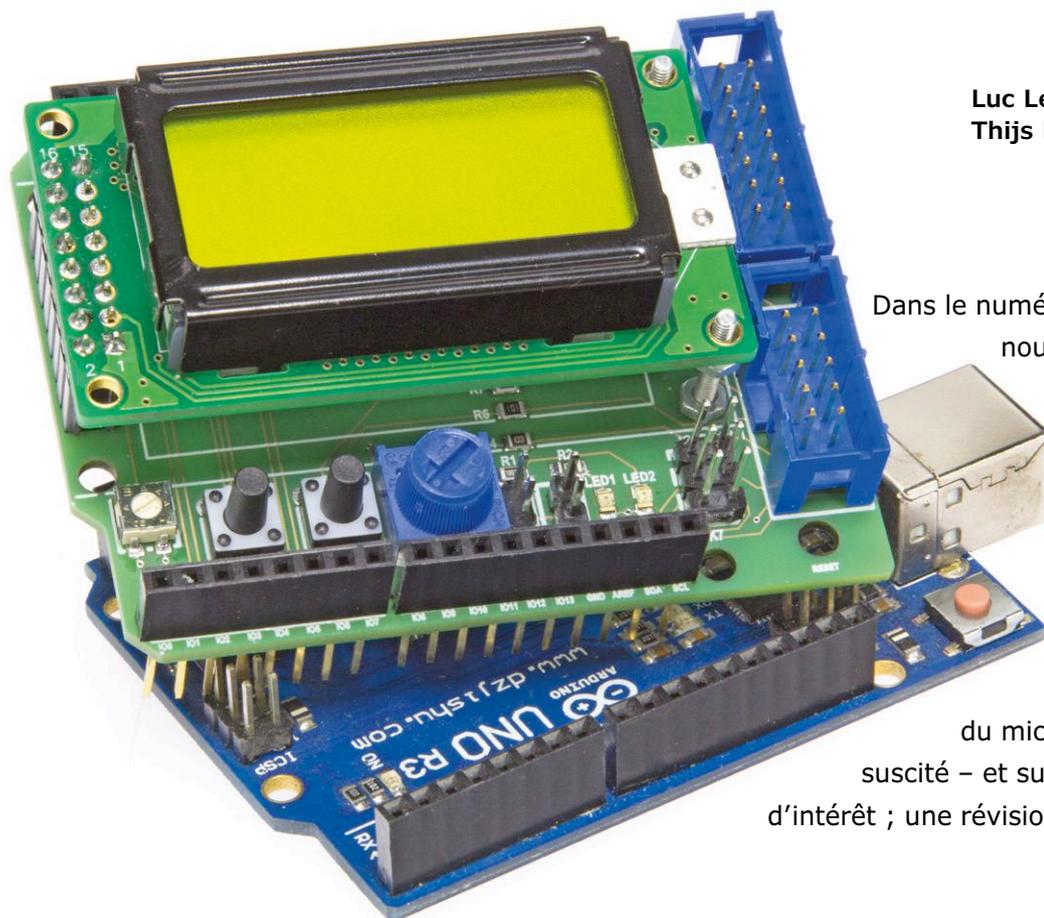
Au labo d'Elektor, nous lançons un concours dès que nous sentons que le moment est venu. Les règles sont indiquées sur la page d'accueil du labo (onglet Elektor Labs sur [elektormagazine.fr](http://elektormagazine.fr)). C'est aussi là que les participants publient leurs propositions et que les gagnants sont annoncés. La page d'accueil du concours d'effet de fondu lumineux à LED contient la liste de toutes les propositions reçues : <https://goo.gl/JWo8oC> ◀ (160558 - version française : Pascal Godart)

# shield d'expérimentation Arduino version 2.0

## les mêmes fonctions, mais écran au choix

Luc Lemmens (labo d'Elektor) &  
Thijs Beckers (rédaction d'Elektor)

Dans le numéro de juillet-août 2014, nous avons décrit une carte d'extension (*shield*) pour l'Arduino Uno [1]. Ce *shield* améliorerait l'Arduino avec des LED, des boutons-poussoirs, un écran LCD et un accès à l'interface UART et aux broches du microcontrôleur. Ce *shield* a suscité – et suscite encore – beaucoup d'intérêt ; une révision s'imposait donc !



Le *shield* d'expérimentation Arduino version 2.0 offre en gros les mêmes possibilités que son prédécesseur ; après trois ans et demi, ça ne fera pas de mal de se rafraichir la mémoire.

### Caractéristiques

- Écran de 2×8 caractères
- 2 boutons-poussoirs
- 1 potentiomètre
- 2 LED d'indication
- Connecteurs d'extension Arduino
- Connecteur ICSP
- Connecteurs d'extension pour bus ECC et EEC

### Schéma

Sur le schéma repris en **figure 1**, nous constatons que le *shield* comporte deux LED, deux boutons-poussoirs, un potentiomètre et un écran LCD de 2×8 caractères, que nous pouvons utiliser librement. Il y a en outre deux connecteurs qui permettent de relier d'autres circuits via un câble en nappe, par exemple une passerelle radio [2] ou un module CA/N à 16 bits [3]. Rien de neuf pour ceux qui connaissent la version précédente du *shield*.

La nouveauté, c'est l'écran et ses connexions. La disposition des connexions de l'écran choisi en 2014, un EA DIPS082,

était quelque peu exotique ; le nouveau *shield* accepte quant à lui tout écran LCD alphanumérique avec connecteur à 16 trous (2×8) standard. Nous ne sommes plus limités à deux lignes de huit caractères, on peut choisir un écran de plus grande capacité. L'écran fourni avec le module complet [4] est un modèle de 2×8 caractères qui permet d'entrer directement dans le vif du sujet.

### Entrées et sorties

Comme c'est le cas sur beaucoup de *shields*, les connecteurs de l'Arduino sont recopiés tels quels, c'est le rôle de K3 à K6 dans notre circuit. Bien entendu, il est impossible de modifier l'emplace-

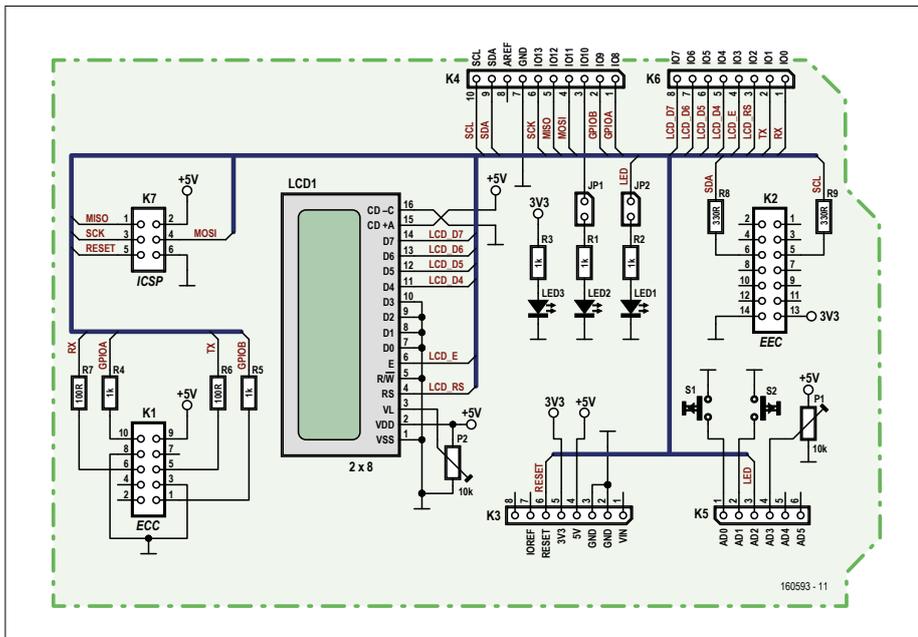


Figure 1. Le schéma n'a pratiquement pas changé. Il y a deux connexions supplémentaires pour l'écran LCD.

► Désormais, connectez n'importe quel écran LCD avec connexion à 2x8 trous

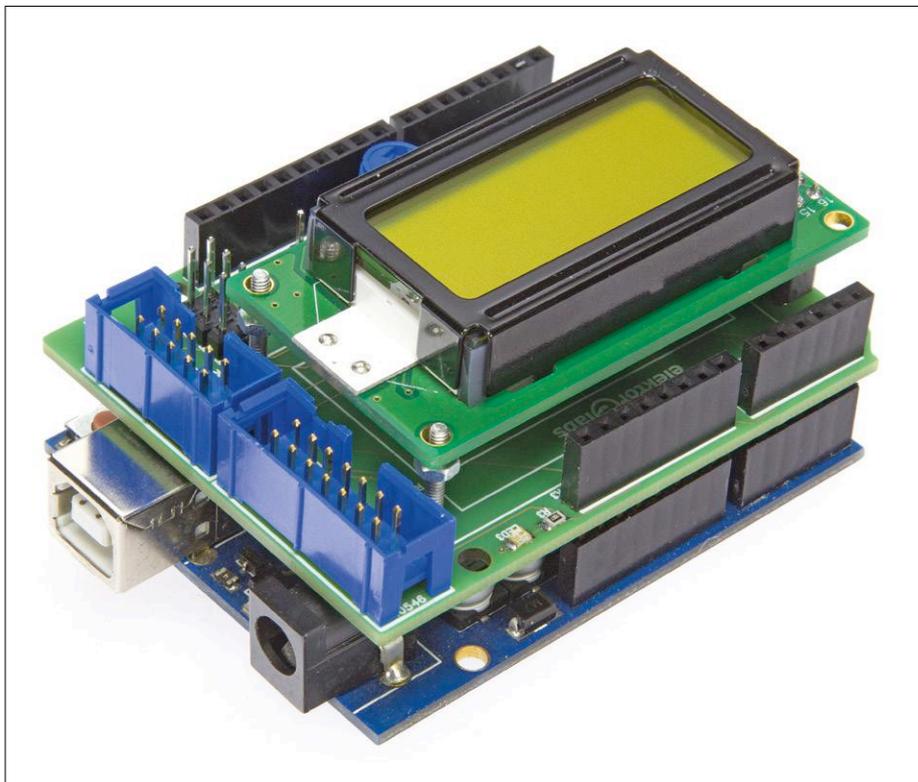


Figure 2. Faites attention en enfichant le shield, il y a un risque de court-circuit.

### INFOS SUR LE PROJET

- Arduino shield LCD
- débutant
- connaissseur expert
- env. ½ h
- Fer à souder et soudeur
- env. 40 €

ment de ces connecteurs, mais la plupart des autres composants ont trouvé une nouvelle place.

La plupart des broches de l'Arduino sont disponibles sur le shield ; en fait, si JP1, JP2 et l'écran ne sont pas utilisés, elles le sont toutes. Il faut juste veiller à la position de P1, qui est relié à AD3. Il est à noter que l'écran LCD, les boutons-poussoirs et le potentiomètre empêchent pratiquement de placer un autre shield au-dessus de celui-ci, mais les broches de l'Arduino sont facilement accessibles pour des fils de connexion.

Les LED sont commandées via AD2 (LED1) et IO10 (LED2) ; il faut alors installer des cavaliers sur JP1 et JP2. LED3 signale la présence de la tension de 3,3 V. S1 et S2 sont reliés respectivement à AD0 et AD1 ; ces deux entrées fonctionnent aussi en entrées analogiques, nous n'avons donc pas utilisé de résistances de rappel vers le haut ou de circuit anti-rebond. Les résistances de rappel internes de l'ATmega328 de l'Arduino sont connectées par logiciel lorsque nécessaire.

#### Écran

La bibliothèque « *LiquidCrystal.h* » de l'environnement de développement intégré (EDI, IDE) de l'Arduino permet de commander l'écran LCD. Si nous voulons par exemple lancer le croquis *HelloWorld*, il suffit de remplacer la ligne

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

par

LiquidCrystal lcd(2, 3, 4, 5, 6, 7);

et la ligne

lcd.begin(16, 2);

par

lcd.begin(8, 2);

### Attention !

Pour que le module ne dépasse pas du circuit imprimé de l'Arduino (Uno), nous avons dû placer les connecteurs mâles au-dessus du connecteur USB en métal (figure 2). En insérant le shield sur l'Arduino, il faudra donc veiller à ce qu'il n'y ait aucun court-circuit entre le connecteur USB et K1. Sur le module que nous proposons, les broches de K3 à K6 sont très longues pour maintenir une bonne distance entre les deux. Le plus sûr est d'isoler le connecteur USB avec un morceau de ruban adhésif isolant ou – encore mieux – de plastique dur. ◀

(160593 – version française : Jean-Louis Mehren)

### Liens

- [1] [www.elektormagazine.fr/140009](http://www.elektormagazine.fr/140009)
- [2] [www.elektormagazine.fr/130023](http://www.elektormagazine.fr/130023)
- [3] [www.elektormagazine.fr/130485](http://www.elektormagazine.fr/130485)
- [4] [www.elektormagazine.fr/160593](http://www.elektormagazine.fr/160593)



## LISTE DES COMPOSANTS

### Résistances

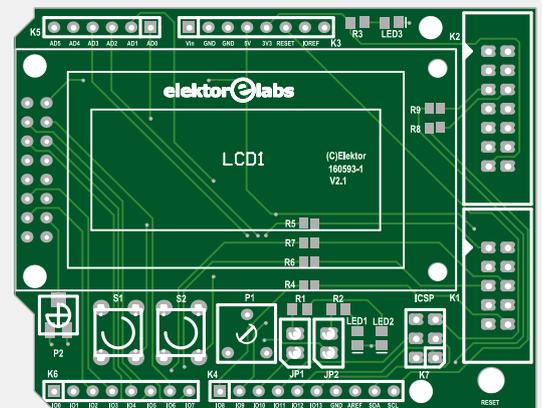
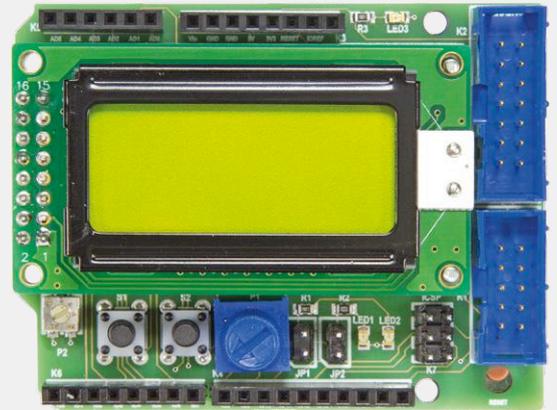
R1-R5 = 1 kΩ  
R6, R7 = 100 Ω  
R8, R9 = 330 Ω  
P1 = potentiomètre ajustable  
10 kΩ avec bouton  
P2 = potentiomètre  
ajustable 10 kΩ  
(CMS, Vishay TS53YJ103MR10)

### Semi-conducteurs

LED1, LED2 = LED rouge  
à faible courant (CMS 0805)  
LED3 = LED verte  
à faible courant (CMS 0805)

### Divers

S1, S2 = bouton-poussoir  
K1 = connecteur HE10  
mâle, à 2×5 contacts,  
pas de 2,54 mm  
K2 = connecteur HE10  
mâle, à 2×7 contacts,  
pas de 2,54 mm  
K3, K6 = barrette femelle  
à 8 contacts \*  
K4 = barrette femelle  
à 10 contacts \*  
K5 = barrette femelle  
à 6 contacts \*  
K7 = connecteur ICSP,  
à 2×3 contacts,  
pas de 2,54 mm  
JP1, JP2 = 2 picots à souder, pas de 2,54 mm,  
avec cavalier  
LCD1 = écran LCD à 2×8 caractères avec  
rétroéclairage, TC0802B  
LCD1 = connecteur mâle à 2 × 8 contacts,  
pas de 2,54 mm  
LCD1 = barrette femelle à 2×8 contacts pour



LCD1, pas de 2,54 mm

LCD1 = 4 vis M2.5×20

LCD1 = 12 écrous M2.5

Circuit imprimé, réf 160593-1 [4] ou

Module complet, réf 160593-91 [4]

\* K3 à K6 = connecteurs pour shield Arduino

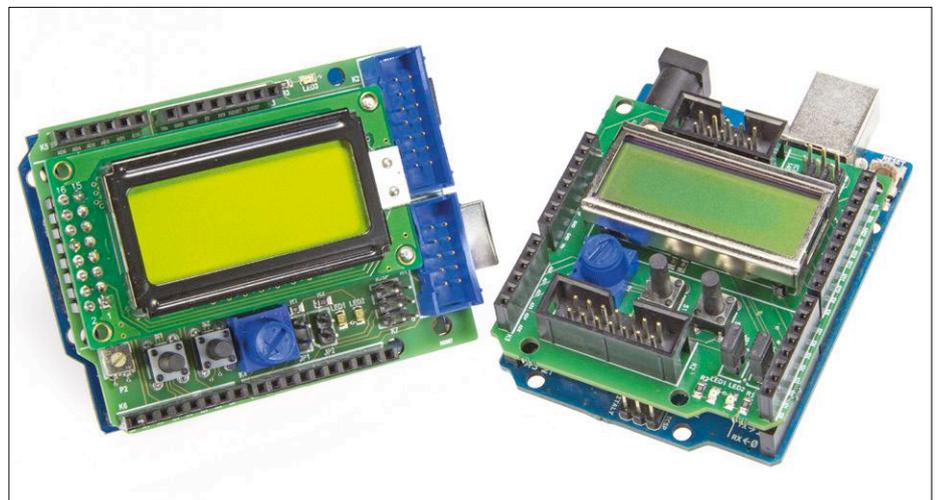


Figure 3. Le nouveau shield (à gauche) et l'ancien (à droite).



### DANS L'E-CHOPPE

→ 160593-1  
circuit imprimé

→ 160593-91  
module complet

# comment consommer moins

## avec du calcul et de la méthode

Robert Lacoste (Chaville)



illustration : shutterstock.com

Si vous êtes comme moi, vous pestez tous les jours face à des appareils électroniques avec une batterie déchargée en quelques jours (voire en quelques heures) ou avec des piles trop souvent mortes. Vous vous êtes sûrement heurté au même problème avec vos montages alimentés par piles ou batteries. Alors, comment augmenter l'autonomie d'un circuit ? Quelles sont les bonnes pratiques à privilégier ? Quelles sont les astuces applicables à quasiment tous les projets ? Cet article répond à ces questions.

### Un exemple...

Trêve de généralités, penchons-nous sur un petit exemple très concret. Imaginez que vous deviez concevoir un petit dispositif de cuisine avec un minuteur et un

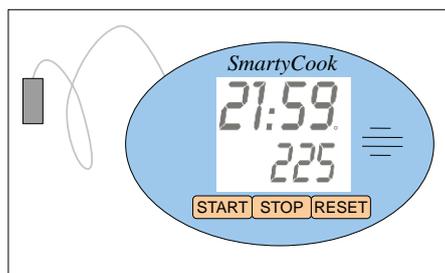


Figure 1. L'exemple pris pour cet article est un petit minuteur de cuisine avec thermomètre.

thermomètre. Je l'ai baptisé SmartyCook (**fig. 1**). Rien de bien compliqué : une sonde de température peut être plantée dans l'aliment à cuire, quelques boutons permettent de programmer la durée de cuisson ou la température maximale à atteindre, enfin un afficheur LCD anime le tout. Bien sûr un logiciel intégré permet de réussir la cuisson de votre rosbif ou foie gras comme un chef. En tant que fidèle lecteur d'Elektor, nul doute que vous pourriez bricoler aisément ce genre de chose, non ?

Plutôt que de partir d'une feuille blanche, imaginons que vous ayez dégotté un schéma quelque part (**fig. 2**). Ce projet repose bien entendu sur un microcontrôleur. Ici on a choisi un petit  $\mu\text{C}$

à 8 bits, à ultra-basse consommation, un PIC16F914 de Microchip. Cette variante du PIC16F dispose d'une interface de pilotage d'un LCD multiplexé qui ne nécessite en externe que deux tensions de référence, respectivement égales à 1/3 et de 2/3 de la tension d'alimentation, produites par R1/R2/R3. Un quartz de 4 MHz cadence le  $\mu\text{C}$ . La sonde de température est une thermistance de type PT1000 (ce qui signifie qu'elle a une valeur de 1 k $\Omega$  à 20 °C), alimentée par une entrée/sortie du  $\mu\text{C}$  via R4. La tension résultante est lue par le convertisseur A/N interne du  $\mu\text{C}$ . Une autre paire de résistances permet de réaliser une mesure en pont et donc d'annuler par logiciel l'influence de la tension d'alimen-

tation. Pour faire « bip-bip » en fin de cuisson, un petit haut-parleur est directement piloté par une autre sortie. Enfin trois boutons-poussoirs sont connectés à d'autres lignes du µC et sont polarisés via des résistances de rappel vers le haut (*pull-up*) internes. Ces lignes sont configurées par le logiciel pour produire une interruption lorsqu'on appuie sur l'un des boutons. Le tout est alimenté par deux piles AAA de 1,5 V en série, donc sous 3 V.

Pas trop mal, non ? Attention quand même : j'ai inventé ce schéma juste pour illustrer cet article, il n'a jamais été physiquement réalisé, donc ne m'en voulez pas s'il y a une coquille quelque part...

### Bilan énergétique

Tout d'abord, comment estimer l'autonomie d'un tel projet, c'est-à-dire la durée de vie de ses piles ? Ce calcul, appelé bilan énergétique, est indispensable avant de réaliser physiquement le circuit, d'une part pour voir si cette autonomie est raisonnable, mais aussi pour identifier les sources d'amélioration potentielles.

L'établissement d'un bilan énergétique compte traditionnellement quatre étapes. Tout d'abord il faut lister ce que j'appelle les modes de fonctionnement du produit. Par définition, il s'agit des états du produit pendant lequel sa consommation est grosso modo constante. Sur le SmartyCook, on distingue trois modes :

- en veille (non utilisé) ;
- actif (décompte du temps et mesure de température en cours) ;
- production du bip-bip (on estime qu'à ce moment-là la consommation est nettement supérieure puisque le HP est alimenté).

Cette décomposition est bien entendu plus complexe si l'appareil possède nombre de fonctions. Vous devez aussi estimer combien de temps l'appareil sera dans chaque mode, par ex. pour chaque journée. À ce stade, des hypothèses sont nécessaires. Considérons que le SmartyCook sera utilisé deux fois par jour en moyenne, chaque fois pour une durée de cuisson de 15 min (toujours en moyenne), et fera bip-bip pendant 30 s avant qu'on ne l'arrête. Vous aboutissez à la liste des modes de fonctionnement résumés sur le **tableau 1**.

La seconde étape est de calculer la consommation effective dans chacun

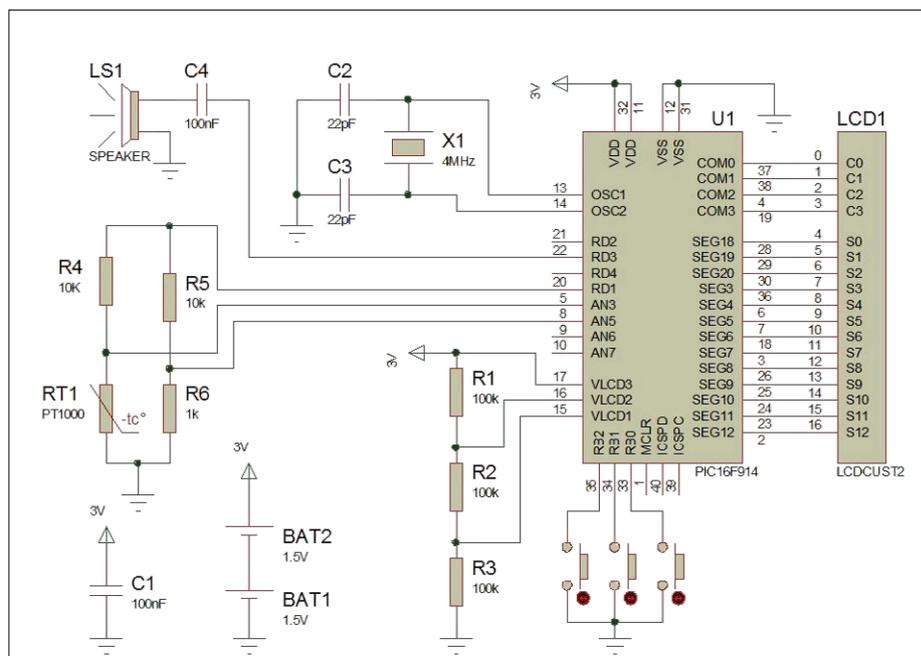


Figure 2. Le schéma du SmartyCook avant optimisation, rien de bien compliqué.

Tableau 1. Modes de fonctionnement.

Mode	Description	Durée (s/jour)	Commentaire
VEILLE	Produit en veille profonde, LCD éteint	84 600	24h – actif
ACTIF	Produit en utilisation, LCD allumé, mesures et décomptes en cours	1 800	2×15 min
BIP	Production du son	60	2×30 s

Tableau 2. Consommation en µA pour chacun des modes de fonctionnement.

Mode	PIC16F914	LCD	R1/R2/R3	R4/R5/R6/RT1	HP	Total
VEILLE	2,300		10,000			12,300
ACTIF	320,000	100,000	10,000	545,455		975,455
BIP	320,000	100,000	10,000	545,455	10 000,000	10 975,455

des modes, par définition supposée constante. Ici, les feuilles de caractéristiques des composants et quelques petits calculs suffisent. Le plus important est bien entendu de ne rien oublier et de ne pas trop se tromper d'ordre de grandeur. Le plus efficace pour cela est de passer en revue chaque fil du schéma, et de se demander si du courant circule dans ce fil dans tel ou tel mode, puis de calculer son intensité. Le **tableau 2** donne les résultats pour le SmartyCook. Les courants estimés sont respectivement de 12 µA, 975 µA et 10,9 mA selon le mode de fonctionnement. À noter que le µC utilisé ne consomme que 320 µA en fonctionnement sous 3 V et à 4 MHz :

une belle performance !

La troisième étape est la plus simple, mais aussi la plus importante : calculez la consommation énergétique moyenne quotidienne, exprimée en mA.h. Pour cela, multipliez simplement le temps passé dans chaque mode de fonctionnement (converti en heures) par le courant consommé dans ce mode (exprimé en mA), et additionnez le tout. Le **tableau 3** contient les résultats ; il ressort qu'avec nos hypothèses, notre SmartyCook consomme chaque jour un peu moins de 1 mA.h. On voit aussi immédiatement que le mode le plus gourmand est le mode « actif » (50% du total). Le mode « bip-bip » ne représente

**Tableau 3.**  
**Consommation énergétique moyenne quotidienne, exprimée en mA.h.**

Mode	Durée (s/j)	Courant instantané (µA)	Énergie (mA.h)	% total
VEILLE	84 600	12,300	0,289050	30,1%
ACTIF	1 800	975,455	0,487727	<b>50,8%</b>
BIP	60	10 975,455	0,182924	19,1%
Total par jour (mA.h)			<b>0,959702</b>	100,0%

**Tableau 4. Autonomie.**

Type de source	Piles AAA zinc/carbone		
Référence	Eveready 1212		
Capacité (mA/h)	Théorique	Projet	Variation (estimation de l'auteur)
Capacité théorique (mA .h)	464		
Tension mini (V)	1,0	1,2	20%
Durée de vie (années)	0,1	2,0	20%
Variation entre fournisseurs			20%
Pics de courant (mA)	16,0	10,0	-5%
Température mini (°C)	20,0	20,0	0%
Capacité estimée (MA.h)			249
Consommation quotidienne (mA.h)			0,960
Autonomie estimée (j)			<b>260</b>

que 19%. Même si l'intensité du courant dans ce mode est très élevée, sa durée d'activation est assez faible. Cette simple analyse donne des pistes sur ce qu'il faut optimiser en priorité. La dernière étape est le calcul de l'autonomie prévisionnelle, en jours. Il suffit de prendre la capacité de la pile, donnée en mA.h, et de la diviser par la dépense énergétique quotidienne, aussi en mA.h, pour trouver le nombre de jours d'autonomie de l'appareil. Ici on a recours à deux piles alcalines AAA de 464mA.h (capacité indiquée par le fabricant). L'appareil consomme 1 mAh par jour. Fonctionnera-t-il pendant 464 jours ? Eh bien non, car cette capacité théorique est donnée par le fabricant pour des conditions très particulières et généralement très optimistes. La température ambiante sera-t-elle toujours de 20 °C ? Le courant de décharge sera-t-il constant ? Quelle est la tension en fin de vie de la pile prise en compte par le fabricant ? Votre appareil fonctionnera-t-il encore avec une pile déchargée ? L'estimation de ces facteurs est malheureusement un peu complexe, car les données nécessaires sont rarement disponibles et relèvent plus de la chimie que de l'électronique. Cela dit, si

l'on prend tous ces facteurs en compte, il n'est pas exceptionnel que la capacité réelle soit réduite de 30 à 50% par rapport aux données théoriques. Avec des hypothèses évidemment discutables, on arrive avec ce type de calcul à une autonomie estimée pour le SmartyCook de l'ordre de 260 jours (**tableau 4**). Plus de huit mois ! Ce n'est pas si mal, non ? Mais peut-on faire mieux ?

**Réduire les courants**

La **figure 3** contient la formule qui permet de calculer la consommation énergétique quotidienne. Gardez bien en tête que chacun de ses termes peut et doit être étudié pour améliorer l'autonomie d'un produit, et que cette démarche doit être systématique : une amélioration peut sembler minime si un autre contributeur est nettement plus important dans le bilan global, mais peut se révéler in fine cruciale si ce contributeur est largement amélioré plus tard... Commençons par la piste la plus évidente : cherchons à réduire le courant consommé lorsque l'appareil est actif. Comment faire ? La première solution, toujours efficace, est de tenter par tous les moyens de réduire la tension de

fonctionnement de l'électronique. Pourquoi ? À cause de la loi d'Ohm,  $I=U/R$  ! Si on abaisse l'amplitude de la tension, on diminue d'autant l'intensité du courant à travers la résistance. De même, tous les semi-conducteurs consomment nettement moins lorsque la tension est réduite. Pour vous en convaincre, téléchargez donc la feuille de caractéristique du PIC16F914 [1] et regardez la partie « power consumption ». Vous verrez que sa consommation est réduite de 320 µA à 180 µA, soit un gain de 50%, lorsqu'on l'alimente en 2 V au lieu de 3 V, pour un service rendu identique ! Faites donc le tour de tous les composants de votre circuit et utilisez toujours la tension minimale compatible avec tout le monde. Si tous les composants tolèrent une tension de 2 V, il serait stupide de les alimenter en 3 V... Pour ce faire, un simple régulateur linéaire peut suffire (à condition d'en sélectionner un qui consomme lui-même très peu). Si vous ajoutez un petit régulateur 2 V de type MPC1700 (consommation propre de 2 µA) entre les piles et le circuit du SmartyCook, l'autonomie totale augmente de 30% ! Là, je vois que certains lecteurs froncent les sourcils, en pensant qu'un régulateur linéaire dissipe de la puissance et qu'il n'y a donc pas de gain, mais détrompez-vous. Ici le µC consomme 320 µA sous 3 V et 180 µA sous 2 V ; avec un régulateur 2 V, le courant tiré des piles passe de 320 µA à un peu plus de 180 µA (182 µA pour être exact), même si  $180 \mu A \times (3 V - 2 V) = 180 \mu W$  sont dissipés dans le régulateur. Un convertisseur DC/DC basse consommation permet un gain encore supérieur, mais avec une complexité accrue. Cela sera le sujet de mon prochain article... Quelles sont les autres voies ? La réduction des fréquences d'horloge est un remède classique. Sur le SmartyCook, on utilise un quartz de 4 MHz pour l'horloge du µC. Pensez-vous réellement qu'il faille faire tourner aussi vite un µC pour compter les secondes sur un thermostat de cuisine ? Évidemment que non ! Il est probablement possible de réduire cette fréquence de plusieurs ordres de grandeur, sans conséquence sur le fonctionnement pour l'utilisateur. Rame-nons le quartz à 32 kHz. Quel impact sur la consommation du PIC ? Elle passe de 180 µA à 4 MHz à 8 µA à 32 kHz ! C'est le cas avec tous les circuits CMOS, dont la consommation est presque proportionnelle à la fréquence de fonction-

nement. Si vous faites le calcul complet, vous constaterez que cette baisse draconienne n'augmente l'autonomie totale que de 10%, simplement parce que d'autres contributeurs deviennent majoritaires ; toutefois cette modification vaut la peine !

Enfin il faut passer en revue chaque composant pour essayer de trouver des variantes frugales en énergie. Encore une fois, il faut de la méthode dans cette démarche. Existe-t-il des variantes plus récentes et moins gourmandes ? Ou d'autres alternatives technologiques ? Peut-on augmenter la valeur des résistances ? Sur l'exemple du SmartyCook, pensez-vous que des améliorations de ce type soient possibles ? Il y en a au moins trois. Regardez à nouveau le schéma (fig. 2) et essayez de les trouver.

Ça y est ? Tout d'abord la sonde de température est une PT1000 (1 kΩ à 20 °C). Pourquoi ne pas la remplacer par une PT10000 qui réduirait tous les courants de mesure d'un facteur 10 ? De même, les trois résistances R1/R2/R3 servent à délivrer des tensions intermédiaires pour le LCD, rien n'empêche de les passer de 100 kΩ à 470 kΩ, puisque les impédances d'entrée du µC sont très élevées. Et hop, encore quelques µA de gagnés. Enfin n'y a-t-il pas une meilleure solution technologique que le haut-parleur ? Un transducteur piézo conviendrait puisqu'il a le même niveau de bruit et il consommerait nettement moins.

De petites améliorations pensez-vous ? Comme souvent, les petits ruisseaux font les grandes rivières. Si l'on accumule ces petites améliorations et que l'on refait le calcul du bilan énergétique, on trouve une autonomie portée de 260 jours à... 1055 jours ! Pas mal, non ?

### Réduire les durées d'activité

Mais on peut aller plus loin. Rappelez-vous que l'énergie consommée est proportionnelle au courant, mais aussi au temps pendant lequel on le consomme. On peut donc aussi gagner beaucoup en réduisant le temps passé dans les modes les plus gourmands, ou en segmentant ces modes en plusieurs modes qui consomment en moyenne moins. Quelques pistes classiques ? Une solution que j'adore est de rendre l'appareil plus efficace. Si l'on arrive à réaliser la même fonction en moins de temps, l'appareil retournera plus vite en veille, d'où un gain pour la pile, mais aussi pour l'utilisateur.

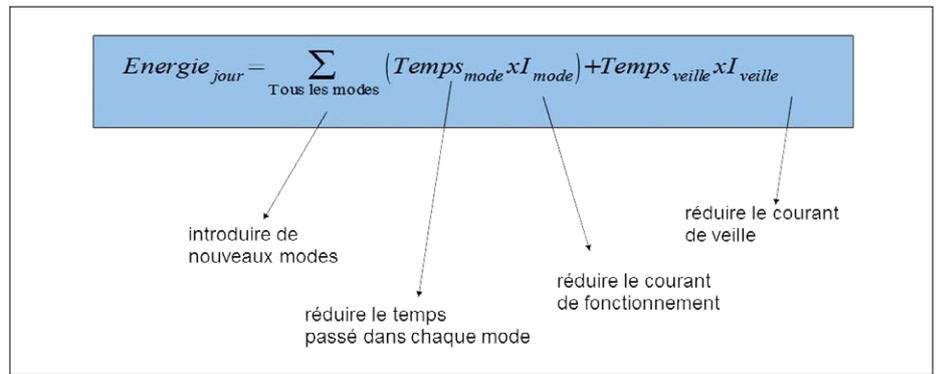


Figure 3. Chaque terme contribuant à l'énergie consommée peut être une piste d'optimisation.

En parlant de méthodes traditionnelles, pensez aussi à utiliser davantage les interruptions pour que le µC retourne en sommeil dès que possible. Évidemment pour exécuter une temporisation dans un logiciel embarqué, il est préférable de proscrire les boucles d'attente (« tant que pas fini, le circuit consomme et la batterie s'épuise »), d'armer un timer et de mettre le µC en veille jusqu'à la fin de la temporisation. Compresser les données avant de les transmettre est aussi une piste efficace pour réduire la durée de transmission. Par exemple optimiser un protocole de communication et transmettre du binaire au lieu de messages en ASCII peuvent faire gagner beaucoup d'énergie. Pensez aussi à la qualité du code en lui-même : dans un de mes projets, j'ai

réduit la consommation de presque 20% simplement en utilisant un meilleur compilateur pour produire le code ! Dans l'exemple du SmartyCook, que pourrions-nous faire ? Peut-être émettre des bip-bip plus brefs, avec un rapport cyclique nettement plus faible ? Si l'on active le transducteur pendant 0,1 s toutes les 2 s, le signal sera toujours audible, et le temps d'activation du mode « bip » sera réduit d'un facteur 20. Le calcul montre que l'impact sur l'autonomie est énorme, car ce mode est devenu, au fil des améliorations, l'un des contributeurs majeurs : on passe de 1 000 jours d'autonomie à 1 500 jours avec cette seule modification ! Une autre règle de bonne pratique est de passer en revue tous les composants et de se demander s'il est vraiment

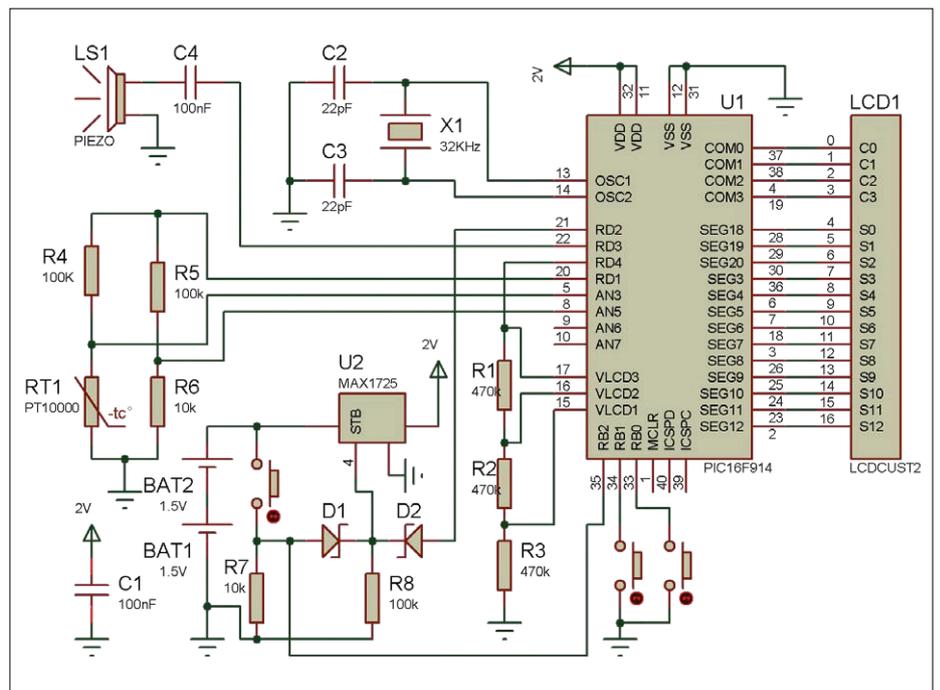


Figure 4. Schéma du SmartyCook après optimisation. En apparence, il est peu différent du schéma initial, mais l'autonomie est multipliée par quinze !

indispensable qu'ils soient tous alimentés dans tous les modes. Regardez le schéma du SmartyCook, en particulier R1/R2/R3. Voyez-vous une grosse amélioration possible ? Ces trois résistances sont connectées directement aux piles, et donc consomment du courant en permanence. Or elles ne sont utiles que lorsque le LCD est actif. On peut donc couper leur alimentation quand le produit est en veille, simplement en les alimentant non pas directement, mais via une sortie du  $\mu$ C. Petite modification, mais très gros impact, l'autonomie prévisionnelle passe de 1500 jours à 2 391 jours...

### Réduire le courant de veille

Dernière piste bien sûr : supprimer les  $\mu$ A consommés lorsque le produit est en veille. Coupez tout ce qui peut être coupé. On peut aller loin dans cette démarche. Par exemple sur le SmartyCook, le  $\mu$ C est tout le temps alimenté. Pourquoi ? Pour qu'une simple pression sur un bouton le réveille. Une petite astuce classique permet d'avoir le même comportement tout en coupant complètement l'alimentation du circuit en veille. Regardez donc la **figure 4**, c'est le schéma optimisé du SmartyCook. J'ai ajouté un régulateur linéaire doté d'une entrée « shutdown » (U2, MAX1725). Une pression sur le bouton active cette entrée, et donc l'alimentation du  $\mu$ C. Ce dernier active alors par logiciel sa sortie RD2, ce qui maintient le régulateur allumé et ce via la fonction « OU » réalisée par D1 et D2. Le circuit fonctionne alors normalement tant que le  $\mu$ C ne remet pas à zéro la broche *shutdown* par RD2 ; sinon, cela coupe l'alimentation, « suicide » le circuit et garantit un courant de repos quasi nul jusqu'à la prochaine pression du bouton.

### Conclusion

Si vous comparez le schéma de départ à celui de la figure 4, les modifications sont assez mineures : quelques changements de valeurs ici ou là, quelques optimisations ponctuelles, et une alimentation légèrement plus complexe. Par contre l'autonomie théorique de l'appareil a radicalement changé. Regardez le nouveau bilan énergétique (**tableaux 5 à 7**) et comparez-le à l'initial : l'autonomie du produit est passée de 260 jours à 3 728 jours, soit de huit mois à plus de dix ans ! Enfin, du moins si l'autodécharge des piles est suffisamment faible. Évidemment cet exemple a été conçu

**Tableau 5. Consommation en  $\mu$ A pour chacun des modes de fonctionnement après optimisation.**

Mode	MAX-1725	PIC-16F914	LCD	R1/R2/R3	R4/R5/R6/RT1	HP	Total
VEILLE	0,700						0,700
ACTIF	2,000	6,000	80,000	1,418			89,418
MESURE	2,000	6,000	80,000	1,418	36,364		125,782
BIP	2,000	8,000	80,000	1,418		250,000	341,418

**Tableau 6. Consommation énergétique moyenne quotidienne après optimisation, exprimée en mA.h.**

Mode	Durée (s/j)	Courant instantané ( $\mu$ A)	Énergie (mA.h)	% total
VEILLE	84600	0,700	0,016450	24,6%
ACTIF	1800	89,418	0,044709	<b>66,8%</b>
MESURE	2	125,782	0,000063	0,1%
BIP	60	341,418	0,005690	8,5%
Total par jour (mA.h)			<b>0,066912</b>	100,0%

**Tableau 7. Autonomie après optimisation.**

Capacité (mA/h)	Théorique	Projet	Variation (estimation de l'auteur)
Capacité théorique (mA .h)	464		
Tension mini (V)	1,0	1,2	20%
Durée de vie (années)	0,1	2,0	20%
Variation entre fournisseurs			20%
Pics de courant (mA)	16,0	10,0	-5%
Température mini (°C)	20,0	20,0	0%
Capacité estimée (MA.h)			249
Consommation quotidienne (mA.h)			0,067
Autonomie estimée (j)			<b>3 728</b>

pour être très démonstratif, mais j'espère que cela vous a convaincu qu'il est souvent possible de « mieux faire » sur le plan énergétique. J'ai aussi essayé de vous montrer que même si les astuces restent assez élémentaires, une telle optimisation doit être réalisée de manière méthodique et systématique. Grâce à ce genre d'approche, de nombreux produits industriels présentent une autonomie remarquable. J'ai entre autres travaillé sur plusieurs projets de compteurs d'eau ou de gaz : ces appareils sont alimentés par piles et ont des autonomies réelles de 15 ou 20 ans, tout en transmettant plusieurs messages radio par jour avec des portées de quelques kilomètres... Comme quoi, c'est possible. Voilà, à vous de jouer ! Je vous recommande de reprendre votre dernier pro-

jet et de tenter d'appliquer la petite méthode proposée dans cet article, peut-être pourrez-vous l'améliorer de manière décisive ? Dans le prochain article, je parlerai en détail d'une autre technique très utile pour l'optimisation énergétique : les convertisseurs à découpage. ◀

(160633)

Cet article a été publié dans la revue Circuit Cellar (n°213, avril 2008).

### Lien

- [1] Feuille de caractéristiques du PIC16F914, Microchip : [www.microchip.com/wwwproducts/en/PIC16F914](http://www.microchip.com/wwwproducts/en/PIC16F914)



**electronica**  
**fast forward**  
powered by elektor

the startup platform

PROPULSEZ VOTRE **PROJET NOVATEUR**

SUR

**LE MARCHÉ  
INTERNATIONAL !**

● Participez à l'édition 2018  
du concours

du 13 au 16 novembre 2018  
à Munich

Les prix du concours electronica fast forward sont offerts par

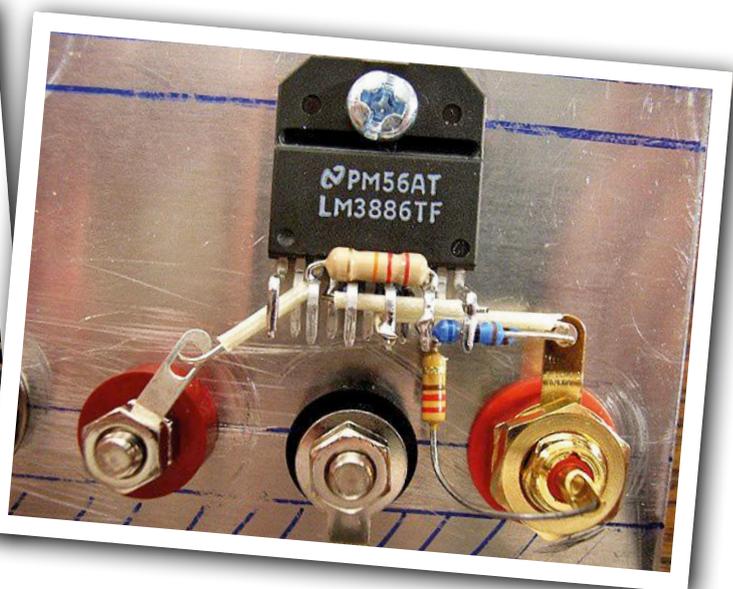
Informations détaillées :  
[www.elektormagazine.com/e-ffwd](http://www.elektormagazine.com/e-ffwd)



**electronica**



**elektor**



# son *chipamp* à soi

## un meilleur son avec moins de composants

Construire soi-même et fignoler ses appareils audio, c'est le passe-temps de beaucoup d'audiophiles. Ensuite ils ont le plaisir d'écouter de la musique sur leur installation. On trouve sur la Toile nombre d'amplificateurs réalisés avec un minimum de composants et qui pourtant sonnent très bien.

**Harry Baggen** (labo d'Elektor)

Tous les amateurs de musique ne sont pas des techniciens, loin de là. Ils cherchent avant tout la meilleure reproduction sonore possible, les données techniques ne les intéressent pas. Ils croient leurs oreilles beaucoup plus que les chiffres de caractéristiques. Ce qui a parfois conduit le monde de l'audio à proposer des produits délirants, tels que des soubassements en bois spéciaux pour les câbles des enceintes acoustiques...

Lors du choix de composants pour un ampli audio, on peut se demander pourquoi certains produiront un meilleur son que d'autres. Le choix du type de l'ampli est aussi un grand sujet de discussion. Lors de mes recherches sur l'internet, j'ai découvert de nombreux sites qui traitent des *chipamps* ou *gainclones*, de simples amplis de puissance faits maison, basés sur un circuit intégré. Il est surtout question de puces de la série *Overture*™ de

National Semiconductors, actuellement Texas Instruments.

La popularité de ces amplis, développés pour des applications audio générales et pas spécialement pour la hi-fi (oublions le haut de gamme), a commencé en 1999 avec la sortie de l'ampli 4706 (*Gaincard*) du fabricant japonais 47 Laboratory [1]. Son objectif est de fournir une reproduction la plus pure possible avec le minimum de composants sur le chemin du signal. Cet ampli ne compte que neuf composants par canal, il opère à partir d'un LM3875, une puce de quelques euros. Dans plusieurs revues, le 4706 a été et est encore estimé au prix astronomique de quelques milliers de dollars. Et quelques autres fabricants de matériel audio l'ont incorporé à leurs amplificateurs.

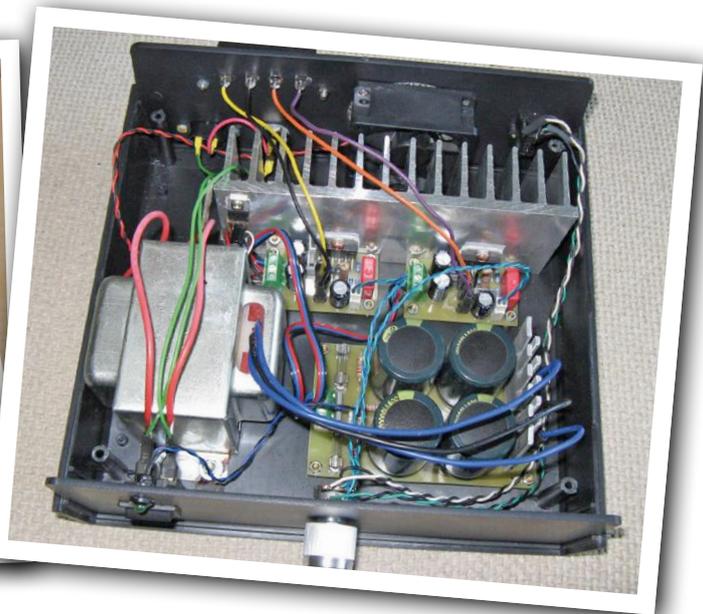
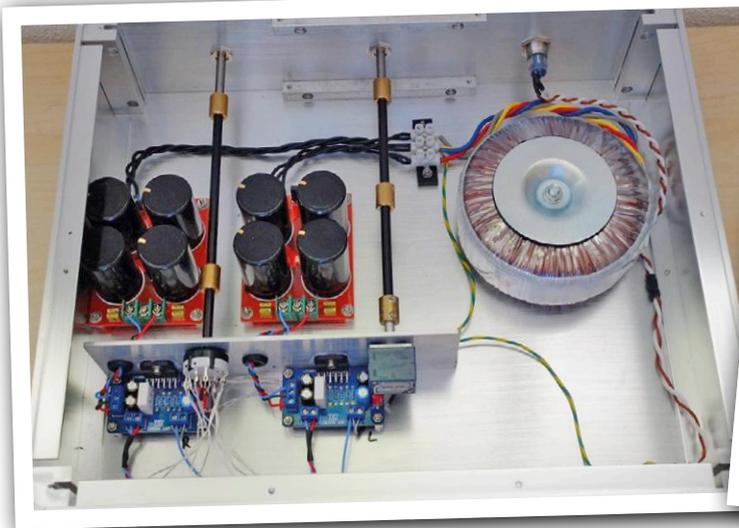
Cela a encouragé de nombreux amateurs à expérimenter avec des puces de la série *Overture*. Il y a même des magasins en ligne qui ne proposent que des composants et des circuits imprimés pour ces

puces. Des fabricants chinois s'y sont mis et proposent sur eBay une carte complète avec un LM3886 [2] pour moins de 10 \$, le plus puissant ampli de la série *Overture*, capable de délivrer près de 70 W sur 4 Ω. Les alimentations correspondantes sont aussi disponibles à des prix ridiculement bas. À ce tarif-là, on devine la qualité des composants utilisés...

L'attrait de ce type d'ampli de puissance simple est qu'on peut facilement faire quantité d'essais avec différentes sortes et qualités de composants. On ajoute un transformateur, un pont redresseur et quelques électrolytiques et en avant la musique ! C'est l'occasion de vérifier la différence avec d'autres types d'électrolytiques, d'en mettre en parallèle tout un tas, de changer le condensateur d'entrée, etc.

Vous trouverez sur différents sites d'intéressantes descriptions de *chipamp*, dont le LM3886. Par exemple, sur [3] Gainclone avec la puce d'ampli LM3886

▶ Que d'expériences à mener avec un ampli de puissance aussi simple !



ou sur [4] « The Beast », LA bête, avec le LM3875.

Mick Feuerbacher prouve dans une large série de photos sur son site Dogbreath [5] qu'on peut construire tout cela sans circuit imprimé.

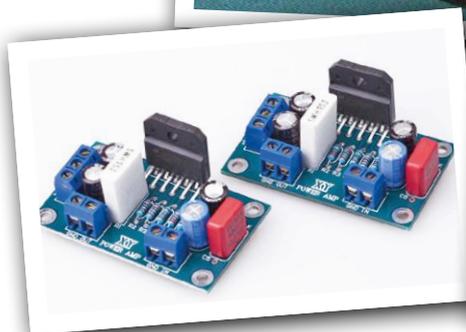
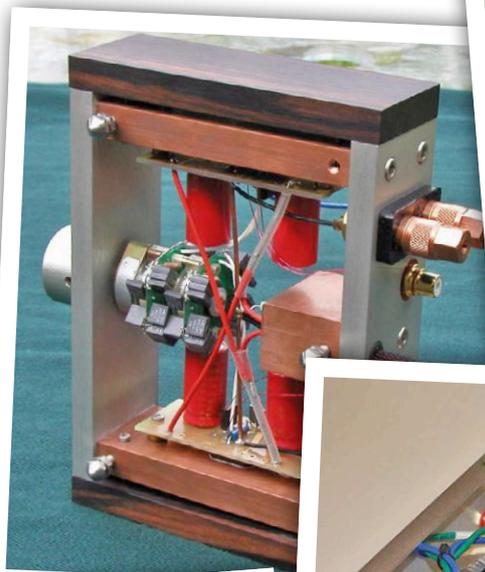
Bien entendu, Elektor a aussi publié le projet *amplificateur 100 W mono-circuit* (984062-I) avec le LM3886 dans le numéro de l'été 1998 [6] quand le circuit intégré est apparu sur le marché, et même avec le circuit imprimé !

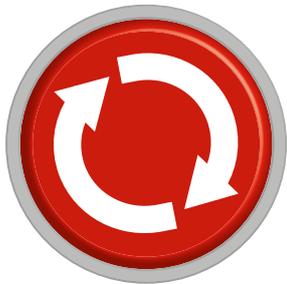
#### Liens

- [1] [www.sakurasystems.com/reference.html](http://www.sakurasystems.com/reference.html)
- [2] [www.ti.com/product/lm3886](http://www.ti.com/product/lm3886)
- [3] [http://diyaudioprojects.com/Chip/LM3886\\_CA/LM3886\\_CA.htm](http://diyaudioprojects.com/Chip/LM3886_CA/LM3886_CA.htm)
- [4] <http://diyaudioprojects.com/Chip/Beast/>
- [5] <http://dogbreath.de/>
- [6] <https://www.elektormagazine.fr/magazine/elektor-199807/36851>

Il y a encore bien d'autres variantes à découvrir sur l'internet, il suffit de chercher. En définitive, il s'agit toujours de projets avec une puce unique et, soudés dessus, quelques composants passifs. Vous pensez peut-être que ce n'est pas possible. Bah ! Des goûts et de la qualité sonore, on ne discutera point ! ◀

(160261 - version française :  
Robert Grignard)





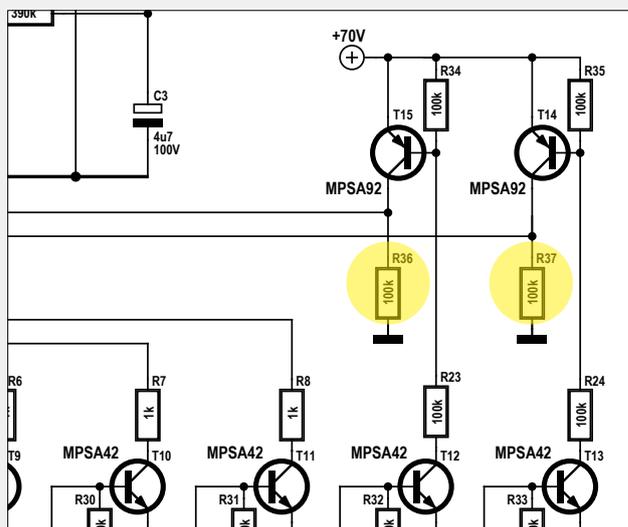
# projet 2.0

## corrections, mises à jour et courrier des lecteurs

### LEditron

Elektor 04/2016, p. 30 (150448)

Lors de la conception du circuit, nous avons remarqué que lorsque le transistor MPSA92 est bloqué, un petit courant de fuite circule malgré tout. Toutefois il est suffisant pour qu'un filament de LED brille très faiblement. Pour éviter cela, dans la version définitive du circuit, nous avons ajouté deux résistances de 100 kΩ (R36 et R37) entre les bornes d'anode communes de l'afficheur et de la masse, de sorte que le courant de fuite circule dans ces résistances et non dans l'afficheur. Ces résistances sont optionnelles et ne doivent être soudées que si la lumière due au courant de fuite est gênante. Malheureusement, les deux résistances n'étaient pas encore incluses dans la première version de la carte proposée au téléchargement. Le dessin du circuit imprimé a été modifié en conséquence et peut maintenant être téléchargé sur la page du projet ([www.elektormagazine.fr/150448](http://www.elektormagazine.fr/150448)). Le kit contient le bon circuit imprimé.



### comprendre l'amplificateur à transistor

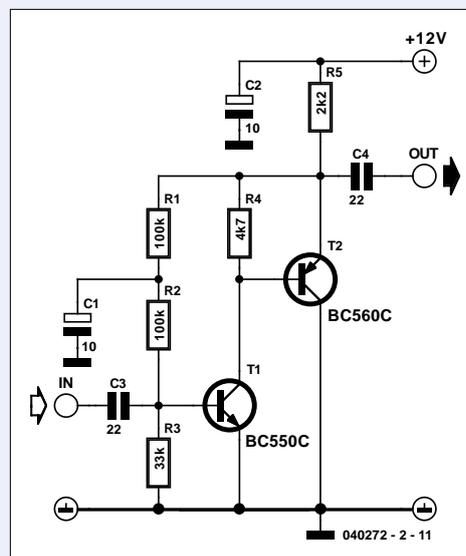
Elektor 09/2015, p. 24 (150403)

Dans cet article, il est écrit que le gain en tension d'un étage à transistors (dans le montage à émetteur commun) est toujours inférieur au paramètre  $h_{FE}$  du transistor. Ce n'est pas tout à fait vrai, comme on peut le voir avec l'amplificateur des « secrets du concepteur » du 10/2004 et du 11/2004 (gain d'environ 1800). Le gain en tension est égal en première approximation à  $S \cdot R_a$  ( $S$ =pente,  $R_a$ =résistance de collecteur). Avec une source de courant comme « résistance de collecteur », on peut donc atteindre des gains très élevés. Cette méthode est souvent utilisée dans les ampli-ops.

Martin Ossmann

Cher lecteur, merci beaucoup pour votre commentaire. Vous avez raison : le paramètre  $h_{FE}$  d'un transistor limite son gain en courant, mais ne peut pas limiter son gain en tension. Encore un sujet à explorer...

Robert Lacoste, auteur



### SDR d'Elektor réinventé (4)

Elektor 01/2017, p. 88 (160165)

Dans l'explication des figures 5 à 7, Burkhard Kainka écrit qu'à 16 kHz, le condensateur de 1 nF constitue avec la résistance de 10 kΩ un déphaseur de 90°. Un filtre passe-bas ou passe-haut simple ne produit, pour sa fréquence à -3 dB, qu'un déphasage de 45°. Pour une fonction correcte, il faudrait donc en connecter au moins deux en série. Au sujet du traitement numérique des signaux IQ, dont il est question dans le dernier paragraphe de l'article, j'ai rédigé il y a quelques années un article Wiki :

[www.mikrocontroller.net/articles/Hilbert-Transformator\\_\(Phasenschieber\)\\_mit\\_ATmega](http://www.mikrocontroller.net/articles/Hilbert-Transformator_(Phasenschieber)_mit_ATmega)

Un Arduino devrait suffire pour réaliser un déphaseur de 90° à large bande. Malheureusement mon premier essai s'est mis à osciller, ce qui n'est pas rare avec les filtres IIR et pour une précision de 16 bits.

*Christoph Kessler*

Bonjour Monsieur Kessler,

Il s'agit en l'occurrence d'un filtre passe-bas, car l'entrée inverseuse de l'ampli op est connectée à une masse virtuelle. Le circuit est un additionneur qui convertit les courants d'entrée en une tension de sortie. Mais pour les courants, on a affaire à un déphasage de 90° pour un large domaine de fréquences.

Merci pour votre référence au transformateur de Hilbert. Là, je dois avouer que j'approche mes limites. Mais s'il n'y avait personne pour convertir cela en choses utilisables par tous, il n'y aurait pas de logiciel SDR.

*Burkhard Kainka*

## shield IdO pour Arduino

Elektor 01/2017, p. 62 (160169)

Dans le deuxième paragraphe de l'article cité, lire « Entrée analogique 2 » au lieu de « Entrée analogique 3 ».



### UniLab 2

Elektor 11/2017, p. 14 (150161)

J'ai trouvé passionnant votre article sur Unilab 2 et je souhaite réaliser le projet prochainement. Mais je trouve que la façon de connecter l'amplificateur opérationnel non utilisé n'est pas la bonne. Voici une référence sur la manière de connecter des amplificateurs opérationnels non utilisés :

[https://e2e.ti.com/blogs\\_/archives/b/thesignal/archive/2012/11/27/the-unused-op-amp-what-to-do](https://e2e.ti.com/blogs_/archives/b/thesignal/archive/2012/11/27/the-unused-op-amp-what-to-do)

*Björn Schuster*

Bonjour Monsieur Schuster,

N'importe quel petit décalage à l'entrée d'un amplificateur opérationnel envoie la sortie au niveau haut ou bas. S'il s'agissait d'un ampli op à décalage ultra-faible, il serait très proche de l'ampli op idéal, ce qui pourrait alors poser d'éventuels problèmes. Mais ici, il n'y a pas lieu de s'en inquiéter. Mais d'une manière générale, le lien indiqué ci-dessus est certainement à prendre en considération.

*Ton Giesberts (labo d'Elektor)*



## gradateur à deux bornes

Elektor 07-08/2017, p. 14 (160380)

Dans la liste des composants, les valeurs pour R2 et R4 sont indiquées en kΩ, c'est une erreur. Les valeurs correctes sont respectivement 330 Ω et 560 Ω, comme indiqué sur le schéma. Notez que dans la version basse tension, le dimensionnement de R4 dépend de la tension d'alimentation. Voir les détails dans l'article.

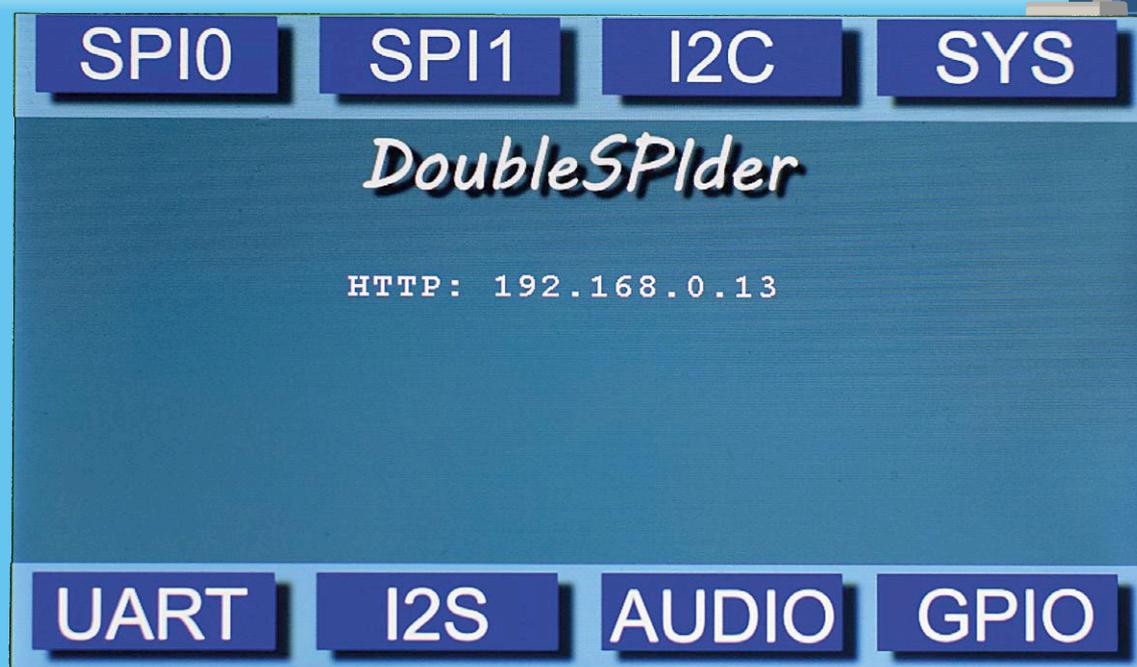
## Lumina – lampe connectée par Bluetooth Low Energy

Elektor 12/2015, p. 56 (130226)

- Il n'y a pas assez de place sur la carte pour le quartz. Ce composant doit être soudé sous la carte, ou alors il faut utiliser l'oscillateur interne du microcontrôleur.
- Étant donné que sous Android 7, il faut autoriser la localisation pour établir la connexion Bluetooth, nous avons ajouté deux lignes au fichier *AndroidManifest.xml* et changé son numéro de version en 1.0a. Deux autres lignes ont été ajoutées au fichier *DeviceScanActivity.java* pour la gestion de cette permission. L'ensemble du code a été recompilé sous Android Studio (version la plus récente) ; l'application fonctionne maintenant aussi sous Android 7.0. Les nouveaux fichiers sont disponibles au téléchargement sur la page du projet : [www.elektormagazine.fr/130226](http://www.elektormagazine.fr/130226).

# DoubleSPIder

adaptateur universel d'interfaces  
pour projets à microcontrôleurs



Torsten Jaekel (États-Unis)

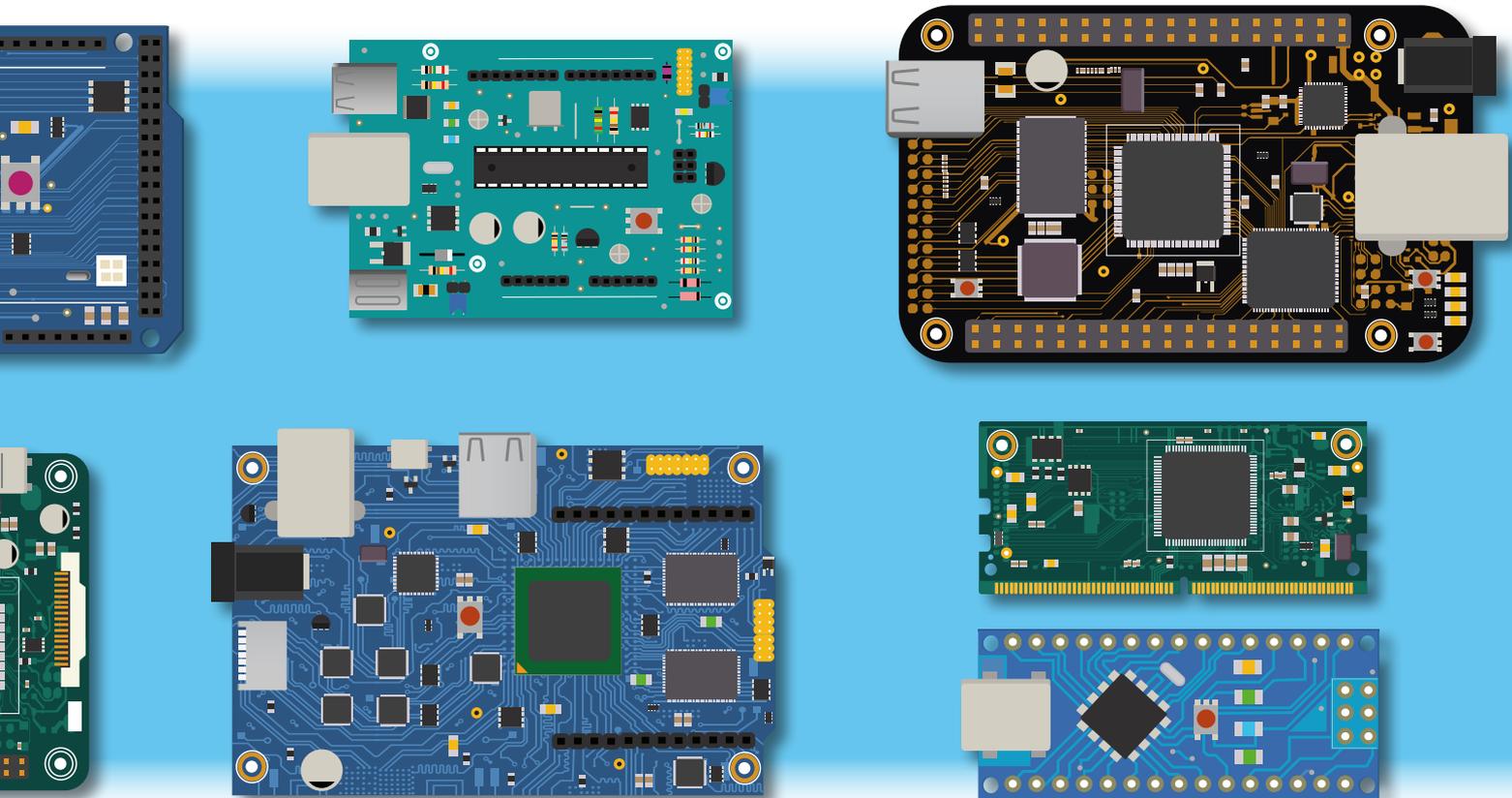
Quand on veut évaluer un nouveau microcontrôleur ou un module périphérique, on se demande souvent comment le piloter par un bus SPI ou I<sup>2</sup>C ou une interface de type UART, ou bien comment y accéder depuis un PC. Comme les PC et les portables ne sont pas équipés de ce type de port, on se met à la recherche de l'adaptateur approprié. S'y ajoute l'installation ou l'adaptation de pilotes. Avec DoubleSPIder, le problème se simplifie, car il offre une capacité d'interconnexion quasi universelle.

DoubleSPIder est un traducteur d'interface, un adaptateur externe, qui fournit trois ports série standards, usuels dans les projets à microcontrôleur. Et il peut bien plus encore : on peut le connecter directement à un réseau TCP/IP, sans nul

besoin d'installer un pilote. Pour cela, l'adaptateur contient un serveur web intégré (**fig. 1**). De cette manière, à partir du navigateur d'un PC, on peut facilement envoyer des commandes ou des paquets de données à un microcontrôleur par

## DoubleSPIDER : faits & chiffres

Base matérielle :	carte de développement STM32F769I-DISCOVERY à µC ARM Cortex-M7
Logiciels :	pilote HAL de STM, FreeRTOS, lwIP ; projet avec micrologiciel à code source ouvert
SPI :	3×, dont 1× isolé + 2× avec <i>Slave Select</i> ; brochage Aardvark
I <sup>2</sup> C :	3× (bus)
Interruptions :	2× (à la place de I <sup>2</sup> C)
GPIO :	8× configurables en entrée ou sortie, dont 1× à drain ouvert et 1× UART (RX, TX)
Réseau :	RJ45 Ethernet, 10 MBit/s, serveur web, serveur REST-API pour Python sur port TCP 8080, DHCP ou adresse IP statique
Interfaces utilisateur :	afficheur à cristaux liquides 800×400 tactile, interpréteur de commande UART, interpréteur Pico-C, serveur web
USB :	USB audio pour les données ou port de mémoire externe pour carte SD
Carte SD :	système de fichiers FAT jusqu'à 4 Go, noms de fichiers longs



les ports SPI, I<sup>2</sup>C ou UART.

Et ce n'est pas tout : il y a un « interpréteur de ligne de commande UART » avec lequel on peut envoyer des commandes. Et il y a un port TCP supplémentaire ; il permet entre autres d'effectuer des tâches de configurations complexes de puces externes, avec des scripts Python sur le PC hôte, ou bien de transférer dans les deux sens, des données quelconques entre le PC et un composant externe, via un bus SPI ou I<sup>2</sup>C ou un port série.

### L'équipement de DoubleSPIDER

Maintenant que sont clairement posés la raison d'être et le but de DoubleSPIDER, venons-en à sa réalisation concrète. Pour le matériel, DoubleSPIDER repose sur une carte d'évaluation « prête à l'emploi » basée sur une puce tout-en-un ARM

à 32 bits de ST (nous y reviendrons), augmentée de connecteurs et de quelques composants électroniques montés sur une carte d'extension. Cela simplifie notablement le montage et offre une bonne dose de confort et de puissance. De fait, l'adaptateur d'interfaces offre bien plus que trois ports série :

- 3× SPI, où un port SPI est combiné à deux signaux de sélection de puce (*slave select*). L'assignation des broches est compatible avec les adaptateurs SPI Aardvark de Totalphase [2].
- 3× I<sup>2</sup>C, bus avec trois connexions, des cavaliers optionnels pour l'activation de résistances de rappel. Les signaux I<sup>2</sup>C peuvent également servir de déclencheurs d'interruption.
- 1× USART, broches RX et TX avec des niveaux TTL.
- 8× GPIO, configurables à volonté en entrée ou sortie. L'une des E/S est configurable en sortie à drain ouvert,

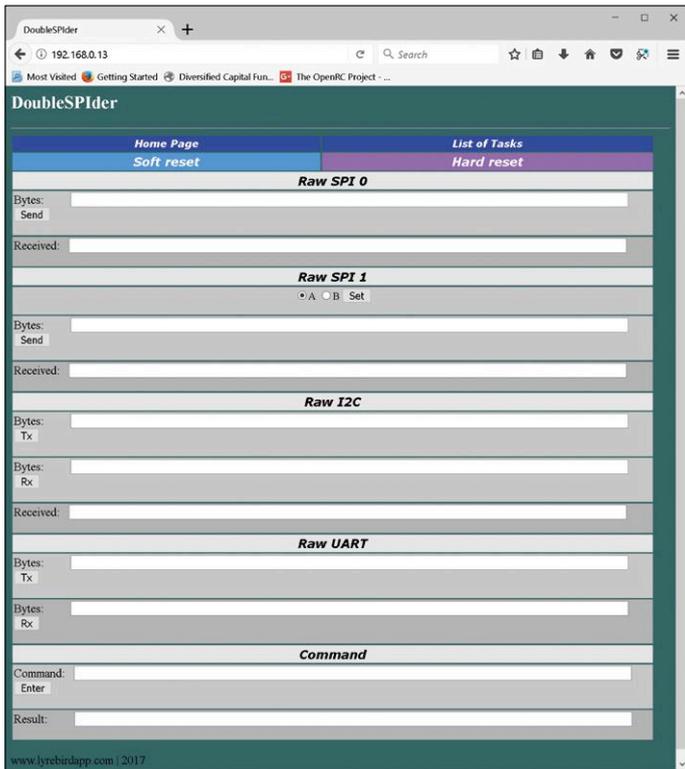


Figure 1. Voici comment le serveur web intégré s'annonce sur le navigateur du PC hôte.

utile par ex. pour réinitialiser une carte connectée.

- 1× USB 2.0, pour être utilisé sans pilote comme carte son.
- 1× emplacement pour carte SD pour stocker de grosses quantités de données
- 1× port Ethernet pour la communication avec le serveur web intégré
- 1× écran tactile de 4 pouces pour la configuration et l'utilisation.

La **figure 2** résume toutes ces caractéristiques et possibilités d'emploi.

Il est évident que DoubleSPIDER est un adaptateur universel haut de gamme pour les ports série SPI, I<sup>2</sup>C, USART, doté d'entrées/sorties à usage général et d'une connexion réseau ; outre le test de microcontrôleurs et de cartes de développement, il permet le raccordement direct de périphériques comme des capteurs série. Depuis un PC avec un navigateur web, il est possible d'accéder à tous les ports.

### Configuration tactile

C'est pour la configuration de l'adaptateur que l'écran tactile montre son intérêt. Celui de DoubleSPIDER rend la configuration manuelle particulièrement simple, en donnant accès à tous les paramètres : le mode SPI, les vitesses des bus SPI et I<sup>2</sup>C ou les directions des broches GPIO sont facilement configurables. Le micrologiciel de l'unité centrale de DoubleSPIDER offre une interface graphique (GUI) sur l'afficheur à cristaux liquides

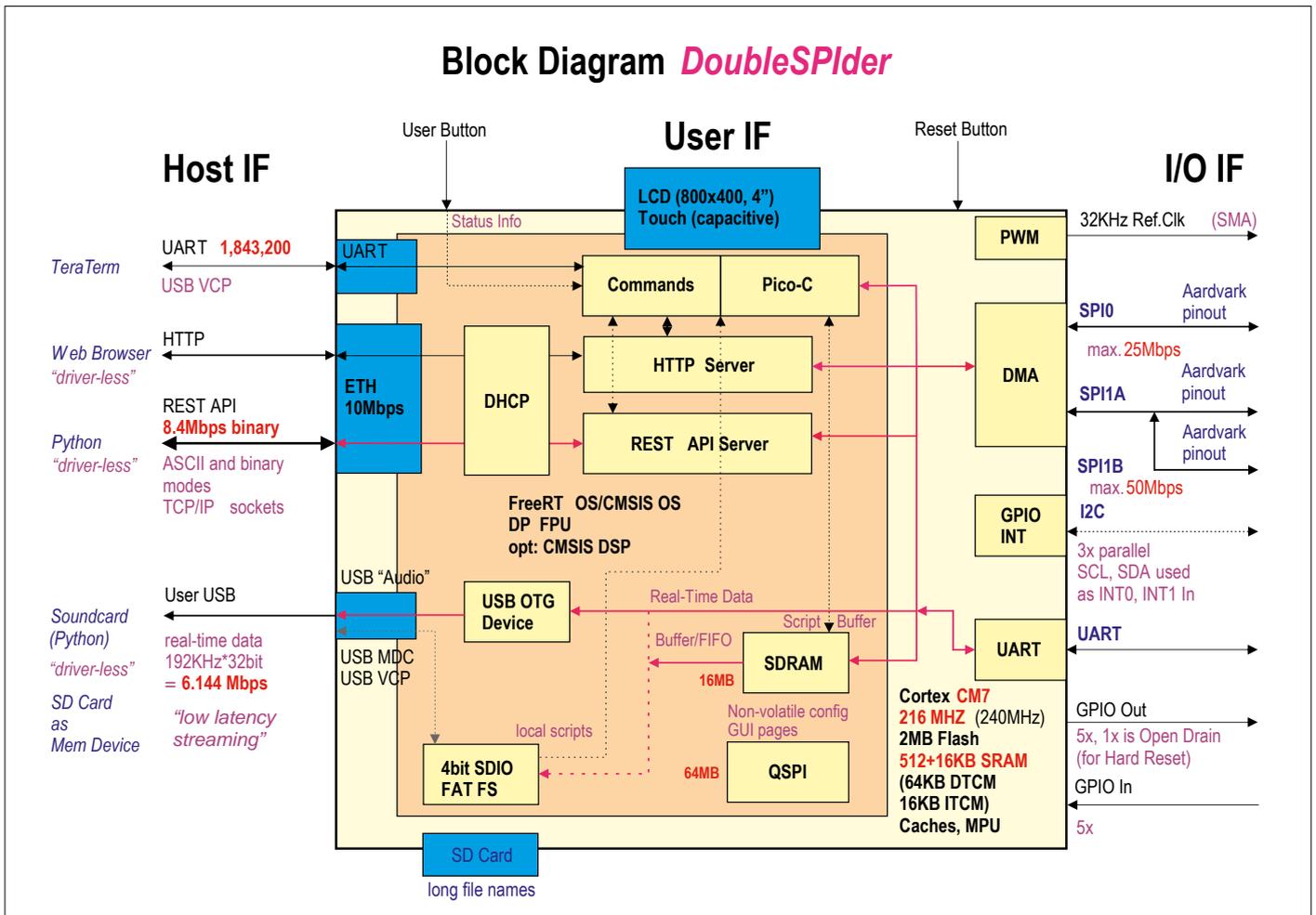


Figure 2. Schéma de principe et connexion du kit Discovery de ST.

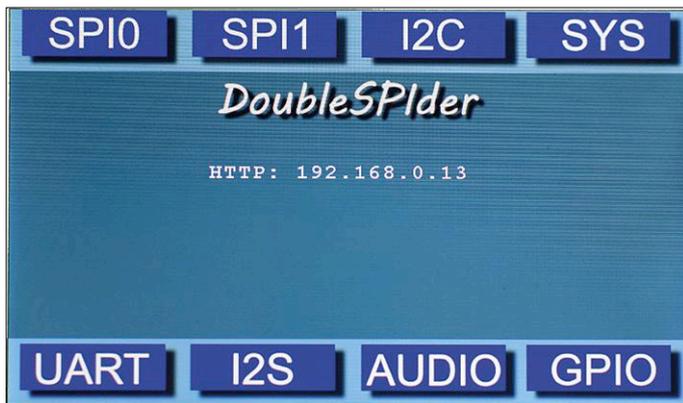


Figure 3. Interface utilisateur sur le clavier tactile de 4" de DoubleSPIDER.

(fig. 3). Tous les réglages peuvent s'effectuer par saisie tactile. Une fois saisis, tous les réglages peuvent être sauvegardés (en option) sur la puce mémoire QSPI et servir automatiquement de configuration de démarrage lors d'une réinitialisation ou de la mise sous tension suivante.

La programmation de l'interface utilisateur est simple. Les éléments graphiques sont des images prédéfinies au format BMP (fig. 4), stockées sur la puce mémoire QSPI. De ce fait, il reste suffisamment de place dans la mémoire flash interne du microcontrôleur pour permettre à l'utilisateur d'ajouter le code de fonctions supplémentaires. Le seul inconvénient de l'absence d'un gestionnaire ou d'un wizard graphique est que cela complique le traitement des événements tactiles (comme les boutons-poussoirs sur l'écran) puisqu'il faut décoder les coordonnées du point de contact.

Outre le réglage de nombreux paramètres et fonctions, on peut aussi afficher l'état des signaux GPIO sur l'écran et modifier l'état des broches GPIO configurées en sorties. Toutefois la production de commandes I<sup>2</sup>C ou SPI, ou de paquets de données n'est pas possible sur l'afficheur. Ces fonctions doivent être effectuées au moyen de l'interpréteur de ligne de commande UART, de l'interface web ou de scripts Python à travers le port TCP.

### Utilisation des scripts

La caractéristique essentielle de DoubleSPIDER repose sur l'utilisation de scripts, écrits par ex. en Python, sur un PC hôte. Avec ces scripts, on peut configurer les puces ou les cartes connectées à DoubleSPIDER, les télécommander, échanger des données, lire des mesures de capteurs, et bien d'autres choses encore. Outre l'interpréteur de ligne de commande UART, utilisable de manière interactive avec un émulateur de terminal (par ex. TeraTerm), DoubleSPIDER contient un interpréteur Pico-C. On peut donc saisir directement des petits programmes en C, lequel est interprété directement, sans conversion ; aucune passe de compilateur n'est donc nécessaire. On peut ainsi programmer directement en C des fonctions exécutables telles qu'elles dans DoubleSPIDER. La figure 5 montre de quelle manière on peut manipuler directement des interfaces sérielles au moyen d'un programme en C.

Les scripts Python, exécutés sur le PC hôte, peuvent également envoyer des commandes par l'UART ou utiliser l'interpréteur Pico-C. Et encore bien mieux : ces scripts peuvent utiliser directement le port dédié 8080. En passant par des interfaces

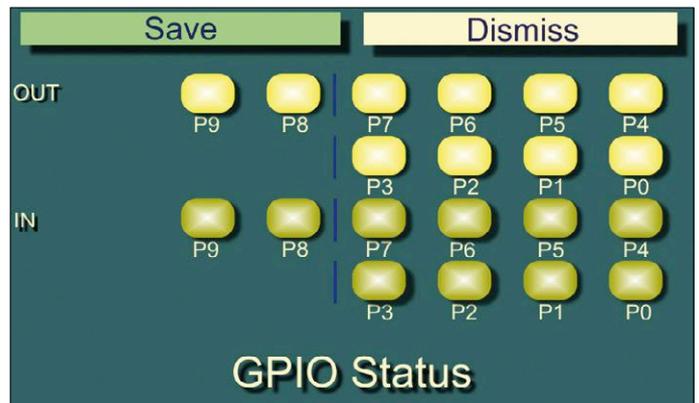


Figure 4. Lecture et changement de l'état des entrées/sorties à usage général.

réseau (sockets), les scripts Python peuvent échanger des commandes et des données à plus haute vitesse ; un mode binaire est même prévu.

Pour optimiser le traitement d'authentiques interruptions (quand les signaux I<sup>2</sup>C sont configurés en déclencheurs d'interruptions), un « accélérateur Python » fonctionne comme un automate à états finis programmable. On peut programmer les interruptions pour qu'elles lancent des actions spécifiques, ce qui permet un traitement accéléré des événements critiques et un allègement de la charge des scripts Python. Sans un accélérateur Python, le PC hôte devrait scruter en permanence les broches concernées – une tâche vraiment laborieuse. Il faudrait alors qu'un registre d'interruption soit continuellement lu par SPI pour voir si un changement s'est produit. Il faut éviter autant que possible la scrutation, car elle est lente, consomme de la puissance de calcul et encombre les voies de communication. Cela s'applique en particulier lorsqu'on utilise des scripts qui doivent s'exécuter en arrière-plan, ce qui complique encore la programmation.

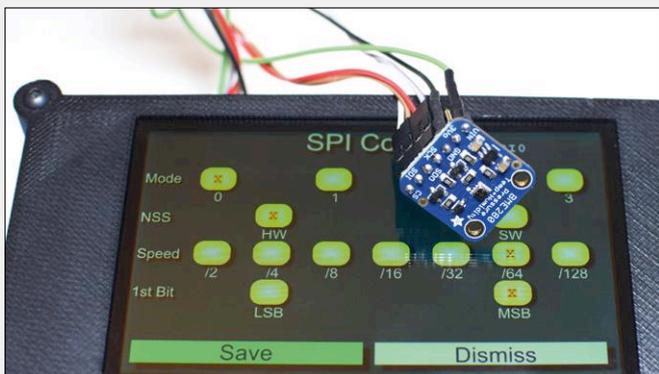
```

COM6 - Tera Term V1
File Edit Setup Control Window Help
*** DoubleSPIDER - V1.6.1 (Jul 31 2017) ***
> pico
***** PICO-C command interpreter *****
(version: 3.1, Jul 31 2017)
: CHelp();
void printf(char *,...);
void print_log(char *,...);
void perror_log(char *,...);
void display(char *,...);
char *sprintf(char *,char *,...);
char *format(char *,...);
void sets(char *,int);
int getchar();
void exit(int);
void malloc(int);
void free(void *);
void strcpy(char *,char *);
void strncpy(char *,char *,int);
int strcmp(char *,char *);
int strncmp(char *,char *,int);
void strcat(char *,char *);
char *index(char *,int);
char *rindex(char *,int);
int strlen(char *);
void memset(void *,int,int);
void memcpy(void *,void *,int);
void memcmp(void *,void *,int);
void quit();
void uvw_info(char *,char *msg,int verb);
void CHelp();
void Pshorts(void *,unsigned long);
void PBytes(void *,unsigned long);
int RunScript(char *);
int ExitValue();
void msleep(unsigned long);
int SpiTransaction(int,unsigned char *,unsigned char *,unsigned long);
int I2CRead(unsigned short,unsigned short,unsigned char *,int);
int I2CWrite(unsigned short,unsigned short,unsigned char *,int);
int ReadReg(int,int,int*,int);
int WriteReg(int,int,int*,int);
int ReadBlock(int,int,int *,int);
int WriteBlock(int,int,int *,int);
int SysCmd(int);
int IrCmd(int,int *);
int IrEnable(int);
void UARTx(unsigned char *,int);
int UARTGetChr();
int i = 10;
printf("i ist: %d\n", i);
i = 10
  
```

Figure 5. Exemple de code pour utiliser des fonctions intégrées de Pico-C.

## BM280 : connexion d'un module capteur

Le BME280 est un module capteur d'environnement de Bosch Sensortec. Il contient des capteurs de température, humidité de l'air et pression atmosphérique, dans un boîtier LGA à 8 broches de 2,5 mm de côté. Comme cette puce est délicate à utiliser telle quelle, il existe des petites cartes d'adaptation qui facilitent les connexions ou le soudage. Cette puce peut être lue via I<sup>2</sup>C ou SPI. Dans cette démo, on utilise une telle carte (disponible dans l'e-choppe [6]) en mode SPI. Nous décrivons ci-après sa connexion à DoubleSPiDer et les différentes possibilités d'y accéder en lecture.



### Connecter le module BME280

Le module BME280 doit être connecté au port SPI2 (H1) de DoubleSPiDer. Pour plus de précision, le tableau suivant indique, outre la désignation des connexions des modules, les numéros des broches selon la feuille de caractéristiques du BME280.

module	broches	DoubleSPiDer
VIN	6 + 8	3,3 V, broche 4
GND	1 + 7	GND, broche 2 ou 10
SCK	4	SPI2_SCLK, broche 7
SDO	5	SPI2_MISO, broche 5
SDI	3	SPI2_MOSI, broche 8
CS	2	SPI2_NSS, broche 9

### Lire le code ID de la puce par SPI

On peut lire le code d'identification de la puce par une transaction SPI. Le premier octet contient l'adresse du registre 0xD0 (lecture et adresse sur 7 bits). Un second octet quelconque doit être envoyé pour récupérer le contenu du registre. Le second octet reçu devrait contenir la valeur 0x60. Dans les méthodes de lecture suivantes, il s'agit (en dehors des copies d'écran) de protocoles de terminal. Les saisies significatives sont représentées en bleu.

#### ...via UART

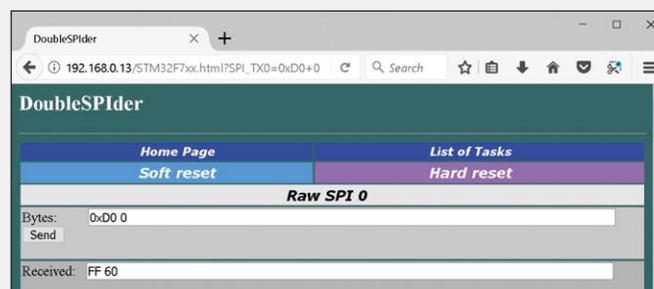
```
*** DoubleSPiDer - V1.1.1 (Jul 31 2017) ***
>
> spi0 0xd0 0
FF60
```

#### ...via Pico-C

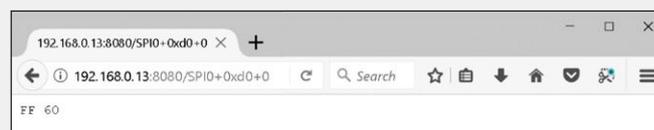
```
> picoc
```

```
***** PICO-C command interpreter *****
      (version: 3.1, Jul 31 2017)
: unsigned char spiTx[2] = {0xD0, 0};
: unsigned char spiRx[2];
: SpiTransaction(0, spiTx, spiRx, 2);
: printf(„BME280 ChipID: %x\n“, spiRx[1]);
BME280 ChipID: 60
```

#### ...via navigateur web



#### ...via serveur REST-API, port 8080



#### ...via script Python et serveur REST-API

```
#!/usr/bin/env python

#Python network access to DoubleSPiDer MCU board via REST-
API-IP: UDP port 8080
#using ASCII format (commands) on REST_API for SPI
transaction

import socket

def Main():
    host = input("Enter IP address shown on display: ")
    port = 8080

    message = 'GET /SPI0+0xD0+0'

    mySocket = socket.create_connection((host, port))
    mySocket.setblocking(1)
    mySocket.sendall(message.encode(encoding='UTF-8'))
    data = mySocket.recv(1024)

    #now print what we have received
    print(data)

    print('Good bye')
    mySocket.close()

if __name__ == '__main__':
    Main()
```

## Audio par l'USB

DoubleSPiDer possède également un port USB utilisateur, qui permet d'échanger facilement de grandes quantités de données. L'utilisation de l'USB sur un PC exige habituellement la présence de pilotes spéciaux. On peut l'éviter si l'on se limite à l'utilisation de pilotes présents par défaut sur chaque PC. C'est pourquoi le port USB de DoubleSPiDer est systématiquement configuré en interface audio. Cette astuce fonctionne sur tous les PC.

Quand le câble USB est branché sur le PC, ce dernier détecte un nouveau périphérique audio. Mais en réalité, ce ne sont pas des paquets audio qui sont transférés, mais des données quelconques, fournies par les puces et modules connectés. Cela pourrait être, par ex., des contenus de FIFO ou des lectures de capteurs, transmis en continu ou de manière cyclique. Le canal audio détourné en transmetteur de données est configuré en 16 bits, stéréo à 48 kHz et dix canaux. Cela autorise une taille maximale de paquets de 960 octets, transmissibles à des intervalles d'une milliseconde, soit un débit maximal de 960 Ko/s. Le grand avantage est que l'USB audio est rapide et avec des temps de latence très faibles.

Avec Python, il est facile de surveiller ce périphérique audio et de capturer les paquets USB, dont les données peuvent être décodées et traitées rapidement et en continu. La **figure 6** précise comment un script Python peut transférer ces données, acquises à l'origine en mode série, et les enregistrer en « USB audio ».

```

C:\Python27\Python.exe
Python 2.7.13 (v2.7.13:a664541af1, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import sounddevice as sd
>>> sd.default.device = 26
>>> sd.default.channels = 2
>>> sd.default.samplerate = 192000
>>> rec = sd.rec(5 * 192000, dtype = 'int16')
>>> with open('USB_recording', 'wb') as f:
...     f.write(rec)
...     f.close()
...
>>>
    
```

Figure 6. Script Python pour le port USB configuré en interface audio.

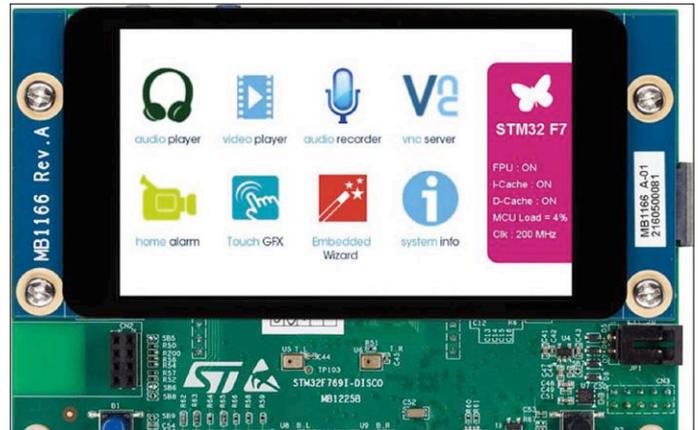


Figure 7. La carte d'évaluation avec le µC STM32F799I de ST coûte environ 80 €, y compris l'afficheur tactile de 4".

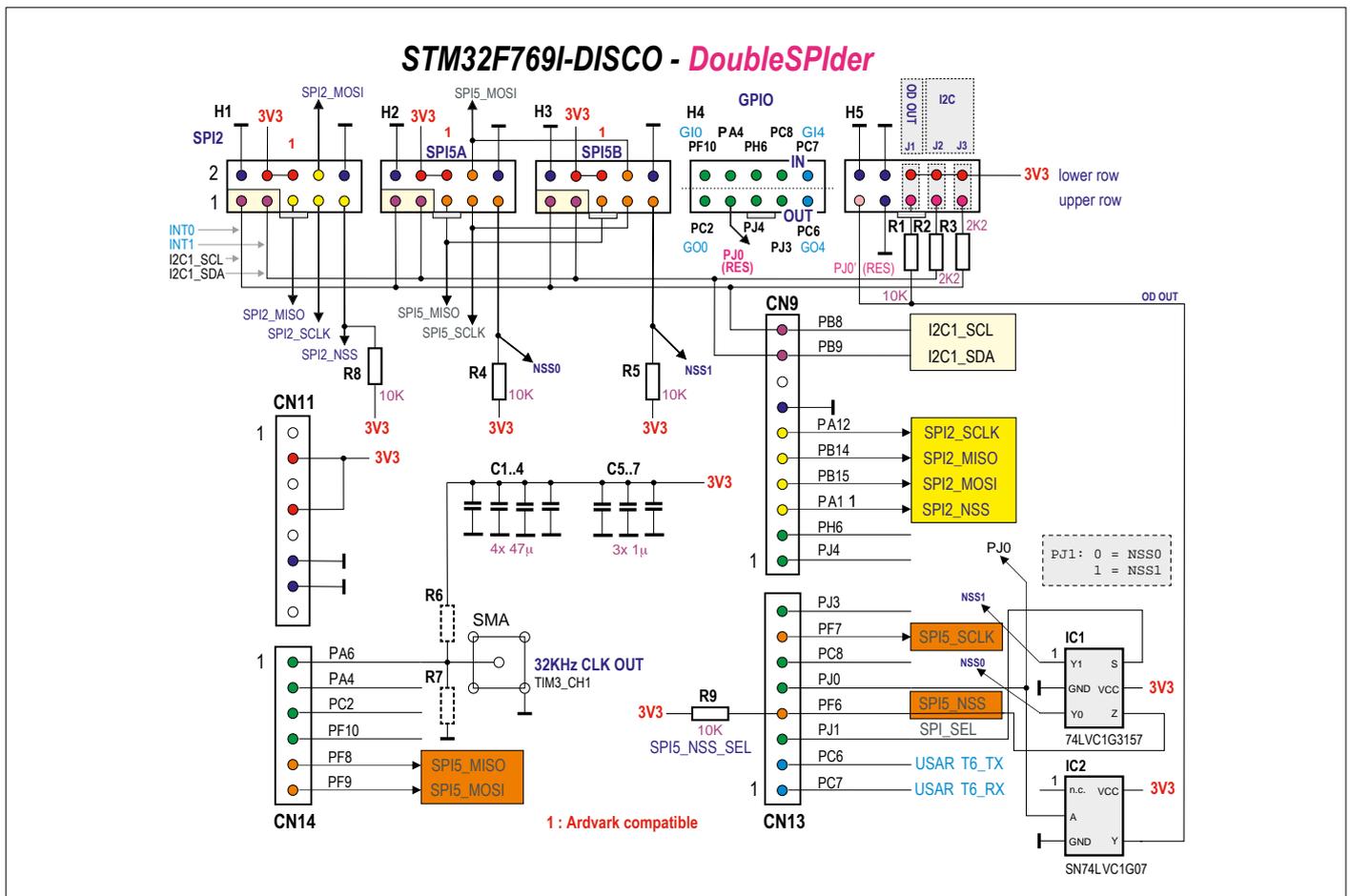


Figure 8. Le « schéma » de la carte d'extension est relativement simple : en dehors de deux puces et de quelques condensateurs et résistances, elle ne comporte que des connecteurs.

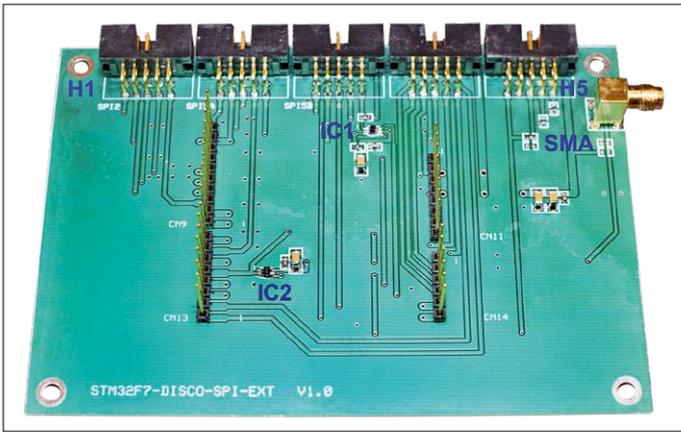


Figure 9. La carte d'extension terminée avec tous ses composants.

### De plus...

Ce projet inclut aussi un emplacement pour une carte SD. Un système de fichiers FAT permet d'y déposer des scripts Pico-C et de les démarrer. En outre, il est possible de lire ou d'écrire des données sur cette carte. Pour éviter de constamment retirer la carte pour la lire ou l'écrire sur le PC, le port USB peut

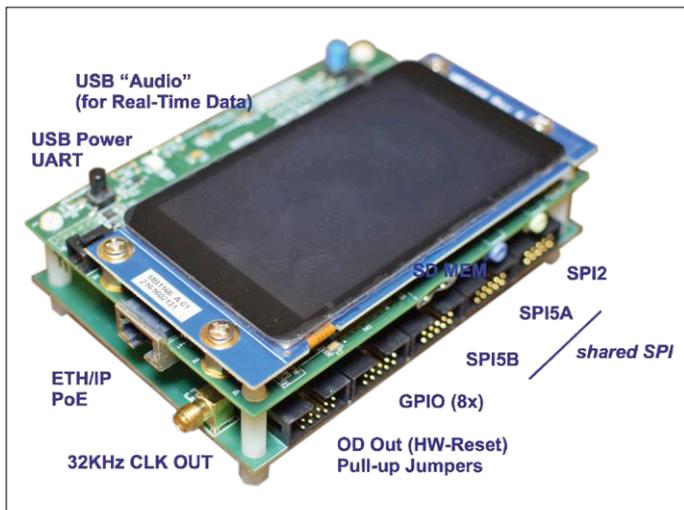


Figure 10. Association des deux cartes.



Figure 11. DoubleSPIDER dans son boîtier sur mesure.

être configuré en interface de mémoire externe, au lieu d'USB audio ; le contenu de la carte SD sera alors directement accessible depuis le PC. Vous pouvez ainsi facilement copier, manipuler ou effacer les scripts Pico-C ainsi que les autres données présentes sur la carte.

En outre, il est concevable de configurer les ports SPI en interfaces audio I2S, utilisables parallèlement à l'interface USB audio déjà disponible. On pourrait ainsi connecter un module audio du commerce du genre CA/N ou CN/A. On pourrait configurer la puce audio par SPI ou I<sup>2</sup>C, alors que les données qui circulerait sur l'audio I2S pourraient atteindre le PC par l'USB. Le micrologiciel de la carte à microcontrôleur est comme toujours à code source ouvert, disponible sur la page du projet sur le site web d'Elektor [3]. Il utilise le pilote libre (HAL) de STMicroelectronics, FreeRTOS comme système d'exploitation, et la pile réseau lwIP [4]. D'autres détails et documents sur DoubleSPIDER sont téléchargeables sous [3]. On y trouvera une documentation PDF contenant des instructions de connexion et d'utilisation.

Comme déjà mentionné, le projet DoubleSPIDER est basé sur une carte de développement de ST, en l'occurrence le « kit Discovery avec µC STM32F769NI » (fig. 7) complété par une carte d'extension comprenant les connecteurs et un peu d'électronique complémentaire. La figure 8 en donne le plan, la figure 9 l'aspect de la carte. On appréciera le résultat final sur la figure 10. L'auteur a également réalisé un boîtier sur mesure au moyen de son imprimante 3D. La figure 11 montre son prototype installé dans ce boîtier.

On trouvera les fichiers de conception du boîtier au format Google SketchUp, ainsi que les fichiers de la carte d'extension (au format du logiciel CAO libre ExpressPCB [5]), avec la liste des composants, le code et des exemples sous [3]. ◀

(160390 - version française : Helmut Müller)

### L'auteur

Torsten Jaekel est ingénieur système chez Broadcom Ltd. en Californie, depuis plus de dix ans.

Il y participe au développement, à la mise en service et au test de capteurs et d'organes de contrôle tactile.

### Liens

- [1] [www.st.com/en/evaluation-tools/32f769discovery.html](http://www.st.com/en/evaluation-tools/32f769discovery.html)
- [2] [www.totalphase.com/products/aardvark-i2cspi](http://www.totalphase.com/products/aardvark-i2cspi)
- [3] [www.elektormagazine.fr/160390](http://www.elektormagazine.fr/160390)
- [4] <http://savannah.nongnu.org/projects/lwip/>
- [5] [www.expresspcb.com](http://www.expresspcb.com)
- [6] [www.elektor.fr/bme280-mouser-intel-160109-2](http://www.elektor.fr/bme280-mouser-intel-160109-2)
- [7] [www.bosch-sensortec.com/bst/products/all\\_products/bme280](http://www.bosch-sensortec.com/bst/products/all_products/bme280)



Photo : Anselm Rapp, Wikimedia Commons.

Figure 1. Le magnétophone UHER 4000 Report-L de 1964.

# UHER 4000 Report-L (1964) fabuleux magnétophone à bande portatif

Karl-Ludwig Butte

Classer les magnétophones à bande au rang des dinosaures, c'est aller trop vite en besogne. En connaissez-vous des portables ? Avec la série des 4000 Report, UHER a réussi un chef-d'œuvre de technologie de précision. Par une utilisation judicieuse des transistors de l'époque et beaucoup de savoir-faire, sous un format compact, le Munichois a fait sortir dans la rue la technique magnétique de prise de son dès 1961.

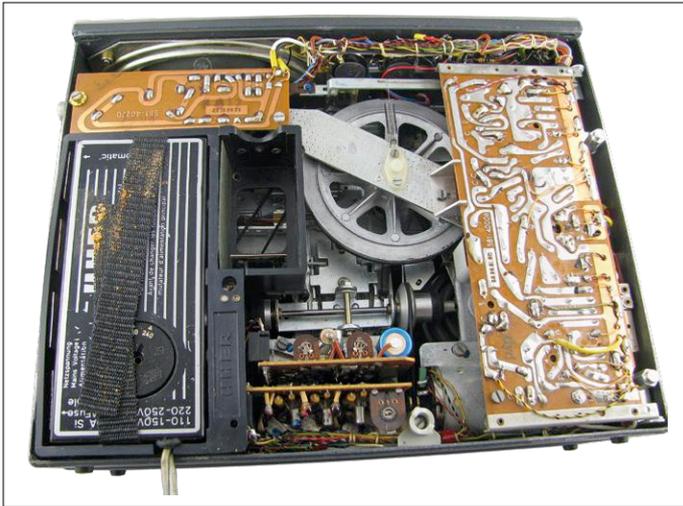


Figure 2. Un premier regard sous le capot du UHER 4000 Report-L.

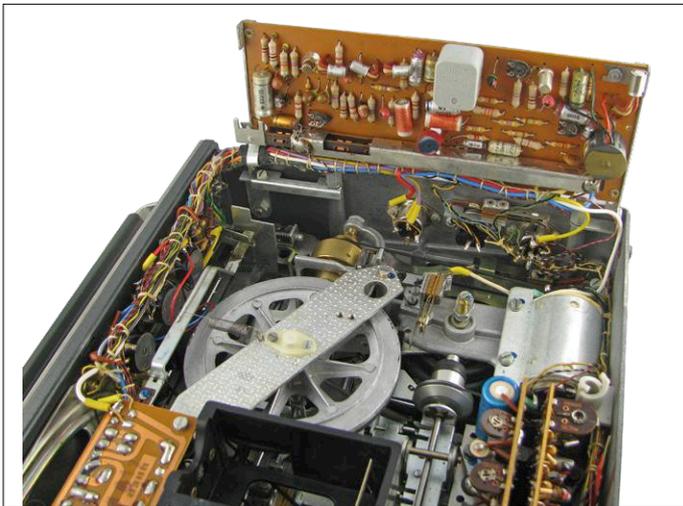


Figure 3. La carte d'amplification enregistrement/lecture est prévue pour être relevée et donner accès aux organes sous-jacents.



Figure 4. La fameuse bobine de démagnétisation.

Les magnétophones produits par UHER n'étaient pas seulement d'une qualité exceptionnelle, ils étaient aussi très chers. Quand j'étais jeune, c'était pour moi un rêve inaccessible. Récemment, j'ai pu me procurer un UHER 4000 Report-L à bon compte et je n'ai pas pu résister à l'envie d'examiner de plus près cette légende. La **figure 1** montre notre star avec son changement de vitesse inimitable qui identifie tout de suite le représentant de la famille des magnétophones. Les bobines sont protégées par un capot métallique avec une fenêtre en plastique. Des fermoirs à ressort les maintiennent sur les axes. Avec une sacoche en cuir adaptée, disponible séparément, pour porter l'appareil en bandoulière, on le transporte et on travaille en cours de route en toute liberté. Robuste, il a servi à de nombreux reportages, mais aussi dans les studios de radio, au service météo allemand et au FBI. Il a même voyagé dans l'espace avec la capsule Gemini. On le retrouve au cinéma aussi, notamment pour 007, l'agent secret de Sa Majesté, dans Opération Tonnerre [1].

### Sous le capot

Si vous êtes aussi avide que moi de découvrir le fonctionnement interne de cette machine, suivez-moi. Le dos de l'appareil n'est fixé que par une seule vis, on peut donc l'ouvrir facilement et même avec une pièce de monnaie. Changer l'accumulateur ou le bloc secteur n'en est que plus rapide. Une plaque de carton noir protège encore l'intérieur après retrait du panneau arrière. On peut aussi l'enlever aisément, elle est fixée par trois boutons-pression. Alors, on a la surprise de la **figure 2**. Tout a l'air fort compressé, mais sans confusion. Les câbles sont rangés bien proprement et ficelés ensemble comme un faisceau d'automobile. La mécanique est faite de métal massif pour affronter le quotidien du métier.

On peut vraiment dire des ingénieurs de UHER qu'ils avaient la technique chevillée au corps et n'ignoraient rien de tout ce qui compose un magnétophone. Tout est bien accessible, pas besoin d'un démontage complet pour échanger des pièces qui s'usent, comme les courroies d'entraînement. La carte d'enregistrement/lecture, à droite dans la figure 2, recouvre une partie de la mécanique. Les ingénieurs munichois en étaient conscients, ils l'ont rendue rabattable (**figure 3**). Il ne faut enlever qu'une seule vis pour la lever. Sur mon appareil, les trois courroies étaient complètement hors d'usage, rien d'étonnant après 53 ans. Heureusement, j'en avais de réserve. D'ailleurs, un grand nettoyage était nécessaire, en particulier de la tête magnétique. Ma bobine de démagnétisation se réjouissait de reprendre du service. La main sur le cœur, cher lecteur de rétronique, avez-vous encore votre bobine de démagnétisation ? À l'intention des plus jeunes, la **figure 4** démystifie cet antique outil qui permet de se débarrasser de la magnétisation indésirable accumulée au fil du temps sur les têtes et les guides de bande et qui porte atteinte à la qualité d'enregistrement et de lecture.

Après ces opérations, un premier test de fonctionnement s'impose. Il faut un moment pour que le moteur démarre, puis il se met à ronronner comme au bon vieux temps. Par chance, j'ai encore une bande magnétique sur une bobine suffisamment petite pour s'insérer. Le diamètre maximum de bobine est de 13 cm, ce qui n'est pas très courant, mais suffisant pour deux heures de service à 9,5 cm/s. Tout est resté impeccable, même la petite ampoule à incandescence qui éclaire le vumètre s'allume encore.

## Données techniques

On trouve encore aujourd'hui les données techniques du 4000 Report-L. La bande passante dépend forcément de la vitesse de défilement. Comme le montre le **tableau 1**, l'appareil ne se contente pas de satisfaire à la norme DIN 45 500 (250 Hz à 6 300 Hz) [2], il surpasse aussi la suivante, DIN 45 511 qui demande une bande passante de 80 Hz à 8 000 Hz.

Il n'y a que la vitesse de 2,4 cm/s qu'il faut oublier quand on veut écouter de la hi-fi. La dérive maximale de vitesse est donnée à  $\pm 0,15\%$  à 19 cm/s et la dynamique, à 56 dB. L'étage de sortie push-pull sans fer donne 1 W dans  $4\ \Omega$ .

## Détails du circuit

Commençons par examiner le diagramme fonctionnel de la **figure 5**. Le magnétophone se compose d'un amplificateur qui combine l'enregistrement et la lecture, d'un ampli de sortie qui attaque le haut-parleur interne ou une connexion pour un haut-parleur externe, d'un générateur d'effacement, d'un régulateur du moteur et d'une alimentation.

Une simple recherche sur l'internet m'a permis de consulter le schéma du 4000 Report-L, il est à la **figure 6** [3], reproduit avec l'aimable autorisation d'ATIS systems GmbH. On y trouve un jeu de transistors éprouvés pour la plupart, mais très courants à l'époque, certains au germanium, mais d'autres déjà au silicium : 3  $\times$  OC 305, 3  $\times$  AC 151, 1  $\times$  AC 153, 1  $\times$  AC 176. Le BCY 51 est au silicium, il est réservé à la partie BF. Le régulateur du moteur est confié aux 3  $\times$  AC 153, 3  $\times$  OC 305 au germanium et 2  $\times$  BFY 39 au silicium. L'ensemble de l'appareil travaille donc avec seulement 17 transistors ! Si on devait en construire aujourd'hui, avec l'inévitable microcontrôleur, on dépasserait les 10 000 transistors.

L'amplificateur combiné enregistrement/lecture dans le tiers supérieur gauche du schéma utilise en première ligne le montage en émetteur commun, universellement adopté pour obtenir une très grande amplification de tension. Étonnant aussi, l'ample réseau de rétroaction qui doit se conformer à la vitesse de défilement imposée par le changement de vitesse. Lors de l'enregistrement, on peut aussi écouter le signal sur haut-parleur. Comme sources de signal, on a le choix entre une entrée pour microphone et une pour radio ou phono.

En haut, à droite dans le schéma, se trouve l'étage final en push-pull. Il est constitué de la paire de transistors complémentaires AC 153 / AC 176, précédés d'un étage intermédiaire de commande à AC 151.

Le générateur d'effacement en bas, à droite, fournit une tension HF de 50 kHz pour la prémagnétisation et l'effacement de la bande. Tout près de lui, on voit aussi la commande du vumètre avec un commutateur pour passer à l'affichage de la tension de batterie.

En bas du schéma, au centre, on voit le régulateur du moteur qui en a inspiré plus d'un. Pour s'en faire une idée, il faut aller consulter le manuel d'entretien [4]. Ici, on doit reconnaître que l'internet a du bon : quelle facilité aujourd'hui d'obtenir les documents de l'appareil et de son entretien, dans le temps, on ne pouvait qu'en rêver. Mais revenons au moteur et à son régulateur. Le moteur est construit avec des enroulements d'induit fixes et un aimant permanent rotatif. C'est ce rotor magnétique qui, par induction, fournit dans les bobines d'induit une tension alternative proportionnelle comme signal

**Tableau 1. Limites de fréquence et vitesses d'entraînement de la bande.**

Vitesse	Fréquences	DIN 45500	DIN 45511
2,4 cm/s	40 à 4 500 Hz	-	-
4,75 cm/s	40 à 10 000 Hz	X	X
9,5 cm/s	40 à 16 000 Hz	X	X
19 cm/s	40 à 20 000 Hz	X	X

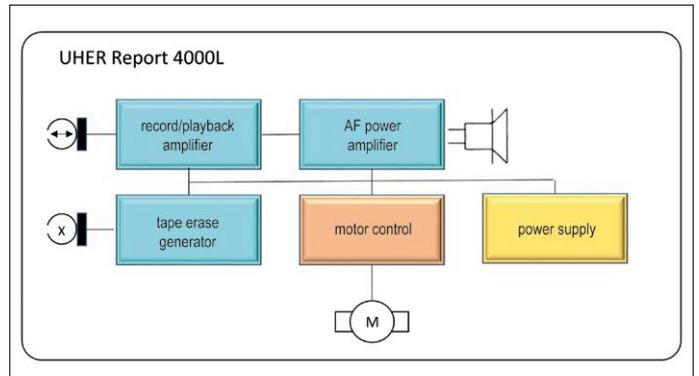


Figure 5. Le diagramme fonctionnel du UHER 4000 Report-L.

de rétroaction pour asservir le régime du moteur. Avec un distributeur spécial de démarrage à commande centrifuge, le moteur démarre en fonction de la position du rotor. Ce distributeur est désactivé par l'action de la force centrifuge avant même que le régime voulu ne soit atteint et la commande du moteur passe en régulation normale. Des images des pièces internes du moteur sont disponibles entre autres sur [5].

Il ne reste plus à voir que l'alimentation en bas, à droite dans le schéma. À part quelques rares composants électroniques, elle se compose principalement d'une simple boîte en plastique noir (**figure 7**). Mais le concept à l'intérieur est simplement génial. Les dimensions de la boîte se basent sur la grandeur de cinq piles LR20 (mono D) qui peuvent servir de source de courant en mobile. On dispose comme accessoires d'un bloc secteur (figure 7) et d'un accumulateur, qui s'adaptent opportunément à la boîte de piles. En outre, on peut charger l'accumulateur sur une batterie auto au moyen d'un câble. Si vous avez regardé attentivement le schéma, vous aurez certainement tiqué en constatant que, fait inhabituel de nos jours, la masse est au pôle positif, mais à l'époque, c'était fort répandu.

## Conclusion

Le magnétophone UHER Report 400L suscite l'enthousiasme aujourd'hui encore par sa robustesse, sa convivialité et sa compacité. Sa construction est orientée dès le départ pour rencontrer les souhaits des utilisateurs et conçue pour une longue durée de service, ce qui est loin d'être resté le cas de nos jours. Souvent, les appareils sont conçus pour qu'il soit impossible de les ouvrir pour remplacer un composant. Si par exemple l'accumulateur intégré est à plat, le seul avenir de l'appareil est la déchetterie. Il est affligeant que des entreprises traditionnelles comme UHER aient périclité parce que leurs produits étaient durables. On a beaucoup écrit à propos des

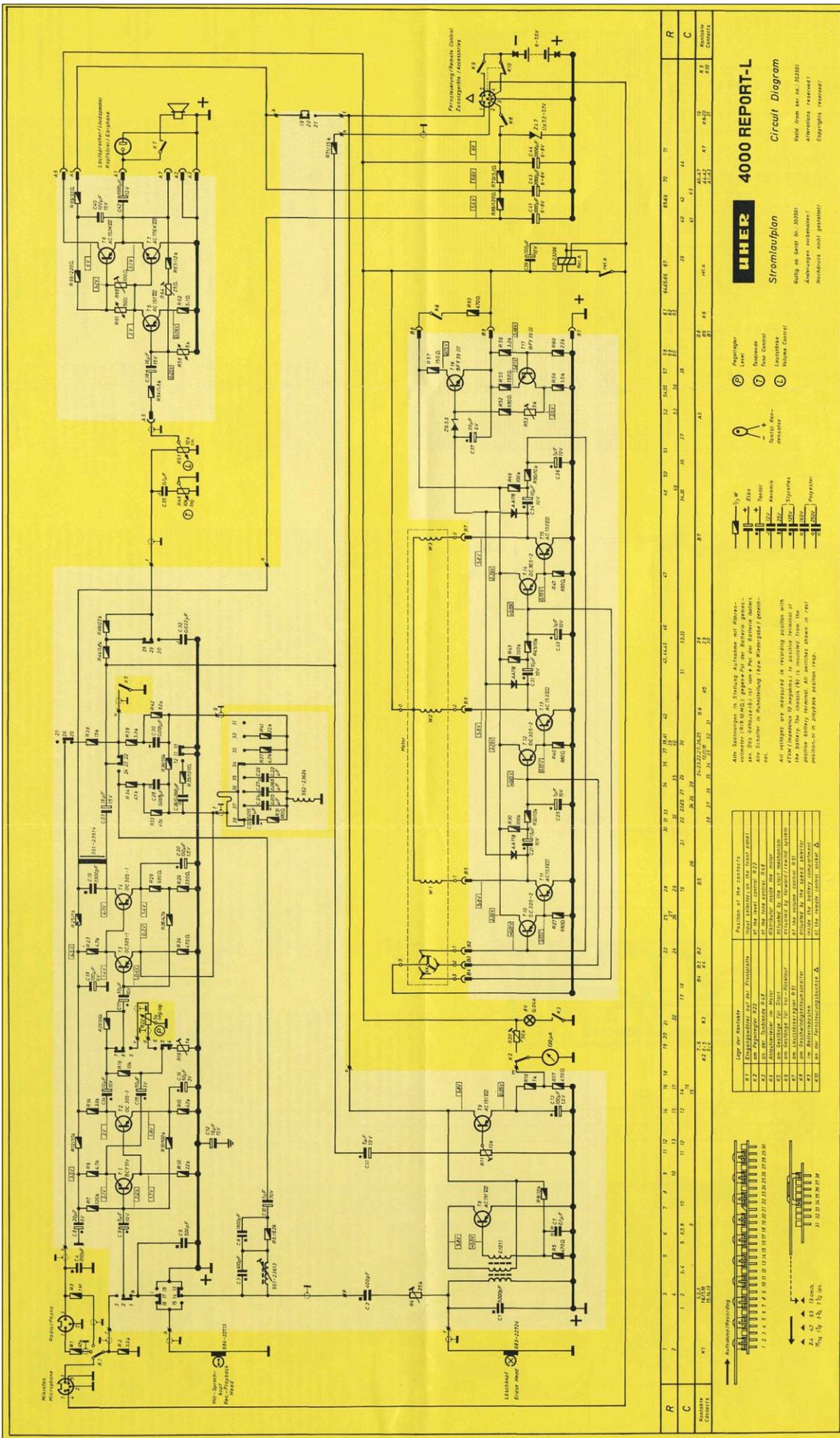


Figure 6. Le schéma détaillé du magnétophone, publié ici avec l'aimable autorisation d'ATIS systems GmbH, détenteur des droits d'auteur.



Figure 7. Le coupleur de piles peut aussi accueillir l'alimentation secteur.

fluctuations qui ont marqué l'histoire de la maison UHER, mais la raconter sortirait du cadre de cet article. Cependant, si elle vous intéresse, dirigez-vous vers les liens [6], [7] et [8] pour vous faire une idée des années brillantes et moins heureuses de l'entreprise. Sur [7], vous trouverez des photos et des souvenirs attachants de quelqu'un qui a travaillé longtemps chez UHER. ◀

(160559 – version française : Robert Grignard)

## Liens

- [1] <https://de.wikipedia.org/wiki/Uher> (en allemand)
- [2] <https://fr.wikipedia.org/wiki/Hi-fi>
- [3] [www.tonband.net/unterlagen/uher/sm\\_4000\\_l.pdf](http://www.tonband.net/unterlagen/uher/sm_4000_l.pdf)
- [4] <http://tonbandwelt.magnetofon.de/uher/4000/serva1.htm>
- [5] <http://forum2.magnetofon.de/board2-tonbandger%C3%A4te/board20-tipps-und-erkenntnisse/16322-der-kollektorlose-b%C3%BChler-motor-des-report-von-innen/>
- [6] [https://de.wikipedia.org/wiki/Uher#cite\\_note-7](https://de.wikipedia.org/wiki/Uher#cite_note-7) (en allemand)
- [7] [www.uher-erinnerungen.de/](http://www.uher-erinnerungen.de/) (en allemand)
- [8] [www.spiegel.de/spiegel/print/d-40350691.html](http://www.spiegel.de/spiegel/print/d-40350691.html) (en allemand)

**EST<sup>®</sup> 2004**

www.elektor.tv



Rétronique est une rubrique mensuelle sur les pages glorieuses et jaunies de l'électronique, avec occasionnellement des montages de légende décrits dans Elektor. Si vous avez des suggestions de sujets à traiter, merci de les télégraphier à [redaction@elektor.fr](mailto:redaction@elektor.fr)

Publicité

## Le coin des petites annonces

Elektor est une plateforme de communication active sur et autour de l'électronique. Grâce à son labo, au magazine, aux livres, à son e-choppe et son rayonnement international, Elektor est aussi un solide support de communication commerciale professionnelle.

Pour animer cette activité et lui donner forme sur papier et en ligne sur le marché français, **nous cherchons un collaborateur indépendant**, rompu aux techniques modernes de communication commerciale.

- Vous parlez français & anglais (ou allemand ou néerlandais).
- Vous avez le goût des contacts humains, de l'imagination et de la pugnacité.
- Vous connaissez le marché francophone de l'électronique.
- Vous pensez que vous pourriez en devenir le trait d'union avec Elektor.

**Contactez-nous**  
([margriet.debeij@eimworld.com](mailto:margriet.debeij@eimworld.com)) !

## Faites découvrir Elektor à vos élèves !

La 29<sup>e</sup> Semaine de la presse et des médias dans l'école se déroulera du 19 au 24 mars 2018 et aura pour thème

**« D'où vient l'info ? ».**

**Cette opération gratuite pour les élèves et leurs enseignants** permet à tout établissement inscrit de recevoir un ou plusieurs exemplaires des magazines qu'il a choisis.

Comme **Elektor participe à l'opération**, tous les lycées de BAC pro et/ou techno, les établissements de formation technique post-BAC (IUT, écoles d'ingénieurs...) ainsi que les établissements de formation des adultes peuvent demander leurs **exemplaires gratuits du magazine Elektor**.

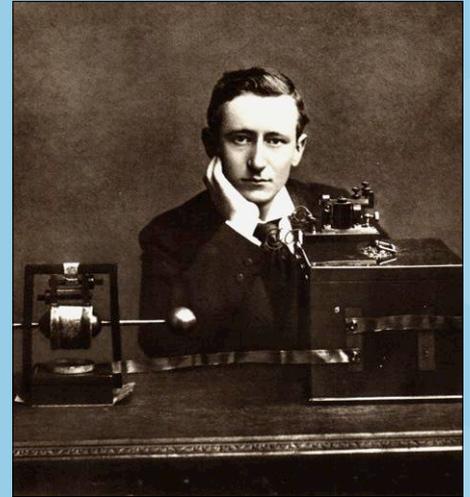
**Inscription obligatoire** sur le site du CLEMI :  
[www.presse.education.gouv.fr/reservation/](http://www.presse.education.gouv.fr/reservation/)  
**du 9 janvier au 7 février 2018**



## QUESTIONS D'ÉTHIQUE

# libérez le spectre radio !

Lorsque Guglielmo Marconi essaie d'obtenir un monopole sur l'utilisation des ondes radio, peu avant 1900, les États y mettent le holà. Un traité international est signé en 1906 ; il spécifie que l'utilisation du spectre relève de la souveraineté nationale. Cette utilisation est depuis strictement réglementée. Bien entendu il y a de bonnes raisons à cela, mais la manière d'appliquer ces règles a freiné le progrès. Avec les nouvelles techniques, il est possible d'allouer une même bande de fréquence à plus d'utilisateurs, ce qui est nécessaire pour faciliter « l'explosion démographique » des appareils connectés. De plus, les liaisons hertziennes sont un outil essentiel pour donner l'accès à l'internet aux quelque trois milliards de personnes qui en sont actuellement privées.



Guglielmo Marconi. Photo : domaine public.

### Tessel Renzenbrink (Pays-Bas)

Marconi a « l'insigne honneur » d'être le seul homme pour qui un traité international a été établi, et ce pour lui barrer la route. Les raisons pour lesquelles les nations ont été unanimes relèvent de leurs préoccupations pour la sécurité ; la radio permettait pour la première fois de communiquer avec des navires en haute mer, et personne ne voulait dépendre – surtout en temps de guerre – du bon vouloir de l'entreprise britannique de Marconi.

### Monopole

Robert Horvitz, de l'*Open Spectrum Foundation*, donne dans un article [1] une seconde raison à la mainmise des États sur le spectre radio : les appareils étaient en ce temps-là un peu primitifs, et ils devaient être manipulés par des opérateurs habiles pour éviter les interférences. En découpant le spectre en bandes, on évitait que des amateurs viennent y mettre la pagaille et en rendent l'utilisation impossible à tous. Celui qui voulait émettre devait demander une licence ; la régulation stricte de l'utilisation des ondes était née.

Horvitz commente à ce sujet : « il est important de noter que la revendica-



Un opérateur radio reçoit un message transmis par voie hertzienne avec le détecteur magnétique de Marconi.

Photo : Archie Frederick Collins (domaine public).



Absence de signal.

Photo : Liz Sullivan (Wikimedia Commons).

tion des nations pour contrôler souverainement le spectre n'est pas basée sur les propriétés physiques des ondes radio. Ces ondes sont après tout identiques aux rayons lumineux, et aucun État n'a jamais voulu s'approprier la lumière du soleil franchissant ses frontières. Cela montre que les efforts des nations pour monopoliser l'accès au spectre radio ne sont pas de nature scientifique, mais proviennent bien d'un agenda politique ».

### Enchères du spectre

Il résulte de cette régulation stricte que le spectre n'est pas utilisé de manière efficace. Par exemple, les autorisations pour certaines bandes de fréquence – dont celles pour la téléphonie mobile – sont attribuées aux enchères. Les entreprises de télécommunications payent des sommes faramineuses pour ces autorisations ; elles reçoivent en retour un accès exclusif à ces bandes de fréquence. Cette exclusivité fait que certaines parties du spectre sont peu ou pas du tout utilisées. Les autorités ne sont bien entendu pas objectives en la matière, car les enchères rapportent des milliards aux finances publiques :

environ 5 milliards à l'Allemagne en 2015, et 3,8 milliards aux Pays-Bas en 2012, pour ne citer que ces deux pays. Les autorisations ne concernent pas seulement une bande de fréquence donnée. Elles couvrent aussi en général un vaste territoire, ce qui n'intéresse pas nécessairement les entreprises : dans les zones urbaines, les installations sont vite rentabilisées, mais il n'en va pas de même en zone rurale. Il n'y a pas de modèle économique viable qui justifierait la mise en place d'un énorme réseau dans ces zones, et certaines régions à faible densité de population sont peu couvertes, voire pas du tout.

### TV White Spaces

La politique des autorités peut donc conduire à une indisponibilité artificielle du spectre, alors que les techniques modernes en permettent une utilisation toujours plus efficace. Les TV White Spaces (TVWS) en sont un bel exemple. Dans la partie du spectre réservée aux émissions TV, il s'agit de bandes de fréquence non utilisées afin de bien séparer les canaux ; comme les émetteurs TV sont particulièrement puissants, ces bandes « tampon » sont relativement larges. Des appareils peu puissants pourraient utiliser ces fréquences « libres » sans gêner les utilisateurs primaires ; une surveillance du spectre et les données géographiques relatives aux émetteurs TV permettraient de déterminer les canaux libres pour les émetteurs « White Space ».

### Tous en ligne

Cette gestion intelligente du spectre serait bien nécessaire pour donner enfin un accès à l'internet à la moitié de la population mondiale qui ne l'a pas encore. D'autant plus que des services vitaux comme l'e-santé et l'Administration électronique deviennent disponibles un peu partout. Une bonne partie des gens qui n'ont pas accès à l'internet vit dans des zones peu peuplées, où il ne serait pas rentable d'installer des câbles ; les liaisons sans fil sont essentielles pour connecter ces personnes. Comme dit plus haut, les entreprises de télécommunications perdraient de l'argent si elles voulaient couvrir ces zones. Il

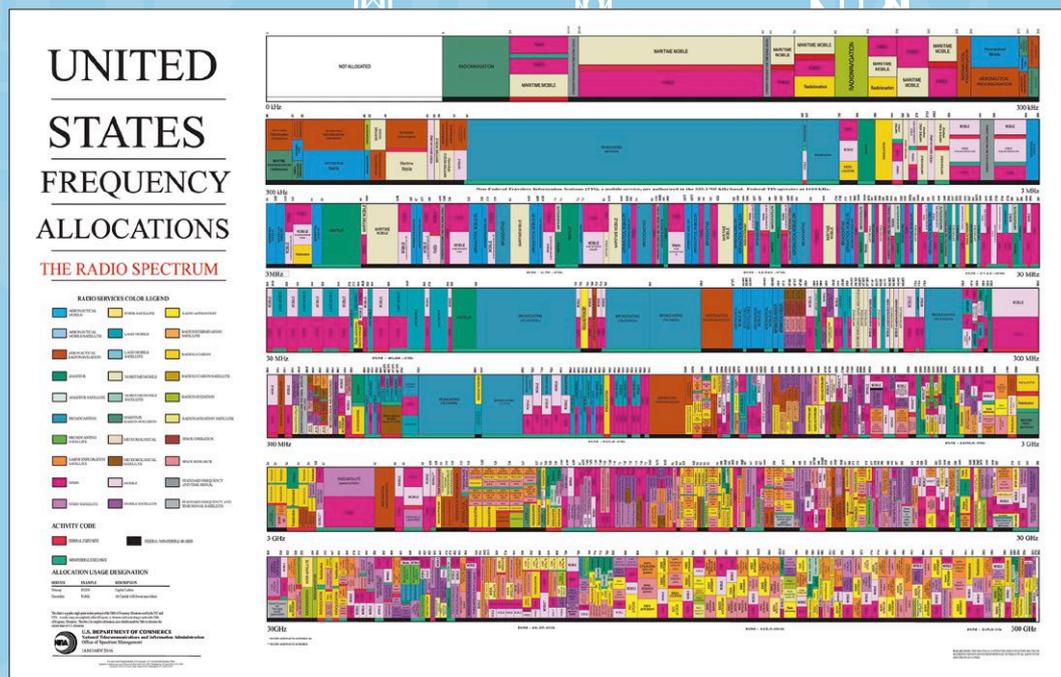
s'agit du problème du dernier kilomètre, expression qui ne concerne pas que les télécommunications : les moyens et coûts nécessaires pour atteindre les utilisateurs finaux individuels. L'Internet Society (ISOC), créée par les pères fondateurs de l'internet que sont Vint Cerf et Bob Kahn – ils ont inventé le protocole TCP/IP, propose une approche alternative. Le but de l'association est de rendre l'internet accessible partout et pour tous, et dans un rapport récent [2] elle défend pour ce faire le principe de réseaux communautaires. Il s'agit de réseaux qui seraient mis en place et entretenus par la communauté, plutôt que d'attendre la bonne volonté des autorités ou des entreprises. Le rapport souligne qu'il s'agit d'un retournement de paradigme, le dernier kilomètre devenant le premier kilomètre : les utilisateurs finaux ne sont plus éloignés des réseaux de télécommunications, ils sont le centre d'un réseau local. Pour permettre l'accès à l'internet, les réseaux locaux devront être reliés à une dorsale internet, mais c'est beaucoup moins cher pour une entreprise de connecter un réseau que de nombreux utilisateurs individuels.

Les réseaux communautaires doivent avoir un accès au spectre radio pour l'établissement des liaisons sans fil, et l'ISOC plaide pour une utilisation optimale du spectre. Les limitations ne sont pas d'ordre technique, mais bien – comme c'est souvent le cas – dues à une politique de gestion établie à et pour une autre époque. ◀

(160561 – version française : Jean-Louis Mehren)

### Liens

- [1] Robert Horvitz : *Geo-database management of white space vs. open spectrum*. Dans *TV Whitespaces, a pragmatic approach*, édité par E. Pietrosemoli et M. Zennaro. Disponible à l'adresse <http://wireless.ictp.it/tvws/book>.
- [2] Internet Society : *Policy Brief – Spectrum approaches for community networks*. Disponible à l'adresse [www.internetsociety.org/policybriefs/spectrum](http://www.internetsociety.org/policybriefs/spectrum).



Allocation des fréquences aux États-Unis. Graphique : NASA (domaine public).



# bienvenue dans votre e-choppe

la rédaction recommande



## Kit Touch Board Pro

Le kit Touch Board Pro contient tout ce qu'il faut à un esprit créatif pour fabriquer une signalétique parlante ou des posters interactifs et sonores, des instruments de musique inédits. Ce kit marie électronique programmée (Atmel ATmega32U4) et capteurs tactiles pour une interactivité immédiate. Il comprend la carte Touch Board, programmée comme une carte Arduino, et une peinture conductrice (dite *electric paint*) qui s'applique comme une encre, au pinceau, au pochoir ou en sérigraphie. Douze électrodes de la carte Touch Board sont déjà

programmées pour déclencher des sons au moyen de capteurs capacitifs ou de proximité. C'est possible sans programmer une ligne de code.

Le kit contient d'autres accessoires pour vous faciliter la réalisation de prototypes originaux et épatants, tel un rouleau de ruban de cuivre adhésif.

Denis Meyer (labo d'Elektor)

[www.elektor.fr/touch-board-pro-kit](http://www.elektor.fr/touch-board-pro-kit)



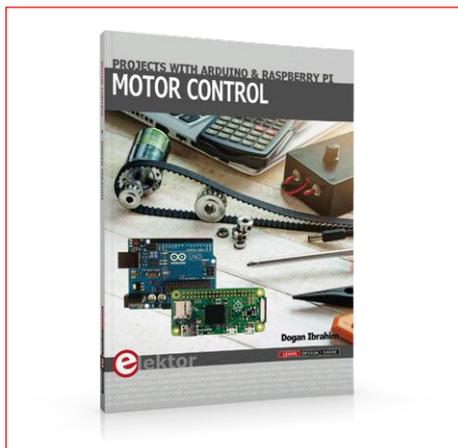
## Vos favoris :

1. Python 3 - Programming and GUIs (livre en anglais)  
[www.elektor.fr/python](http://www.elektor.fr/python)



2. Livre officiel de l'ESP32 (livre en anglais)  
[www.elektor.fr/esp32-book](http://www.elektor.fr/esp32-book)
3. Horloge de sable Arduino  
[www.elektor.fr/horloge-de-sable-arduino](http://www.elektor.fr/horloge-de-sable-arduino)
4. Microscope USB numérique A1  
[www.elektor.fr/usb-microscope-a1](http://www.elektor.fr/usb-microscope-a1)
5. Lampe Peltier  
[www.elektor.fr/lampe-peltier](http://www.elektor.fr/lampe-peltier)
6. STEMtera  
[www.elektor.fr/stemtera](http://www.elektor.fr/stemtera)

## Motor Control Projects with Arduino and RPi (livre en anglais)



Ce livre traite des moteurs électriques à courant continu et de leur utilisation dans des projets à base d'Arduino et de Raspberry Pi Zero W. Il comprend de nombreux projets testés sur les moteurs CC standard, les moteurs pas-à-pas, les servomoteurs et les robots mobiles. Le livre s'adresse aux étudiants, aux amateurs et à tous ceux qui souhaitent concevoir des montages à microcontrôleur, également avec commande à distance (Wi-Fi & Bluetooth).

Prix (membres) : 29,66 €

[www.elektor.fr/motor-control](http://www.elektor.fr/motor-control)

## Écran tactile de 25 cm pour RPi (HDMI/VGA/BNC/AV)

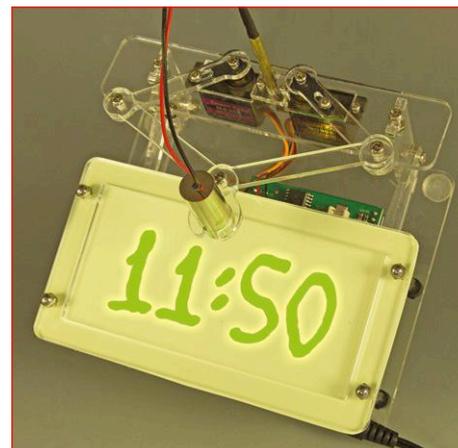


Écran tactile de 25 cm (10 pouces), logé dans un élégant et robuste boîtier métallique avec finition noire mate et dalle à vitre durcie (3H). Au dos de l'écran, une plaque peut recevoir n'importe lequel des ordinateurs monocartes courants (Raspberry Pi, etc). Alimentation par USB ou adaptateur secteur (fourni). Cet écran sert de moniteur de PC standard, grâce à ses connecteurs HDMI et VGA. Le compagnon idéal du microscope HDMI Andonstar ADMS201.

Prix (membres) : 130,46 €

[www.elektor.fr/ecran-tactile-rpi](http://www.elektor.fr/ecran-tactile-rpi)

## Laser Time Writer : kit de transformation



Kit de modification de l'horloge de sable d'Elektor (cf. janvier 2017) : le sable est remplacé par un autocollant phosphorescent sur lequel un laser vient écrire l'heure. Le kit contient toutes les pièces nécessaires pour transformer votre horloge de sable en graveur laser : pièces acryliques rassemblées sur une plaque prédécoupée, autocollant phosphorescent, module laser, bouton-poussoir et fil, entretoise en laiton et petite quincaillerie.

Prix (membres) : 40,46 €

[www.elektor.fr/laserclock](http://www.elektor.fr/laserclock)

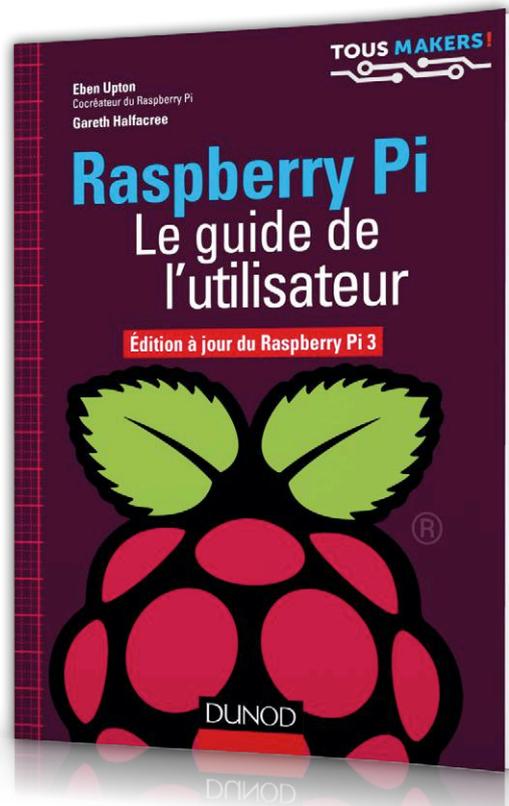
NOUVEAU  
LIVRE

# Raspberry Pi

## Le guide de l'utilisateur

Ouvrage coécrit par Eben Upton, le créateur du RPi –  
édition à jour pour RPi 3, 288 pages

Le nano-ordinateur Raspberry Pi 3 apporte son lot de nouveautés : intégration du Wi-Fi et du Bluetooth, processeur plus puissant de 33% (toujours quatre coeurs, mais en 64 bits), meilleure gestion de la puissance et nouveau modèle de connecteur micro-SD. Écrit par le créateur du Raspberry Pi, cet ouvrage est un vrai manuel d'utilisation destiné à tous ceux qui souhaitent tirer le meilleur parti de cette carte. Il permet de comprendre le matériel et ses principes de fonctionnement (installation, configuration) et d'apprendre les bases de deux langages de programmation simples, Scratch et Python.



Prix (membres) : 24,21 €

[www.elektor.fr/raspberry-pi-le-guide-de-l-utilisateur](http://www.elektor.fr/raspberry-pi-le-guide-de-l-utilisateur)

### Kit de démarrage du Raspberry Pi Zero W



Un kit avec tout ce qu'il faut pour démarrer avec le nouveau Raspberry Pi Zero W ! Il comprend une bande de huit LED multicolores *Blink!*, un boîtier acrylique, une carte micro-SD de 8 Go avec NOOBS 2.2 préchargé, un connecteur mâle à 2x20 broches, 2 câbles USB, un adaptateur HDMI et bien sûr un Raspberry Pi Zero W avec puce monocœur, Wi-Fi et Bluetooth intégrés. Parfait pour effectuer vos premiers pas en Python.



Prix (membres) : 47,66 €

[www.elektor.fr/zero-kit](http://www.elektor.fr/zero-kit)

### Maîtrisez les microcontrôleurs à l'aide d'Arduino



3<sup>e</sup> édition augmentée avec AVR Playground & Elektor Uno R4

L'ambition de cet ouvrage est de vous faire entrer dans le monde Arduino puis de vous en faire ressortir victorieux pour vous emmener plus loin dans l'apprentissage de la programmation des microcontrôleurs. Il met la théorie en pratique sur une carte Arduino avec l'environnement de programmation Arduino.



Prix (membres) : 32,85 €

[www.elektor.fr/maitrisez-les-microcontrolleurs-3](http://www.elektor.fr/maitrisez-les-microcontrolleurs-3)

### TAPIR – détecteur ultrasensible d'électrosmog



Ce détecteur de pollution électromagnétique à large bande et ultra-sensible permet de détecter toute source de champ électrique E ou magnétique H, normalement inaudible. Le TAPIR, acronyme de *Totally Archaic but Practical Interceptor of Radiation* est ultra simple à utiliser. Branchez les écouteurs et une antenne, allumez-le. Approchez-vous d'appareils électriques et vous entendrez différents bruits (suivant le type et la fréquence du champ émis).



Prix (membres) : 22,46 €

[www.elektor.fr/tapir](http://www.elektor.fr/tapir)

# Hexadoku casse-tête pour elektorniciens

Votre magazine se termine toujours et encore par une grille de sudoku particulière. Éteignez le fer à souder, oubliez les écrans de PC et d'oscilloscope, 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 €**.

À vos crayons !

Envoyez votre réponse (les chiffres sur fond grisé) avec vos coordonnées par courriel, avant le **23 janvier 2018** à l'adresse [hexadoku@elektor.fr](mailto:hexadoku@elektor.fr)

### Les gagnants

La solution de la grille du numéro de novembre/décembre est **749AF**

Les cinq bons Elektor d'une valeur de **50 €** vont à :

Ricardo **Claudino** (Portugal), Jean-François **Dedecker** (Belgique), John **Jones** (États-Unis), Monique **Notot** (France), Siem **Oppe** (Pays-Bas).

Bravo à tous les participants et félicitations aux gagnants !

			B	2				D	1							
8	3			F	5			0	6					7	D	
1	2		C	8	D	E	7	B	F		0			6	A	
	F	0	4	6					2	3	5			B		
9	5		D								3			A	E	
7			E		A	C	D	9	1	3		4			2	
0	A	C					4	6						1	9	5
				9			F	A			2					
				6			1	8			E					
A	4	3					C	D						B	8	0
C			F		0	8	A	2	4	B		6				1
5	E		6									C		4	9	
	6		5	F	1				4	8	D			0		
4	D		3		C	E	8	B	2	0		F		5	6	
E	0			4	3				6	1					2	8
			8	B					9	E						

0	1	4	A	9	B	6	C	5	2	8	D	F	3	7	E	
5	6	B	3	F	7	1	8	E	0	4	C	A	D	2	9	
7	9	C	E	0	D	A	2	1	F	6	3	8	4	B	5	
8	D	F	2	4	3	E	5	7	9	A	B	C	0	1	6	
F	B	E	0	C	1	D	6	8	A	5	2	7	9	3	4	
9	8	A	4	5	F	0	3	B	C	7	E	6	1	D	2	
1	C	2	D	7	4	9	A	F	6	3	0	B	E	5	8	
3	5	6	7	2	8	B	E	4	D	1	9	0	A	C	F	
B	F	7	5	D	0	2	4	9	1	C	A	E	8	6	3	
D	A	8	6	3	C	F	9	0	E	2	5	1	7	4	B	
C	E	0	9	8	A	7	1	6	3	B	4	2	5	F	D	
2	4	3	1	6	E	5	B	D	7	F	8	9	C	0	A	
A	0	9	B	1	5	3	7	2	4	E	6	D	F	8	C	
E	2	D	C	A	6	8	F	3	5	0	1	4	B	9	7	
4	3	1	F	B	2	C	D	A	8	9	7	5	6	E	0	
6	7	5	8	E	9	4	0	C	B	D	F	3	2	A	1	

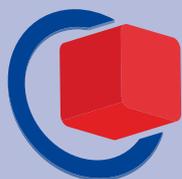
Tout recours est exclu, de même que le sont, de ce jeu, les personnels d'Elektor International Media et leur famille. Un seul gagnant par foyer.

**2ew18P**

Code donnant droit à l'entrée gratuite  
▶ [embedded-world.de/voucher](http://embedded-world.de/voucher)

Nuremberg, Allemagne

**27.2 – 1.3.2018**



**embeddedworld**

Exhibition & Conference

... it's a smarter world

**DÉCOUVRIR LES INNOVATIONS**

Plongez dans l'univers des systèmes embarqués et découvrez des innovations couronnées de succès.

▶ [embedded-world.de](http://embedded-world.de)

Partenaires médias

**Markt&Technik**  
Die unabhängige Wochenzeitung für Elektronik

**DESIGN & ELEKTRONIK**  
KNOW-HOW FÜR ENTWICKLER

elektroniknet.de  
**Elektronik**  
Fachmedium für industrielle Anwender und Entwickler

Elektronik  
**automotive**  
Fachmedium für professionelle Automobilelektronik

**Computer & AUTOMATION**  
Fachmedium der Automatisierungstechnik

**SmarterWorld**  
Solutions for a Smarter World

**MEDIZIN+elektronik**  
Fachmedium für Elektronik in der Medizintechnik

[computer-automation.de](http://computer-automation.de)

[elektroniknet.de](http://elektroniknet.de)

**MEDIZIN-UND-elektronik.DE**

Organisateur du salon professionnel

NürnbergMesse GmbH

T +49 9 11 86 06 - 49 12

F +49 9 11 86 06 - 49 13

[visitorservice@nuernbergmesse.de](mailto:visitorservice@nuernbergmesse.de)

Organisateur des congrès

WEKA FACHMEDIEN GmbH

T +49 89 2 55 56 - 13 49

F +49 89 2 55 56 - 03 49

[info@embedded-world.eu](mailto:info@embedded-world.eu)

NÜRNBERG MESSE



# LA CONSOLE DE JEUX CHIC ET RÉTRO POUR VOTRE SALON RASPBERRY PI 3 DESIGN NES

Avec ce pack, vous pouvez placer une console de jeux haut de gamme, chic et rétro, dans votre salon. Pour l'installation, nous recommandons RetroPie, ainsi que notre rubrique « How-To » dans le magazine reichelt.



Référence :  
RPI NESPI BUNDLE

**NOUVEAU** **75,55**

## Notre pack comprend :

- une console Raspberry Pi 3 B
- un boîtier de grande qualité avec le design NES
- une manette de commande avec le design NES
- une manette de commande avec le design SNES
- un ventilateur pour le boîtier
- un bloc d'alimentation puissant
- une carte microSD de 16 Go (classe 10) avec NOOBS pré-installé (6 systèmes d'exploitation)
- un câble HDMI de 1,2 m



## RASPBERRY PI 3

Des millions d'exemplaires vendus dans le monde entier ...  
...et maintenant chez nous, plus économique que jamais !

- Processeur Quad-Core 1,2 GHz ARM Cortex-A53 64-Bit BCM2837
- 1 Go SD-RAM & VideoCore IV avec Dual-Core-GPU
- WiFi, Bluetooth, HDMI, USB, LAN, fente microSD, CSI, DSI, 40x E/S, ...



Référence :  
au lieu de 33,53 RASPBERRY PI 3

**ÉCONOMISEZ**  
**13%** **29,33**



## Alimentation électrique pour cartes enfichables

- Intensité maximale : 700 mA
- Entrée 1 : Connecteur creux 5,5/2,1 mm, DC IN 6...12 V
- Entrée 2 : USB 5 V
- Entrée 3 : Micro USB 5 V
- Sortie : 3,3 V, 5 V, masse, USB

Référence :  
DEBO BREAD POWER

**4,12**



## How-To dans le magazine reichelt

### Le plaisir du jeu rétro avec la Raspberry Pi :

voici comment transformer votre ordinateur miniature en une console de jeux rétro en quelques étapes seulement

Nous vous expliquons pas à pas dans le magazine reichelt quels outils vous permettent d'installer une véritable console rétro dans votre salon, mais aussi comment obtenir de véritables classiques à faire fonctionner dessus.

### S'INFORMER MAINTENANT ▶



<http://www.rch.lt/NES-FR>

