



elektor

DÉCOUVRIR

CRÉER

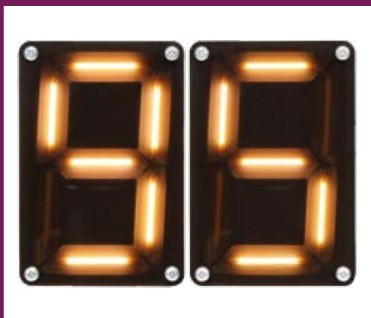
PARTAGER



ESP8266 sur la
carte d'entrées/sorties
Android



i-Pendulum
tient en équilibre
et se relève tout seul



LEDitron
LED en filament pour
afficheur à 7 segments

programmeurs et débogueurs

outils de
développement
pour tous

Dans cette édition :
5 projets de labo
2 projets de lecteur
1 cours Windows & RPi
des trouvailles sur
l'internet ...



mini-kit ampli-op pour myDAQ : expérimenter avec
dix configurations différentes • MILIPOL PARIS 2015 •

journal défilant avec Arduino & matrice de 16 x 16 LED • relais
à mercure • **analyseur de signal par Ethernet (2) :** du code, des maths, du
concret • trucs & astuces • **WLAN compact et autonome,** ou comment utiliser
la puce ESP8266 sans μ C • tuyaux électroniques pour réparer et concevoir •
Windows sur la carte RPi (3) : SPI et I²C • bruits de labo • **LED en filament**



Développer plus rapidement et plus facilement vos applications embarquées grâce à la gamme d'analyseurs logiques et générateurs de signaux ScanaQuad !

Avec 4 voies configurables en entrées ou en sorties, et une fréquence d'échantillonnage allant jusqu'à 200MHz, vous pourrez travailler sereinement sur tous vos bus de communication séries (UART, I2C, SPI, 1-Wire, RS232/485, CAN, LIN,...). Les voies sont protégées jusqu'à $\pm 35V$ et bénéficient d'un seuil de tension d'entrée ajustable. Ce qui rends ScanaQuad unique parmi la concurrence, c'est son aptitude à générer et enregistrer des signaux simultanément : vous apprécierez alors la possibilité de générer des signaux I2C en quelques cliques, et enregistrer la réponse d'un circuit.

ScanaQuad SQ25



- 25 MSPS @ 4 voies
- 256K points / voie
- Générateur de signaux*
- Protection $\pm 12V$
- Trigger: Rising, falling, level, pulse width

69€ H.T.

ScanaQuad SQ50



- 50 MSPS @ 4 voies
- 1M points / voie
- Générateur de signaux*
- Protection $\pm 12V$
- Trigger: Rising, falling, level, pulse width, serial protocol
- Tension sortie variable
- Seuil: 0.5V \rightarrow 3V

89€ H.T.

ScanaQuad SQ100



- 100 MSPS @ 4 voies
- 2M points / voie
- Générateur de signaux*
- Protection $\pm 35V$
- Trigger: Rising, falling, level, pulse width, serial protocol
- Tension sortie variable
- Seuil: 0.5V \rightarrow 3V
- Impédance variable
- Sorties « Open-drain »
- Option Pull-up.
- Paires différentielles : 1

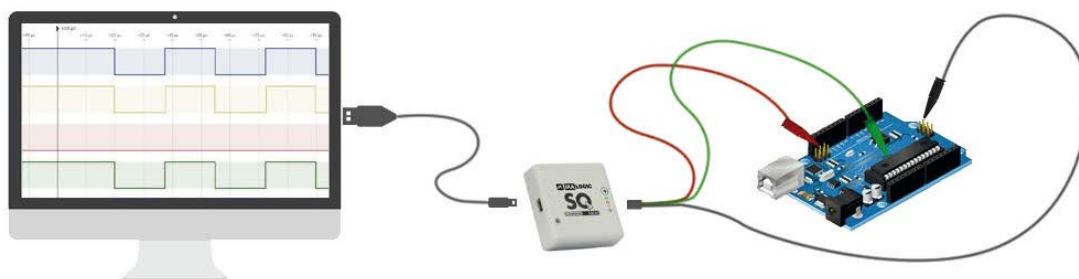
119€ H.T.

ScanaQuad SQ200



- 200 MSPS @ 4 voies
- 4M points / voie
- Générateur de signaux*
- Protection $\pm 35V$
- Trigger: Rising, falling, level, pulse width, serial protocol
- Tension sortie variable
- Threshold: 0.5V \rightarrow 3V
- Impédance variable
- Sorties « Open-drain »
- Option Pull-up.
- Paires différentielles : 2

149€ H.T.



Choisissez votre ScanaQuad sur www.ikalogic.com !

* Signaux numérique.

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

Tél. : (+33) 01.49.19.26.19
du lundi au vendredi de 10h à 13h

Fax : (+33) 01.49.19.22.37

www.elektor.fr | www.elektormagazine.fr

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

Publicité :

Fabio Romagnoli +32 485 65 40 90
fabio.romagnoli@eimworld.com

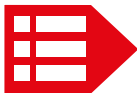
DROITS D'AUTEUR :

© 2016 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 editrice 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 editrice. La Société editrice 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 editrice 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 editrice 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.



Le râteau de la TNT

Le déploiement de la télévision numérique terrestre simple définition (codée au format MPEG-2) a commencé en mars 2005 à Paris, Lyon et Marseille. Presque toute la France était couverte en novembre 2011. Les quatre premières chaînes gratuites de TNT haute définition (codage MPEG-4) sont apparues en octobre 2008. Pour mémoire, les chaînes à péage diffusaient dès 2006 également en MPEG-4.

Le déploiement de la TNT a duré sept ans parce que les fréquences étaient rares. Ce n'est que lorsque la diffusion analogique a complètement cessé que le nombre de fréquences disponibles a augmenté et que la diffusion en HD sur tout le territoire a été possible. Toutefois les opérateurs de téléphonie mobile lorgnaient aussi ces fréquences libérables, appelées « dividende numérique ».

Le résultat de cette bataille entre télévision et téléphone pour ce trésor est que, durant la nuit du 4 au 5 avril 2016, les chaînes de la TNT vont être rassemblées sur six multiplex (au lieu de huit auparavant) afin de libérer une partie de la bande de fréquences pour la téléphonie mobile.

Lors de cette opération, le nombre de chaînes diffusées en HD passera de 11 à 29 chaînes nationales (sur un total de 31). C'est pourquoi l'État nous invite depuis plusieurs semaines à vérifier la compatibilité HD de notre matériel de réception de la TNT.

En effet, le matériel SD des premiers téléspectateurs de la TNT était déjà obsolète dès l'achat puisqu'il ne permettait pas d'accéder à la TNT payante ni de recevoir les chaînes HD. Que vont devenir les centaines de milliers de décodeurs TNT SD et décodeurs satellites SD périmés dès leur fabrication ? Vont-ils grossir les rangs des DEEE* ou rejoindre les ordinaux du grenier (cf. édito de mars) ? Quel gâchis !

Mariline Thiebaut-Brodier

* Déchets d'Équipements Électriques et Électroniques.

Notre équipe

Rédactrice en chef :	Mariline Thiebaut-Brodier (redaction@elektor.fr)
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 :	Robert Grignard, Jean-Louis Mehren, Denis Meyer, Hervé Moreau, Kévin Petit, Guy Raedersdorf
Service de la clientèle :	Cindy Tijssen
Graphistes :	Giel Dols, Mart Schroijen, Patrick Wielders
Elektor en ligne :	Daniëlle Mertens

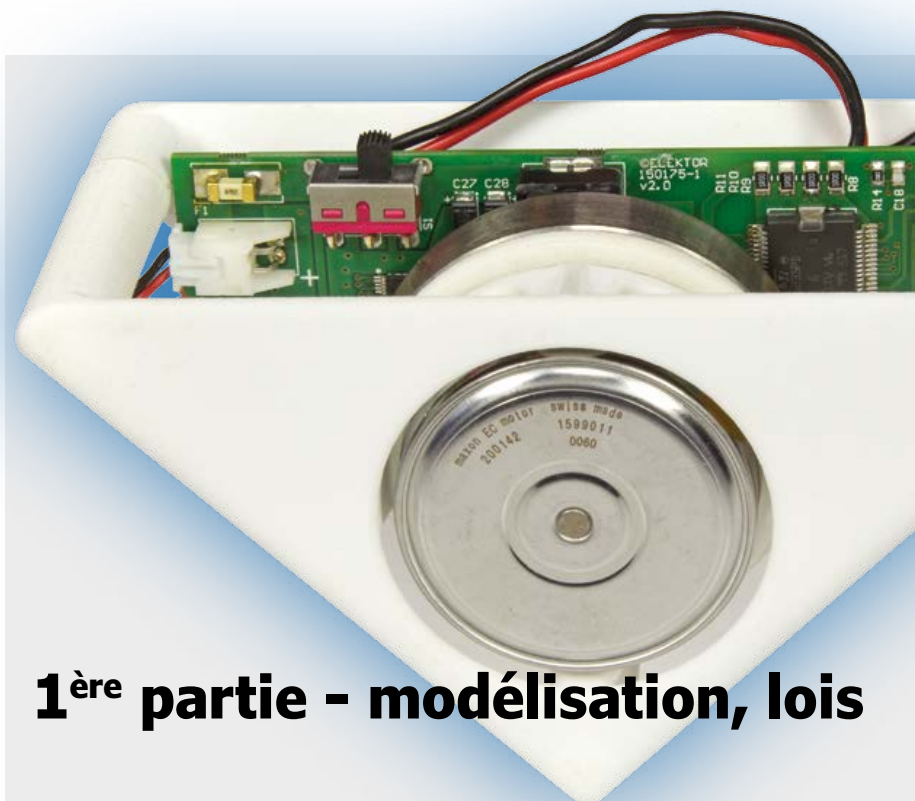
- 5 Bientôt dans Elektor
- 6 Elektor : guide de connexion(s)
- 24 ElektorBusiness
morceaux choisis
- 26 MILIPOL PARIS 2015
à point nommé ?
- 28 agenda
avril 2016
- 70 l'e-choppe d'Elektor
- 80 des nouvelles du monde d'Elektor
- 82 hexadoku
casse-tête pour elektorniciens

DÉCOUVRIR CRÉER PARTAGER

- 8 bienvenue dans la section DÉCOUVRIR
- 9 relais à mercure
drôle de composant n°23
- 10 programmeurs et débogueurs
outils de développement pour tous
- 18 Windows sur la carte RPi (3)
SPI et I²C
- 23 trucs & astuces
moteurs à courant continu - piles boutons

DÉCOUVRIR CRÉER PARTAGER

- 29 bienvenue dans la section CRÉER
- 30 LEDitron
LED en filament pour afficheur à sept segments
- 35 i-Pendulum
1^{ère} partie - modélisation, lois de contrôle,
filtre de Kalman
- 42 ESP8266 sur
la carte d'entrées/sorties Android
lancez-vous dans la mise à jour du micrologiciel
- 47 mini-kit ampli-op pour myDAQ
expérimenter avec dix configurations
différentes



1^{ère} partie - modélisation, lois

ESP8266 sur la carte d'entrées/sorties Android

lancez-vous dans la mise à jour du micrologiciel

La carte d'E/S Android offre la possibilité de commander facilement des appareils électroniques à partir d'un ordiphone ou d'une tablette. Il y a de la place sur la carte pour au moins sept modules différents de liaison sans fil, dont l'ESP8266, populaire et bon marché. Il faut pour cela reprogrammer le module avec un nouveau micrologiciel. Nous vous expliquons ici comment faire.

42



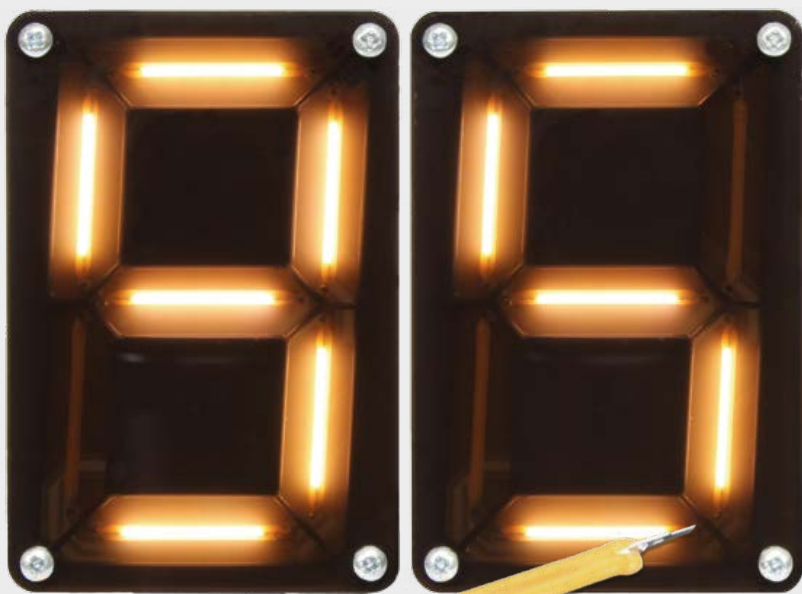


Lorsque j'ai présenté mon prototype de pendule inversé, un collègue m'a demandé : « et donc tu t'es réveillé un matin en te disant : aujourd'hui je vais fabriquer un pendule inversé ! ». Ce projet devrait susciter le même étonnement dans votre entourage parce que ce pendule est capable non seulement de tenir en équilibre, mais aussi de se relever tout seul.

35

i-Pendulum

de contrôle, filtre de Kalman



29

LEDitron

LED en filament pour afficheur à sept segments

Les lampes fluorescentes compactes sont passées de mode, les LED sont aujourd'hui les reines de l'éclairage. Le LEDitron utilise des LED en filament pour créer un afficheur à sept segments. Un peu comme un tube Numitron, mais à faible consommation.

- 52 journal défilant**
avec Arduino & matrice de 16 x 16 LED
- 58 analyseur de signal par Ethernet (2)**
du code, des maths, du concret
- 64 WLAN compact et autonome**
ou comment utiliser la puce ESP8266 sans µC

DÉCOUVRIR CRÉER PARTAGER

- 74 bienvenue dans la section PARTAGER**
- 75 tuyaux électroniques**
pour réparer et concevoir
- 76 LED en filament fragiles**
de nombreuses LED sur un substrat de verre
- 78 bruits de labo...**
génie et diversité

 **bientôt sur ces pages**

Alimentation modulaire

Pour le générateur de fonctions pas DDS du numéro de novembre 2015, nous avons utilisé une alimentation dans laquelle le transformateur et le filtre prenaient beaucoup de place (*alimentation universelle* de janvier-février 2016). Nous vous montrerons comment rendre ce circuit plus compact et plus efficace grâce à un module de conversion CC/CC de Recom. Il est possible d'adapter cette alimentation à nombre d'autres projets.

Robot de téléprésence

Si vous êtes un fan de la série télé « The Big Bang Theory », vous connaissez le robot de téléprésence de Sheldon. Croyez-nous, notre robot est encore plus sympathique. Il a été entièrement pensé par l'inventeur de Wheelie, Chris Krohne. Vous le découvrirez dans le prochain numéro ainsi que comment sont reliés de manière astucieuse la tablette, l'ordiphone de commande et l'électronique logée dans le pied du robot.

Bien entendu d'autres projets sont en préparation comme la mesure de distance avec RPi ou une horloge Nixie à six chiffres.

Sous réserve de modification.

Le numéro de mai 2016 paraîtra le 20 avril 2016.

la communauté d'Elektor

DÉCOUVRIR

CRÉER

PARTAGER

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 cave de d'Ali Baba des électroniciens ! 10 % de remise permanente pour les membres GREEN et GOLD. www.elektor.fr



Elektor Magazine : dix numéros par an d'électronique inédite (projets, banc d'essais, cours, trucs et astuces, actualité...). www.elektormagazine.fr



Elektor PCB Service : gravure à la demande de vos circuits imprimés, en un ou plusieurs exemplaires. www.elektorpcbservice.com



Elektor GRATUIT & SANS PAPIERS : lettre d'information hebdomadaire à l'affût des nouveautés. Gratuite : www.elektor.fr/inscription



Elektor Academy : l'apprentissage par la pratique au travers de webinaires, séminaires, ateliers, DVD... www.elektor-academy.com



Elektor Boeken & DVD's: Arduino, Raspberry Pi, microcontrôleurs, Linux et bien d'autres. 10 % de remise pour les membres ! www.elektor.fr



Elektor TV : présentations, comptes-rendus, déballage et évaluation de produits. Pour y voir plus clair. 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.elektor-labs.com

connectez-vous à notre communauté

formule GREEN

92,50 € par an

- ✓ l'accès à l'archive d'Elektor
- ✓ 10% de remise dans l'e-shoppe
- ✓ 10 x magazine numérique
- ✗ 10 x magazine imprimé
- ✓ des offres exclusives
- ✓ l'accès à **Elektorlabs**
- ✓ un DVD annuel

connectez-vous

www.elektor.fr/formule-greencard

formule GOLD

127,50 € par an

- ✓ l'accès à l'archive d'Elektor
- ✓ 10% de remise dans l'e-shoppe
- ✓ 10 x magazine numérique
- ✓ 10 x magazine imprimé
- ✓ des offres exclusives
- ✓ l'accès à **Elektorlabs**
- ✓ un DVD annuel

connectez-vous

www.elektor.fr/formule-goldcard

formule GRATUITE

- ✗ l'accès à l'archive d'Elektor
- ✗ 10% de remise dans l'e-shoppe
- ✗ 10 x magazine numérique
- ✗ 10 x magazine imprimé
- ✓ des offres exclusives
- ✗ l'accès à **Elektorlabs**
- ✗ un DVD annuel

connectez-vous

www.elektor.fr/inscription

79

pays

247031

membres actifs

1034

experts & auteurs

485

publications

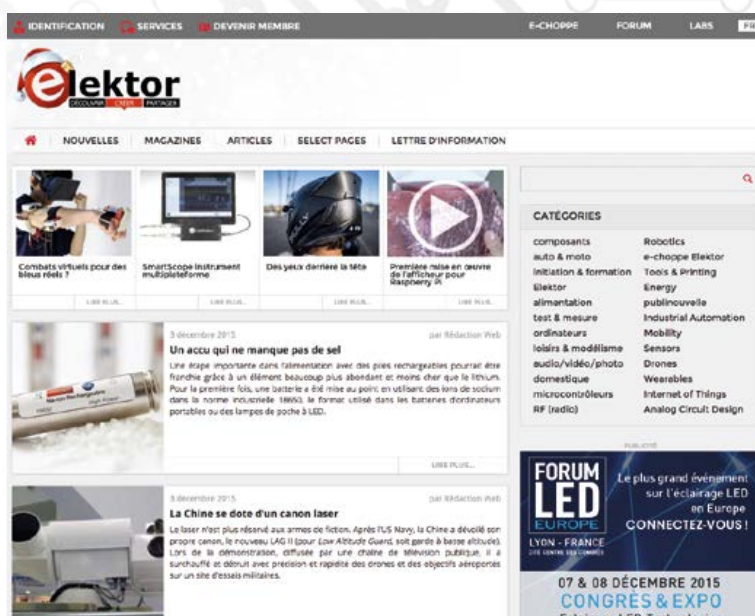
235102

visiteurs (par mois)

www.elektormagazine.fr

Explorez le monde de l'électronique

Entièrement rénové, notre site offre aux mordus d'électronique tout un monde de projets, d'articles et vidéos. Un bouton en haut du menu donne accès aux versions française, néerlandaise, allemande et anglaise du site. Le moteur de recherche intelligent permet de trouver rapidement informations et articles. Avec l'abonnement GREEN ou GOLD, vous pouvez naviguer dans toutes les archives d'Elektor et profiter de services complémentaires : offres et remises exclusives... Vous avez également accès aux préférences de votre compte, y compris l'abonnement au magazine papier et à la lettre d'information Elektor.



Elektor : gratuit & sans papiers

Restez informé, restez passionné !

Comme plus de 120 000 autres électroniciens, grâce à notre lettre d'information hebdomadaire gratuite Elektor, vous êtes au courant des dernières tendances et des récents développements de l'électronique. En outre, les abonnés à Elektor reçoivent régulièrement des offres et remises spéciales pour la boutique en ligne.

Pour vous abonner :
www.elektor.fr/inscription



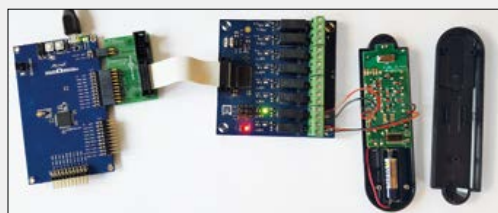
bienvenue dans la section DÉCOUVRIR

Jens Nickel



Mon périple dans l'Internet des Objets

L'apprentissage par la pratique est un principe que j'applique bien volontiers lorsque je chemine dans l'Internet des Objets – et vous pouvez me suivre sur www.elektormagazine.fr (cherchez les articles « Mon voyage dans le nuage »). Dans l'épisode 5 de cette série, j'ai réalisé pour la première fois une commande de type IdO qui fonctionne. Depuis n'importe quel PC dans le monde, je peux allumer ou éteindre la lampe de mon bureau privé. À l'aide d'une bibliothèque .NET disponible gratuitement, j'ai écrit un petit client MQTT qui peut envoyer et recevoir des messages. Comme station relais sur le réseau, j'utilise un serveur de test Mosquitto (<http://test.mosquitto.org>). Pour commander ma lampe



sans fil depuis le PC récepteur, j'ai vraiment bricolé. Les ordres sont transférés depuis le PC, via un port COM virtuel, vers la carte déjà connue SAM-D20 avec une extension à relais. J'ai raccordé aux relais une télécommande du commerce « piratée » qui, elle, commande la prise sur laquelle

est branchée la lampe. Toujours est-il que ça marche ! Grâce au protocole MQTT et au serveur sur le réseau, je n'ai plus à me préoccuper de la configuration du routeur et autres joyeusetés qui seraient nécessaires avec un serveur web personnel. Il y a encore du travail pour optimiser cette solution.

Le point faible reste la sécurité : n'importe qui peut allumer ou éteindre ma lampe à partir du moment où il connaît MQTT et ses commandes... Même si cela vous tente, ce n'est pas la peine d'essayer de jouer avec cette lampe, je l'ai depuis retirée du réseau ;-)

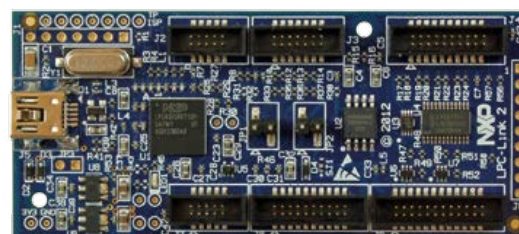
La puce ESP8266 ...

... semble devenir petit à petit un standard de fait lorsqu'il s'agit de relier un microcontrôleur à un réseau local sans fil (WLAN). Cela me fait penser aux puces FT232 de FTDI. Nombre de lecteurs et de développeurs de notre labo en attrapent une presque automatiquement lorsqu'une carte doit être dotée d'une interface USB pour communiquer avec PC.

Aujourd'hui, pour certaines expériences, je travaille avec la carte IoT WiFi de Pretzel (disponible sur www.elektor.fr). Elle contient un ATmega328, une puce ESP8266, l'alimentation et d'autres composants indispensables. Je n'ai pas rien à souder, ni à configurer, je peux attaquer directement la programmation et les essais. Pour moi, c'est idéal. Et pour vous ?



Programmateurs



et débogueurs

Dans le précédent numéro, nous avons passé en revue différentes cartes et divers kits à microcontrôleurs ; nous poursuivons le travail avec les programmeurs et débogueurs. Une fois encore, cet article a été écrit par un fanatique des contrôleurs, Viacheslav Gromov, l'un de nos plus jeunes lecteurs, mais qui est déjà un fin connaisseur de ce monde. C'est pourquoi cette année je l'ai volontiers chargé de rechercher sur le salon « Embedded World » de nouveaux contrôleurs et cartes pour Elektor. ◀

(150731)

relais à mercure

drôle de composants n°23

David Ashton (Australie)

Jadis, certains relais utilisaient le mercure comme milieu de contact. Les plus anciens d'entre vous se souviennent peut-être de ceux qui équipaient les organes de la télégraphie : les relais à tiges à contacts mouillés au mercure. Le mince film de mercure liquide qui recouvrait leurs contacts avait pour rôle d'augmenter la surface de contact, et par-là même d'en diminuer la résistance. Le film éliminait aussi en grande partie le rebond des contacts, car après la première fermeture la tension de surface du mercure maintenait la jonction, du moins si les contacts ne rebondissaient pas trop loin ; on pouvait donc construire des relais à la fois petits et rapides – les téléimprimeurs pouvaient fonctionner jusqu'à 100 bauds (= Hz ici). Il fallait régulièrement secouer ces relais afin que le mercure recouvre leurs contacts, et il fallait les monter d'une certaine façon pour éviter que ce même mercure ne provoque un court-circuit. Ce risque était généralement indiqué sur le composant (ce n'est pas le cas de celui de la **fig. 1**), tout comme était souvent mentionné le danger à ouvrir le relais en raison de la toxicité du mercure. Jan Buiting possède un oscilloscope Tektronix de 1959 qui utilise un relais à mercure pour produire des impulsions avec un temps de montée de 4 ns. Mais il vous en parlera dans sa rubrique Rétronique.

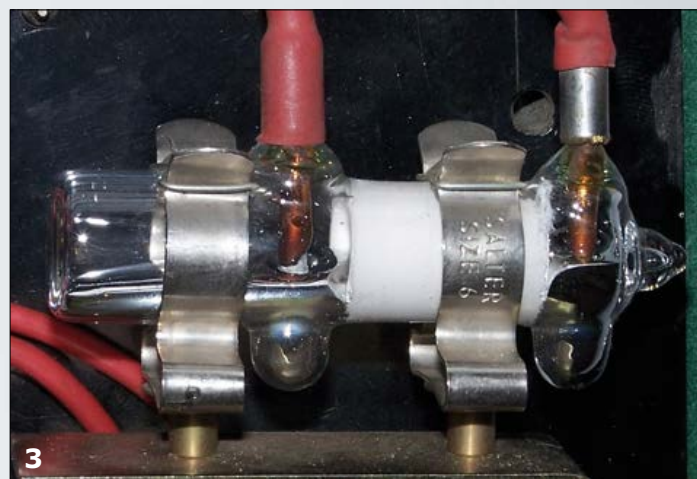
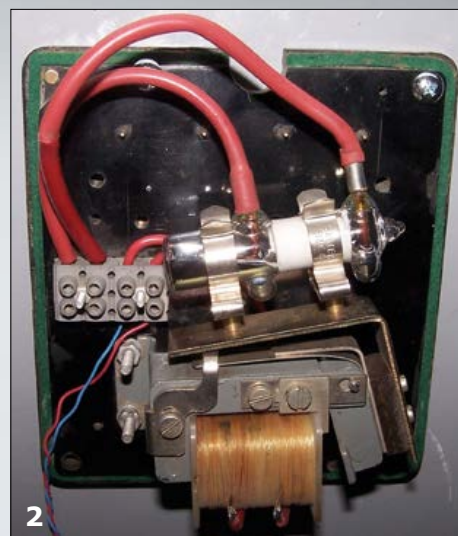
Le mécanisme à relais de la **figure 2** est l'une de mes belles et récentes découvertes. La bobine de 240 V CA est classique, mais au lieu de contacts de travail on a ici un relais dont l'armature incline un interrupteur à mercure formé d'un tube de verre rempli de mercure, et doté de deux contacts d'environ 2 mm de diamètre et 6 mm de long. Lorsque le relais est activé, le tube s'incline vers la droite et l'écoulement du mercure met les contacts en continuité électrique (**fig. 3**). Lorsque le relais est relâché, le liquide s'écoule vers la gauche et la quantité restante ne suffit plus à recouvrir les contacts et à maintenir la continuité électrique. L'interrupteur et la bobine du relais sont reliés à des bornes au moyen de câbles souples multibrins. Le diamètre du tube est réduit au niveau du col par un tube intérieur, apparemment en céramique. Cette réduction garantit que le mercure n'établit la connexion entre les contacts qu'au moment où le relais est enclenché. Le volume de mercure est sans doute de 1 cm³.

J'ai aussi vu un relais à mercure dont le noyau de fer est attiré magnétiquement dans ou hors un bain de mercure pour en élever et abaisser le niveau, et ainsi établir ou rompre la connexion entre les contacts.

De nos jours, on ne rencontre plus guère de relais et interrupteurs à mercure. Je me souviens qu'enfant j'avais joué avec le vif-argent d'un thermomètre cassé, et je sais que certaines lampes fluocompactes contiennent encore du mercure, mais c'est une autre histoire... Le mercure était aussi utilisé dans

les interrupteurs à inclinaison (le contact s'ouvre ou se ferme selon l'inclinaison du composant) dont certains modèles, bien étanches, sont d'ailleurs encore fabriqués. Quoi qu'il en soit et malgré sa mauvaise réputation, ce métal argenté reste un liquide fascinant. ◀

(150672 – version française : Hervé Moreau)



Envoyez-moi vos idées en quelques lignes ou sous forme d'un article déjà rédigé : neil@gruending.net

programmeurs

outils de développement pour tous

Viacheslav Gromov (Allemagne)

Le mois dernier, nous avons vu qu'il existait de nombreuses cartes de développement à débogueur intégré. On gagne en indépendance avec son propre programmeur/débogueur pour transférer du logiciel de son ordinateur vers le microcontrôleur souhaité, débuser les pannes ou améliorer le code.

Pour à peu près chaque famille de microcontrôleurs, on peut obtenir du fabricant de semi-conducteurs un débogueur ou un programmeur, mais il y a aussi de nombreux outils universels de développement chez d'autres sources, indépendantes d'une marque. En comparaison du programmeur « normal », un débogueur offre davantage de fonctions supplémentaires. Il permet surtout au concepteur d'avoir une vue sur les adresses importantes en mémoire et dans les registres, en cours d'exécution ou lors d'un arrêt du programme en un point précis (*breakpoint*). Souvent, on peut aussi accéder activement au programme ou à la mémoire, ainsi que demander une exécution du programme instruction par instruction.

Au-delà de ces fonctions de base, les fabricants inventent sans arrêt d'autres fonctions de débogage, quand ce ne sont pas des traceurs qui enregistrent en cours d'exécution des informations, depuis la consommation jusqu'aux données en mémoire, sans interrompre le programme, de manière à examiner la situation avec le plus de réalisme possible.

Sur un simple programmeur, ces fonctions de dépannage n'existent pas. Les appareils de programmation sont faits pour écrire et lire la mémoire du microcontrôleur vite et facilement. Durant la phase de développement, un programmeur ne peut s'utiliser pour rechercher des erreurs que quand le logiciel est chargé dans le microcontrôleur et que l'on peut observer l'exécution des fonctions de l'extérieur. Après cette phase, le programmeur devient le moyen le plus aisé et le plus rapide pour transférer le micrologiciel sous forme d'un fichier de données. Il faut évidemment avoir sur son ordinateur un outil de débogage correspondant. Si vous en utilisez un obtenu du fabricant, il est toujours préférable d'avoir l'outil de débogage intégré dans l'environnement de développement (EDI) pour pouvoir à tout moment passer dans ce mode-là. Il fonctionne toujours mieux quand les paramètres sont exacts et que le pilote adéquat est installé. C'est souvent là que le bât blesse, du moins chez les débutants, parce qu'il peut y avoir une quantité impressionnante de réglages à faire.

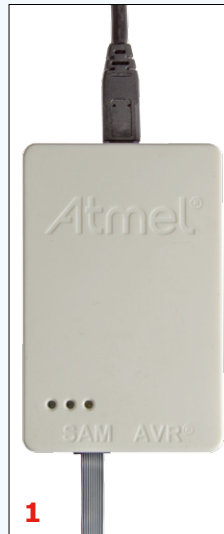
Nous allons passer en revue plusieurs débogueurs et programmeurs de différentes firmes. Brièvement et sans vouloir être exhaustif, nous verrons les débogueurs et programmeurs proposés par les fabricants de semi-conducteurs (ou des sociétés liées à ces fabricants), puis les débogueurs et programmeurs de sociétés indépendantes.

et débogueurs



Atmel

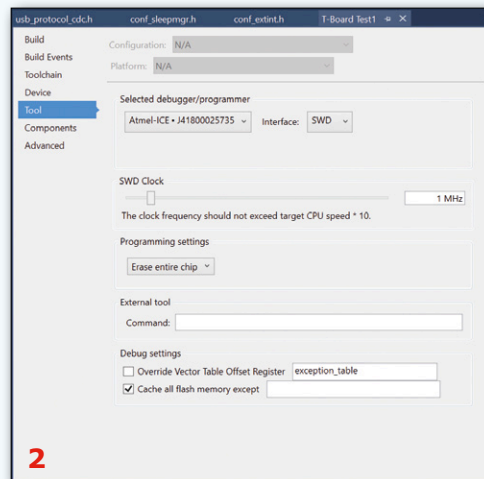
Atmel ICE [1] (**figure 1**) permet de déboguer et programmer *in situ* deux familles de microcontrôleurs Atmel très connues et étendues : les ARM Cortex M SAM à 32 bits et les AVR à 8 bits. Atmel ICE est disponible en différents kits. Le moins cher est une carte nue sans boîtier ni câble pour un peu plus de 40 €. Le kit de base contient le débogueur avec boîtier ainsi que le câble USB et le câble plat pour la programmation, mais il coûte plus de 60 €. Parmi les kits plus étoffés, il y en a un avec encore un câble plat supplémentaire et une petite carte d'adaptation pour le connecteur de programmation, mais pour plus de 110 €, il est nettement plus cher. Bien entendu, on peut ajouter soi-même le boîtier, le câble plat et la carte d'adaptation quand on a démarré avec un modèle à bon compte.



7,5 MHz. Trois LED de couleurs différentes sur le débogueur indiquent son état. La carte d'adaptation n'est nécessaire que pour une certaine configuration du connecteur, sinon le câble plat suffit. Le microcontrôleur à programmer est normalement alimenté de l'extérieur sous une tension de 1,6 V jusqu'à 5,5 V.

À la recherche d'autres de ces appareils, on trouve par exemple pour 50 €, l'AVR ISP mkII [2] qui programme exclusivement les contrôleurs AVR. Comme autre débogueur pour AVR, il y a l'AVR ONE! à plus de 700 €, mais extrêmement puissant et, disponible depuis plus longtemps, le JTAGICE3 [4] pour les familles SAM D et AVR, qui coûte moins de 130 €. Sur l'internet aussi, il y a quantité de programmeurs AVR de constructeurs plus ou moins connus. Le moins cher, c'est le mySmartUSB light de Laser & Co. Solutions GmbH [5] : pratique, travaille par ISP et coûte environ 16 €.

En plus de la connexion USB 2.0 pour l'ordinateur, l'ICE d'Atmel a deux connecteurs de programmation pour AVR et pour la famille SAM. Celui pour la famille AVR dispose des interfaces aWire, debugWire, une **Program and Debug Interface** (PDI), une **Tiny Programming Interface** (TPI) et enfin une interface SPI (ISP). Indépendamment l'un de l'autre, les deux connecteurs sont équipés des interfaces **Serial Wire Debug** (SWD) et **Joint Test Action Group** (JTAG), sur lesquelles le brochage diffère de l'une à l'autre. Sur chaque interface de programmation, l'ICE d'Atmel est en mesure de travailler avec une fréquence d'horloge sur le bus de



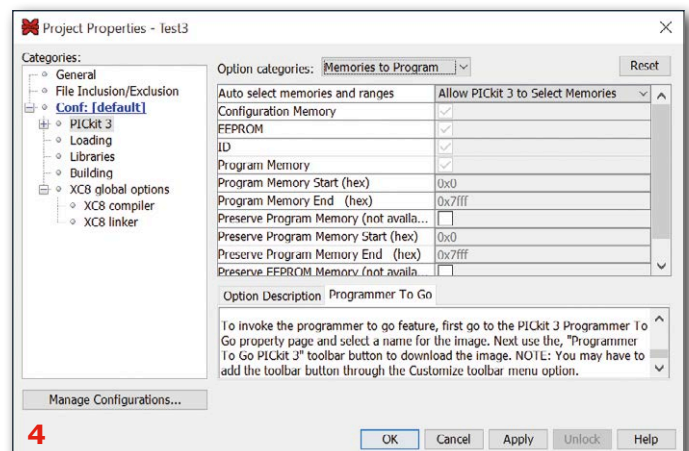
Tous les débogueurs fonctionnent avec le Studio 7 d'Atmel (**figure 2**) qui est gratuit et la plupart des autres EDI courants. Le Studio d'Atmel propose automatiquement une mise à jour du micrologiciel des débogueurs et programmeurs quand il y découvre une ancienne version.

Microchip



Le débogueur PICKit 3 [6] à environ 70 € pour la famille de microcontrôleurs (ds)PIC de Microchip (**figure 3**) est bien connu et fort répandu. Il accompagne aussi différents kits. Outre l'interface USB pour la liaison au PC, pour laquelle le câble est offert, on n'y trouve qu'un seul connecteur à six pôles pour la programmation *in situ* de tous les PIC courants avec l'interface typique ICSP (**In-Circuit Serial Programmer**), il peut donc aussi servir de programmeur. Le débogueur PICKit 3 est très puissant, il garantit de très hauts débits de données, il est aussi rapide que n'importe quel microcontrôleur. Il peut leur fournir 30 mA entre 1,8 V et 5 V. Il présente aussi trois témoins à LED (différentes

couleurs) de l'état du débogueur. Un bouton-poussoir donne accès à une fonction spéciale : **Programmer-To-Go**. Dans ce mode, on peut charger le logiciel dans le PICKit 3 et de là le copier dans le microcontrôleur d'une simple pression sur le poussoir, sans avoir recours à l'ordinateur. Pour de plus grands projets ou une

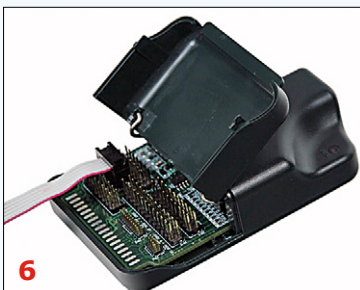
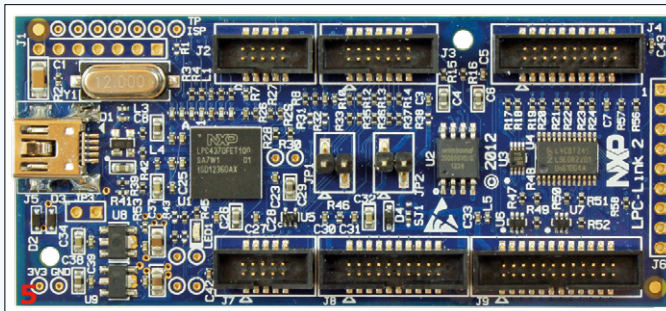


production en série, il y a de plus gros appareils, mieux adaptés, comme le programmeur en circuit Softlog ICP2 Production Quality [7] à partir de 400 €. Le PICKit 3 a été conçu pour l'EDI MPLAB X de Microchip et se

connecte sans aucun souci (**figure 4**), notamment du fait que les pilotes HID s'installent automatiquement sans objection de Windows.

NXP

Pour les familles LPC « normales » de NXP, il y a de longue date le LPC-Link2 (**figure 5**) qui s'installe sous une forme un peu plus simple sur la plupart des cartes de développement Xpresso et coûte seulement 20 € [8]. On trouve sur la carte plusieurs embases à collerette, sur les broches de l'une d'elles on accède, par les cavaliers J6 à J8, aux interfaces SWD et JTAG. Les autres connecteurs donnent accès aux broches du contrôleur de débogage à trois cœurs LPC4370. On peut réellement utiliser le débogueur comme carte de développement en programmant et déboguant le microcontrôleur LPC4370 de l'extérieur au moyen d'un autre débogueur. On dispose aussi du câble plat pour le relier au microcontrôleur cible.



Comme ce débogueur est basé sur la technologie CMSIS-DAP et J-Link, il est compatible avec la plupart des EDI comme l'actuel LPCXpresso. L'outil logiciel LPCScrypt permet de modifier et mettre à jour le micrologiciel sur le débogueur. Pour la récente série Kinetis (anciennement Freescale), c'est

le débogueur USB Multilink Universal (**figure 6**) de la firme P&E [9] qui s'accorde le mieux, il coûte environ 200 €. Son grand frère, avec le suffixe FX, est deux fois plus cher, mais plus puissant et s'accorde avec plus de microcontrôleurs : Kinetis, ColdFire, LPC, STM32, PSoC 4, etc. Sous le couvercle en plastique de ce débogueur se trouvent des barrettes pour lesquelles un câble plat est aussi fourni. En plus des interfaces SWD et JTAG, il y en a beaucoup d'autres spécifiques aux nombreuses familles de microcontrôleurs dont ce débogueur peut s'occuper. Les fréquences d'horloge peuvent aller jusqu'à 50 MHz et la tension d'alimentation peut se situer entre 1,6 V et 5,25 V. Le débogueur FX peut délivrer selon le besoin 3,3 V ou 5 V au microcontrôleur.

En théorie, il peut travailler avec la plupart des EDI actuels (Kinetis, Design Studio, Keil, IAR, etc.) qui acceptent le protocole et le pilote P&E.

STMicroelectronics

Le débogueur à 30 € ST-Link/V2 [10] basé sur un microcontrôleur STM32 ARM Cortex M3 convient aussi bien pour les microcontrôleurs STM32 ARM que pour ceux à 8 bits de la famille STM8 de STMicroelectronics (en bref : ST). Il est livré avec tous les câbles nécessaires (**figure 7**). Il est disponible sous forme amincie sur de nombreuses cartes ST.

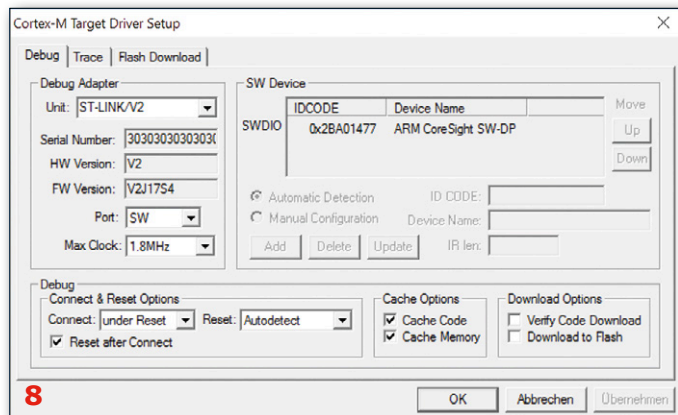
Outre l'interface USB 2.0 pour la liaison au PC, il est équipé d'un petit connecteur à quatre broches pour les STM8 et d'une double rangée de broches pour les STM32 pour pouvoir déboguer et programmer les microcontrôleurs directement sur la carte. L'interface pour STM8 s'appelle SWIM (**Single Wire Interface Module**) et ne requiert, en plus de la ligne d'alimentation, que de deux autres pour la RAZ et les données. Le ST-Link/V2 peut travailler en deux modes, à basse vitesse : 9,7 Ko/s ou à haute vitesse,



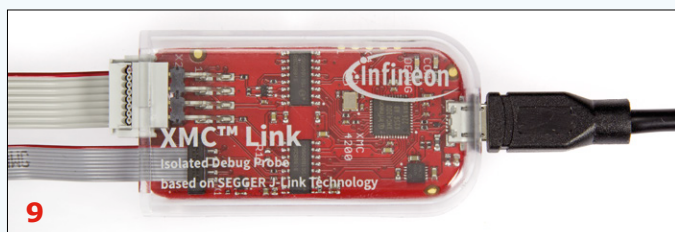
à 12,8 Ko/s et sous une tension entre 1,65 V et 5,5 V, alimentation à fournir de l'extérieur. Le gros connecteur pour la famille STM32 donne accès tant à l'interface SWD, typique d'ARM, qu'à l'interface JTAG plus élaborée. Les tensions d'alimentation permises vont de 1,65 V à 3,6 V, mais les entrées de données tolèrent aussi le 5 V. L'état de fonctionnement est affiché sur une LED bicolore qui s'allume ou clignote en différentes couleurs.

Son grand frère coûte plus du double, s'appelle ST-Link/V2-ISOL et diffère de l'autre par deux points. Il est doté, entre la prise USB et le connecteur de programmation, d'un isolateur (jusqu'à 2 500 V_{eff}) par souci de sécurité sur des applications critiques ; en outre, la connexion SWIM se trouve sur les 20 contacts de l'embase à collerette. Les cordons livrés sont câblés en conséquence.

Il va de soi qu'il existe des débogueurs plus chers qui couvrent la totalité de la gamme des microcontrôleurs ST, par exemple pour 150 € vous pouvez acheter le STX-RLINK [11], un débogueur universel qui marche avec pratiquement tous les EDI connus, depuis IAR et Keil en passant par Atolic, TASKING et STVD jusqu'à CooCox. La mise en service est généralement très simple, il n'y a que peu de paramètres à régler (**figure 8**) et Windows installe la plupart du temps les pilotes automatiquement. La mise à jour du micrologiciel est aussi possible à l'aide d'un outil gratuit disponible sur la page du produit.



Infineon

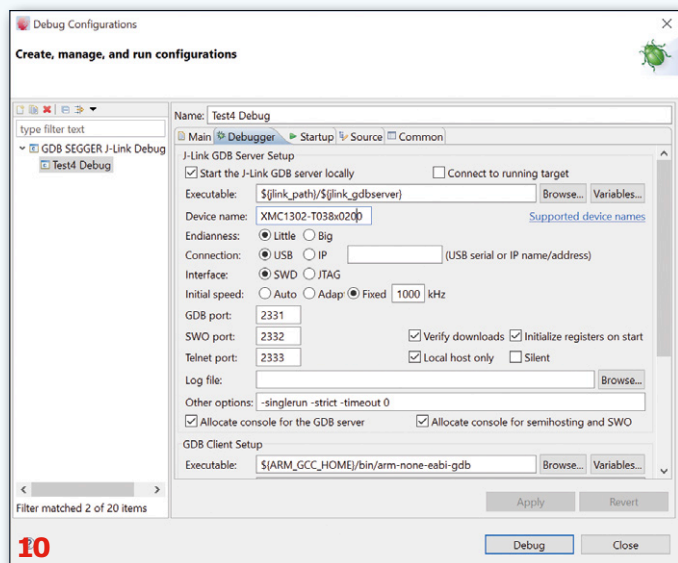


Chez Infineon il y a un nouveau débogueur, en forme de sonde (**figure 9**), pour la grande famille des microcontrôleurs XMC ARM Cortex M, le XMC Link [12]. Ce petit appareil construit à partir d'un contrôleur XMC4200, avec logiciel d'inspiration J-Link de SEGGER, remplit toutes les attentes de celui qui veut réaliser son projet avec un microcontrôleur XMC. Il dispose d'un connecteur à huit contacts pour les familles XMC1000 Cortex M0 et XMC4000 Cortex M4 ainsi que d'un connecteur typique et standardisé à dix contacts Cortex-Debug, uniquement pour la puissante famille XMC4000 Cortex M4.

Le premier connecteur sort une interface de débogage à une seule broche (**Single Pin Debug**, SPD) et une interface SWD, on trouve aussi une passerelle de UART vers USB, une interface série virtuelle pour la phase de développement. Sur le gros connecteur se trouve aussi une autre ligne pour le Serial Wire Viewer (SWV), c'est une fonction de traceur, et l'interface JTAG. Les deux connecteurs doivent être alimentés de l'extérieur séparément du microcontrôleur dans la plage de 2,5 V à 5,5 V. En outre, sur le débogueur sont implantées deux puces d'isolation qui protègent le PC de tensions jusqu'à 1 kV sur les

contacts de programmation. Les deux LED voisines de la prise USB indiquent si le débogueur est alimenté et s'il est occupé à quelque chose. Le débogueur coûte bien moins de 100 €, mais le prix exact ne nous a pas encore été communiqué. Un câble pour chaque connecteur fait partie de la livraison.

Tous les EDI actuels acceptent ce nouveau débogueur, y compris le DAVE d'Infineon, évidemment, où la relation avec le débogueur est sans nuages (**figure 10**). Prochainement sortira un outil XMC Flash qui simplifiera encore la programmation.



Texas Instruments

Le MSP-FET [13] de la **figure 11**, d'une valeur un peu supérieure à 100 €, couvre la totalité des familles de microcontrôleurs MSP430 et CC430 de Texas Instruments (TI). La famille CC430 a, en comparaison avec la famille à 16 bits MCP430 connue de longue date, différents périphériques supplémentaires embarqués pour assurer la transmission sans fil. Le MSP-FET ressemble aux eZ-FET bâtis sur LaunchPad, mais offre beaucoup plus de fonctions. Au lieu d'une prise USB, on y trouve un connecteur de programmation à 16 contacts qui mène à une interface JTAG.

Cela revient au mode Spy-Bi-Wire avec moins de lignes que d'habitude. La fonction EnergyTrace(++), une particularité de MSP-FET, aide à l'amélioration du code par rapport à la consommation de puissance. Elle est mesurée durant l'exécution et affichée. Les UART, I²C et SPI sont aussi disponibles, notamment pour les échanges de données avec le PC.

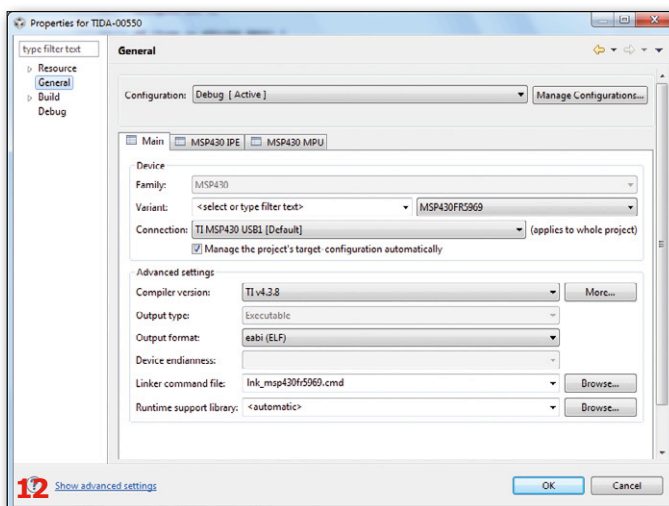
On peut programmer le microcontrôleur via UART ou I²C, à condition d'avoir installé au préalable le chargeur d'amorçage correspondant. Le MSP-FET est capable de fournir jusqu'à 100 mA



11

au microcontrôleur sous une tension réglable entre 1,8 V et 3,6 V. En plus de l'interface de programmation, il y a deux LED de couleurs différentes pour indiquer l'état actuel. L'ensemble comprend aussi un câble USB et un câble plat pour le connecteur de programmation.

À part le **Code Composer Studio (CCS)** en version normale et version Cloud, ce débogueur est aussi accepté par IAR (**figure 12**). Pour tous les autres compilateurs, on peut avoir recours au



12 Show advanced settings

besoin à MSP-Flasher qui peut programmer les données issues de n'importe quel compilateur dans le microcontrôleur MSP430.

Renesas

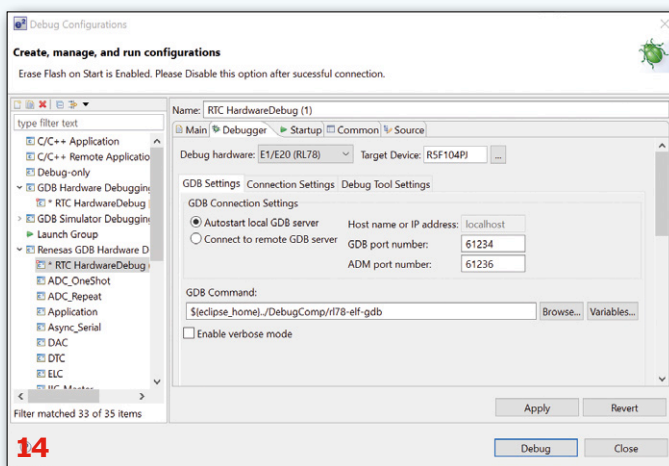


13

Comme Renesas met des familles de microcontrôleurs toujours plus intéressantes sur le marché, il faut bien sûr considérer aussi leur offre en débogueurs et programmeurs. Le plus avantageux, mais pourtant universel E1-Emulator (**figure 13**) peut fonctionner aussi bien comme débogueur que comme programmeur et coûte environ 200 € [14]. Il est fourni avec un câble USB et un câble de programmation. Il convient à plusieurs familles de microcontrôleurs Renesas, pas seulement pour les plus connues comme RL78, RX ou R8C, mais aussi V850, RH850 (pour l'automobile) et la famille de microcontrôleurs Smart Analog. Il y a deux connecteurs à 14 contacts sur l'Emulator pour les fonctions habituelles de débogage et de programmation. Le connecteur à l'opposé de la prise USB conduit directement aux interfaces UART, SPI et JTAG, protégées par un couvercle en plastique, tandis que la fonction d'auto-test reste accessible en face avant. Au moyen d'un outil sur la page du produit (après vous être annoncé), il est possible de tester les fonctions du débogueur pas à pas, l'un d'eux nécessite de relier ensemble les deux connecteurs à 14 bro-

ches. Sur la « véritable » connexion de programmation, on peut alimenter le microcontrôleur raccordé avec du 3,3 V ou 5 V et un courant jusqu'à 200 mA. Les LED sur les côtés témoignent des différents états du débogueur. E20-Emulator [15], le grand frère, dispose de plus de fonctions, mais coûte largement plus de 1 000 €.

Bien entendu, ce débogueur est compatible avec e²studio, la CubeSuite+ et beaucoup d'autres EDI connus (**figure 14**). Sur la page du produit, vous trouverez aussi des outils de programmation externes et gratuits, comme le programmeur Flash de Renesas.



14

Cypress

Pour la très particulière famille PSoC, il y a un petit débogueur du nom de MiniProg3 (**figure 15**) au prix de près de 80 € [16]. Il s'apparente à une clé USB et est doté de deux connecteurs de programmation. Il peut programmer les PSoC 1 jusqu'à 5LP, mais

ne peut déboguer que de PSoC 3 à PSoC 5LP. Il est prévu une embase femelle à cinq contacts pour brancher une petite carte et un connecteur normal à dix voies pour un câble plat joint à la livraison. Sur la prise à cinq contacts, il y a une interface ISSP

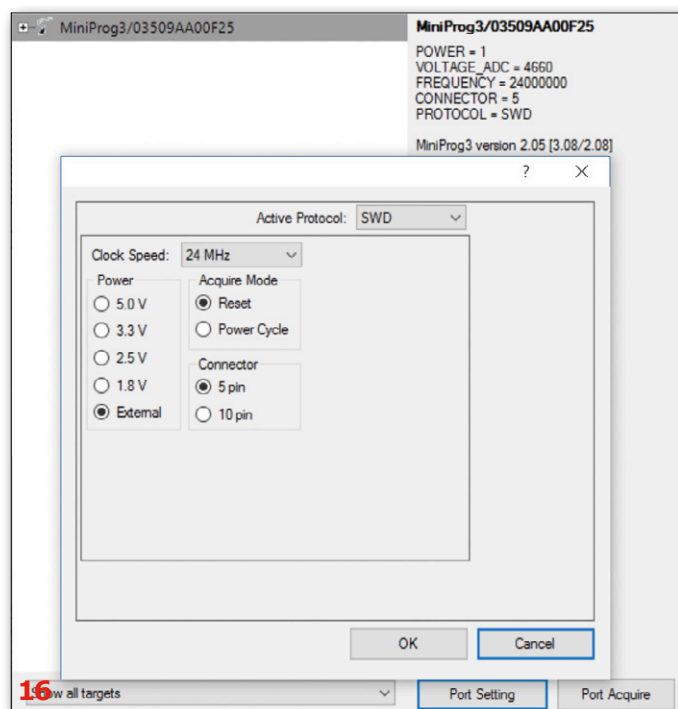


15

(*In-System Serial Programming*) et une interface SWD. Il y a en outre une interface I²C qui n'est prévue que pour l'échange de données avec un ordinateur par l'intermédiaire d'une passerelle I²C vers USB.

Sur le connecteur à dix voies, on peut trouver l'interface JTAG et une interface soit SWD, soit SWV. On se sert de JTAG et SWV uniquement sur les PSoC 3 et PSoC 5LP, l'interface SWD convient à tous les microcontrôleurs qu'il peut déboguer. Avec le PSoC 1, il faut utiliser l'interface ISSP. Le MiniProg3 est capable de délivrer tout au plus 200 mA pour alimenter le microcontrôleur à programmer, sous quatre tensions différentes au choix : 1,8 V, 2,5 V, 3,3 V et 5 V. Les cinq LED incluses dans le débogueur indiquent son état actuel.

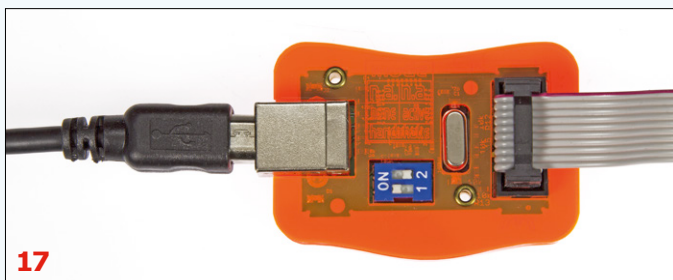
L'outil logiciel le plus diffusé pour ce débogueur se nomme PSoC Programmer, il est intégré à PSoC Creator et Designer (**figure 16**) et donc prêt à l'emploi au sein de l'EDI.



16

Appareils universels pour déboguer et programmer

Venons-en à présent aux constructeurs indépendants de débogueurs et de programmeurs universels. Il en existe de nombreux types largement répandus, mais faute de place nous en limiterons le panorama à trois d'entre eux, généralement peu coûteux et très utilisés.

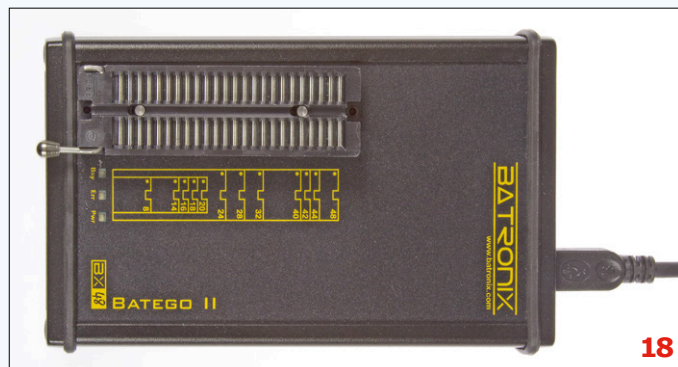


17

Commençons par le PROG-S [17] à une vingtaine d'euros de la société Diamex (**figure 17**). Avec ce programmeur, on peut lire et écrire du logiciel sur certains membres des familles de microcontrôleurs AVR, LPC et STM. Les câbles USB et de programmation sont fournis. Selon la famille de microcontrôleurs à programmer, il y a lieu d'adapter la position des deux microinterrupteurs sur le débogueur. Les deux LED indiquent l'état dans lequel il se trouve. Outre l'interface USB pour l'ordinateur, il n'y a qu'un connecteur de programmation à dix contacts par UART et SPI (ISP). Le SPI est conçu pour la programmation en ISP des microcontrôleurs AVR. PROG-S travaille alors avec un STK500 et le Studio d'Atmel peut l'adresser directement. Avec l'UART et deux autres lignes (RAZ et Boot), on peut programmer avec le PROG-S les microcontrôleurs LPC (logiciel : Flash Magic) ou STM32 (logiciel : STM32Prog), s'ils ont été munis du chargeur d'amorçage *ad hoc*. Il faut aussi que les microcontrôleurs à pro-

grammer soient toujours alimentés en courant de l'extérieur. Il y a aussi une configuration des interrupteurs qui permet de se servir de l'appareil de programmation comme passerelle UART vers USB. Pour connaître la liste des microcontrôleurs que cet appareil peut programmer, voyez la page du produit. Depuis peu, un successeur est apparu, un peu plus puissant sous le nom de PROG-S2 [18].

Une plus lourde pièce d'artillerie est certainement constituée par le BX48 Batego II de Batronix (**figure 18**), pour le portefeuille aussi : avec un prix d'environ 500 €, il est le matériel le plus cher de cette sélection [19]. Le microcontrôleur à programmer y est inséré seul dans un support de circuit intégré, l'appareil est conçu pour des lignes de production en série avec la stabilité et la qualité adéquates. Comme accessoires, on offre le câble USB et plusieurs logiciels sur CD. Cet instrument de programmation est principalement destiné – ses petits frères eux sont exclusivement destinés – aux puces de mémoire de toutes sortes, un avantage spécialement pour les grands projets quand le microcontrôleur requiert une mémoire externe. Si la puce est



18

CMS, on peut acquérir chez Batronix l'adaptateur correspondant. Au besoin, il existe aussi un adaptateur qui transforme le Batego en programmeur *in situ*.

Il peut programmer de nombreuses familles de microcontrôleurs, depuis les célèbres AVR et PIC jusqu'aux (bien connues des anciens, moins des modernes) 80C51 de Goldstar, Dallas, Intel et Philips. La programmation est très rapide, le logiciel Prog Express qui s'y rapporte est très intuitif et propose toutes les fonctions réclamées par la production en petite série. On peut par exemple faire reconnaître automatiquement par Batego des puces inconnues ou ajouter sur chaque puce programmée un numéro de série supplémentaire. L'un dans l'autre, un outil très professionnel !

Quand on associe ARM avec débogage, on pense tout de suite au débogueur J-Link de SEGGER (**figure 19**). Souvent, les fondateurs de silicium ajoutent sur leurs cartes d'évaluation le tracé d'un J-Link allégé ou développent leur propre débogueur en se basant sur la technologie J-Link. Pourtant, on peut se procurer le J-Link tout seul [20] et on en trouve des exécutions de bonne qualité et bien équipées.

La version économique pour les étudiants coûte environ 60 € et le modèle BASE, un peu plus de 300 €. À l'inverse du modèle bon marché, le nombre de points d'arrêt en mémoire flash n'est pas limité. Qui en veut davantage prendra le modèle PLUS, lui avec la licence nécessaire pour le logiciel, qui coûtera environ 600 €. Il existe des modèles encore plus puissants avec 3 Mo/s au lieu de 1 Mo/s de vitesse de téléchargement pour les fonctions spéciales de traçage ou de réseaux.

Tous les débogueurs J-Link fonctionnent avec tous les microcontrôleurs basés sur ARM 7, ARM 9 ou ARM11, ainsi que les

ARM Cortex. Indépendamment de cela, ils s'accommodent aussi des microcontrôleurs à 32 bits de Renesas RX et de la famille PIC32 de Microchip. Pour les programmer, il y a entre autres une interface SWD et une JTAG disponibles sur le connecteur de programmation à 20 contacts, les câbles USB et de programmation sont joints. Le J-Link peut si nécessaire fournir au circuit du microcontrôleur au maximum 5 V et 300 mA. On peut aussi y inclure un isolateur.

Fondamentalement, le J-Link dispose de nombreuses fonctions particulièrement intéressantes comme le mode de transfert en temps réel (RTT) dans l'échange de données avec le microcontrôleur pendant l'exécution ou un logiciel System View pour visualiser et analyser des paramètres déterminés en cours d'exécution. En tout cas, des logiciels pour le J-Link, on n'en manque pas. Pour la programmation on peut utiliser l'outil logiciel J-Flash. Pour la programmation pure, Segger propose aussi la série Flasher. Avec J-Link, vous avez à vos côtés un débogueur universel dans le monde des 32 bits (ARM) que pratiquement tous les EDI reconnaissent ! ◀

(150725 – version française : Robert Grignard)



Liens

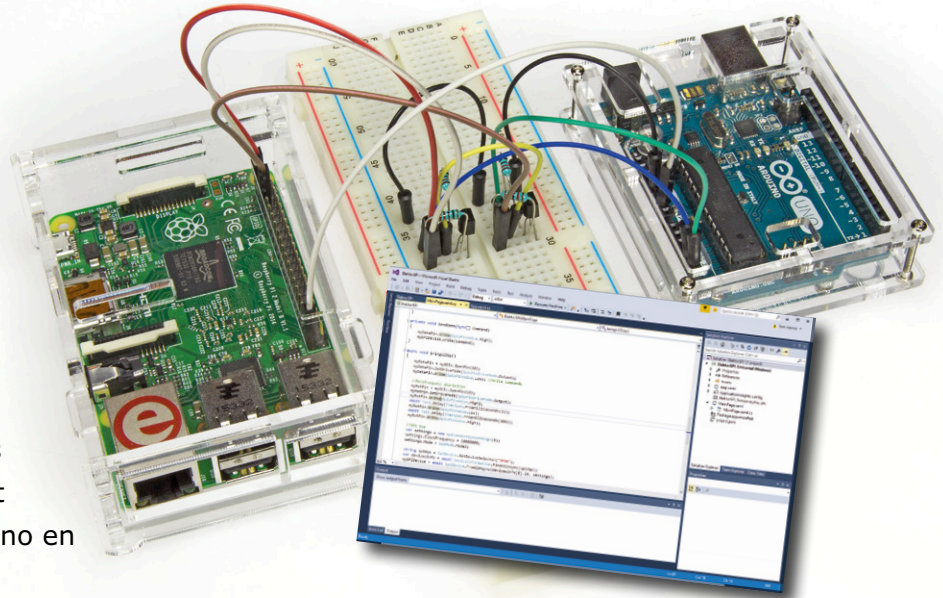
- [1] www.atmel.com/tools/atatmel-ice.aspx
- [2] www.atmel.com/tools/avrispmkii.aspx
- [3] www.atmel.com/tools/avrone_.aspx
- [4] www.atmel.com/tools/jtagice3.aspx
- [5] <http://shop.myavr.com/index.php?sp=article.sp.php&artID=200006>
- [6] www.microchip.com/DevelopmentTools/ProductDetails.aspx?PartNO=PG164130&utm_source=&utm_medium=MicroSolutions&utm_term=&utm_content=DevTools&utm_campaign=PIckit+3
- [7] www.microchip.com/Developmenttools/ProductDetails.aspx?PartNO=TPG100001
- [8] www.nxp.com/products/microcontrollers-and-processors/arm-processors/lpc-cortex-m-mcus/lpc-cortex-m4-single-multi-core/lpc4300-series/lpc-link2:OM13054
- [9] www.nxp.com/products/interface-and-connectivity/wireless-connectivity/sub-1-ghz-wireless-solutions/universal-multilink-development-interface:UMultilink
- [10] www.st.com/web/catalog/tools/FM146/CL1984/SC724/SS1677/PF251168?sc=internet/evalboard/product/251168.jsp
- [11] www.st.com/web/catalog/tools/FM146/CL1984/SC724/SS1677/PF122903
- [12] www.infineon.com/cms/en/product/productType.html?productType=5546d462501ee6fd015023aeb65733b3
- [13] www.ti.com/tool/MSP-FET#
- [14] www.renesas.com/products/tools/emulation_debugging/onchip_debuggers/e1/
- [15] www.renesas.com/products/tools/emulation_debugging/onchip_debuggers/e20/index.jsp
- [16] www.cypress.com/documentation/development-kitsboards/cy8ckit-002-psoc-minipro3-program-and-debug-kit
- [17] (en allemand) www.diamex.de/dxshop/mediafiles//Sonstiges/Prog-S-Anleitung.pdf
- [18] (en allemand) www.diamex.de/dxshop/USB-ISP-Programmer-fuer-AVR-STM32-NXP-Cortex-Prog-S2
- [19] (en anglais) www.batronix.com/shop/programmer/BX48/batego-II.html
- [20] (en anglais) www.segger.com/j-link-debugger.html

Windows sur la carte RPi (3)

SPI et I²C

Tam Hanna

Dans les projets à microcontrôleur, on utilise souvent les bus SPI et I²C pour la communication avec l'électronique externe. Windows pour RPi le permet aussi, nos deux démonstrations en apportent la preuve. Tout d'abord, nous allons piloter un mini-afficheur OLED, et ensuite nous utiliserons une carte Arduino en esclave de mesure.



Cela fait belle lurette que Microsoft opère dans le domaine de l'enfouï (*embedded*) – le *Gadgeteer* est un exemple de cette tradition de longue date. Tous les petits de Redmond ont en commun la misère de leurs performances : il n'existe pas de système Microsoft pour ordinateur industriel, abordable, qui réponde aux critères rigoureux du temps réel.

Sous Windows 10 pour RPi, la situation n'est pas meilleure lorsque les applications doivent se présenter sous la forme de *Managed Code*. Cela signifie que l'accès au matériel se fait par un détour, via le système d'exploitation, c'est cher en temps. Heureusement, le principal processeur du RPi, développé par Broadcom, intègre des moteurs pour I²C et SPI. La communication via les deux bus se fait par matériel ; le système d'exploitation se contente d'envoyer des ordres et de recueillir les résultats, deux opérations peu critiques en ce qui concerne le temps.

Afficheur OLED

L'intérêt majeur de RPi est sa capacité à fournir de l'HDMI.

Malheureusement, les panneaux compatibles HDMI sont souvent « hénarques » et gourmands.

Aujourd'hui, les fabricants chinois inondent le marché d'afficheurs OLED de 128 x 64 pixels monochromes à faible bande passante. L'auteur a acheté l'afficheur de la **figure 1** chez AliExpress. On en trouve de similaires sur eBay et dans les magasins d'électronique bien achalandés.

Le contrôleur intégré dans cet afficheur est réputé pour sa flexibilité. Outre I²C, il « parle » aussi deux variantes de SPI ; le choix de la forme de communication active se fait par soudage de résistances sur l'arrière de l'affichage. La **figure 2** montre les composants soudés d'origine, qui paramètrent le module dans un mode de communication propriétaire, très proche de SPI. La **figure 3** dévoile la connexion à RPi, le projet est facile à monter sur une plaque d'essais (**fig. 4**).

Le lecteur attentif n'aura pas manqué de remarquer ici l'absence de retour de l'afficheur vers RPi. Il ne s'agit pas d'une erreur de conception : notre afficheur ne peut pas envoyer de données à l'hôte.

Nous pouvons envoyer en alternance des données et des instructions de commande au contrôleur de l'afficheur, la broche DC lui indique de quoi il retourne. L'interconnexion de l'entrée de réinitialisation et de la broche GPIO est nécessaire, car le contrôleur a besoin d'une impulsion de réinitialisation au démarrage.

Maintenant nous sommes prêts à réaliser une nouvelle appli universelle. Le code complet est, comme toujours, disponible au téléchargement sur le site d'Elektor [3] sous forme de projet Visual Studio.

Instructions pour l'afficheur

Nous commençons par déclarer un groupe de variables indispensables pour la communication :

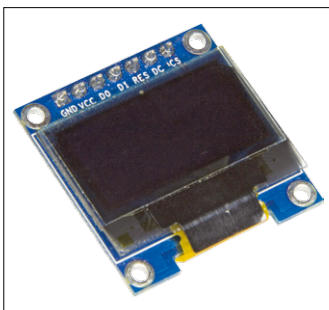


Figure 1. L'afficheur OLED – petit mais costaud !

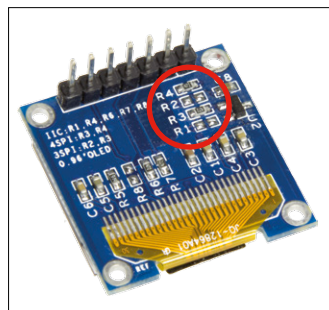


Figure 2. Des résistances CMS paramètrent le mode de commande de l'afficheur (SPI dans le cas présent).

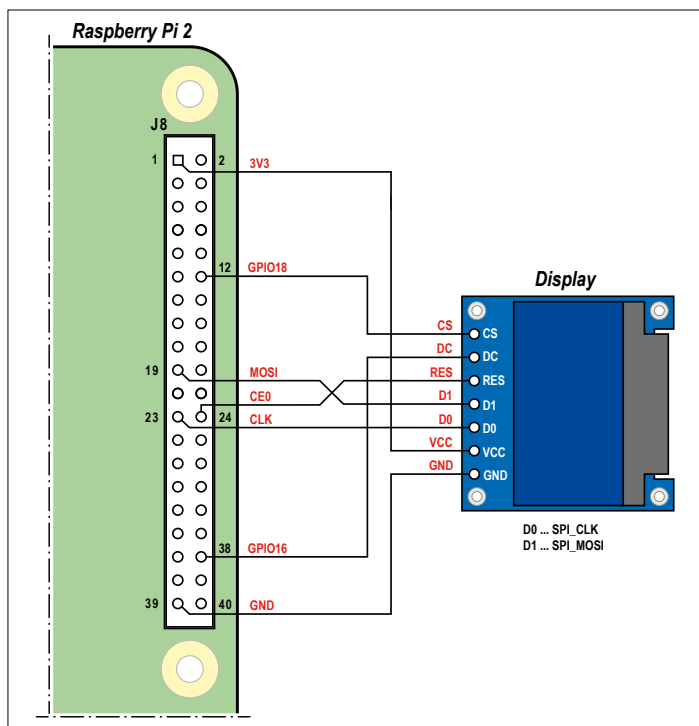


Figure 3. Cinq lignes suffisent pour piloter l'afficheur.

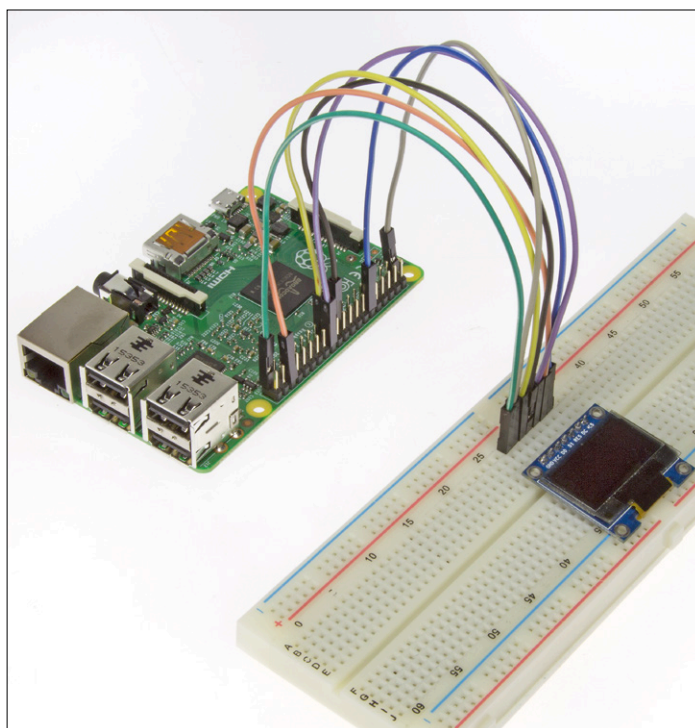


Figure 4. Il est possible de monter notre projet « affichage » sur une simple plaque d'essais.

```
GpioController myGCIc;
GpioPin myDataPin;
SpiDevice mySPIDevice;
```

```
GpioPin myRstPin;
```

```
byte[] gfxBuf = new byte[128 * 64 / 8];
```

Ici, deux points pertinents. Nous définissons une matrice d'octets qui sert de mémoire pour les données sur les pixels à transférer vers l'afficheur. Ensuite nous créons les instances globales de toutes les broches et du périphérique SPI – c'est important, sinon le ramasse-miettes (*Garbage Collector*) risque de faire des siennes (cf. encadré).

On trouve ensuite quelques variables globales remplies avec les octets d'instructions à envoyer. Ici on peut parfaitement s'aider de pilotes pour Arduino, proposés par ex. par Adafruit ou d'autres fournisseurs/distributeurs. Dans notre cas, la situation fut encore plus simple puisque Microsoft met à disposition un exemple de code :

```
private static readonly byte[] CMD_DISPLAY_OFF =
    { 0xAE };
private static readonly byte[] CMD_DISPLAY_ON =
    { 0xAF };
private static readonly byte[] CMD_CHARGE_PUMP_ON =
    { 0x8D, 0x14 };
private static readonly byte[] CMD_MEM_ADDR_MODE =
    { 0x20, 0x00 };
...
```

Listage 1. Démarrage de l'afficheur.

```
public MainPage()
{
    this.InitializeComponent();
    myGCIc = GpioController.Default;
    Random myRND = new Random();
    myRND.NextBytes(gfxBuf);

    bringLCDUp();
}

sync void bringLCDUp()
{
    myDataPin = myGCIc.OpenPin(16);
    myDataPin.SetDriveMode(GpioPinDriveMode.Output);
    myDataPin.Write(GpioPinValue.Low); //Write commands

    //Resetsequenz abarbeiten
    myRstPin = myGCIc.OpenPin(18);
    myRstPin.SetDriveMode(GpioPinDriveMode.Output);
    myRstPin.Write(GpioPinValue.High);
    await Task.Delay(TimeSpan.FromMilliseconds(1));
    myRstPin.Write(GpioPinValue.Low);
    await Task.Delay(TimeSpan.FromMilliseconds(100));
    myRstPin.Write(GpioPinValue.High);

    ...
}
```

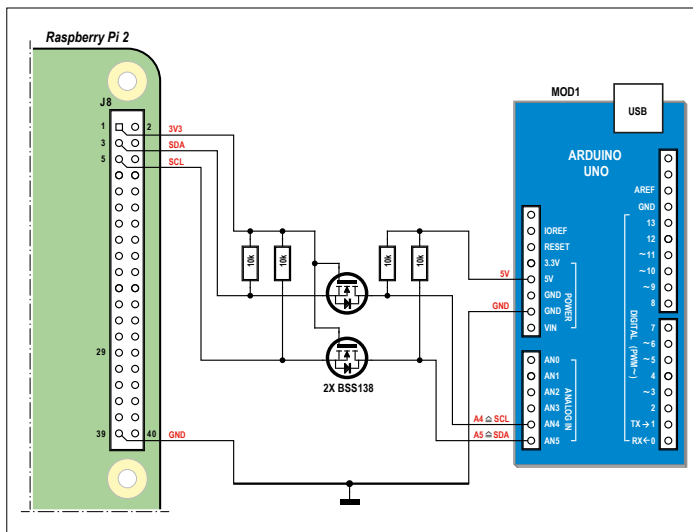


Figure 5. Les convertisseurs de niveau protègent le Raspberry Pi 2 ; la tension de signal de 5 V de l'Arduino Uno ne risque donc pas de l'endommager.

Dans le constructeur de *MainPage* (cf. partie 1 [1]), nous appelons la fonction `bringLCDUp`, et engrangeons des données aléatoires dans la mémoire de l'afficheur (**listage 1**). Les données aléatoires sont importantes parce qu'un afficheur OLED complètement noir ne donne pas le moindre signe de vie – nombre d'informaticiens ont, lors de la programmation de jeux pour téléphone intelligent à écran OLED, redémarré leur machine pensant à un plantage total.

`BringLCDUp` commence par exécuter la séquence de réinitialisation prévue par la fiche de caractéristiques.

Il manquera ici, aux développeurs ayant l'expérience de .NET, la fonction de temporisation `thread.Sleep`. Cette méthode éprouvée est passée de mode : les applications en Windows RT ne doivent pas bloquer, mais toujours rester réactives.

SPI

Lors de l'étape suivante, on initialise le bus SPI. Sous Windows 10 pour IdO, le pilotage de bus matériels externes se fait toujours via un objet déclaré selon la procédure suivante :

```
//SPI bus
var settings = new SpiConnectionSettings(0);
settings.ClockFrequency = 10000000;
settings.Mode = SpiMode.Mode3;

string spiAqs = SpiDevice.GetDeviceSelector("SPI0");
var devicesInfo = await DeviceInformation.
    FindAllAsync(spiAqs);
mySPIDevice = await SpiDevice.
    FromIdAsync(devicesInfo[0].Id, settings);
```

Pour que l'afficheur soit prêt à fonctionner, il ne nous reste plus qu'à transmettre l'instruction d'initialisation et à écrire les informations contenues dans le tampon d'affichage. C'est ce dont se charge la dernière partie de `bringLCDUp` :

```
SendCommand(CMD_CHARGEUMP_ON);
SendCommand(CMD_MEMADDRMODE);
SendCommand(CMD_SEGREMAP);
SendCommand(CMD_COMSCANDIR);
SendCommand(CMD_DISPLAY_ON);

SendCommand(CMD_RESETCOLADDR);
SendCommand(CMD_RESETPAGEADDR);
SendData(gfxBuf);
```

Les dernières lignes informent le contrôleur de l'arrivée de nouvelles données, qui auront leur place dans le bord supérieur gauche. Ensuite, l'ensemble du bitmap est transféré en une fois vers le contrôleur d'afficheur qui l'écrit sur l'écran et l'affiche. Les fonctions `SendCommand` et `SendData` ne diffèrent l'une de l'autre que par la mise à une valeur différente de la broche DC, avant l'activation de la transmission SPI.

```
private void SendCommand(byte[] Command)
{
    myDataPin.Write(GpioPinValue.Low);
    mySPIDevice.Write(Command);
}
```

RPi autonome

En fait, notre RPi est un piètre calculateur industriel : après une coupure de l'alimentation, il redémarre dans l'« appli de commande » (cf. partie 1 [1]), et ne poursuit l'exécution de son programme qu'après une nouvelle invitation via Visual Studio. Pour résoudre ce problème, il faut se connecter au Raspberry Pi 2 via *PowerShell*. Pour cela, ouvrez *PowerShell* sur votre ordinateur de bureau et entrez la commande suivante :

```
IotStartup list
```

Windows 10 y répond en affichant la liste des applis qui se trouvent dans le RPi. Si aucun nom n'apparaît, vous pouvez reconnaître votre programme à l'aide des UUID affichées dans les propriétés du projet.

Lors de l'étape suivante, on peut ajouter le programme avec l'instruction `IotStartup add ...`. On trouvera de plus amples infos à ce sujet sur <https://ms-iot.github.io/content/en-US/win10/tools/CommandLineUtils.htm>. Microsoft modifie la syntaxe de la commande assez souvent ; il faudra donc, avant de l'utiliser, jeter un coup d'œil sur le site internet.

Dernier point – ne serait-ce que pour des raisons d'exhaustivité – mentionnons que le RPi Windows 10 est un ordinateur Windows complet. Cela signifie qu'il aime bien être mis à l'arrêt. Cette tâche peut être accomplie en tapant `shutdown /s /t 0` (lorsque l'écran de démarrage apparaît, le RPi peut être déconnecté).

```
private void SendData(byte[] Command)
{
    myDataPin.Write(GpioPinValue.High);
    mySPIDevice.Write(Command);
}
```

SPI permet également la lecture de données ; pour en savoir plus, rendez-vous sur le lien internet [4].

Bus 1-Wire...

SPI peut être une norme de transmission très rapide ; cependant pour l'acquisition de température, le standard reste le bus 1-Wire de Maxim.

Il permet l'installation de réseaux de capteurs, alimentés tant en énergie qu'en informations par une ligne de données et une ligne de masse. Sous Linux, le Raspberry Pi en version native peut dialoguer avec ces capteurs : les développeurs de Raspbian ont créé un module de noyau (*kernel*) qui émule 1-Wire par *bit-banging* (exécution d'un protocole série par logiciel). Sous Windows RT, cela n'est pas possible puisque la structure asynchrone du système d'exploitation rend impossible la définition de retards précis (de l'ordre des microsecondes). À cause du ramasse-miettes, il est impossible de définir avec précision des temporisations inférieures à la milliseconde.

Jusqu'à présent, les tentatives de *bit-banging* ont échoué. Lors de certains débats, même les employés de Microsoft ont admis qu'il est impossible d'utiliser Windows 10 comme base pour du *bit-banging*. En tant qu'alternative on propose – comme bien souvent – d'utiliser un contrôleur externe. Maxim propose un circuit, le DS2482, qui permet d'accéder à son réseau 1-Wire via I²C.

... et Arduino

Tous ceux qui ont assimilé les enseignements du misanthrope Murphy, connaissent les problèmes d'approvisionnement en composants. Le jour où vous avez impérativement besoin de l'adaptateur 1-Wire en question, il est impossible à trouver en Europe.

C'est pour cette simple raison qu'il serait bon de convertir une carte Arduino Uno en esclave de mesure chargé de nous envoyer les valeurs via I²C. Pour présenter cette collaboration, nous avons doté l'Uno d'un logiciel de calculateur simple (**listage 2**) qui réagit aux demandes de notre RPi (quatre octets 123, 100, 128, 200) avec une réponse prédéfinie de 16 octets (« HELLO... »). L'interrogation de capteurs 1-Wire n'a pas été implémentée ici, la programmation d'un Arduino ne constitue pas le sujet de cet article.

Connectez l'Arduino au RPi 2 selon le schéma de la **figure 5** ; le tout devrait alors ressembler à ce que montre la **figure 6**. Côté Windows 10, il faut à nouveau une appli universelle, qui cette fois crée une instance de la classe de périphérique I²C. L'initialisation proprement dite se déroule globalement comme dans le cas du bus SPI. On y fixe aussi la vitesse ; on utilise soit le mode standard, qui travaille à 400 kHz, soit une variante plus rapide.

Le **listage 3** reproduit le logiciel du RPi. Nous lisons les données I²C en provenance de l'Arduino avec la fonction `ReadPartial`. Leur résultat `result` comporte des informations concernant la communication, entre autres le nombre d'octets retournés

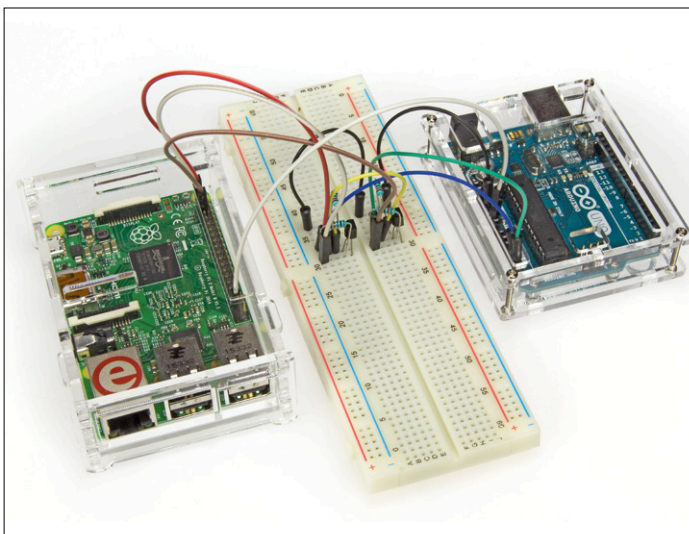


Figure 6. Arduino et Raspberry Pi travaillent « en tandem ».

par l'Arduino. Nous pouvons nous en servir pour une analyse.

En conclusion

Le Raspberry Pi 2 sous Windows 10 ne répond pas aux attentes d'un développeur d'applications industrielles. Il ne fait aucun doute que la pile de l'interface graphique (*GUI*) et la connexion en réseau sont excellentes – malheureusement les performances pour des tâches de régulation difficiles sont loin d'être satisfaisantes. La possibilité de relier des capteurs adressables via I²C ou SPI simplifie l'acquisition de données, mais ne permet

Listage 2. Arduino transformé en « esclave de mesure ».

```
#include <Wire.h>

void setup() {
    Wire.begin(13);
    Wire.onReceive(receiveEvent);
    Serial.begin(9600);
}

void loop() {
    delay(100);
}

void receiveEvent(int howMany) {
    char b1=Wire.read();
    char b2=Wire.read();
    char b3=Wire.read();
    char b4=Wire.read();
    if(b1==123 && b2 == 0 && b3 == 128 && b4==200)
    {
        Wire.write("HELLOHELLOHELLO!");
    }
}
```

Listing 3. Logiciel Raspi et I2C.

```

namespace ElektorI2C
{
...

    public sealed partial class MainPage : Page
    {
        int myArduinoAdress = 13;
        private I2cDevice myI2CArduino;

        public MainPage()
        {
            this.InitializeComponent();
            initI2c();
        }

        async void initI2c()
        {
            string aqs = I2cDevice.GetDeviceSelector();
            var dis = await DeviceInformation.FindAllAsync(aqs);
            var settings = new I2cConnectionSettings(myArduinoAdress);
            settings.BusSpeed = I2cBusSpeed.StandardMode;
            myI2CArduino = await I2cDevice.FromIdAsync(dis[0].Id, settings);

            //Prozessrechner abfragen
            myI2CArduino.Write(new byte[] { (byte)128, (byte)0, (byte)128, (byte)200 });
            byte[] i2cReadField = new byte[16];
            var result = myI2CArduino.ReadPartial(i2cReadField);
            if (result.Status == I2cTransferStatus.PartialTransfer || result.Status == I2cTransferStatus.FullTransfer)
            {
                if (result.BytesTransferred >= 4)
                {
                    System.Diagnostics.Debug.WriteLine("Arduino works!");
                }
            }
        }
    }
}

```

aucune opération en temps réel : si le ramasse-miettes est au travail, rien ne l'arrête.

Ajouter une carte Arduino permet de contourner ces faiblesses. Si vous construisez vous-même la partie « temps réel » avec différents éléments (MCU AVR et platine), vous économiserez quelques euros. Cependant la question de la répartition des tâches – que fait RPi, que fait le calculateur en temps réel – reste un sujet qui remplirait tout un manuel ! ◀

(150520 – version française : Guy Raedersdorf)

Liens

- [1] www.elektormagazine.fr/150465
- [2] www.elektormagazine.fr/150519
- [3] www.elektormagazine.fr/150520
- [4] <https://msdn.microsoft.com/en-us/library/windows.devices.spi.aspx>

Ramasse-miettes

Si vous souhaitez voir le ramasse-miettes en action, supprimez la déclaration globale *myRstPin* et remplacez-la par une déclaration locale dans la méthode concernée. Lors de l'exécution d'un tel programme, le contenu de

l'afficheur apparaît brièvement, puis disparaît. C'est parce que le ramasse-miettes supprime l'objet « broche GPIO », et dans le cadre de cette opération, remet la broche à son état d'origine.

trucs et astuces

les lecteurs écrivent aux lecteurs

Encore des solutions futées qui facilitent la vie des électroniciens



Aux limites des moteurs à courant continu

Martin Weiß

On se sert de moteurs à courant continu pour mettre en mouvement ou déplacer des objets. Mais souvent, ces déplacements ou rotations du moteur ne doivent pas dépasser certaines limites. Une bonne solution consiste à mettre des interrupteurs de fin de course qui coupent l'alimentation du moteur à la position limite. Leurs contacts à ouverture sont mis en série avec les bornes du moteur, en général on met deux interrupteurs afin de limiter le mouvement dans les deux sens. Seulement voilà,

quand on commute un des interrupteurs, il n'est plus possible de renvoyer le moteur dans l'autre sens, puisque sa ligne d'alimentation est interrompue.

Comment faire ?

Il suffit d'ajouter une diode en parallèle sur chaque interrupteur de fin de course. Lorsque le moteur est stoppé dans un sens, la diode du contact ouvert est passante. Le moteur peut fonctionner, éventuellement au ralenti. En effet, la tension de déchet

de la diode diminue la tension sur le moteur, ce qui en réduit la vitesse ou le couple d'entraînement. Mais dès que l'interrupteur de fin de course est à nouveau au repos, le moteur reprend sa vitesse normale.

Encore faut-il que les diodes soient judicieusement choisies en fonc-

tion du courant du moteur pour mettre en œuvre cette solution simple et peu coûteuse. Dans le circuit de l'article *Commande pour moteurs à courant continu* [1] par exemple, on peut utiliser des diodes ordinaires de type 1N5401 qui supportent un courant de 3 A.

[1] www.elektormagazine.fr/140562

(150662 – version française : Robert Grignard)



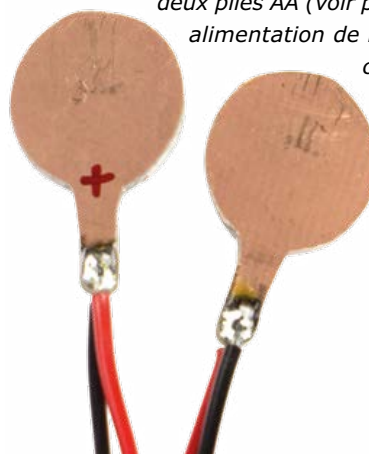
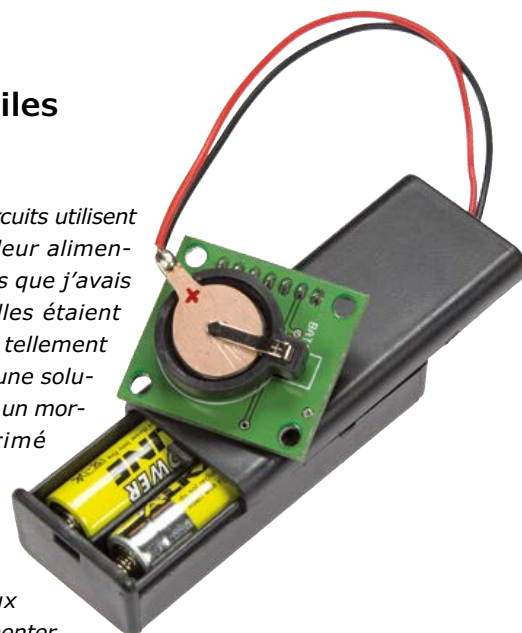
truc pour les piles boutons

Jean-Robert Pecheur

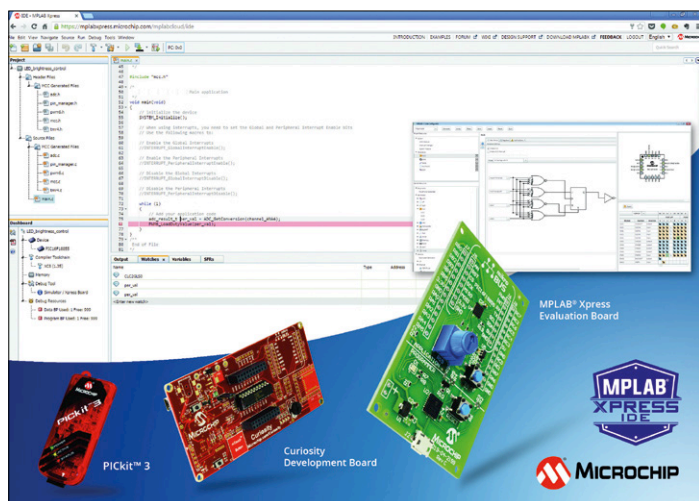
Certains appareils ou circuits utilisent une pile bouton pour leur alimentation. Mais chaque fois que j'avais besoin de ces piles, elles étaient déchargées. J'en ai eu tellement assez que j'ai pensé à une solution de remplacement : un morceau de circuit imprimé double face, coupé aux bonnes dimensions, peut facilement être inséré dans un support de pile. Soudez-y deux fils, et vous pouvez alimenter votre circuit avec par exemple

deux piles AA (voir photo). Si vous voulez utiliser une alimentation de laboratoire, réglez-la sur 3 V, et c'est parti ! Cela fonctionne aussi pour des piles plus grosses : il suffit d'empiler les morceaux de circuit imprimé. ◀

(150596 – version française : Jean-Louis Mehren)



Vous avez une solution futée pour arranger une bricole... Une façon bien à vous d'utiliser un composant ou un outil... Vous savez comment résoudre un problème plus facilement ou mieux qu'avec la solution actuelle... Écrivez-nous – chaque astuce publiée vous rapportera 40 € !



Plateforme de développement pour PIC, gratuite et en nuage

Microchip lance *MPLAB Xpress*, un EDI en nuage pour PC, portables et tablettes. Il est gratuit et ne nécessite ni téléchargement, ni inscription, ni installation. Xpress offre la plupart des fonctions populaires de l'EDI MPLAB® X. Il comprend une bibliothèque d'exemples validés par Microchip, l'outil *MPLAB Code Configurator* (MCC) 3.0 pour configurer un µC et produire du code en mode graphique, des compilateurs MPLAB XC, une prise en charge des programmeurs/débogueurs matériels, et 10 Go d'espace de stockage en ligne accessible avec un compte *myMicrochip*. La migration d'un projet Xpress vers la version X complète est aisée. *MPLAB Xpress Community* permet en outre de partager du code, des idées et des connaissances avec d'autres utilisateurs.

La carte d'évaluation MPLAB Xpress est équipée d'un programmeur, du µC PIC16F18855, et d'un connecteur mikroBUS™ qui permet d'utiliser plus de 180 cartes Click™ de MikroElektronika. Xpress est égale-

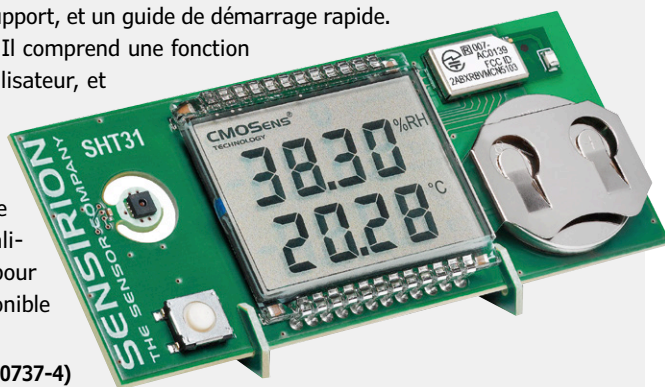
ment compatible avec la *Curiosity Development Board* de Microchip, une carte bon marché dotée d'un programmeur/débogueur ainsi que de connexions externes pour différentes cartes d'extension. L'EDI en ligne Xpress peut aussi être utilisé avec le programmeur/débogueur PICKit™3 de Microchip, un outil compatible avec plus de 1000 µC PIC®. (150737-1)

Kit de développement pour le capteur de température et d'humidité SHT31

Le kit de développement « SHT31 Smart Gadget » de Sensirion comprend une carte principale équipée du capteur de température et d'humidité SHT31, un afficheur LCD, un poussoir, un module BLE, une pile, un support, et un guide de démarrage rapide.

Le capteur SHT31 est réputé pour son intelligence, sa fiabilité et sa précision. Il comprend une fonction de traitement du signal, deux adresses I²C distinctes sélectionnables par l'utilisateur, et son débit va jusqu'à 1 MHz. Son boîtier DFN a une surface de 2,5 x 2,5 mm² pour une hauteur de 0,9 mm. Avec une tension de fonctionnement comprise entre 2,4 et 5,5 V, une consommation de 0,2 µA en mode veille et de seulement 2 µA en mode actif (pour une mesure par seconde), il trouve sa place dans de nombreuses applications. Le capteur SHT31 fournit une sortie numérique calibrée, linéarisée et compensée en température. Sa précision type est de ±2 % pour l'humidité relative, et de ±0,3 °C pour la température. Le nouveau kit est disponible auprès de Mouser Electronics.

www.sensirion.com/humidity-development-kit (150737-4)



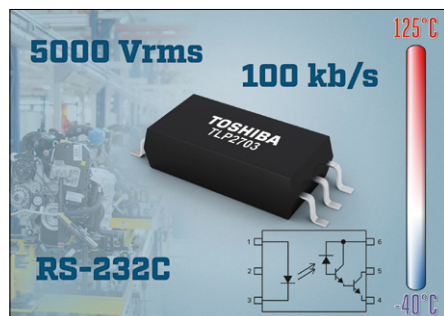
Isolateurs numériques LVDS pour environnements industriels difficiles

Les isolateurs numériques ADN465x d'Analog Devices pour transmission LVDS (*Low Voltage Differential Signaling*) améliorent les performances, la fiabilité et la consommation des applications à instrumentation et automate programmable industriel (API). La technologie d'isolement galvanique iCoupler® garantit la sécurité et la fiabilité des transmissions, et autorise des débits de 600 Mb/s avec une gigue de seulement 70 ps, et un temps de propagation max. de 4,5 ns. Avec les puces ADN465x, les signaux sériels LVDS très rapides peuvent désormais être isolés directement, sans « désérialisation ».

www.analog.com/en/products/interface-isolation/isolation/isolated-lvds.html (150737-3)



Photocoupleur pour communications RS-232C à 100 kbits/s



Compatible avec des débits jusqu'à 100 kbits/s, le photocoupleur TLP2703 de Toshiba Electronics Europe affiche un prix compétitif par rapport aux mêmes circuits intégrés de 1 Mbit/s. Il peut

être piloté par un courant de seulement 1 mA, et affiche un taux de transfert en courant (I_C/I_F) élevé, 900 % (min.) pour un courant d'entrée de 0,5 mA. Avec un temps de propagation garanti de 25 μ s (max.) pour un courant d'entrée $I_F = 1,6$ mA, et de 7 μ s (max.) pour $I_F = 12$ mA, la puce est adaptée aux interfaces isolées comme RS-232C. Le TLP2703 est logé dans un boîtier SO6L de 2,3 mm de hauteur. Avec une ligne de fuite de 8 mm et une tension d'isolement de 5 kV_{eff}, il convient aux applications nécessitant un isolement fort. Sa température de fonctionnement est garantie jusqu'à +125 °C.

www.toshiba.semicon-storage.com (150737-6)

Transformateurs de détection de courant compacts

Les nouveaux transformateurs de détection de courant CMS CST7030 de Coilcraft ne mesurent que 5,2 x 7,0 mm (hauteur max. 3,0 mm) et détectent jusqu'à 20 A à des fréquences comprises entre 10 kHz et 1 MHz. Les applications types sont la mesure de courants de charge, la commande d'alimentations à découpage et la protection contre les courts-circuits/surcharges. Les dispositifs satisfont la norme automobile AEC Q200 Grade 1 (-40 °C à +125 °C) et conviennent donc à la mesure de courant dans les moteurs de traction et les systèmes de gestion de batterie des véhicules conventionnels et électriques, dont les 48 V. Les CST7030 sont déclinés en cinq modèles de cinq rapports de transformation allant de 1/20

à 1/150 (primaire sur secondaire). Ils affichent une résistance CC du primaire de 0,0015 Ω , et 500 V_{eff} 1 min d'isolement entre les enroulements (tests d'isolement et diélectrique).

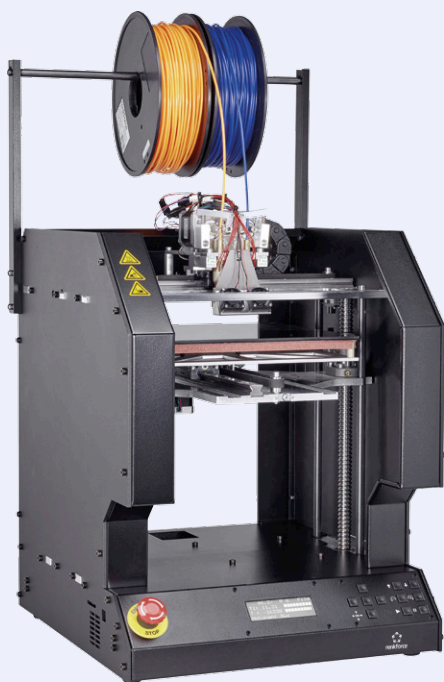
Il existe aussi une famille à isolement de 1500 VCC, la CU8965-AL. Le site de Coilcraft propose des échantillons gratuits.



www.coilcraft.com

(150737-8)

Imprimante 3D à double buse et unité de commande autonome



La nouvelle imprimante 3D RF2000 de Conrad succède à la RF1000 lancée en 2013. Elle possède deux extrudeuses, offre deux couleurs d'impression, et permet de créer des objets solubles dans l'eau. Pour plus de commodité, l'imprimante peut maintenant être commandée par une unité appelée Printbox (en option) en lieu et place d'un PC.

Comparée à la RF1000, l'éclairage de la chambre de la RF2000 offre une meilleure visibilité du processus d'impression, le plateau en vitrocéramique renforce l'adhérence des pièces, et un large écran LCD à contraste élevé améliore l'interface utilisateur. L'ajout d'un ventilateur supplémentaire et d'un bouton d'arrêt d'urgence améliore en outre le refroidissement et la sécurité de l'imprimante. L'unité de commande Printbox est prête à l'emploi. Elle est compatible avec la RF1000 et la RF2000, mais aussi avec d'autres imprimantes de marques connues, comme Maker-

Bot. La Printbox peut fonctionner en réseau local ou via le nuage (gratuit) Astroprint. Elle permet aussi de stocker des fichiers de modèles 3D et des images à haute résolution. Il est par ailleurs possible de surveiller le processus d'impression au moyen d'une webcam reliée à Astroprint.

www.conrad.com (150737-5)



MILIPOL PARIS 2015 : à point nommé ?



Patrick GUEULLE (France)

Pour sa 19^e édition en 30 ans, de nouveau jumelée avec *CARTES Secure connexions*, le salon mondial de la sécurité intérieure des États s'est tenu dans la lourde ambiance des attentats parisiens de novembre 2015. Une aubaine pour cet évènement de référence en matière d'antiterrorisme, mais aussi de risques majeurs en général, qui semble avoir ainsi fait le plein de visiteurs exceptionnellement motivés.

Des opportunités à saisir

La sécurité coûte cher, et ne bénéficie pas toujours de la priorité voulue en matière budgétaire : en France, le marché de la sécurité intérieure de l'État n'aurait ainsi progressé (selon l'atlas « En toute Sécurité » 2015) que de 1,3% en 2014 (à 3,4 milliards d'euros), contre un record de 34,2% en 2002 ! Même si les forces de l'ordre prennent grand soin des coûteux équipements dont on veut bien les doter, cela ne suffit sans doute pas à financer leur renouvellement et leur adaptation à la constante évolution des menaces.

Moyennant l'engagement des dépenses nécessaires (et peut-être la négociation à la baisse de tarifs souvent astronomiques), on trouve à Milipol tout ce qu'il faut pour assurer très correctement la sécurité des populations, ce qui a déjà été fait avec succès dans bon nombre de pays.

Même s'il est fascinant d'approcher, voire de manipuler, toutes sortes de matériels réservés aux troupes d'élite, nous avons plutôt centré notre visite sur la découverte de technologies relevant de l'électronique, aussi bien numérique qu'analogique. Aucun risque d'être déçu !

Le brouillage intelligent

Le brouillage et l'interception des radiocommunications ont toujours été un enjeu stratégique majeur, mais l'évolution vertigineuse des usages du spectre hertzien a totalement changé la donne. Aujourd'hui, les mouvements de foule s'organisent en temps réel *via* les réseaux sociaux, les explosions se déclenchent par téléphone portable (parfois depuis une cellule de prison !), tandis que des drones télécommandés ou programmés sont potentiellement capables de commettre, *incognito*, des attaques chimiques ou bactériologiques en tout point du territoire.

Des véhicules bourrés d'électronique peuvent cependant écarter temporairement ces menaces dans des secteurs fixes ou mobiles : à proximité d'une émeute, ou au sein d'un convoi d'officiels. Le groupe SESP a bien voulu nous expliquer comment fonctionne son fourgon MULTITASKER (**figure 1**), capable de faire pâlir d'envie bien des auteurs de fictions...

En présence d'une manifestation susceptible de dégénérer, le véhicule se gare à quelque distance et redresse, sur son toit, un panneau bardé d'antennes de téléphonie mobile assez directives. Grâce à un groupe électrogène embarqué (insonorisé et alimenté sur le réservoir principal), cela permet d'émettre des

puissances considérables (1600 W et plus !) en direction du rassemblement, sur les bandes de fréquences utilisées par tous les types de téléphones portables (430 MHz à 4 GHz) : plus de téléphonie, plus d'internet ou de réseaux sociaux, ni même de SMS. Un malaise s'installe, suite à cet isolement inattendu... Au bout d'un moment, le dispositif de brouillage passe en mode « chasseur d'IMSI », dispositif ultra-confidentiel capable de se déguiser en relais pour collecter les identifiants de tous les mobiles qui vont automatiquement tenter de se réinscrire sur leur réseau, dans un périmètre bien délimité. Cela suffit pour diffuser immédiatement, massivement, et sans passer par les opérateurs, des SMS d'intimidation du genre : « Nous savons qui vous êtes, rentrez chez vous ! ». La manifestation est alors censée se disperser piteusement et dans le calme.

En version fixe, des équipements comparables permettent d'éteindre complètement les réseaux de téléphonie mobile à l'intérieur des pénitenciers, avec toutefois le risque de perturber les communications légitimes dans le voisinage (ce qui est en principe interdit !). Un brouillage dit « intelligent » peut cependant être mis en œuvre, qui ne se déclenche que lorsqu'une tentative de communication illicite est détectée.

En version roulante, un brouillage omnidirectionnel à plus large spectre (20 MHz à 10 GHz) peut créer une zone de sûreté permanente autour d'un convoi circulant en terrain hostile : d'éventuelles bombes télécommandées ne recevront tout simplement pas leur ordre de mise à feu ! Et une variante émettant vers le ciel existe même pour envoyer « au tapis » les drones radiopilotes (qui étaient parallèlement les grandes vedettes du salon...).

Ajoutons des versions portables et autonomes, de puissances comprises entre 90 et 500 W dans 6 à 12 bandes. Et dire que l'on s'inquiète, dans le civil, de la possible nocivité de téléphones portables qui rayonnent, au maximum, 1 à 2 W !

Détection de jonctions non linéaires

Cette étonnante technique de détection de dispositifs électroniques dissimulés (en fonctionnement ou même à l'arrêt) n'est pas nouvelle : imaginée pendant la Seconde Guerre mondiale par Charles Bovill (un brillant ingénieur radio britannique), elle a d'abord servi à repérer des points de corrosion invisibles sur les avions !

Son principe est simple, mais génial : on illumine la zone suspecte avec une puissante

émission UHF (quelques centaines de MHz et quelques W), et on guette l'apparition d'harmoniques en retour. L'harmonique 3 est surtout renvoyée par des métaux corrodés (effet redresseur des oxydes), tandis que l'harmonique 2 est plus caractéristique des circuits électroniques (jonctions P-N). À quelques nuances près...

À partir des années 70 à 80, des applications dans le domaine des contre-mesures ont été



Figure 1. Système mobile de brouillage, mais également de diffusion de messages d'intimidation.

identifiées, mais ce n'est qu'assez récemment que de remarquables progrès ont été accomplis grâce aux techniques numériques. Ces avancées sont parfaitement illustrées par le dernier modèle ORION 2.4 HX de Research Electronics International (États-Unis), que nous avons eu l'occasion d'essayer de nos propres mains (**figure 2**).

L'émission se fait ici dans la bande « ISM » des 2,4 GHz (celle du Bluetooth et du WiFi !) en agilité de fréquence, modulation numérique, et large bande (1,25 MHz), avec une puissance apparente rayonnée (PAR) de 3,3 W ou même (uniquement pour les forces de l'ordre) de 6,6 W.

Des circuits de réception ultra-sensibles (-140 dBm) et très sélectifs (filtrage par DSP), associés à une antenne à polarisation circulaire, permettent typiquement de repérer une carte SIM à travers plusieurs centimètres de béton armé ! Mais nous avons fait mieux, en détectant avec succès la puce bien plus petite (0,8 x 0,8 mm) d'un simple ticket de bus RFID dont nous avions préalablement détruit l'antenne. Stupéfiant !

De puissants algorithmes de corrélation numérique entre les niveaux respectifs et les irrégularités des deux harmoniques, améliorent énormément la discrimination entre les composants électroniques et les artefacts, sans toutefois dispenser l'opérateur d'un certain savoir-faire.

Ce genre de matériel sert couramment à « faire le ménage » dans les salles de réunion, chambres d'hôtel, cellules pénitentiaires, véhicules, etc. Après tout, le premier modèle commercialisé ne s'appelait-il pas « the broom » (le balai) ?

Les éclairages tactiques

Au détour d'une allée, un stupéfiant faisceau lumineux nous a brusquement remis en mémoire une phrase lue dans un roman d'espionnage des années 60 (*Langelot et les saboteurs*) : « Une pile spéciale, de format minuscule, mais de puissance considérable, permettait à ce porte-clefs d'éclairer aussi bien qu'une grosse torche ». Aujourd'hui, ce n'est plus de la science-fiction : nous y sommes, et même au-delà ! Grâce aux

LED blanches à très haute luminosité (genre CREE XP-L HI) et aux chimies à base de lithium.

Figure 2. Détecteur de jonction non linéaire pour dénicher tous les circuits électroniques, même s'ils sont éteints.





Figure 3. M3XS-UT, une lampe torche à très longue portée : jusqu'à 1 km.

Si des fabricants français comme PETZL ont su innover avec des lampes frontales ultra-miniatures fonctionnant sur piles CR2032, le *made in China* à vocation militaire fait plutôt appel à des CR123A (piles d'appareils photo) ou des batteries lithium-ion rechargeables (typiquement 3,6 V / 3,6 Ah). La plus puissante des lampes-torche « durcies » présentées par OLIGHT (la M3XS-UT) éclaire ainsi facilement (de nuit) jusqu'à 1 km, grâce à une seule LED équipée d'un réflecteur optimisé (**figure 3**). Son intensité lumineuse peut atteindre 250 000 Cd (pendant une heure) pour un flux de 1200 lumens, à comparer avec l'intensité réglementaire (15 000 Cd) des feux de détresse pyrotechniques pour la marine !

Et comme si cela ne suffisait pas, il existe de véritables projecteurs équipés de six LED : 5000 lumens pour une portée de plus de 2 km !

À de telles puissances, l'éclairage peut être considéré comme une arme, capable d'aveugler durablement un adversaire agressif.

Et n'oublions pas que la technologie LED permet d'émettre des lumières colorées quasiment monochromatiques, ainsi

que des rayonnements infrarouges (aux fins de désignation de cibles ou d'illumination invisible pour la vision nocturne), et même ultraviolets (détection de sang, de faux documents, etc.). Les effets stroboscopiques (clignotements plus ou moins rapides) ont également leur utilité, et pas seulement pour accroître l'autonomie.



Bien entendu, l'usage de sources d'énergie performantes ne se limite pas à l'éclairage : piles à combustible portables, panneaux solaires déployables, batteries à chimies inhabituelles dans le civil, piles amorçables, ne sont que quelques exemples de solutions permettant d'alimenter dans la durée, « au milieu de nulle part », des équipements informatiques ou de communication, voire de petits systèmes d'armes.

Un peu comme dans le secteur automobile où les innovations de la Formule 1 finissent par entraîner des progrès sur les voitures de série, ces techniques pour le moment fort coûteuses s'inviteront probablement un jour dans notre vie quotidienne. Certaines ont d'ailleurs déjà embarqué sur des voiliers de course pour la dernière Transat Jacques Vabre !

Mais entretemps, elles nous auront peut-être sauvé la vie, en aidant nos « anges gardiens » à accomplir efficacement leurs missions de protection des personnes et des biens. ◀

(150641-1)



avril 2016

avril 2016

♦ Industrie Paris 2016

04 au 08/04 – Paris

www.industrie-expo.com

♦ Documentation management de l'information stratégique en entreprise

06 au 07/04 – Paris

www.documentation.fr



♦ Europadrone

06 au 07/04 – Lyon

www.europadrone-event.com



♦ SidO – Salon de l'Internet des Objets

06 au 07/04 – Lyon

www.sido-event.com



♦ RICV

rencontres internationales de cerfs-volants

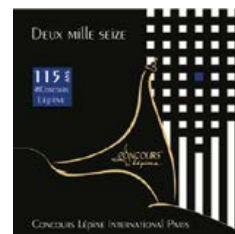
09 au 17/04 – Berck-sur-Mer

www.cerf-volant-berck.com

♦ concours Lépine

29/04 au 08/05 – Paris

www.concours-lepine.com



♦ Maker Faire Paris

30/04 au 01/05 – Paris

www.makerfaireparis.com

Maker Faire

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.

bienvenue dans la section **CRÉER**

Clemens Valens, labo d'Elektor



Y a-t-il une vie après Arduino ?

Arduino a été l'objet de toutes les attentions ces dernières années, au point d'éclipser quelque peu d'autres plateformes de prototypage, excepté Raspberry Pi. Certaines n'en sont pourtant pas moins intéressantes. C'est le cas de la plateforme mbed d'ARM, que nous vous avions présentée dès sa sortie, en 2010. Cette année-là vous aviez d'ailleurs été plusieurs milliers à participer à notre concours mbed. Joli départ, belle promesse de succès, seulement voilà, Arduino est arrivée en fanfare, Raspberry Pi a suivi, et mbed a fini dans les tiroirs de la plupart des utilisateurs. mbed n'a pourtant jamais cessé d'évoluer. De plateforme à une seule carte, elle est devenue au fil des ans un écosystème multi-cible soutenu par de nombreux fabricants de semi-conducteurs comme NXP/Freescale, ST, Atmel, Renesas et Silicon Labs.

La principale différence entre Arduino et mbed, par ailleurs toutes deux à code source et matériel ouverts, vient de ce qu'Arduino est une sorte d'incendie de forêt incontrôlable, alors que mbed est gérée de façon rigoureuse par une entité commerciale. Les industriels ont pondu de

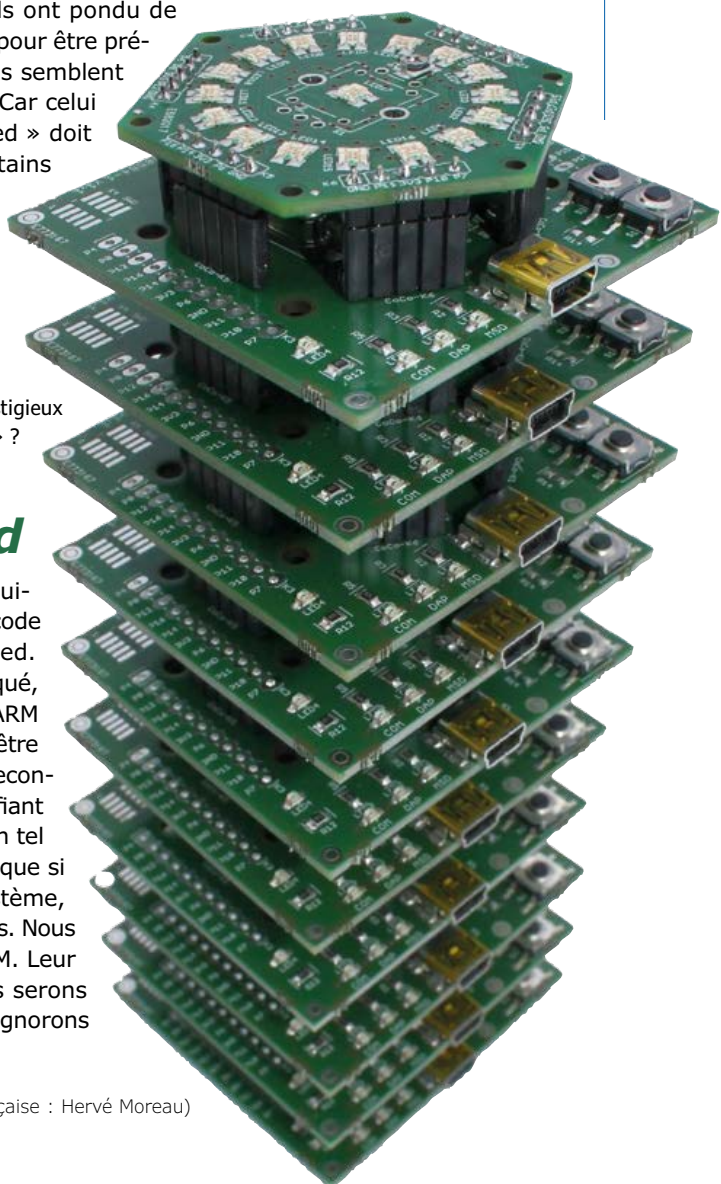
nombreuses cartes compatibles Arduino juste pour être présent dans la (basse-)cour Arduino, toutefois ils semblent quand même disposés à investir dans mbed. Car celui qui souhaite vendre une carte « mbed-enabled » doit obtenir l'approbation d'ARM, donc réussir certains tests. Elektor a décidé de tenter l'aventure. Réussirons-nous à créer la première carte mbed Elektor ?

<https://developer.mbed.org>

Ces dix cartes obtiendront-elles le prestigieux label « mbed-enabled » ?

Le label *mbed-enabled*

Une carte mbed est par définition une carte équipée d'un processeur ARM capable d'exécuter le code binaire produit par le compilateur en ligne mbed. Répondre à cette définition n'est pas trop compliqué, dès lors que la carte est dotée d'un processeur ARM compatible mbed. Mais cela ne suffit pas. Pour être estampillée *mbed-enabled*, une carte doit être reconnue par le compilateur en ligne grâce à un identifiant unique. Or, seule l'équipe mbed peut produire un tel ID et l'intégrer dans ses outils. Et elle ne le fait que si la carte satisfait certains critères. Sur notre système, notre carte a réussi les tests associés à ces critères. Nous en avons donc envoyé dix échantillons chez ARM. Leur examen est en cours. S'ils sont acceptés, nous serons invités à passer l'épreuve suivante, mais nous ignorons encore sa nature. Affaire à suivre... ◀



(150734 - version française : Hervé Moreau)

LEDitron

LED en filament pour afficheur à sept segments

Ilse Joostens & Peter S'heeren (Belgique)

Les lampes fluorescentes compactes sont passées de mode, les LED sont aujourd'hui les reines de l'éclairage. Le *LEDitron* utilise des LED en filament pour créer un afficheur à sept segments. Un peu comme un tube Numitron, mais à faible consommation.

Caractéristiques

- Tension de sortie ajustable entre 35 et 100 V
- Courant de sortie maximal 100 mA
- Tension d'entrée 12 V, 1 A
- Pour 4 à 6 afficheurs (en fonction du réglage de luminosité)
- Uniquement avec des composants à fils ou à pattes

L'avènement des lampes à basse consommation dans les années 80 n'est pas si loin. Les premières lampes fluorescentes compactes (LFC) n'étaient d'ailleurs pas si compactes que ça, et ressemblaient parfois à un pot de confiture avec un pas de vis ; leurs dimensions ont heureusement bien diminué depuis ! L'interdiction des ampoules à incandescence dans l'Union européenne a accéléré la prolifération des lampes à basse consommation, et vers 2010 les lampes à LED se sont présentées comme successeur des LFC. Une évolution récente est l'apparition de lampes à basse consommation avec des « filaments » de LED ; elles ressemblent aux ampoules classiques (**fig. 1**), et pour les nostalgiques il y en a même qui ont l'aspect de lampes à filament de carbone.

Ceux qui s'y connaissent un peu en électronique « vintage » ont sans doute entendu parler des tubes Numitron ; il s'agit en fait de lampes à incandescence

avec plusieurs filaments qui forment un afficheur à sept segments. L'inconvénient est qu'on n'en trouve en général que de petite taille, ce qui nous a donné l'idée de réaliser un plus gros afficheur avec des LED en filament.

On trouve actuellement ces « filaments » hors de leur ampoule (**fig. 2**), notamment via des revendeurs chinois sur eBay. Le résultat est impeccable, l'afficheur est en plus parfaitement lisible, même en plein soleil.

Principe de fonctionnement des LED en filament

Le développement des LED bleues à haute luminosité par Shuji Nakamura (Nichia Corporation) en 1993 a rendu possible l'utilisation de LED pour l'éclairage. Une LED blanche est une LED bleue superluminieuse, enduite d'une substance fluorescente qui modifie le spectre ; la couleur est alors perçue comme blanche par l'œil humain.

Les LED en filament ne sont qu'une évolution de l'utilisation des LED blanches. Un filament est constitué d'une série de LED placées sur un substrat transparent, en général du verre ou du saphir synthétique. Lorsque le courant est très limité, on peut voir sur le substrat les points bleus qui correspondent aux différentes LED (**fig. 3**). La barrette de LED est recouverte d'un enduit au phosphore et munie de fils à ses extrémités, afin d'en permettre le montage dans des armatures de lampes à incandescence à l'aide de points de soudure.

Vu le nombre élevé de LED en série sur chaque segment, la tension nécessaire à l'illumination est relativement élevée, de l'ordre de 70 V. Pour une alimentation par le secteur c'est plutôt un avantage, l'électronique nécessaire à placer dans le culot de l'ampoule n'en sera que plus limitée. Pour notre circuit, nous devrons bien entendu en tenir compte.



Alimentation et gestion des afficheurs

Le circuit se compose en gros de trois parties (**fig. 4**) :

- un convertisseur élévateur (*boost*) qui augmente le 12 V à 70-80 V,
- les pilotes des sept segments,
- et deux pilotes d'anode.

Le convertisseur élévateur fournit une tension de sortie ajustable avec P1,

d'environ 35 V à 100 V, pour un courant maximal d'environ 100 mA. Comme la tension de service du condensateur de sortie est de 100 V, il est préférable de ne pas dépasser 80 à 85 V, ce qui est plus que suffisant pour notre application. Nous avons choisi une configuration standard autour d'un temporisateur 555. Dans la pratique une telle alimentation est bon marché, relativement stable, et le dessin du circuit imprimé n'est pas trop critique.

Cependant nous avons optimisé le dessin de ce circuit imprimé. Les liaisons entre les condensateurs d'entrée, le MOSFET, la bobine, la diode et le condensateur de sortie sont les plus courtes possible, et les pistes de cuivre sont larges. La contre-réaction, bâtie autour de R38, R10, P1 et T1 est éloignée autant que faire se peut des zones du circuit où circulent de forts courants. Les condensateurs d'entrée et de sortie sont des modèles à



Figure 1. LED en filament dans une « ampoule à incandescence ».



Figure 2. Des LED en filament hors de leur ampoule invitent à l'expérience.



Figure 3. Lorsque le courant est très limité, on peut voir les différentes LED sur le substrat.

faible résistance série (*Equivalent Series Resistance – ESR*). Le résultat est un circuit qui malgré sa simplicité présente des caractéristiques honorables, et qui ne s'échauffe pas même lorsqu'il doit fournir 100 mA. Avec une tension de sortie de 82 V et une charge de 820 Ω , nous obtenons un courant de 100 mA (8,2 W), tandis qu'à l'entrée (12 V) le courant est d'environ 790 mA (9,48 W) ; le rendement dépasse donc les 86%.

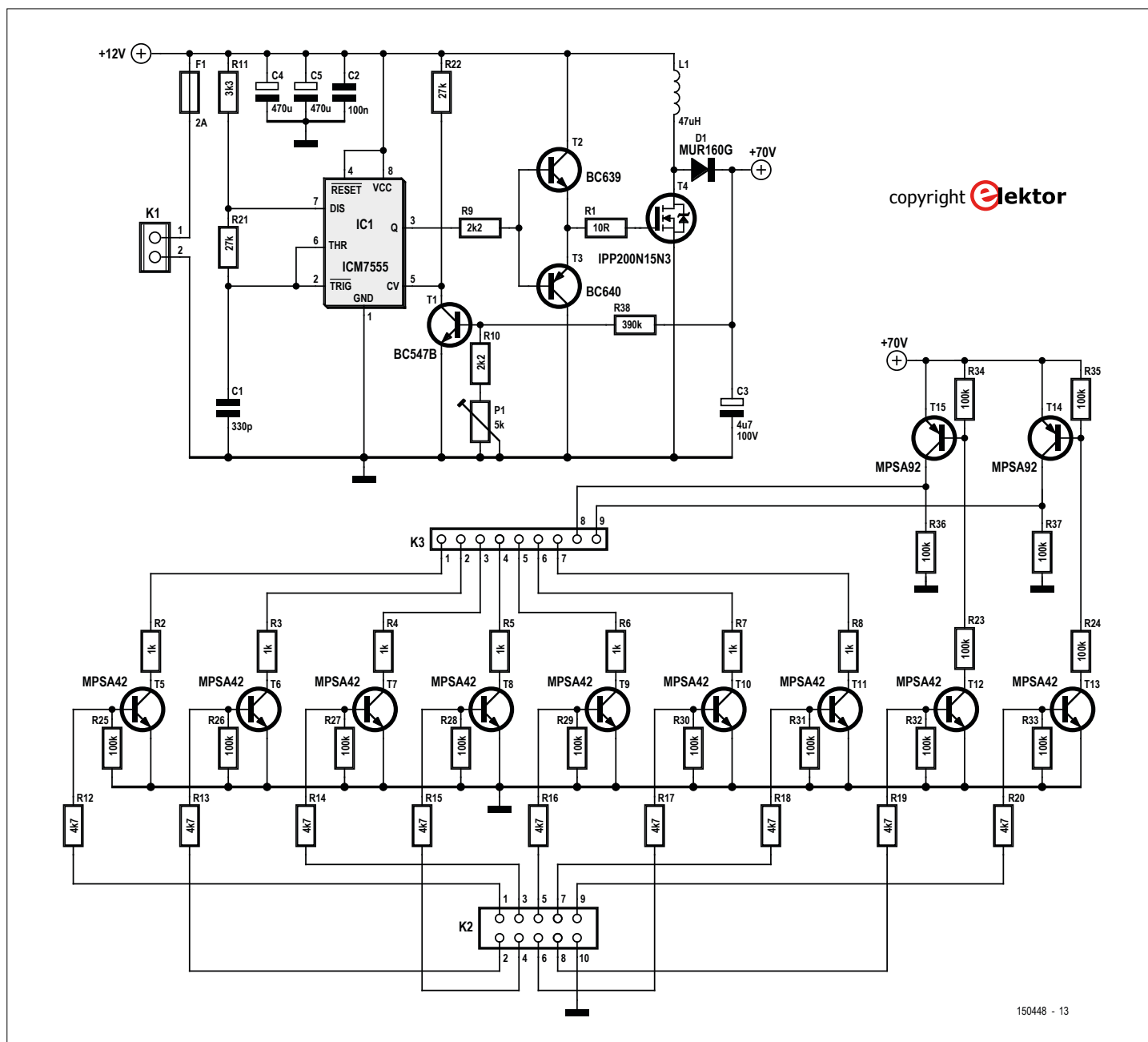
Le but est que le circuit soit monté à demeure et fasse partie d'un plus grand ensemble. Nous n'avons donc pas prévu de protection contre les inversions de

polarité. Nous n'avons pas non plus remplacé la diode – qui peut s'échauffer avec de forts courants – par un MOSFET canal P, ce qui aurait fait grimper les coûts. Nous avons tout de même prévu un fusible réarmable de 2 A.

Les LED en filament sont ici connectées en anode commune. Les pilotes de segment (cathode) sont construits autour de transistors NPN à haute tension MPSA42. Un niveau haut sur les broches 1 à 7 de K2 rend, via un transistor, le segment concerné conducteur ; le courant est limité par R2 à R8. En faisant varier la tension de sortie du convertisseur ou en

modifiant la valeur de R2 à R8, on peut ajuster la luminosité des segments. On pourrait aussi le faire – à partir d'un logiciel – en les alimentant avec une tension à rapport cyclique variable.

Il y a encore deux pilotes d'anode, bâtis autour de deux paires de transistors NPN-PNP (MPSA42-MPSA92), de manière à pouvoir multiplexer deux afficheurs. Si l'on veut connecter plus que deux afficheurs, on peut facilement ajouter des pilotes d'anode sur un morceau de circuit imprimé. Le nombre maximal d'afficheurs avec une seule alimentation est de 4 à 6, en fonction de la luminosité souhaitée.



copyright elektor

150448 - 13

Figure 4. L'alimentation présente un haut rendement du fait du découpage, et dégage donc peu de chaleur.

Composants de l'alimentation

Résistances :

R1 = 10 Ω
 R2 à R8 = 1 k Ω
 R9, R10 = 2,2 k Ω
 R11 = 3,3 k Ω
 R12 à R20 = 4,7 k Ω
 R21, R22 = 27 k Ω
 R23 à R37 = 100 k Ω
 (R36, R37 = en option)
 R38 = 390 k Ω
 P1 = potentiomètre ajust. 5 k Ω

Condensateurs :

C1 = 330 pF
 C2 = 100 nF
 C3 = 4,7 μ F / 100 V,
 à résistance série (ESR) réduite
 C4, C5 = 470 μ F / 25 V,
 à résistance série (ESR) réduite

Inductance :

L1 = 47 μ H (Würth WE-PD3 744 591 47)

Semi-conducteurs :

D1 = MUR160G
 T1 = B547B
 T2 = BC639
 T3 = BC640
 T4 = IPP200N15N3 G
 T5 à T13 = MPSA42
 T14, T15 = MPSA92

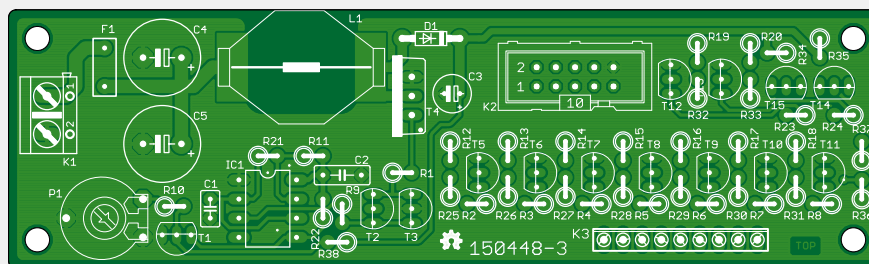
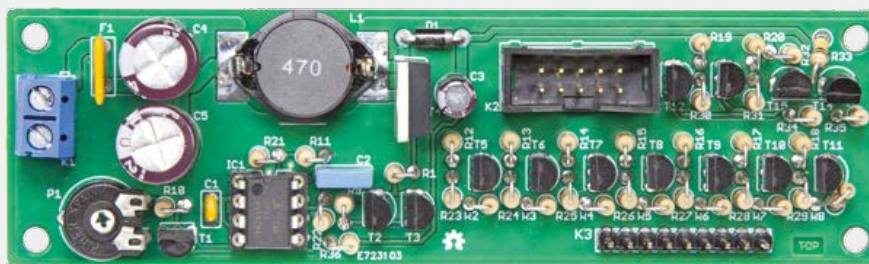


Figure 5. Un circuit imprimé facilitera certainement le montage du circuit.



IC1 = ICM7555

Divers :

F1 = fusible 1,85 A / 30 V (MCC33161)
 K1 = bornier pour circuit imprimé à 2 broches,
 pas de 5 mm

K2 = connecteur mâle 2 x 5 broches,
 pas de 2,54 mm
 K3 = embase mâle à 9 broches
 Circuit imprimé 150448-3
 (disponible sur www.elektor.fr)

Et pour encore faciliter le montage, un circuit a été dessiné pour le pilotage des LED (**fig. 5**). Les LED en filament elles-mêmes sont sur un circuit séparé, dont le schéma est reproduit en **figure 6** et le dessin en **figure 7**. Un circuit imprimé « relais » a aussi été dessiné pour le pilotage de deux afficheurs (**fig. 8**). Les tracés des circuits imprimés sont disponibles gratuitement sur le site du magazine [1].

Logiciel

Il est tout à fait possible de remplacer des afficheurs à sept segments dans un circuit existant par les afficheurs à LED en filament, si nécessaire via des inverseurs logiques. Il est aussi possible d'utiliser un microcontrôleur, une carte Arduino ou un ordinateur monocarte.

On peut ainsi construire une horloge, un thermomètre ou un voltmètre avec Arduino et les exemples de programmes publiés dans l'article sur le *shield* à VFD (*Vacuum Fluorescent Display*) dans le numéro de septembre 2015 [2].

La temporisation du pilotage des afficheurs devra sans doute être adaptée, sinon l'affichage pourrait trembloter ; dans les nouveaux croquis écrits pour cet article [1], la fréquence de rafraîchissement des afficheurs passe ainsi de 250 à 500 Hz.

Assemblage des afficheurs à sept segments

Le plus simple pour monter les LED en filament est d'utiliser des embases femelles ou des supports de circuit intégré. Les supports tulipe peuvent être montés sur un circuit imprimé, ou à travers du Plexiglas® noir (préperçage de 3 mm de diamètre) ; la connexion se fera alors à l'arrière par des fils.

Les fils de connexion des LED en filament doivent être pliés délicatement à 90° et enfoncés dans les supports tulipe. Attention à la polarité, l'anode est marquée par un point rouge ou un petit trou dans le fil de connexion. Pour le point décimal, on utilisera une LED de 3 mm, jaune ou blanche. Il ressort de notre expérience que les LED en filament sont très fragiles, et ne survivent souvent pas à leur voyage. Une alternative est d'acheter des lampes à filament de LED et de les démonter précautionneusement ; il faut surtout être très attentionné lors de manipulation des LED en filament.

Pour augmenter la visibilité de l'afficheur, on pourra enfermer les LED en filament sous un écran, par exemple en Plexiglas® noir, et ajouter un filtre optique. Pour ce dernier, nous avons obtenu de bons résultats avec du Plexiglas® Umbra de 3 mm de couleur brune, qui donne à l'ensemble

un joli look rétro ; ce Plexiglas® existe en plus de 40 coloris standard, on pourra donc expérimenter à souhait.

Si vous utilisez une découpeuse au laser, un schéma de conception assistée par ordinateur (CAO) est disponible sur le site du magazine [1], avec un exemple pour un afficheur d'environ 67 x 100 mm².

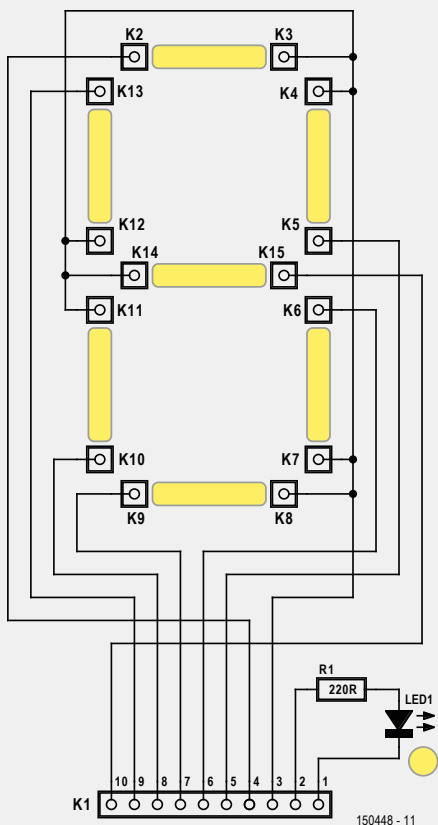
Un kit avec du Plexiglas® Umbra et un circuit imprimé noir pour les afficheurs est aussi disponible chez Elektor.

Montage et test du circuit

Le montage du circuit est franchement simple. Commencez avec les composants les moins hauts, comme la diode, la bobine, le potentiomètre ajustable, le support de circuit intégré et les condensateurs céramiques. Poursuivez avec les résistances et les transistors en boîtier TO-92. Viennent ensuite les connecteurs, le condensateur de sortie de 4,7 μ F et le fusible. Et pour finir, les composants un peu plus volumineux : les condensateurs d'entrée et le MOSFET.

Placez ensuite IC1 sur son support et tournez P1 à fond vers la gauche. Reliez le circuit à une alimentation de 12 V (qui débite au moins 1 A) et mesurez la tension de sortie entre la cathode de D1 et la masse, vous devriez obtenir de l'ordre de 35 V. Augmentez la tension avec P1

Composants de l'afficheur à sept segments



Résistance :

$R1 = 220 \Omega$

Semi-conducteur :

LED1 = LED 5 mm

Divers :

K1 = embase femelle à 10 contacts,
pas de 2,54 mm
Circuit imprimé 150448-1
(disponible sur www.elektor.fr)

Figure 6. Les sept LED en filament sont pilotées via un connecteur à dix broches...

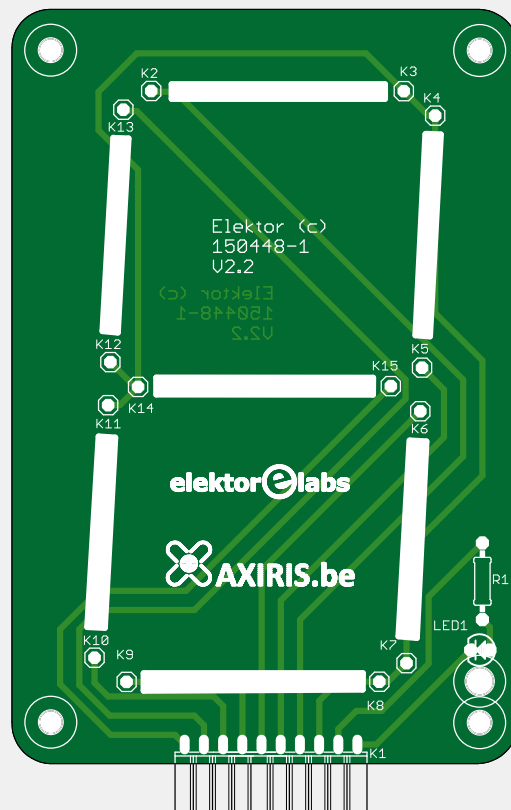


Figure 7. ... et forment sur le circuit imprimé un afficheur à sept segments.

Composants de carte « relais »

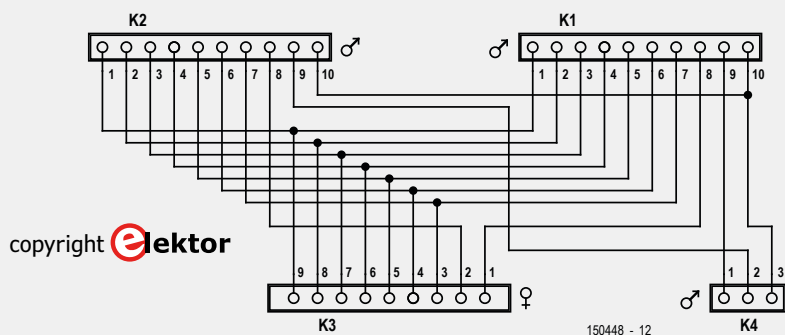
Divers :

K1, K2 = embase mâle à 10 broches,
pas de 2,54 mm
K3 = embase femelle à 9 contacts,
pas de 2,54 mm

K4 = embase mâle à 3 broches,
pas de 2,54 mm
Circuit imprimé 150448-2
(disponible sur www.elektor.fr)



Figure 8. Avec la carte « relais », on peut piloter simultanément deux afficheurs à partir d'un seul circuit.



jusque 75-80 V, c'est prêt à l'emploi !
Après avoir connecté les afficheurs, vous pourrez ajuster à nouveau P1 en fonction de la luminosité souhaitée.

Le courant de fuite de T14 et T15 (MPSA92) à l'état bloqué est parfois suffisant pour illuminer faiblement les LED en filament qui devraient être éteintes. Cela dépend de l'application et peut certes être gênant ; si tel est le cas, on peut monter une résistance de 100 kΩ (R36, R37) entre l'anode des afficheurs et la masse.

Et pour terminer, un avertissement : la tension de sortie n'est pas sans danger ! Évitez donc de toucher les parties sous tension lors du montage et des tests, et prévoyez un boîtier isolant pour le circuit. ◀

(150448 - version française : Jean-Louis Mehren)

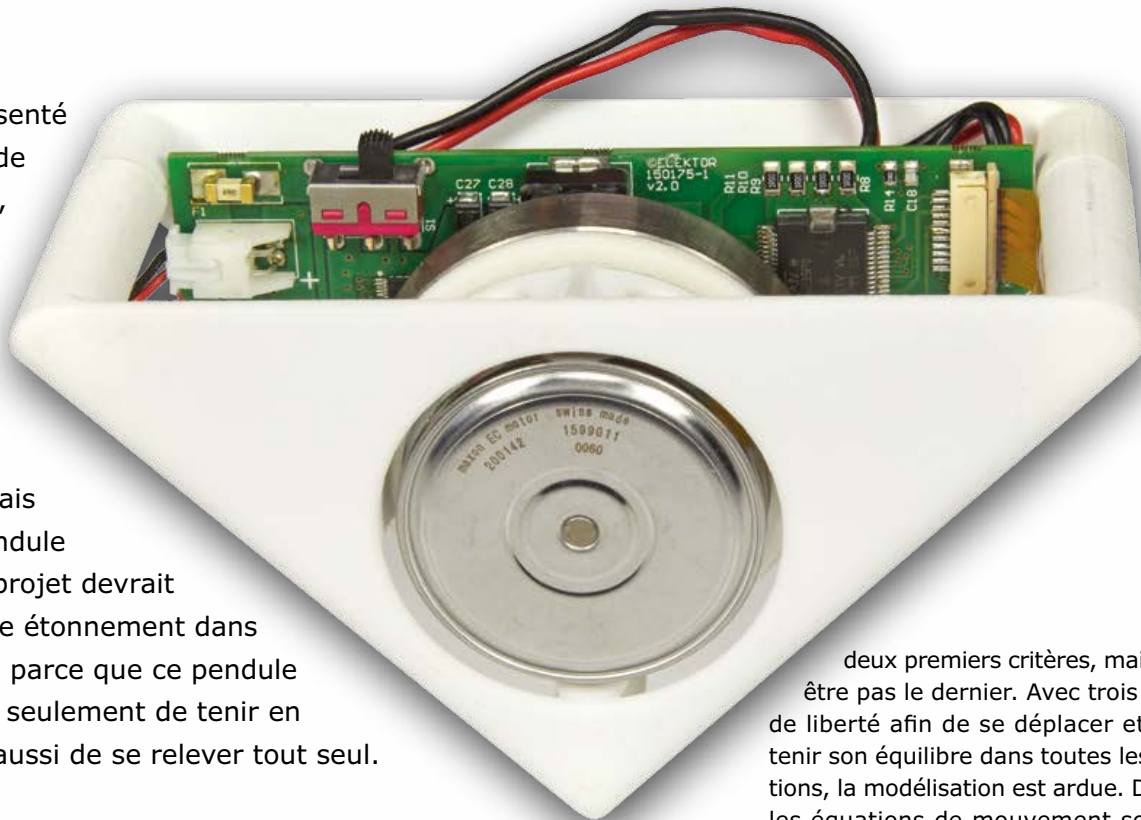
Liens

- [1] Informations supplémentaires sur ce projet : www.elektormagazine.fr/150448
- [2] Article sur le « shield à VFD » : www.elektormagazine.fr/150064

i-Pendulum

1^{ère} partie - modélisation, lois de contrôle, filtre de Kalman

Lorsque j'ai présenté mon prototype de pendule inversé, un collègue m'a demandé : « et donc tu t'es réveillé un matin en te disant : aujourd'hui je vais fabriquer un pendule inversé ! ». Ce projet devrait susciter le même étonnement dans votre entourage parce que ce pendule est capable non seulement de tenir en équilibre, mais aussi de se relever tout seul.



deux premiers critères, mais peut-être pas le dernier. Avec trois degrés de liberté afin de se déplacer et maintenir son équilibre dans toutes les directions, la modélisation est ardue. De plus, les équations de mouvement sont loin d'être linéaires, d'où des lois de contrôle complexes. Pour simplifier, je suis passé du cube (3D) au carré (1D) avec un peu d'épaisseur bien sûr, capable de maintenir son équilibre autour d'un seul axe. Comme ce carré n'avait pas besoin de « rouler », je l'ai coupé en deux, d'où cette forme triangulaire.

Jean-Sébastien Gonsette (Belgique)

La particularité du pendule inversé est que son centre de gravité est au-dessus de son axe de rotation ou de pivotement. C'est donc un pendule avec la tête en bas qui ne demande qu'à tomber, comme le corps humain. Toutefois le cerveau travaille avec les yeux et oreilles pour envoyer des impulsions nerveuses d'ajustement vers les muscles lorsque notre buste se déplace par rapport à nos chevilles (point de pivot). Les pendules inversés reposent sur ce principe. La vidéo en [1] montre mon pendule en action. Sur l'internet, vous trouverez dix ans de vidéo sur ce sujet universitaire récurrent. Il permet aux étudiants de tester toutes sortes de techniques de régulation, de se frotter à un sujet pluridisciplinaire, et de déclencher à tous les coups des *ouah* !. Il y en a pour tous les goûts : à deux roues façon Segway, barre (simple, double ou

triple) à la verticale sur un chariot en mouvement, quadricoptère coiffé d'une tige sur laquelle trône un verre. Aucun de ces modèles ne correspond à mes critères de conception, à savoir :

- Dimension réduite (env. 10 cm) et léger pour pouvoir parader sur mon bureau.
- Complètement autonome, c.-à-d. que le pendule transporte sa batterie.
- Simple, afin de démythifier le principe général sans ajouter de complexité superflue.

J'ai trouvé un projet récent des scientifiques de l'École polytechnique fédérale de Zurich (ETH), spécialisés dans les systèmes dynamiques et le contrôle : *Cubli*. C'est un cube qui tient en équilibre sur l'une de ses arêtes ou l'un de ses coins, et saute pour passer d'une position à une autre (vidéo en [2]). *Cubli* respecte mes

Anatomie d'un pendule inversé

La **figure 1** est une vue éclatée des différents éléments du pendule ; son nombre de pièces est restreint. Son squelette est constitué de deux demi-coques en plastique, fabriquées avec une imprimante 3D (nous y reviendrons dans la 2^e partie de cet article). La première demi-coque sert de support au circuit imprimé, tandis que la deuxième contient le moteur et sa roue d'inertie. Le cerveau du pendule est un microcontrôleur PIC qui commande et alimente le moteur. La batterie de type LiPo (courant en modélisme) est glissée directement dans son support derrière le circuit imprimé. Le moteur, unique organe de commande du pendule, est logé dans

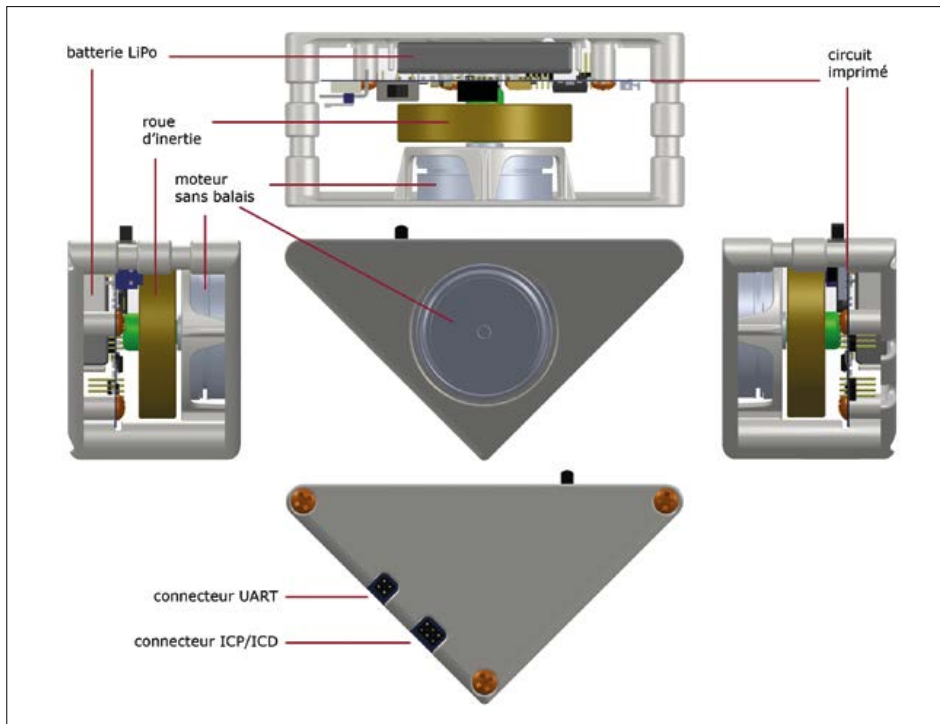


Figure 1. Entrailles du pendule inversé.

une niche prévue à cet effet dans la coque en plastique. Sur ce moteur vient se greffer une roue d'inertie en couplage direct. Cette roue d'inertie est constituée d'un axe en plastique, également imprimé en 3D, sur lequel est serti un anneau métallique dont la masse apporte l'inertie désirée.

Comment ça marche ?

Comme Cubli, mon pendule repose sur le principe physique de la *conservation du moment angulaire* à l'aide d'une roue

d'inertie pour assurer son équilibre. Même si la rotation de cette roue fait penser à l'effet gyroscopique d'une toupie, ce dernier n'est pas en cause ici. D'abord, passons par quelques rappels de physique, en particulier les Lois du mouvement de Newton :

1. Dans un référentiel galiléen (c.-à-d. soumis à aucune force extérieure), le vecteur vitesse du centre d'inertie d'un système est constant si et seulement si la somme des vecteurs

force qui s'exercent sur le système est nulle. En clair : la quantité de mouvement d'un corps se conserve si aucune force n'est appliquée dessus.

2. Dans un référentiel galiléen, la somme des forces s'exerçant sur un système est égale à la dérivée de la quantité de mouvement. Il s'agit de la célèbre formule :

$$\sum \vec{f}_i = \frac{d\vec{p}}{dt}$$

ou plus vulgairement $f = m \cdot a$. En clair : la quantité de mouvement d'un corps ($p=mv$) est modifiée au prorata de la résultante des forces qu'il subit.

3. Tout corps A exerçant une force sur un corps B subit une force d'intensité égale dans la même direction, mais de sens opposé : c'est le principe d'action-réaction. Ce qu'un corps gagne, l'autre le perd ; c'est un transfert d'argent d'un compte vers un autre, ce n'est jamais perdu pour tout le monde.

Ces principes généraux s'appliquent aux particules. Pour tout un système, on s'intéresse plutôt à son centre de gravité ainsi qu'à son orientation angulaire autour de ce point. Cette décomposition revient à scinder les forces en deux catégories :

- celles qui modifient la *quantité de mouvement linéaire* du centre de gravité du système,
- celles qui modifient la *quantité de mouvement angulaire* du système autour de son centre de gravité.

Mathématiquement, il est plus facile de travailler sur la quantité de mouvement angulaire avec le *couple*. Il dépend de force appliquée pour faire tourner le système autour d'un point de pivot et de la distance entre le point de pivot et le point d'application de la force. Le **tableau 1** compare mouvement linéaire et mouvement angulaire.

Le fonctionnement du pendule est presque identique à celui des volants d'inertie embarqués dans les engins spatiaux. Un satellite en orbite autour de la terre doit corriger son orientation afin de réaligner ses antennes vers un point précis de la terre. Avant de manœuvrer, son moment angulaire est nul (il ne tourne pas sur lui-même). Lorsqu'il applique un couple moteur sur l'un de ses volants d'inertie, celui-ci accélère sa rotation dans

Tableau 1. Comparaison entre mouvement linéaire et mouvement angulaire.			
déplacement du centre de gravité d'un système		rotation d'un système autour de son centre de gravité	
Masse en kg	m	Inertie en kg.m ²	I
Vitesse en m/s	\vec{v}	Vitesse angulaire en rad/s	$\vec{\omega}$
Accélération en m/s ²	$\vec{a} = \frac{d\vec{v}}{dt}$	Accélération angulaire en rad/s ²	$\vec{\alpha} = \frac{d\vec{\omega}}{dt}$
Force en N	\vec{f}	Couple en N.m	$\vec{\tau}$
Quantité de mouvement linéaire	$\vec{p} = m \cdot \vec{v}$	Quantité de mouvement angulaire	$\vec{L} = I \cdot \vec{\omega}$
Énergie cinétique	$K = \frac{m \cdot v^2}{2}$	Énergie cinétique angulaire	$K = \frac{I \cdot \omega^2}{2}$
Second principe	$\sum \vec{f} = m \cdot \vec{a}$	Second principe	$\sum \vec{\tau} = I \cdot \vec{\alpha}$
Troisième principe	$\vec{f}_{A \rightarrow B} = -\vec{f}_{B \rightarrow A}$	Troisième principe	$\vec{\tau}_{A \rightarrow B} = -\vec{\tau}_{B \rightarrow A}$

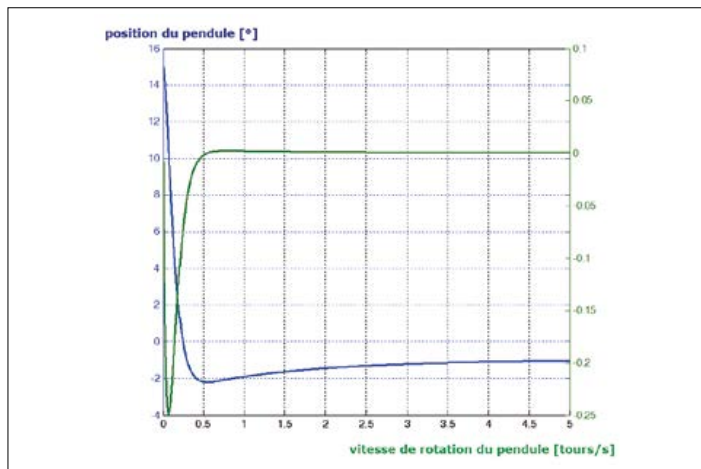


Figure 2. Pour le plaisir des yeux, la simulation de la stabilisation du pendule.

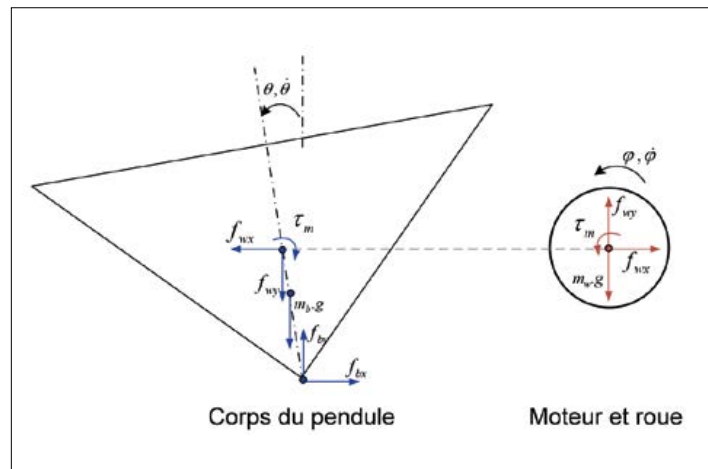


Figure 3. Dynamique du pendule et de sa roue d'inertie.

un sens, mais dans le même temps la reste du satellite subit un couple de même valeur, mais de sens opposé (3^e principe). Résultat, le satellite tourne dans le sens contraire. Sans aucun point d'appui extérieur dans l'espace, un satellite peut donc utiliser un volant d'inertie interne pour modifier son moment angulaire. Si nous regardons le satellite complet (volant d'inertie inclus), l'absence de couples externes implique aussi que le moment angulaire total ne change pas, c.-à-d. qu'il reste nul (2^e principe). C'est bien le cas : la somme « moment angulaire du volant d'inertie (« + ») + moment angulaire du reste du satellite (« - ») » ne varie jamais si les seuls couples en jeu sont des couples entre des éléments internes. Pour revenir à l'analogie sur les comptes en banque, sans apport d'argent externe, vous aurez beau transférer de l'argent entre votre compte courant et vos comptes épargnes dans tous les sens, vous ne serez pas plus riche.

Pour le pendule, la situation est pratiquement identique, toutefois le couple extérieur n'est pas nul à cause de l'action de la gravité. Si vous essayez de maintenir le pendule en équilibre sur sa pointe, la gravité transformera le moindre petit déséquilibre en une accélération angulaire non nulle, jusqu'à l'écrasement au sol. Pour la contrer, il faudrait amener le centre de gravité du pendule exactement au-dessus de son point de pivot. Comme ce point est petit, c'est impossible en pratique. Par contre il est tout à fait possible d'amener le centre de gravité du pendule au-dessus du pivot, *en moyenne*. Le pendule est donc tiré vers la gauche quand il dévie vers la droite, et inversement, cela

permet de le maintenir en équilibre et de contrer l'action de la gravité. C'est là que le moteur et sa roue d'inertie entrent en jeu. Comme pour le satellite, le corps du pendule accélère dans un sens quand la roue d'inertie part dans l'autre sens. À chaque instant, il faut détecter vers où tombe le pendule et en déduire la valeur du couple moteur à appliquer pour compenser le déséquilibre. Si la commande est bien réalisée, le pendule décrit de petites oscillations, tantôt vers la gauche, tantôt vers la droite, mais ne tombe pas.

Et le saut ?

La capacité du pendule à sauter (passer de la position couchée à la position en équilibre) est surprenante, mais simple à expliquer. Il suffit d'accélérer la roue d'inertie dans la direction vers laquelle on veut sauter, jusqu'à ce qu'elle atteigne une vitesse adéquate. À ce moment-là, il faut bloquer la roue pour que tout le pendule décolle. Revenons à Newton :

- Quand le pendule est au sol, roue à l'arrêt, son moment angulaire est nul.
- Il accélère sa roue d'inertie et gagne ainsi en moment angulaire en s'appuyant sur ce sol. Sans support, cette opération est impossible ; cela ne fonctionne pas dans l'espace.
- Lorsque la vitesse de rotation est suffisante, bloquer la roue revient à faire passer le moment angulaire à zéro. Comme le moment angulaire d'un système isolé se conserve et ne peut pas disparaître, il est transféré quelque part : au pendule lui-même ! Le moment angulaire total ne varie pas.

Modélisation de la dynamique du pendule

Pour réguler un système physique tel que ce pendule, il faut connaître les lois qui régissent ses mouvements en fonction des commandes appliquées. Il faut donc modéliser la dynamique du pendule, ce qui permet d'étudier plus finement le problème à résoudre, puis de dimensionner correctement les différents éléments : taille du pendule, masse de la batterie, couple du moteur, puissance du circuit électronique... Comme je pars de rien, je choisis de fixer arbitrairement certains paramètres et de vérifier par la simulation que mon modèle tient la route, avant de construire le prototype (voir **figure 2**, résultat d'une régulation testée sur PC). Le modèle limite le risque d'échec, mais il est aussi indispensable pour écrire les algorithmes embarqués dans le pendule (commande du moteur, analyse des données reçues des capteurs et estimation de l'état du pendule).

Pour obtenir les équations de la dynamique du pendule, il faut déterminer le nombre de degrés de liberté du système, dresser la liste exhaustive des forces et couples qui agissent sur chacun des éléments, décrire les effets dynamiques de chacun d'eux, puis secouer le tout pour ne garder que les paramètres intéressants. Le pendule a deux degrés de liberté :

- le corps qui pivote autour de son point de contact avec le sol, avec un angle θ
- la rotation du moteur et sa roue d'inertie, avec un angle φ .

Le but du jeu est de trouver comment évoluent les dérivées secondes de ces deux paramètres, c.-à-d. leur accéléra-

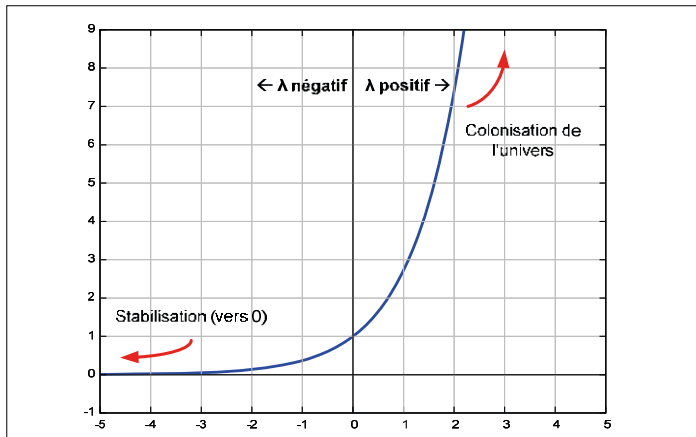


Figure 4. La valeur propre, maîtresse de l'univers.

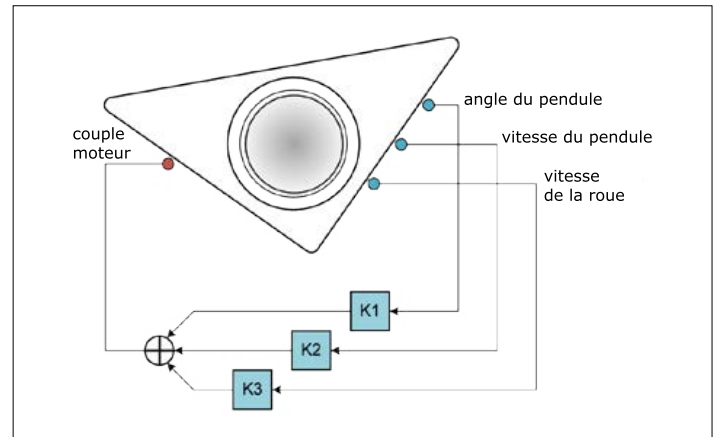


Figure 5. Commande par retour d'état.

tion, en fonction du couple délivré par le moteur. La **figure 3** reprend la liste de tous les paramètres qui agissent sur le corps du pendule et la roue :

- Roue : le couple moteur lui-même τ_m , le couple de frottement entre le rotor et le stator τ_r , et aussi les deux forces qui maintiennent le rotor en place f_{wx} et f_{wy} .
- Pendule : les forces et les couples qui agissent sur la roue, mais en sens opposé ; les deux forces entre le pendule et le sol f_{bx} et f_{by} qui maintiennent le pivot du pendule en place.

La gravité g agit à la fois sur la masse du pendule m_b et sur celle de la roue m_w . Je vous épargne les calculs ; on obtient les deux équations dynamiques suivantes :

$$\ddot{\theta} = \frac{M \cdot g \cdot \sin \theta - \tau_m + \tau_f}{I}$$

$$\ddot{\varphi} = \frac{(I + I_w) \cdot (\tau_m - \tau_f)}{I \cdot I_w} - \frac{M \cdot g \cdot \sin \theta}{I}$$

avec $M = m_b \cdot l_b + m_w \cdot l_w$
et $I = I_b + m_b \cdot l_b^2 + m_w \cdot l_w^2$

Les termes m_b , I_b , m_w et I_w sont respectivement la masse et l'inertie du corps du pendule et celles de la roue d'inertie ; l_b et l_w sont la hauteur du centre de gravité du corps du pendule et la hauteur de la roue d'inertie par rapport au sol. Ces équations servent non seulement à simuler le comportement du pendule, mais aussi à établir les lois de contrôle du moteur.

Stabilité du pendule

Pour étudier mathématiquement l'instabilité du pendule, je propose une analogie avec la biologie. Une boîte de Petri permet de cultiver des micro-organismes

afin d'étudier leur évolution. Voici mes hypothèses :

- y : nombre d'organismes dans la boîte.
- n : taux de naissance des organismes ; d : leur taux de décès.
- Si ces deux taux sont proportionnels à la taille de cette population, alors $y' = n \cdot y - d \cdot y$.

Si ces taux sont constants, l'intégration de cette équation est aisée, le résultat est la courbe d'évolution de la population. Si la population de départ est y_0 , on obtient :

$$y(t) = y_0 \cdot e^{(n-d) \cdot t} = y_0 \cdot e^{\lambda \cdot t}$$

La fonction exponentielle montre que, selon le signe de $n - d$, l'évolution de la population prendra une allure radicalement différente. Si le taux de décès est supérieur au taux de naissance, l'exposant est négatif : la population s'éteint progressivement. Dans l'autre cas, la population croît sans fin et colonise l'univers (du moins si les équations ont encore du sens à cette échelle). L'exposant λ de cette équation est appelé *valeur propre du système*. C'est le signe de cette valeur qui détermine si le système est stable (valeur « - ») ou instable (valeur « + ») (**fig. 4**). Modifier le pH dans la boîte permet de contrôler le taux de croissance/décroissance. Si ce contrôle est proportionnel à la quantité d'individus, les équations de l'évolution deviennent :

$$y' = n \cdot y - d \cdot y + k \cdot y ;$$

$$y(t) = y_0 \cdot e^{(n-d+k) \cdot t}$$

C'est là que réside toute l'astuce de la manœuvre. Comme le paramètre k peut être choisi librement (dans les limites

physiques du système), il est possible de forcer une valeur propre positive à devenir négative, ce qui permet au système de retrouver sa stabilité.

Les équations qui régissent le comportement du pendule sont un peu plus compliquées que dans cet exemple. Les différents états du système, comme l'angle et la vitesse du pendule ou la vitesse du moteur, doivent être transformés en vecteurs, tandis que les multiples équations différentielles doivent être transposées dans des matrices. Au final, on calcule les valeurs propres d'une matrice, il est alors moins évident de voir pourquoi le système est stable ou pas. Si l'une des valeurs propres est positive, alors quelque chose de la forme $e^{\lambda \cdot t}$ apparaît quelque part dès que vous effectuez une intégration sur votre système, et plus rien ne va plus.

Régulation

La régulation de type PID est plutôt adaptée à système à une entrée de commande et une variable de sortie. Or, notre pendule a deux degrés de liberté : l'angle de son corps et la vitesse de rotation de sa roue d'inertie. Il faut contrôler ces deux grandeurs à la fois avec une seule entrée de commande, le couple moteur. Le point délicat ici est qu'un moteur ne peut délivrer un couple que dans une plage de vitesse de rotation donnée. Si je ne régule pas la vitesse de rotation, elle dérivera jusqu'à atteindre des valeurs inacceptables et le pendule ne sera plus contrôlable. Exit la régulation PID !

Je choisis donc la **commande par retour d'état** qui repose sur la valeur des états internes du système. Concrètement, le bouclage s'effectue en introduisant un coefficient de gain réel pour chaque possi-

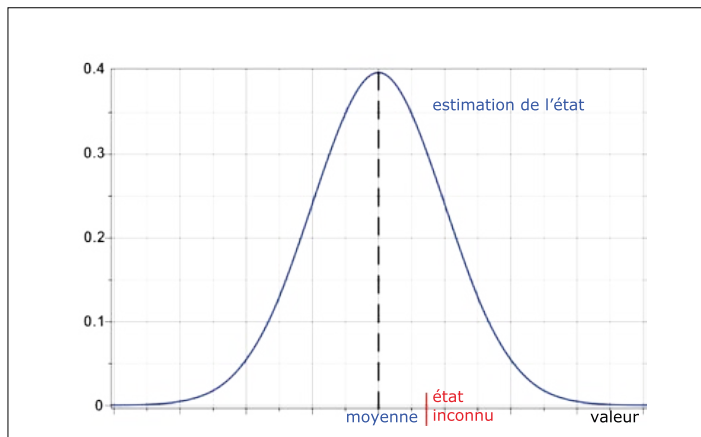


Figure 6. Fonction de densité de probabilité de type gaussien.

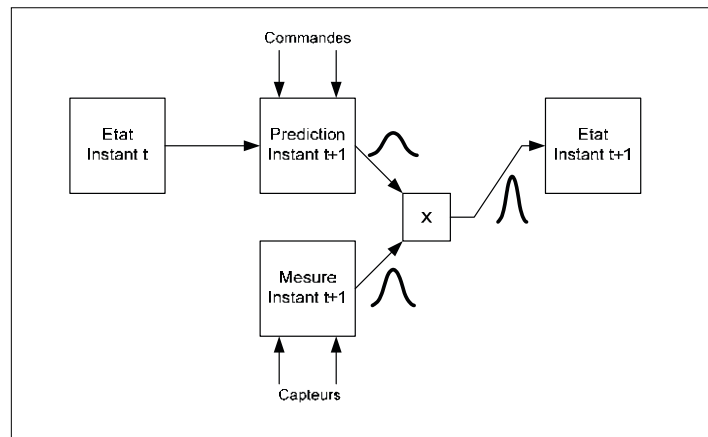


Figure 7. Principe du filtre de Kalman.

bilité de retour d'une sortie sur une entrée. Dans le cas du pendule, les états contrôlés sont l'angle du pendule, la vitesse de rotation du pendule et la vitesse de rotation de la roue d'inertie, donc trois sorties. La boucle de rétroaction repose donc sur la somme de trois gains appliquée à l'unique entrée de commande (fig. 5). Il est possible de calculer des gains tels que les valeurs propres du pendule soient toutes négatives, d'où un pendule stable.

Différentes approches sont possibles, mais il est assez commode d'appliquer la théorie du « régulateur linéaire quadratique » (*Linear Quadratic Regulator*, LQR). Elle fournit un algorithme de calcul automatique de ces gains, en fonction des équations du système et d'une série de coefficients de pondération. Ces poids définis par le concepteur correspondent à l'importance accordée aux différents états du système et à la commande lors de la régulation. Plus un poids est élevé, plus l'algorithme optimisera l'état ou la commande associé ; il ou elle convergera plus rapidement vers sa valeur de consigne. Le plus important ici est de rétablir l'équilibre avec un minimum de couple, que ce soit en une ou trois secondes. En effet, le couple du moteur est limité : le but est donc de tirer le maximum du moteur tout en compensant le plus grand déséquilibre possible. C'est pourquoi la commande a un poids infini et un poids égal à zéro aux états. Je m'arrête là sur la théorie du contrôle. Le sujet est trop vaste pour être traité entièrement ici. Allez voir la série de tutoriels de l'université du Michigan, consacrée au contrôle des systèmes avec Matlab, dont un sur la régulation d'un pendule inversé [3].

État du pendule après passage par le filtre de Kalman

Comment connaître les états du pendule à chaque instant : angle du pendule, vitesse du pendule et vitesse du moteur ? Soit prédire ces états, soit les mesurer. Si l'état de départ est connu, je peux prédire l'évolution en fonction de la consigne de couple appliquée au moteur. J'intègre les équations différentielles du modèle pour savoir où le pendule sera un instant plus tard, et je recommence pour l'instant suivant... C'est comme essayer de conduire une voiture les yeux fermés, en se basant sur la représentation mentale de la route et le comportement de la voiture. Cela marchera... pendant quelques secondes ! À cause d'erreurs de modélisation, la prédiction s'écarte de la réalité. La mesure semble donc une méthode plus fiable. Toutefois tous les capteurs présentent une certaine imprécision et sont sujets aux bruits de mesure. En clair, deux mesures successives d'un état identique ne donneront pas deux fois le même résultat. De plus, il n'est pas toujours possible de tout mesurer. Selon le processus, certains états seront déduits de manière indirecte. J'ai d'une part une méthode précise à court terme, mais dont l'erreur augmente rapidement, et d'autre part une méthode avec une plage d'erreur fixe, mais non négligeable. L'idéal est d'utiliser celle qui offre l'erreur la plus faible à un instant donné. Prenons un exemple : je mesure une résistance avec un voltmètre haut de gamme (68 Ω) puis avec un modèle à quelques euros (70 Ω). La première mesure semble la plus fiable. En fait il faut effectuer la moyenne pondérée des deux mesures avec des poids calculés au prorata de la précision de chacun des instru-

ments. Plus l'instrument est précis, plus le poids de sa mesure est élevé. J'applique ce principe au pendule. L'état du pendule sera déterminé en fonction du modèle et des mesures des capteurs, pondérés en fonction de leur validité respective.

Ce principe constitue le cœur du **filtre de Kalman** : un filtre sur les états qui effectue une moyenne pondérée des différentes mesures et prédictions d'un processus dynamique. Ce filtre ôte autant que possible le bruit qui entache les mesures et les prédictions afin d'obtenir une estimation optimale de l'état du processus. Il se décompose en deux étapes :

1. Utiliser le modèle du système, ainsi que les commandes produites à un instant t , pour prédire l'état du système à $t + dt$. La prédiction estime aussi comment l'incertitude courante sur l'état du système évolue en fonction de l'incertitude sur le modèle lui-même. Toutes les incertitudes d'un filtre de Kalman font l'objet d'approximations de Gauss. Pour un système qui ne contient qu'un seul état, la classique courbe de Gauss est représentée à la **figure 6**. Dans les autres cas, cela devient une courbe multi-gaussienne à plusieurs dimensions.
2. Mesurer l'état du système à $t + dt$. Ces mesures ont aussi leur lot d'incertitudes, une fois encore modélisées par des courbes de Gauss. Toutes les mesures sont pondérées par leur incertitude respective, comme dans l'exemple du voltmètre.

Ensuite, le résultat des mesures est fusionné avec la prédiction en tenant

compte là encore des incertitudes respectives des deux estimations. La **figure 7** donne une vue synthétique du processus. Si le sujet vous intéresse, lisez le livre référencé en [4].

Ici, un gyroscope et un accéléromètre mesurent l'angle du pendule avec le sol. Le gyroscope ne s'intéresse qu'à un seul axe orienté dans la direction Z, perpendiculaire au circuit imprimé. L'accéléro-

mètre comporte trois axes, X, Y et Z, mais seuls les deux premiers sont utilisés dans le filtre de Kalman. Ce dernier est plutôt rudimentaire, mais fournit un résultat amplement suffisant.

Comment sont fusionnées les informations issues des deux capteurs ?

- Le gyroscope fournit la vitesse de rotation du pendule. Grâce à une intégration sur un pas de temps dt , on peut

prédire le futur angle du pendule.

- L'accéléromètre mesure principalement comment l'accélération due à la gravité est projetée sur les axes X et Y. Par inversion trigonométrique, il donne donc directement une mesure de l'angle que fait le pendule avec le sol.

La prédiction et la mesure sont combinées au prorata des erreurs de chacun des cap-

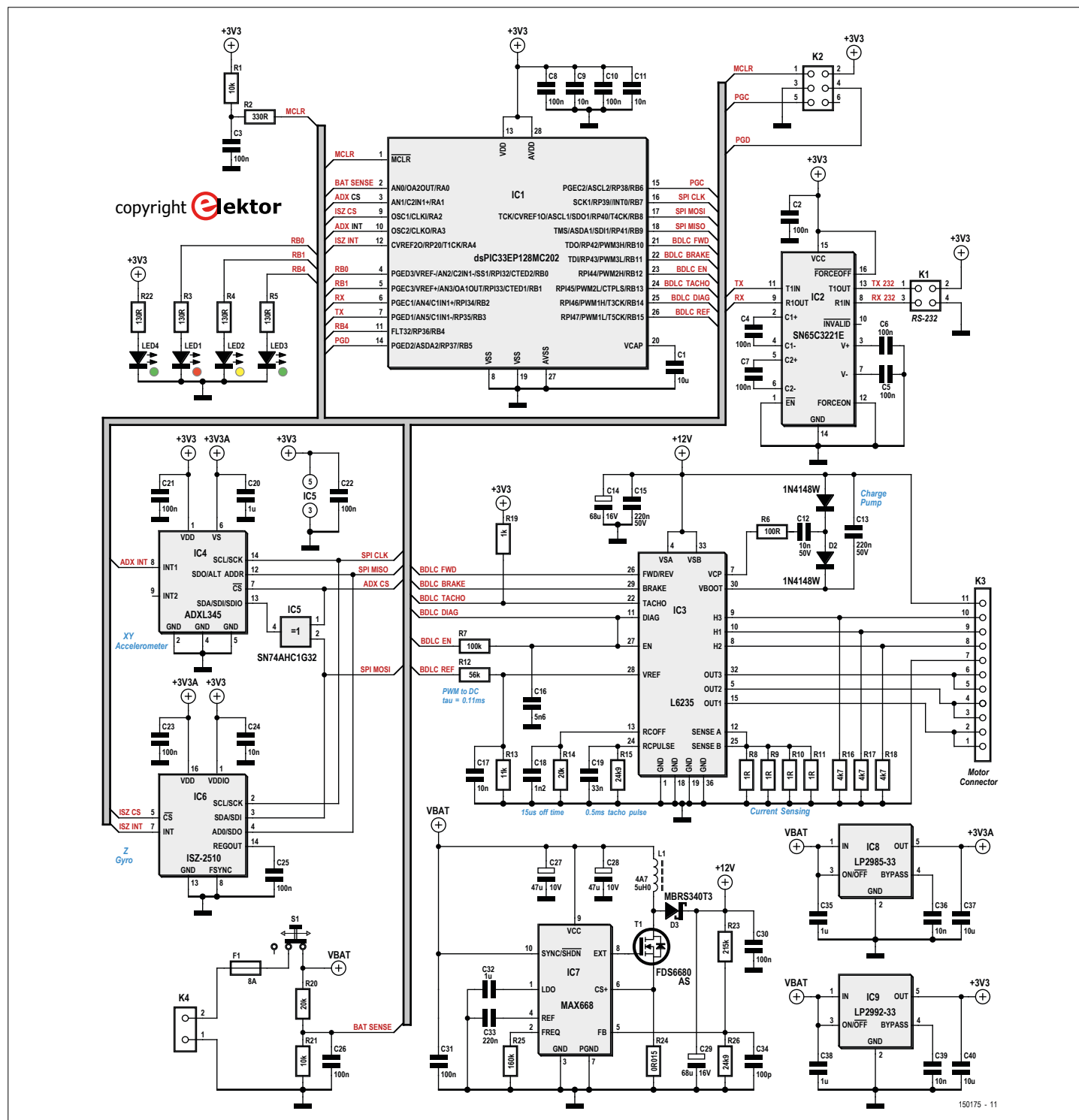


Figure 8. Schéma complet du circuit du pendule inversé.

teurs. L'estimation de la vitesse de la roue suit le même principe. Le couple demandé au moteur permet de réaliser une prédiction sur l'accélération ou la décélération de la roue, et donc sur sa vitesse après un pas de temps dt . Cette prédiction est ensuite associée à la mesure de vitesse effectuée à chaque fois qu'une impulsion tachymétrique est réceptionnée.

Électronique du pendule

Assez de théorie, passons à la pratique. L'électronique embarquée du pendule est très classique et doit assurer quatre grandes fonctions réunies sur le schéma de la **figure 8** :

1. Convertir la tension issue de la batterie LiPo pour alimenter le moteur sans balais (*brushless*)
2. Délivrer les signaux de commande du moteur
3. Mesurer les paramètres dynamiques du pendule via un accéléromètre et un gyroscope
4. Mettre en œuvre la loi de commande du moteur (μC) en fonction des valeurs retournées par les capteurs.

Alimentation

La tension délivrée par la batterie LiPo varie entre 7 V et un peu plus de 8 V, selon son état de charge. Le moteur Maxon requiert une tension nominale de 12 V. L'alimentation à découpage articulée autour d'un circuit MAX668 réalise l'adaptation de puissance nécessaire (régulateur *boost*). Le moteur absorbe un courant de 2 A pour un couple de 55 mNm, l'alimentation fournira au moins le double d'intensité. Lorsque la broche 8 du MAX668 est à l'état haut, le transistor de puissance T1 conduit et l'intensité du courant à travers l'inductance de puissance L1 grimpe. La diode D3 empêche que le condensateur de sortie C29 ne se décharge par la même occasion. Lorsque la broche 8 passe à l'état bas, le transistor est bloqué, ce qui oblige l'énergie magnétique accumulée dans L1 à se déverser dans C29 via la diode. Le MAX668 régule les cycles de ce processus à une fréquence configurable, fixée ici à 300 kHz avec R25. Cette régulation se base sur la tension de sortie prélevée aux bornes de R23 et R26, ainsi que sur le courant circulant dans l'inductance et mesuré à travers R24. La tension de sortie +12 V contient inmanquablement des ondulations à hautes fréquences dues aux charges et décharges répétées de C29.

Cette ondulation est toutefois dominée par la valeur de sa résistance série équivalente (ESR) et les courants à hautes fréquences qui la traversent. C'est pourquoi il faut apporter un soin tout particulier au choix de ce condensateur.

Moteur

J'ai choisi un moteur sans balais de la série EC Flat de Maxon. Par rapport à un moteur CC traditionnel, l'absence de balais évite une usure mécanique prématurée et les bruits électromagnétiques associés. En outre ce modèle permet une meilleure conversion de puissance. Toutefois, son câblage et sa logique de commande sont plus complexes. On ne peut pas se contenter de raccorder ce moteur à une source de tension : il faut l'alimenter via trois phases, en permutant les flux magnétiques de celles-ci à un rythme de rotation précis du rotor. Un système électronique est donc nécessaire pour superviser la séquence d'alimentation des bobines. C'est le rôle du circuit intégré L6235 (IC3), qui contient toute la logique pour piloter ce moteur, y compris les transistors de puissance. Il est doté d'un ensemble d'E/S simples :

- **Fwd/Rev** : sens de rotation souhaité
- **Brake** : active la fonction de freinage électromagnétique (c'est elle qui permet au pendule de sauter)
- **En** : active la rotation du moteur
- **Vref** : entrée analogique pour choisir l'intensité limite du courant délivré au moteur
- **Diag** : signale le déclenchement d'une protection interne
- **Tacho** : délivre les impulsions tachymétriques.

La conversion d'un signal MLI en tension analogique grâce au filtre passe-bas constitué par R12, R13 et C17 permet de définir la valeur limite du couple. La connexion aux trois phases du moteur est effectuée sur les sorties **Out1** à **Out3**,

tandis que les signaux de position issus des capteurs à effet Hall du moteur sont décodés sur les entrées **H1** à **H3**. Le circuit contrôle le courant absorbé par le moteur en le mesurant au travers des résistances R8 à R11 et en le comparant avec la consigne fixée sur **Vref**. Ce circuit permet bien de contrôler le couple délivré puisque le couple de ce moteur est proportionnel au courant absorbé.

Capteurs

Le pendule ne peut rester en l'équilibre que s'il mesure sa position dans l'espace et la corrige en conséquence. Des capteurs lui permettent de mesurer son angle par rapport au sol : gyroscope à un axe (ISZ-2510, IC6) et accéléromètre à trois axes (ADXL345, IC4). Ces deux capteurs analogiques sont chacun logés sur un circuit numérique alimenté séparément. Ainsi la mesure analogique est le moins possible polluée par les bruits numériques du reste du circuit. Les deux circuits dialoguent avec le μC via un bus SPI.

Microcontrôleur

Toute l'intelligence du pendule est embarquée dans un microcontrôleur dsPIC à 16 bits. Par rapport aux PIC à 32 bits, la famille à 16 bits est idéale pour un tel projet où la réactivité prime sur la puissance de calcul. De plus, l'espace mémoire en RAM et ROM est assez généreux et permet de développer des applications plus complexes que sur un μC à 8 bits. Enfin, les dsPIC bénéficient également de sorties MLI, utilisées ici pour commander le moteur. La programmation du μC n'a rien de particulier : interface ICD/ICP reliée au connecteur K2. L'une des UART sert à réaliser un port série via le circuit SN65C3221 (IC2).

Le prochain article est consacré à des tâches plus concrètes comme le câblage du circuit, la construction mécanique et le calibrage des capteurs. ◀

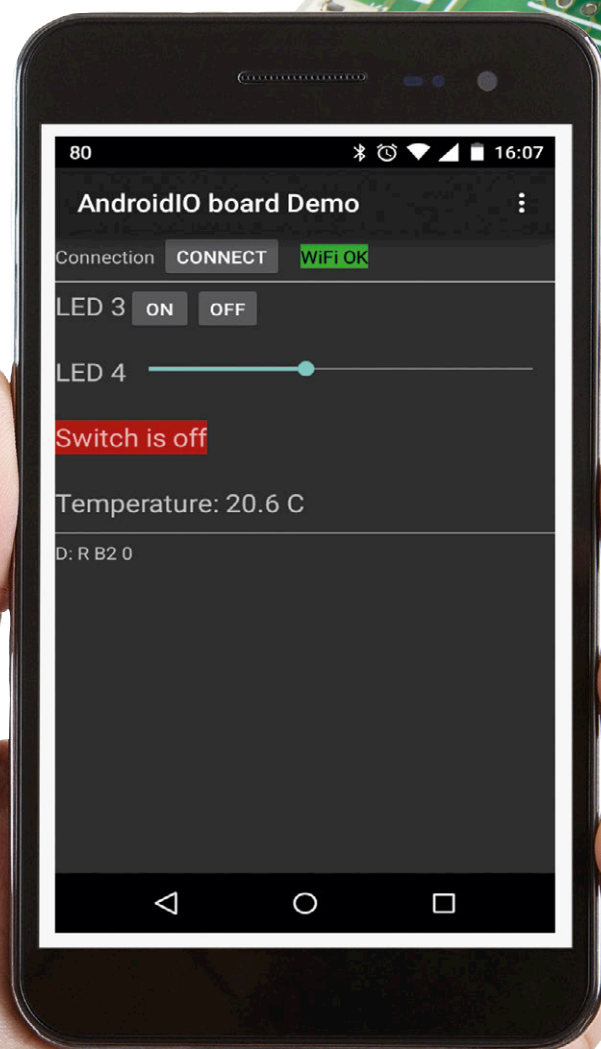
(150175)

Liens

- [1] www.youtube.com/watch?v=6xe19XnX5L0
- [2] www.youtube.com/watch?v=bMuCACqwI4s
- [3] <http://ctms.engin.umich.edu/CTMS/index.php?example=InvertedPendulum§ion=SystemModeling>
- [4] Livre de Dan Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*

ESP8266 sur la carte d'entrées/sorties Android

lancez-vous dans la mise à jour du micrologiciel



Elbert Jan van Veldhuizen (Pays-Bas)

La carte d'E/S Android offre la possibilité de commander facilement des appareils électroniques à partir d'un ordiphone ou d'une tablette. Il y a de la place sur la carte pour au moins sept modules différents de liaison sans fil, dont l'ESP8266, populaire et bon marché. Il faut pour cela reprogrammer le module avec un nouveau micrologiciel. Nous vous expliquons ici comment faire.

L'ESP8266 est un module Wi-Fi très bon marché. Dans la 2^e partie de l'article sur la carte d'E/S Android (10/2015), nous vous avons expliqué que dès que le micrologiciel serait stable, nous reviendrions sur sa mise en œuvre. C'est chose faite maintenant, le micrologiciel est suffisamment stable pour que le module puisse être utilisé. Cet article décrit la procédure à suivre.

ESP8266

Le module ESP8266 est équipé d'un processeur Xtensa, et peut être utilisé en Wi-Fi comme point d'accès (AP, *Access Point*) ou comme client (STA, *Station*).

En mode AP, le téléphone Android se connecte directement au réseau Wi-Fi de l'ESP8266.

En mode STA, l'ESP8266 se connecte à un routeur Wi-Fi. Si le téléphone est aussi connecté à ce routeur, la liaison peut être établie via le « réseau local ». Un téléphone Android peut même servir de point d'accès (modem affilié – *Tethered Modem*) ; l'ESP8266 pourrait alors se connecter en mode STA au téléphone. Dans ce cas, l'ESP8266 doit bien entendu connaître le nom du réseau (SSID) et son mot de passe ; la procédure est expliquée plus loin dans l'article.

L'ESP8266 est disponible sous différentes formes, chacune ayant son numéro de

type. Pour la carte d'E/S Android, c'est la version ESP-01 qui convient ; ce module comporte un connecteur de 2x4 contacts au pas de 2,54 mm (**fig. 1**) et s'adapte directement au connecteur MOD4 de la carte d'E/S. Il y a deux LED sur le module ESP-01, qui indiquent la mise sous tension et le transfert de données. La version ESP-12 est aussi populaire, elle est pratiquement au même prix que l'ESP-01. Ce module comprend plus de broches d'entrées/sorties, mais ne s'adapte pas directement à la carte d'E/S Android. On peut utiliser de la bande adhésive double face et établir les connexions avec des fils, ou réaliser un adaptateur avec un morceau de plaque d'essai pastillée (**fig. 2**). Avec certains modules ESP-12, la broche GPIO15 doit être reliée à la masse via une résistance de moins de 1 kΩ. Cette broche est en fait parfois utilisée pour détecter la présence d'une carte SD externe, sur laquelle démarrer. En outre, il y avait sur le module ESP-12 de l'auteur une LED qui ne s'allumait pas, alors que tout fonctionnait normalement.

L'ESP8266 existe aussi avec diverses capacités de mémoire. Les versions plus anciennes (en général de couleur bleue) possèdent 512 Ko (4 Mbits) de mémoire ; les plus récentes (souvent noires) ont une capacité de 1 Mo (8 Mbits). Pour cette application avec la carte d'E/S Android, 512 Ko suffisent, mais la version 1 Mo convient aussi.

L'ESP8266 produit beaucoup de bruit sur la ligne d'alimentation. Le condensateur présent par défaut sur la carte d'E/S Android a une capacité trop faible pour assurer un fonctionnement stable du module. Il est possible d'ajouter un condensateur entre les broches 2 (+) et 3

(-) du connecteur MOD6 (pour le module à six broches HC-06), voir **figure 3** ; une valeur de 100 µF est plus que suffisante.

Micrologiciel ESP8266

Le micrologiciel standard de l'ESP8266 présente un inconvénient (tout comme celui du RN-141 et des autres modules adaptables à la carte d'E/S Android) : les données ne sont pas transférées de manière transparente. Le micrologiciel utilise les codes AT, qui étaient en usage au temps des modems téléphoniques. Par exemple, pour un transfert de données sur le port série, on doit d'abord indiquer via la commande AT combien de caractères seront transmis. Le débit

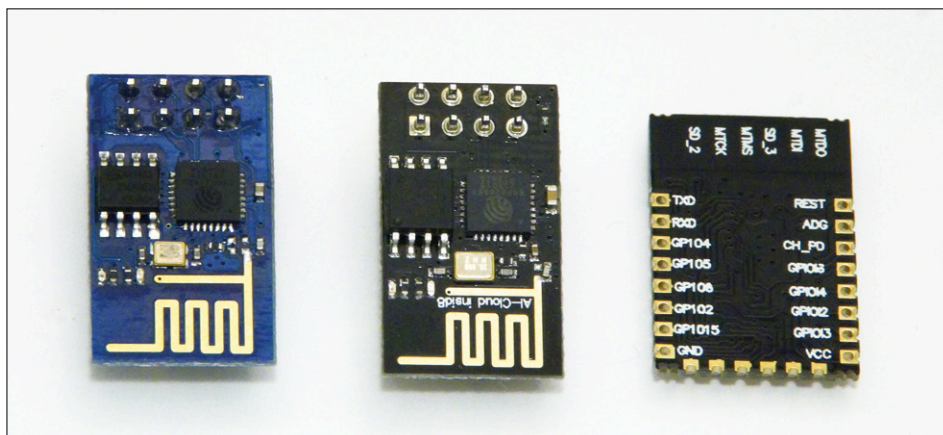
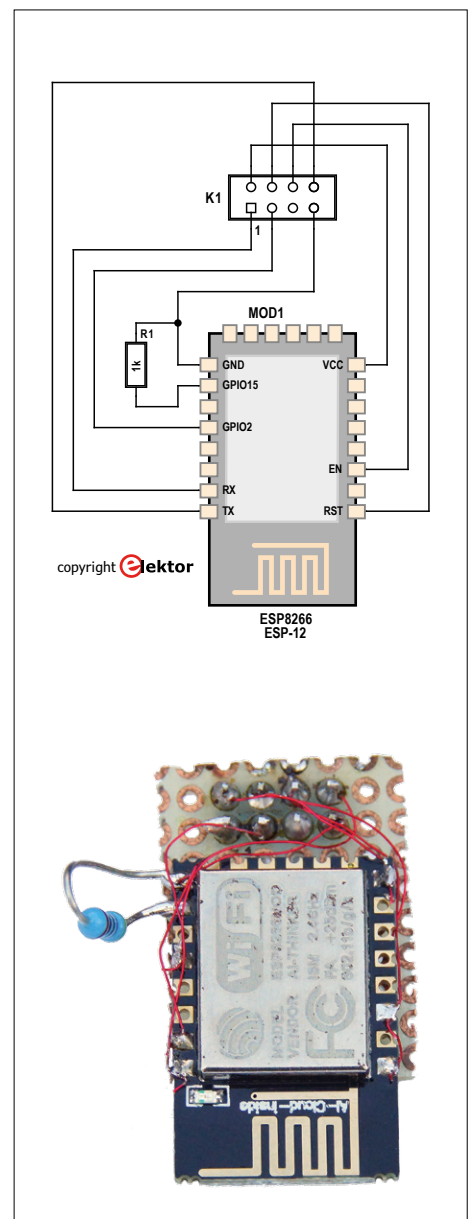


Figure 1. De gauche à droite, trois versions de l'ESP8266 : ESP-01 avec 512 Ko, avec 1 Mo, et la face inférieure d'un ESP-12 avec 1 Mo.

Figure 2. Schéma pour convertir un ESP8266-ESP-12 en un ESP8266-ESP-01.

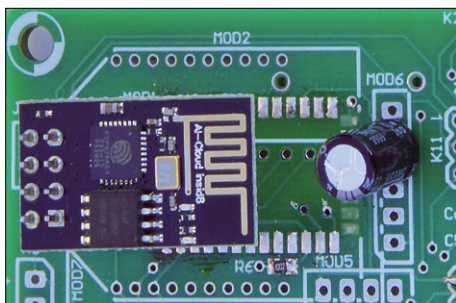


Figure 3. Condensateur additionnel sur le connecteur MOD6.

du port série est fixé à 115 200 bauds, ce qui n'est pas compatible avec la carte d'E/S Android, qui utilise un débit de 9 600 bauds.

Ce qu'il y a de bien avec l'ESP8266, c'est que de nombreuses autres versions du micrologiciel sont disponibles. Il y en a même qui permettent d'établir une liaison série transparente. L'auteur a opté pour le micrologiciel *Beckdac* [1], qui semble stable et ne contient que l'essentiel. Si la programmation vous rebute ou

demande trop de travail, vous pouvez commander un ESP8266 programmé chez Elektor [2].

Paramètres AP et STA standard

La configuration peut être exécutée avec les commandes `+++AT` via la liaison Wi-Fi. Un aperçu complet des commandes se trouve en [3]. Ci-après les principales :

+++AT BAUD baudrate. Réglage du débit du port série. Pour la carte d'E/S Android : `+++AT BAUD 9600`. Si on introduit la commande sans paramètre (et cela vaut pour toutes les commandes), la valeur configurée est retransmise. Ainsi `+++AT BAUD` donne `BAUD=9600 8 N 1`.

+++AT MODE mode-number. Avec 1, l'ESP8266 est en mode STA, avec 2 en mode AP. Et avec 3, aussi bien en STA qu'en AP.

+++AT STA SSID password pour entrer le nom (SSID) et le mot de passe du routeur auquel l'ESP doit se connecter.

+++AT AP SSID password authentication-method permet le réglage des paramètres du réseau Wi-Fi de l'ESP8266. La méthode d'authentification 2 utilise WPA, la 3 WPA2. Avec `+++AT SSID password 3`, l'ESP8266 est protégé contre les tentatives d'intrusion.

D'ailleurs, on peut se connecter avec plusieurs ordinateurs et appareils Android, tant en mode AP qu'en mode STA. L'ESP8266 combine alors toutes les liaisons et envoie aussi les résultats à toutes les connexions.

Programmation de l'ESP8266

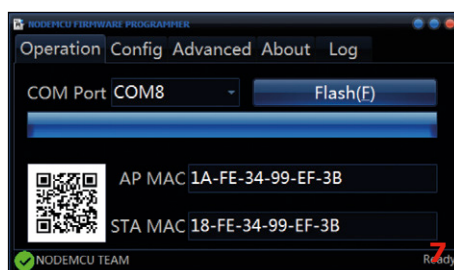
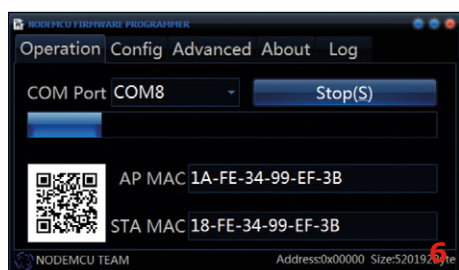
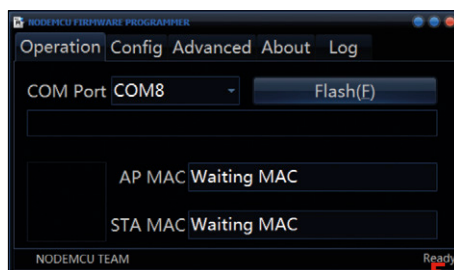
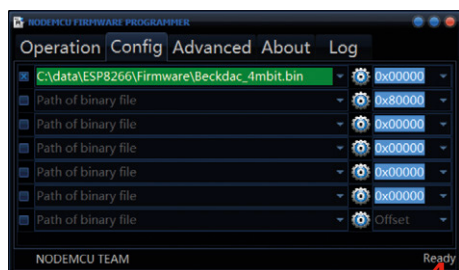
Le micrologiciel se compose de plusieurs parties qui doivent être programmées à partir d'adresses mémoire séparées. Cette procédure est non seulement lourde, certains programmeurs ne s'en accommodent pas bien non plus. L'auteur soupçonne que certains blocs de la mémoire contiennent encore des données, ce qui entraîne un fonctionnement incorrect de l'ESP8266 ; il a donc créé un gros fichier binaire, où les blocs de mémoire non utilisés sont aussi effacés. Le **tableau 1** montre le contenu de ce fichier, une combinaison des fichiers *Beckdac* et du code original [4]. On peut télécharger le fichier en [2] ; en fonction de la version – 512 Ko (bleue) ou 1 Mo (noire) – on utilisera *beckdac_4mbit.bin* ou *beckdac_8mbit.bin*, bien que le fichier de la version 4 Mbits fonctionne en principe aussi avec la version 8 Mbits.

Il existe divers programmeurs pour l'ESP8266, l'auteur en a sélectionné trois : *FLASH_DOWNLOAD_TOOLS* [5], *esp8266_flasher* [6] et *esp8266flasher* [7], la seule différence entre les deux derniers est le tiret bas dans le nom. Bien que l'auteur ait réussi une bonne programmation avec les trois, il estime cependant que la combinaison du dernier programmeur et du gros fichier binaire précité donne le résultat le plus stable.

Instructions pour la programmation

Afin d'éviter les maux de tête, il faut suivre les étapes suivantes :

- Préparation : téléchargement et installation du logiciel et du matériel.
- Option : test de l'ESP8266 (avant programmation).



Figures 4-7. Le logiciel *esp8266flasher* pendant la programmation.

Tableau 1. Contenu des fichiers *beckdac_4mbit.bin* et *beckdac_8mbit.bin*.

adresse	fichier Bin	source
0x00000	0x00000.bin	Beckdac [1]
0x3E000	blank.bin	Espressif [4]
0x40000	0x40000.bin	Beckdac [1]
0x7E000	blank.bin	Espressif [4]
en sus pour la version 8 Mbits (1 Mo)		
0x80000	0x40000.bin	Beckdac [1]
0xFE000	blank.bin	Espressif [4]

- Programmation du nouveau micrologiciel.
- Configuration et installation des paramètres corrects de l'ESP8266.
- Test.

1. Préparation

Téléchargez le micrologiciel [2] et décompressez le fichier zip.

Téléchargez le programme *esp8266flasher.exe* [7].

Munissez-vous d'un convertisseur USB-série avec une tension de 3,3 V (par ex. un des convertisseurs à FT232H décrits dans les numéros d'avril et novembre 2015), et installez éventuellement les pilotes *ad hoc*. Réalisez un connecteur pour relier le module USB-série à la partie gauche de JP2 sur la carte d'E/S Android. N'utilisez que la masse et les deux lignes de données, et pas l'alimentation du module. Prévoyez une alimentation séparée pour la carte, la plupart des modules USB-série ne peuvent fournir suffisamment de courant.

2. Test de l'ESP8266 (avant programmation)

Insérez l'ESP 8266 sur la carte d'E/S Android et connectez le module USB-série et l'alimentation à la carte d'E/S Android. Le LED rouge de l'ESP8266 s'allume si la tension d'alimentation est présente. La LED bleue clignote plusieurs fois lors du démarrage.

Démarrez un émulateur de terminal sur l'ordinateur (par exemple *Tera-term*), et établissez la liaison USB-série, à 115 200 bauds (pour la plupart des modules ESP8266).

Tapez **AT**, vous recevrez **Error** en retour, c'est OK.

Recherchez les réseaux Wi-Fi présents. Vous devriez trouver un nouveau réseau, par exemple « AI-THINKER_XXXXX » (le nom dépend du fournisseur) ; les six derniers caractères représentent la valeur hexadécimale des trois derniers octets de l'adresse MAC.

Le module fonctionne.

3. Programmation de l'ESP8266

Placez le cavalier JP1 sur la carte d'E/S Android, pour que l'ESP8266 soit en mode programmation. Si vous utilisez une résistance, veillez à ce que sa valeur soit inférieure à 1 kΩ.

Connectez le module USB-série à JP2 et l'alimentation à la carte d'E/S Android. Démarrez *esp8266flasher.exe*.

Cliquez sur l'onglet *Config*.

Cliquez sur l'une des roues dentées, chargez le fichier *beckdac_4mbit.bin* ou *beckdac_8mbit.bin*, réglez le décalage (*offset*) sur 0x00000, et cochez la case à gauche du nom du fichier (**fig. 4**).

Sélectionnez dans l'onglet *Operation* le port série correct si ce n'est déjà fait (**fig. 5**).

Appuyez sur *Flash*.

Le programme devrait montrer l'adresse MAC après quelques secondes (**fig. 6**). Si ce n'est pas le cas, redémarrez la carte d'E/S Android en supprimant l'alimentation un instant.

La programmation dure une minute et demie pour la version 4 Mbits, trois minutes pour la version 8 Mbits. Attendez jusqu'à ce que *Ready* apparaisse (**fig. 7**). La programmation est terminée. Retirez le cavalier JP1 et le module USB-série, remettez les cavaliers en place sur JP2 et redémarrez la carte d'E/S Android en supprimant l'alimentation un instant.

4. Configuration

Réglez le réseau Wi-Fi sur *ESP_XXXXXX*. Établissez une liaison *Telnet* (avec par ex. *Teraterm*) vers l'adresse 192.168.4.1 (port 23).

Tapez :

- **+++AT FLASH 1**
- **+++AT BAUD 9600**
- **+++AT MODE 3**

Tapez (avec l'adresse et le mot de passe de votre réseau local Wi-Fi) :

- **+++AT STA SSID Password**

Tapez (avec une adresse et un mot de passe de votre choix) :

- **+++AT AP SSID Password 3**

L'ESP8266 va redémarrer automatiquement et installer un nouveau réseau Wi-Fi. Attendez que cette opération soit terminée.

L'ESP8266 est maintenant prêt à l'emploi. Si vous n'utilisez l'ESP8266 qu'en mode STA ou AP, vous devez encore régler respectivement MODE 1 ou MODE 2, pour ne plus devoir configurer AP ou STA.

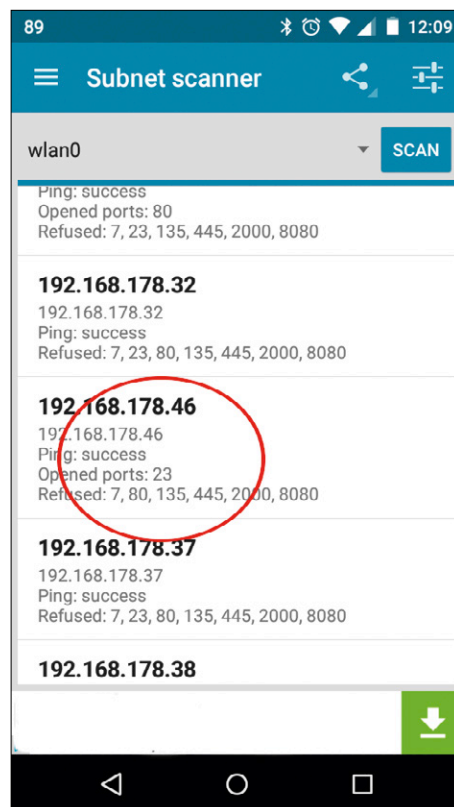


Figure 8. Scan Subnet sur le port 23 pour dépister l'ESP8266 sur le réseau.

5. Test

Avec la commande **+++AT AP** vous pouvez vous connecter au réseau Wi-Fi de l'ESP8266, à partir d'un téléphone ou d'une tablette Android, et avec les données installées précédemment.

Si l'ESP8266 est à portée du réseau Wi-Fi, il se connectera. Vous pouvez rechercher l'adresse – avec par ex. l'appli *Ping Tools* [8] – en réalisant un scan vers le port 23 (**fig. 8**).

Vous pouvez maintenant tester la carte d'E/S Android, en lançant par exemple l'appli de démonstration, ou en envoyant une commande (G Z par ex.) via l'appli *Telnet* [9].

Et si ça ne fonctionne pas ?

Si vous avez bien suivi les instructions ci-devant, cela devrait fonctionner. Mais si un problème subsiste malgré tout, parcourez les points ci-après.

L'ESP8266 produit beaucoup de bruit sur la ligne d'alimentation. Un bon antiparasitage est nécessaire. Placez le condensateur additionnel.

L'ESP8266 consomme beaucoup en pointe (200 à 300 mA). La plupart des

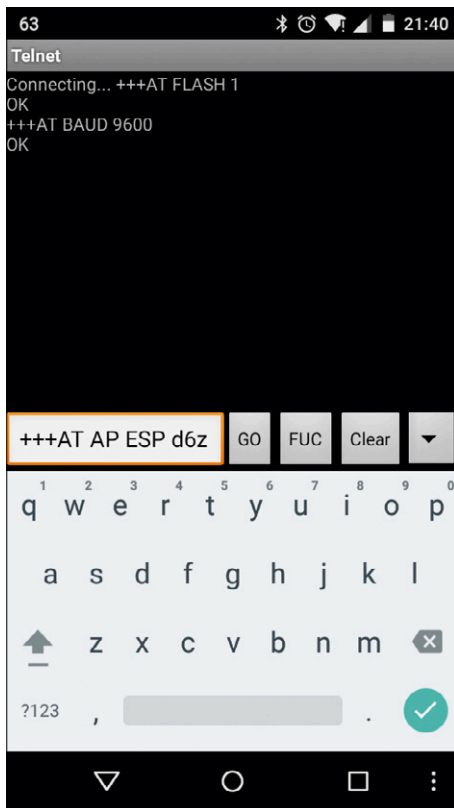


Figure 9. Configuration de l'ESP8266 avec *Simple Telnet*.

modules à FT232 (inclus le BoB d'Elektor) ne peuvent pas délivrer un tel courant. Pendant la programmation, il faut utiliser une alimentation séparée.

Soyez patient. Il a déjà fallu à l'auteur deux heures pour installer le logiciel sur l'ESP8266. Le signal Wi-Fi était bien présent, mais à part ça, rien ! Il s'en est tenu là pour la journée ; le lendemain, il a branché l'alimentation, et tout marchait ! Les raisons exactes de ce dysfonctionnement sont inconnues (il n'y a pas de gros condensateur sur le circuit) ; il est

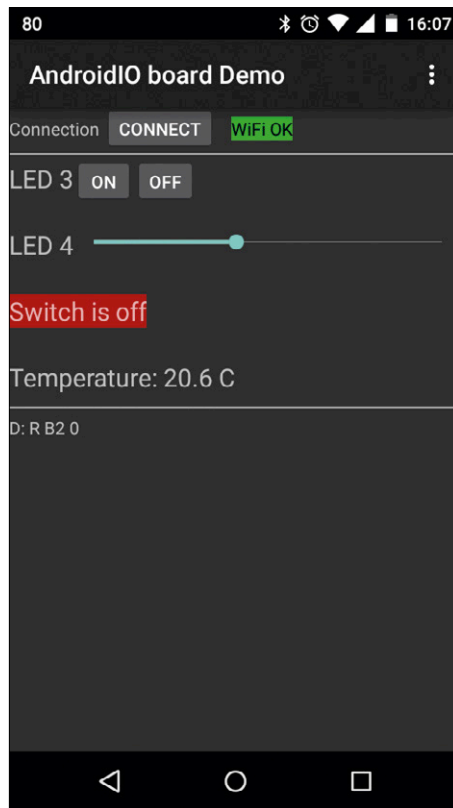


Figure 10. Appli de démonstration pour la combinaison ESP8266 / carte d'E/S Android.

possible que le module USB-série reçoive encore quelque courant via les lignes de données, ce qui entrave la remise à zéro. Leçon à retenir : supprimez l'alimentation du module quelque temps, il redémarrera peut-être correctement.

Il n'est pas possible de vérifier si le micrologiciel est correctement enregistré. Si les paramètres ont été mal réglés, des erreurs peuvent en résulter. Avec *Flash_download_tools*, les paramètres peuvent être sélectionnés manuellement. L'option la plus sûre est d'écrire 20 MHz et de régler DIO.

La connexion au module (réseau Wi-Fi *ESP_XXXXXX*) ne fonctionne souvent pas sous Windows 8.1. Avec un téléphone ou une tablette Android, ou un ordinateur sous Linux, cela fonctionne en principe bien, sauf qu'il faut parfois attendre une petite minute. L'auteur a donc utilisé *Simple Telnet* de Google Play [9] pour la configuration (fig. 9).

Il semble parfois que la liaison *Telnet* n'est pas encore établie (message *Connecting...*). Vous pouvez cependant taper, les caractères apparaissent bien.

Les commandes *+++AT* ne fonctionnent que via le Wi-Fi (via la liaison *Telnet*), pas via le port série de l'ESP8266.

N'oubliez pas de fixer le débit à 9 600 bauds.

Appli de démonstration

Dans les trois articles sur la carte d'E/S Android, des applis sont présentées pour diverses applications. Pour cet article, une appli simple a été développée pour l'utilisation des LED, des touches sensibles et de la résistance CTN. La figure 10 montre une capture d'écran de l'appli. La LED 3 de la carte d'E/S Android peut être allumée ou éteinte, et la luminosité de la LED 4 réglée entre 0 et 100% ; un changement d'état des touches sensibles est aussi transmis et montré, et enfin la température de la CTN est lue chaque seconde. Dans le menu de configuration, on peut, comme dans celui de toutes les autres applis, régler le module de communication en Bluetooth, Wi-Fi, périphérique USB ou hôte USB. Le fichier APK de l'appli et le code source peuvent être téléchargés en [2]. ◀

(150703 – version française : Jean-Louis Mehren)

Liens

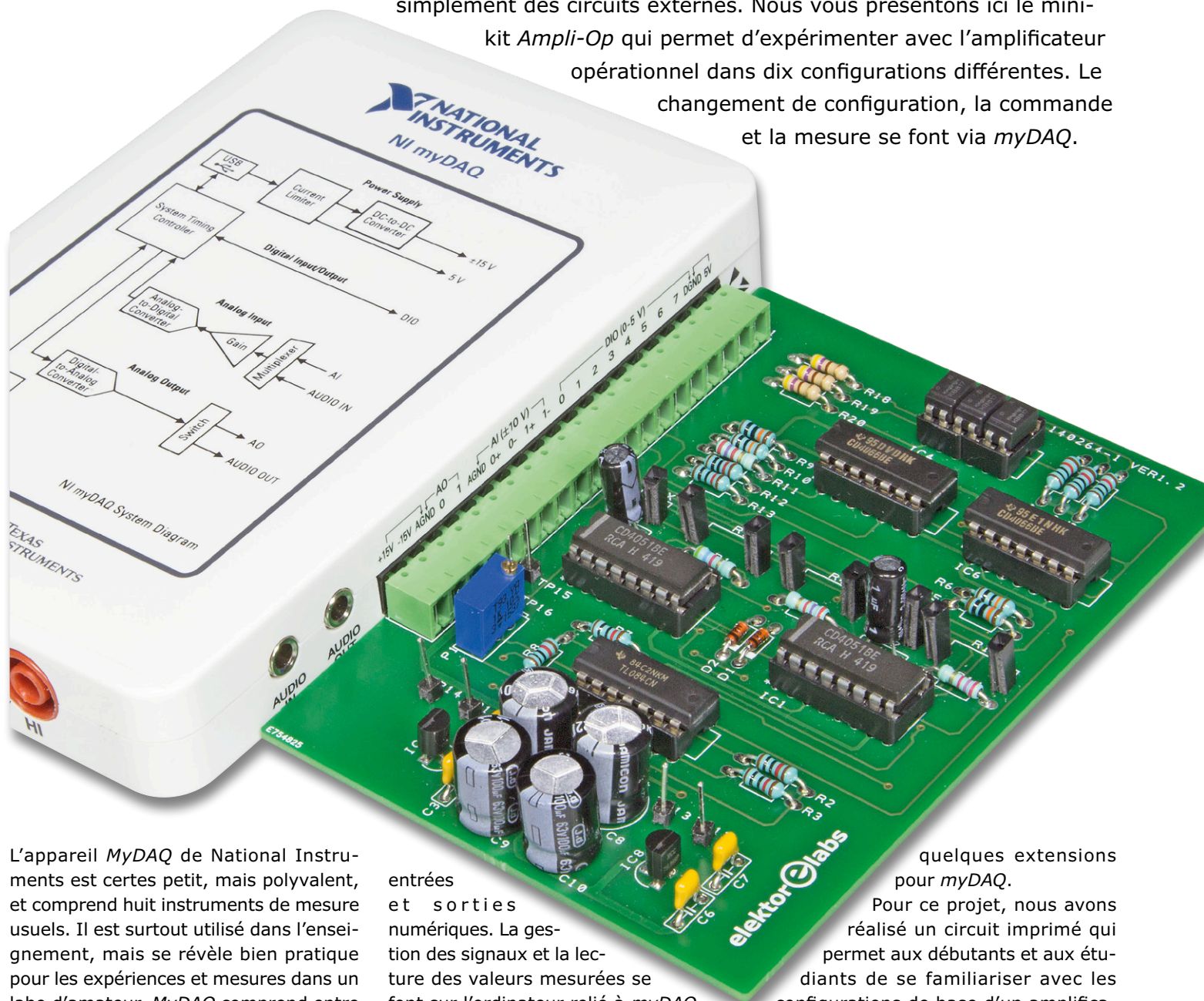
- [1] <https://github.com/beckdac/ESP8266-transparent-bridge>
- [2] www.elektormagazine.fr/150703
- [3] README.md sur <https://github.com/beckdac/ESP8266-transparent-bridge>
- [4] https://github.com/espressif/esp8266_at/tree/master/bin
- [5] FLASH_DOWNLOAD_TOOLS_v1.2_150512.zip : https://drive.google.com/file/d/0B_ctPy0pJuW6V2ViNDR0NGdnYTQ/view?pli=1
- [6] esp8266_flasher : <https://drive.google.com/file/d/0B3dUKfqzZnlwVGc1YnFyUjgxeE/view?usp=sharing>
https://github.com/Stadslab/ESP8266_example/tree/master/ESP8266_flasher_V00170901_00_Cloud%20Update%20Ready
- [7] esp8266flasher : <https://github.com/nodemcu/nodemcu-flasher>
- [8] Ping Tools : <https://play.google.com/store/search?q=ping%20tools&c=apps&hl=fr>
- [9] Simple Telnet : <https://play.google.com/store/apps/details?id=com.telnet&hl=fr>

mini-kit ampli-op pour myDAQ

expérimenter avec dix configurations différentes

Sunil Malekar et Roy Aarts (Labo d'Elektor)

L'instrument de mesure *myDAQ* de National Instruments (NI) est un outil didactique polyvalent, qui peut aussi être utilisé par l'amateur. Grâce à un connecteur sur le côté de l'appareil, on peut y relier très simplement des circuits externes. Nous vous présentons ici le mini-kit *Ampli-Op* qui permet d'expérimenter avec l'amplificateur opérationnel dans dix configurations différentes. Le changement de configuration, la commande et la mesure se font via *myDAQ*.

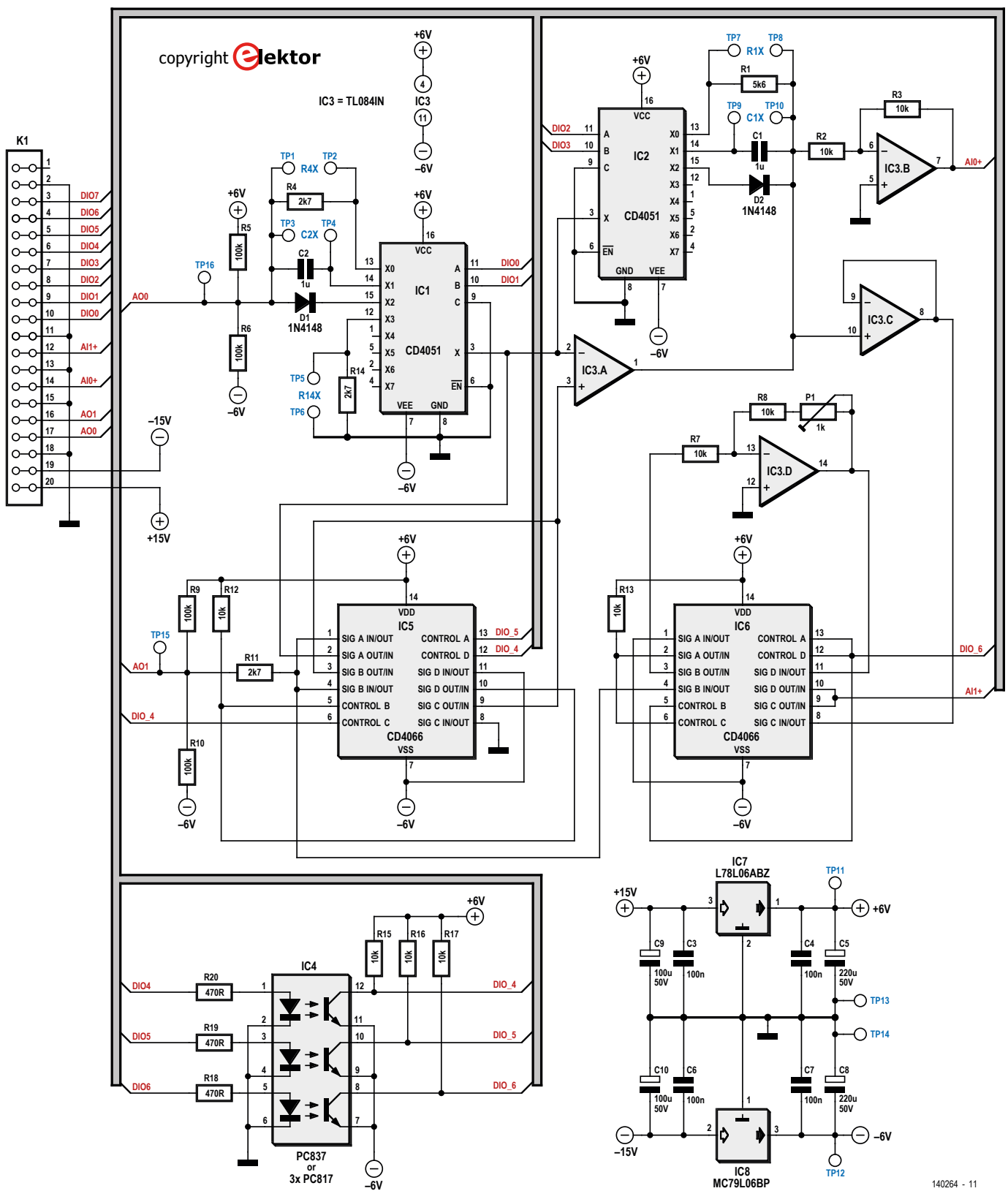


L'appareil *MyDAQ* de National Instruments est certes petit, mais polyvalent, et comprend huit instruments de mesure usuels. Il est surtout utilisé dans l'enseignement, mais se révèle bien pratique pour les expériences et mesures dans un labo d'amateur. *MyDAQ* comprend entre autres un oscilloscope à deux canaux, un multimètre numérique, un générateur de fonctions à deux voies, et quelques

entrées et sorties numériques. La gestion des signaux et la lecture des valeurs mesurées se font sur l'ordinateur relié à *myDAQ* (liaison USB). Le connecteur sur le côté permet d'y relier ses propres circuits, et chez Elektor, nous avons déjà pensé à

quelques extensions pour *myDAQ*.

Pour ce projet, nous avons réalisé un circuit imprimé qui permet aux débutants et aux étudiants de se familiariser avec les configurations de base d'un amplificateur opérationnel (AOP), ainsi qu'avec les filtres. Nous avons rassemblé sur le circuit quelques AOP et un grand nombre



140264 - 11

Figure 1. Le circuit semble complexe, mais il n'en est rien : juste quelques AOP qui peuvent être configurés de différentes manières à l'aide d'interrupteurs.

d'interrupteurs électroniques, ce qui permet – à partir de *myDAQ* – le changement de configuration et des mesures en temps réel. À l'aide de boutons virtuels sur l'écran, il est possible de choisir la configuration et de régler divers paramètres, tels que la fréquence, la forme d'onde et l'amplitude. Il y a quatre fenêtres graphiques qui permettent de visualiser les signaux d'entrée et de sortie des deux canaux. En fait le logiciel de ce projet a été écrit dans *LabVIEW*. Ceci permet donc aussi d'apprendre comment les signaux de sortie sont programmés, et comment les signaux d'entrée de *myDAQ* sont transférés sur l'écran.

Le boîtier *myDAQ*

L'instrument d'acquisition de données *myDAQ* de NI est pourvu de deux entrées analogiques avec un taux d'échantillonnage maximal de 200 éch/s et une résolution de 16 bits, de deux sorties analogiques qui présentent les mêmes caractéristiques, huit entrées/sorties numériques, un compteur-temporisateur universel de 32 bits, un multimètre numérique à 3½ chiffres, et d'une partie alimentation qui peut délivrer 5 V/100 mA et ±15 V/32 mA. Sur le côté, il

y a un connecteur à vingt broches pour relier des circuits externes.

La liaison avec l'ordinateur se fait via une interface USB 2.0 à grande vitesse. Un certain nombre d'instruments virtuels pour les fonctions usuelles sont fournis avec *myDAQ* (*ELVISmx Software Instruments*) ; ils peuvent fonctionner de manière autonome. Pour réaliser d'autres applications, il faut aussi disposer d'une version de *LabVIEW*.

Notre circuit

Le circuit de la **figure 1** peut sembler complexe de prime abord, mais il n'en est rien. Il tourne essentiellement autour de l'AOP IC3.A, dont la configuration peut être sélectionnée à l'aide de deux multiplexeurs/démultiplexeurs analogiques à huit canaux, IC1 et IC2. Avec IC1, on sélectionne le composant relié à l'entrée de l'AOP ; cela peut être une diode (D1), un condensateur (C2), ou une résistance (R4) en série avec l'entrée inverseuse, ou une résistance (R14) entre cette entrée et la masse pour une configuration non-inverseuse. Avec IC2, on choisit le composant dans la boucle de contre-réaction de l'AOP : une diode (D2), un condensateur

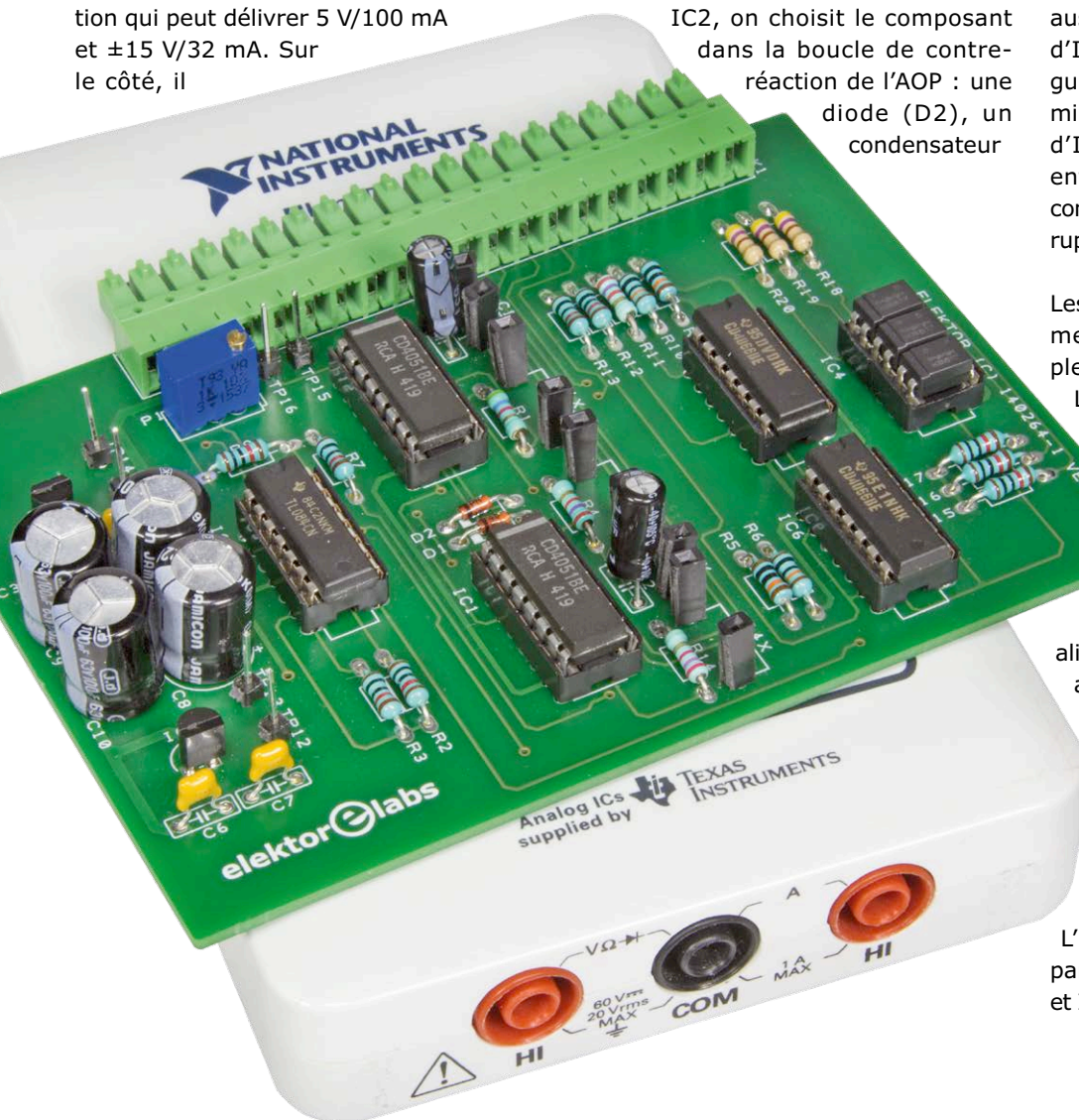
(C1) ou une résistance (R1). Cela permet déjà de mettre en œuvre nombre de configurations différentes. En outre, il y a encore des petits connecteurs qui permettent de mettre une résistance ou un condensateur (R1X, C1X, R4X, C2X, R14X) en parallèle avec la valeur déjà sélectionnée. À la sortie d'IC3.A se trouvent un étage inverseur (IC3.B) et un non-inverseur (IC3.C). La sortie de l'étage inverseur est toujours raccordée à la première entrée analogique (AI0+) de *myDAQ* ; en fonction de la configuration choisie via le panneau de gestion, la sortie de l'étage non-inverseur ou celle d'un AOP configurable (IC3.D) est reliée à la deuxième entrée (AI1+) de *myDAQ*.

Nous trouvons encore sur le circuit deux 4066 (IC5 et IC6), quadruples interrupteurs bidirectionnels. Avec IC5, il est possible de connecter la deuxième sortie analogique de *myDAQ* via R11 à une des deux entrées d'IC3.A, afin de réaliser un sommateur ou un soustracteur. Il est encore possible d'envoyer le signal après la résistance R1 vers l'étage construit autour d'IC3.D. L'interrupteur C d'IC5 permet aussi de mettre l'entrée non-inverseuse d'IC3.A à la masse dans certaines configurations. Les interrupteurs d'IC6 déterminent si c'est la sortie d'IC3.B ou celle d'IC3.C qui est envoyée à la deuxième entrée de *myDAQ* (interrupteur C), et conditionnent l'amplificateur IC3.D (interrupteur B pour l'entrée, D pour la sortie).

Les signaux numériques DI0 à DI6 permettent la mise en œuvre des multiplexeurs et interrupteurs analogiques.

Le **tableau 1** montre le niveau des signaux pour les diverses configurations. Une mention particulière pour DI4, DI5 et DI6, qui pilotent IC5 et IC6 : vu que les lignes d'entrée/sortie délivrent des signaux de commutation à 3,3 V et qu'IC5 et IC6 sont alimentés symétriquement en ±6 V, une adaptation de niveau est nécessaire. C'est le rôle des trois photocoupleurs d'IC4 : lorsque le signal sur l'une des entrées passe au niveau haut, le transistor du photocoupleur conduit et tire le signal de +6 V vers -6 V. Une adaptation de niveau avec inversion incluse !

L'alimentation du circuit est assurée par les deux régulateurs de tension IC7 et IC8, qui abaissent les tensions de ±15V



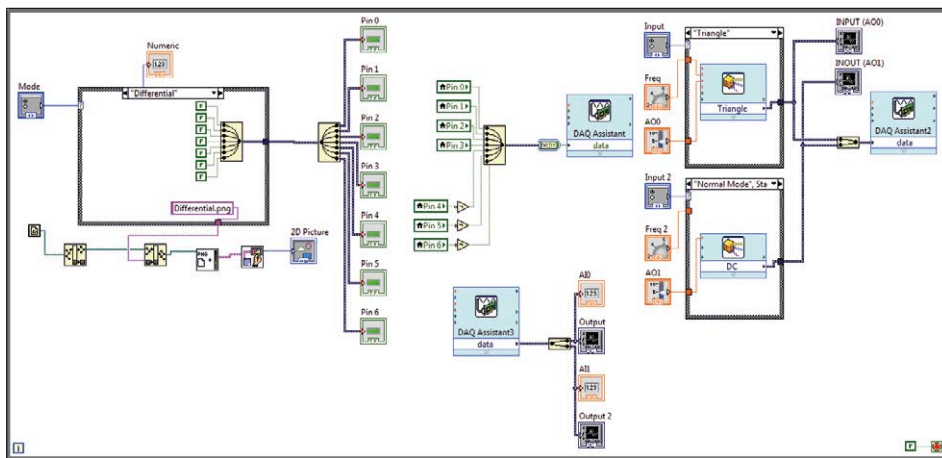


Figure 3. Le schéma fonctionnel du programme LabVIEW, écrit pour le mini-kit Ampli-Op.

fournies par myDAQ à ± 6 V.

Enfin, le connecteur à vingt broches K1 permet toutes les liaisons nécessaires entre le circuit et myDAQ.

Construction

Nous avons dessiné un circuit imprimé pour ce projet, son tracé est en **figure 2**. Nous l'avons voulu très aéré et n'avons utilisé que des composants à fils ou à pattes, et ce pour en faciliter la construction. Les circuits intégrés peuvent être placés sur des supports ; c'est très pratique pour les AOP si on veut tester plusieurs modèles (avec un brochage identique, bien entendu). Nous n'utilisons que des signaux à basse fréquence, les

risques d'interférence sont donc faibles. Pour TP1 à TP16, on peut utiliser des barrettes d'embases mâles ou femelles à dix points, que l'on coupe en fonction des besoins ; les picots à souder conviennent aussi, mais c'est moins pratique.

Après avoir garni le circuit imprimé et monté le connecteur à vingt broches, on peut le relier au connecteur sur le côté de myDAQ. Il est conseillé de munir le circuit imprimé de supports aux coins extérieurs ; le circuit ne sera donc pas soutenu que par le connecteur, et on n'exercera pas de pression sur celui-ci lors de l'installation de composants supplémentaires ou de l'ajustage du potentiomètre par exemple.

Pour ceux qui veulent se lancer dans la construction, nous avons prévu un petit kit (réf. 140264-71) comprenant le circuit imprimé, le connecteur à vingt broches – qui n'est pas si courant – et le régulateur de tension MC79L06 – plus très courant lui non plus.

Le logiciel

Pour l'utilisation du circuit, nous avons écrit un module logiciel dans LabVIEW, qui permet de sélectionner la configuration et de « jouer » avec les signaux d'entrée, tout en regardant ce qui se passe à la sortie. Ce module est gratuit et téléchargeable sur le site du magazine [1] ; son utilisation nécessite cependant que vous ayez installé une version de LabVIEW sur votre ordinateur.

La **figure 3** montre le schéma fonctionnel du programme. Sur la gauche, la sélection de la configuration, qui conditionne les sorties numériques de myDAQ ; à droite, la production des signaux des sorties analogiques. Et en bas au milieu, la reproduction des signaux analogiques d'entrée et de sortie dans quatre fenêtres graphiques. Le schéma est donc divisé en une partie logique numérique, un générateur de fonction et un oscilloscope.

Après démarrage du programme (**figure 4**), on peut commencer les expérimentations. Sur la gauche de l'écran, il y a dix cases à cocher pour choisir la

Liste des composants

Résistances :

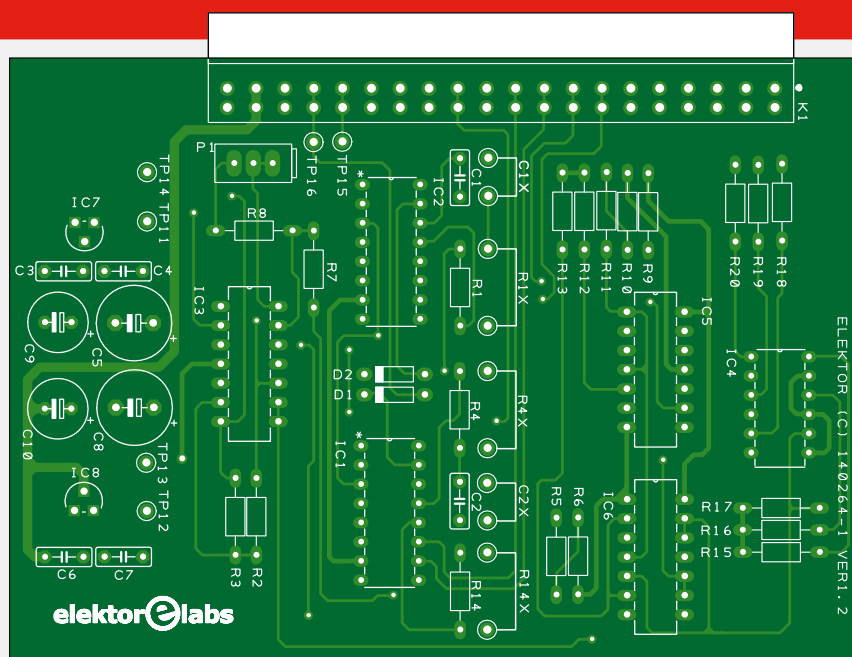
(5% / 250 mW)

R1 = 5,6 k Ω
 R2, R3, R7, R8, R12, R13, R15,
 R16, R17 = 10 k Ω
 R4, R11, R14 = 2,7 k Ω
 R5, R6, R9, R10 = 100 k Ω
 R18, R19, R20 = 470 Ω
 P1 = ajustable 1 k Ω

Condensateurs :

C1, C2 = 1 μ F / 50 V, radial
 C3, C4, C6, C7 = 100 nF
 C5, C8 = 220 μ F / 50 V, radial, pas de 10 mm
 C9, C10 = 100 μ F / 50 V, radial, pas de 8 mm

Figure 2. Le circuit imprimé de ce mini-kit Ampli-Op n'est pas surchargé. Il est facile à assembler avec des composants à fils ou à pattes.



configuration des AOP du circuit ; la configuration réglée est représentée juste en dessous. À droite des cases, sept indicateurs verts qui représentent les niveaux des sorties numériques de *myDAQ*. Ceux-ci correspondent au **tableau 1**, et actionnent les multiplexeurs analogiques du circuit. Nous avons ensuite quelques réglages des signaux de sortie de *myDAQ* : la forme peut être un signal continu, un sinus, un triangle ou un signal carré. Les curseurs verticaux ajustent la valeur et la polarité du décalage continu, ainsi que l'amplitude du signal. La fréquence est choisie à l'aide des boutons rotatifs « freq » ; il vaut mieux la maintenir basse, pour bien suivre l'évolution des signaux dans les fenêtres graphiques. Les deux fenêtres de gauche représentent les signaux d'entrée, celles de droite les signaux de sortie.

Vous pouvez maintenant expérimenter à souhait et en voir directement le résultat. Amusez-vous bien !

(140264 – version française : Jean-Louis Mehren)

Lien

[1] www.elektormagazine.fr/140264

Tableau 1.

configuration	DIO7	DIO6	DIO5	DIO4	DIO3	DIO2	DIO1	DIO0
inverseur	X	0	0	1	0	0	0	0
non-inverseur	X	0	0	0	0	0	1	1
sommateur	X	0	1	1	0	0	0	0
soustracteur	X	0	0	0	0	0	0	0
différentiateur	X	0	0	1	0	0	0	1
intégrateur	X	0	0	1	0	1	0	0
logarithme	X	0	0	1	0	1	0	0
antilogarithme	X	0	0	1	0	0	1	0
comparateur	X	0	0	1	1	1	0	0
amplificateur ajustable	X	1	0	1	0	0	0	0

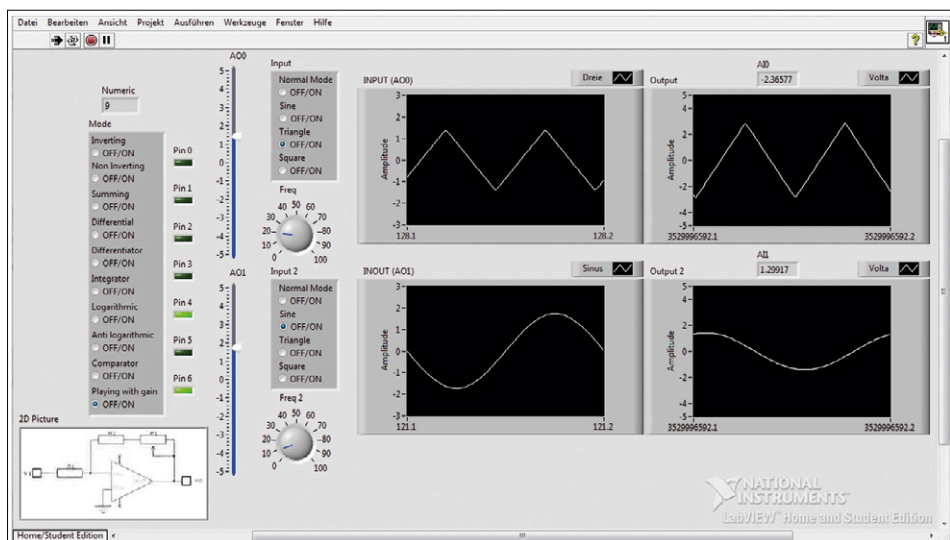
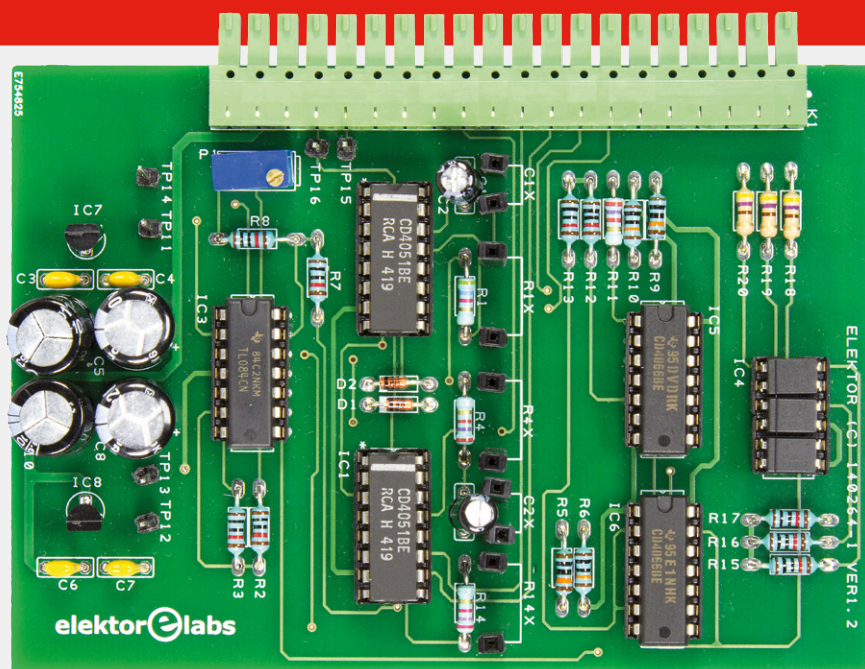


Figure 4. Le programme tel qu'il apparaît sur l'écran. On sélectionne la configuration et on règle les signaux d'entrée à gauche, et les fenêtres de droite montrent les traces des signaux d'entrée et de sortie.



Semi-conducteurs :

IC1, IC2 = CD4051BE
 IC3 = TL084
 IC4a, IC4b, IC4c = 1x PC837 ou 3xs PC817
 IC5, IC6 = CD4066BE
 IC7 = L78L06
 IC8 = MC79L06
 D1, D2 = 1N4148

Divers :

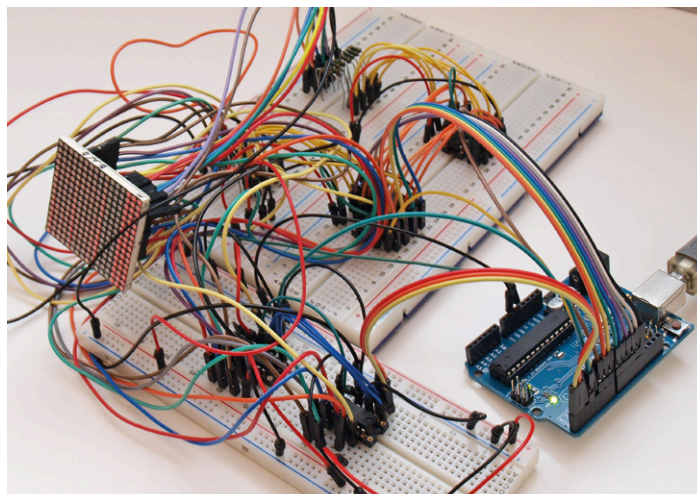
Support pour circuit intégré DIP à 6 broches
 Support pour circuit intégré DIP à 14 broches
 Support pour circuit intégré DIP à 16 broches
 K1 = connecteur à 20 broches,
 pas de 3,81 mm, pour liaison avec *myDAQ*
 TP1 à TP10 = embase femelle, à 1 rangée,
 au pas de 2,54 mm
 TP11 à TP16 = embase mâle, à 1 rangée,
 au pas de 2,54 mm
 Mini-kit Ampli-Op réf. 140264-71 (disponible
 sur www.elektor.fr) qui comprend le circuit
 imprimé, le connecteur *myDAQ* à 20 broches
 et un régulateur de tension MC79L06

journal défilant avec Arduino

& matrice de 16 x 16 LED

Tam Hanna

Avec une carte Arduino, un afficheur à matrice de LED et quelques autres composants, réalisez un journal lumineux défilant. Pour le logiciel, un outil gratuit de Mikroelektronika vous aidera à traduire, en un clin d'œil, toute une police de caractères en lignes de code.



Les reportages qui nous arrivent des différentes capitales à travers le monde contiennent inmanquablement un ou plusieurs plans sur des journaux lumineux qui défilent sur les façades des bourses ou autres places publiques. Réaliser pareil jour-

nal est une application intéressante à plus d'un titre, parce qu'elle réclame des connaissances sur le matériel autant que sur le logiciel. Pas de panique, nous vous tiendrons la main pour la traversée !

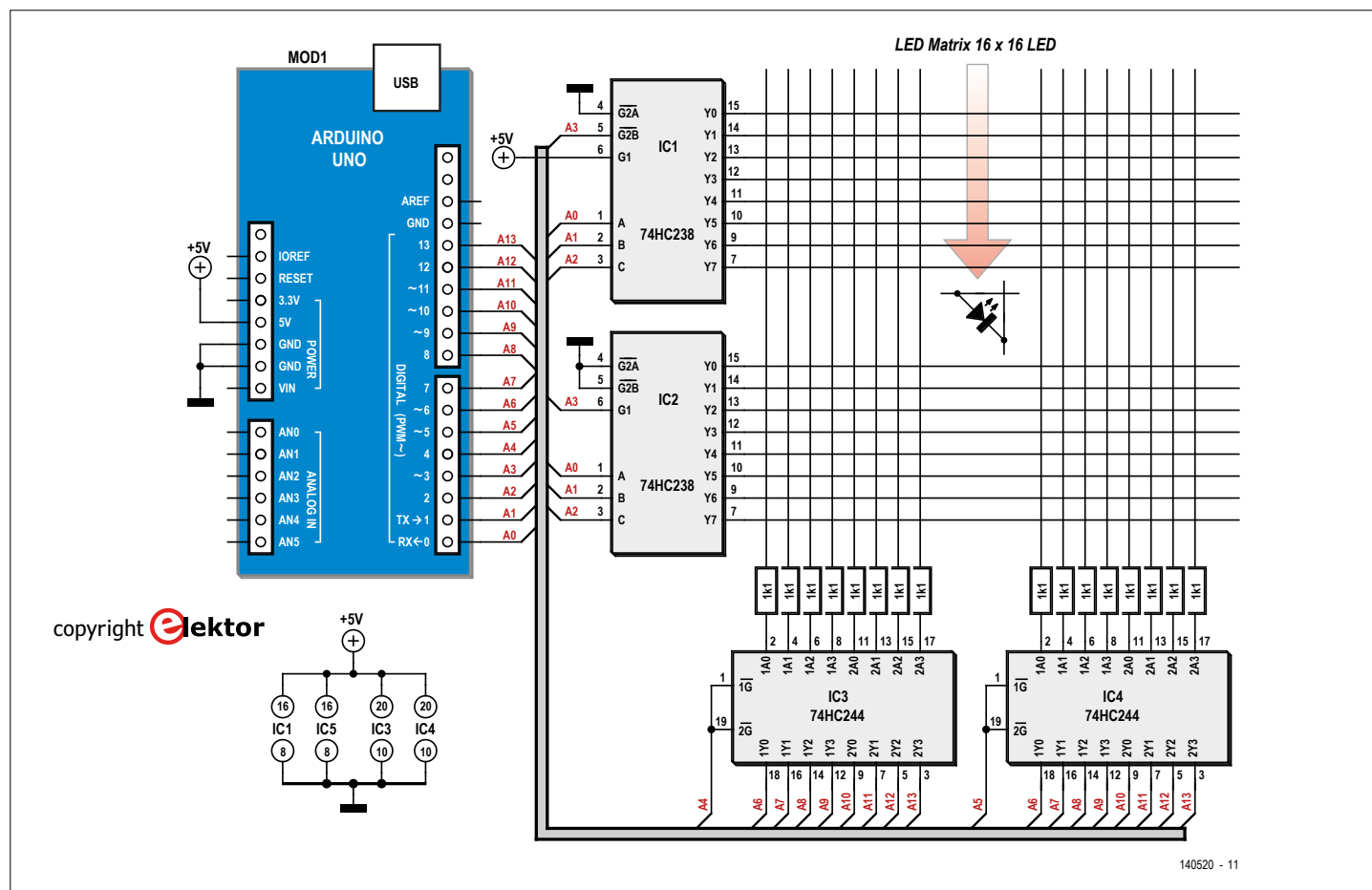


Figure 1. Le schéma. Les deux 74238 alimentent chacun une rangée de la matrice de LED. Les commutations de chaque LED dépendent des 74244. Leur pilotage réclame la totalité des lignes de port numériques de l'Arduino.

Le circuit

Attaquons par la **figure 1** qui représente l'ensemble du schéma. À chaque croisement de lignes se trouve une diode électroluminescente. Arduino pilote notre matrice de 16x16 LED réparties en quatre blocs. Les puces 74238 ont pour tâche d'activer les rangées les unes après les autres.

En fait la commutation de chaque LED d'une rangée active est produite au moyen de deux puces 74244 auxquelles sont branchées les cathodes que chaque colonne peut attirer au niveau bas. Ce pilotage occupe toutes les lignes de port numériques d'Arduino.

Pour augmenter la luminosité, il faut ajouter des transistors, c'est un sujet que nous n'aborderons pas ici, faute de place.

Premiers pas vers le code

Notre but final est de mettre en œuvre une carte Arduino classique. Toutefois pour me simplifier la tâche de mise au point, j'ai commencé avec une carte plus puissante, l'Arduino Zero, qui facilite le développement de l'algorithme grâce au débogueur et à des ressources en abondance – l'optimisation suivra. Ce qui, au premier abord, semble une méthode peu propice peut se révéler en pratique très prometteuse. En général, un système est plus facile à « simplifier » dès lors qu'il a fonctionné – pour arriver à ce résultat, c'est confortable de pouvoir disposer de beaucoup de ressources.

Comme les puces utilisées ressortissent à la famille HC, elles fonctionnent aussi sous 3,3 V et sont donc compatibles avec les deux cartes Arduino ; cette particularité sera utile lors de la transplantation. Il faut seulement veiller lors du travail sur Zero à prendre la tension d'alimentation sur la broche 3V3 d'Arduino. Exécutez alors le programme du **listage 1**.

Avec les deux 74238, l'adressage des rangées s'effectue en appliquant un nombre binaire à quatre chiffres sur les quatre lignes A0 à A3. Dans la fonction `activateRow`, nous mettons d'abord les quatre lignes au niveau bas. Puis, au moyen de quatre tests If – sur les bits de la variable `_which` – nous passons la ligne correspondante au niveau haut.

Le code du **listage 2** fait s'allumer les LED.

Nous activons successivement les rangées en appelant en boucle la fonction `activateRow`. Dans la boucle intérieure, nous mettons d'abord toutes les colonnes au niveau haut pour les désactiver. On les active ensuite une à une. L'effet de déplacement des points s'obtient en insérant une pause avec `delay`.

Détection de panne

Il n'est pas rare de constater un défaut dans le câblage de la matrice. Plutôt que de tout démonter et remonter, un logiciel permet de passer en revue le câblage pour trouver de petites pannes. J'ai donc écrit la routine `mapRow` qui met en lumière la correspondance entre les lignes logiques et les physiques (**listage 3**).

Que fait la police ?

Tout ceci nous permet de commander les points individuellement. Mais pour représenter des signes, il faut une police de caractères. Sur le PC, on en utilise communément au format *TrueType*. Il s'agit de données vectorielles, généralement hors de portée d'un petit microcontrôleur. En outre, la largeur des caractères d'une police TTF est variable (*on parle de police à chasse variable en typographie*). Un « I » prend moins de place

Listage 1. Sélection des rangées

```
void setup() {
  for(int i=0;i<13;i++)
  {
    pinMode(i,OUTPUT);
    if(i>=6)digitalWrite(i,LOW);
  }
}

const int TAM_A0=0;
const int TAM_A1=1;
const int TAM_A2=2;
const int TAM_A3=3;

void activateRow(unsigned char _which)
{
  _which=mapRow(_which);
  digitalWrite(TAM_A0,false);
  digitalWrite(TAM_A1,false);
  digitalWrite(TAM_A2,false);
  digitalWrite(TAM_A3,false);
  if(_which&1) {
    digitalWrite(TAM_A0,true);
  }
  if(_which&2) {
    digitalWrite(TAM_A1,true);
  }
  if(_which&4) {
    digitalWrite(TAM_A2,true);
  }
  if(_which&8) {
    digitalWrite(TAM_A3,true);
  }
}
```

Listage 2. Un point lumineux mobile

```
void loop() {
  while(1==1)
  {
    for(int j=0;j<16;j++)
    {
      activateRow(j);
      for(int i=0;i<8;i++)
      {
        digitalWrite(13,true);
        digitalWrite(12,true);
        digitalWrite(11,true);
        digitalWrite(10,true);
        digitalWrite(9,true);
        digitalWrite(8,true);
        digitalWrite(7,true);
        digitalWrite(6,true);
        digitalWrite(6+i,false);
        delay(25);
      }
    }
  }
}
```

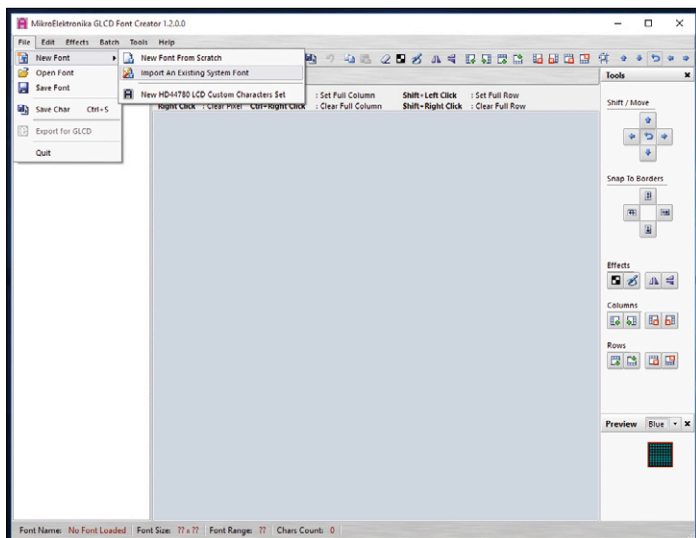


Figure 2. C'est l'outil *GLCD Font Creator* qui assure l'importation de la police de caractères du système.

qu'un « M », par exemple. Ici, il nous faut des lettres de largeurs égales. Dans un premier temps, j'ai utilisé la police *Courier*. Le fabricant d'outils Mikroelektronika propose un programme très utile qui convertit une police en lignes de code – dans sa version de base, il est gratuit. Téléchargez le programme prêt à l'emploi *GLCD Creator* en [1] pour créer une nouvelle police basée sur Courier. Il faut pour cela, comme à la **figure 2**, importer une nouvelle police.

Mettez comme taille de caractères la valeur 13. Puisqu'elle n'est pas dans les présélections, il faut, comme à la **figure 3**, l'introduire manuellement. L'étape suivante consiste à régler les paramètres d'importation de la **figure 4**, le processus d'optimisation prend ensuite un certain temps.

C'est le moment de vérifier la grandeur des signes produits qui s'affiche en bas de l'écran. Si elle n'est pas de 10x13, alors on continue par *Batch > Columns > Ins Column*. Ensuite, il est nécessaire de passer *par Batch > Invert All*.

Cliquez alors sur *Export*, puis sur l'onglet *MicroC*. Le résultat

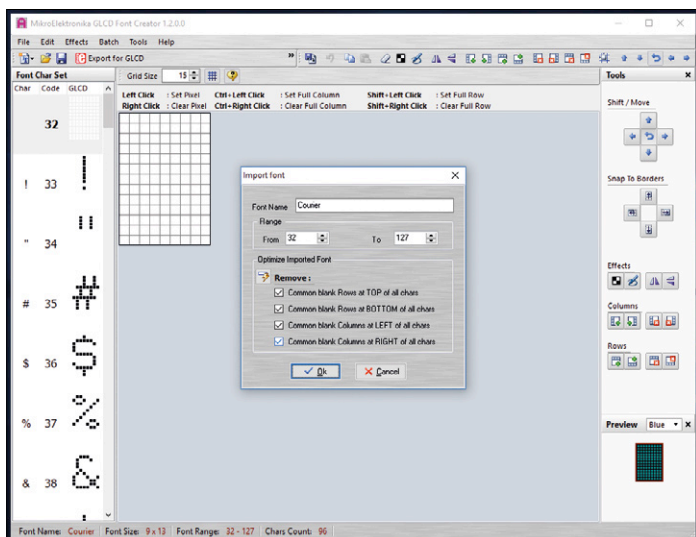


Figure 4. *GLCD Font Creator* élimine automatiquement les pixels inutiles.

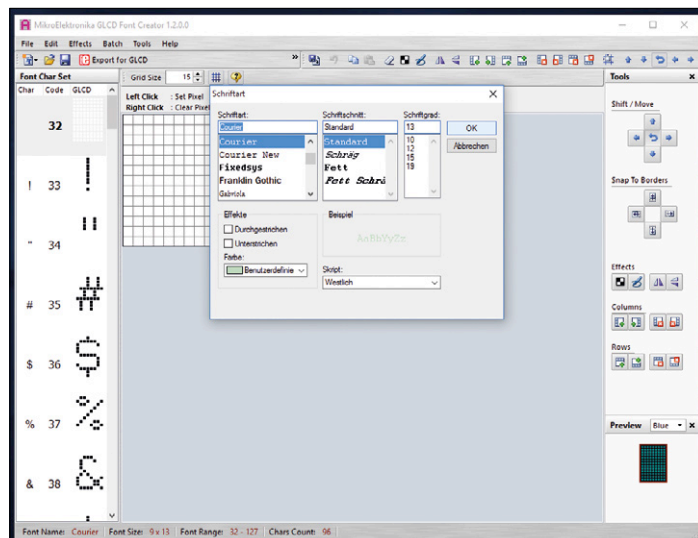


Figure 3. Le choix de la police et de la taille des caractères se fait dans une boîte de sélection. Si la hauteur voulue n'y est pas renseignée, il suffit de l'inscrire soi-même

est un tableau en C comme à la **figure 5**.

Copiez ensuite le code dans le bloc-notes et mettez en surbrillance l'octet de longueur ici inutile, comme le montre la **figure 6**. Copiez ce texte dans le champ *Rechercher* du sous-menu *Édition > Remplacer*. Éliminez toutes les occurrences de ce texte avec des *Rechercher-Remplacer par* (**figure 7**). Puis remplacez la désignation du type de donnée *short* par *int* pour terminer ce travail.

On continue en copiant dans le fichier *.c* le tableau *Courier10x13*. Notre version de la commande de LED doit encore distinguer la partie gauche de la partie droite de l'affichage (c'est pourquoi nous n'avons pas encore traité plus haut le défilement du point allumé). La méthode du **listage 4** active soit la puce 74244 de gauche, soit celle de droite, selon l'état du bit *_hi*.

Un lecteur averti pourrait critiquer ici le choix d'activer un côté de l'afficheur directement avec une ligne de port plutôt qu'avec

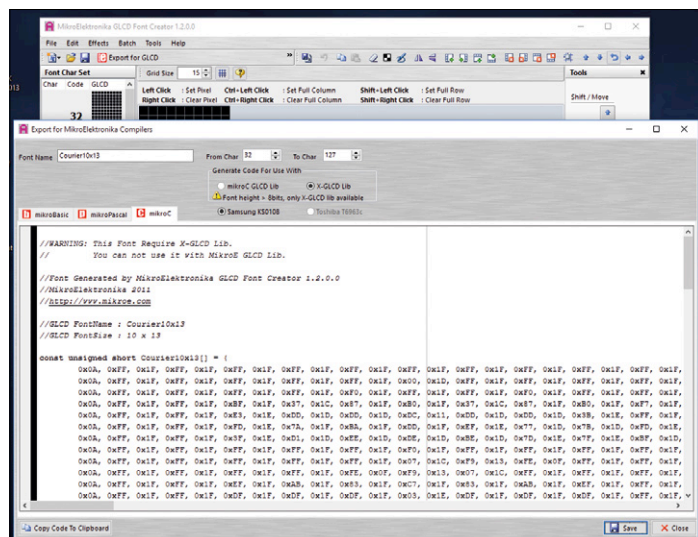


Figure 5. Le résultat : un tableau en C de la description des caractères.

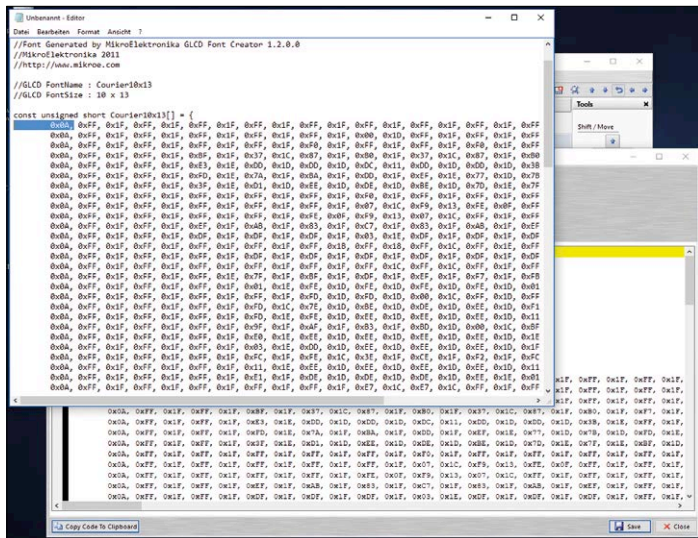


Figure 6. Cet octet est inutile.

un inverseur. C'est bien vrai, mais cela aurait augmenté la complexité du matériel.

Il nous faut encore envisager le rendu de chaque caractère. Une routine de démonstration va nous servir tant pour le développement que pour les tests. Elle travaille avec une boucle For avec la plage de caractères codés 32 à 127 de la police. Chaque caractère est redessiné 120 fois consécutivement, sinon, il aurait l'air délavé (**Listage 5**).

Par défaut, une police créée avec *GLCD Font Creator* contient les caractères codés 32 à 127, les caractères précédents ne font pas partie du tableau. On peut donc simplifier la donne avec une soustraction de 32.

On peut dès lors s'offrir le vrai rendu des caractères (cf. **listage 6**).

Après le calcul du décalage du premier octet du caractère, la boucle parcourt successivement les dix colonnes de l'afficheur. On prend d'abord un octet `whichToWrite` qui commande les colonnes de la partie gauche du schéma (remarquez que les caractères y sont représentés avec une rotation de 90 °). Pour chaque colonne de la représentation du caractère, une rangée de LED est toujours activée dans le schéma.

On utilise une astuce pour appliquer les bits de données sur

Listage 3. Dépistage des pannes matérielles

```
unsigned char mapRow(unsigned char row)
{
    if(row<8)
    {
        return row+8;
    }
    else
    {
        return row-8;
    }
}
```

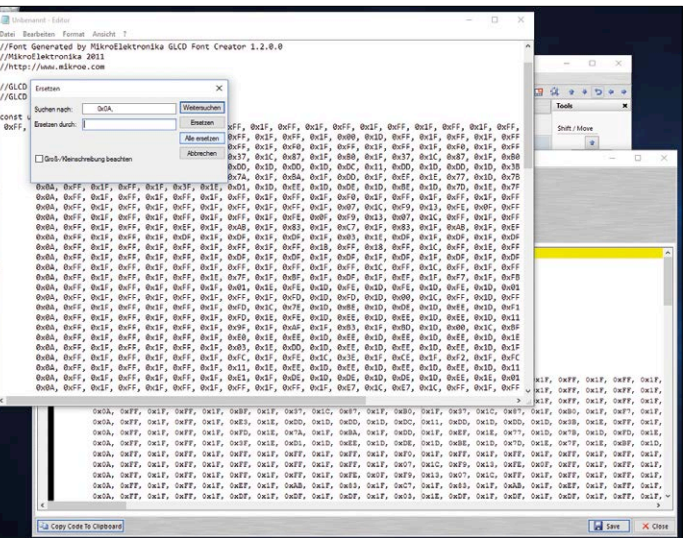


Figure 7. Laisser vide le champ « Remplacer par » supprime aisément toutes les occurrences de l'octet de longueur.

les lignes des colonnes : `digitalWrite` considère toutes les valeurs supérieures à 0 comme vraies. Pour chaque bit, on combine avec un opérateur ET la valeur `whichToWrite` et un masque obtenu en décalant vers la droite l'adresse du bit.

Listage 4. Commande des moitiés gauche et droite de l'afficheur

```
void pickPartOfDisplay(bool _hi)
{
    if(_hi)
    {
        digitalWrite(4,HIGH);
        digitalWrite(5,LOW);
    }
    else
    {
        digitalWrite(4,LOW);
        digitalWrite(5,HIGH);
    }
}
```

Listage 5. Affichage de la police de caractères

```
void loop() {
    while(1==1)
    for (int offsetC=0;offsetC<127-32;offsetC++)
    for(int dela=0;dela<120;dela++)
    {
        //Rendern
    }
}
```

Le fonctionnement de la partie droite de la matrice de LED est semblable. La variable `offset` est incrémentée d'une unité pour atteindre un octet `whichToWrite` qui commandera les colonnes de la partie droite du schéma.

Elle arrive sur les lieux !

Pour réaliser un journal lumineux, nous devons pouvoir écrire les caractères à la bonne position dans la matrice. Pour développer un micrologiciel, il est souvent recommandé d'utiliser une logique fonctionnelle éprouvée, de la transplanter dans une fonction et de la retravailler sur place.

La routine `displayChar` reçoit, en plus du décalage, un autre nombre entier qui indique le décalage vers la droite (positif) ou vers la gauche (négatif), voir **listage 7**.

Le principe de base de la fonction de rendu reste inchangé. La nouvelle version diffère de la précédente par le fait que

les coordonnées atteintes (pour les rangées dans le schéma) sont soumises au décalage `_shoveHowMuch`. Si cela produit un résultat inacceptable, le contenu d'`offset` est incrémenté de deux et la boucle suivante est lancée.

Les caractères s'inscrivent dix fois à la même place (d'où la boucle tout à l'extérieur) pour éviter que l'affichage soit trop pâle.

La boucle principale `loop` ne se limite pas à un seul caractère, elle contient un petit message de bienvenue de plusieurs caractères, envoyés ici avec un certain décalage pour se succéder sur la matrice de LED. La boucle `For` assure la mise en file nécessaire (**listage 8**).

Migration sur AVR

Et maintenant, nous pouvons passer sur Arduino Uno. La police tient tout juste dans la RAM ; grâce à la structure plus éco-

Listage 6. Rendu d'un caractère

```
int offset=widthOfChar*offsetC;

for(int j=0;j<10;j++)
{
    char whichToWrite=Courier10x13[offset];

    activateRow(j);
    pickPartOfDisplay(false);
    for(int i=0;i<8;i++)
    {
        digitalWrite(6+i,(whichToWrite&(1<<i)));
    }
    delayMicroseconds(20);

    offset++;
}
```

```
whichToWrite=Courier10x13[offset];

activateRow(j);
pickPartOfDisplay(true);
for(int i=0;i<8;i++)
{
    digitalWrite(6+i,(whichToWrite&(1<<i)));
}
delayMicroseconds(20);

offset++;

}
```

Listage 7. Déplacement d'un caractère

```
void displayChar(int _offset, int _shoveHowMuch)
{
    for(int dela=0;dela<10;dela++)
    {
        int offset=widthOfChar*_offset;
        for(int j=0;j<10;j++)
        {
            if((j+_shoveHowMuch)>15 || (j+_shoveHowMuch)<0)
            {
                offset++;
                offset++;
                if((j+_shoveHowMuch)>15) break;
                //Speed optimization
                continue;
            }

            char whichToWrite=Courier10x13[offset];

            activateRow(j + _shoveHowMuch);

            pickPartOfDisplay(false);
            for(int i=0;i<8;i++)
```

```
{
    digitalWrite(6+i,(whichToWrite&(1<<i)));
}
delayMicroseconds(20);

offset++;

whichToWrite=Courier10x13[offset];

activateRow(j + _shoveHowMuch);

pickPartOfDisplay(true);
for(int i=0;i<8;i++)
{
    digitalWrite(6+i,(whichToWrite&(1<<i)));
}
delayMicroseconds(20);

offset++;
}
}
```

nome en ressources du programme, nous pouvons ignorer les avertissements du compilateur. Malheureusement, le résultat n'est pas vraiment utilisable à cause d'un fort scintillement. Pour calmer le jeu, j'ai modifié le contenu de la boucle de la manière suivante :

```
void loop() {
  displayChar('F'-32,0);
  /*for(int i=0;i<90;i++)
  ...
```

À présent, le contrôleur n'est plus surchargé par la représentation des caractères.

La solution consiste à optimiser la routine d'affichage. La nouvelle version est dans le **listage 9**.

Ce qui est intéressant, outre l'éloignement des appels de `delayMicroseconds`, c'est le contrôle effectué par `shoveHowMuch`. La nouvelle version de `displayChar` vérifie la validité des plages de nombres immédiatement après le démarrage et économise le processeur le cas échéant lors de l'exécution des boucles For imbriquées. La boucle extérieure est particulièrement critique en raison de la multiplication qu'elle contient.

Conclusion

L'affichage du petit mot de bienvenue met un point final à cette étape du projet : l'afficheur à matrice de 16x16 LED fonctionne. Pour des projets plus conséquents, il existe de nombreuses possibilités d'optimisation. D'abord choisir un processeur doté de plus de ports d'E/S pour pouvoir afficher simultanément davantage de données.

Un deuxième changement concerne la gestion des caractères. La boucle primitive actuelle gaspille de la puissance de calcul du fait que chaque caractère est affiché indépendamment de la position qu'il occupera.

Enfin, notre programme arrive juste sous la limite de la mémoire disponible dans le processeur AVR utilisé. L'emploi d'instructions PROGMEM (dans la mémoire Flash au lieu de la SRAM) aiderait déjà à ce stade, vous trouverez des informations complémentaires dans la documentation d'Arduino.

L'ensemble du code est à télécharger en [2].

(140520 – version française : Robert Grignard)

Liens

[1] www.mikroe.com/glcd-font-creator/

[2] www.elektormagazine.fr/140520

Listage 8. Un petit coucou

```
void loop() {
  for(int i=0;i<90;i++)
  {
    displayChar(' '-32,0-i);
    displayChar('H'-32,10-i);
    displayChar('A'-32,20-i);
    displayChar('L'-32,30-i);
```

```
    displayChar('L'-32,40-i);
    displayChar('O'-32,50-i);
    displayChar('!'-32,60-i);
    displayChar(' '-32,70-i);
    displayChar(' '-32,80-i);
  }
}
```

Listage 9. Routine améliorée

```
void displayChar(int _offset, int _shoveHowMuch)
{
  if(_shoveHowMuch<-10)return;
  if(_shoveHowMuch>16)return;

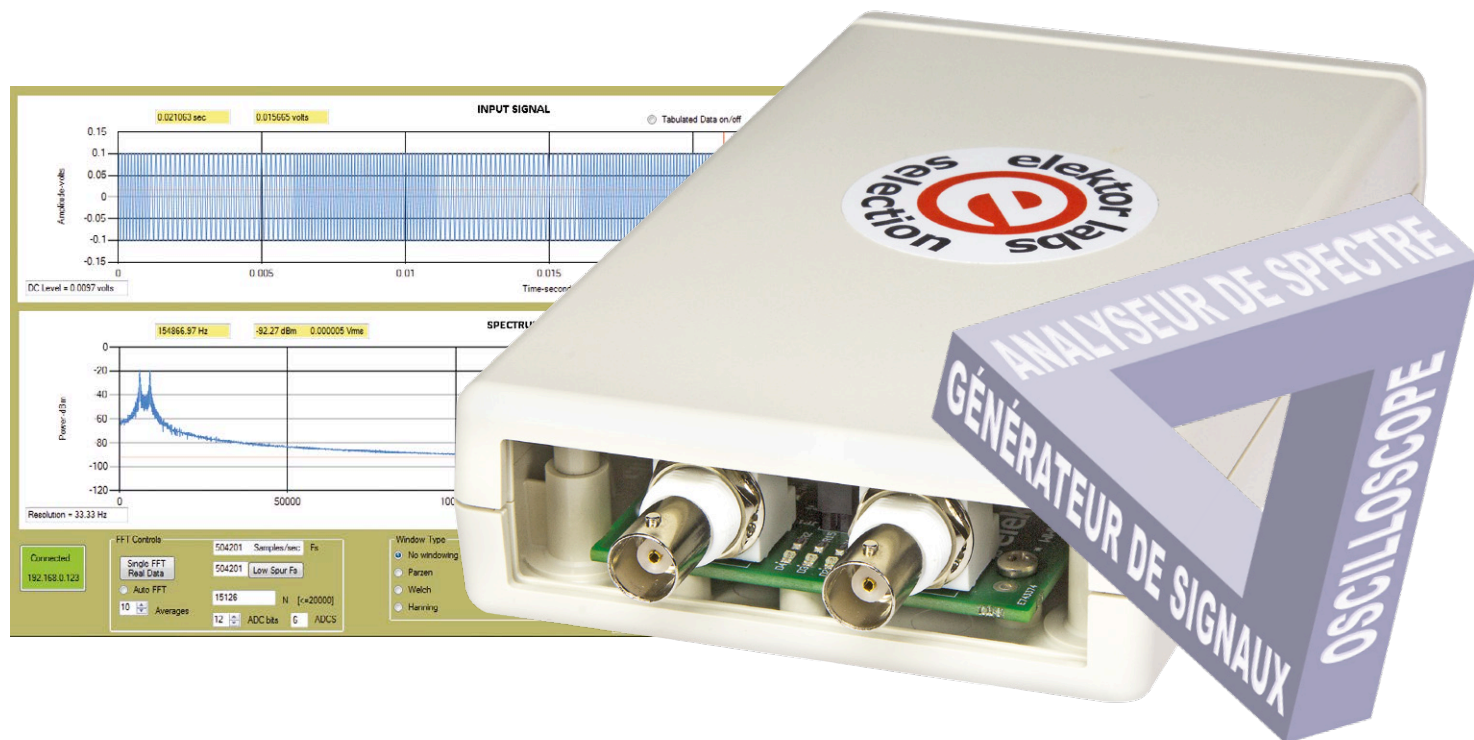
  for(int dela=0;dela<5;dela++)
  {
    int offset=widthOfChar*_offset;
    for(int j=0;j<10;j++)
    {
      if((j+_shoveHowMuch)>15 || (j+_shoveHowMuch)<0)
      {
        offset++;
        offset++;
        if((j+_shoveHowMuch)>15)break;
        //Speed optimization
        continue;
      }
      char whichToWrite=Courier10x13[offset];
      activateRow(j+_shoveHowMuch);
```

```
      pickPartOfDisplay(false);
      for(int i=0;i<8;i++)
      {
        digitalWrite(6+i,(whichToWrite&(1<<i)));
      }
      offset++;

      whichToWrite=Courier10x13[offset];
      activateRow(j+_shoveHowMuch);
      pickPartOfDisplay(true);
      for(int i=0;i<8;i++)
      {
        digitalWrite(6+i,(whichToWrite&(1<<i)));
      }
      offset++;
    }
  }
}
```

analyseur de signal par Ethernet (2)

du code, des maths, du concret



Neal Martini (États-Unis)

L'article précédent [1] décrivait le matériel de l'ASPE, un instrument compact et bon marché qui combine sur une même carte un oscilloscope, un générateur de signal et un analyseur de spectre. Nous nous intéressons ici aux rouages logiciels et mathématiques de l'instrument.

Je présente d'abord le micrologiciel et le programme du PC, puis les principes mathématiques qui vous permettront de comprendre les mécanismes du traitement du signal numérique.

Caractéristiques

- Oscilloscope, analyseur de spectre et générateur de signal, trois-en-un
- Fréquences d'échantillonnage jusqu'à 1 MHz
- Sous-échantillonnage
- Signal d'entrée max. : 0 dBm ($0,225 V_{eff}$)
- Sensibilité : -80 dBm ($22,5 \mu V_{eff}$)
- Connexion Ethernet
- Code source ouvert

Communication par Ethernet

Le microcontrôleur et le PC communiquent par Ethernet grâce au module W5500. Côté contrôleur, ce sont les pilotes *ioLibrary_BSD* [2] qui se chargent de la communication avec ce module. Avant de pouvoir les utiliser, il faut écrire quatre fonctions de commande du périphérique SPI du dsPIC33, puisque c'est par SPI que dialoguent le W5500 et le dsPIC. Voici leur code en C :

```
// Declare W5500 driver SPI
Functions
reg_wizchip_cs_cbfunc(wizchip_select, wizchip_deselect);
reg_wizchip_spi_cbfunc(wizchip_read, wizchip_write);
```

```
//Functions
void wizchip_select(void)
{
    WIZCS = 0;
}

void wizchip_deselect(void)
{
    WIZCS = 1;
}

void wizchip_write(uint8_t wb)
{
    uint8_t dummy;
    SPI1BUF = wb;
    // write to buffer for TX
    while( !SPI1STATbits.SPIRBF );
    // wait for TX complete
    dummy = SPI1BUF;
```

```

}

uint8_t wizchip_read()
{
    SPI1BUF = 0x00;
    // write to buffer for TX
    while( !SPI1STATbits.SPIRBF );
    // wait for TX complete
    return SPI1BUF;
    // read the received values
}

```

Une fois ces fonctions déclarées, la communication entre la puce W5500 et le PC se fait avec les fonctions `recv(sn,buf,size)` et `send(sn,buf,size)` des pilotes `ioLibrary_BSD`.

Côté PC, le code C# qui établit la communication Ethernet et transmet les commandes/données est très simple. Lorsque l'application PC est lancée, le code détecte laquelle des quatre adresses IP possibles est présente sur le réseau local, et l'utilise ensuite pour toutes les communications de la session. Il est donc important de connecter l'analyseur au réseau avant de lancer l'application, puis d'attendre quelques secondes afin que la puce W5500 ait le temps de s'initialiser.

Le programme PC utilise la classe `TcpClient` de Windows pour la gestion proprement dite de la communication. Voici le code C# qui établit la connexion entre le PC et la puce W5500 :

```

var result = client.
    BeginConnect(IPAddress.
        Parse(ipaddress),4000,null,null);
result.AsyncWaitHandle.
    WaitOne(TimeSpan.
        FromSeconds(1));
//timeout if no PCB
if (client.Connected)
{
    clientStream = client.
        GetStream();
    //get a client stream
}

```

Une fois la connexion établie, les fonctions `clientStream.Write(TxBuff,offset,size)` et `clientStream.Read(RxBuff,offset,size)` se chargent de l'échange de commandes et de données entre le PC et la puce W5500.

Générateurs de signaux

L'ASPE possède deux générateurs de signaux séparés. Celui du μC fournit un

signal analogique produit par synthèse numérique directe (*Direct Digital Synthesis*, DDS). Vous pouvez l'utiliser comme source de signal pour une application externe, ou l'envoyer à l'instrument à des fins d'analyse. Ce générateur a surtout été inclus pour vous permettre d'expérimenter avec l'ASPE, autrement dit il ne saurait remplacer un générateur de signal de laboratoire.

L'autre est le générateur de signal synthétique du programme PC. Il permet de créer des tableaux de nombres représentant divers types de signaux que l'on peut visualiser et auxquels on peut appliquer une transformation de Fourier.

Examinons d'abord brièvement le générateur du μC (**fig. 1**). L'association du modulateur de largeur d'impulsion (MLI) et du filtre passe-bas (FPB) forme un convertisseur analogique/numérique. Si le rapport cyclique du MLI est par ex. maintenu constant, la sortie du FPB sera une valeur CC. Si le rapport cyclique varie suivant l'amplitude du signal souhaité, la sortie du FPB produira ce signal.

La fréquence MLI du dsPIC33 est :

$$f_{\text{sys}} / 256 = 120 \times 10^6 / 256 = 468\,750 \text{ Hz}$$

Cette fréquence est également la fréquence d'actualisation de la DDS. Une fois par cycle MLI, nous calculons l'amplitude de la forme d'onde souhaitée, puis nous définissons le rapport cyclique qui donnera cette amplitude. Pour trouver la valeur de ce rapport, nous utilisons un tableau de valeurs représentant un cycle de l'onde. Nous y sélectionnons alors simplement l'amplitude qui correspond à l'endroit où l'onde souhaitée

serait au moment où la DDS est mise à jour, et actualisons le rapport cyclique en conséquence. Un accumulateur de phase de 32 bits (DDSp dans le code) mémorise la phase de l'onde souhaitée. Une fois par cycle d'actualisation de la DDS, cet accumulateur est incrémenté par un incrément de phase (DDSD dans le code). Sa valeur dépend de la fréquence de l'onde souhaitée selon la relation :

$$\text{incrément de phase} = 2^{32} \times \text{fréquence}_\text{souhaitée} / \text{taux d'actualisation DDS}$$

Pour un sinus de 10 000 Hz par ex., l'incrément de phase vaut 91 625 698. Cette valeur est ajoutée au contenu de l'accumulateur de phase à la même fréquence que celle de l'actualisation de la DDS. Par conséquent l'accumulateur suit à tout moment et précisément la valeur de la phase de l'onde souhaitée. Comme un tableau de 2^{32} entrées occuperait trop de place en RAM, nous ne prenons que les huit premiers bits de l'accumulateur à 32 bits et les utilisons pour l'adressage du tableau qui couvre une période du signal.

Le code source de l'ASPE permet de choisir entre sinus, ondes carrée et triangulaire, et de sélectionner une fréquence. Avec un peu d'effort (et d'ambition) vous devriez être en mesure d'y ajouter d'autres ondes.

À propos de fréquence, la fréquence de coupure du filtre passe-bas de la sortie MLI doit être choisie avec soin : suffisamment basse pour lisser les impulsions MLI (ondulation), mais suffisamment haute pour laisser passer le signal produit. L'uti-

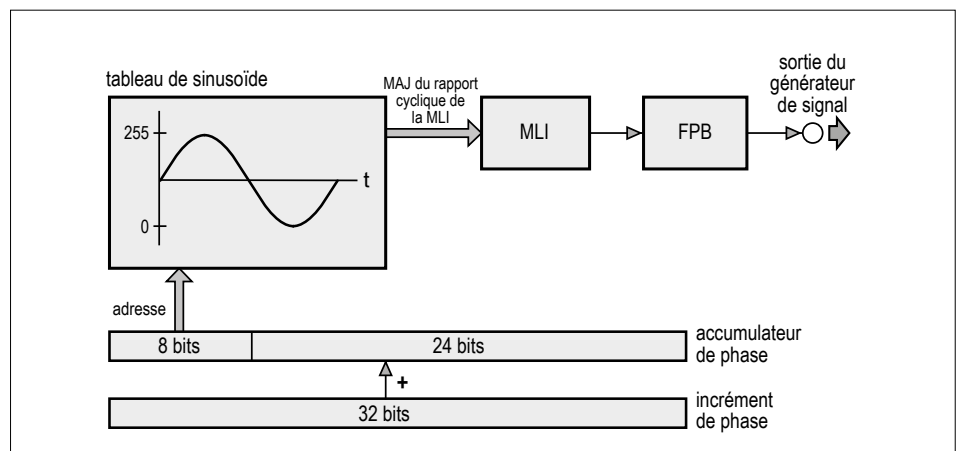


Figure 1. Diagramme fonctionnel du générateur de signal à synthèse numérique directe.

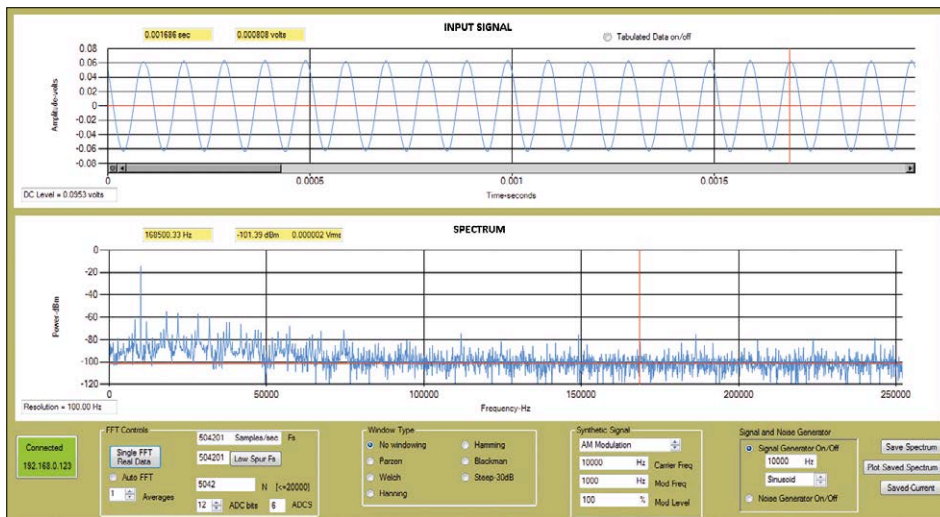


Figure 2. L'interface du programme lorsque le générateur de signal de l'ASPE est paramétré pour produire une sinusoïde de 10 kHz. La sortie du générateur est reliée à l'entrée de l'analyseur.

litaine en ligne du lien [3] est à cet égard bien utile. Le rôle principal du filtre est de maintenir l'ondulation à un niveau faible. On voit clairement le lissage qu'il produit avec une onde carrée : plus on augmente la fréquence, plus les bords s'arrondissent.

La **figure 2** montre l'interface PC lorsque le générateur de signal de l'ASPE est paramétré pour produire une sinusoïde de 10 kHz. La sortie du générateur est ici reliée à l'entrée de l'analyseur. La fenêtre du haut affiche la sinusoïde échantillonnée, et sa forme est plutôt satisfaisante. On peut se faire une idée plus exacte de sa qualité en examinant la fenêtre *Spectrum*. On distingue le pic du signal

à 10 kHz, mais aussi des composantes parasites. Ces distorsions sont environ 40 dB sous le signal souhaité, ce qui pour un générateur aussi simple peut être considéré comme acceptable. Cela dit ce sont bien des distorsions.

Voici les deux blocs de fonctions principaux du générateur DDS :

```
// Generate a lookup table; one
// cycle of the waveform.
for (i=0; i<256; i++)
{
    Switch (RxBuff[7]) // Selects
    the type of signal to generate.
    {
        case 1: Wave[i] = (uint8_t)
```

```
(128+127*sin((PI*i)/128)+.5);
//sine
        break;
        case 2: Wave[i] = (i<128) ?
255 : 0; // one cycle square
        break;
        case 3: Wave[i] =
abs((i % 256) - 128);
//one cycle triangle
        break;
    }
}
```

```
// Calculate the phase increment
DDSD = ((FrequencyDesired *
pow(2,32))/468750.0) + .5;
```

```
// Timer1 interrupt service routine
void __attribute__((__interrupt__,
no_auto_psv)) _T1Interrupt(void)
{
    PDC1 = Wave[(DDSp>>24)];
    //do table lookup using
    upper 8 bits of DDSp
    DDSp += DDSd;
    //increment phase accumulator
    IFS0bits.T1IF = 0;
    //Clear Timer1 interrupt flag
}
```

De son côté, le générateur de signal synthétique de l'application PC offre quatre types de signaux, définis par la fonction *GenerateData* du fichier *myFFTWstuff.cs*. Là encore vous pouvez modifier le code de ce fichier pour y définir des ondes de forme arbitraire.

Notez que l'onde carrée synthétique (case 2 ci-dessus) est volontairement « non idéale ». Elle est produite par sommation de cinq sinusoïdes de façon à ce que vous puissiez observer les effets de cette approximation sur les représentations temporelle et fréquentielle du signal.

Transformation de Fourier

Je vais maintenant tenter d'expliquer la transformation de Fourier, une méthode inventée par Joseph Fourier pour recréer des formes d'onde par sommation de diverses fonctions sinusoïdales (trigonométriques). La transformation de Fourier permet de calculer les amplitudes et phases de ces sinusoïdes, autrement dit décompose un signal en composantes sinusoïdales. Son équivalent pour un ensemble de données échantillonnées est la transformation de Fourier discrète (TFD). Voici l'équation utilisée pour sa mise en œuvre algorithmique :

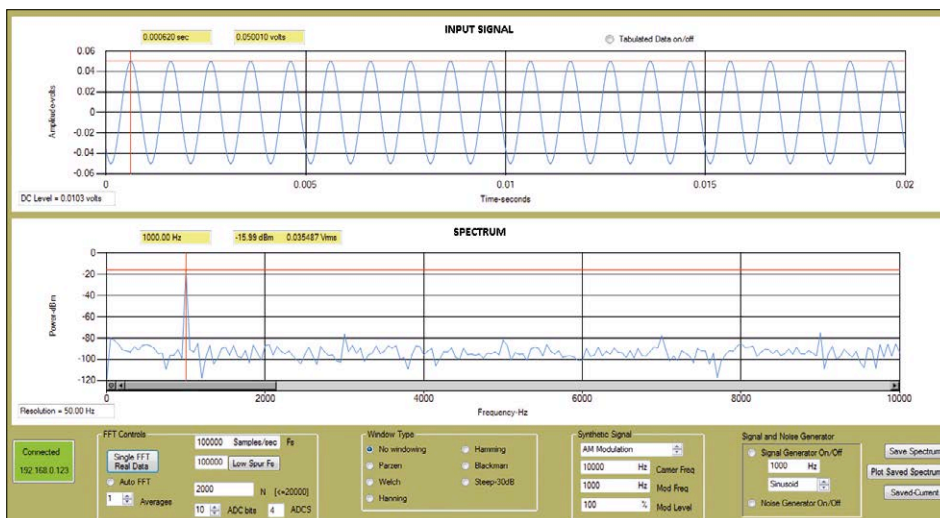


Figure 3. Le générateur de signal de l'ASPE a produit un sinus de 1 kHz. Son spectre est calculé et affiché par l'application PC.

$Spectre(m) =$

$$\sum_{n=0}^{N-1} x(n) \cos\left(\frac{2\pi nm}{N}\right) - j \sum_{n=0}^{N-1} x(n) \sin\left(\frac{2\pi nm}{N}\right)$$

$x(n)$ est la valeur de la n -ième donnée échantillonnée ;

N est le nombre total d'échantillons ;

n est le numéro de l'échantillon considéré ;

m correspond à la position dans le domaine fréquentiel.

Pour chaque valeur de m , l'équation renvoie un nombre complexe de la forme $(a + jb)$. Si vous regardez attentivement les termes de l'équation, vous verrez que a est la somme des N produits de l'échantillon $x(n)$ par le cosinus d'une fréquence définie par la valeur de m . Idem pour b , mais avec un sinus. Plus la valeur de l'échantillon $x(n)$ est proche du cosinus/sinus de la fréquence déterminée par m , plus la somme est grande. Plus la valeur $x(n)$ s'écarte, plus la somme est petite. Le cosinus/sinus de plus basse fréquence que l'équation utilise est une sinusoïde dont le cycle couvre la période d'échantillonnage. Toutes les sinusoïdes suivantes de la TFD sont des multiples entiers de cette fréquence fondamentale. Prenons un exemple concret pour mieux comprendre. Supposons que la fréquence d'échantillonnage f_s est de 5000 éch/s et que $N = 1000$ points de données. Ces points représentent donc le signal sur une période $T = 1000/5000 = 0,2$ s. La première sinusoïde qu'utilise la TFD effectue un cycle en T s, soit ici $1/0,2 = 5$ Hz. Il s'agit de la résolution du spectre de sortie de la TFD. Si vous effectuez la TFD sur $N = 1000$ points, vous obtenez 1000 nombres complexes. En pratique, avec des données réelles, les premiers $N/2$ points sont les mêmes que les derniers $N/2$ points, donc la partie non redondante du spectre de sortie contient seulement $N/2$ points. Dans ce cas, la plage de fréquences du spectre de sortie est :

$$(N/2) \times \text{résolution} = 500 \times 5 \\ = 2\,500 \text{ Hz}$$

soit $f_s/2$.

L'application de la TFD nous donne donc $N/2$ nombres complexes. À quoi servent ces nombres ? À calculer les niveaux de puissance des différentes composantes sinusoïdales dont a besoin l'analyseur. Le nombre $(a + jb)$ représente sous forme complexe l'amplitude et la phase de chaque sinusoïde. Puisque nous mesurons cette amplitude via une résistance de $50 \, \Omega$ placée à l'entrée de l'ASPE, la puissance de la sinusoïde est :

$$P_{\text{eff}}(m) = (a^2 + b^2) / 50$$

Cette puissance est convertie en dBm, l'unité standard employée par la plupart des analyseurs de spectre. P_{dBm} exprime la puissance mesurée par rapport à un milliwatt. Elle se calcule ainsi :

$$P_{\text{dBm}} = 10 \log_{10} P_{\text{eff}} / 0,001$$

En plus de la valeur en dBm, la valeur efficace et la fréquence correspondantes sont affichées au-dessus du spectre, pour la position du curseur.

La **figure 3** illustre le processus complet de la transformation, depuis l'amplification de l'étage d'entrée jusqu'à la conversion de la puissance en dBm. L'entrée est un sinus de 1 kHz et 100 mV_{cc}, échantillonné par le CA/N de 10 bits à une fréquence f_s de 100 kHz, par blocs de $N = 2000$ échantillons. Comme expliqué ci-dessus, cela correspond à une fenêtre de temps $T = N/f_s = 20$ ms et à une plage

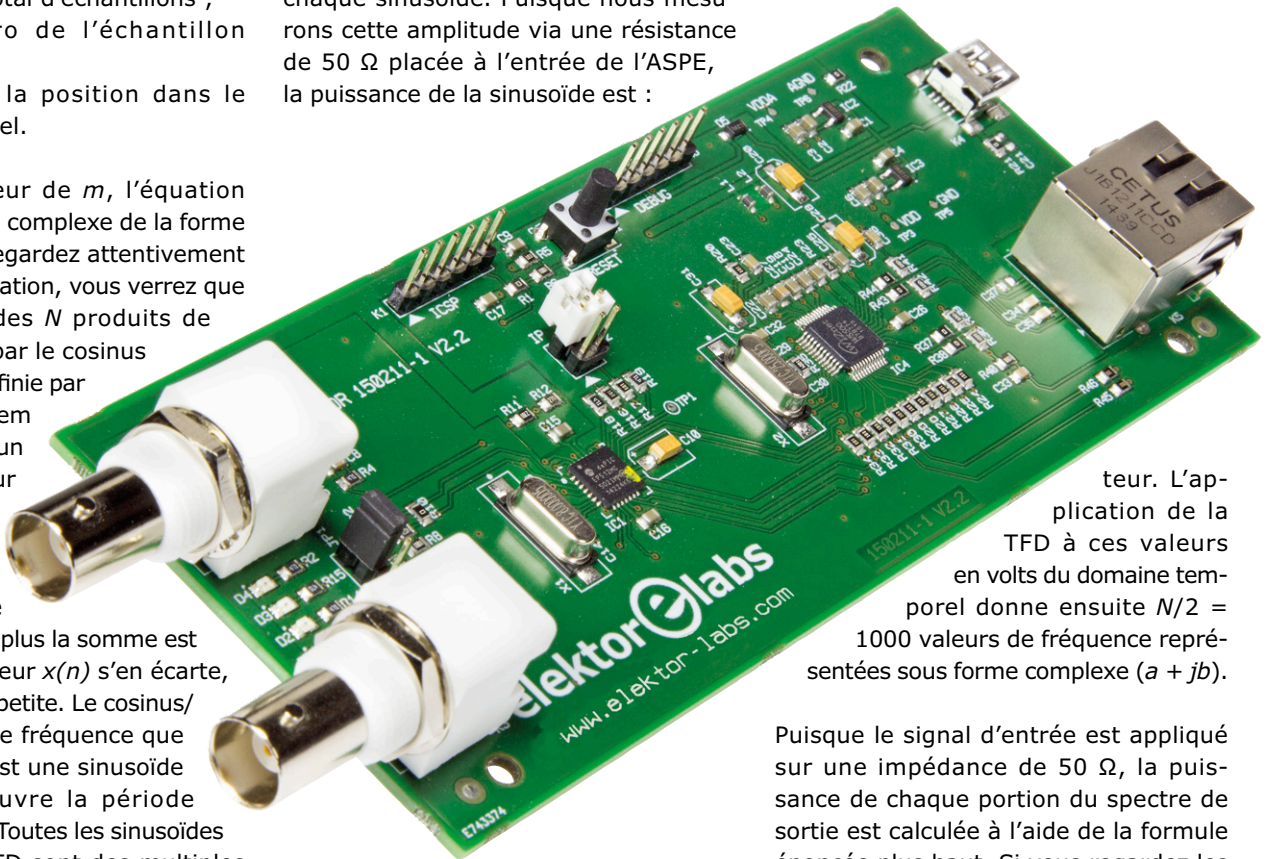
de fréquences $f_s/2 = 50$ kHz (la fig. 3 est en mode zoom et ne couvre que l'intervalle 0 à 10 kHz). Après la conversion A/N, le niveau CC est supprimé et les échantillons sont convertis en volts par mise à l'échelle (*scaling*) du nombre de bits et compensation du gain de l'amplifica-

teur. L'application de la TFD à ces valeurs en volts du domaine temporel donne ensuite $N/2 = 1000$ valeurs de fréquence représentées sous forme complexe $(a + jb)$.

Puisque le signal d'entrée est appliqué sur une impédance de $50 \, \Omega$, la puissance de chaque portion du spectre de sortie est calculée à l'aide de la formule énoncée plus haut. Si vous regardez les valeurs données par la position du curseur sur la courbe, vous verrez que la puissance calculée par TFD pour la portion de spectre à 1000 Hz est de $-15,99$ dBm (35,487 mV_{eff}), comme attendu.

Ce que j'ai appelé jusqu'ici la TFD du programme PC est en fait une transformation de Fourier rapide (FFT, *Fast Fourier Transform*), un algorithme de calcul très puissant de la transformation de Fourier discrète. Cette méthode élaborée en 1965 par Cooley et Tukey réduit fortement le nombre de calculs, car elle évite les multiplications redondantes qui apparaissent lors de l'évaluation des sommes de l'équation TFD. C'est une TFD, mais beaucoup plus efficace.

Les FFT sont calculées par l'application PC. Le microcontrôleur aurait pu s'en charger, mais il l'aurait fait plus lentement. J'ai utilisé les routines d'une bibliothèque FFT datant de 1999 (voir www.fftw.org). Elles étaient en C, mais j'en



Fenêtrage

Certaines TFD ont besoin d'un fenêtrage. Appuyons-nous sur un exemple pour le comprendre. La **figure 4** montre le spectre d'un sinus qui effectue exactement dix cycles durant la période d'échantillonnage. La **figure 5** montre un signal qui effectue un peu plus de dix cycles durant la même période d'échantillonnage. La différence entre les deux spectres

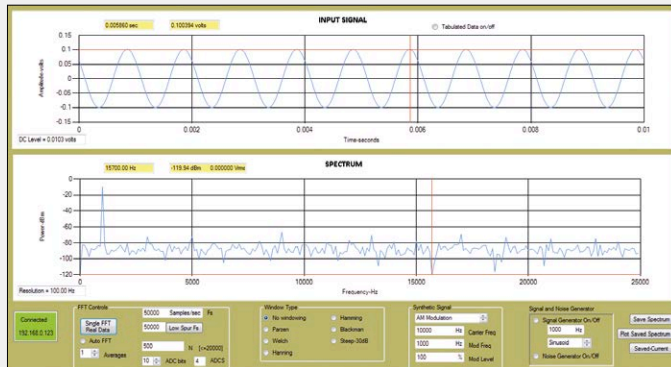


Figure 4. Le signal de 1000 Hz analysé ici effectue un nombre entier de cycles pendant la durée d'échantillonnage. Notez que les premières valeurs échantillonnées et les dernières sont à peu près les mêmes.

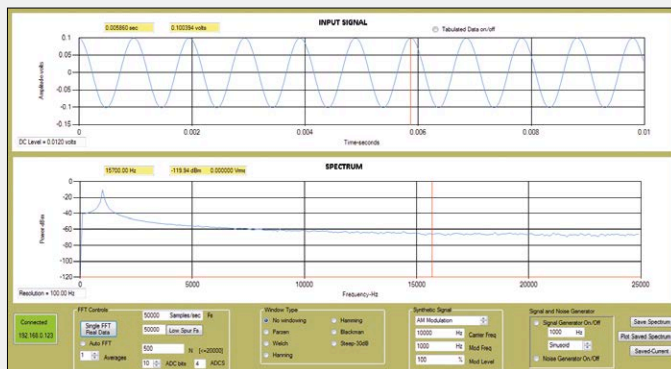


Figure 5. Voici ce qui arrive lorsqu'il n'y a pas un nombre entier de cycles pour le signal analysé (ici un sinus de 1020 Hz). On voit que les premiers et derniers échantillons sont très différents. Le pic principal est étalé et masque le domaine qui l'entoure. Le reste du spectre a également des niveaux plus hauts pouvant masquer des signaux.

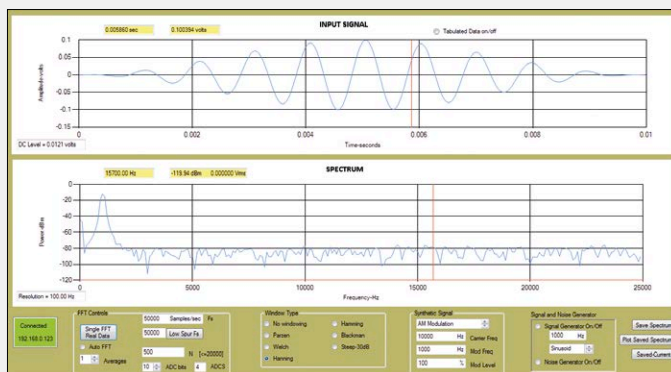


Figure 7. Après l'application d'un fenêtrage de Hanning au sinus de 1020 Hz de la fig. 5, on voit clairement que le spectre est meilleur.

est spectaculaire. Notez, sur la fig. 5, les niveaux élevés qui n'appartiennent pas au signal, ainsi que l'étalement du pic principal. Ces niveaux de fond élevés (appelés « fuite spectrale ») peuvent masquer des signaux du spectre. Pourquoi cette fuite ? La TFD tente de reproduire les données échantillonnées sur une période T d'échantillonnage, donc d'une certaine façon l'algorithme suppose qu'il travaille avec une période du signal qui se répète toutes les T secondes. Lorsque le début et la fin des données échantillonnées ne présentent pas tout à fait les mêmes niveaux, cette différence provoque la production artificielle de composantes à hautes fréquences dont l'injection dans le spectre entraîne des distorsions spectrales. On pourrait jouer avec la fréquence d'échantillonnage jusqu'à obtenir un nombre entier de cycles dans le signal testé, mais évidemment on ne sait pas toujours à quoi doit ressembler le signal. C'est là que le fenêtrage entre en jeu.

Pour adoucir cette discontinuité, le fenêtrage abaisse les amplitudes des échantillons au début et à la fin du bloc d'échantillons. Pour cela les valeurs échantillonnées sont multipliées par une fonction (fenêtre) qui atténue les amplitudes des échantillons de début et de fin. La **figure 6** montre la fenêtre de Hanning de 500 points utilisée pour cet exemple. Si vous multipliez le bloc des données d'entrée de la

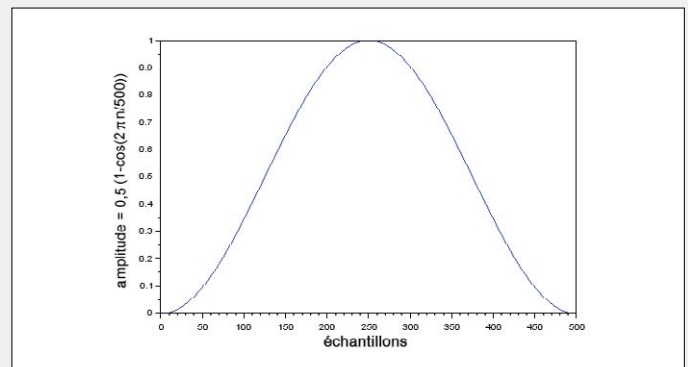


Figure 6. Le fenêtrage de Hanning.

figure 5 par la fenêtre de Hanning, vous obtenez les données « fenêtrées » du haut de la **figure 7**. Notez l'atténuation des discontinuités du début et de la fin du bloc de données. Notez aussi l'amélioration apportée au spectre (fenêtre *Spectrum* de la fig. 7). La fuite spectrale est beaucoup moins intense. Le fenêtrage a cependant un coût : il produit un certain étalement spectral, notamment au niveau du pic du signal. La puissance totale reste la même, mais elle est répartie sur quelques portions de spectre adjacentes. Cet étalement n'est généralement pas un problème comparé aux bénéfices qu'apporte le fenêtrage.

Le programme possède six fonctions de fenêtrage. Chacune agit différemment sur l'étalement spectral et sur le nombre de composantes à hautes fréquences qu'elle laisse passer. De nombreuses ressources en ligne exposent les avantages et inconvénients de ces fenêtres, mais il est bien plus amusant de découvrir leurs effets par soi-même.

ai heureusement trouvé une implantation en C# [4], ce qui m'a permis d'extraire les blocs de code nécessaires au programme PC.

La vitesse de traitement de l'ASPE dépend de la taille N des blocs d'échantillons. Un N élevé offre une meilleure résolution spectrale, un N petit un traitement plus rapide. Le tableau suivant donne une idée de la relation entre N et la vitesse (sur mon ordinateur portable). La fréquence d'actualisation comprend toutes les étapes : échantillonnage, transfert des données, fenêtrage, FFT et tracé des courbes.

N	fréquence d'actualisation (Hz)
5 000	8,33
10 000	3,96
15 000	2,48
20 000	1,70

Pour finir, voici le code C#, le cœur de l'application PC, qui montre comment sont transférées les données de l'ASPE ainsi que les étapes du traitement numérique :

```
// Receive data over the network
int read = 0, offset = 0,
toRead = 2*N;
```

```
while (toRead>0 &&
(read = clientStream.
Read(RxBuff,offset,toRead))>0)
{
    toRead -= read;
    offset += read;
}
```

```
DCterm = myFunctions.DCtermCalc(N,
    RxBuff, ScaleFactor);
//calculates DC term
```

```
// Subtracts DC, scales for ADC
bits, puts data in real part
of din[]
```

```
myFunctions.Filldin(N, DCterm,
    RxBuff, ScaleFactor, AmpGain,
    din);
```

```
myFunctions.windowing(N, din,
    WindowType, 1, din);
//window data
plotTime.PerformClick();
//plot input time series
fftwf.execute(fplan); //Do FFT
```

```
//convert to dBm and average
spectra (code not shown here)
```

```
plotFreq.PerformClick(); // Plot
frequency domain data (spectrum)
```

Conclusion

L'instrument est séduisant par bien des ASPECTS : une interface interactive, deux générateurs de signaux, la possibilité d'ajouter des formes d'onde arbitraires, un code ouvert qui permet d'adapter le système à une application particulière, des représentations en temps et en fréquence pour bien comprendre les signaux analysés, et de nombreux paramètres de configuration. Autant d'atouts qui développeront chez l'utilisateur un sens pratique de la numérisation et de la transformation de Fourier des signaux... tout en s'amusant ! ◀

(150694 - version française : Hervé Moreau)

Liens

- [1] www.elektormagazine.fr/150211
- [2] <http://wizwiki.net/wiki/doku.php?id=products:w5500:driver>
- [3] <http://sim.okawa-denshi.jp/en/PWMtool.php>
- [4] <https://github.com/tszalay/FFTWSHarp>
- [4] <https://github.com/tszalay/FFTWSHarp>

Sous-échantillonnage

Le sous-échantillonnage mérite quelques explications puisqu'il augmente considérablement le potentiel de l'analyseur. Si on envoie à l'entrée de l'ASPE un signal sinusoïdal de 10 kHz produit par un générateur de signal du commerce, et si on règle la fréquence d'échantillonnage de l'ASPE sur 500 kHz, on obtient un ensemble d'échantillons. Dans cet environnement à faible bruit, si l'on répète l'opération avec le générateur réglé sur 510 kHz, 1010 kHz, 1510 kHz et ainsi de suite, on obtient le même ensemble d'échantillons. Pourquoi ? Parce que l'échantillonnage de ces signaux se fait au même instant du cycle où était échantillonné le signal de 10 kHz. Gardez toutefois à l'esprit que des fréquences de 490 kHz, 990 kHz ou 1490 kHz (dites fréquences-images) donneraient un ensemble d'échantillons semblable à celui obtenu avec le signal de 10 kHz.

Quand le bruit augmente dans le système, le bruit dans le spectre sous-échantillonné augmente aussi. S'il y a un nombre significatif de composantes fréquentielles là où sont les fréquences-images, le spectre subit une distorsion. Ces limitations n'empêchent pas l'utilisation du sous-échantillonnage pour voir des signaux situés bien au-delà de la fréquence f_s de l'analyseur. C'est la technique que j'ai utilisée pour analyser avec une fréquence d'échantillonnage de 504 201 Hz le signal de 1 140 kHz d'une station radio AM

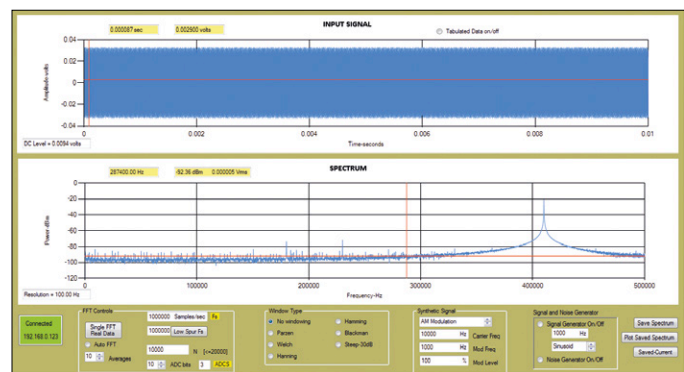


Figure 8. Le signal analysé ici est une sinusoïdale de 4,41 MHz. Le sous-échantillonnage fait apparaître le signal à 410 kHz. Cette technique permet de visualiser des signaux de fréquences bien supérieures à la fréquence d'échantillonnage maximale de 1 MHz de l'analyseur.

(fig. 7 de l'article précédent [1]). Sur le spectre, le pic apparaît à $1\,140\,000 - 2 \times 504\,201 = 131\,598$ Hz. La seule limitation matérielle est la largeur de bande permise par l'amplificateur d'entrée. Le produit gain-bande de cet amplificateur est de 6 MHz. Si une amplitude réduite ne vous gêne pas, vous pouvez même analyser des signaux de plus de 1 MHz (fig. 8). C'est sympathique !

WLAN compact et autonome

ou comment utiliser la puce ESP8266 sans μ C

Walter Trojan (Allemagne)

Introduire un microcontrôleur (μ C) dans un réseau local sans fil (WLAN) pour le faire entrer dans le monde de l'Internet des Objets ? Rien de plus facile aujourd'hui ! De nombreuses cartes ou puces le permettent. Toutefois il est possible de remplacer ces solutions combinées par un circuit plus compact sur lequel la puce WLAN remplit des tâches typiquement réservées à un μ C.

Dans le numéro de janvier/février 2016 [1], nous avons vu comment la puce ESP8266 pouvait être intégrée à des circuits avec μ C et pilotée avec des commandes AT. Toutefois elle sait faire bien plus. En effet cette puce comprend, outre sa circuiterie spécialisée WLAN, un puissant processeur à 32 bits cadencé à 80 MHz, voire 160 MHz. À en croire le fabricant, les fonctions WLAN ne consomment que 20% de ses capacités. Comment résister à l'idée d'utiliser intelligemment les 80% restants pour d'autres activités ?

Un coup d'œil au diagramme fonctionnel (fig. 1) montre clairement la frontière entre la technologie analogique à haute fréquence, un émetteur-récepteur à 2,4 GHz avec adaptateur d'antenne et les générateurs d'horloge incontournables, et la commande numérique à base de MCU L106 de Tensilica. Ce μ C contient, outre une mémoire RAM de 32 + 80 Ko, des périphériques utiles, accessibles de l'extérieur par le biais de 17 lignes. Selon l'application, elles peuvent être commutées en E/S à usage général (GPIO), ou encore en interfaces UART,

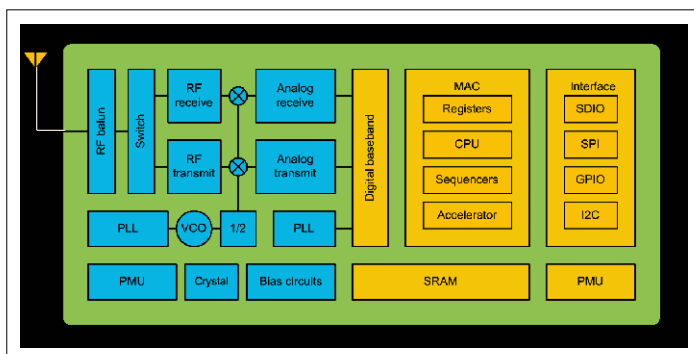
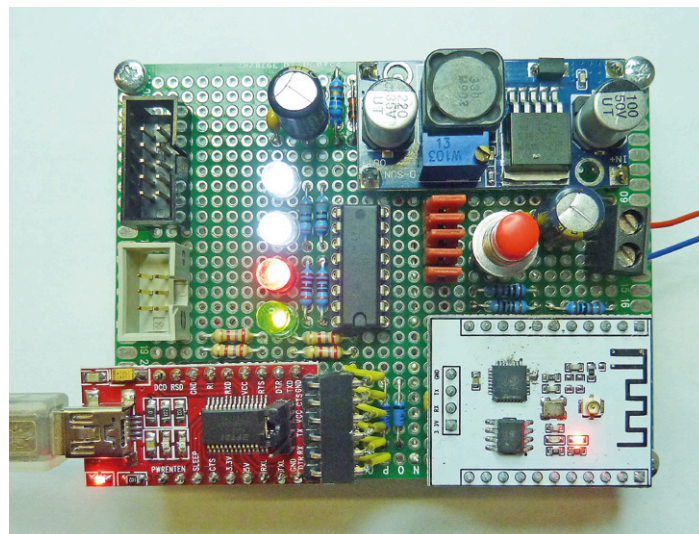


Figure 1. Les entrailles de la puce ESP.



SPI, I²C ou MLI. Le micrologiciel se trouve dans une mémoire Flash externe, pilotée par une interface SDIO puissante (quatre lignes de données, deux de commande). La consommation de courant est de ± 70 mA en fonctionnement normal, d'au maximum ± 170 mA ; en mode veille, elle est abaissée à 10 μ A env. par les unités de gestion de l'alimentation (PMU). Au vu de ces caractéristiques, l'ESP8266 est utilisable de manière autonome pour des projets WLAN, petits et moyens.

Tous les chemins mènent... au micrologiciel

Espressif, le fabricant de la puce, propose un système de développement, qui tourne sous une machine virtuelle Linux, Ubuntu. Comme ce n'est pas l'environnement favori de nombre d'utilisateurs, la communauté Maker a ouvert de nouvelles voies pour le développement avec la puce ESP8266. Les trois les plus populaires sont :

Lua – langage de script d'Amérique du Sud

Lua (Lune en portugais) est un langage de script impératif et extensible [2], exécuté par un interprète très compact (120 Ko) indépendant de la plateforme. Il peut accéder aux bibliothèques en C ; il est également possible, depuis du code en C, d'appeler des routines Lua.

Si l'on flashe l'ESP8266 avec l'interpréteur Lua appelé NodeMCU [3], le préprocesseur AT d'origine est écrasé. À partir de là, la puce n'accepte plus que des instructions Lua et les exécute. Il existe même des cartes spéciales, qui outre une puce ESP, comportent aussi un convertisseur UART/USB et sont préchargées avec ce micrologiciel. Avec ESPlorer [4], on dispose d'un EDI confortable pour l'édition et l'exécution de scripts Lua.

La **figure 2** propose une capture d'écran : on y voit, dans la

fenêtre de gauche, un script Lua, et à droite le dialogue avec l'ESP. Sur l'exemple présenté (clignotement de LED), on voit la simplicité et la clarté de la structure de Lua. L'inconvénient de cette approche est la nécessité d'apprendre un nouveau langage de programmation et l'occupation par le micrologiciel NodeMCU de quelques ressources de l'ESP.

IDE Arduino

Oui, même la puissante communauté Arduino a répondu à l'appel ; elle a intégré l'ESP8266 dans l'EDI d'Arduino. L'ajout de cette extension est rapide [5] : sous « Additional Boards Manager URLs », on spécifie l'adresse de l'extension pour EDI [6]. Ensuite le « Boards Manager » propose l'installation du « Generic ESP8266 Module ». Après leur installation, les modules ESP supplémentaires apparaissent dans la liste des cartes disponibles.

Sélectionnez le « Generic ESP8266 Module ». Vous pouvez alors commencer le développement du micrologiciel. En cas d'utilisation de version la plus récente de l'EDI, il peut arriver que les bibliothèques de l'extension pour EDI ne soient pas encore adaptées. Il faudra alors installer la version précédente (v1.6.5 à la place de la v1.6.6 par ex.).

Un document numérique gratuit de 300 pages, rédigé par l'« activiste ESP » Neil Kolban [7], décrit fort bien le déroulement de cette intégration. Cet ouvrage donne aussi de nombreuses informations essentielles et décrit les spécificités de l'ESP8266. Chaudement recommandé !

Arduino épaulé son *shield* WLAN avec une puissante bibliothèque de plus de cent fonctions. Maintenant ces routines sont aussi disponibles pour l'ESP. La **figure 3** montre une capture d'écran du célèbre EDI Arduino avec, dans la fenêtre principale, une portion de croquis d'un serveur Web. Avec près de cent commandes, on doit pouvoir réaliser un simple serveur web pour commander quelques LED. La **figure 4** montre la fenêtre de navigateur avec ce qu'on peut faire.

Eclipse et C

Grâce à un greffon (*plugin*), les amateurs de l'EDI Eclipse peuvent programmer des croquis dans l'EDI d'Arduino (intégration : cf. Neil Kolban). Si vous voulez être le maître de votre micrologiciel et profiter des subtilités de l'ESP8266-SDK, utilisez la version professionnelle d'Eclipse et programmez en C. Eclipse est un EDI à code source ouvert, mis à la disposition de la communauté *open source* par IBM en 2001. Conçu initialement comme plateforme de développement pour Java, cet environnement a été, depuis lors, enrichi de nombreuses *perspectives* (approches spécialisées) pour différents langages de programmation.

Pour C et C++, il existe CDT (*C Development Tools*). Autres avantages : Eclipse tourne sous Linux et Windows, et son développement se poursuit. Depuis 2006, il en existe dix versions. La version Luna, la plus récente date de 2015 et baptisée Mars, est très répandue.

Des plates-formes de développement de renom telles qu'Atollic TrueSTUDIO ou OpenSTM utilisent Eclipse comme EDI. Pour la communauté ESP, l'utilisateur Cherts a mis au point un paquet pour Windows qui contient les éléments suivants :

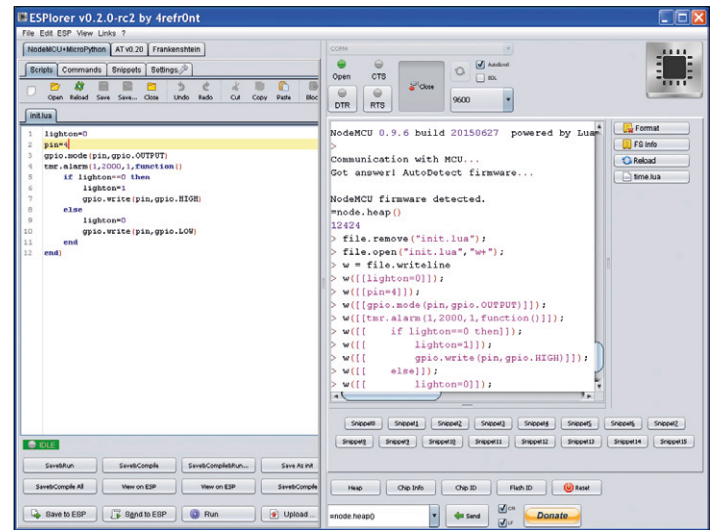


Figure 2. ESPlorer : dans la fenêtre de gauche le script Lua, dans celle de droite le dialogue avec l'ESP.

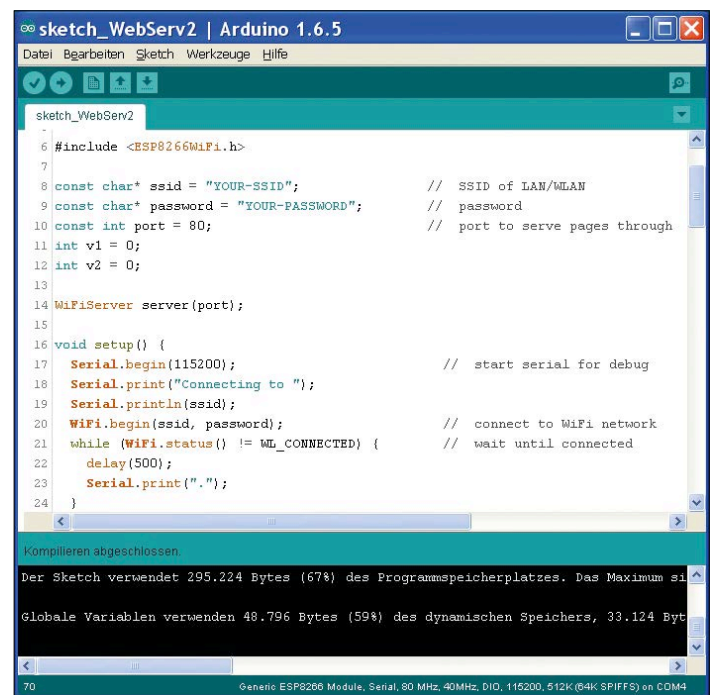


Figure 3. Un serveur web rustique dans l'EDI Arduino.



Figure 4. Voilà à quoi ressemble le « Simple Webserver ».

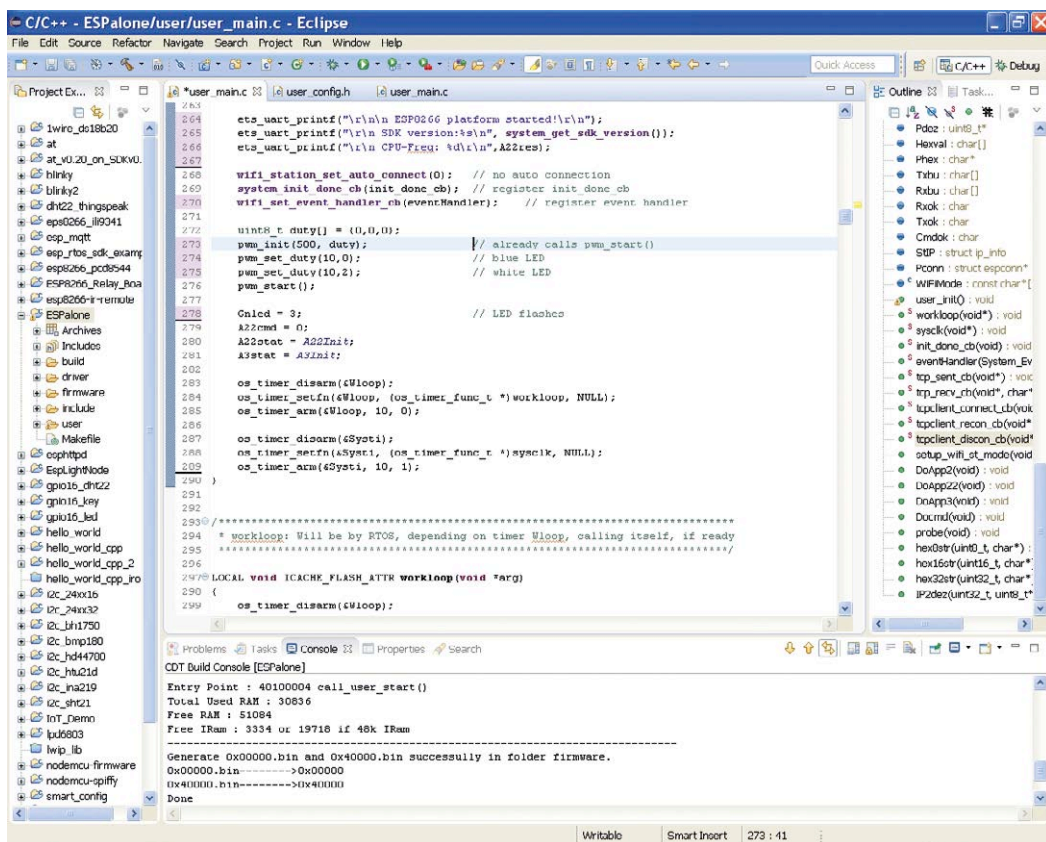
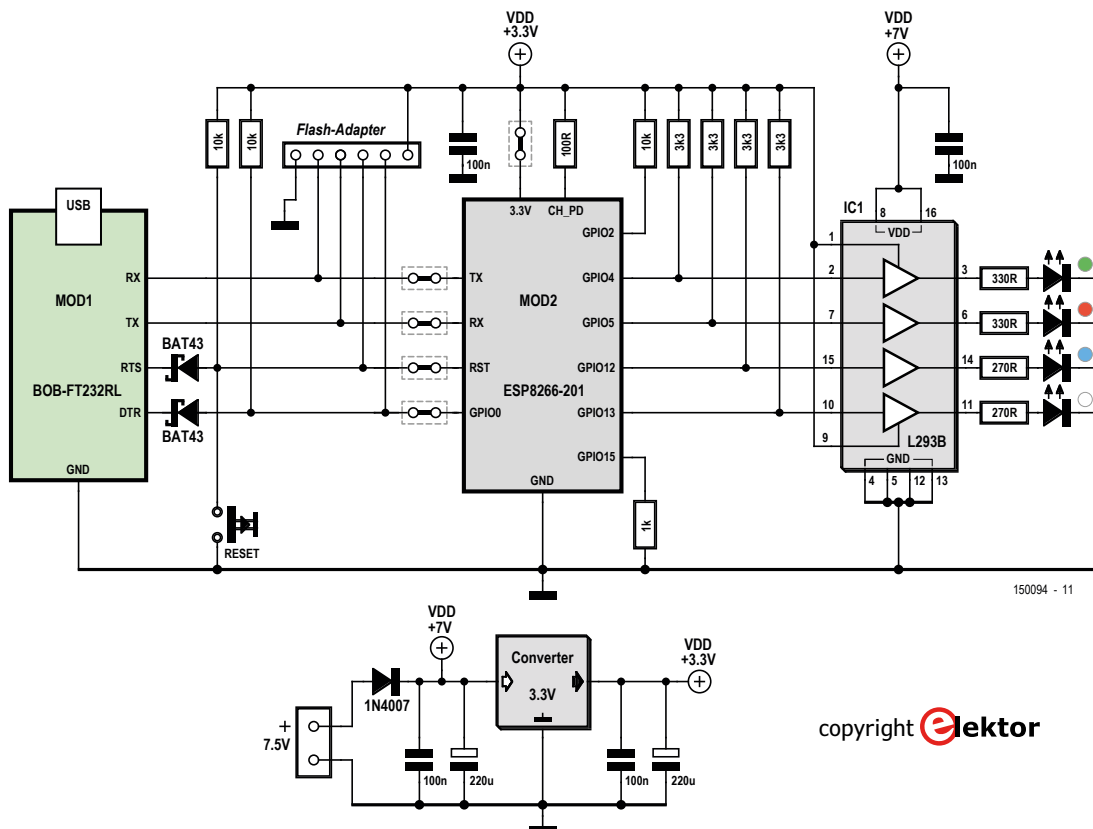


Figure 5. Un écran d'Eclipse bien garni avec de nombreuses informations et options.



copyright elektor

Figure 6. Matériel nécessaire pour faire clignoter les LED.

- ESP8266 *System Development Kit* (SDK) avec documentation et de nombreux exemples
- EDI Eclipse Mars
- Traitement de paquet et compilateur MinGW
- Script pour l'installation automatique de paquets additionnels

On trouvera des instructions et des adresses de téléchargement sous [8]. Après son installation, Eclipse se présente comme en **figure 5**. À gauche dans la fenêtre de projet sont répertoriés les exemples et les projets de l'utilisateur disponibles, la fenêtre centrale affiche le code en C. Via « Outline » dans la fenêtre de droite, on peut accéder rapidement aux variables et fonctions du programme. Dans la fenêtre du bas, on voit s'afficher, entre autres, des messages de la console, d'erreur et les résultats de recherche. Des outils supplémentaires tels que le flashage de l'ESP depuis Eclipse sont apportés par les fichiers Make intégrés aux projets.

Comme j'utilise déjà Eclipse pour le développement avec des µC STM32, j'y suis aussi à l'aise pour la programmation de l'ESP et je l'ai utilisé pour créer la solution décrite ci-dessous.

Passons à des choses plus matérielles !

Après ce défilé des environnements de développement, venons-en à un projet concret. Le cahier des charges est : pilotage de trois LED comme dans l'article de janvier/février [1], mais sans µC hôte ; l'ESP8266 avec son µC à 32 bits doit se débrouiller seul. La commande se fera par WLAN avec les mêmes clients pour PC, iPhone et tablette Android.

Jetons un coup d'œil au schéma complet de la **figure 6**. Au centre, une ESP-201 avec plusieurs E/S à usage général. Le pilotage des quatre LED se fait par GPIO 4, 5, 12 et 13. La LED verte indique l'état de la communication :

- **éteinte** : pas de connexion au réseau sans fil
- **clignotante** : l'ESP est connectée au routeur
- **allumée fixe** : connexion d'un client

Comme la charge maximale en courant d'un port GPIO est de 12 mA, nous avons intercalé un L293B, un pilote (*driver*) capable de fournir une tension de sortie plus élevée pour attaquer les LED blanche et bleue. Un pilote, cela peut sembler démesuré, mais j'ai l'intention d'utiliser cette carte pour une expérience avec un moteur gourmand en courant. Des résistances de polarisation basse aux entrées du pilote veillent à ce que leur état soit défini au démarrage de l'ESP. Un point important pour la sélection de l'adresse de démarrage est l'état des GPIO 0, 2 et 15 lors de la phase initiale : lorsque GPIO 0 et GPIO 2 sont au niveau haut et GPIO 15 au niveau bas, l'ESP est forcée à démarrer depuis la mémoire Flash externe. Des résistances de polarisation haute/basse définissent les niveaux requis.

Un convertisseur USB/RS232 à puce FT232RL sur la carte facilite le chargement du micrologiciel. Le transfert du logiciel vers l'ESP se fait par les lignes TX/RX. Pour cela il faut d'abord activer le mode Flash en mettant au niveau bas GPIO 0 et

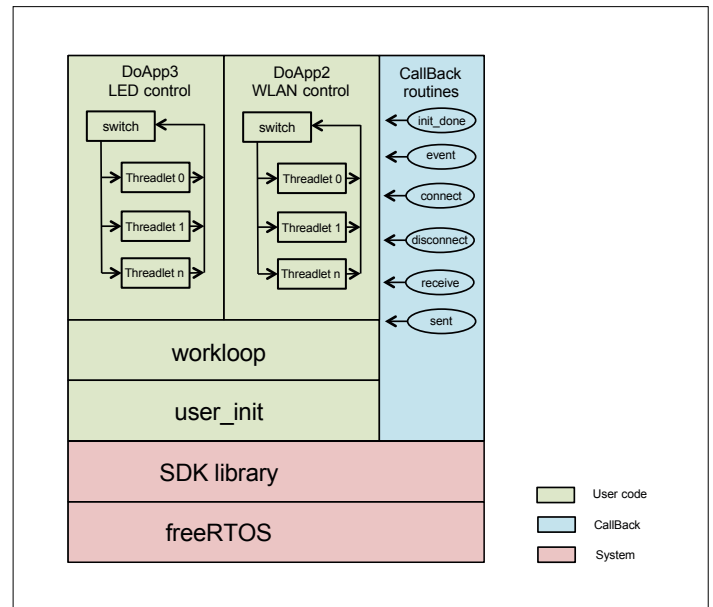


Figure 7. Structure du micrologiciel.

RST par les sorties RTS et DTR du FT232. Les diodes BAT43 prises dans ces lignes servent au découplage. Le transfert se fait à 115200 bits/s, Eclipse se charge de la commande. Le connecteur USB assure l'alimentation du convertisseur USB/RS-232. Un bouton-poussoir permet un redémarrage manuel du micrologiciel de l'ESP, en mettant au niveau bas la ligne RST. L'important lors du choix du convertisseur est qu'il doit pouvoir être réglé sur une tension de sortie de 3,3 V. Une tension plus élevée détruirait l'ESP !

Pour permettre le flashage d'autres cartes, j'ai ramené les lignes de programmation sur un connecteur à six broches et connecté l'ESP par le biais de cavaliers. Pour une programmation extérieure, on enlève les cavaliers et on connecte la carte cible externe avec un câble court.

L'alimentation est classique : une tension d'entrée de 7 à 9 V_{CC} est appliquée au pilote (qui peut supporter des tensions jusqu'à 45 V) via une diode de protection contre une inversion de polarité ainsi qu'à l'entrée du régulateur à découpage qui produit une tension de sortie fixe de 3,3 V. Comme la consommation de l'ESP est en moyenne de 70 à 80 mA, avec des crêtes de 170 mA, il faut dimensionner en conséquence la source d'alimentation. Des condensateurs de 220 µF tamponnent les crêtes de courant. La photo du montage de l'autour montre comment agencer la platine à trous.

Puce habillée par freeRTOS

Espressif met à disposition des kits de développement logiciel (SDK) avec ou sans RTOS (*Real Time Operating System*). Les interfaces vers les fonctions système et WLAN sont identiques. Si vous n'avez jusqu'à présent travaillé que sur des contrôleurs « tout nus », c'est-à-dire uniquement avec le micrologiciel installé par le fabricant, la transition vers un environnement avec système d'exploitation en temps réel demande une certaine adaptation. Perte de contrôle, mais gain de confort.

Pour la commande de l'ESP8266, nous utilisons un système

Listage 1. Boucle sans fin sous FreeRTOS

```
LOCAL void ICACHE_FLASH_ATTR workloop(void *arg)
{
    os_timer_disarm(&Wloop); // désactive le Timer Wloop

    DoApp2(); // automate fini Wi-Fi
    DoApp3(); // automate fini LED
    os_timer_setfn(&Wloop, (os_timer_func_t *)workloop, NULL);
    // appelle workloop une nouvelle fois
    os_timer_arm(&Wloop, 1, 0); // après une pause de 1 ms
}
```

Listage 3. Des instructions Print permettent de documenter le déroulement du micrologiciel

```
La plateforme ESP8266 démarre !
                                // démarrage dans user_init

SDK version:1.3.0              // version de SDK utilisée
CPU-Freq: 80                   // horloge CPU 80 MHz
Init done.                     // modules WLAN initialisés
A2ConnStation                  // DoApp2 prend la main et...
ESP8266 in STA mode configured.
                                // commute en mode station

Event: EVENT_STAMODE_CONNECTED
                                // connexion d'ESP au routeur réussie

Event: EVENT_STAMODE_GOT_IP
                                // ESP a obtenu une adresse IP

A2CheckIP                      // elle est interrogée puis...
IP OK 192.168.178.28           // imprimée
A2StartServer                  // démarrage du serveur TCP
                                // attente du client

tcpclient_connect_cb           // le client s'est connecté
IP OK 192.168.178.20           // adresse IP du client
                                // en attente d'une commande

tcp_rcv_cb B2,Lon
                                // le client envoie une commande, LED = allumée

A2ProcMsg                      // traitement de la commande
A2SendMsg Data sent, Txbu: ack
                                // accusé de réception pour le client

tcp_sent_cb                    // émission réussie
                                // en attente d'une commande

tcp_rcv_cb B1,Lof
                                // le client envoie une commande, LED = éteinte

A2ProcMsg                      // traitement de la commande
A2SendMsg Data sent, Txbu: ack
                                // accusé de réception pour le client

tcp_sent_cb                    // émission réussie
tcpclient_discon_cb           // le client s'est déconnecté
```

Listing 2. Fonction tcpclient_connect_cb appelée après la connexion (login) d'un client

```
static void ICACHE_FLASH_ATTR tcpclient_connect_cb(void *arg)
{
    Pconn = arg; // prise en charge de l'argument
    espconn_regist_sentcb(Pconn, tcp_sent_cb);
    // init. tcp_sent_cb
    espconn_regist_recvcb(Pconn, tcp_rcv_cb);
    // init. tcp_rcv_cb
    ConnState = TCP_CONNECTED; // état du signal
}
```

connu : FreeRTOS. Il coordonne à la fois les fonctions WLAN et les activités propres au micrologiciel. Lors de la programmation, il faudra tenir compte des aspects suivants :

- éviter les boucles de temporisation qui chargent la CPU ; les remplacer par des temporisations (logicielles) structurées.
- ne jamais définir une variable *Timer* en local, mais en statique, cela va de soi d'ailleurs.
- une fonction de l'utilisateur ne doit pas durer plus de 15 ms, sinon des fonctions WLAN en concurrence seront perturbées ou le temporisateur du chien de garde déclenchera une réinitialisation.

Cela dit, nous pouvons passer à notre micrologiciel-maison dont on découvre la structure en **figure 7**. Comme indiqué, il repose sur le système d'exploitation FreeRTOS et les nombreuses fonctions de service du SDK. Comme pour tous les programmes en C, une fonction *main*, d'ailleurs cachée des yeux de l'utilisateur, s'occupe du contrôle général. Elle tourne en arrière-plan et commande, entre autres, toutes les fonctions WLAN.

Après le démarrage de l'ESP8266, on appelle une fois la fonction *user-init*. Cette fonction :

- définit les paramètres pour les périphériques, UART et GPIO dans le cas présent,
- démarre une fonction de rappel (*callback*) *Init* et un gestionnaire d'événement (*Event Handler*),
- active une fonction *Systick* pour un timer logiciel propre (cadence de 10 ms ici),
- et appelle une boucle principale propre, baptisée ici *workloop*.

Le principe est évident, on a affaire une série de routines de type *Callback*. Avec *Init*, on signale que l'ESP est fonctionnelle : tous les périphériques WLAN fonctionnent correctement. Le gestionnaire d'événement est lui aussi une routine *Callback* qui signale, entre autres, les événements utiles ici, à savoir :

- **EVENT_STAMODE_CONNECTED** : la puce ESP est connectée au routeur en tant que station
- **EVENT_STAMODE_DISCONNECTED** : la connexion avec le routeur est coupée
- **EVENT_STAMODE_GOT_IP** : le routeur a donné une adresse IP à l'ESP



Avec l'ESP8266, la communauté des *faiseurs* dispose d'un module qui permet de mettre en œuvre des projets liés à l'Internet des Objets sans μ C hôte

Dans ces routines *Callback*, comme dans toutes les autres encore à mentionner, il n'y a pas de traitement significatif, simplement la mise du signal correspondant dans une variable globale.

Le traitement se fait dans les automates finis *DoApp2* et *DoApp3* appelée par *workloop* sachant que *DoApp2* se charge des activités spécifiquement WLAN et *DoApp3* du pilotage des LED. Ici, *workloop* ne prend pas la forme d'une boucle sans fin classique puisqu'après l'appel successif des Apps mentionnées plus haut, elle est terminée ; toutefois après une courte temporisation, elle se rappelle elle-même. Cela évite le blocage du processus et permet aux fonctions du système de reprendre le contrôle. Le **listage 1** montre une boucle sans fin sous FreeRTOS.

Lors du traitement, d'autres routines *Callback* que j'ai définies sont initialisées :

- *tcpclient_connect_cb* : un client s'est connecté
- *tcpclient_disconnect_cb* : un client s'est déconnecté
- *tcp_rcv_cb* : réception d'un message TCP
- *tcp_sent_cb* : l'émission d'un message TCP a réussi

Dans ces fonctions aussi, il n'y a que signalisation de l'événement dans une variable et transfert à *DoApp2/3*. Le **listage 2** montre à titre d'exemple la fonction *tcpclient_connect_cb*, appelée après connexion d'un client. Elle se contente d'activer les routines d'émission et de réception et de signaler le nouvel événement. Les fonctions *espconn* sont implémentées dans les bibliothèques fournies.


DoApp2 et *DoApp3* travaillent en *multithreading* coopératif. Après l'initialisation globale dans *user_int*, la boucle principale *workloop*, dans laquelle a lieu un appel séquentiel des *threads*, est activée. Ces derniers sont conçus comme des automates finis dans lesquels des séquences de code – appelées ici *threadlets* – sont traitées en fonction d'une variable d'état. Si le *threadlet* a été traité avec succès, il change la

variable *switch*, de sorte qu'il y a, à l'appel suivant, exécution d'un autre *threadlet* en fonction de la répartition des tâches. Des instructions *Print* permettent de documenter le déroulement du micrologiciel, cf. le **listage 3**.

Les commandes client que j'ai définies, cf. le **tableau 1** ci-dessous, sont exécutées par l'ESP. Dans les interfaces utilisateur des clients se trouvent des boutons et glissières, la variable <value> peut varier de 0 à 255 (0 = éteint, 255 = plein pot). Le programme source Eclipse, téléchargeable en [10], donne plus de détails.

Cette solution autonome fonctionne tout aussi bien que la combinaison avec le STM32 décrite en [1].

Conclusion et perspectives

Avec l'ESP8266, la communauté des *faiseurs* dispose d'un module qui permet de mettre en œuvre des projets liés à l'Internet des Objets, petits et moyens, sans μ C hôte. Pour le développement du micrologiciel, il existe des outils appropriés pour (presque) tous les goûts. L'aventure continue : le grand frère de la puce vient déjà d'être annoncé. Elektor suit les choses de près et vous tiendra au courant ! 

(150094 – version française : Guy Raedersdorf)

Tableau 1. Commandes et (ré)actions de l'ESP

activation	commande du client	action
bouton 1	B1, Lof	éteindre LED rouge
bouton 2	B2, Lon	allumer LED rouge
bouton 3	B3, Lbl	clignotement de la LED rouge
bouton 4	B4, Lzz	flash de la LED rouge
glissière 1	S1, <value>	valeur de la luminosité pour LED bleue
glissière 2	S2, <value>	valeur de la luminosité pour LED blanche

Liens

- [1] WLAN pour microcontrôleurs (Elektor 01-02/2016) : www.elektormagazine.fr/150093
- [2] Langage de script Lua : www.lua.org/
- [3] NodeMCU : <https://github.com/nodemcu/nodemcu-firmware>
- [4] ESPlorer : <http://esp8266.ru/esplorer/>
- [5] EDI Arduino : www.arduino.cc/en/Main/Software
- [6] Extension pour EDI Arduino : http://arduino.esp8266.com/stable/package_esp8266com_index.json
- [7] Livre de Kolban : <http://neilkolban.com/tech/esp8266/>
- [8] Système de développement à base d'Eclipse : www.esp8266.com/viewtopic.php?f=9&t=820
- [9] FreeRTOS : www.freertos.org/
- [10] Page du projet chez Elektor : www.elektormagazine.fr/150094



bienvenue dans votre e-choppe

Elektor recommande



Analyseur de signal par Ethernet

Besoin d'un oscilloscope simple, d'un générateur de signal rudimentaire et d'un analyseur de spectre pratique ? L'Analyseur de Signal Par Ethernet, compact et bon marché, est fait pour vous. Il numérise des signaux avec des fréquences d'échantillonnage jusqu'à 1 MHz. Les signaux numérisés, présentés comme sur un oscilloscope, sont affichés sur un PC sous Windows. Le logiciel sur PC permet aussi de produire un spectre de puissances dans le domaine fréquentiel. Les variables

contrôlant l'échantillonnage et la transformée de Fourier sont paramétrables avec les commandes intuitives de l'interface graphique. Pour utiliser l'ASPE, il suffit de le connecter à un réseau local via un routeur, le logiciel pour PC le détectera sur l'une des quatre adresses IP possibles.

Clemens Valens, labo d'Elektor

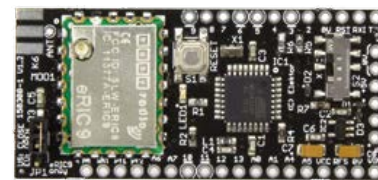
www.elektor.fr/network-connected-signal-analyser



Vos favoris :

1. eRIC Nitro

www.elektor.fr/eric-nitro



2. Red Pitaya V1.1

www.elektor.fr/red-pitaya-instrument

3. Authentique écran tactile de 7" Raspberry Pi

www.elektor.fr/rpi-touchscreen

4. Kit de 37 capteurs Raspberry Pi

www.elektor.fr/raspberry-pi-37-sensor-kit

5. Franzis Maker Kit Internet des Objets

www.elektor.fr/franzis-maker-kit-IoT

6. Abonnement Formule Green

www.elektor.fr/green

7. Carte d'E/S Android

www.elektor.fr/ardroid-i-o-board

Domotique avec Zelio Logic



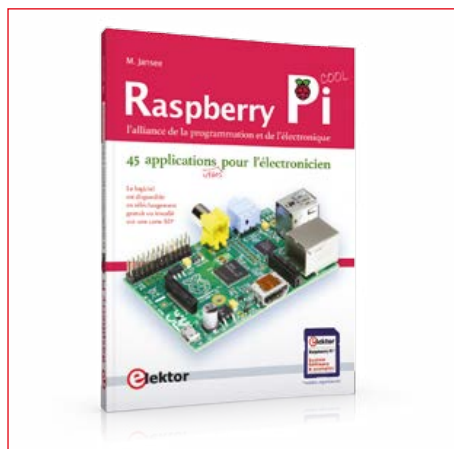
Le module programmable compact Zelio Logic avec 12 entrées/sorties est une solution technique bon marché et facile à mettre en œuvre, dans un logement, pour automatiser l'éclairage, piloter le chauffage, optimiser la consommation d'électricité ou encore pour se protéger contre les intrusions. Ce livre montre comment exploiter Zelio Logic avec son outil de programmation graphique en LADDER ou FBD (logiciel pour PC Zelio Soft 2).



Prix (membres) : 41,85 €

www.elektor.fr/domo-zelio

Raspberry Pi : 45 applications utiles pour l'électronicien



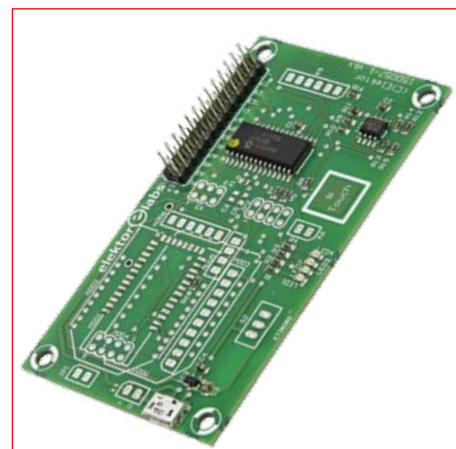
Après une courte présentation du R. Pi, vous installez le logiciel. Suit une introduction brève au système Linux et à la programmation en Bash, Python et Javascript. L'auteur expose seulement ce qui est nécessaire pour comprendre les projets et les adapter à vos besoins, puis il vous propose des projets plaisants, certains simples (clignotant alternatif, commande de moteurs), d'autres compliqués (régulation de vitesse de moteur, serveur météo avec CGI...)



Prix (membres) : 30 €

www.elektor.fr/rpi

Carte d'E/S Android



La carte d'E/S Android permet de piloter les ports d'un microcontrôleur PIC avec un appareil sous Android, par Wi-Fi, USB ou Bluetooth (versions 2.0 et 4.0). Les classes JAVA (bibliothèques) spécialement développées pour commander la carte d'E/S Android permettent de se concentrer sur l'application plutôt que sur la manipulation des entrées/sorties. La carte d'E/S Android basée sur un PIC16F1938-I/SO permet d'accéder aux 25 broches d'E/S.



Prix (membres) : 22,46 €

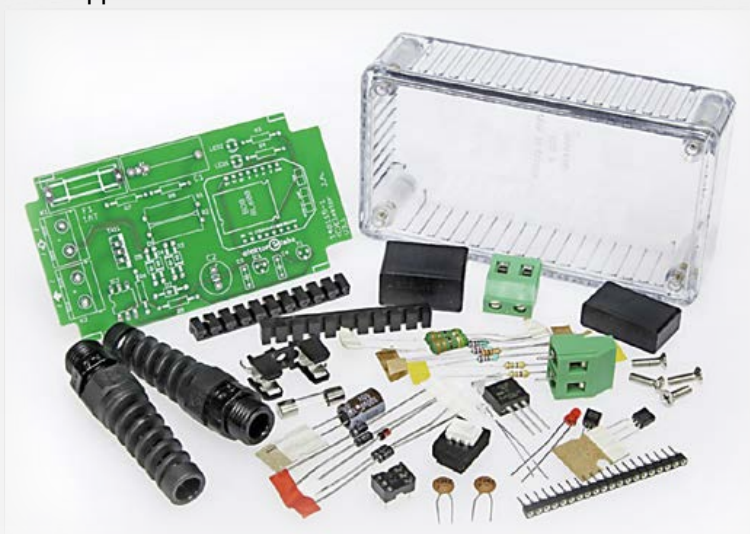
www.elektor.fr/android-i-o-board



Kit de l'interrupteur Bluetooth

L'interrupteur 230 V commandé par Bluetooth (Low Energy), présenté dans le numéro de décembre 2015, permet de commander un triac avec le BL600 à partir d'un ordiphone sous Android, par exemple pour allumer et éteindre une lampe.

Grâce à la carte de liaison e-BoB BL600 d'Elektor, il est possible d'accéder aux principaux signaux du module de communication BL600 de Laird Technologies, même sur un circuit d'application soudé à la main.



Ce kit contient une partie des composants nécessaires pour assembler l'interrupteur 230 V (max. 200 W) commandé par Bluetooth.

Cet interrupteur est basé sur l'e-BoB BL600 (non inclus).

Ce kit contient entre autres le boîtier Hammond, modèle 1591BCTL+ (140115-72).



Prix (membres) : 31,46 €
www.elektor.fr/kit-bluetooth

Kit de l'interrupteur Bluetooth

avec boîtier Hammond

Buggy Formula AllCode

un cours complet de robotique

e-choppe Elektor en ligne

Votre panier d'achats pour l'électronique

Générateur de fonctions par DDS

DVD Elektor 2015

Kit de l'anneau de LED



Module assemblé du générateur de fonctions compact (sinus/carrés/dents de scie) à synthèse numérique directe (DDS) avec amplitude max. de 15 Vcc, commande d'offset (± 10 V) et petit afficheur graphique (128x128 pixels). La plage de fréquence s'étend de 1 Hz à 10 MHz. Un petit clavier permet de régler les paramètres et de naviguer dans les menus comme le calibrage et le balayage.



Prix (membres) : 161,96 €

www.elektor.fr/dds-function-generator



Ce DVD-ROM réunit tous les articles d'ELEKTOR parus au cours de l'année 2015, en français (mais aussi en anglais, allemand et néerlandais). Il contient le texte des articles ainsi que les schémas, et tous les dessins des circuits imprimés, sous forme de fichiers à haute résolution. Une fonction de recherche dans la table des matières vous permet de trouver immédiatement l'article souhaité.

Prix (membres) : 24,75 €

www.elektor.fr/dvd-2015



Décorez votre maison avec cet anneau de LED. Il est programmable comme une carte Arduino Uno avec l'environnement de développement Arduino ; cela permet de créer facilement ses propres motifs lumineux. Cet anneau à LED est alimenté par deux piles CR2032. Attention : il faut prendre son fer à souder pour monter les composants fournis sur le circuit imprimé (livré nu). Si vous achetez trois kits, vous pourrez monter un anneau rouge, un jaune et un vert.



Prix (membres) : 26,06 €

www.elektor.fr/kit-anneau-de-led



par **Ernie Woollard**

Je suis ingénieur en électronique à la retraite depuis peu. J'étais à la recherche d'un document qui rassemble tout ce qu'il faut savoir sur Raspberry Pi et qui permette de se familiariser rapidement avec Linux et Python.

Ce livre répond parfaitement à mes besoins. Il couvre le matériel RPi, les commandes en ligne Linux, l'interface graphique, Python, l'interfaçage du matériel, et contient plusieurs exemples. Les projets proposés permettent de se confronter de manière pratique aux aspects à la fois matériels et logiciels de la conception. Il y a également un

chapitre sur le *Program Description Language* (PDL), un langage pour créer et documenter du code fiable et efficace.

Ce livre est bien écrit. En outre j'ai trouvé que les projets sont clairement expliqués avec de nombreux exemples de code, d'où une véritable immersion dans le sujet.

Les chapitres sur la programmation réseau et système en Python sont particulièrement intéressants et informatifs, ce sont des notions essentielles pour approfondir ses connaissances.

Ce livre conviendra aussi à tous ceux qui manipulent déjà la carte RPi et qui veulent l'utiliser dans des projets qui vont bien au-delà des notions de base.

www.elektor.fr/raspberry-pi-advanced-programming

... vous aussi envoyez-nous votre avis sur votre produit favori d'Elektor ; si nous le retenons pour une publication, vous recevrez en échange un bon d'achat de 100 € à valoir dans l'e-shoppe d'Elektor...

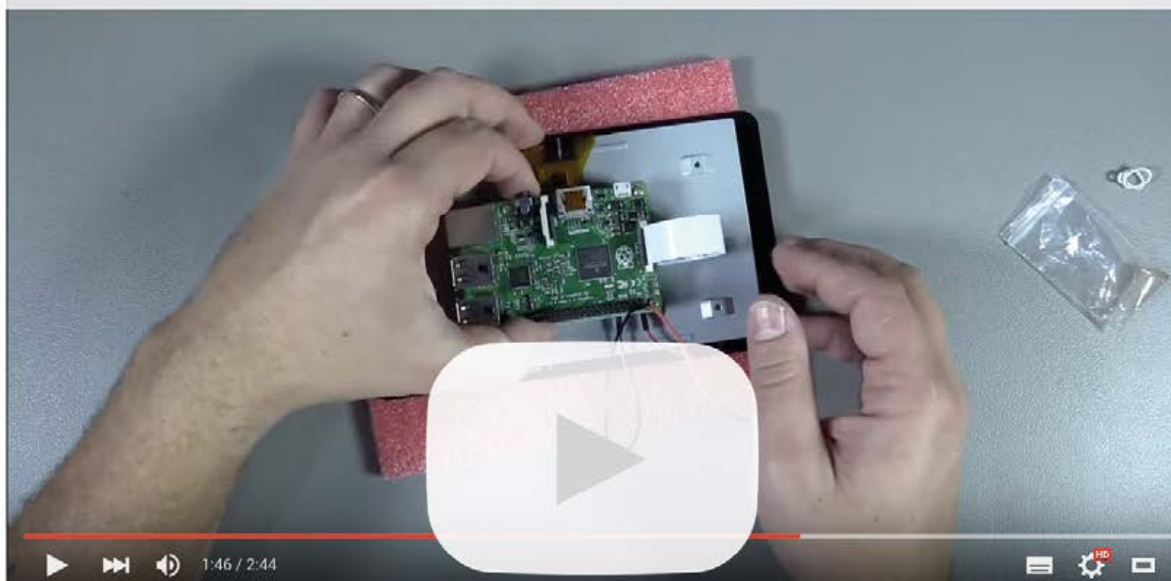
J'essaye : www.elektor.fr/rotm



layar
INTERACTIVE PRINT

- 1  Téléchargez l'appli gratuite Layar
- 2  Scannez cette page
- 3  Découvrez le contenu interactif

SCANNEZ CETTE PAGE ET REGARDEZ LA VIDÉO





Buggy Formula AllCode

Que vous soyez déjà un passionné de robots ou bien un débutant qui souhaite développer ses connaissances en matière de robotique, Formula AllCode est fait pour vous. Le principal avantage de Formula AllCode est que vous pratiquerez la robotique avec les plateformes matérielle et logicielle de votre choix. Formula AllCode comprend un cours complet de robotique. Le robot lui-même est compatible Bluetooth et peut devenir l'esclave de différentes plateformes (appareils Android et Apple, Raspberry Pi). Le cours gratuit fourni avec le robot comprend une palette d'activités avec différents niveaux de difficulté : depuis la simple détection de ligne jusqu'à l'analyse de labyrinthe. Le système repose sur un microcontrôleur dsPIC à 16 bits et il est programmable avec à peu près n'importe quoi : Flowcode 6, MATLAB, LabVIEW, Python, App Inventor, Visual Basic/C#/C++ etc.



Prix (membres) : 224,10 €

www.elektor.fr/formula-allcode-buggy

Kit de l'interrupteur Bluetooth

avec boîtier Hammond

Buggy Formula AllCode

un cours complet de robotique

e-choppe Elektor en ligne

Votre panier d'achats pour l'électronique

DVD Elektor 1990-1999



Ce DVD-ROM contient tous les numéros d'Elektor des années 1990 à 1999. Complétez votre schémathèque dans les domaines les plus divers (alimentation, audio, auto, domestique, hautes-fréquences, microcontrôleurs...) ! Les 2 200 articles d'Elektor réunis sont présentés par ordre de parution (mois/année), mais aussi classés par ordre alphabétique et par rubriques. Un index général permet d'effectuer des recherches dans tous les fichiers PDF.



Prix (membres) : 80,10 €

www.elektor.fr/dvd-elektor-1990-1999-french

Compilation BL600



Compilation BL600 en PDF : retrouvez les huit articles parus dans Elektor sur le BL600, un module de communication radio à consommation ultra-faible. Avec la carte de liaison pré-assemblée, vous pourrez accéder facilement et rapidement aux principaux signaux du module BL600 (I2C et son capteur de température, SPI et son convertisseur numérique-analogique) pour réaliser un thermomètre extérieur sans fil ou bien un interrupteur 230 V commandé.



Prix (membres) : 7,16 €

www.elektor.fr/e-select-compilation-bl600

Maker Kit - Internet des Objets



Avec ce kit, construisez vos propres projets pour l'Internet des Objets : horloge internet, serveur web, système d'alarme Twitter, etc. Entrez dans le monde de l'Internet des Objets avec cette carte (et les 21 autres composants) et commandez du matériel via l'internet. Vous apprendrez tout ce qu'il faut pour faire vos premiers pas (livre en anglais et allemand) : configuration des circuits, paramétrage du réseau, programmation en HTML, etc.



Prix (membres) : 71,96 €

www.elektor.fr/franzis-maker-kit-IoT

bienvenue dans la section **PARTAGER**

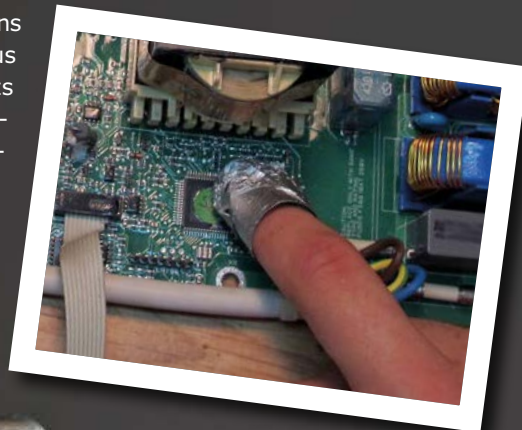


Thijs Beckers (Elektor NL)

Se mettre du papier d'aluminium sur le doigt, pourquoi pas ?

Comme à l'accoutumée dans la section PARTAGER, vous trouverez dans les bruits de labo un aperçu de plusieurs projets dignes d'intérêt à suivre sur www.elektor-labs.com. C'est MiBu qui a recyclé le fascinant afficheur fluorescent de sa

vieille radio de cuisine pour en refaire une neuve connectée. Toujours dans le remploi, la mécanique d'une parabole orientable sur les satellites peut vous servir à positionner une caméra ou une autre antenne. Écoutez les bruits du labo...



Nous vous racontons nos mésaventures avec les LED en filament et surtout nous vous donnons un truc bien pratique pour réaliser le projet Leditron décrit dans ce numéro.

Harry n'est pas seulement un chercheur en ligne, c'est un dénicheur hors pair ! Il vous a concocté une sélection d'astuces pour électroniciens. Le truc de la feuille d'aluminium sur l'index m'a particulièrement séduit.

Le projet Platino est revenu sur le devant de la scène dans le dernier numéro et les nouveautés s'enchaînent ici. Si vous n'avez pas suivi, sachez que Platino est né dans le magazine d'octobre 2011 et qu'il n'a pas cessé d'évoluer depuis. Pour la facilité d'emploi, nous lui avons ajouté le 3,3 V à bord, un deuxième port sériel, un bouton de RAZ et une sérigraphie plus complète. Il n'y a pas que le matériel qui en a profité, le logiciel aussi a été revu avec des fichiers de type *Board Packages*. Le nouvel EDI d'Arduino vous facilite la vie : désormais, vous pouvez sélectionner la carte cible depuis le *Board Manager* (gestionnaire de cartes). Pour finir, n'oubliez pas vous de concentrer sur l'Hexadoku. ◀

(150744 - version française : Robert Grignard)



tuyaux électroniques

pour réparer et concevoir

Harry Baggen (labo d'Elektor)

Vous connaissez bien sûr notre rubrique « trucs et astuces » qui apporte des solutions pratiques aux petits et grands problèmes de l'électronique. Sur l'internet aussi, nombre d'électroniciens proposent des solutions originales et ingénieuses. Nous en avons sélectionné quelques-unes.



Chaque électronicien, qu'il soit professionnel ou amateur, développe au fil du temps ses propres trucs pour le montage de circuits ou la réparation d'appareils. En fouinant sur l'internet, on trouve une foule d'astuces, et il est grand temps de s'y attarder.

Celui qui dépanne occasionnellement des circuits électroniques est souvent confronté à l'absence de schéma, et par conséquent à un circuit imprimé où il faut trouver soi-même les connexions entre composants. Si on veut par exemple savoir où mène la piste venant d'un transistor sur un circuit imprimé à double face ou à couches multiples, c'est relativement facile avec un testeur de continuité (multimètre). Mais cela devient très vite fastidieux de tester de nombreuses connexions avec deux pointes de touche en espérant entendre le fameux bip. Il est possible d'accélérer le processus en recouvrant un doigt d'une feuille d'aluminium raccordée avec un fil et une pince crocodile à une des deux pointes de touche. On place l'autre touche sur le point dont on veut connaître la terminaison sur le circuit imprimé, et on touche avec le doigt « métallisé » divers autres endroits du circuit. Ce doigt permet de toucher un certain nombre de points en une fois, et on établit donc une connexion

assez rapidement ; une fois celle-ci trouvée, on affine avec la pointe de touche en parcourant la zone concernée du circuit. Le gain de temps est appréciable et garanti ! [1]

Un truc bien pratique pour diminuer la chute de tension sur une diode est décrit en [2]. Si vous voulez mettre une diode de protection entre par exemple une batterie de 6 V et un circuit alimenté en 5 V, la chute de tension ne doit pas être trop élevée. On pense évidemment à utiliser une diode Schottky ; la chute est inférieure à celle d'une diode en silicium, mais elle atteint tout de même 1 V pour un courant de 100 à 200 mA. On peut aussi mettre deux ou trois diodes en parallèle, pour se trouver en deçà du coude de la courbe caractéristique et réduire la chute de tension à moins de 0,5 V. Il y a bien sûr de meilleures solutions, comme la mise en œuvre d'un transistor FET, que nous avons déjà utilisée chez Elektor ; mais si vous êtes pressé, souder quelques diodes entre elles est une solution rapide ! Sur le site *Electronics for Bharat* [3], on trouve beaucoup de trucs et astuces pour l'utilisation de platines d'expérimentation. Beaucoup sont bien connus de la plupart des électroniciens, mais il y en a quelques-uns dont on se dit : « Bon sang ! Mais c'est bien sûr ! Pourquoi n'y

avais-je pas pensé plus tôt ? ». Comme celui qui consiste à introduire une pile bouton CR2032 entre les broches d'un connecteur mâle 2 x 2 et ensuite monter le tout sur un circuit imprimé. Ou encore comment relier une carte SD à un circuit à l'aide d'une barrette d'embases femelles au pas de 2,54 mm.

Quelque chose de tout à fait différent sur le site de *Starlino Electronics* [4], pour distinguer un vrai circuit intégré RT232R ou RT232RL d'un faux. En effet ces dernières années, un véritable marché de contrefaçon des circuits intégrés s'est développé, et ces imitations peuvent poser de réels problèmes dans un circuit si on ne s'en aperçoit pas à temps. ◀

(150749 – version française : Jean-Louis Mehren)

Liens

- [1] www.instructables.com/id/How-To-Quickly-Find-And-Trace-PCB-Tracks-1/
- [2] www.daycounter.com/LabBook/Electronics-Tips-Tricks.phtml
- [3] <http://m8051.blogspot.nl/2012/08/updated-bread-board-tips-and-tricks.html>
- [4] www.starlino.com/ftdi-chip-real-of-fake-how-to-spot-a-fake-rt232r-rt232rl-and-others.html

LED en filament fragiles

de nombreuses LED sur un substrat de verre

Harry Baggen (labo d'Elektor)

Les lampes à LED ne sont plus vraiment une nouveauté, on en trouve de toutes tailles et formes ; la plupart tentent même d'imiter les bonnes vieilles lampes* à incandescence. Le rayonnement de nombre de lampes bon marché n'est hélas pas suffisamment omnidirectionnel, et les fabricants essaient de pallier ce défaut en ajoutant des réflecteurs et en plaçant les LED de manière particulière dans la lampe. Toutefois, la lampe a beau ressembler à un modèle à incandescence, ce n'est pas forcément esthétique dans un luminaire ouvert.

Ces derniers temps, une nouvelle variété est apparue sur le marché : les lampes à LED en filament. Au lieu de LED individuelles, on emploie des « filaments » de LED, qui ressemblent à un bâton et sont constitués d'un grand nombre de LED en série. La chute de tension est telle qu'on peut alimenter quelques LED en filament en série directement par le secteur, via un redresseur. Dans une ampoule, ces LED en filament sont semblables aux anciens filaments de carbone ; un autre avantage est que le rayonnement est bien plus régulier, du moins pour les meilleurs spécimens.

Pour la fabrication, on utilise la technique *chip-on-glass* : le matériau semi-conducteur des LED est placé directement sur un substrat de verre. Les barrettes sont ensuite recouvertes d'un enduit au phosphore. La composition de cet enduit et la couleur



* Une « lampe » est composée d'une ampoule (l'enveloppe) et d'autres éléments comme le culot, le filament...

des LED déterminent le spectre lumineux de l'ensemble. Ces barrettes sont munies d'une languette métallique à chaque extrémité, pour le montage et la connexion à l'alimentation.

Les LED en filament sont disponibles à l'unité
a u p r è s d e
revendeurs
chinois.

Nous avons eu l'idée de « détourner » leur emploi. C'est pour-quoi nous avons pensé à réaliser un grand afficheur à sept segments : c'est le LEDitron qui fait l'objet d'un article dans ce numéro. Nous avons commandé en Chine une grosse quantité de LED en filament, mais à l'arrivée 50% semblaient ne pas fonctionner. La raison ? Les substrats de verre sont très sensibles aux chocs et aux vibrations, et malgré un emballage soigneux, un bon nombre d'entre eux ne résiste pas au voyage. Nous avons bien sûr cherché – et déniché – un autre « fournisseur » : la lampe à LED en filament prête à l'emploi ! On la trouve pour quelques euros pièce chez les grossistes. Enveloppez l'ampoule dans un linge, brisez le verre avec précaution, et retirez les LED en filament : pour quelques euros, vous récupérez trois ou quatre LED en filament, qui conviennent parfaitement pour le LEDitron ! En outre, ces LED en filament semblent bien moins fragiles que leurs homologues achetées à l'unité en Chine ; un autre matériau que le verre est sans doute utilisé pour le substrat. Et elles résistent même à un petit choc ! ◀

(150747 – version française : Jean-Louis Mehren)

Lien

www.ledinside.com/knowledge/2015/2/the_next_generation_of_led_filament_bulbs





bruits de labo...

Chaque jour le génie et la diversité

Viennent nous enrichir de quelque nouveauté (de J.B. Gresset)

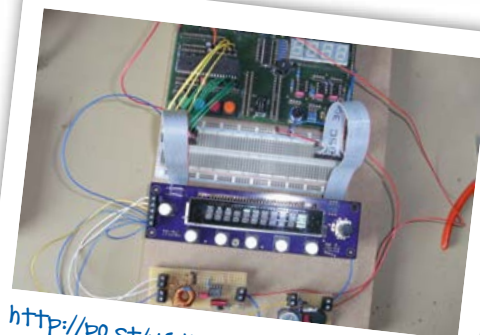
« Diversité », c'est le mot qui vient à l'esprit lorsqu'on parcourt les projets présentés sur www.elektor-labs.com. Voici une sélection éclectique de projets qui accaparent leurs auteurs pendant leur temps libre.

Afficheur fluorescent pour radio « de cuisine »

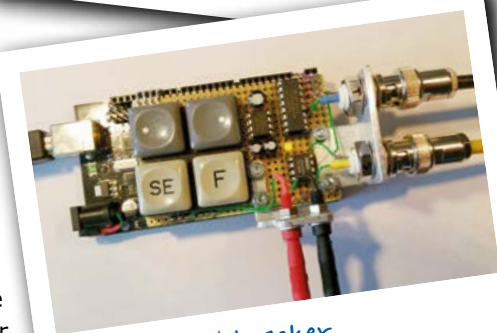
Le bleu troublant et le vert fantomatique des afficheurs fluorescents sont bien plus séduisants que le noir sur fond verdâtre des écrans à cristaux liquides. Lorsque la vieille radio de cuisine de l'auteur de ce projet est tombée en panne, il en a récupéré l'afficheur fluorescent avec l'idée de construire un nouvel appareil qui disposerait d'une horloge et d'un minuteur pour la cuisson. Un Atmega32 commande l'écran et les boutons, tandis qu'un RPi se charge de diffuser la radio via l'internet. À terme, l'auteur prévoit d'ajouter quelques fonctions domotiques à sa radio.

Suiveur de signaux lents pour oscillo

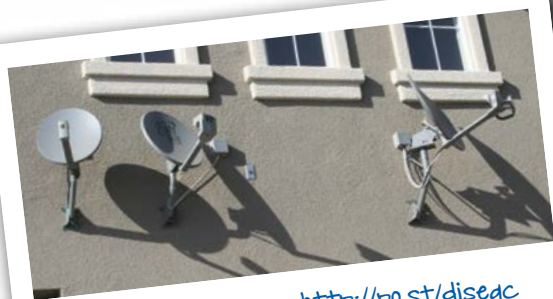
Pour étudier la courbe de décharge d'une batterie (signal très lent), le balayage horizontal de l'oscillo devrait se dérouler en heures plutôt qu'en fractions de seconde. Il faut donc ralentir le générateur de base de temps ou *timebase* de l'oscillo. C'est ce que fait ce projet à l'aide d'une carte Arduino Mega 2560 et d'un *shield* sur mesure. Il permet d'étendre jusqu'à 12h ou plus la plage du générateur de balayage (ou de poursuite) d'un oscilloscope normal. En outre tous les paramètres sont affichés sur l'écran de l'oscillo. Enfin ça marche aussi sur les bons vieux oscillos analogiques.



<http://po.st/vfdkitchenradio>



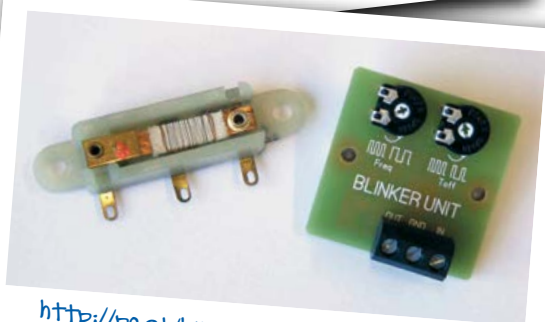
<http://po.st/vtracker>



<http://po.st/diseqc>

Commande d'orientation d'antenne ou de caméra de surveillance

Grâce à des vis sans fin, la mécanique d'orientation des grandes paraboles est robuste et puissante. Ce principe convient aussi pour l'orientation d'antennes terrestres, de caméras de surveillance... sur un angle de rotation de 180°. Le protocole DiSEqC de commande du matériel de réception (*Digital Satellite Equipment Control*) fait appel à des séquences particulières d'impulsions de 22 kHz : octet de départ, octet d'adresse, octet de commande, octet de donnée (chaque octet est suivi d'un bit de parité). Un μC ATtiny2313 suffit pour engendrer ce protocole en BASCOM dont l'instruction SOUND permet de produire le signal de 22 kHz sur n'importe lequel de ses ports de sortie.



<http://po.st/blinky>

Un clignotant qui marche pas... marche... marche pas...

La variété des projets sur elektor-labs.com reflète les nuances infinies de nos préoccupations d'électroniciens. Les uns rêvent d'aller se poser sur Mars (sans souci de retour), d'autres ont besoin d'un simple clignotant pour modèle réduit. Ce circuit simple, mais universel puisqu'il peut être alimenté entre 6 et 30 V CA ou 9 et 42 V CC, est idéal pour remplacer de vieux clignotants à bilame. Son encombrement est modeste : 36 x 36 mm. L'archiclasique temporisateur 555 (version CMOS) offre la possibilité de varier aussi bien la fréquence que la durée (ou rapport cyclique) du clignotement. ◀

(150736)

DÉCOUVRIR

CRÉER

PARTAGER

elektor•PCB Service

le fruit de la coopération d'elektor et d'eurocircuits



**confiez-nous
la production de vos circuits imprimés,
vous avez tout à y gagner !**

➡ **le prix !**

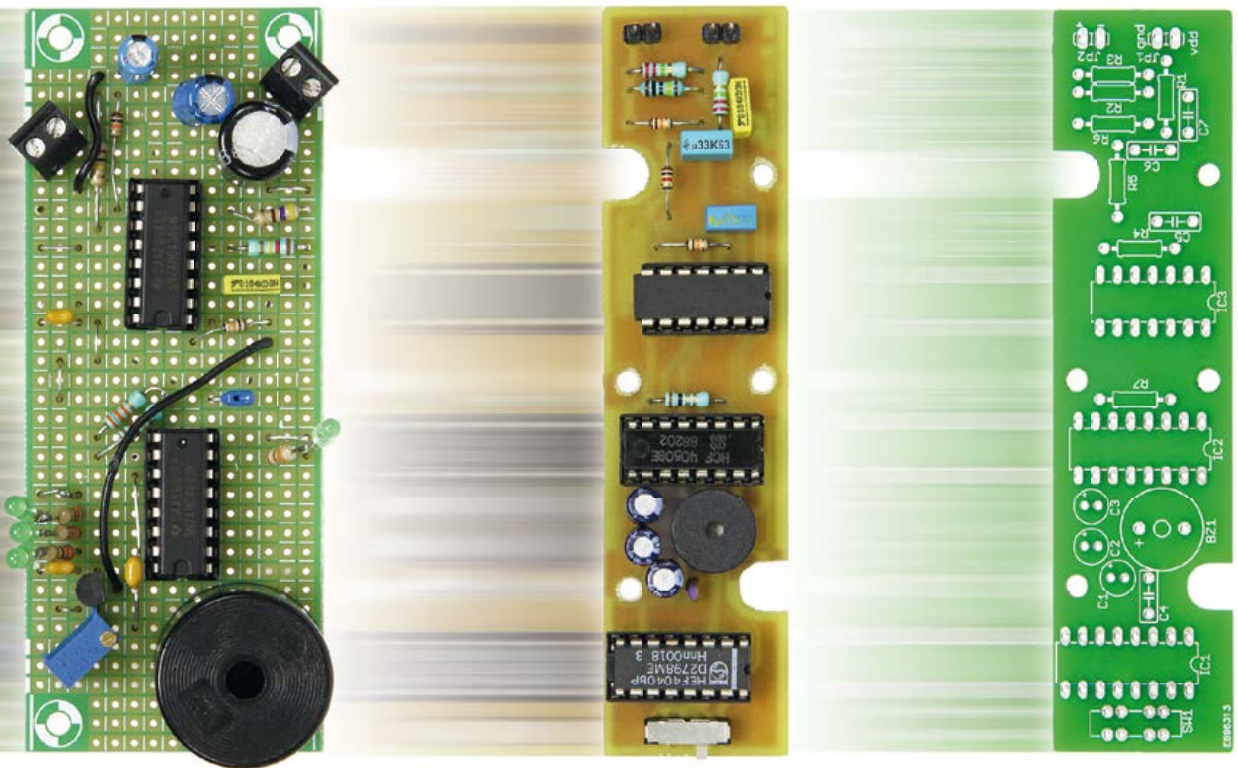
irrésistible...

➡ **la qualité !**

imbattable...

➡ **la fiabilité !**

inégalée...



elektor PCB service est, à l'échelle européenne, le service le plus complet de fabrication sur mesure de circuits imprimés. Ce service commence en ligne, par le confort et l'efficacité d'outils faits sur mesure, étudiés pour vous permettre de visualiser votre commande et de l'analyser *avant* de payer.

- Pour vos débuts, vous utiliserez éventuellement le service de **prototypage sans masque de soudure** ; dans ce cas, vos circuits imprimés simple ou double face sont nus.
- Pour les utilisateurs exigeants, nous proposons les options sans compromis. Le **PCB Visualizer** montre les circuits imprimés tels qu'ils seront livrés, le **PCB checker** procède à une vérification technique de votre circuits imprimés (*design rules check*), et enfin le **PCB configurator** facilite la préparation de la commande.

Des menus bien conçus et un guidage par options accélèrent le processus de commande sans négliger aucun détail. Au moment de passer votre commande, vous savez exactement ce qui sortira de nos machines.



Qu'il soit perso ou pro, confiez votre prochain PCB à :

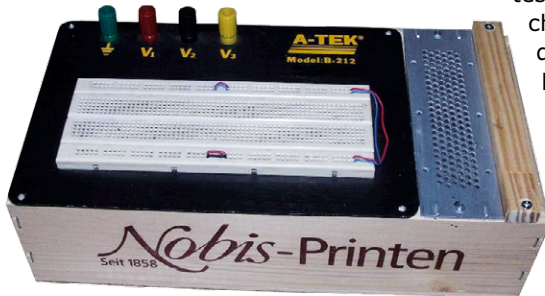
www.elektorPCBservice.com

compilées par **Aniek Reuling**

Tout sert, tout ressort

Après avoir assemblé sur un panneau un circuit destiné à tester un vieux afficheur, notre traducteur australien David Ashton s'est mis en quête d'un boîtier. Rien par ici ? Rien dans ce bureau non plus ? Soudain David s'est souvenu du cadeau

qu'Elektor lui avait offert pour Noël : une boîte de *Printens*, délicieuse spécialité pâtissière d'Aix-la-Chapelle. Et comme la pantoufle de Cendrillon, elle s'adapta à merveille au circuit !



Elektor-ID

Afin d'unifier les procédures de connexion sur nos différents sites, nous avons mis en place fin 2015 un identifiant (*ID-Elektor*) pour les membres Green et Gold. Ce compte Elektor-ID

ELEKTOR-ID

vous permet actuellement d'accéder au site du magazine (www.elektormagazine.fr) et à celui du labo (www.elektor-labs.com).

READ ONLY MEMORY

Le magazine Elektor et son éditeur sont fiers de leur histoire. Cette rubrique montre pourquoi.



Avant les écrans tactiles, il y eut les claviers à membrane. Leur coût de fabrication était bien moindre que ceux des claviers ordinaires. Ils n'avaient aucune pièce mobile et se reliaient

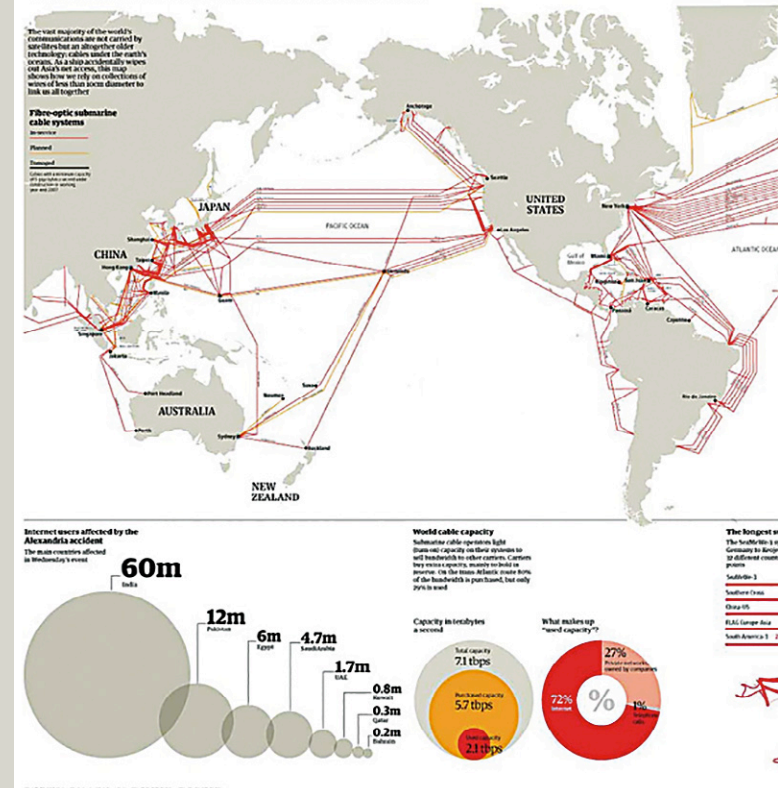
au monde extérieur au moyen d'une barrette de connexion. En outre ils pouvaient être de n'importe quelle forme et couleur, et ne craignaient pas les tasses de café renversées. Ce qui explique sans doute que cette « technologie de l'ère spatiale » ait fait fureur au début des années 1980.

fracture et dividendes

elektorethics par Tessel Renzenbrink

A priori, les technologies numériques pourraient bénéficier à l'humanité entière : moteur de croissance économique, création d'emplois, nouvelles prestations de services, le potentiel est là. D'après la Banque Mondiale, ce potentiel ne serait cependant pas exploité au mieux. Plus important, 60 % de la population mondiale n'a pas accès à l'internet. Cette fracture numérique accentue les inégalités entre ceux qui ont accès au réseau et ceux qui ne l'ont pas.

The internet's undersea world



Carte des câbles Internet sous-marins. Alexander van Dijk. Licence CC-BY.

Dans son rapport « Les dividendes du numérique », la Banque mondiale classe les bénéfices des technologies numériques en trois catégories principales : *inclusion*, car plus de personnes – même en zone rurale – peuvent participer à l'économie ; *efficacité*, car les transactions commerciales sont devenues plus rapides, meilleur marché ou plus commodes ; *innovation*, car l'automatisation des plateformes d'e-commerce favorise la création de nouveaux modèles commerciaux.

Risques émergents

Ces bénéfices ne sont ni acquis ni automatiques, prévient

INDISCRÉTIONS • La nouvelle antenne allemande d'Elektor située à Aix-la-Chapelle a organisé une pendaison nouvel habillage d'influence « steampunk » qu'elle a créé pour l'horloge Nixie (nous en reparlerons en mai) Raoul Morreau et Johan Dijk, ont travaillé dur à l'installation d'une base de données en ligne pour Elektor Cette rubrique sera désormais compilée par Robert van der Zwan.

hexadoku casse-tête pour elektorniciens

Comment ne pas perdre la mémoire ?

Nous savons tous qu'une activité physique régulière, une alimentation variée et un bon sommeil sont bénéfiques pour la santé, donc pour la mémoire aussi. Il semblerait que la concentration soit également le carburant de la mémoire. Les jeux de lettres et de chiffres sont donc conseillés pour stimuler le cerveau et entretenir sa mémoire. Concentrez-vous sur votre hexadoku mensuel !

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 tirerons au sort **trois bonnes réponses** internationales reçues dans les délais ; leurs auteurs recevront chacun un chèque-cadeau d'une valeur de **50 €** à valoir dans l'e-choppe d'Elektor. À vos crayons !

Envoyez votre réponse (les chiffres sur fond grisé) avec vos coordonnées par courriel, avant le **28 avril 2016** à l'adresse **hexadoku@elektor.fr**

Les gagnants

La solution de la grille du numéro de janvier-février est **BE840**

Les trois bons Elektor d'une valeur de 50 € vont à :

Gilbert **Dufieux** (France), Klaus **Kohlmann** (Allemagne), Susanne **Müller-Furrer** (Suisse).

Bravo à tous les participants et félicitations aux gagnants !

0	7	F			E	6			8	2			9	3	5
2	B	C	5			8			3				6	4	7
6	3	9		0	7				A	F			B	E	C
	8			3						D				1	
		D	6	9						5	B	8			
8		2			F		5	7		3			C		E
5	9					0	8							2	F
					6	4			B	E					
					0	5			F	8					
4	A						E	3						F	6
D		3			2		7	A		9			1		4
		7	F	4							C	2	D		
	1			5							A			B	
F	6	5		7	B					C	8		A	D	9
9	D	B	0			C			5			7	2	6	8
7	4	A			D	3			2	B			0	C	1

7	6	4	5	E	D	F	A	C	2	9	1	8	0	3	B
9	D	8	C	1	B	4	6	3	F	E	0	A	2	5	7
A	F	B	1	2	0	3	7	4	D	5	8	C	9	6	E
E	0	2	3	8	C	5	9	6	7	A	B	D	F	1	4
6	A	C	2	9	7	B	E	8	4	0	F	3	5	D	1
8	1	3	7	0	A	C	F	5	E	D	6	2	B	4	9
B	5	F	4	D	1	2	8	9	3	7	A	E	6	0	C
D	9	E	0	3	4	6	5	B	C	1	2	7	8	A	F
C	B	1	E	A	6	8	4	D	0	F	7	9	3	2	5
2	3	D	F	B	9	7	C	A	1	8	5	6	4	E	0
0	7	5	9	F	2	D	3	E	6	C	4	1	A	B	8
4	8	A	6	5	E	0	1	2	B	3	9	F	7	C	D
F	E	0	A	C	3	9	2	7	5	4	D	B	1	8	6
1	C	6	B	4	8	A	0	F	9	2	E	5	D	7	3
5	4	7	8	6	F	E	D	1	A	B	3	0	C	9	2
3	2	9	D	7	5	1	B	0	8	6	C	4	E	F	A

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.

Une qualité professionnelle à prix discount!



- ✓ Plus de 45 ans d'expérience
- ✓ Envoi en 24 heures
- ✓ Plus de 50 000 produits

Les langues de notre boutique:

Payment Methods:



Connecteur enfichable D-Sub avec SnapLock

- Pas de vissage requis
- Serrage fixe tout comme pour une fixation à vis

CONEC



1 Connecteur D-Sub

DSUB ST2093	9 pôles	2,73
DSUB ST2113	25 pôles	4,37
DSUB ST2123	37 pôles	5,80
DSUB ST2133	50 pôles	7,06

2 Prise D-Sub

DSUB BU2143	9 pôles	3,03
DSUB BU2153	15 pôles	3,86
DSUB BU2163	25 pôles	5,54
DSUB BU2173	37 pôles	7,22
DSUB BU2183	50 pôles	8,36

3 Couvercle D-Sub avec SnapLock, plastique

KAPPE 1810	pour 9 pôles	1,85
KAPPE 1820	pour 15 pôles	1,98
KAPPE 1830	pour 25 pôles	2,10
KAPPE 1840	pour 37 pôles	2,14
KAPPE 1850	pour 50 pôles	2,23

4 Couvercle D-Sub avec SnapLock, métallique

KAPPE 1750	pour 9 pôles	2,90
KAPPE 1760	pour 15 pôles	3,32
KAPPE 1770	pour 25 pôles	3,74
KAPPE 1780	pour 37 pôles	4,16
KAPPE 1790	pour 50 pôles	4,79

S'ABONNER MAINTENANT !

Newsletter

Recevez chaque semaine les toutes dernières informations

- ✓ Nouveautés
- ✓ Les meilleures offres
- ✓ Réductions des prix



opticalCON, robuste système de connexion à fibres de verre

opticalCON est un robuste système de connexion à fibres de verre destiné à de multiples applications. Grâce à son faible poids, à son design compact, à son robuste boîtier et à ses cycles élevés de connexion, ce système offre une connexion sécurisée et au coût avantageux.



NEUTRIK

Câble de connexion à fibres de verre à 2 canaux, 3 m

- ☐ Étanche à l'eau conformément à IP65
- ☐ Blocage Push-Pull fiable



122,02
OCON 3

Prise encastrée opticalCON DUO, noire

- ☐ Étanche à l'eau conformément à IP65
- ☐ Capuchon de protection avec joint d'étanchéité au silicone protège la connexion visuelle



57,94
OCON BU BK

NEUTRIK etherCON

Prise encastrée noire CAT6 en forme D

- ☐ Apte à CAT6
- ☐ IP65
- ☐ Blocage Push-Pull

18,15
EBU CAT6 BK

Débits de données jusqu'à 10 Gbit/s



Borniers à ressort unipolaire de MetzConnect

- ☐ Apte au brasage
- ☐ Avec bouton poussoir pratique

un morceau de **0,26**



RIA CONNECT BTR NETCOM MCQ TECH



SR 901 CW

SR 901 YE

SR 901 GR

SR 901 OR

SR 901 RD

SR 901 BK

Commander maintenant! www.reichelt.fr

Assistance téléphonique en anglais : **+49 (0)4422 955-360**

Prix du jour ! Prix à la date du : 29.2.2016

Prix en € hors T.V.A., plus coûts d'expédition
reichelt elektronik, Elektronikring 1, 26452 Sande (D)

mouser.fr

Les dernières nouveautés pour
vos conceptions les plus récentes™



La plus **vaste sélection** de produits les **plus récents**.

Plus de **4 millions** de produits de plus de **500 fabricants**.

Distributeur agréé de semi-conducteurs
et de composants électroniques



MOUSER
ELECTRONICS