

# eLektor

www.ektor.fr

## votre serveur web avec la carte Linux d'Elektor

- + VU-mètre stéréo à tubes
- + périphériques maison pour téléphone Android
- + bibliothèque C pour ElektorBus

## régulateurs de tension à découpage

ils restent froids et ne sont  
pas plus encombrants que  
les 78xx & 79xx



L 19624 -413- F: 7,20 €





**Spécial: Économisez 50% Spécial: Économisez 50%**

# Fêtez le 25ème Anniversaire de *Circuit Cellar*



**\$25** Papier ou Numérique :: **\$50** Les deux éditions

**Fêtez le 25ème Anniversaire** de la date depuis laquelle *Circuit Cellar* propose à ses lecteurs une analyse éclairée de la technologie à la base de l'électronique enfouie.

Allez rendre visite à [www.circuitcellar.com/el912](http://www.circuitcellar.com/el912) pour bénéficier de ces superbes offres.

**OFFRE BONUS! OFFRE BONUS! OFFRE BONUS! OFFRE BONUS! OFFRE BONUS!**

Abonnez-vous aujourd'hui et recevez également, outre votre abonnement, **le Numéro Spécial du 25ème Anniversaire!**

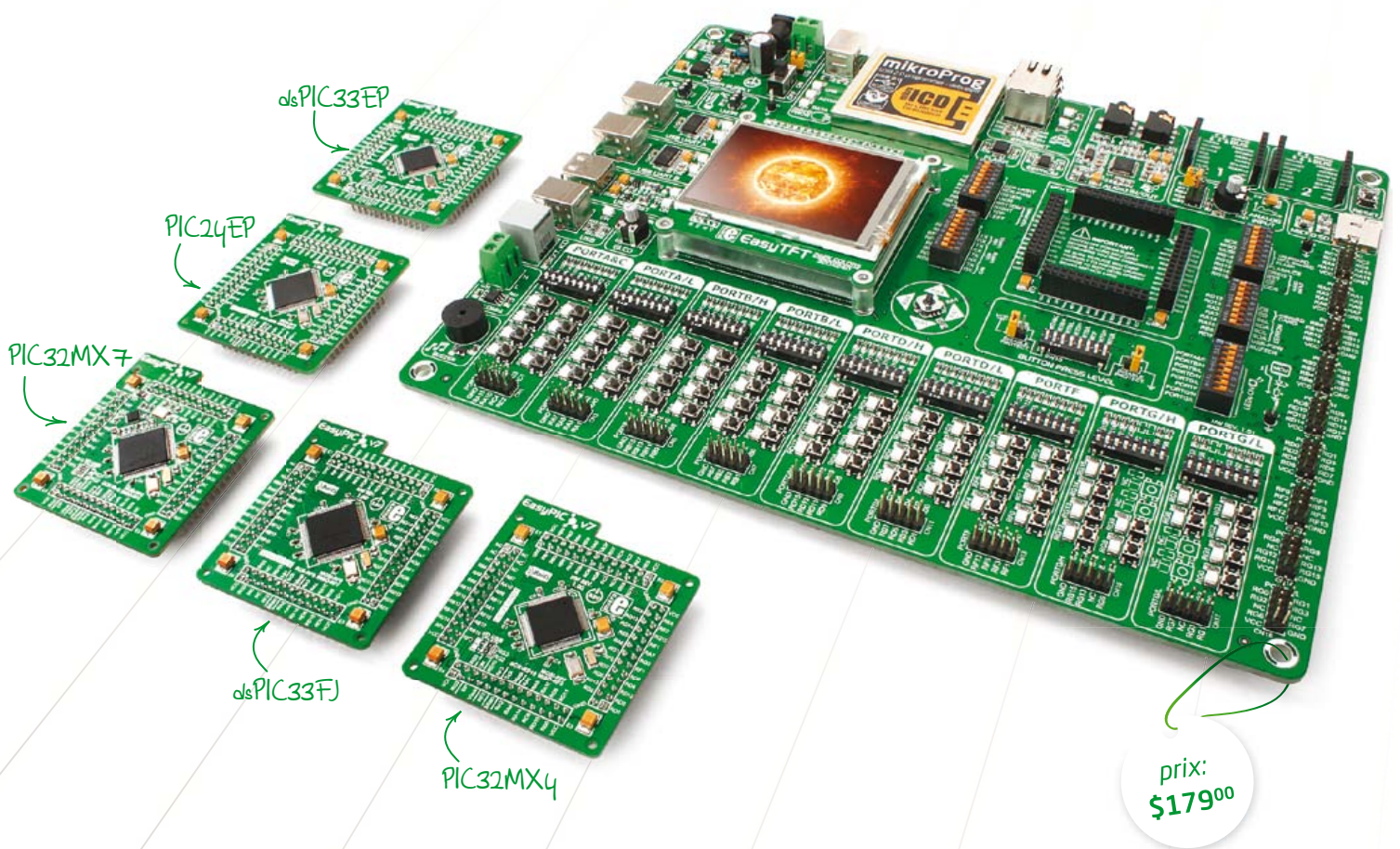


# Union des 16 bits et des 32 bits

L'intelligence est la faculté de s'adapter rapidement. Pouvoir changer de microcontrôleur de façon transparente pour trouver la solution la meilleure pour votre projet, met entre vos mains la clé du succès. Cela n'a jamais été facile. Jusqu'à aujourd'hui. **EasyPIC Fusion™ v7** est la première carte de développement au monde à combiner le support de trois architectures différentes : dsPIC33®, PIC24® et PIC32®. Empoignez une carte et lancez-vous.

Travaillez smart, pas dur.

## EasyPIC Fusion<sup>v7</sup>





## L'art de l'électronique

Ce matin, en route vers le bureau, je suis doublé par une DS en parfait état, d'un curieux modèle break, très sixties. Cela m'a rappelé qu'une DS de 1957 était la première voiture entrée dans les collections d'un musée d'art et pas n'importe lequel\*. Aussitôt mes cogitations *artisticautomobiles* sont parties dans tous les sens ; je me suis demandé par exemple si l'électronique était elle aussi de l'art. De fil en aiguille, j'ai repensé au choix fait, en 1996, de ne pas traduire littéralement le titre de *the Art of Electronics* quand Elektor a édité ce livre\*\*. Un choix peut-être fondé au siècle dernier, mais que je ne referais pas aujourd'hui. Avec la patine du temps et la raréfaction qui en découle, comme toutes les techniques raffinées, l'électronique d'hier finit par acquérir une dimension mythique. Je doute que cela suffise à en faire un art. Tous utilisent grosso modo les mêmes techniques, mais il y a des chirurgiens, des cuisiniers, des jardiniers et des... électroniciens plus doués que les autres. Cela n'en fait pas encore des artistes ? La musique la plus sublime se laisse réduire à des phénomènes physiques ou à une arithmétique mélodique ou harmonique. Les ordinateurs de l'IRCAM apprennent d'ailleurs à dialoguer avec des musiciens improvisateurs. J'ai rencontré bien des concepteurs et des programmeurs qui évoquent leurs créations comme le fruit de leur expression. Comme des artistes ? Je ne suis pas arrivé beaucoup plus loin dans mes réflexions contradictoires, le break DS a bifurqué et mon attention a été captée par le CD que je venais de mettre. Une merveille comme je n'en ai pas entendu depuis longtemps, *Unison* du trio de jazz *Aka Moon*. Émotion si forte qu'une fois arrivé au parking, je suis resté dans la voiture un bon moment pour l'écouter jusqu'au bout. Tant de bonheur, grâce à quoi ? À l'**électronique**, bien sûr. Celle qui a permis à l'ingénieur magicien du son, Michel Andina, de l'enregistrer aussi parfaitement, celle qui permet d'en faire un CD et celle qui permet de le reproduire plus que convenablement dans un milieu aussi improbable qu'une auto. Et si c'était tout simplement ça, *l'art de l'électronique* ?

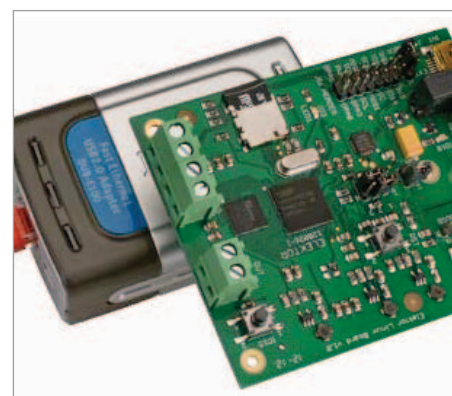
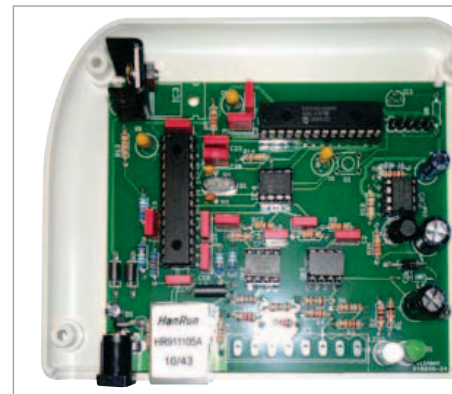
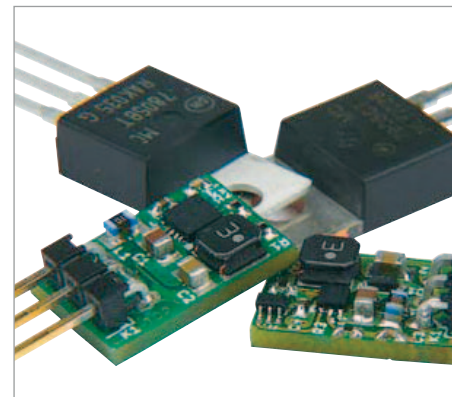
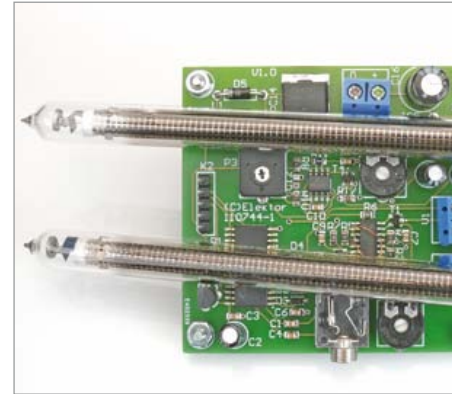
Denis Meyer

\* Musée d'art moderne à New-York

\*\* Traité de l'électronique analogique & numérique  
[www.elektor.fr/TT1](http://www.elektor.fr/TT1)

# elektor

- 6 de nous à vous : le réseau Elektor**  
Informations pratiques & légales
- 8 elektor : (bien) plus qu'un magazine**  
Par vagues rapprochées, les innovations de l'électronique inondent le monde. Pour se maintenir au sommet de ces déferlantes, Elektor aussi se renouvelle.
- 10 elektor.labs : visite guidée**  
Visite guidée d'elektor.labs, la pompe à idées d'Elektor.
- 12 retour aux sources (9)**  
Les oscillateurs sont les éléments clés de nombreux appareils et le socle d'une grande variété d'applications.
- 16 VU-mètre à Nixie**  
Association de deux composants séparés par 45 ans : un PSoC et un tube Nixie IN-9.
- 22 régulateurs de tension froids**  
Désormais pour remplacer les 78xx vous disposez de régulateurs à découpage qui ne dissipent rien.
- 28 mettez le cap sur Arduino 3a**  
L'un des inventeurs d'Arduino vous invite à jouer avec la carte Arduino Mega ADK pour la relier à des téléphones ou tablettes Android.
- 34 DCF sur l'enregistreur météo USB**  
L'enregistreur de données autonome paru en sept. 2011 dans une version agrémentée d'un module DCF.
- 36 Software Defined NIC**  
Difficile à croire, mais un AVR à 8 bits qui tourne à 20 MHz est capable d'expédier des paquets sur un réseau Ethernet !
- 44 Rigol : analyseur de spectre abordable**  
Un analyseur de spectre dont la plage s'étend sur 1,5 GHz avec un générateur de poursuite pour seulement 1 058 €.
- 45 bip-bip-bip... c'est quoi, ce bip ?**  
Mystérieux gazouillis entendu sur un récepteur de balise de détresse de modèle réduit.





# SOMMAIRE

35<sup>e</sup> année  
novembre 2012  
n° 413



## 16 VU-mètre à Nixie

Entre générations de composants, les frontières s'évaporent aussi. Cet article associe deux composants séparés par 45 ans d'évolution technique vertigineuse. D'un côté un PSoC (*Programmable System on Chip*) et de l'autre, le tube Nixie IN-9. À l'âge d'internet, voyons comment collaborent sous le fer d'Elektor, un pur produit Sovjet-elektronisjk et un zeste de technologie de la Silicon-Valley.



## 22 régulateurs de tension froids

Adieu 7805 et 7905 ! Voici des remplaçants pour ces régulateurs, légendaires pas seulement pour leurs qualités mais aussi pour leur dissipation si forte que l'usage d'un radiateur est souvent une contrainte incontournable et parfois rédhibitoire. Désormais vous disposez de régulateurs à découpage qui ne dissipent rien mais ne sont pas plus encombrants qu'un 7805. Et compatibles broche à broche !



## 48 interface internet pour compteur électrique domestique

Frémissez-vous de culpabilité quand on vous parle de votre empreinte carbone ? Si vous souhaitez réduire l'une et l'autre, il faut commencer par connaître à tout moment la quantité d'électricité que vous consommez et que vous payez. Et si l'internet pouvait nous aider à la déterminer et à la surveiller ? Voici *openEnergy* et *openSource* pour énergimètre So.



## 68 embarquez Linux ! (5)

La carte Linux d'Elektor offre de nombreuses possibilités d'applications. L'une d'entre elles est de lire des signaux analogiques et numériques au travers d'une connexion réseau, et de se servir d'un serveur web léger pour afficher les informations obtenues. Ici nous commanderons à distance la LED de la carte, et nous afficherons son état sur une page web créée de façon dynamique.

- 48 interface internet pour compteur électrique domestique**  
Pour tous ceux qui se préoccupent de la quantité d'électricité qu'ils consomment.
- 52 amplificateur de mesure à transconductance**  
Un amplificateur à transconductance est un amplificateur différentiel dont le courant de sortie est une fonction de la différence de tension entre les entrées.
- 56 bibliothèque C pour ElektorBus**  
Grâce à cette bibliothèque C pour ElektorBus, il n'y a plus à s'emm... brouiller avec les protocoles de bus.
- 66 déphaseur commandé en MLI**  
Faire varier le déphasage entre deux signaux carrés peut être bien utile.
- 68 embarquez Linux ! (5)**  
Ici nous commanderons à distance la LED de la carte, et afficherons son état sur une page web.
- 76 3 mois de stage au labo d'Elektor**  
Francois-Xavier Maurille, étudiant en 2e année d'École d'Ingénieurs en Électronique et Informatique à Angers a effectué un stage au labo d'Elektor.
- 80 Rétronique : générateur AM-HF TF801D/1**  
Le TF801D/1 de Marconi Instruments est un générateur HF (10 MHz à 470 MHz), produit aux environs de 1960.
- 84 Hexadoku**
- 85 circuits intégrés pour la mesure d'énergie**  
La question de la puissance et de la forme d'onde se pose, que ce soit pour un luminaire ou un gros moteur, pour un onduleur ou un panneau solaire qui fournit de l'énergie au réseau électrique.
- 86 avant-première**  
Pendant que vous lirez ce numéro, nous préparons les suivants. Vos idées, vos suggestions, vos propres schémas sont les bienvenus.




## Notre équipe

Rédacteur en chef :	Denis Meyer (redaction@elektor.fr)	Directeur éditorial :	Wisse Hettinga
Rédaction internationale :	Harry Baggen, Jan Buiting, Thijs Beckers, Eduardo Corral, Jens Nickel, Clemens Valens		
Laboratoire :	Thijs Beckers, Ton Giesberts, Luc Lemmens, Raymond Vermeulen, Jan Visser		
Graphistes :	Giel Dols, Jeanine Opreij, Mart Schroijen	Secrétaire de rédaction :	Hedwig Hennekens
Directeur <i>Elektor online</i> :	Daniëlle Mertens		
Ont coopéré à ce numéro :	Gerry Atrix, Jean-Paul Brodier, Robert Grignard, Hervé Moreau, Kévin Petit,, NN		

## Nos réseaux



## Nos équipes internationales

 <b>United Kingdom</b> Wisse Hettinga +31 (0)46 4389428 w.hettinga@elektor.com	 <b>Spain</b> Eduardo Corral +34 91101 93 95 e.corral@elektor.es	 <b>India</b> Sunil D. Malekar +91 9833168815 ts@elektor.in
 <b>USA</b> Hugo Vanhaecke +1 860-875-2199 h.vanhaecke@elektor.com	 <b>Italy</b> Maurizio del Corso +39 2.66504755 m.delcorso@inware.it	 <b>Russia</b> Nataliya Melnikova +7 (965) 395 33 36 Elektor.Russia@gmail.com
 <b>Germany</b> Ferdinand te Walvaart +31 46 4389417 f.tewalvaart@elektor.de	 <b>Sweden</b> Wisse Hettinga +31 46 4389428 w.hettinga@elektor.com	 <b>Turkey</b> Zeynep Köksal +90 532 277 48 26 zkoksal@beti.com.tr
 <b>France</b> Denis Meyer +31 46 4389435 d.meyer@elektor.fr	 <b>Brazil</b> João Martins +55 11 4195 0363 joao.martins@editorialbolina.com	 <b>South Africa</b> Johan Dijk +27 78 2330 694 / +31 6 109 31 926 j.dijk@elektor.com
 <b>Netherlands</b> Harry Baggen +31 46 4389429 h.baggen@elektor.nl	 <b>Portugal</b> João Martins +351 21413-1600 joao.martins@editorialbolina.com	 <b>China</b> Cees Baay +86 21 6445 2811 CeesBaay@gmail.com

35<sup>ème</sup> année, n°413 novembre 2012  
ISSN 0181-7450 Dépôt légal : octobre 2012  
CPPAP 113 U 83713

ELEKTOR / PUBLITRONIC SARL  
c/o Regus Roissy CDG - 1, rue de la Haye - BP 12910  
FR - 95731 Roissy CDG Cedex - France  
Tél. : (+33) 01.49.19.26.19 - Fax : (+33) 01.49.19.22.37  
www.elektor.fr

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

DROITS D'AUTEUR : © 2012 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



## Nos adhérents

Nous  
avons

264880

adhérents  
dans 83 pays

Adhérez vous aussi au réseau Elektor...

**ElektorHebdo en ligne**

notre lettre hebdomadaire d'information électronique gratuite et à ...

**Elektor, le mensuel d'électronique**










Tarifs, conditions, offres, nouveautés :

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

## Vos correspondants

**Nous sommes à votre service pour toute question relative à votre commande ou votre abonnement par téléphone (+33) 01.49.19.26.19 ou par courriel : [service@elektor.fr](mailto:service@elektor.fr)**

## Nos annonceurs

 Beta Layout <a href="http://www.beta-layout.com">www.beta-layout.com</a> . . . . . 15	 Jackaltac <a href="http://www.jackaltac.com">www.jackaltac.com</a> . . . . . 71
 Cadsoft <a href="http://www.element14.com">www.element14.com</a> . . . . . 88	 Microchip <a href="http://www.microchip.com">www.microchip.com</a> . . . . . 43
 DesignSpark <a href="http://www.designspark.com">www.designspark.com</a> . . . . . 27	 MikroElektronika <a href="http://www.mikroe.com">www.mikroe.com</a> . . . . . 3
 Eurocircuits <a href="http://www.elektorpcbservice.com">www.elektorpcbservice.com</a> . . . . . 63	 Reichelt <a href="http://www.reichelt.fr">www.reichelt.fr</a> . . . . . 21
 ForumLED <a href="http://www.forumled.com">www.forumled.com</a> . . . . . 42	 MiniPlacer <a href="http://www.miniplacer.com">www.miniplacer.com</a> . . . . . 15

**Pour placer votre annonce dans le prochain numéro d'Elektor**

veuillez contacter Mme Ilham Mohammédi par téléphone au (+33) 01.49.19.26.19  
ou par courrier électronique : [i.mohammedi@elektor.fr](mailto:i.mohammedi@elektor.fr)

dans laquelle elles sont incorporées (Loi du 11 mars 1957 -art. 40 et 41 et Code Pénal art. 425).

Certains circuits, dispositifs, composants, etc. décrits dans cette revue peuvent bénéficier de droits propres aux brevets; la Société éditrice n'accepte aucune responsabilité du fait de l'absence de mention à ce sujet. Conformément à l'art. 30 de la Loi sur les Brevets, les circuits et schémas publiés dans Elektor ne peuvent être réalisés que dans des buts privés ou scientifiques et non commerciaux. L'utilisation des

schémas n'implique aucune responsabilité de la part de la Société éditrice. La Société éditrice n'est pas tenue de renvoyer des articles qui lui parviennent sans demande de sa part et qu'elle n'accepte pas pour publication. Si la Société éditrice accepte pour publication un article qui lui est envoyé, elle est en droit de l'amender et/ou de le faire amender à ses frais; la Société éditrice est de même en droit de traduire et/ou de faire traduire un article et de l'utiliser pour ses autres éditions et activités, contre la rémunération en usage chez elle.

Elektor est édité par Elektor International Media B.V.  
Siège social : Allee 1 - 6141 AV Limbricht, Pays-Bas

Imprimé aux Pays-Bas par Senefelder Misset - Doetinchem

Distribué en France par M.L.P. et en Belgique par A.M.P.



# elektor

Par vagues rapprochées, les innovations de l'électronique inondent le monde. Pour se maintenir au sommet de ces déferlantes et pour rester digne de la confiance que vous lui accordez, Elektor aussi se renouvelle. Dans le numéro du mois prochain, nous vous servirons les fruits de cette revitalisation en cours. L'âme du projet reste le service de qualité assuré à nos lecteurs, aux visiteurs de notre site et à nos clients. Tous bénéficieront de la même expertise d'Elektor qui renforcera sa position de portail et de plateforme dynamique pour les électroniciens les plus exigeants du monde entier.

Don Akkermans, Directeur  
Wisse Hettinga, Éditeur en chef international

---

## L'électronique change le monde, Elektor change l'électronique. Pour vous, avec vous !

---

Aujourd'hui, Elektor est déjà *bien plus* qu'un magazine, même si beaucoup de membres de la communauté Elektor sont attachés au papier imprimé & agrafé et à sa cadence mensuelle. Les uns l'achètent en kiosque, les autres, de plus en plus nombreux, s'y abonnent. Que vous nous lisiez depuis plus de 25 ans ou depuis quelques mois, vous appréciez cette parution qui rythme votre vie de professionnel ou de passionné d'électronique. Et nous ne nous lasserons pas de relever chaque mois le défi qui consiste à essayer de plaire à tous : débutants et experts, étudiants et enseignants, amateurs et professionnels.

Elektor n'a pas cessé de se diversifier, soit pour répondre à la demande soit parce qu'il fallait anticiper. Vous aurez sans doute visité au moins une fois notre site, acheté un circuit imprimé dans notre e-shoppe ; dans votre bibliothèque technique il y a certainement un ou plusieurs volumes de la collection *30x circuits*. Votre nom ne figurerait-il pas déjà parmi les participants à nos webinaires en ligne ou à nos concours de conception internationaux ? Vous ne seriez pas justement l'auteur de l'un des projets publiés par Elektor ?

L'industrie de l'électronique ne dédaigne pas ce qui se passe chez nous. Des fabricants d'envergure cherchent activement à coopérer avec Elektor et sa communauté : test bêta, échantillonnage, concours de conception, échange d'information... toutes les formes de coopération sont envisagées.

D'autre part, des électroniciens de plus en plus nombreux découvrent Elektor autrement que sous la forme du magazine qui nous a valu notre réputation. Songez un instant que par

l'intermédiaire d'Elektor vous êtes connecté à un réseau de quelque 300.000 électroniciens dans 80 pays.

En résumé, grâce à l'internet, Elektor est devenu ce qu'il était déjà : une communauté d'aficionados de l'électronique. En faisant sauter le carcan imposé par le seul papier, nous allons donner à cette communauté l'ampleur qu'elle mérite. Pour marquer le coup, abandonnons la terminologie liée au papier. Désormais, vous n'êtes plus ni lecteur ni abonné, mais membre d'une communauté : *elektor.community*. Et voici de quoi elle est composée.

**elektor**  **labs**

Un nouveau site accueille vos propositions et vous permet de suivre la progression des projets de vos pairs, jusqu'à leur aboutissement sous forme de réalisations reproductibles.

Une version bêta de ce site est en ligne depuis quelques mois, sous le nom *elektor-projects*. Il devient *elektor.labs* où palpitera désormais le cœur d'Elektor. C'est là que s'élabore la connaissance, là que se partage et s'acquiert le savoir.

**elektor**  **community**

Le portail [www.elektor.com](http://www.elektor.com) devient le point d'ancrage d'*elektor*.

# (beaucoup) plus qu'un magazine



**community**, où se croisent les voies qui mènent à l'e-choppe [elektor.store](#), aux archives [elektor.files](#) et aux forums. L'internaute retrouvera les guichets auxquels il pourra, selon ses besoins, télécharger les PDF des articles, commander des composants, des livres ou des kits. Bien entendu, il reste possible d'accéder directement à chacune de ces sections.

## elektor magazine

Désormais, le magazine imprimé s'appellera [elektor.magazine](#). Il paraîtra dix fois par an. La mise en page rafraîchie, le logo redessiné et les nouvelles rubriques témoigneront de la revitalisation annoncée, mais les schémas et les réalisations soigneusement décrites comme nulle part ailleurs demeurent l'essentiel de ce qu'Elektor aura à offrir. Au numéro double de l'été répondra, au début de l'année, une deuxième édition double. Le magazine reste téléchargeable sous forme de PDF.

## elektor post

Notre lettre d'information électronique [ElektorHebdo](#) s'inscrit dans un service de communication directe, élargi et amplifié, appelé [elektor.post](#), où le mot anglais *post* est pris dans son sens moderne de message posté sur l'internet. Ce service comportera par exemple l'envoi bimensuel par courrier électronique de schémas électroniques pour enrichir votre collection.

### Pour qui ?

L'abo conventionnel est remplacé par une carte de membre qui donne droit à :

- dix numéros par an du magazine Elektor restructuré autour du noyau classique de circuits et de réalisations, enrichi par de nouvelles rubriques et d'autres éléments inédits
- dont deux numéros doubles d'Elektor par an
- le contenu de toute l'année sous forme numérique
- une remise permanente de 10 % au moins sur TOUS les produits et les services offerts dans [elektor.store](#)
- 10 % de remise (au moins) sur tous les circuits imprimés d'Elektor
- l'accès direct à [elektor.labs](#)
- les publications occasionnelles d'[elektor.post](#) par voie de courrier électronique
- une carte de membre GOLDCARD avec clé USB intégrée sur laquelle se trouve le contenu de toute l'année sous forme numérique

Il y aura également une carte de membre dite GREENCARD, avec les mêmes services et les mêmes avantages mais exclusivement sur l'édition numérique d'Elektor (sans papier).

Quand votre abonnement actuel arrivera à expiration, vous recevrez une notification personnelle dans laquelle tous ces bonus seront décrits en détail, ainsi qu'une proposition qui ne se refuse pas.



# elektor labs: visite guidée

Salut à tous ! Je vois qu'il y a du monde aujourd'hui pour la visite d'**elektor.labs**.

*Cool.* En temps normal, pour visiter le Saint des Saints, il faut une carte de membre.

Ou une invitation. Profitez-en aujourd'hui, car c'est portes ouvertes.

Mon nom est Clemens et je suis votre guide. Suivez-moi et n'hésitez pas à poser des questions.

elektor.labs, c'est là que ça se passe. Dot-labs, comme ça se prononce, c'est la pompe à idées de votre communauté électronique préférée, plus connue sous le nom d'Elektor. Le pouvoir d'attraction des génies qui travaillent dans .labs vaut celui de nos plus charmantes secrétaires, mais eux, ils ont en plus la connaissance de l'électronique. À ma droite (et donc sur votre gauche) voici Ton G. avec le g de... ? Génial, bien sûr ! Lui-même est modeste et préférerait qu'on l'ignore, mais à cause de ses talents de concepteur dans le domaine des amplificateurs audio, son expérience et sa connaissance, tout le monde le harcèle de questions. Essayez de vous retenir... et suivez-moi, par ici. Au milieu, c'est le domaine de Luc. On l'appelle Lucky Luc parce que ses projets marchent toujours et qu'ils sont *cool*. Ne me demandez pas comment il fait, je serais

bien embêté pour répondre. Demandez-le lui, en français si vous voulez, il se débrouille bien dans notre langue. Luc a un côté *lone-some cowboy*, et comme le héros de BD il est charmant et galant. Il connaît une foule de trucs précieux en électronique, il vous en révélera peut-être l'un ou l'autre.

Permettez-moi maintenant de vous présenter Raymond la Science. Il mérite bien son illustre surnom, car il est toujours en train de compulser des tonnes de doc à la recherche de composants et de techniques nouvelles, il veut tout apprendre et, tel une éponge, des pieds à la tête il est imbibé d'information. Pas besoin d'insister longtemps pour qu'il la partage avec vous. Essayez, mais pas en français, sinon il va encore se croire obligé de se mettre au français.





[www.elektor-labs.com](http://www.elektor-labs.com)



Et voici Sunil, Sunil et Sunil. Non seulement ils sont Indiens, mais ils sont **en** Inde. Leur contribution orientale donne à nos projets électroniques une note épiciée. Pour que ça reste intéressant, rien de tel que le coup de fouet d'un goût bien relevé. Maintenant que vous connaissez le chemin qui mène à .labs, vous reviendrez et vous pourrez entrer en contact avec nos trois Sunil quand bon vous semblera. Laissez-leur un commentaire sur le site.

Si vous pensiez qu'elektor.labs n'est qu'une obscure officine dans la cave d'un château, vous aviez tout faux ! elektor.labs c'est *beaucoup* de pièces pas forcément bien éclairées, rarement bien rangées, dans beaucoup de garages, de greniers, de studios, au sein de nombreuses sociétés, écoles, universités et bien davantage, dans le monde entier. elektor.labs n'est pas localisable géographiquement. C'est un concept d'envergure mondiale.

Il y a Ivan à Saint-Petersbourg, en Russie, qui développe des applications à microcontrôleurs pour le sport. Il y a Gina et Nina à Lima, au Chili, qui sont folles d'électronique vestimentaire. Elles ont conçu un poncho qui... pardon, que dites-vous ? Lima est au Pérou ? Oh, oui bien sûr, et vous qui êtes-vous ? Païkan... et vous êtes... ? Japonais. Bonjour Païkan. Et à part les capitales, vous avez d'autres passions ? Celle de l'électronique peut-être ? Ah, vous cherchez de l'aide pour la conception d'un anémomètre sans pièces mobiles ? Excellent. Vous devriez passer par le site elektor.labs et y publier votre projet. Donnez une brève description de ce que vous imaginez et ajoutez quelques mots pour dire ce que vous attendez des autres. N'oubliez pas les illustrations, photos et dessins.



elektor.labs doit rassembler. Les membres qui partagent le même intérêt pour tel ou tel sujet pourront s'entraider. Ensemble nous créerons, mettrons au point et concrétiserons des réalisations simples ou

complexes. D'ailleurs, il me revient à l'esprit que j'ai vu un anémomètre sans pièce mobile sur .labs. Allez-y. Le projet m'a semblé intéressant. Comme tant d'autres, sur tant de sujets.

Sur le site elektor.labs, les visiteurs sont invités à évaluer les projets. Cette participation du plus grand nombre est cruciale pour orienter le choix des sujets. Il suffit de cliquer sur les étoiles. Plus une proposition sera populaire, plus la rédaction d'Elektor sera

encline à en faire un article en coopération avec son auteur. Et là ça devient lucratif pour vous. Si votre projet est publié par Elektor, vous êtes rémunéré ! Non seulement vous aurez votre nom dans toutes les éditions d'Elektor, mais vous pourrez aussi inviter votre moitié à dîner sans avoir, pour une fois au moins, à vous préoccuper de la colonne de droite sur la carte des vins. Si vous craignez d'être reconnu dans la rue, vous aurez de quoi vous acheter des lunettes de star. *Get published, get famous, get rich !*

La rédaction ne retiendra pas nécessairement que des montages très étoilés. Tout le monde n'est pas fan de Björk, des Spice Girls ou des Beatles. Nos éditeurs eux-mêmes ont des goûts variés. Et savent repérer les perles que les autres ne voient pas (encore). Ils sauront trouver la vôtre.



Parmi les nombreuses formes que prendra l'électronique dans la communauté d'Elektor, il y a les concours et les échantillons gratuits. Il ne vous a pas échappé qu'Elektor s'est associé à des opérations de grande envergure avec des partenaires susceptibles de devenir des parrains généreux. C'est précisément ce qui est en train de se passer avec le PSoCaMorph, projet de synthétiseur musical, dont les participants peuvent gagner un kit de développement PSoC 5 simplement en se montrant actifs dans l'élaboration du projet. Celui-ci est conduit par le fameux expert du filtrage, Kendall Castor-Perry. Si la synthèse sonore vous intéresse, vous devriez vous frotter au projet PSoCaMorph !

Voilà, chers visiteurs, vous aurez compris qu'elektor.labs est bien plus qu'un laboratoire d'électronique ordinaire. C'est une plateforme sans limites sur laquelle nous laissons libre cours à notre passion. Quand vous aurez votre propre clé pour entrer à elektor.labs vous y passerez quand vous voudrez et resterez le temps qu'il vous plaira. Pour discuter, interroger, fouiner, comparer, proposer... ne vous gênez pas.

C'est la fin de votre visite guidée d'elektor.labs. N'oubliez pas le guide. Merci.

Ah, une dernière chose. Quand vous serez le dernier à partir, le soir, ne vous souciez pas d'éteindre la lumière et le fer à souder. elektor.labs est ouvert 24/24 même le dimanche et les jours fériés.

(120608-I)



# retour aux sources (9)

## oscillateurs sinusoïdaux

Les possibilités de l'électronique ne seraient pas ce qu'elles sont si nous ne pouvions traiter que les signaux venus de l'extérieur, sans pouvoir créer – à première vue ex nihilo – des oscillations. Les oscillateurs sont les éléments clés de nombreux appareils et le socle d'une grande variété d'applications. Ce sont eux qui permettent de créer des sons, mais aussi des signaux pour tester certains circuits.

Burkhard Kainka (Allemagne)

### Oscillateurs RC

Qui n'a pas déjà eu les oreilles agressées par un effet Larsen ? Ce sifflement provient d'une réaction acoustique entre un haut-parleur et un microphone. Le seul moyen d'arrêter ces sifflements, qui par ailleurs peuvent être de différentes hauteurs, est d'éloigner le micro du haut-parleur, ou de diminuer l'amplification.

Toute rétroaction suffisamment forte conduit en principe à des oscillations. La réaction peut également être de nature électrique, un signal de sortie étant dans ce cas appliqué de façon rétro-active sur l'entrée. La réaction ne se produit que pour un certain déphasage, identique à celui requis par un amplificateur à deux étages.

Le circuit de la **figure 1** fonctionne comme un multivibrateur, si ce n'est qu'ici la réaction est réglable. Outre les signaux rectangulaires du multivibrateur, nous pouvons

donc aussi délivrer des signaux sinusoïdaux ou d'autres formes. Le potentiomètre permet d'ajuster la réaction jusqu'à ce qu'apparaissent de faibles oscillations, le plus souvent sinusoïdales.

Un unique transistor fait aussi l'affaire, bien que celui-ci déphase le signal de 180 degrés. On peut le redresser à l'aide de combinaisons RC placées à la queue leu leu. L'oscillateur à réseau déphaseur de la **figure 2** produit un signal sinusoïdal d'environ 800 Hz. Idéal comme oscillateur d'entraînement au morse, il fera aussi un bon générateur de signaux pour tester un amplificateur BF. L'oscillateur à réseau déphaseur fonctionne tout aussi bien avec un transistor à effet de champ BS170. Le circuit de la **figure 3** a été conçu pour être à haute impédance, et délivre une fréquence de 10 Hz. Son courant de service n'est que de 30  $\mu$ A.

### Oscillateur en anneau

Jusqu'à présent nous n'avons utilisé qu'un ou deux étages. Qu'en est-il d'un montage à

émetteur commun à trois étages ? Avec un déphasage total de 180 degrés, vous pensez sans doute que la réaction devient une contre-réaction. Le circuit produit pourtant bel et bien des oscillations (**fig. 4**). Leurs fréquences augmentent avec la tension de service et peuvent atteindre 1 MHz.

Comment expliquer ces oscillations ? Nous sommes en présence d'un amplificateur à trois étages à contre-réaction doté d'un gain en tension très élevé. Chaque étage provoque toutefois un léger retard dont le cumul entraîne, à une certaine fréquence, un déphasage supplémentaire de 180 degrés. La contre-réaction devient ainsi réaction à ladite fréquence, et les oscillations apparaissent (**fig. 4**).

Si l'on souhaite utiliser ce circuit non pas comme oscillateur mais comme amplificateur sensible, il faut donc à tout prix éviter l'apparition de la réaction. Il est toutefois difficile d'empêcher les oscillations propres du circuit avec une amplification aussi grande.

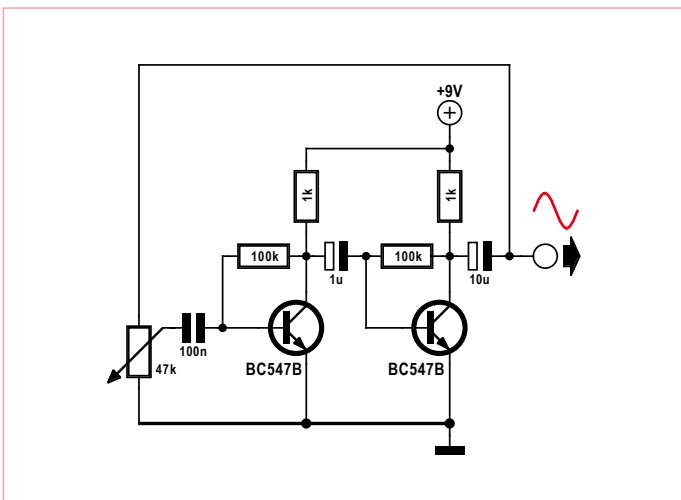


Figure 1. Comment créer des oscillations à l'aide d'une réaction.

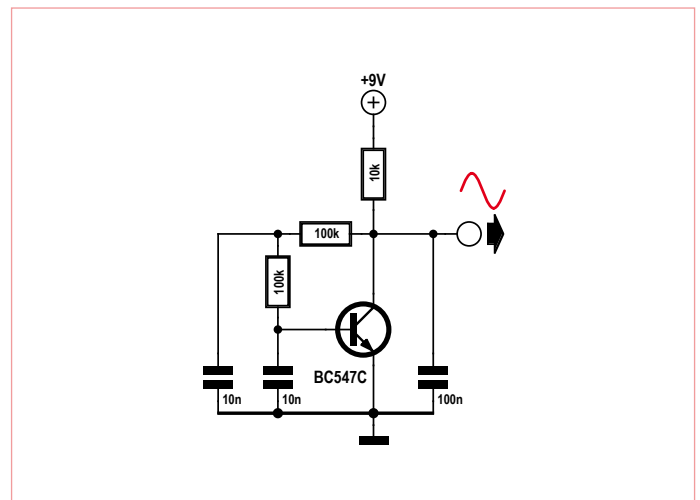


Figure 2. Oscillateur avec décalage de phase.

## L'effet Miller

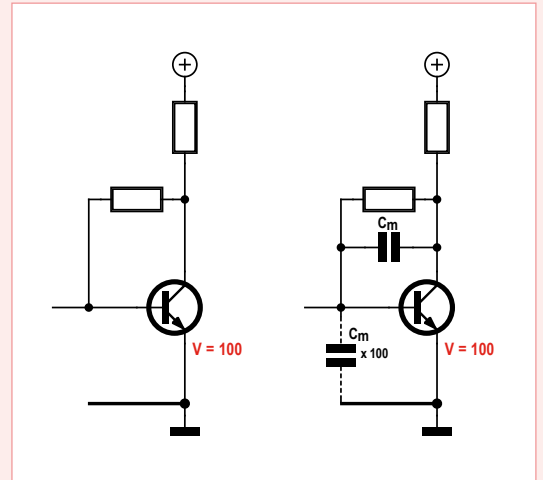
Le gain typique d'un montage amplificateur à émetteur commun est d'environ 100. Cet ordre de grandeur prévaut pour les hautes fréquences, mais le gain n'est parfois pas aussi élevé que ce qui était souhaité. Si la fréquence de transition d'un BC547 se situe vers 300 MHz (à cette fréquence le gain en courant est tombé à 1), la fréquence de coupure de ce circuit amplificateur est néanmoins beaucoup plus basse. L'effet sera d'autant plus prononcé que les résistances du circuit seront grandes. Les capacités de transition internes du transistor jouent ici un rôle prépondérant.

La capacité base-collecteur  $C_{bc}$ , bien que sur le BC547 elle ne soit que de 5 pF, joue ici un rôle crucial. L'inversion de la polarité du condensateur de Miller  $C_m (= C_{bc})$  entre l'entrée et la sortie de l'amplificateur inverseur intervient en effet des deux côtés à la fois. Si par exemple le potentiel de la base s'élève de 1 mV, le potentiel du collecteur baisse de 100 mV, ce qui implique une multiplication par 100 de la charge.

Tout se passe en somme comme si un condensateur était présent à l'entrée, dont la capacité était plus grande que la capacité de Miller d'un facteur égal au gain en tension, soit ici une capacité d'environ 500 pF. C'est ce que l'on appelle l'effet Miller. En fonction de la valeur de la résistance interne de la source de signal reliée au circuit, il se forme un filtre passe-bas qui annihile la fréquence de coupure haute.

Dans le cas d'un amplificateur, cela signifie qu'avec une fréquence de coupure haute le circuit doit être dimensionné pour avoir l'impédance la plus basse possible. Dans certains cas il peut être judicieux de travailler avec un gain en tension plus petit, p. ex. en choisissant une résistance de sortie moins élevée. On peut également exploiter les transistors HF qui possèdent de très faibles capacités de transition.

Pour ce qui est des oscillateurs, la capacité Miller permet de les construire sans recourir à des condensateurs fixant la fréquence, puisque le transistor contient déjà leur équivalent.



Un oscillateur à trois étages est plus facile à construire qu'un amplificateur à trois étages. Plus le courant de collecteur moyen est faible, plus l'impédance du circuit est haute, et plus les capacités internes du transistor ont de l'influence. Une petite tension de service aura donc pour résultat des retards plus grands et une fréquence plus basse.

Les étages amplificateurs de ce circuit forment un anneau, d'où son nom d'oscilla-

teur en anneau. Il fonctionne également avec cinq, sept ou neuf étages, sous réserve qu'une contre-réaction soit présente si la tension est continue. Un nombre pair d'étages forment en revanche toujours une bascule statique.

Un oscillateur en anneau à trois étages peut posséder une très haute impédance, et ainsi être alimenté de façon économique : avec trois résistances de collec-

teur de 1 M $\Omega$ , il fonctionne dès 0,5 V et a besoin de moins de 1  $\mu$ A. Une photodiode BPW34 utilisée comme cellule solaire suffit donc à son alimentation (fig. 5). On mesure à sa sortie une fréquence de 5 kHz. Comme cette fréquence augmente avec la luminosité, vous pourriez sans doute également vous servir de ce circuit comme d'un capteur de lumière. Comment ça seulement 5 kHz sans condensateur ? Curieux quand on sait que les capa-

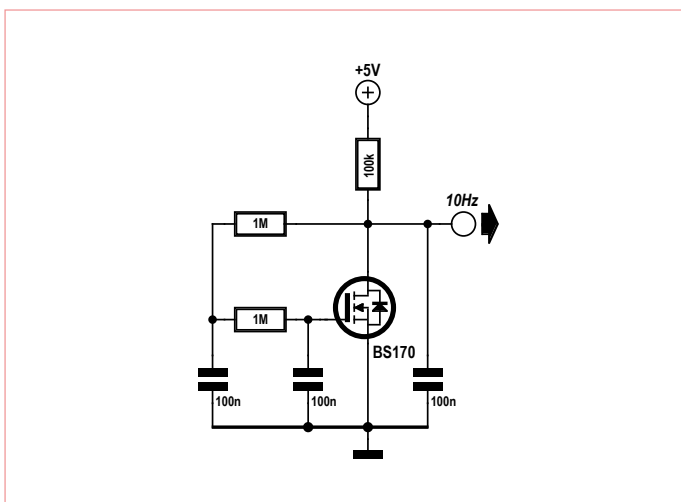


Figure 3. Oscillateur à réseau déphaseur avec FET.

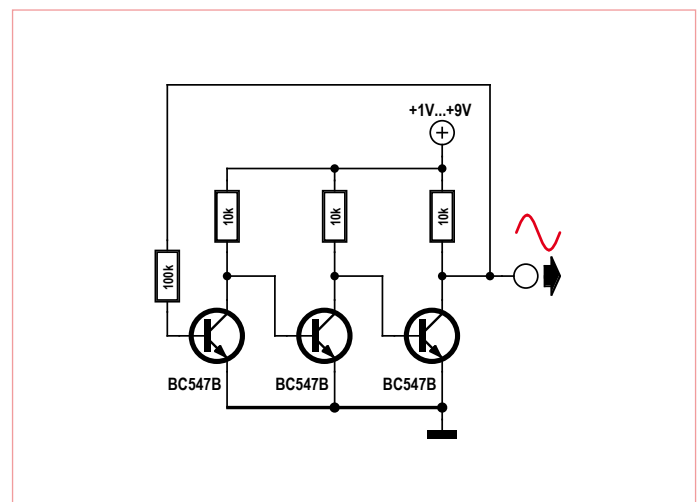
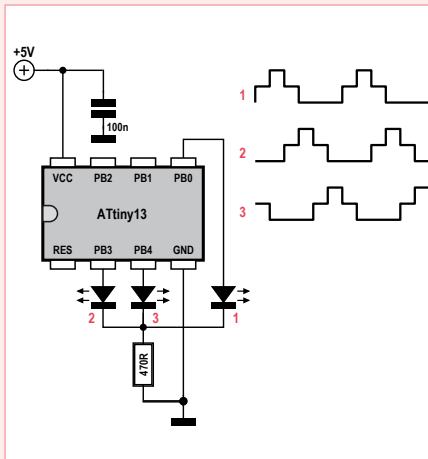


Figure 4. Un oscillateur sans condensateur.



## Clignotant triphasé



Utiliser un microcontrôle est un autre moyen de créer facilement des signaux (ou courants) triphasés. Pour cela nous répar-tissons six points de commutation le long

d'un intervalle temporel. Bien que numé-rique, le résultat avec trois LED est sem- blable à celui obtenu avec un oscillateur en anneau triphasé. Particularité du circuit, une unique résistance-talon partagée par les LED. Conséquence, chaque LED fonctionne avec deux niveaux de luminosité. Une LED allumée peut être dans trois configurations : dans la première elle brille en même temps que celle qui est à sa gauche, et partage avec elle le même courant. Même chose dans la deuxième, mais le duo se fait cette fois-ci avec sa voisine de droite. Elle brille seule dans la configuration intermédiaire, et reçoit donc tout le courant. Chaque LED prend donc les états : duo, solo, duo, éteinte, éteinte, éteinte. Et ceci avec chaque fois un déphasage de 180 degrés par rap- port à sa voisine.

```
'Clignotant triphasé, 1500 ms,
0,67 Hz
$regfile = "attiny13.dat"
$crystal = 1200000
Config Portb = Output
```

```
Do
  Portb.0 = 1
  Waitms 250
  Portb.3 = 0
  Waitms 250
  Portb.4 = 1
  Waitms 250
  Portb.0 = 0
  Waitms 250
  Portb.3 = 1
  Waitms 250
  Portb.4 = 0
  Waitms 250
Loop
End
```

cités entre électrodes d'un transistor n'ex- cèdent pas quelques picofarads, non ? Le phénomène en jeu est ici l'effet Miller (voir encadré) : la capacité base-collecteur se comporte en entrée comme si elle était multipliée par le gain en tension. Ce même effet multiplicateur permet d'abaisser considérablement la fréquence en ajoutant

un ou plusieurs condensateurs entre la base et le collecteur (fig. 6). Les trois condensateurs de 100 nF donnent une fréquence d'environ 1 Hz.

### Clignotant à LED triphasé

Des fréquences aussi basses permettent de créer des effets de lumière attrayants. Les

trois LED du circuit de la figure 7 créent ainsi un clignotement d'ensemble agréable. Chaque LED de cet oscillateur triphasé s'al- lume avec un certain temps de retard. Le courant qui les traverse est pratiquement sinusoïdal, d'où des transitions en douceur. Selon la nature de l'alimentation, directe- ment sous 9 V ou par un courant bridé par

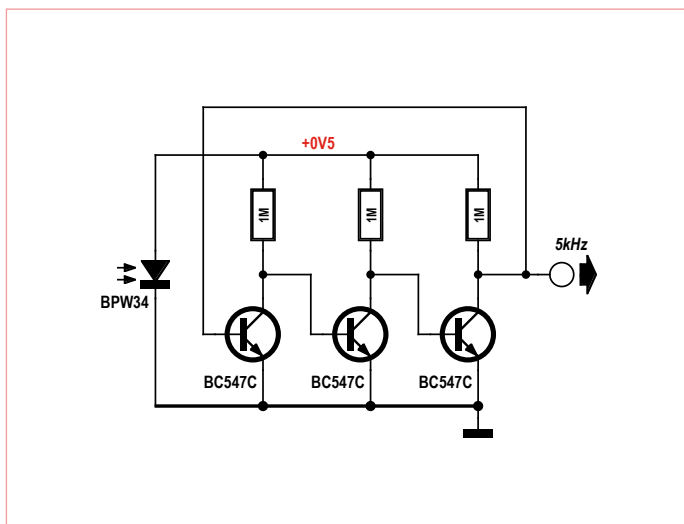


Figure 5. Une cellule solaire pour alimenter l'oscillateur en anneau.

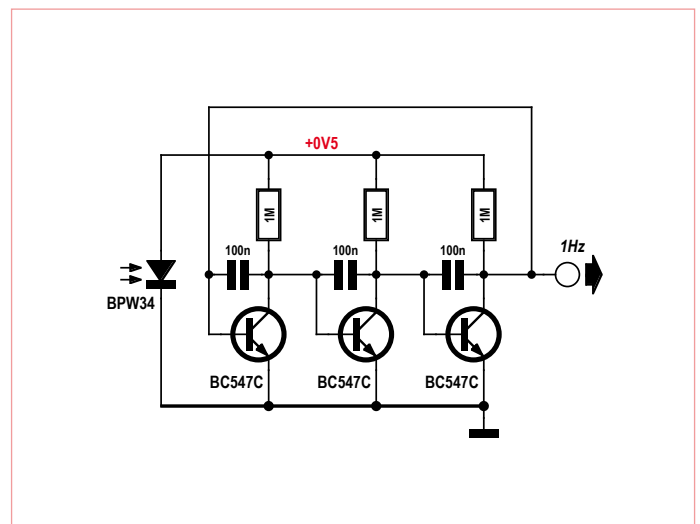


Figure 6. Un oscillateur en anneau plus lent et plus économique.

## Solution du jeu-concours de septembre

La solution était le code CFG :

### Solution 1 :

La durée d'une période entière est :  $T = 2 \times 0,65 \text{ ms} = 1,3 \text{ ms}$ .  
 $f = 1 / T = 1 / 1,3 \text{ ms} = 0,769 \text{ kHz}$ , soit environ 770 Hz. La bonne réponse était donc la C).

### Solution 2 :

La tension de base chute moins profondément et atteint plus rapidement +0,6 V, donc la fréquence augmente : réponse F).

### Solution 3 :

Le courant d'émetteur doit être plus fort, donc la résistance d'émetteur plus faible. Ainsi le courant qui traverse le haut-parleur augmentera lui aussi. G) était donc le bon choix.

le potentiomètre, le clignotement se montre vif et rapide, ou pâle et lent. Car, encore une fois, la fréquence dépend ici aussi étroitement du courant de fonctionnement.

(120009 – version française : Hervé Moreau)

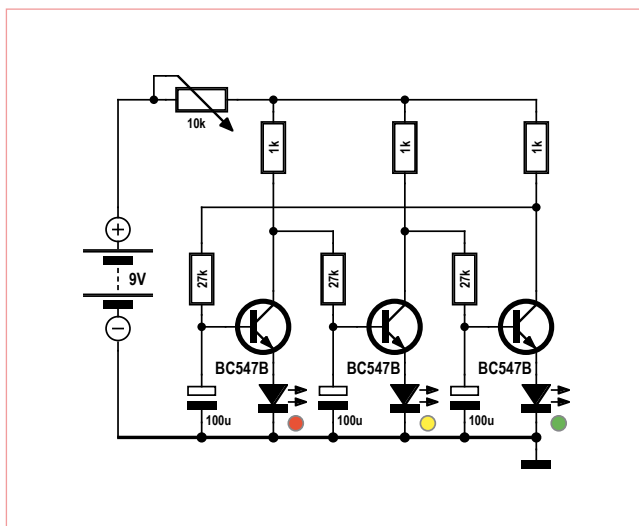


Figure 7. Effets de lumière triphasés.



L'ORIGINAL DEPUIS 1994  
**PCB-POOL**  
 Beta LAYOUT



### Pochoir CMS gratuit

avec chaque commande  
"Prototype"



### EAGLE : Bouton de commande

[pcb-pool.com/download-button](http://pcb-pool.com/download-button)  
 sur votre première commande de PCB

**20% de remise**

Appel Gratuit : FR 0800 90 33 30  
[sales@pcb-pool.com](mailto:sales@pcb-pool.com)

[www.pcb-pool.com](http://www.pcb-pool.com)

**Beta**  
 LAYOUT  
 create.electronics

## SolderTech



*SolderTech est un distributeur de pâte à braser économique, simple et efficace.*

*Très facile à utiliser, il est complet et prêt à l'emploi. Il comprend la source d'air comprimé, et ne nécessite qu'une prise de courant.*

**Plus d'informations sur notre site [www.MiniPlacer.com](http://www.MiniPlacer.com)**

MiniPlacer - 302, avenue de Neuville - 78950 Gambais  
[www.miniplacer.com](http://www.miniplacer.com) / tel : 01 34 87 19 05

# VU-mètre à Nixie

## indicateur de niveau nostalgique pour ampli à tubes

Quoi de plus approprié, pour rehausser un amplificateur à tubes, qu'un affichage lumineux aux accents rétro ?

Ce circuit est un étonnant cocktail qui allie un PSoC de Californie à deux Nixie d'Ukraine... et du passé !



Harald W. Cintra (Brésil)

À notre époque, les distances et les frontières semblent s'évaporer. En irait-il de même des générations de composants électroniques ? Oui, si l'on en croit cet article publié en 2012, qui associe deux composants séparés par 45 ans d'évolution. D'un côté un PSoC (*Programmable System on Chip*) et de l'autre, le tube Nixie IN-9. Voyons comment collaborent un pur produit *Sovjet-elektronisk* et un zeste de technologie de la *Silicon-Valley* à l'âge d'internet et sous l'égide d'Elektor.

### Pourquoi un IN-9 ?

La lumière rose orangé qu'émet ce tube à décharge gazeuse fait de l'IN-9 un candidat idéal pour ce circuit. La hauteur de la colonne ardente dépend directement du courant appliqué, avec une tension sur l'anode de 150 V<sub>CC</sub>. Le **figure 1** montre la quasi-linéarité du comportement de l'afficheur. Entre 0 et 8 mA, tout va comme sur des roulettes à environ 10,3 mm par mA. Ensuite, ça part plutôt en banane... Vous pourrez vous procurer ces tubes sur l'internet, ce sont des NOS (*new old stock*) ou vieilleries rutilantes, sur un site d'enchères

## Caractéristiques

- VU-mètre stéréo de type colonne
- 2 tubes linéaires IN-9 à décharge gazeuse
- Couleur d'affichage : rose orangé
- Hauteur de colonne : 85 mm (environ)
- Niveau de ligne audio (environ 0,7 V)
- Processeur Cypress PSoC CY8C27443
- PSoC disponible prêt à l'emploi
- Logiciel de conversion lin/log
- Alimentation 150 VC interne
- Platine compacte (65 x 65 mm)

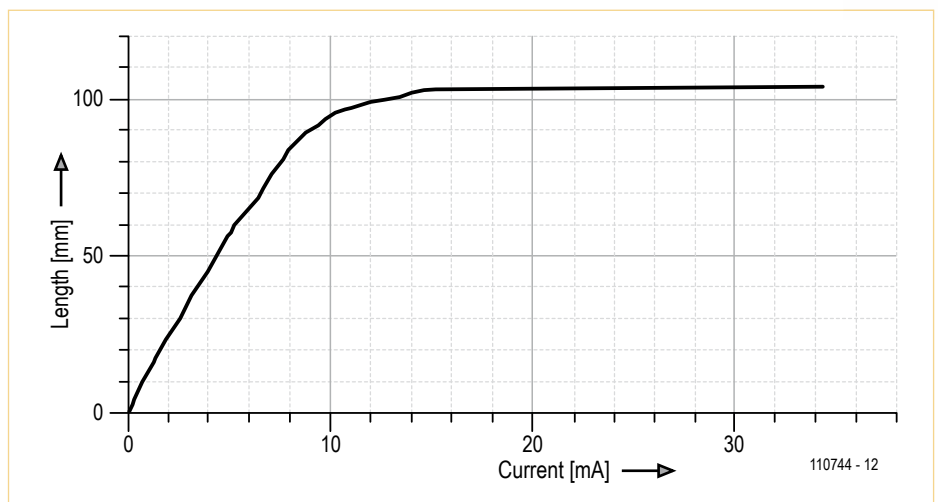


Figure 1. Correspondance mesurée (et non spécifiée par le fabricant) entre la hauteur de la colonne lumineuse et le courant appliqué à un IN-9. Mieux vaut se cantonner à la fourchette 0 à 8 mA. Illustration redessinée à partir d'un original, avec l'autorisation de Jeff Malins (2004).





en ligne bien connu dont nous tirons le nom, situé tout près de la Cypress Corp en Californie.

Elektor compte déjà quelques projets à tubes Nixie comme un thermomètre hygromètre [1] ou une horloge Spoutnik [2], autant d'articles dans lesquels vous trouverez certainement une mine de références utiles pour faire ce

beau voyage dans le passé, au temps des tubes électroniques ou des lampes de Papy. Par comparaison, on imagine difficilement susciter le même émoi chez un électronicien en 2060 avec un PSoC, mais restons optimistes !

N'espérez pas trouver la fiche technique officielle de l'IN-9. Si aucune notice n'accompagne vos tubes, interrogez l'ami Google. La figure 2 expose le colis qui est arrivé au laboratoire d'Elektor (Babouchka non incluse). Outre l'emballage typiquement ukrainien, on y découvre quelques tubes Nixie numériques, un culot et divers composants commandés simultanément sur Ebay (zut, on l'a tout de même dit !).



Figure 2. Tubes Nixie et leur emballage « à la russe ». Ceci dit, toutes les pièces étaient en parfait état de marche.

### Ajoutez un PSoC

Aussi glamour soit-il, un VU-mètre (*volume unit*) sans amplificateur logarithmique ne vaudrait guère mieux qu'un dentiste dans un poulailler. Pour cette application, nous avons choisi la voie logicielle au lieu de la voie matérielle, et avons sélectionné le PSoC CY8C27443 de Cypress, un circuit qu'on ne voit pas souvent dans Elektor.

Le processeur CY8C27443 convient particulièrement bien à ce circuit, en raison notamment de toute la puissance que recèle le processeur SoC et des nombreuses options en A/N, N/A, amplificateurs et autres, ce qui réduit d'autant le nombre de composants externes. À l'exception des redresseurs et des sources de courant, le PSoC contient tous les éléments présentés sur le diagramme de la figure 3. De gauche à droite : amplificateur d'entrée, redresseur, autre ampli, convertisseur A/N, conversion logicielle logarithmique, puis convertisseur N/A et enfin source de courant pour piloter le tube. Le tout en double pour les canaux G et D, aujourd'hui les amplis à tubes sont stéréo !

### Circuit pratique

Le schéma de la figure 4 démontre, si nécessaire, la polyvalence du PSoC qui va nous permettre de réduire drastiquement le nombre de composants nécessaires.

L'alimentation provient d'un adaptateur externe de 12 V<sub>CC</sub>. Nous avons construit un petit inverseur élévateur basé sur un bon vieux NE555 (IC6), câblé en multivibrateur astable et qui utilise l'entrée de commande de tension pour réguler la tension de sortie, afin de produire les 150 V de tension d'anode pour le tube IN-9. Les composants R10, P3 et R13 forment un diviseur de tension qui sert de rétroaction vers l'entrée de la tension de commande du 555 par l'intermédiaire de T4. Nous réglerons l'ajustable pour qu'il délivre 150 V à l'anode de l'IN-9.

Nous utilisons un 7805 (IC7) pour fournir le +5 V au PSoC, puis un second (IC5) pour atteindre les 10 V nécessaires pour les amplificateurs opérationnels, en insérant une diode zener de 5,1 V dans son retour à la masse

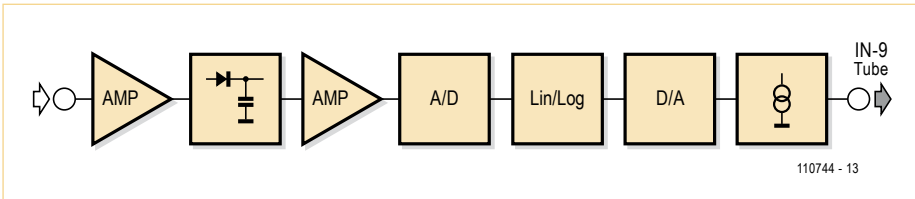


Figure 3. Parcours du signal audio – presque tout se passe à l’intérieur du PSoC.

## 1960’s Выпущено-удулекщтыюл meets 2012 Silicon-Valley tech hype

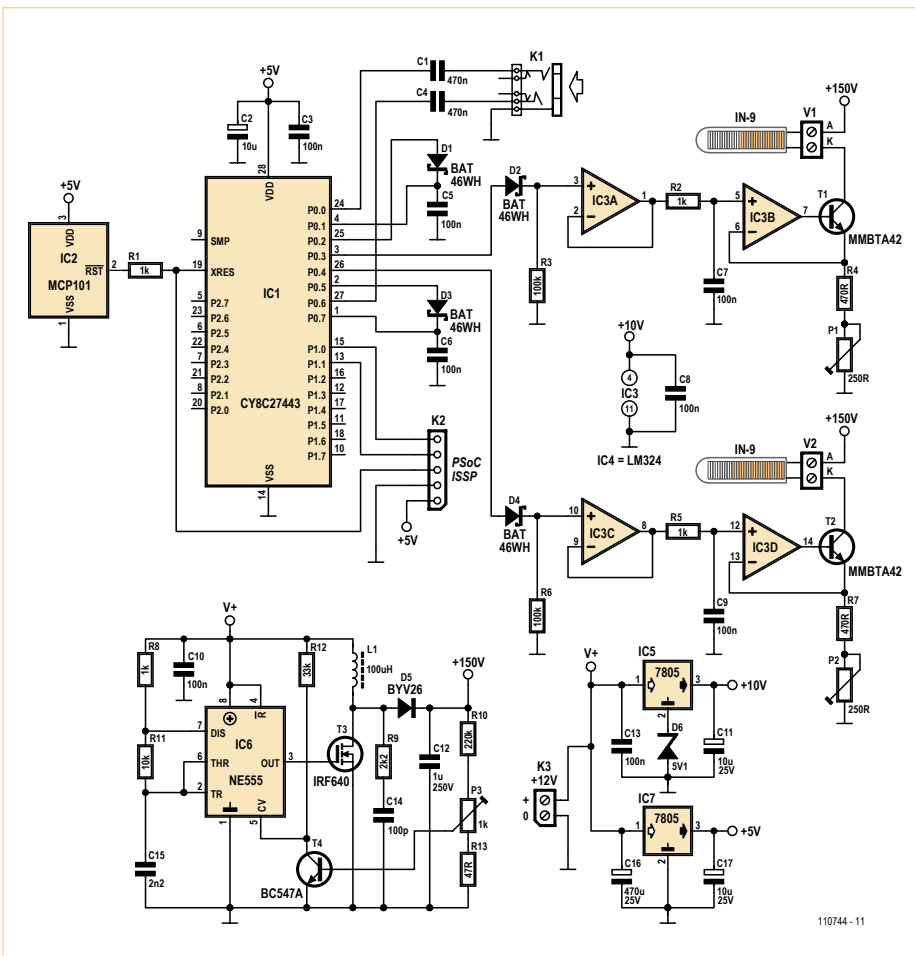


Figure 4. Ce circuit est un bel exemple de l’exceptionnelle polyvalence du PSoC qui joue ici l’homme-orchestre. Reconnaissons qu’il ne servirait strictement à rien en l’absence des tubes Nixie IN-9 et du NE555, deux composants qui datent respectivement des années 60 et 70.

Du 5 V et un signal de mise à zéro suffisent au bonheur d’un PSoC. Pour garantir un signal sûr et certain à la mise sous tension, nous utilisons le générateur dédié MCP101 (IC2).

Du côté du signal audio, l’entrée de niveau ligne (stéréo) passe d’abord par les condensateurs de découplage (C1/C4) pour arriver au PSoC et y être amplifié d’un facteur 2. La sortie de cet amplificateur est acheminée vers une autre broche du PSoC pour redressement par diodes Schottky BAT46WH (faible perte) (D1/D3) avant de retourner dans le PSoC.

Le signal redressé y est légèrement amplifié puis arrive au convertisseur A/N à 8 bits. La valeur convertie transite alors par une table de conversion linéaire à logarithmique avant de rejoindre le convertisseur N/A, toujours à l’intérieur du PSoC. Retour à l’extérieur, le signal est envoyé, via une diode (D2/D4) dans un tampon (IC3A/C), puis traverse un filtre RC (R2/C7, R5/C9), et commande finalement un drain de courant construit autour d’un amplificateur opérationnel LM324 (IC3B/D), un transistor haute tension (T1/T2) et une résistance de réglage (P1+R4, P2+R7). Le but de D2 (D4) est d’établir un seuil de déclenchement pour empêcher les signaux trop faibles d’activer les tubes IN-9. P1 et P2 sont étalonnés pour qu’un signal d’entrée maximal illumine une barre complète.

Pour en revenir au PSoC, la **figure 5** dresse le portrait de son fonctionnement interne. La **figure 6** détaille les amplificateurs d’entrée, PGA\_2 et PGA\_4, ainsi que la sortie des signaux hors du circuit puis transférés via PGA\_1 et PGA\_3 pour les tamponner et enfin dans DUALADC8 où ils sont numérisés. On y aperçoit aussi la sortie du convertisseur N/A, conduite aux bornes de sortie. Le site d’Elektor [3] propose gratuitement les fichiers de conception du PSoC. Cypress offre un grand nombre d’outils logiciels pour programmer en long et en large leurs PSoC, mais pour ceux que l’opération rebute, Elektor [3] propose des CI programmés, prêts à l’emploi.

### Assemblage

Le circuit imprimé du projet est représenté à la **figure 7**. Il s’agit d’un circuit hybride



qui mélange des composants à fils avec des CMS. Il faut monter en premier les CMS. Souder à la main le PSoC et le LM324 n'a rien d'impossible, avec un fer à souder assez fin et un bon entraînement pour assurer la précision nécessaire. Il existe plusieurs techniques, qui ont déjà été décrites de nombreuses fois dans la revue.

Tous les condensateurs électrolytiques sont de type radial et à fils. Faites surtout attention à la polarité au moment de les implanter.

Les tubes IN-9 ont un « talon d'Achille » à l'endroit où les connecteurs émergent de l'ampoule de verre. Pour leur éviter le même sort (qu'Achille), on les monte sur sup-

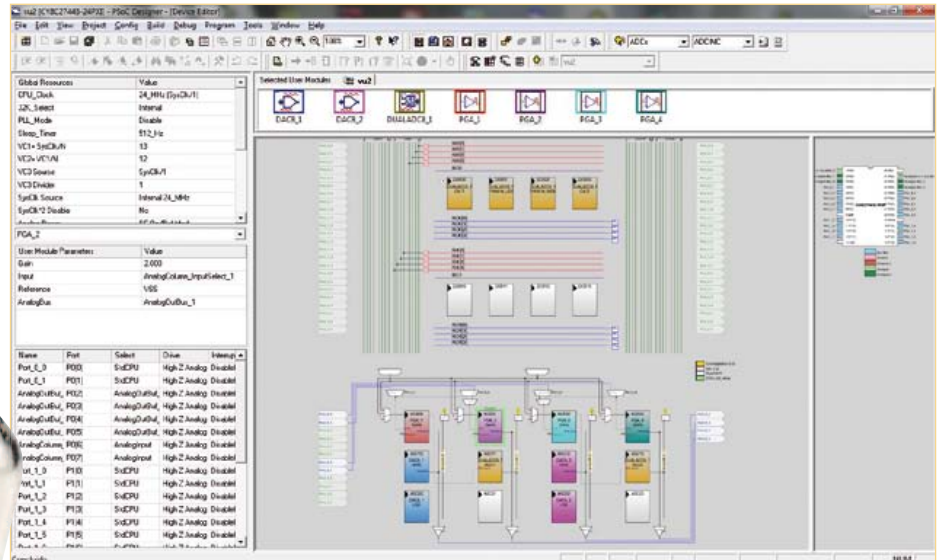


Figure 5. Un PSoC est un peu comme une super boîte de Lego. Vous choisissez les modules, vous imaginez un projet, vous le testez, le corrigez puis vous exposez le résultat. Cette image résume la structure de programmation du VU-mètre Nixie.

port. Toutefois, si les tubes sont solidement fixés à la bonne place, on peut garder leurs connecteurs longs de 40 mm jusqu'aux vis marquées V1 et V2 sur la platine. « K » désigne la cathode et « A » l'anode. N'oubliez pas que ces fils portent une haute

tension et qu'ils méritent par conséquent toutes les précautions et les protections d'usage.

Visionnez sur le canal Youtube d'Elektor, baptisé ElektorIM [4], la petite vidéo qui

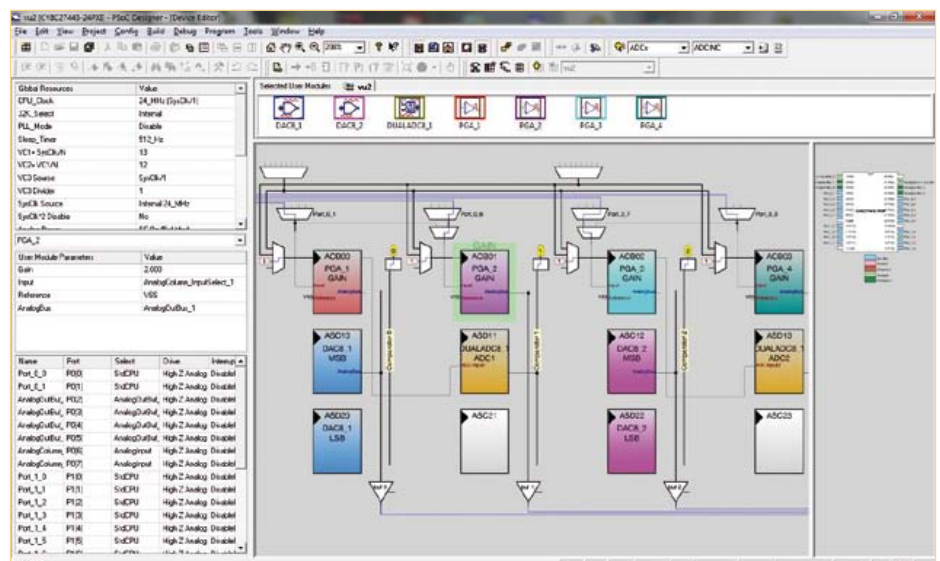


Figure 6. Une vue rapprochée, quoique virtuelle, du bloc d'amplification du PSoC.



montre le VU-mètre Nixie en action. Nous ne pouvions pas terminer cet article sans chaudement vous recommander de ne pas construire ce circuit si vous ne maîtrisez pas les dangers inhérents au travail sur de hautes tensions continues. C'est le cas des raccordements audio à l'intérieur d'un amplificateur à tubes.

(110744 – version française : Robert Grignard)

- [1] Thermomètre hygromètre à tubes Nixie, juin 2012, [www.elektor.fr/110321](http://www.elektor.fr/110321)
- [2] Spoutnik donne l'heure, janvier 2007, [www.elektor.fr/050018](http://www.elektor.fr/050018)
- [3] [www.elektor.fr/110744](http://www.elektor.fr/110744)
- [4] [www.youtube.com/user/elektorim?feature=results\\_main](http://www.youtube.com/user/elektorim?feature=results_main)

Publicité

**SCOPES and more**  
**HAMEG®**  
 Instruments  
 A Rohde & Schwarz Company  
**Great Value in TEST & MEASUREMENT**

## Liste des composants

### Résistances

(CMS 0805, sauf ajust.)

R1, R2, R5, R8 = 1 kΩ  
 R3, R6 = 100 kΩ  
 R4, R7 = 470 Ω  
 R9 = 2,2 kΩ  
 R10 = 220 kΩ  
 R11 = 10 kΩ  
 R12 = 33 kΩ  
 R13 = 47 Ω  
 P1, P2 = 220 Ω ajust. horiz.  
 P3 = 1 kΩ ajust. horiz.

### Condensateurs

C1, C4 = 470 nF (CMS 0805)  
 C2, C11, C17 = 10 μF 16V rad., pas de 2,5 mm  
 C3, C5, C6, C7, C8, C9, C10, C13, C14 = 100 nF (CMS 0805)  
 C12 = 1 μF 250 V (CMS 1825)

C15 = 2,2 nF (CMS 0805)  
 C16 = 470 μF 25 V radial, pas de 5 mm

### Inductances

L1 = 100 μH, 860 mA, Multicomp  
 MCSDRH73B-101MHF

### Semi-conducteurs

D1, D2, D3, D4 = BAT46WH  
 D5 = BYV26  
 D6 = BZD27C5V1P zener 5,1 V  
 T1, T2 = MMBTA42  
 T3 = IRF640NSPBF  
 T4 = BC847  
 IC1 = CY8C27443-24SX1, Elektor 110744-41\*  
 IC2 = MCP101-475HI/TO  
 IC3 = LM324ADT (SOIC-14)  
 IC4 = NE555DT (SOIC-8)  
 IC5, IC6 = 78L05

### Divers

K1 = 3,5 mm jack stéréo, encartable (Lumberg 1503 09)  
 K2 = embase à 5 picots SIL au pas de 2,54 mm  
 K3 = bornier à 2 vis au pas de 5 mm  
 V1, V2 = bornier à 2 vis, au pas de 5 mm  
 2 x tube Nixie IN-9  
 circuit imprimé 110744-1\*

\* cf. [www.elektor.fr/110744](http://www.elektor.fr/110744)

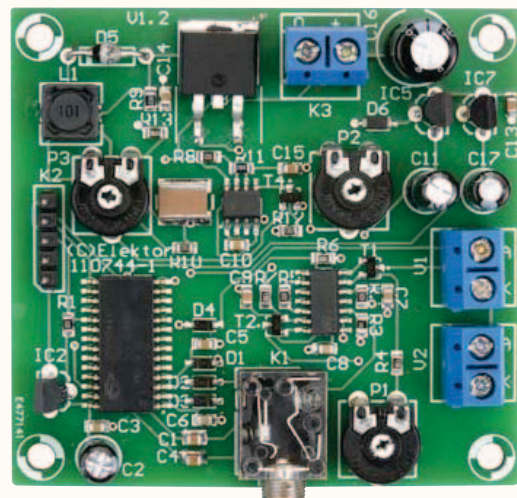
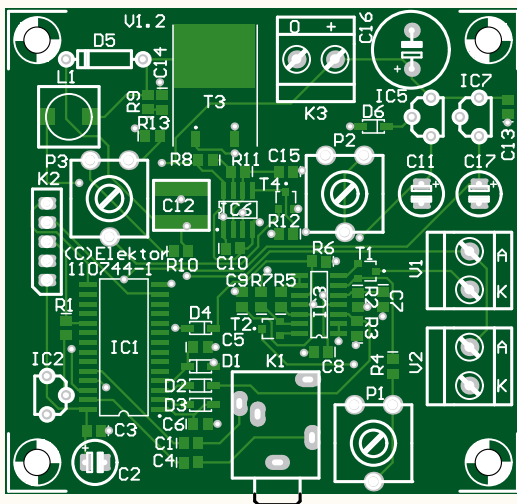


Figure 7. La platine conçue dans les laboratoires d'Elektor mélange les CMS et les composants à fils.



- ✓ plus de 40 ans d'expérience
- ✓ plus de 40 000 produits en réserve
- ✓ pas de supplément pour les petites quantités
- ✓ envoi rapide en 24 heures

+49 (0)4422 955-333

+49 (0)4422 955-360

**Qualité professionnelle à des prix réduits !**

### Technologie de refroidissement

Composants



**SUNON**  
FAN-ML 4010-12  
**2,70**

#### Système de ventilateur 12 V brevetés MagLev

Rotors tenus en équilibre par un champ magnétique. Le ventilateur peut être ainsi utilisé en toute circonstance.

Bestellnummer:	Dimension (mm)	tr/min.	dB	m³/h
FAN-ML 4010-12	40x40x10	8500	39	16,2
FAN-ML 4020-12	40x40x20	8200	27	18,4
FAN-ML 5010-12	50x50x10	6100	35	25,9
FAN-ML 6010-12	60x60x10	4000	28	27,2
FAN-ML 8020-12	80x80x20	3200	38	61,1

#### Dissipateur thermique pour un montage par vis multiples

- Profil aluminium
- pour T0220, 218, TOP3
- Canal de la vis M3
- Résistance thermique du dissipateur : 3,6 K/W
- H x L x P : 94 x 55 x 28 mm



**V 6716Z**  
**3,25**

Tous les dissipateurs thermiques sur : <http://r.ch.it/5J>

#### Réglage du ventilateur selon la température

Pour des ventilateurs de 2,5 W max.

de tension de service : 12 V

Optimisation de la vitesse de rotation selon la température en vue d'un niveau sonore minimal.

**LÜFTER-REGLER 2,70**



#### Pâte thermique

Assure une meilleure conductibilité thermique entre le semi-conducteur et le dissipateur thermique.

**LEITPASTE 5GR 1,25** Pulvérisateur  
**LEITPASTE 35GR 2,80** Tube



Besoin de l'atelier

### Optoélectronique

Alimentation en courant électrique

**ELECTRONIC ASSEMBLY**  
making things easy



EA EDIP-TFT43A  
**183,25**

#### Unité d'affichage 4,3" TFT en état de fonctionner immédiatement

Outre les diverses polices, des fonctions graphiques puissantes sont déjà intégrées et immédiatement utilisables.

- Affichage graphique TFT avec fonctions graphiques
- 480 x 272 pixels / 4,3 (320 x 240 pixels / 3,2), couleur 16 bits (65 536 couleurs) avec éclairage LED blanc
- Alimentation +5 V / 180 mA

EA EDIP-TFT43A **183,25** Module TFT  
EA EDIP-TFT43ATP **192,95** Module TFT avec ÉCRAN TACTILE

#### LED 5 mm, ultra clair, Lumen Ligot

Boîtier : incolore, clair

**LUCKY LIGHT®**  
ELECTRONICS CO., LTD.

Bestellnummer:	m / couleur	UF / IF (V / mA)
LED 5-16000 RT	630 m super rouge	2
LED 5-08000 GN	520 m pur vert	3,2
LED 5-14000 GE	592 m super jaune	2
LED 5-12000 BL	466 m bleu	3,2
LED 5-05000 WS	6000 K blanc	3,3
LED 5-25000 WS	5500 K blanc	3,3
LED 5-16000 WW	4000 K blanc chaud	3,3

up to **25.000** mcd

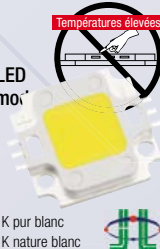


#### Module LED 10 W haute puissance

Le meilleur de la technologie LED  
**ATTENTION !** Température du module plus de 150 °C.

Assurer un refroidissement suffisant.

LED H10WG PWS **13,95** 6000 K pur blanc  
LED H10WG NWS **14,95** 4000 K nature blanc



Technologie réseau et PC

### Alimentation en courant électrique

Technologie satellite et TV

**TRACO POWER**



TSR 1-2433  
**8,95**

#### Série TSR-1, 1A Régulateurs à découpage ultra performants

Remplacer un régulateur linéaire 78xx inefficace par un régulateur à découpage simple de la série TSR-1 !

- Température de travail : -40°C à +85°C
- Tenue permanente aux courts-circuits

	Tension d'entrée	Tension de sortie	Courant de sortie max.
TSR 1-2433	<b>8,95</b> 4,75-36 VDC	3,3 VDC	1000 mA
TSR 1-2450	<b>8,95</b> 6,50-36 VDC	5,0 VDC	1000 mA
TSR 1-2490	<b>8,95</b> 12,00-36 VDC	9,0 VDC	1000 mA
TSR 1-24120	<b>8,95</b> 15,00-36 VDC	12,0 VDC	1000 mA

Tous les produits TRACO <http://r.ch.it/5K>

#### Série TPMW Convertisseur continu-continu, module

- Puissance de sortie : 4 watts
- Entrée réseau universelle : 85-264 VAC, 47-440 Hz
- cul/UL 60950-1, IEC/EN 60950-1, Rapport CB
- Gamme de température de travail : -25°C à +60°C
- Protection de surcharge et protection contre les courts-circuits ; classe de protection II

	Sortie	Rendement
TMPM 04105	<b>16,40</b> 5 VDC / 800 mA	72 %
TMPM 04112	<b>16,40</b> 12 VDC / 333 mA	76 %
TMPM 04124	<b>16,40</b> 24 VDC / 167 mA	77 %

#### Série TIW Convertisseurs continu-continu, 5-12 W

- Installation sur des prises encastrées
- Protégé contre la saleté / l'eau, IP 67
- Entrée réseau universelle : 93-264 VAC, 47-63 Hz
- Autorisations de sécurité selon IEC/EN 60950-1, EN 60335-1 et EN 50178
- Protection de surcharge et protection contre les courts-circuits

	Tension de sortie	Tension d'entrée	Courant de sortie max.
TIW 06-105	<b>27,80</b> 5 W	5,0 VDC	1000 mA
TIW 12-112	<b>29,95</b> 12 W	12,0 VDC	1000 mA
TIW 12-115	<b>29,95</b> 12 W	15,0 VDC	800 mA
TIW 12-124	<b>29,95</b> 12 W	24,0 VDC	500 mA



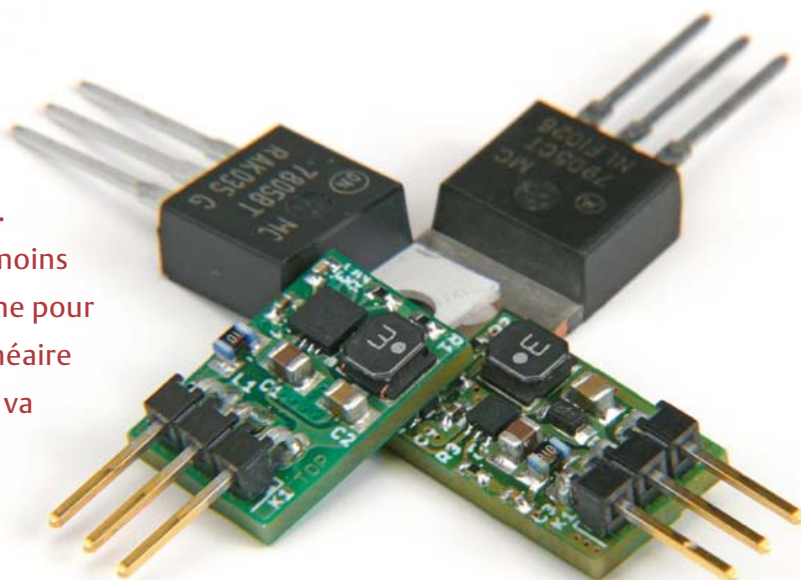
# régulateurs de tension froids

## variantes à découpage des 7805 et 7905

Raymond Vermeulen (Elektor)

«Enfin ! Mon circuit marche du tonnerre. Compact, bon rendement et tout... du moins sur l'alim du laboratoire ! Il lui en faut une pour lui tout seul. Alors, vite un régulateur linéaire et il sera opérationnel. Sauf que ce 78xx va chauffer, dissiper toute la puissance. Il faudra un radiateur, plus de place, etc. La galère...»

Fini le calvaire, voici enfin une variante verte, respectueuse du rendement, le régulateur à découpage pas plus gros qu'un 78xx.



C'est mon collègue Ton Giesberts qui s'est pointé un jour dans mon bureau en me demandant si je voyais la possibilité de remplacer les 7805, 7905 et consorts par un truc à commutation. J'ai cherché, et trouvé, une solution commerciale, disponible, oui mais volumineuse et très limitée en courant. J'en ai tout de suite conclu qu'on n'était jamais mieux servi que par soi-même. Quelques mois plus tard, voici le résultat : **un régulateur à découpage**, basé sur les convertisseurs abaisseurs (*buck*) standard, **moulé dans le costume du 7805**.

Aussitôt, mon collègue Harry Baggen, de la rédaction, suggère de faire pareil pour les tensions négatives. Je sais comment faire du négatif avec du positif, mais de négatif à moins négatif ? Je tombe alors sur une vieille note d'application de *National Semiconductor* qui explique le principe de la topologie *negative buck*. On va s'en servir !

### Le positif d'abord !

Ce circuit utilise la tactique du convertisseur série, donc abaisseur, encore appelé *stepdown* ou *buck*. C'est un commutateur qui utilise une diode, une bobine et un

condensateur (**fig. 1**). Tandis que l'interrupteur est fermé, de l'énergie se stocke dans la self. Quand il s'ouvrira, la bobine va restituer cette énergie au condensateur réservoir et à la charge à travers la diode. Le rapport cyclique, c'est-à-dire le quotient des temps de conduction et de blocage du commutateur électronique, détermine la grandeur de la tension de sortie. Elle sera d'autant plus élevée que la conduction sera longue, jusqu'à s'approcher de la tension d'entrée. Si l'on s'en tient à ce schéma simpliste, la tension de sortie faiblit quand la demande de courant s'élève. Pour l'éviter, on ajoute une rétroaction : on compare la tension de sortie à celle qu'on souhaite obtenir et l'on adapte le rapport cyclique en conséquence.

Les convertisseurs abaisseurs sont de plus en plus utilisés dans l'appareillage électronique et de nombreux fabricants de puces spécialisées dans cet usage. Elles contiennent les composants de commutation et aussi l'électronique de régulation et de protection. Nous avons choisi le TPS62150 de *Texas Instruments*, un convertisseur abaisseur synchrone qui peut four-

nir 1 A et travaille à une fréquence relativement haute, 1,25 MHz, de quoi réduire considérablement la taille des composants environnants. Le diagramme fonctionnel de cette puce est visible à la **figure 2**. Si vous observez le schéma de notre remplaçant du 7805 (**fig. 3**), vous verrez qu'à la diode du schéma théorique de la figure 1, on a substitué un MOSFET intégré pour réduire les pertes. Peu de choses à raconter sur le schéma, vu que presque tout est intégré dans la puce, à l'exception de la bobine, de quelques résistances et condensateurs. Nous avons ajouté un filtre LC (L1/C1) à l'entrée pour atténuer d'éventuels parasites. Nous y avons inséré R3 pour empêcher le filtre d'osciller, même si en pratique la RSE, la résistance série équivalente, de L1 et C1 est probablement suffisante pour obtenir l'effet désiré. Comme il ressort du diagramme fonctionnel, le CI de convertisseur contient différents circuits qui règlent avec précision la tension de sortie, si bien que, soumis à des échelons de charge, de brusques variations de la demande en courant, il ne présente qu'un minimum de creux et de pics de tension de sortie.



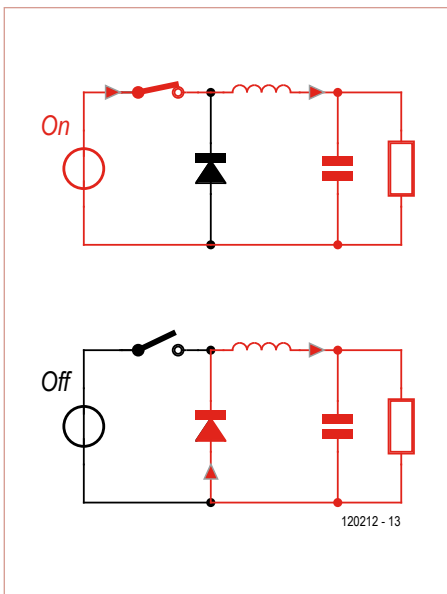


Figure 1. Dispositif de base du convertisseur abaisseur de tension.

À l'examen de la taille du circuit complet, admettez que nous avons choisi les composants avec un soin particulier. Les dimensions d'un boîtier TO-220 sont spécifiées avec une certaine tolérance et varient d'une marque à l'autre. Nous sommes restés dans la moyenne avec notre projet : 15,5 x 10,1 x 4,75 mm. L'endroit où se trouve normalement la plaquette métallique percée d'un trou fait ici partie de la platine. Impossible de la fixer sur un radiateur comme un 7805 traditionnel, direz-vous. Aucun souci : la variante à découpage dissipe si peu qu'elle s'en passe. En *rawette*<sup>1</sup>, le **tableau 1** vous donne les valeurs de résistances à adopter pour des tensions de sortie différentes de 5 V.

La conception de la platine pour une alimentation à découpage joue un rôle primordial. Le **figure 4** vous montre le tracé de la platine pour le régulateur positif. La bonne pratique dans ce domaine consiste à réduire autant que faire se peut les boucles là où doit circuler un courant fort et la surface cuivrée pour de grandes variations de tension. Comme on souhaite aussi que le circuit soit petit, on fait d'une pierre deux coups. À l'égard des prestations, la version commutatrice ne le cède en rien au 7805 ordinaire : elle accepte 17 V maximum à l'entrée et la sortie peut fournir largement 1 A. Les condensateurs standard que l'on trouve dans la configuration normale du 7805 peuvent rester en place, mais on peut aussi bien s'en passer pour un nouveau projet.

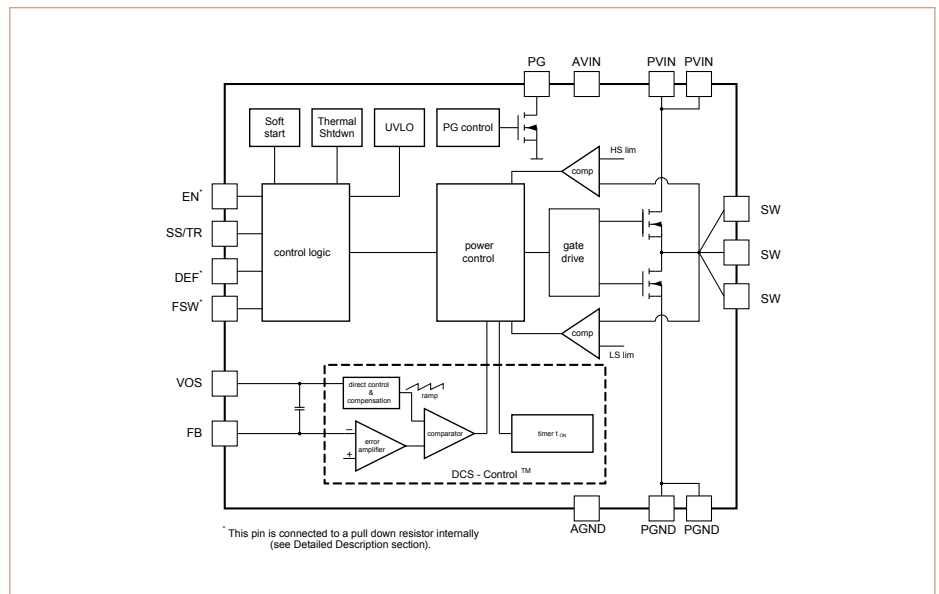


Figure 2. Diagramme fonctionnel et structure interne du TPS62150.

## Caractéristiques techniques

### Régulateur positif :

- Tension de sortie : 5 V (réglable par 2 résistances)
- Tension d'entrée : 6 à 17 V
- Courant de sortie max. : 1 A
- Ondulation de sortie max. : 80 mV sous 17 V entrée et 1 A sortie
- Dimensions et brochage identiques au 7805 standard

### Régulateur négatif :

- Tension de sortie : -5 V (réglable par 2 résistances)
- Tension d'entrée : -6 à -17 V
- Courant de sortie max. : 0,85 A
- Ondulation de sortie max. : 36 mV sous -17 V entrée et -0,85 A sortie
- Dimensions et brochage identiques au 7905 standard

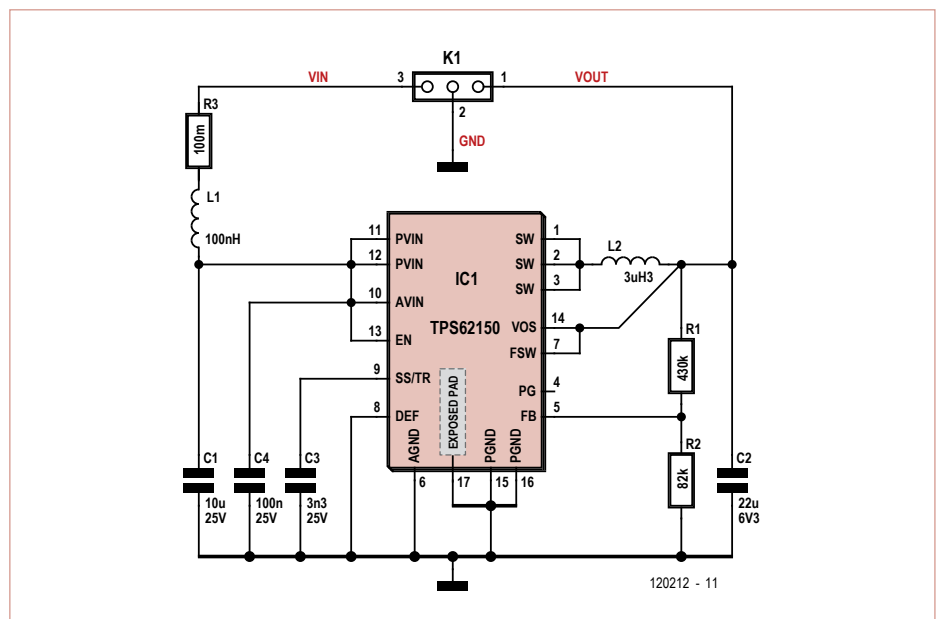


Figure 3. Schéma du régulateur positif à découpage

## Liste des composants du régulateur positif

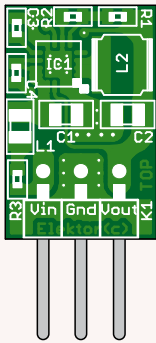


Figure 4. La platine à double face pour le régulateur positif. (200% de sa taille réelle)

### Résistances :

R1 = 430 kΩ 1%/0,1 W, CMS 0402  
 R2 = 82 kΩ 1%/0,1 W, CMS 0402  
 R3 = 0,1 Ω 1%/0,1 W, CMS 0402

### Condensateurs :

C1 = 10 μF/25 V, X5R, CMS 0805 (p.ex. TMK212BBJ106KG-T)  
 C2 = 22 μF/6,3 V, X5R, CMS 0805 (p.ex. JMK212BJ226MG-T)  
 C3 = 3,3 nF/25 V, X7R, CMS 0402  
 C4 = 100 nF/25 V, X5R, CMS 0402

### Inductances :

L1 = 100 nH, CMS 0805 (p.ex. MLZ2012DR10D)  
 L2 = 3,3 μH, 4x4 mm (p.ex. NRS4018T3R3MDGJ)

### Semi-conducteurs :

IC1 = TPS62150RGTT buck-converter, QFN-16

### Divers :

K1 = embase à 3 picots coudés, au pas de 2,54 mm  
 circuit imprimé 120212-1 (cf. [www.elektor.fr/120212](http://www.elektor.fr/120212))

Tableau 1. Valeurs des résistances pour différentes tensions du régulateur positif.

U <sub>uit</sub>	R1	R2
5,0 V	430 kΩ	82 kΩ
3,3 V	256 kΩ	82 kΩ
2,5 V	174 kΩ	82 kΩ
1,8 V	102 kΩ	82 kΩ
1,5 V	72 kΩ	82 kΩ
1,2 V	41 kΩ	82 kΩ

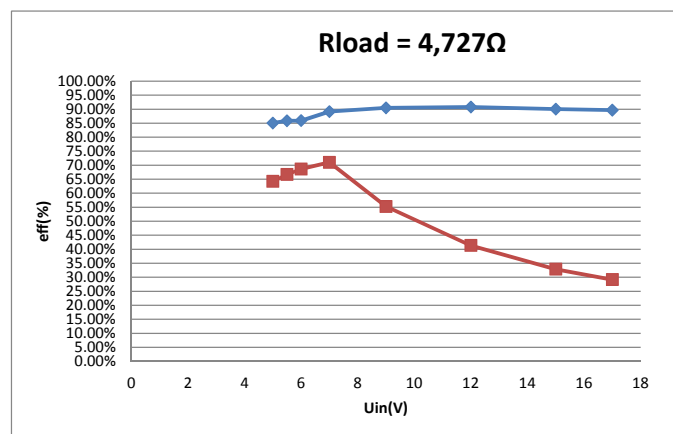
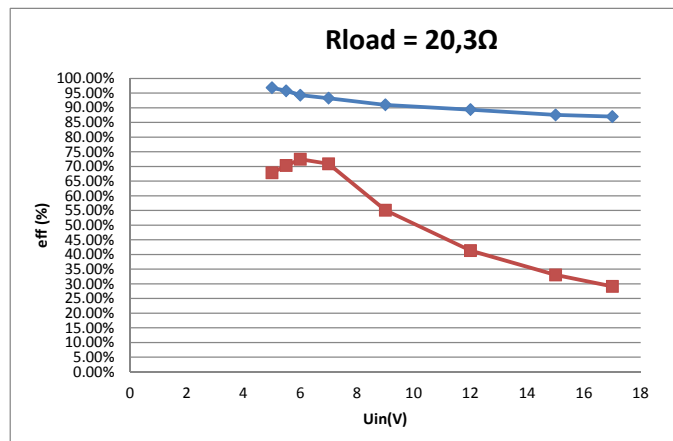


Figure 5. Mesure du rendement du régulateur positif à 250 mA (a) et 1 A (b) de courant de sortie. En bleu, le régulateur positif à découpage ; en rouge, le 7805 standard.

La figure 5 montre le rendement atteint par le régulateur positif pour des tensions d'entrée et des courants de sortie différents. Avec de basses tensions d'entrée, le rendement pour un courant jusqu'à 250 mA avoisine 95 %, pour la tension maximale, il descend à 85 % environ. Pour le courant maximum, il se situe à peu près à 90 % sur

toute la plage de tensions d'entrée. Le plus intéressant, c'est d'observer le comportement du circuit soumis à un échelon de charge, une brusque variation du courant de sortie. Pour le vérifier, nous avons construit un montage dans lequel la charge passe de 10 à 85 % du maximum à 50 kHz avec un rapport cyclique de 50 %. La figure 6 vous donne à voir la mesure de la variation de tension de sortie. Le grand pic est principalement causé par la position de la sonde de mesure et des fils. Le montage utilisé ne constitue pas une charge réaliste, mais plutôt le cas le plus défavorable. En usage normal, le pic à la sortie serait beaucoup plus petit.

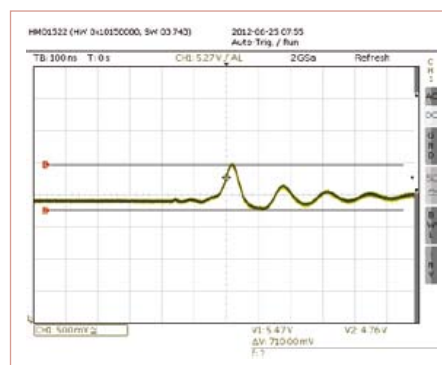


Figure 6. Comportement sur charge commutée.

### Le régulateur négatif

Si la structure du régulateur positif et le choix des composants respectaient les données du cahier de caractéristiques de la puce utilisée, avec la version négative, c'est une autre paire de manches. D'abord, il existe bien toutes sortes de circuits inté-

grés pour alimentation à découpage, pour des élévateurs ou pour des inverseurs de tension, mais point de puce qui accepte une tension négative pour la réduire en sortie. Sur la toile, nous avons trouvé la description d'un circuit qui le permet.

On part d'un convertisseur élévateur (*boost*) que l'on branche littéralement à l'envers, pour en faire un convertisseur abaisseur (*buck*) négatif. Si vous regardez bien le schéma du régulateur négatif à la **figure 8**, vous aurez l'impression que les dénominations des broches sont mal placées : la masse du circuit est reliée au  $V_{in}$  de la puce et l'entrée de tension négative, à la broche GND de la puce. Et voilà comment d'une tension négative on fait une autre qui l'est moins.

Et maintenant, que vais-je faire... de la rétroaction ? La question délicate, c'est que le signal de contre-réaction doit, dans cette configuration hors du commun, être en relation avec la broche GND de la puce. Utiliser un diviseur de tension ordinaire ferait que, lors d'un changement du niveau d'entrée par rapport à la broche  $V_{CC}$ , le rapport des résistances ne serait plus correct et alors le circuit ne pourrait plus fonctionner que pour une tension d'entrée déterminée. On s'en tire avec un moniteur de courant de *shunt*, le MAX4073, pour le signal de rétroaction. Il « traduit » la tension de sortie par rapport à GND en une tension en relation avec  $V_{in}$ .

Pour le convertisseur élévateur, nous avons cherché une puce qui supporte une grande tension d'entrée, mais de petit format pour cette application. Le choix s'est porté sur le TPS61170 de *Texas Instruments* (**fig. 7**). Il tient 20 V entre  $V_{in}$  et GND.

Malgré la configuration spéciale, nous pouvons utiliser les mêmes formules pour le calcul des composants que pour un convertisseur abaisseur. Nous gardons aussi les mêmes filtres LC d'entrée et de sortie du régulateur positif.

Au bout du compte, ce projet garde des dimensions assez petites pour pouvoir être implanté sur une platine aux mensurations

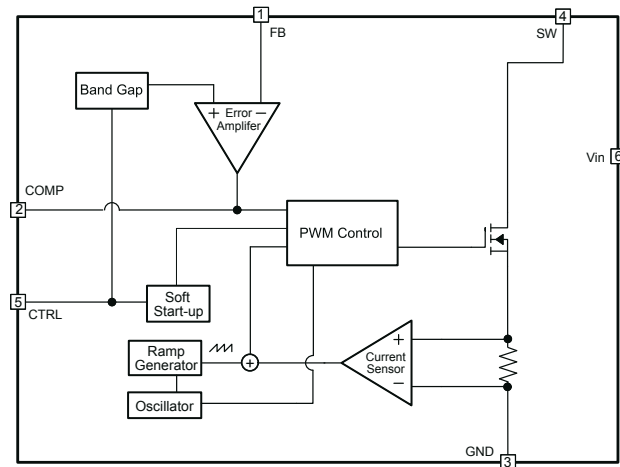


Figure 7. Diagramme fonctionnel du TPS61170.

du 7905 habituel (**fig. 9**). En gracieuseté, nous vous proposons également dans le **tableau 2** les valeurs de R1 et R2 pour différentes tensions de sortie.

Les résultats obtenus avec cette configuration particulière sont très bons. La **figure 10** montre que le rendement est

bien supérieur à celui du 7905 ordinaire. Seul le courant maximum de sortie est inférieur par rapport à notre version positive. La limite est imposée par le TPS61170. En outre, il semble que la limitation de courant du régulateur négatif soit dépendante de la température. Si nous soufflez dessus pendant qu'il limite, la tension de sortie

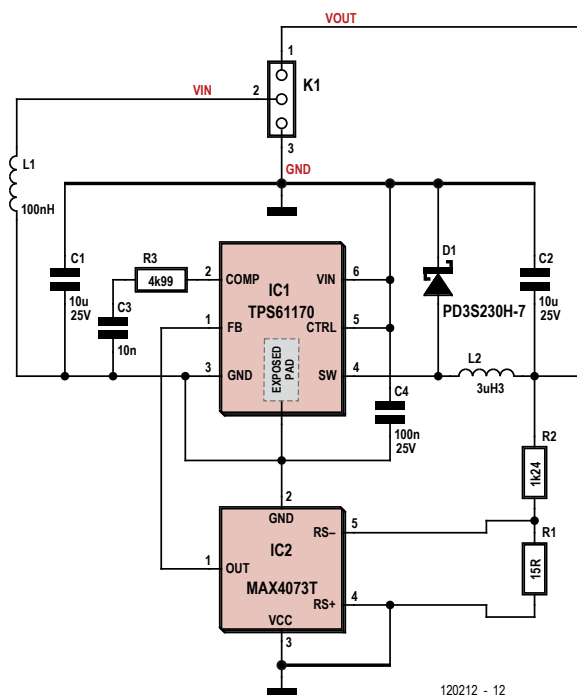


Figure 8. Schéma du régulateur négatif à découpage.



## Liste des composants du régulateur négatif

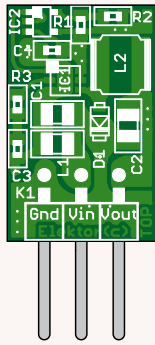


Figure 9. La platine à double face pour le régulateur négatif. (200% de sa taille réelle)

### Résistances :

R1 = 15 Ω 1%/0,1 W, CMS 0402  
 R2 = 1,24 kΩ 1%/0,1 W, CMS 0402  
 R3 = 4,99 kΩ 1%/0,1 W, CMS 0402

### Condensateurs :

C1 = 10 μF/25 V, X5R, CMS 0805 (p.ex. TMK212BBJ106KG-T)  
 C2 = 22 μF/6,3 V, X5R, CMS 0805 (p.ex. JMK212BJ226MG-T)  
 C3 = 10 nF/25 V, X7R, CMS 0402  
 C4 = 100 nF/25 V, X5R, CMS 0402

### Inductances :

L1 = 100 nH, CMS 0805 (p.ex. MLZ2012DR10D)  
 L2 = 3,3 μH, 4x4 mm (p.ex. NRS4018T3R3MDGj)

### Semi-conducteurs :

IC1 = TPS61170DRVR boost-converter, 6-SON  
 IC2 = MAX4073TAXK+T current-sense amplifier, SC-70-5  
 D1 = PD3S230H-7 diode Schottky, PowerDI323

### Divers :

K1 = embase à 3 picots soudés, au pas de 2,54 mm  
 circuit imprimé 120212-2 (cf. [www.elektor.fr/120212](http://www.elektor.fr/120212))

Tableau 2. Valeurs des résistances pour différentes tensions du régulateur négatif.

U <sub>uit</sub>	R1	R2
-5,0 V	15 Ω	1,24 kΩ
-3,3 V	23,2 Ω	1,24 kΩ
-2,5 V	30,1 Ω	1,24 kΩ
-1,8 V	42,2 Ω	1,24 kΩ
-1,5 V	51 Ω	1,24 kΩ
-1,2 V	63,4 Ω	1,24 kΩ

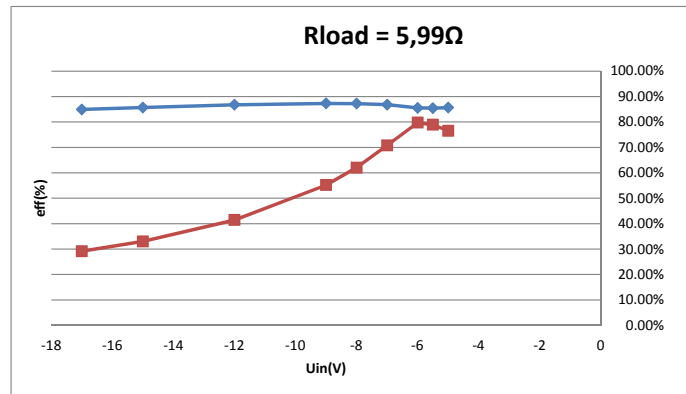
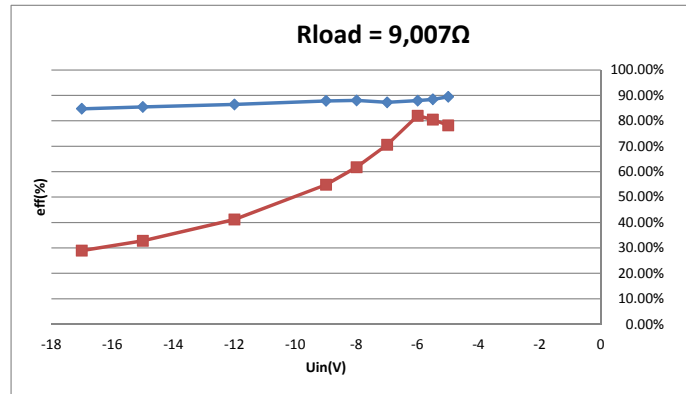


Figure 10. Mesure du rendement du régulateur négatif à 0,5 A (a) et 0,85 A (b) de courant de sortie. En bleu le régulateur négatif à découpage ; en rouge le 7905 standard.

remonte quelque peu ou redevient plus négative, question de point de vue.

Nous avons pratiqué les mêmes tests de tenue aux variations de la charge et ce circuit-ci les a tout aussi bien supportés. Comme l'indique la figure 11, le régulateur est stable sur toute la plage de tension

d'entrée, même dans le cas le plus critique avec -17 V à l'entrée.

**Les deux régulateurs à découpage présentés ici constituent des substituts de choix aux régulateurs linéaires ordinaires. Ils produisent bien un peu plus d'ondulation en sortie et la construction est forcément plus complexe, mais dès que le rendement et la dissipation deviennent des critères importants, la solution est excellente !**

(120212 – version française : Robert Grignard)

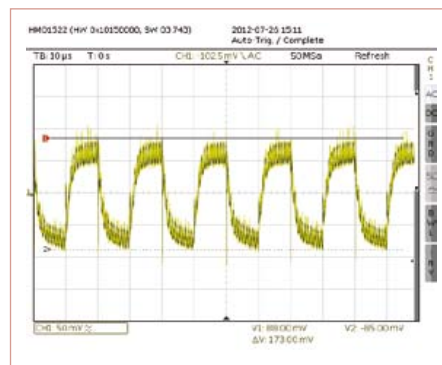


Figure 11. Comportement en impulsions à 50 kHz du courant de sortie entre 60 et 560 mA pour -17 V à l'entrée, couplage en alternatif.

### Liens

- [1] TPS61170: [www.ti.com/product/tps61170](http://www.ti.com/product/tps61170)
- [2] TPS62150: [www.ti.com/product/TPS62150](http://www.ti.com/product/TPS62150)
- [3] [www.elektor.fr/120212](http://www.elektor.fr/120212)

<sup>1</sup>rawette : petit supplément gratuit, en wallon



DESIGNSPARK

DÉVELOPPÉ PAR 

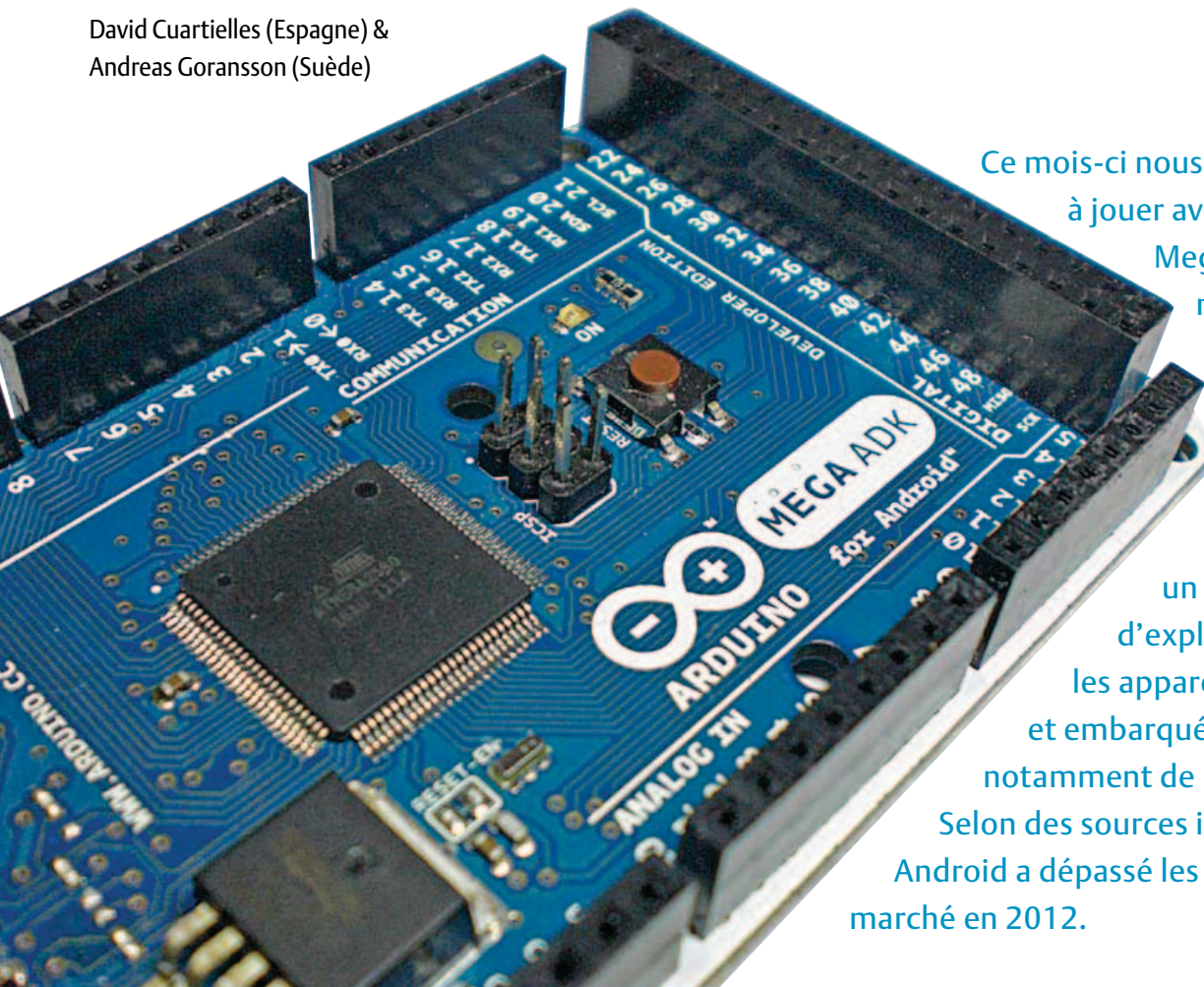
# TRÈS BIENTÔT UN NOUVEL ÉCOSYSTÈME DE CONCEPTION

[www.designspark.com](http://www.designspark.com)

# mettez le cap sur Arduino

## 3a: des périphériques maison pour votre téléphone Android

David Cuartielles (Espagne) &  
Andreas Goransson (Suède)



Ce mois-ci nous allons commencer à jouer avec la carte Arduino Mega ADK, et vous montrer comment la relier à des téléphones ou tablettes Android. Android est un système d'exploitation pour les appareils mobiles et embarqués concurrent notamment de l'IOS d'Apple. Selon des sources indépendantes [1], Android a dépassé les 50 % de parts de marché en 2012.

Grâce à Android, n'importe qui est en mesure de créer des périphériques pour les téléphones tactiles. Cet article vous aidera à préparer votre système pour le développement de tels périphériques et des applications qui vont avec.

### Un petit tour au marché

Voici la liste des courses pour nos expériences :

- une carte Arduino Mega ADK (en photo ci-dessus) ou carte Arduino Uno avec *shield* USB Host
- un téléphone Android avec Android OS 2.3.4 ou plus récent
- câbles USB et microUSB (un de chaque)

Remarque : nous allons dans cet article nous concentrer sur la préparation de votre système pour le développement d'applications Arduino à connecter à votre téléphone. Nous vous donnerons des applis prêtes à l'emploi (avec les sources) pour que vous puissiez

échanger des données avec votre téléphone.

Conseil : avant de continuer la lecture de cet article, téléchargez tous les codes qui y sont utilisés ; à cause de leur longueur, nous n'en montrerons ici que des extraits.

### Présentation d'Android

Android est un système d'exploitation (OS) conçu pour les appareils mobiles et embarqués. Basé sur Linux, il utilise une machine virtuelle de type Java, appelée Dalvik. À l'instar des autres OS utilisés sur le marché de la téléphonie mobile, Android présente un certain nombre de différences vis-à-vis de Linux.

Pour commencer, les opérations typiques d'un téléphone ont une haute priorité dans le système, utilisation en téléphonie mobile oblige. Par exemple, et en l'absence de configuration contraire, un appel (de votre moitié) vous empêchera de faire quoi que ce soit d'autre (sur votre téléphone).



Aussi, tous les appareils possèdent une série de capteurs tels que des accéléromètres, capteurs de température ou de luminosité, etc. Ils vous permettent d'accéder à l'internet ou à d'autres appareils de multiples façons : GPRS/3G/4G, Wi-Fi, Bluetooth, câble USB, etc. Enfin, vous ne programmerez pas par défaut d'applications natives puisque ces dernières tourneront sur la machine virtuelle Dalvik. Ainsi, vos programmes ressembleront, du point de vue de la syntaxe, à du Java. La plupart du temps, vous pouvez importer des bibliothèques Java (depuis les sources) directement dans vos applications pour téléphone. Android se porte garant d'un certain niveau de portabilité entre les appareils des différents constructeurs. En d'autres termes, une même appli, une fois compilée, a toutes les chances de fonctionner sur différents téléphones pour peu qu'ils fassent tourner la même version de l'OS.

### Périphériques Android

L'équipe de développement d'Android y a introduit en 2011 la possibilité de créer des périphériques, appelés accessoires, à l'aide d'une série d'outils à code source ouvert. Le célèbre *Accessory Development Kit* [2] est composé de schémas de référence et de l'*Android Open Accessory Protocol* (AOAP).

Ces schémas sont dérivés de différentes cartes Arduino. Dans cet article, nous nous concentrerons sur l'Arduino Mega ADK, compatible avec la première version de l'AOAP. Le code présenté dans cet article devrait fonctionner avec les cartes compatibles ADK plus récentes (type Arduino Due et consorts). Il devrait également être possible de reproduire les expériences présentées ici à l'aide d'un Arduino Uno et d'un *shield* USB Host comme celui de la **figure 1**.

Une des principales fonctions apportées par AOAP est le réveil d'application (*App Wakeup*) suite à la détection d'un accessoire. Lorsqu'un accessoire est branché au téléphone ou à la tablette, il déclenche le lancement d'une application de son choix. Si l'application n'est pas installée, une URL à partir de laquelle on pourra la télécharger et l'installer sera proposée.

Au moment où j'écris ces lignes, il y a, outre l'Arduino Mega ADK, une palanquée de cartes compatibles avec le Google ADK. D'un autre côté, il n'y avait aucune carte autre que l'Arduino Due compatible avec l'ADK2.

### Préparatifs : côté Arduino

Par souci de simplicité, commençons par configurer notre EDI Arduino pour la compilation de code compatible avec AOAP pour l'Arduino Mega ADK. En plus du matériel adéquat, il vous faudra installer :

- Arduino 1.0.1 ou plus récent
- La bibliothèque *UsbHost* pour Arduino (comprend la classe *AndroidAccessory*). Si vous n'avez jamais installé une bibliothèque Arduino, le petit guide ci-dessous vous aidera.

Dès que votre code sera fonctionnel sur l'Arduino Mega ADK, votre appareil Android, grâce au mode de fonctionnement d'AOAP, se connectera à l'internet et vous proposera un site pour télécharger

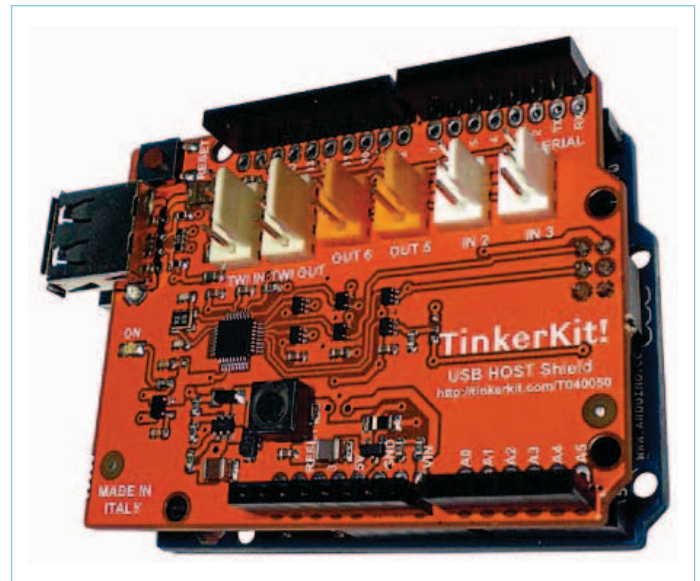


Figure 1. Un Arduino Uno relié à un shield USB Host.

ger l'application utilisée dans cet exemple (**fig. 2**), même s'il n'y a sur votre téléphone aucune application capable de tirer parti de l'accessoire.

Remarque : Pour que cette installation automatique fonctionne, il vous faudra activer l'option qui autorise votre téléphone à installer des applications qui ne proviennent pas du *Google Play Market* officiel. Rendez-vous dans *Settings* → *Applications* et assurez-vous que l'option *Unknown sources* est active. Assurez-vous également que votre téléphone dispose d'une connexion à l'internet, via le Wifi ou le réseau téléphonique, afin d'être en mesure de télécharger l'application depuis nos serveurs.

### Installation de la bibliothèque *USB Host* pour Arduino

Au moment où j'écris ces lignes, l'EDI Arduino n'inclut pas la bibliothèque *USB Host* dont nous allons parler. Vous ne verrez donc pas



Figure 2. Un Arduino Mega ADK relié à un PC et un téléphone.

apparaître *USB Host* dans le menu *Sketch / Import Library* et il vous faudra la télécharger depuis le site officiel Arduino qui héberge ce projet [3] et l'installer. En quelques mots, l'ajout d'une nouvelle bibliothèque à l'EDI se fait en créant un dossier *libraries* dans votre *sketchbook* et en y décompressant directement le fichier que vous aurez téléchargé. Il ne vous restera plus qu'à redémarrer l'EDI Arduino pour que la bibliothèque apparaisse dans le menu susmentionné. Si vous avez déjà une version précédente de cette bibliothèque ou que vous l'avez téléchargée depuis un autre site que l'officiel, je vous conseille fortement de la désinstaller avant d'ajouter la nouvelle version, les deux pouvant être incompatibles.

L'installation de la bibliothèque vous donnera accès non seulement au code permettant d'exécuter différentes commandes sur le modem, mais également à une série d'exemples qui vous permettront de :

- tester si le mode ADK fonctionne correctement
- envoyer des valeurs numériques/analogiques de la carte au téléphone
- recevoir sur la carte des valeurs en provenance du téléphone
- déboguer les divers types de périphériques USB reliés à la carte Arduino Mega ADK

Pour accéder aux exemples, c'est très simple, il suffit de se rendre dans le menu : *File / Examples / USB Host*.

### Démarrer une appli

Afin de vérifier que tout fonctionne bien pour vous, commençons avec le code le plus simple qui soit. Vous allez charger sur votre Arduino un programme qui fera télécharger à votre téléphone une application très simple, *Elektor\_MIAU*, qui affichera quelque chose sur l'écran. (*code listing 1*)

Une fois ce code chargé sur votre Arduino Mega ADK, ouvrez le moniteur de port série de l'EDI pour voir ce qui s'y passe. Lorsque vous relierez votre téléphone via un câble microUSB à la carte Arduino, vous verrez dans le moniteur quelque chose de similaire à la **figure 3**.

Dans le même temps, votre téléphone détectera qu'un accessoire compatible a été branché et, comme aucune application ne



Figure 3. Capture d'écran du moniteur de port série lors de la connexion d'un appareil Android à l'Arduino Mega ADK.

sera capable de le gérer, il vous affichera un message indiquant où trouver l'application adéquate sur l'internet (**fig. 4**).

Vous remarquerez que l'URL visible sur l'écran du téléphone est la même que celle contenue dans le programme ci-dessus. En d'autres termes, c'est dans votre code que vous déterminez l'emplacement de l'application sur l'internet. Ici, nous avons mis en place un site afin que vous puissiez essayer tous les exemples depuis votre téléphone. Ouvrez le navigateur sur votre appareil Android et contemplez (**fig. 5**).

Pour installer l'application, il vous suffira de cliquer sur le fichier *Elektor\_MIAU.apk* que vous aurez téléchargé. Si l'application adéquate est déjà présente sur votre téléphone, c'est une invitation à la démarrer qui appa-

raîtra lors d'un nouveau branchement de l'accessoire (**fig. 6**). Une fois que vous aurez autorisé l'application à démarrer, elle affichera une image GIF depuis une boucle (**fig. 7**). Nous pensions qu'il serait sympa de montrer que l'on peut utiliser les capacités graphiques de votre terminal et éviter le banal « Hello World » textuel. Cependant, cette appli n'utilise aucune information en provenance de la carte Arduino et ne lui envoie aucune — tout ce qu'elle permet c'est vérifier que votre téléphone/tablette prend correctement en charge les accessoires.

Remarque : Il convient de préciser qu'il est possible d'utiliser plusieurs applis avec un seul accessoire. Dans ce cas, la boîte de dialogue qui apparaît au branchement vous donnera le choix.

### Préparatifs : côté Android

La construction d'applications Android peut être complexe. Elle nécessite une longue liste de différents paquets logiciels de provenances variées. Nous avons écrit un guide de démarrage passant en revue les différents paquets dont vous aurez besoin pour expérimenter à moindre peine [3].

Le plus facile pour développer une application Android simple est probablement d'utiliser l'outil à sources ouvertes *Processing* [4] de C. Reas et B. Fry. Il offre un EDI simplifié permettant de compiler des applications Java. La dernière version permet également de compiler du code pour les téléphones Android, et du Javascript pour les sites web. Nous avons développé un outil complémentaire à l'EDI *Processing* qui compile et charge du code sur votre appareil Android pour commander des accessoires.

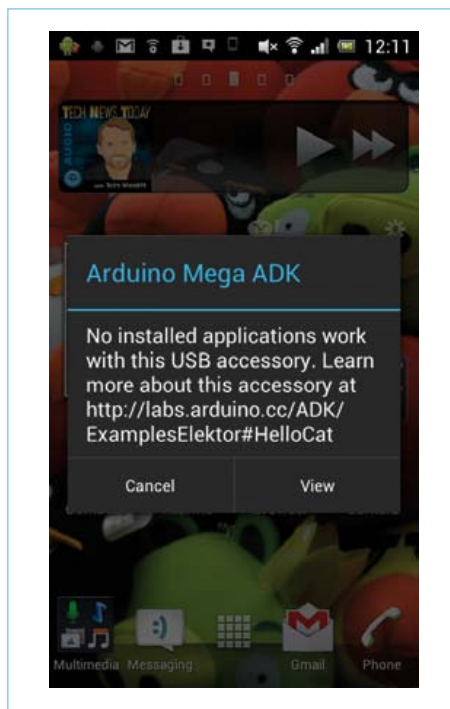


Figure 4. Capture d'écran d'un téléphone Android affichant un avertissement lorsqu'il ne dispose pas de l'application nécessaire à la prise en charge d'un accessoire.

**(code listing 1)**

```
// bibliothèques
#include <AndroidAccessory.h>

AndroidAccessory acc(
  "Arduino, SA",
  "Miau",
  "Arduino Mega ADK",
  "1.0",
  "http://labs.arduino.cc/ADK/
  ExamplesElektor>HelloCat",
  "0000000012345678");

// variable utilisée pour détecter si le périphérique
// Android a été relié récemment
boolean wasConnected = false;
```

```
void setup()
{
  // initialisation de la communication série
  Serial.begin(115200);
  Serial.print("\r\nStart");

  // initialisation de l'objet accessoire
  acc.begin();
}

void loop()
{
  if (acc.isConnected() && !wasConnected) {
    Serial.print("Accessory connected. ");
    wasConnected = true;
  }
}
```

Voici un résumé de ce dont vous aurez besoin pour reproduire les expériences présentées dans cet article :

- le SDK Android avec toutes les mises à jour jusqu'à la dernière version des API
- *Processing 2.0a8* ou plus récent
- l'outil Arduino ADK pour *Processing*
- Eclipse [5] + le greffon ADT (tous deux optionnels et non utilisés ici)

Comme souvent dans le monde des logiciels, il existe une pléthore d'outils différents pour écrire vos applications. Le SDK Android est un outil externe à votre éditeur de texte préféré qui ne se chargera

que de compiler, éditer les liens, simuler, compresser et signer vos applications Android pour une version particulière de l'OS.

D'un autre côté, *Processing* est vraiment un bon outil si vous commencez la programmation, c'est pourquoi nous y avons adjoint un outil de notre création comme expliqué dans le guide de démarrage [3]. Si vous tenez à utiliser un ensemble d'outils professionnels, pensez à Eclipse [5], l'EDI à sources ouvertes, et les outils qui associés. Sur le site du développeur Android [6], un guide explique la marche à suivre pour mettre en place une chaîne d'outils complète basée sur Eclipse.

Je vais supposer que vous aurez réussi à installer *Processing* et les autres outils sans encombre. Tous sont multiplateformes et

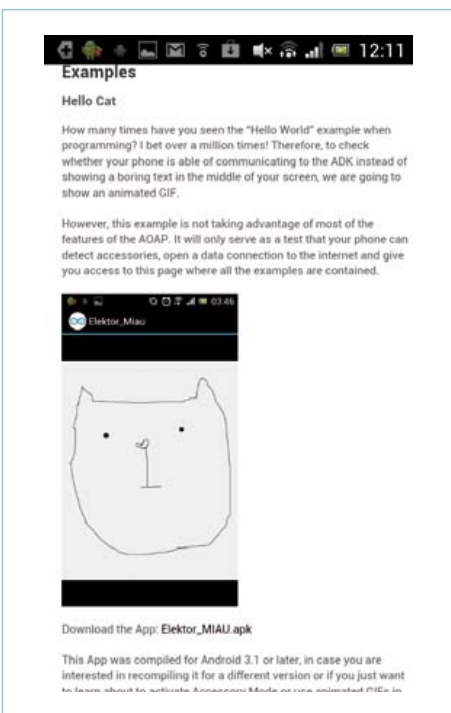


Figure 5. Un téléphone Android affichant des informations sur une application.

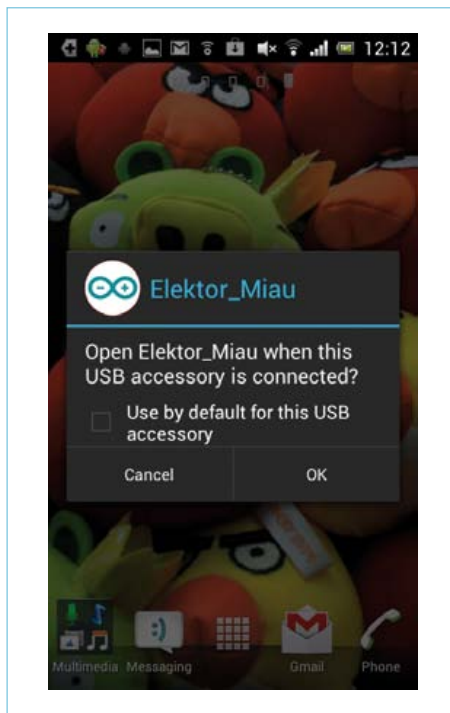


Figure 6. Votre téléphone Android vous demandera une confirmation avant de charger une application pour un accessoire.

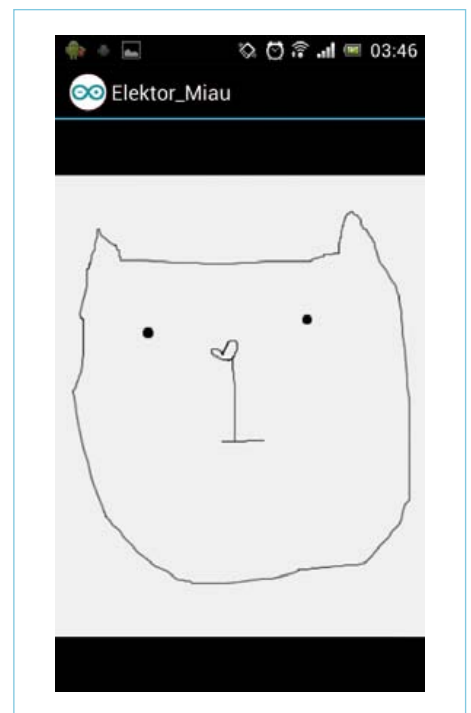


Figure 7. L'appli MIAU tournant sur Android.



**(code listing 2)**

```
void setup() {
  // assurons-nous que l'orientation de l'écran
  // sera fixe
  orientation(PORTRAIT);
}
```

```
void draw() {
  // dessin à partir du centre de la forme
  rectMode(CENTER);

  // dessin d'un carré de 50x50 pixels à l'aide
  // du jeu de couleurs par défaut
  rect(width/2, height/2, 50, 50);
}
```

**(code listing 3)**

```
AndroidAccessory acc(
  "Arduino, SA",
  "Elektor_Rectangle",
```

```
"Arduino Mega ADK",
"1.0",
"http://labs.arduino.cc/ADK/
ExamplesElektor#Rectangle",
"0000000012345678");
```

devraient fonctionner sur pratiquement n'importe quel OS. Nous allons maintenant écrire le code côté Android à l'aide de *Processing*.

**Votre première appli**

Avant même de penser à créer une application pour commander votre accessoire, nous allons utiliser *Processing* afin d'en créer une très simple pour afficher quelque chose sur l'écran du téléphone. Pour commencer à écrire des applis Android, vous devrez avoir installé le SDK Android et réglé le logiciel *Processing* pour compiler des applications Android (fig. 8). L'EDI devrait maintenant s'être mis au vert, signe que vous avez choisi le bon mode.

Créons une petite appli et exécutons-la dans la foulée sur votre appareil. Le bout de code suivant affichera un carré au centre de l'écran. Plus tard, nous ajouterons le code qui permettra à ce carré de changer en fonction de données issues de capteurs, capturées par l'accessoire Arduino. (code listing 2)

*Processing* est un outil destiné aux graphistes, il n'est donc pas étonnant que la fonction principale du programme ne s'appelle pas *loop* mais *draw*. Le code que vous écrirez dans l'EDI de *Processing* est du Java, mais vous n'aurez pas à vous soucier de la complexité sous-jacente. Pour vérifier que tout est installé correctement, sélectionnez l'entrée *Sketch* → *Run on Device* du menu (fig. 9) ; le code sera compilé puis chargé



Figure 8. Capture d'écran de Processing en mode Android ; vous remarquerez le bouton de sélection de mode dans le coin supérieur gauche.

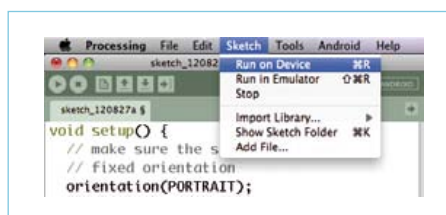


Figure 9. Lancez l'appli directement sur votre téléphone.

automatiquement sur le téléphone.

Au démarrage de l'application, vous devriez voir une image semblable à celle de la figure 10 sur l'écran de votre téléphone. L'appli démarrera directement et restera à l'écran. Vous devriez la trouver dans votre menu d'applications et pourrez la lancer autant de fois que vous le désirerez.

Remarque : A l'heure où j'écris ces lignes, *Processing* ne permet pas de signer les applis. La signature d'une appli est un processus, semblable à un coup de tampon, qui valide le fait qu'elle est prête à être distribuée sur d'autres appareils. Toute appli créée avec *Processing* ne fonctionnera que sur les appareils sur lesquels vous aurez effectué directement le chargement. Si vous souhaitez envoyer vos applis à un ami, il faudra les compiler avec Eclipse, ce qui est beaucoup plus compliqué et ne sera pas expliqué ici. Par exemple, l'application *Elektor\_MIAU* est signée, et tout le monde peut donc la télécharger à partir d'internet.

**Faire démarrer votre appli au branchement de l'accessoire**

Maintenant que vous avez créé une application très simple qui tourne sur votre téléphone, il est temps de la faire démarrer au branchement d'un accessoire. C'est ici que notre outil Arduino ADK pour *Processing* entre en jeu. Si vous l'avez installé comme expliqué [3], vous ne devriez pas rencontrer de problèmes en l'appelant depuis l'EDI *Processing*. L'outil dont vous aurez besoin pour

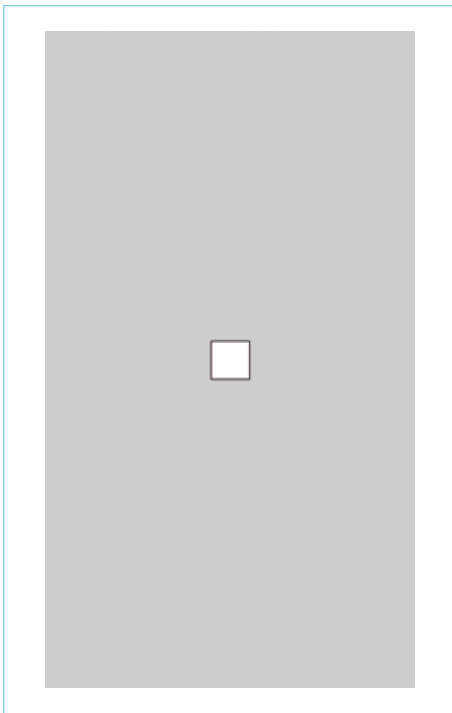


Figure 10. Capture d'écran de l'appli tournant sur le téléphone.

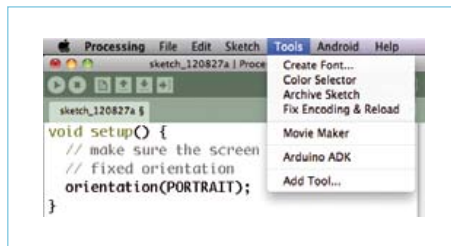


Figure 11. Capture d'écran du menu Tools de l'EDI Processing.

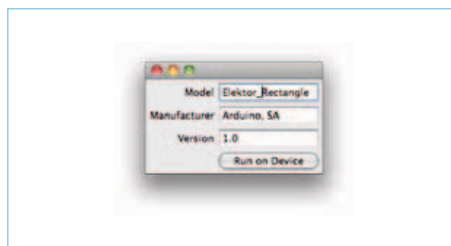


Figure 12. Cette boîte de dialogue permet de configurer l'appli pour qu'elle démarre au branchement de l'accessoire.

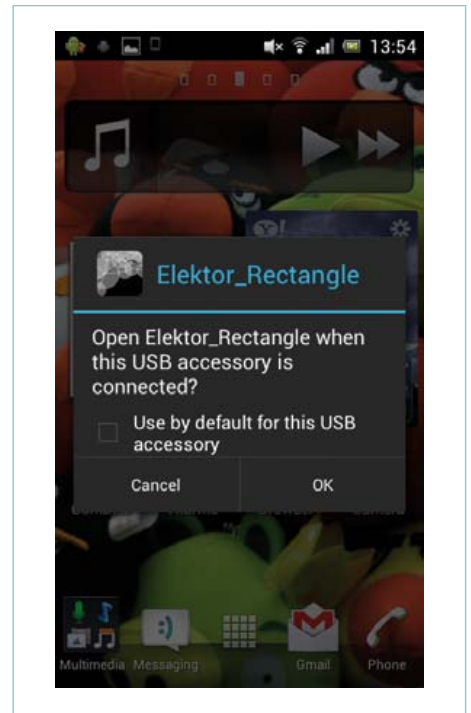


Figure 13. Cette boîte de dialogue vous demande de confirmer le lancement de votre application de dessin de rectangle lors du branchement de l'accessoire associé.

charger le code pour l'accessoire sur votre appareil Android devrait se trouver dans le menu *Tools* → *Arduino ADK* (fig. 11).

Lorsque vous sélectionnez cette entrée du menu, l'EDI ouvrira une boîte de dialogue où vous pourrez saisir le nom de l'appli et le fabricant de l'accessoire (fig. 12). Exactement ce dont les accessoires Android ont besoin pour identifier l'appli à laquelle se connecter. Il faudra que cela corresponde au code présent sur l'Arduino, comme vous pourrez le voir sur le **listing 3**.

Essayez maintenant de modifier le code du **listing 1** qui appelait l'application *HelloCat* afin qu'il appelle cette nouvelle application. (code *listing 3*)

Il ne s'agit là que de l'extrait de code dont vous aurez besoin pour modifier le code du **listing 1**.

Une fois ceci fait et le code chargé sur votre Arduino Mega ADK, votre appli de dessin de rectangle démarrera quand vous brancherez votre téléphone sur la carte (fig. 13).

### Le mois prochain

Nous voici arrivés au terme de cette brève introduction à la création d'accessoires Android à base d'Arduino. Vous venez également d'apprendre les bases de la création d'applications Android. Cependant, il ne s'agit là que d'une introduction. Rejoignez-nous le mois prochain pour apprendre à échanger des données entre un appareil Android et une carte Arduino.

(120539 – version française : Kévin PETIT)

### Remerciements

Nous remercions l'équipe de Circuits@Home pour leur travail initial sur la bibliothèque USB Host pour Arduino et Philip Lindsay pour son travail de débogage du premier ensemble officiel de bibliothèques Arduino pour Android.

### À propos du co-auteur

Andreas Goransson est concepteur d'interactivité à Malmö, en Suède. Il enseigne le logiciel à l'université de Malmö et prépare avec David Cuartielles un livre sur Android et Arduino.

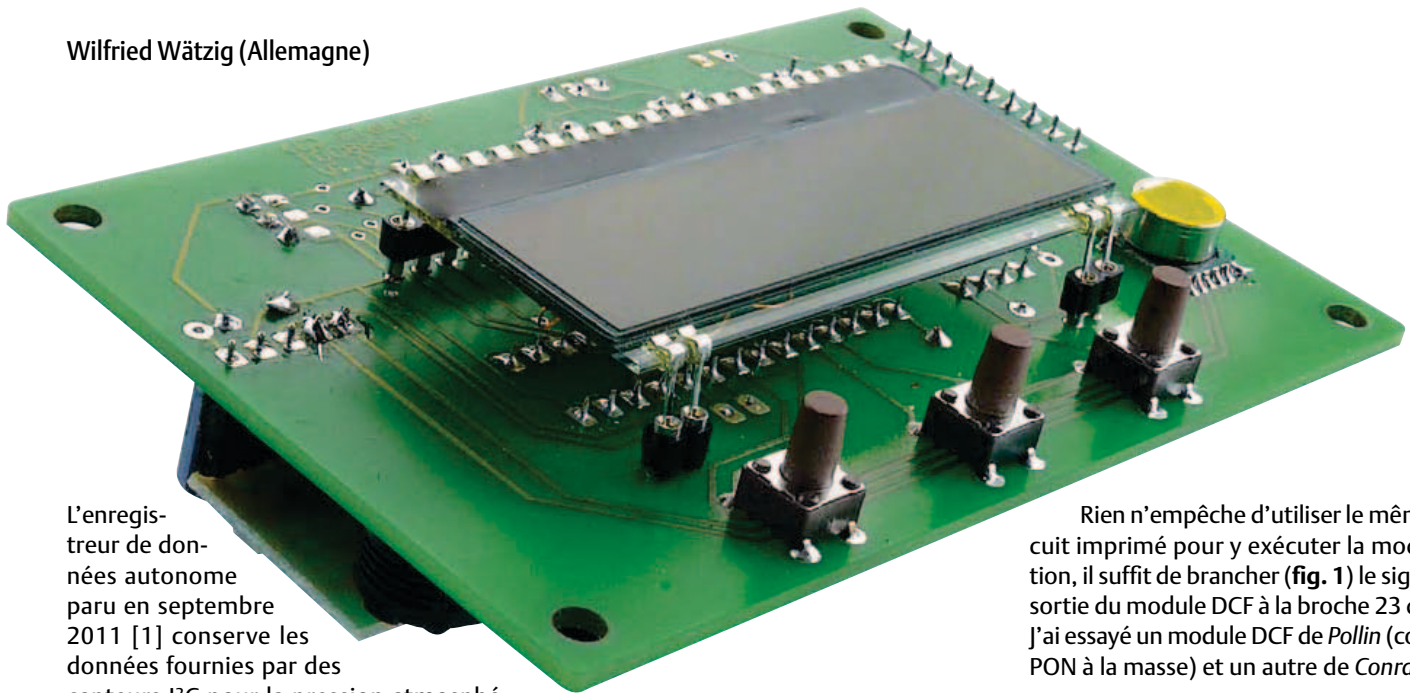
### Liens

- [1] Le marché mondial des téléphones tactiles, par systèmes d'exploitation : <http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats/a#smartphoneos>
- [2] L'Accessory Development Kit : <http://developer.android.com/tools/adk/index.html>
- [3] Arduino avec Android sur Arduino.cc : <http://labs.arduino.cc/ADK/AccessoryMode>
- [4] Le projet Processing : <http://processing.org>
- [5] Eclipse : <http://eclipse.org>
- [6] Guide Android d'installation du SDK Android SDK à l'aide d'Eclipse : <http://developer.android.com/tools/index.html>

# DCF sur l'enregistreur météo USB

À la demande de nombreux lecteurs, l'auteur de l'enregistreur météo de longue durée du numéro de septembre 2011 en a construit une version agrémentée d'un module DCF. Il vous la présente ici.

Wilfried Wätzig (Allemagne)



L'enregistreur de données autonome paru en septembre 2011 [1] conserve les données fournies par des capteurs I<sup>2</sup>C pour la pression atmosphérique, la température et l'humidité, il les affiche aussi sur un écran LCD. On peut également lire les résultats via USB par l'intermédiaire d'un module série/USB et les représenter graphiquement sur un PC avec *GNUplot*. Comme il utilise des modules de capteurs numériques, le coût du matériel est faible et il ne demande aucun réglage. Les mesures sont effectuées à intervalles réguliers et stockées dans une mémoire EEPROM sérielle, de sorte qu'elles persistent, même sans alimentation. L'instrument de mesure fonctionne six à huit semaines sur trois piles AA.

Je sais que de très nombreux lecteurs ont réalisé cet enregistreur, Elektor en propose d'ailleurs un kit réunissant le circuit imprimé, le contrôleur programmé et les capteurs d'humidité et de pression atmosphérique. Beaucoup ont exprimé le désir d'y raccorder ensuite un module DCF, ce qui exige une adaptation du logiciel. J'ai donc implémenté cette modification et j'en ai profité pour y inclure quelques fonctions supplémentaires.

Rien n'empêche d'utiliser le même circuit imprimé pour y exécuter la modification, il suffit de brancher (fig. 1) le signal de sortie du module DCF à la broche 23 du  $\mu$ C. J'ai essayé un module DCF de *Pollin* (contact PON à la masse) et un autre de *Conrad*. Les

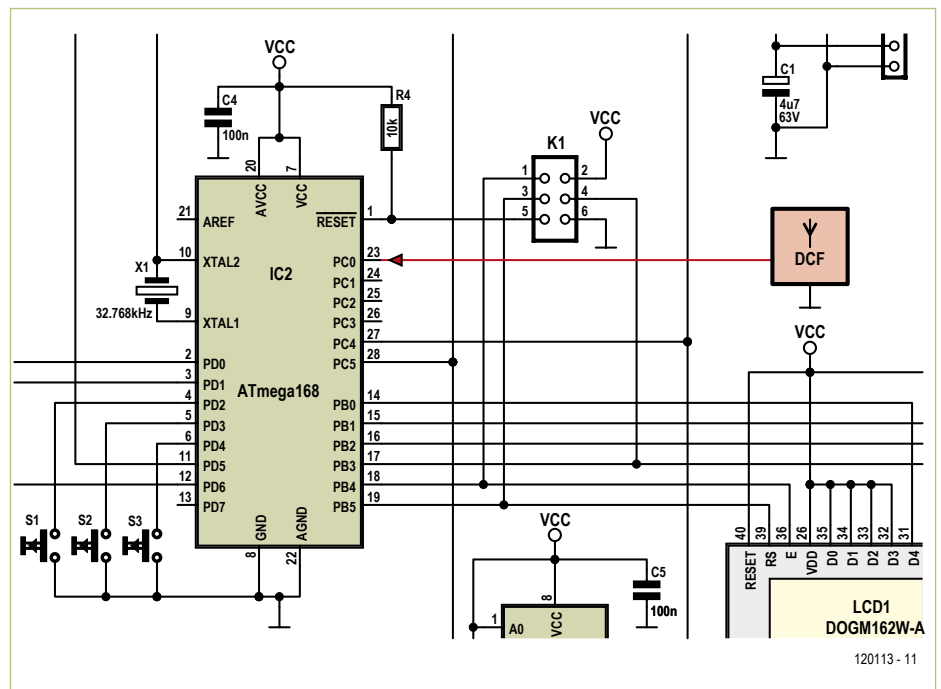


Figure 1. La mise à niveau : maintenant un module DCF, mais aussi un nouveau microcontrôleur.



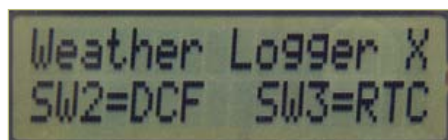
deux sont sensibles aux parasites provoqués par les alimentations à découpage, les lampes économiques, etc. Il faut attendre plusieurs minutes avant que le module ne délivre un signal propre, dont le programme puisse décoder les informations de DCF77. Autrement, le  $\mu$ C peut se servir de l'horloge intégrée (RTC).

### Fonctions avancées

L'extension du code de programme – on en arrive à près de 12 Ko – conduit au remplacement de l'ATmega88 par un ATmega168 au brochage identique.

Le nouveau programme (V 1.1) pour le microcontrôleur est à télécharger du site d'Elektor [2] sur la page du projet. Les fusibles de l'ATmega168 restent les mêmes que pour l'ATmega88, comme on le voit dans le **tableau 1**.

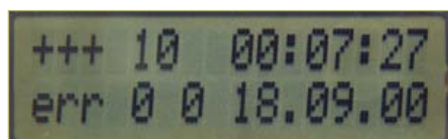
Dès sa mise sous tension, l'enregistreur météo demande dans quel mode temporel il doit fonctionner :



#### L'heure DCF :

Au cours de la synchronisation sur le signal de DCF, il utilise les impulsions des secondes de l'horloge en temps réel.

Voici ce qui s'affiche à l'écran :



+++ xx hh:mm:ss xx=compteur DCF,  
hh:mm:ss=temps  
err c b DD:MM:YY c=erreur du compteur DCF, b=bit DCF, DD:MM:YY=date  
Une fois synchronisé sur DCF, le symbole « DCF » remplace +++.  
Retour au programme principal avec S1.

Le **tableau 2** donne un aperçu de tous les réglages et fonctions accessibles par les boutons S1 à S3.

Dans les commandes interactives pour obtenir les données, il y a maintenant un format de sortie supplémentaire qui donne le mois, le jour, l'heure et la minute (cf. **tableau 3**).

(120113 – version française : Robert Grignard)

### Liens

[1] [www.elektor.fr/100888](http://www.elektor.fr/100888)

[2] [www.elektor.fr/120113](http://www.elektor.fr/120113)

**Tableau 1. Réglage des fusibles pour l'ATmega168**

Fusibles :	EXT.	0xF9	oscillateur interne à 8 MHz: 8 => horloge CPU = 1 MHz
	HIGH	0xDF	CKDIV8 validé, <i>brown-out</i> désactivé
	LOW	0x62	65 ms au démarrage

**Tableau 2. Réglage des fonctions (boutons S1 à S3)**

Fonction choisie par S1	Fonction de S2	Fonction de S3
		choix heure DCF
0 : affichage normal	indique la pression	indique l'humidité
1 : régler les heures	+ heures	+ minutes
2 : fixer M/N	incrémente M : 0 à 6	efface N
3 : commande UART	continuer	sortir
4 : afficher les valeurs	continuer	sortir
M = mesures par heure ; N = nombre de mesures		

**Tableau 3. Commandes interactives pour consulter les données**

Commande	Fonction
h =help	imprime les commandes disponibles, p.ex. # h=help/a=show-p/p#=print#/m#=set-fm#/c=clear/x=exit
a =show-p	imprime le nombre de mesures
p# =print#	p0 imprime valeurs de mesure et signification, p.ex. 123 12:30:00 T= 25.6 degC H=43% P= 987.6 hPa p1 n'imprime que les nombres, p.ex. 123 12:30:00 256 43 9876 p2 imprime l'axe du temps en heures, les autres valeurs sans la signification au format de saisie pour GNUplot, p.ex. 68.50 256 43 9876 p3 imprime la date (mois/jour) et l'heure (heure/minute) avec les valeurs pour Excel, p.ex. 10 23 12 30 256 43 9876
m# =set-fm#	fixe le nombre de mesures par heure #=0 à 6 (60/30/20/15/12/10 minutes) m0 aucune mesure m6 mesurer 6 fois par heure, soit toutes les 10 minutes
c =clear	efface N, le nombre de mesures
x =exit	termine la liaison série

# Software Defined NIC

## Ethernet maison pour microcontrôleur



Des interfaces réseau toutes faites pour microcontrôleurs, c'est comme s'il en pleuvait. Elles ont quand même des inconvénients : le matériel n'est pas spécialement bon marché, leur mise au point prend du temps

et la programmation est onéreuse. Or, pour envoyer promener des paquets de données d'un contrôleur vers le réseau, on n'a pas besoin de grand-chose. Développer pareil

*Software Defined NIC (Network Interface Card)* réclame surtout de s'y connaître en protocoles de transmission. Après, l'utilisation est d'une grande simplicité.

### Merten Joost (Allemagne)

La série d'articles des mois passés sur SDR-AVR nous a montré tout ce qu'il est possible de faire avec un contrôleur AVR à 8 bits. Difficile à croire, mais un de ces microcontrôleurs qui tourne à 20 MHz, pour quelques composants de plus, devient capable d'expédier des paquets sur un réseau Ethernet ! Mais avant de vous montrer comment une carte de *Software Defined Network (SDN)* Interface fonctionne, il nous faudra une petite dose de théorie.

#### Le modèle TCP

Dès qu'il est question de réseaux, on tombe inmanquablement sur le modèle de référence OSI pour la communication. Dans cet article, pourtant, je préfère le modèle TCP/IP parce qu'il est plus proche de la réalité.

Comme on le voit à la **figure 1**, ce modèle repose sur quatre niveaux superposés. Celui du bas se définit comme la couche liaison qui donne accès au réseau et place les paquets de données sur le câble Ethernet. Un étage plus haut, la couche réseau

s'arrange pour que les paquets trouvent leur chemin à travers l'Internet. Elle établit un canal de communication entre deux ordinateurs branchés sur Internet. Profitant de ce service, la couche de transport qui la surplombe construit des voies de communication entre les processeurs de ces ordinateurs. Le protocole le plus connu pour ce processus est le TCP (*Transmission Control Protocol*) qui établit une liaison point à point entre les deux processeurs. En divergence avec IP (*Internet Protocol*), le TCP garantit également que les paquets arrivent intacts et dans le bon ordre au récepteur, ce qu'il fait en demandant un accusé de réception pour chaque paquet. Mais il existe dans cette couche d'autres protocoles, moins connus. L'UDP (*User Datagram Protocol*) se distingue du TCP par le fait qu'il n'est pas sécurisé, parce qu'il ne garantit ni la bonne arrivée ni l'ordre de réception des paquets. Avec la diffusion (*broadcasting*), il offre un service que TCP ne contrôle pas. Par diffusion, on entend la transmission simultanée d'un paquet à plusieurs destinataires dans un réseau. Il convient d'ailleurs de relativiser l'incertitude dans le transfert. Toutes

les couches non sûres comme IP et UDP font de leur mieux pour que les paquets soient bien livrés. Si l'on s'aperçoit que des paquets sont perdus, ils seront envoyés une nouvelle fois.

La strate qui coiffe le modèle TCP/IP est la couche application, qui à son tour utilise les services des couches sous-jacentes. Le protocole le plus célèbre qui constitue le TCP est probablement le http, chargé de la communication entre les serveurs Internet et les navigateurs sur la toile. Mais il existe aussi des applications qui, délibérément, utilisent l'UDP tout en tenant compte du degré d'incertitude de la transmission de données. Il est intéressant de constater que le NFS (*Net File System*), mis au point par *Sun Microsystems*, permet d'accéder aux fichiers sur un réseau avec l'UDP.

#### Encapsuler ses paquets

Chaque couche organise les données reçues des couches inférieures dans une trame qui lui est propre. Dans cette trame, les informations sont stockées de manière à ce que les paquets arrivent à bon port dans

le processus de réception. Certains protocoles y ajoutent un en-tête (*header*) alors que d'autres y accrochent une remorque. La **figure 2** donne une idée de la fabrication de ces paquets de données qui sont envoyés sur le câble Ethernet. Les données utilisateur fournies par l'application, par exemple un serveur Internet, sont d'abord dotées d'un en-tête et transmises à la couche TCP, laquelle rajoute devant l'ensemble du paquet un autre en-tête qui lui attribue, entre autres, un numéro unique. De plus, l'en-tête TCP contient un numéro de port pour l'identification lors du processus de réception dans l'ordinateur cible. En même temps, on peut inclure dans l'en-tête la confirmation d'un paquet reçu précédemment. Le paquet est alors transmis à la couche IP. Celle-ci aussi complète le paquet de son propre en-tête avec, par exemple, les adresses IP des ordinateurs émetteur et récepteur. Le paquet sera ensuite envoyé à l'interface réseau qui ne manquera pas de l'agrémenter d'autres données.

En plus de l'en-tête Ethernet qui, notamment, contient les adresses MAC des ordinateurs impliqués, la carte réseau calcule ensuite le mot de contrôle de redondance cyclique selon l'algorithme CRC32 et ajoute le résultat en remorque du paquet. Elle ajoute en outre un préambule au paquet. Tous ces en-têtes, remorques et le préambule doivent être reconstruits dans une formule logique.

### La couche électrique

Pour faire simple, limitons-nous à la description de l'Ethernet à 10 Mbit/s. Dans sa forme d'origine [1], on utilisait comme support le câble coaxial. Il portait un signal différentiel dont l'excursion de tension prescrite était de  $\pm 0,7$  V. Sur les paires torsadées actuellement en usage, on travaille avec des niveaux de  $\pm 2,5$  V. Le codage des données s'effectue en mode *Manchester* selon la norme IEEE 802.3. Dans ce cas, un « 1 » est représenté par un flanc positif (montant) et un « 0 » par un flanc négatif (cf. **fig. 3**). Ce type de codage contient indirectement le signal d'horloge, puisque chaque bit garantit un changement de flanc. Avec un taux de transfert de données de 10 Mbit/s, doublé logiquement de ce fait, la fréquence d'émission est de 20 MHz. La figure 3 montre par

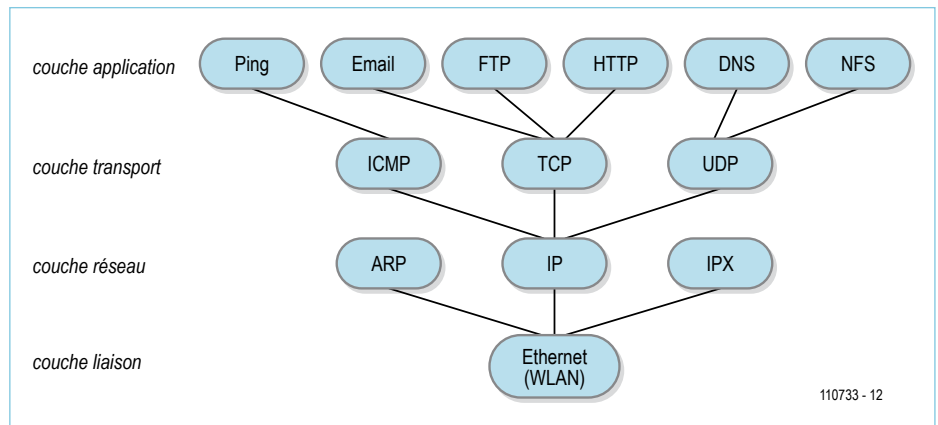


Figure 1. Le modèle de couches TCP/IP.

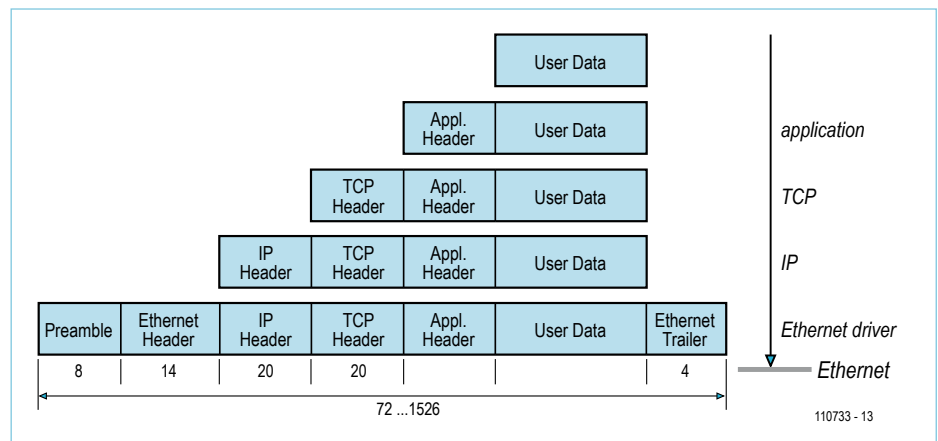


Figure 2. L'encapsulation de données TCP.

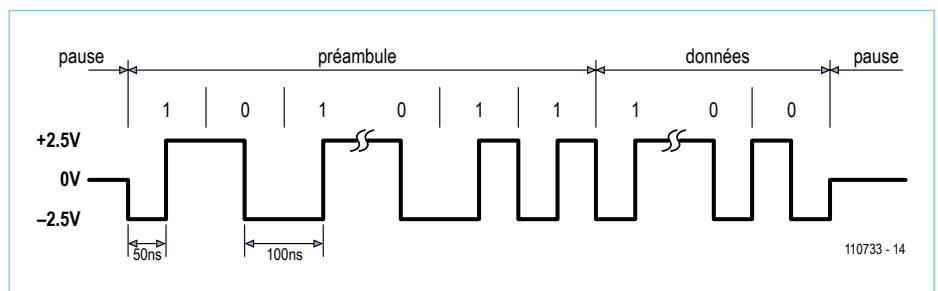


Figure 3. Le préambule d'un paquet Ethernet en code Manchester.

la même occasion le préambule qui entame chaque paquet Ethernet. Le préambule a une longueur de huit octets qui contiennent une alternance de 1 et de 0 jusqu'aux deux derniers bits, lesquels sont des 1 et signalent l'arrivée des données proprement dites. Le transfert de données s'effectue au format *Little-endian* (petit-boutiste) – donc les bits de poids faible en premier. Après les données vient le mot de contrôle CRC32 qui couvre quatre octets. La norme prescrit que les paquets Ether-

net aient une longueur comprise entre 72 et 1 526 octets. S'il y a moins de données à transmettre, il faut bourrer les paquets pour atteindre la longueur minimale. Pour toute implémentation aussi, il y a lieu d'en tenir compte.

Si vous coupez une carte à 10 Mbit/s à une autre capable de monter à 100 Mbit/s, le système reconnaîtra tout de suite que le débit binaire possible est de 10 Mbit/s. Les appareils de réseau sont rétrocompatibles et adaptent leur débit au compo-



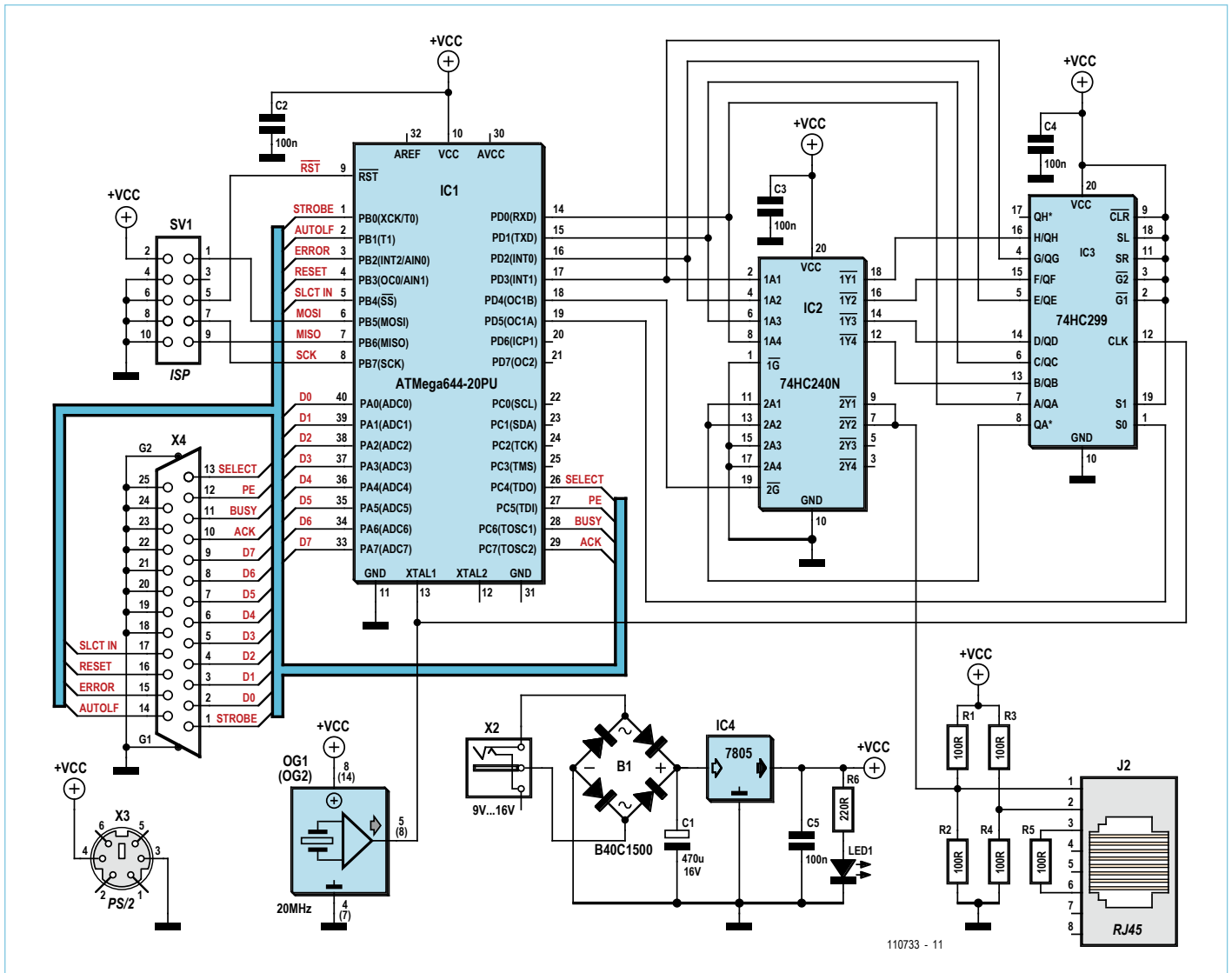


Figure 4. Le schéma du circuit du Software Defined NIC.

sant le plus lent. Pour l'identification, ces appareils envoient de courtes impulsions à intervalle régulier au partenaire capable de reconnaître la capacité de transmission. Pour 10 Mbit/s, il transmet toutes les 16 ms ( $\pm 8$  ms) des impulsions dites NLP (*Normal Link Pulse*). Il s'agit d'impulsions positives de 100 ns de large. Ici aussi, le logiciel doit reproduire ces impulsions.

### AVR-NIC

La question qui vient à l'esprit, c'est : comment un processeur à 20 MHz pour-

rait-il émettre et recevoir des signaux à 20 MHz ? À première vue, c'est tout simplement impossible. Rien que la réception et l'inscription dans la mémoire principale demandent plusieurs cycles de l'unité centrale. Interpréter en même temps le résultat, autant ne pas y penser.

Donc, sans réception, l'utilisation du TCP est impossible, puisqu'il faut accuser réception de chaque paquet, ce qui implique une communication bidirectionnelle. Il ne reste plus que l'UDP comme protocole de trans-

mission. Le transfert est moins sûr, d'accord, mais on a l'avantage de la diffusion par laquelle aucun destinataire spécifique ne doit être inscrit. De quoi permettre la diffusion sur le réseau local de données vidéo ou audio que n'importe quel ordinateur connecté peut recevoir.

L'envoi au rythme de l'horloge à 20 MHz représente aussi une tâche difficile du fait qu'il faut lire les données de la mémoire, les coder au format *Manchester* et les faire partir au bon moment. Simultanément, on doit aussi vérifier la fin des données et fina-

lement fournir le mot de contrôle CRC32. Et pourtant, le schéma à la **figure 4** vous le confirmera, le matériel supplémentaire nécessaire ne se compose que du tampon inverseur à 8 bits, le 74HC240 doté de sorties à trois états et d'un registre à décalage de 8 bits, le 74HC299, à entrée parallèle et sortie sérielle. Le microcontrôleur délivre les données en moitiés d'octets. Avec quatre inverseurs d'IC2, on en fait huit bits qui codent les quatre bits de donnée en procédé appelé *Manchester* inversé. Les huit bits, le registre à décalage IC3 les prend en parallèle et les sort en série à 20 MHz. On monte en parallèle deux inverseurs pour attaquer le câble de manière à atteindre une plus grande puissance. Du même coup, ils inversent le signal pour le mettre dans le bon sens. Chacun des deux conducteurs du câble est pourvu d'un diviseur de tension formé de deux résistances de 100  $\Omega$  pour le polariser à  $V_{CC}/2$  comme niveau au repos. Une ligne de commande maintient à haute impédance les deux inverseurs tant qu'aucun paquet n'est émis. Au début d'une transmission, ils sont libérés et génèrent le signal différentiel d'une amplitude voisine de  $\pm 2$  V. Un transformateur spécial n'est pas nécessaire, le signal est envoyé directement à une prise modulaire à 8 contacts RJ45.

### Calculs Ethernet

La sortie est ainsi capable de suivre en temps réel, mais il faut que les paquets de données assemblés avec les en-têtes et les remorques soient calculés au préalable et inscrits en RAM, ensuite le contrôleur sera utilisé à plein régime pour émettre les données. Par conséquent, avant de commencer vraiment la transmission, il faut que les données adéquates des parties variables de l'en-tête soient déjà remplies. Le mot de contrôle aussi doit avoir été calculé.

La **figure 5** montre la structure et le contenu de l'en-tête utilisé. Dans celui pour Ethernet, il y a trois champs à remplir. L'adresse MAC de l'émetteur est au choix, elle doit seulement être différente de celles des autres dans le réseau local. S'il n'y a qu'un seul ordinateur ciblé, on peut utiliser pour le récepteur une adresse MAC réelle. Comme variante, on pourra se servir de l'adresse MAC de diffusion, elle se compose de six

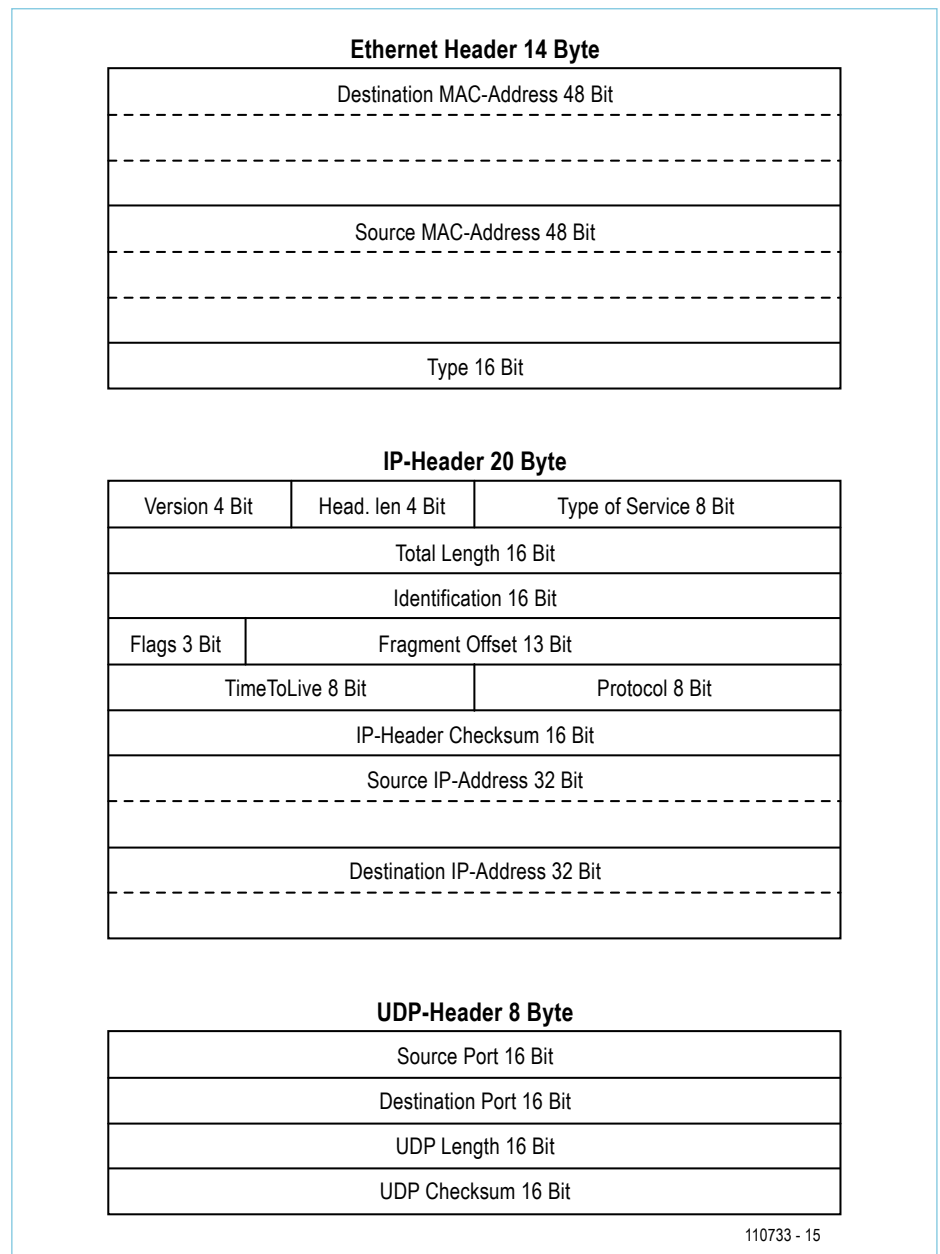


Figure 5. L'en-tête Ethernet utilisé par SDN.

octets 0xFF. Avec elle, il est théoriquement possible d'atteindre n'importe quel ordinateur dans le monde. Dans le dernier champ de l'en-tête Ethernet, le Type, on inscrit la donnée 0x0800, une constante qui indique qu'il s'agit d'un paquet IP. Dans un en-tête IP, on peut attribuer à plusieurs champs des constantes fixes. Il y a entre autres le numéro de version IP, la durée de vie du

paquet et UDP comme couche incrustée. Les adresses IP de l'émetteur et du récepteur peuvent aussi demeurer statiques, puisqu'elles visent le réseau local. Pour la diffusion, il reste à remplir de « 1 » les derniers bits du masque de réseau. Le masque de réseau divise l'adresse en une partie réelle du réseau et la partie qui concerne les participants au sous réseau local.

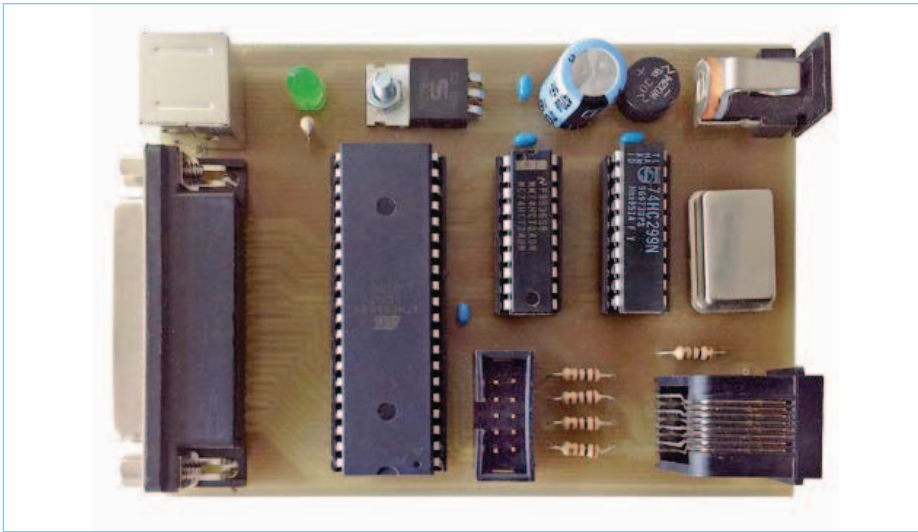


Figure 6. Le prototype terminé de l'auteur.

Avec comme adresse réseau 192.168.1.0 et un masque de 255.255.255.0, l'adresse de diffusion est 192.168.1.255. Mais il existe aussi des champs à déterminer de nouveau pour chaque paquet de données. Par

exemple, le champ *Total Length* qui englobe la longueur du paquet IP entier, en-têtes et données compris. Ainsi, quand on peut envoyer des données de différentes longueurs, celle du paquet sera à recalculer

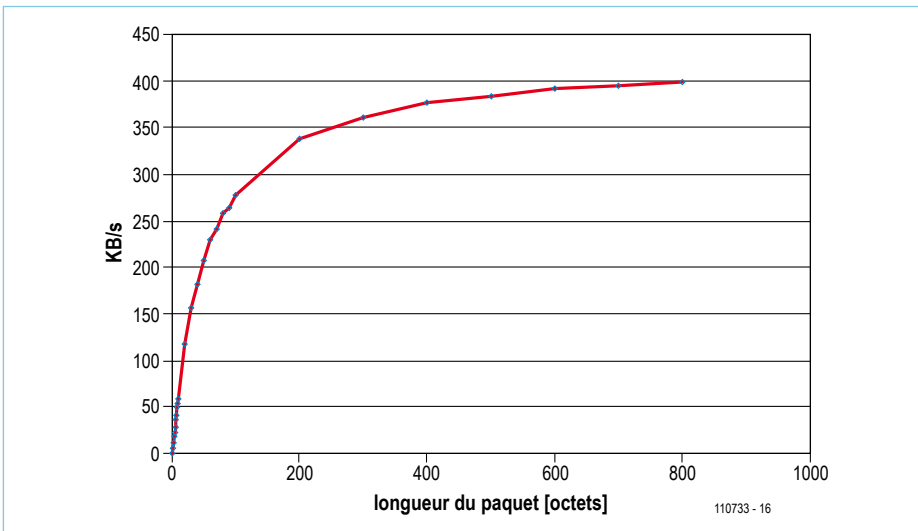


Figure 7. Le débit binaire de SDN en fonction de la longueur des paquets.

chaque fois. En outre, il y a lieu de calculer le mot de contrôle de l'en-tête IP et de le stocker dans le champ *IP Header Checksum*. Dans le champ *Identification*, on peut éventuellement insérer un numéro de série pour identifier le paquet.

Dans l'en-tête UDP, on inscrit les ports des processus d'émission et de réception. Ce sont normalement des valeurs constantes qui correspondent au port du processus de réception. Les numéros de port inférieurs à 1 000 sont à éviter, car nombre d'entre eux sont déjà attribués. Le champ *UDP Length* est comparable à *IP Length* et sera recalculé chaque fois en fonction de la grandeur du paquet. Le mot de contrôle UDP est facultatif, nous pouvons par chance nous en passer dans cette version-ci.

Finalement, on vérifie si le paquet entier a la longueur minimale requise. Dans le cas contraire, on bourre le paquet d'un nombre correspondant d'octets. Le contenu de ces « octets de bourrage » (*padding*) est arbitraire. Ce n'est qu'après que le mot de contrôle du paquet de données final pourra être calculé. Le préambule Ethernet n'intervient pas dans le calcul, mais bien les octets de bourrage. Le résultat du calcul est ajouté en terminaison du paquet. Le code source pour le calcul du CRC32 a été emprunté à un noyau Linux, compilé, puis désassemblé et en fin de compte optimisé à la main.

Il existe une bibliothèque en assembleur GNU qui remplit toutes les tâches évoquées et qui est également compatible C. Avec elle, on peut programmer toutes les applications en GCC AVR (par exemple, dans l'AVR Studio 5). La bibliothèque met à disposition deux fonctions. `UDPInit()` demande comme paramètre un pointeur vers une

## Webcam IP

Comme application d'expérimentation, j'ai connecté un ancien modèle de caméra, par un vieux port parallèle, aux lignes restées libres du microcontrôleur. Le code du pilote « `cam.c` » et l'en-tête « `cam.h` » pour la caméra sont inspirés d'un pilote DOS sous GPL que j'ai adapté pour le contrôleur. Il prend le signal vidéo ligne par ligne et forme de chacune d'elles un paquet UDP

en lui attribuant un numéro de série. Le logiciel pour PC « `canvas.exe` » reçoit les paquets UDP et reconstitue l'image. Cela donne environ deux images par seconde, ce qui suffit pour des tâches simples comme la surveillance. La caméra IP simple ainsi réalisée avec du matériel pour environ 10 euros constitue une belle démonstration du *Software Defined NIC* décrit ici.

Si vous voulez la reproduire telle quelle, la caméra est une *Creative Labs Video Blaster WebCam II* modèle 1100001424 pour port parallèle, on en trouve certainement pour quelques sous sur [ebay.com](http://ebay.com).



zone de mémoire dans laquelle sera stocké ultérieurement l'ensemble du paquet Ethernet. La fonction inscrit ensuite toutes les informations d'en-tête statiques dans cette zone de mémoire. En outre, elle initialise un temporisateur à 8 bits, Timer0, qui appelle toutes les 13 ms une routine d'interruption pour envoyer les NLP. Les données sont inscrites 50 octets après le début du domaine de stockage des paquets. La seconde fonction, UDPSEND() calcule, sur base du paquet de données, les informations qui manquent encore dans l'en-tête, corrige au besoin la longueur du paquet, calcule le mot de contrôle CRC32 et envoie le paquet sur le réseau. Cette fonction attend comme paramètres deux pointeurs qui désignent le début du paquet et la fin des données utiles.

### Construction et mise en service

Le SDN se sert des six lignes inférieures de n'importe quel port d'un contrôleur AVR. Sur quatre d'entre elles passent les données, les deux autres commandent le registre à décalage et les inverseurs à trois états. Le contrôleur prend le rythme d'un oscillateur externe à 20 MHz. Ce signal d'horloge sert également au registre à décalage. Un matériel aussi réduit, juste IC2 et IC3, peut être installé sur une platine à simple face très compacte. Vous trouverez sur le site d'Elektor [2] les dessins de la platine ainsi que le fichier du code source, gratuits au téléchargement. La **figure 6** vous montre mon prototype.

En AVR Studio, on utilise le fichier « udp.S » comme source dans le projet. Le listage udp\_avr.c montre un programme de test simple. Il alloue une plus grande place en mémoire que les 54 octets de données utiles. Les 50 premiers octets sont réservés à l'en-tête et les 4 derniers au mot de contrôle. Pour pouvoir aussi envoyer des paquets plus petits sans avoir à changer chaque fois la longueur du tampon de mémoire, on passe à la fonction d'émission un second paramètre qui pointe vers la fin du domaine des données à transmettre. Quand le circuit est connecté à un réseau, on peut observer l'arrivée des données sur un PC où tourne le logiciel gratuit *Wireshark* [3].

#### Listage de udp\_pc.c

```
#define BUFLen 800
#define PORT 6668

main()
{
    SOCKET s;
    struct sockaddr_in adresse;
    unsigned char buf[BUFLen];
    int num;

    if((s = socket(AF_INET, SOCK_DGRAM, 0)) < 0)
        printf("no socket %d\n", s), exit(0);
    adresse.sin_family = AF_INET;
    adresse.sin_port = htons(PORT);
    adresse.sin_addr.s_addr = htonl(INADDR_ANY);

    if(bind(s, (struct sockaddr*) &adresse, sizeof(adresse)) != 0)
        printf("bind failed\n"), exit(0);

    num = recv(s, buf, BUFLen, 0);
    // do something
}
```

### Source PC

Pour utiliser les données envoyées par SDN sur un PC avec un programme adéquat, il faut être en mesure de recevoir des paquets UDP. Le listage udp\_avr.c montre que quelques lignes en C (GCC sous cygwin) suffisent pour recevoir les données par l'intermédiaire d'un *socket UDP*. Après avoir constitué le socket, il n'y a plus qu'à appeler la fonction « recv() ». Mais sur PC, nul n'est tenu par le C. Pour la réception de données, les sockets nécessaires existent aussi dans d'autres langages évolués comme Java ou Delphi.

### Il est beau le débit

Avec l'Ethernet 10 Mbit, le débit n'excède pas 1 Mo/s. La **figure 7** montre la courbe du débit binaire que l'on peut atteindre comme fonction de la longueur des paquets. On s'aperçoit que l'on ne peut atteindre « que » 400 ko/s, à cause de la complexité du calcul du mot de contrôle CRC32. Les mesures ont montré que le calcul de ce mot prend en fait plus de temps que la transmission effective des données.

(110733 – version française : Robert Grignard)

### Liens

- [1] [http://research.microsoft.com/en-us/um/people/gbell/Ethernet\\_Blue\\_Book\\_1980.pdf](http://research.microsoft.com/en-us/um/people/gbell/Ethernet_Blue_Book_1980.pdf)
- [2] [www.elektor.fr/110730](http://www.elektor.fr/110730)
- [3] [www.wireshark.org](http://www.wireshark.org)
- [4] <http://userpages.uni-koblenz.de/~physik/informatik/techC.pdf>

## À propos de l'auteur

Le Dr Merten Joost est professeur à l'université de Coblenz et enseigne l'électronique numérique, les microcontrôleurs et la robotique. Le contenu de cet article fait partie de son cours sur les microcontrôleurs [4] dans lequel, à côté des bases des réseaux informatiques, il expose les méthodes efficaces de programmation des microcontrôleurs.

# FORUM LED EUROPE

CONFERENCES  
+ EXPO

21/22 NOVEMBRE 2012

PARIS – FRANCE

Grande Halle de la Villette

LA PLATEFORME LA PLUS EFFICACE  
POUR LES MATERIELS D'ÉCLAIRAGE  
ET LES TECHNOLOGIES LED

## CONGRÈS

2 jours pour rencontrer les  
meilleurs spécialistes  
internationaux

## EXPO

ForumLED  
Technologies

**new**  
ForumLED  
Lighting Products

Organisé par

**Cdo**  
CONSEIL DÉVELOPPEMENT ORGANISATION

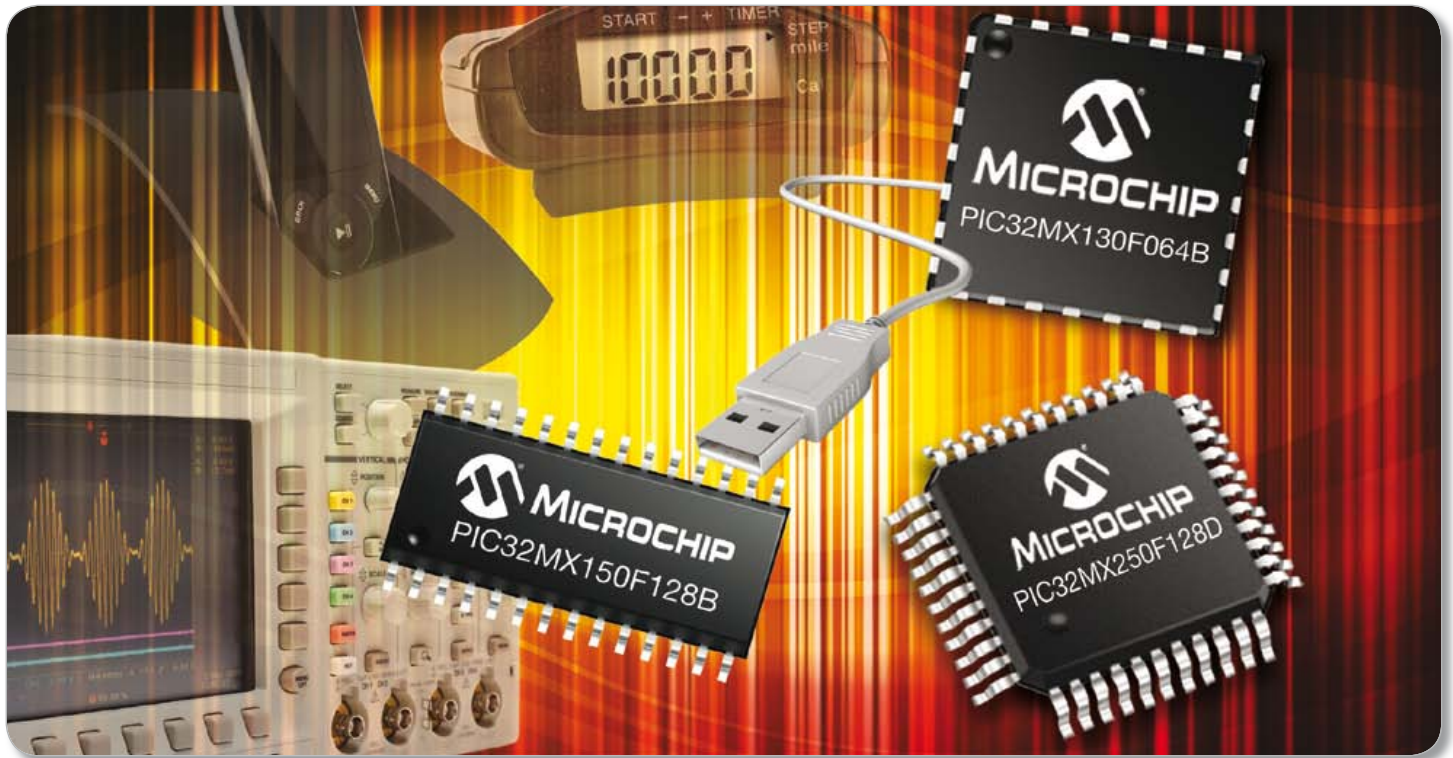
Partenaire principal

**Cluster  
Lumière**  
Performance & Innovation

[www.forumled.com](http://www.forumled.com)

# Offrez des performances 66 DMIPS à vos systèmes à microcontrôleurs 32 bits

Microcontrôleurs PIC32 32 bits économiques et peu encombrants



**Les microcontrôleurs PIC32 les plus petits et les plus économiques de Microchip vous permettent désormais d'ajouter des fonctions de lecture/enregistrement audio, des boutons ou curseurs tactiles capacitifs et l'USB On-the-Go (OTG), le tout pour une vitesse de 66 DMIPS à seulement 40 MHz, dans des boîtiers miniatures de 5 mm x 5 mm.**

Les nouveaux microcontrôleurs PIC32MX1 et MX2 intègrent une mémoire Flash de 128 ko et 32 ko de RAM, deux interfaces CODEC audio I<sup>2</sup>S™, le tactile capacitif, un module port parallèle maître (Parallel Master Port) PMP 8 bits pour l'ajout de graphismes ou de mémoire externe, un CAN sur puce à 1 millions d'échantillons par seconde (MSPS) ainsi que l'USB On-the-Go (OTG) et d'autres périphériques de communication série. Ils sont disponibles en boîtiers miniatures de 5 mm x 5 mm, avec un brochage compris entre 28 à 44 broches (et VTLA 36 broches), et possèdent une plage de température de fonctionnement qui s'étend de -40°C à +105°C.

Ces nouveaux composants vous permettent d'économiser de l'espace et de simplifier vos architectures grâce à la sélection PPS permettant de remapper numériquement des broches. Pour autant, la compatibilité avec la famille des PIC24 16 bits est maintenue, garantissant ainsi une migration facile vers la puissance de calcul 66 DMIPS de l'architecture 32 bits des PIC32.

## OUTILS DE DÉVELOPPEMENT POUR UN DÉMARRAGE RAPIDE

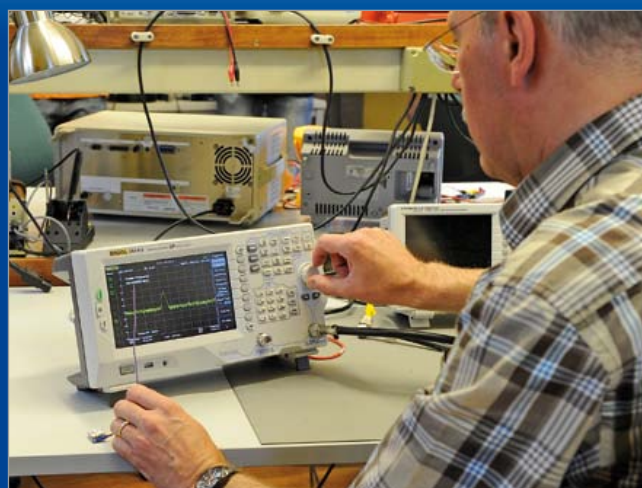
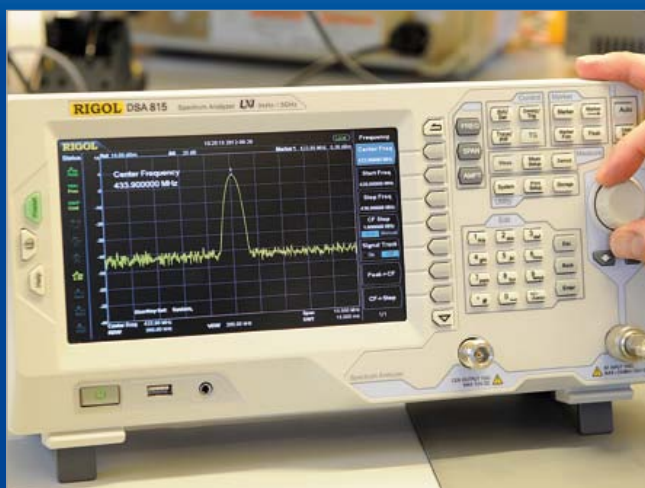
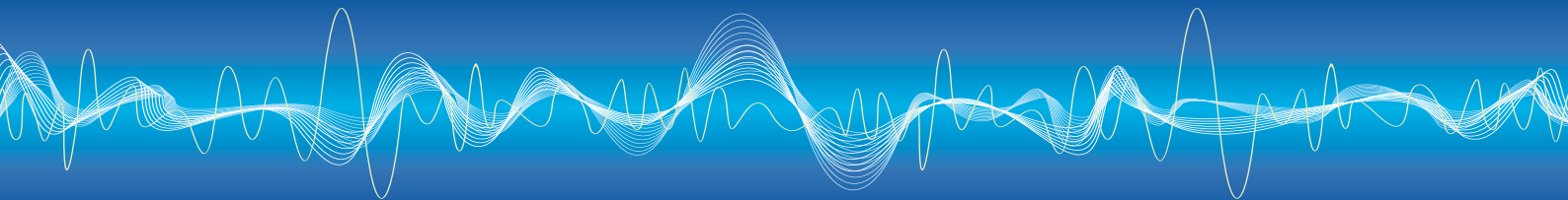
- Kit d'évaluation MPLAB® pour PIC32MX1XX/2XX (DM320013)
- Carte d'évaluation CTMU PIC32MX (AC323027)
- Module plug-in PIC32MX220F032D (MA320011) pour Explorer 16
- Plateforme de développement Microstick II pour microcontrôleurs et DSC 16 et 32 bits (DM330013-2)

**Ajoutez des possibilités audio et tactiles à vos systèmes à faible coût dès aujourd'hui : [www.microchip.com/get/eurmx1mx2](http://www.microchip.com/get/eurmx1mx2)**



**Microcontrollers • Digital Signal Controllers • Analog • Memory • Wireless**





# Rigol : analyseur de spectre abordable

Harry Baggen (Elektor)

« J'ai ici un superbe analyseur, vous devriez l'essayer. C'est fou ce qu'il offre comme possibilités pour un millier d'euros », me disait récemment au téléphone le distributeur pour le Benelux de la firme chinoise *Rigol*. C'était prometteur, en effet : un analyseur de spectre dont la plage s'étend sur 1,5 GHz avec un générateur de poursuite pour seulement 1 058 € hors TVA. Sûr que j'avais envie de tester un tel appareil dans notre laboratoire, même si, entre nous, « des instruments de mesure aux possibilités étonnantes », j'en ai déjà vu défilé des régiments entiers !

J'ai donc pris rendez-vous et j'espérais disposer de quelques jours pour me faire la main avec cet appareil, mais avant même que j'aie pu mettre les choses en route, nous étions déjà en pleine production du numéro suivant d'Elektor et la période de réservation de l'analyseur approchait de sa fin. Pire encore, Jan Buiting, le seul radioamateur de la maison, qui tenait à participer aux essais, venait de partir en vacances (comment fait-il pour boucler son édition anglaise avant les autres, mystère...).

Sachant que les amateurs sans-filistes nous traitent de « péquenots du courant continu » parce que notre boulot ne nous emmène que rarement au-delà des 100 kHz, mon collègue Ton Giesberts, notre Monsieur Audio, et moi, nous demandions comment nous y prendre. Comme nous tenions à réaliser ces épreuves sérieusement pour connaître exactement les possibilités de l'appareil, nous avons rassemblé quelques circuits HF sur lesquels on travaillait au labo et les mesures ont pu commencer.

## Une bonne impression

Le *Rigol DSA 815-TG* est un analyseur de spectre doté d'une bande passante allant de 9 kHz jusqu'à 1,5 GHz. Avec pareilles spécifications, on paye largement le double pour un tel analyseur de spectre et on n'a toujours pas de générateur de poursuite. Il a de quoi appâter les radioamateurs avec ses nombreuses possibilités. Pour 1 058 euros, on peut le considérer

comme une bonne affaire, du moins pour des amateurs qui ont les moyens de satisfaire leurs désirs !

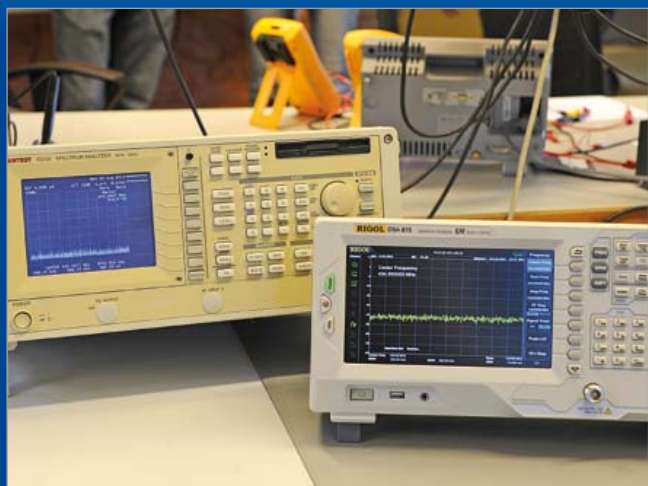
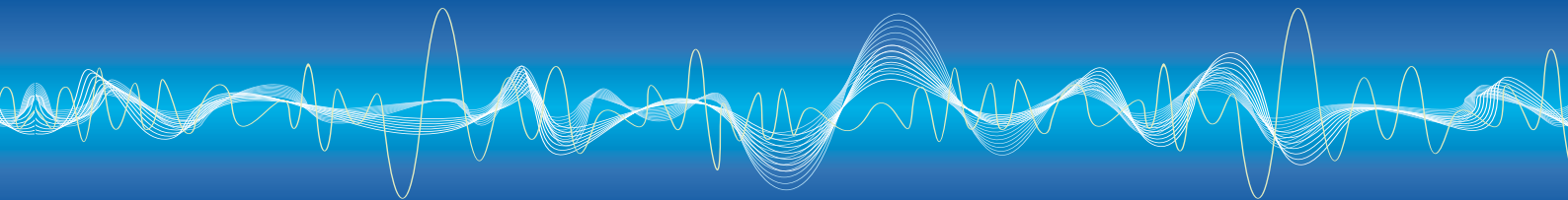
Ce qui frappe en débattant l'analyseur, c'est son poids. D'un bon quatre kilos, c'est assurément plus lourd qu'un oscilloscope de la même gamme de prix. La finition est bonne et, contrairement à l'habitude de beaucoup d'autres fabricants chinois, la présentation est heureuse, volontairement sobre avec des couleurs discrètes. Vous trouvez sur l'appareil à la fois une connexion USB hôte et une esclave de manière à brancher une clé USB tout en conservant la liaison avec un ordinateur. Il y a même un connecteur Ethernet pour se servir de l'analyseur à distance dans un réseau.

À l'allumage, l'écran de 20 cm (8 pouces) offre une grande image de haute qualité, de 800 x 400 pixels, qui reste clairement visible de biais. À côté de l'affichage, un panneau de commande comporte de nombreuses touches et un bouton rotatif. En bordure d'écran, autour de l'image de mesure, une variété de réglages figurent à côté des points du menu qui correspondent à l'enfilade de touches voisines. Cette disposition ressemble fort à celle d'oscilloscopes et d'analyseurs d'autres marques. On est tenté d'appuyer sur les points du menu affichés à l'écran plutôt que sur les touches... la prochaine version sera peut-être tactile ! Dans le contexte actuel, nous sommes gâtés avec les tablettes et les téléphones tactiles. Espérons que l'importateur n'aura pas trop de mal à effacer de l'écran nos empreintes digitales !

## Les premières mesures

Au début, on chipote un peu dans tous les menus pour trouver son bonheur. Non que leur disposition soit mauvaise, loin de là, mais l'appareil dispose de tant de possibilités et de réglages que parfois l'arbre cache la forêt. Le manuel sur CD qui accompagne l'instrument n'est pas d'un grand secours à cet égard, il compte pas moins de 220 pages. Chaque fonction ou réglage y est minutieusement décrit, mais on s'y perd tout autant. Après quelques temps, on finit par trouver son chemin, c'est du moins l'expérience que nous avons eue avec nos nouveaux oscilloscopes au laboratoire.

En manière de test, nous avons branché le prototype d'un circuit d'émetteur à 433 MHz destiné à la recherche et la



# bip-bip-bip... c'est quoi, ce bip ?

Thijs Beckers (Elektor)

récupération de modèles réduits d'avions. Il a vite trouvé la porteuse à 433 MHz, puis la touche Auto de l'analyseur a placé un agrandissement de la porteuse au centre de l'écran. En retournant au balayage complet sur 1,5 GHz, nous avons constaté que la deuxième harmonique n'était qu'à 12 dB sous la fondamentale. Eh là, est-ce bien conforme à la norme ? C'est à approfondir, tout cela !

C'est principalement pour la mesure de filtres que l'association de l'analyseur et du générateur dans le même appareil est idéale. On n'a besoin de rien d'autre pour vite déterminer par l'image le comportement d'un circuit. Dans ce cas, la fonction de normalisation est très pratique, vous pouvez compenser toute dérive dans la combinaison d'un générateur et d'un analyseur pour exécuter la mesure proprement dite.

Lors de l'examen sur écran des résultats de mesure, l'analyseur de spectre offre énormément de possibilités de réglages et de mesures, comme la résolution de bande passante, la période de balayage, les divisions de l'échelle, etc. Sur l'écran, l'image procure une grande richesse d'informations sans qu'elles ne troublent la lecture de la fenêtre de mesure. Vous pouvez bien sûr y placer différents marqueurs pour les mesures, enregistrer aussi bien les valeurs mesurées que les réglages et les exporter. Nous pourrions encore étendre la liste des fonctions, mais restons-en là.

En l'absence de notre radioamateur, nous n'avons pas pu faire d'autres expériences avec l'analyseur de spectre, mais après cette brève exploration, nous avons gardé une impression positive du DSA 815. Son prix comparativement bas et la profusion de possibilités, ainsi que la présence du générateur de poursuite rendent cet instrument légitimement désirable pour tout radioamateur ou mordu de haute fréquence.

P.-S. Sur la photo, le Rigol cotoie notre propre analyseur **Advantest**, qui a plus de 20 ans. Remarquez la similitude dans la disposition des commandes !

(120560 - version française : Robert Grignard)

[www.arbenelux.com/en/](http://www.arbenelux.com/en/)  
<http://rigol-france.com/>  
<http://eu.rigol.com/>

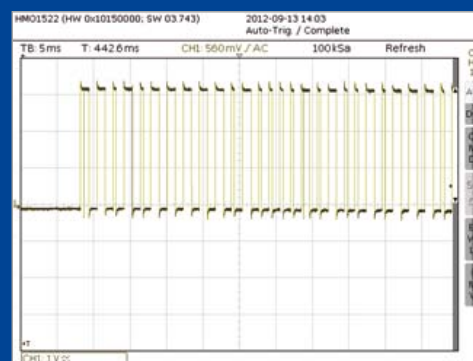
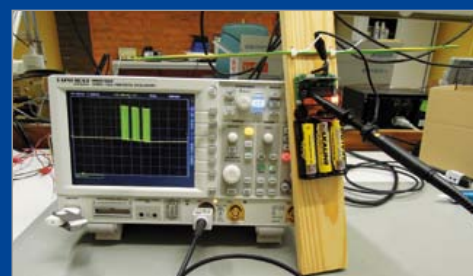
Au labo d'Elektor, Ton G. (voir la visite guidée, ailleurs dans ce numéro) a travaillé sur une balise pour vous aider à retrouver un modèle réduit volant... qui a cessé de voler plus tôt que prévu. Le prototype a marché. Et pourtant, si j'en parle ici, c'est qu'il n'y avait pas quelque détail croustillant à révéler. L'émetteur fait ce qu'on lui demande, mais le récepteur en fait un peu plus. Il ramasse un signal répétitif qui de trois bips successifs, espacés d'une minute environ. À l'oscillo, il apparaît que ce gazouillis ressemble à des signaux de communication à codage Manchester (ou codage biphasé). Le récepteur est calé sur la bande ISM de 433 MHz, mais nous n'avons pas l'ombre d'une idée de l'origine de ce signal. S'agirait-il de notre centrale d'alarme ou d'un détecteur de fumée ?

Il est vrai que notre récepteur ramasse effectivement le signal émis par une clé d'auto. Un truc utile à retenir d'ailleurs, pour la prochaine fois que vous aurez à bidouiller sur le décodage d'une alarme pour auto. Ah, ce n'est donc pas vous qui avez imaginé le stratagème suivant pour voler des voitures sans les abîmer : vous survolez un immense parking plein de voitures avec un petit hélico télécommandé, équipé d'un émetteur qui envoie des salves de codes de déverrouillage. Une petite caméra Wifi accrochée à l'hélico vous permet de repérer les voitures qui répondraient au code produit aléatoirement. Si le modèle vous convient, il n'y a plus qu'à ouvrir la portière et partir avec le véhicule sans casse ni intrusion.

Mais revenons à nos parasites déjà variés. Ces essais ont montré qu'il ne fallait pas négliger non plus le rayonnement intense du spectre de bruit des écrans TFT, ou plus précisément de leur alimentation à découpage, que ce soit l'écran de l'oscilloscope ou celui de l'ordinateur. Ton G. voit là une raison de plus de prévoir un blindage pour son récepteur de balise de détresse.

Et toujours pas de trace de la source de ces étranges bip-bip-bip. Il va falloir que je me mette en chasse un de ces jours, le TAPIR [1] à la main...

(120452)



## Liens

[1] [www.elektor.fr/120354](http://www.elektor.fr/120354)



# Flowcode 5 pour concev

## FLOWCODE5

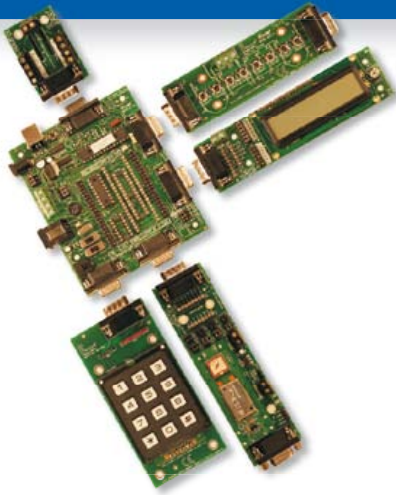


**Flowcode 5** est l'un des langages de programmation graphique pour microcontrôleurs (PIC, AVR, ARM et dsPIC/PIC24) les plus avancés au monde. Son avantage principal est de permettre la création de systèmes électroniques et robotiques complexes même si l'on manque encore d'expérience.

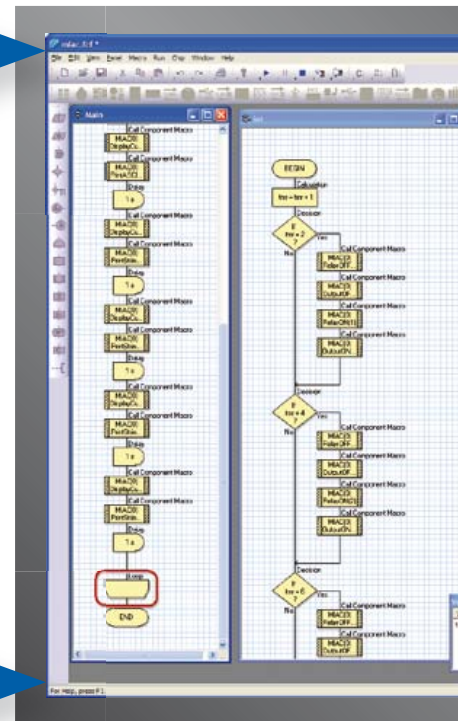
**Flowcode est utilisé :**

- dans l'enseignement, pour initier les étudiants à la programmation
- dans l'industrie, autant pour le prototypage rapide que pour des projets d'envergure

## ... en électronique



Les E-blocks sont des circuits électroniques compacts, correspondant chacun à une fonction autonome comme on les trouve dans les circuits embarqués. Il en existe une quarantaine, dont la complexité va croissant, depuis le simple afficheur à LED jusqu'au circuits de programmation, aux modules Bluetooth ou TCP/IP. Les E-blocks peuvent être assemblés aisément pour élaborer des systèmes propices à l'apprentissage par l'expérimentation. Ils conviennent aussi pour le prototypage rapide de systèmes complexes. L'ensemble est complété efficacement par une gamme étendue et sans cesse renouvelée de logiciels puissants, et de capteurs variés.



## ... pour la commande industrielle



MIAC (**M**atrix **I**ndustrial **A**utomotive **C**ontroller) est une unité de commande industrielle pour circuits électroniques variés avec pour champs d'application privilégiés la capture, la mesure, la surveillance et l'automatisation.

Le MIAC lui-même est construit autour d'un puissant microcontrôleur PIC de la série 18 qui se connecte directement au port USB et se programme en Flowcode, en C ou en assembleur. Flowcode est fourni avec le MIAC, lequel est équipé d'origine du bus CAN, qui facilite la connexion en réseau de plusieurs MIAC.

## Outil de débogage FlowKit

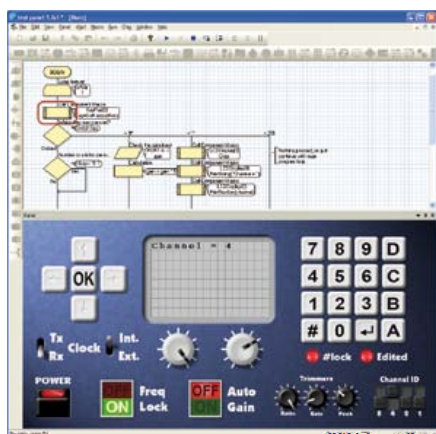
L'outil FlowKit offre la fonction ICD (In Circuit Debug) pour une gamme étendue d'applications Flowcode dans des projets PIC et AVR :

- marche, arrêt, pause et pas-à-pas pour programmes en Flowcode en temps réel
- suivi des variables de votre programme
- modification des variables
- débogage en circuit du robot mobile Formula Flowcode Buggy, ECIO et de projets autour de MIAC





# ... et pour se former...

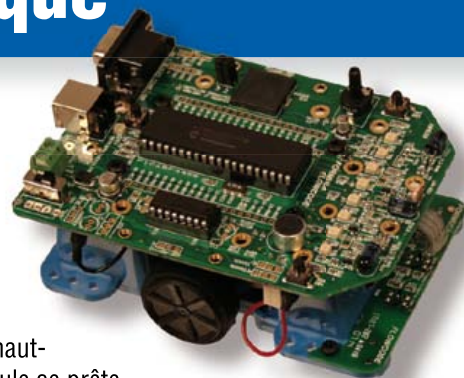


## NOUVEAU dans Flowcode 5 :

- Nouvelle présentation personnalisable du code C
- Simulation améliorée
- Fonction de rechercher et de remplacement
- Nouveaux types et nouvelles fonctions des variables, des constantes et des variables de port
- Documentation automatique du projet
- Codage facilité par le nouvel explorateur de projet
- Mise en place de signets de code pour la navigation dans le programme
- La refonte complète du système d'interruption offre aux développeurs l'accès direct à plus de fonctions intégrées
- Amélioration de la signalisation des erreurs de compilation
- Désactivation de fonctions des icônes
- Amélioration des annotations
- Amélioration des liens vers les supports média

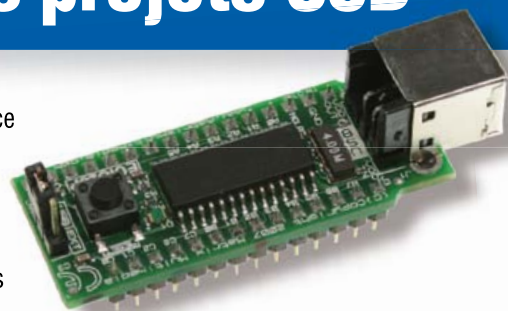
## ... en robotique

Formula Flowcode Buggy est le nom d'un petit robot mobile remarquable par son rapport performances/prix. Ce véhicule fournit aussi bien un support adéquat pour l'apprentissage de la robotique, qu'une plateforme idéale pour des compétitions de robotique. Loin d'être un jouet, ce robot programmable par l'USB, est doté d'une détection de ligne, de capteurs de proximité, de 8 LED incorporées, d'un capteur sonore, d'un haut-parleur et du connecteur d'extension E-blocks. Ce véhicule se prête à de nombreux exercices de robotique depuis la simple détection de ligne jusqu'à l'analyse de labyrinthe. Le connecteur d'extension E-blocks autorise l'adjonction d'afficheurs, ou de modules Bluetooth, ZigBee ou d'un GPS.



## ... pour les projets USB

Les ECIO sont de puissants modules à microcontrôleurs programmables par l'interface USB, avec une empreinte au choix de 28 ou 40 broches au standard DIL (0,6 pouce). Construits sur des microcontrôleurs des séries PIC18 et des ARM7, les modules ECIO conviennent bien à l'étudiant et à l'autodidacte. Ils sont programmables en Flowcode, en C ou en assembleur. De nouvelles routines USB sous Flowcode favorisent le développement rapide de projets autour de l'USB, avec y compris les fonctions USB HID, USB esclave, et USB bus sériel (PIC seulement). Vous pouvez incorporer ECIO à vos propres circuits de façon à les doter de la fonction de reprogrammation.

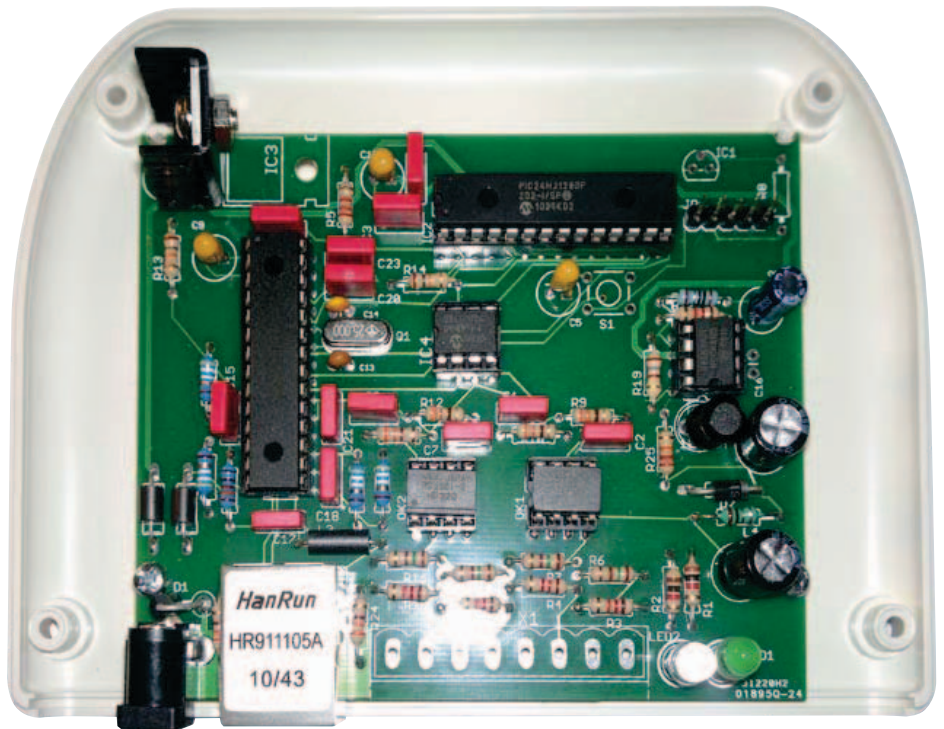


Retrouvez les E-blocks et leur documentation sur :  
**[www.elektor.fr/eblocks](http://www.elektor.fr/eblocks)**

# interface internet pour compteur électrique domestique

Søren Mikkelsen (Danemark)  
avec le concours de  
Hans Henrik Skovgaard (Danemark)

Frémissez-vous de culpabilité quand on vous parle de votre empreinte carbone ? Si vous souhaitez réduire l'une et l'autre, il faut commencer par connaître à tout moment la quantité d'électricité que vous consommez et que vous payez. Et si l'internet pouvait nous aider à la déterminer et à la surveiller ?



## openEnergy et openSource pour énergimètre So

Parmi les différentes formes d'énergie qui alimentent les ménages, l'électricité occupe incontestablement une place prépondérante. Dans certaines maisons, c'est même la seule source d'énergie.

Le raccordement au réseau électrique implique normalement la présence d'un compteur (énergimètre) que l'on peut relever de différentes manières selon la technologie employée. Les plus anciens modèles (à disque Ferraris) n'ont qu'un compteur mécanique à cadran qu'il faut lire. D'autres modèles, plus évolués, sont capables de transmettre directement la consommation au fournisseur d'électricité.

Parfois, on aimerait mesurer la consommation d'un appareil domestique en particulier. Récemment dans les médias, on conseillait justement d'installer un comp-

teur intermédiaire sur les pompes à chaleur afin d'analyser leur consommation électrique et, au final, de s'assurer que le compresseur fonctionne à bon escient.

### Interface S0

Il existe plusieurs façons de mesurer la consommation électrique. Intéressons-nous ici à un énergimètre du commerce muni d'une interface S0, ainsi qu'une passerelle Ethernet à usage domestique.

L'interface S0 répond à la norme DIN 43864. Elle définit une interface électrique qui transmet des impulsions à partir d'un générateur jusqu'à un appareil de tarification. La sortie d'un S0 supporte une tension maximale de  $27 V_{CC}$  et le courant maximum à sa sortie est de 27 mA. Une impulsion dure généralement 30 ms.

Outre la consommation électrique, cette interface sert aussi pour les consommations d'eau et de gaz. La quantité consommée et représentée par une impulsion dépend de l'appareil qui transmet les impulsions. Dans le cas de l'électricité, ce pourrait être 1 000 impulsions par kilowattheure (kWh). L'usage d'un compteur à interface S0 implique généralement qu'il soit intercalé entre le réseau électrique et l'appareil qui consomme. Dans la plupart des pays, cette intervention ne peut être réalisée que par un électricien agréé, à cause des risques liés à l'électricité.

Certains compteurs installés par les fournisseurs d'électricité contiennent déjà une interface S0. Si le vôtre a une LED qui clignote, il y a de grandes chances qu'il soit équipé d'une interface S0.

## Enregistrer les données de consommation

Il ne suffit pas de compter les impulsions, il faut encore les enregistrer pour analyser les données récoltées. L'enregistrement peut s'effectuer localement ou bien sur un

serveur extérieur (cloud). Avec les capacités actuelles des connexions internet, la quantité de données à stocker ne risque pas de saturer la bande passante. Pour enregistrer et afficher ces données de consommation, nous avons choisi une solu-

tion à code source ouvert. Cette solution provient du projet *openenergymonitor* [1]. Elle utilise le langage PHP et la base de données MySQL. Difficile de trouver une solution plus abordable. En tant que projet *open source*, son code est libre. Pas besoin d'être

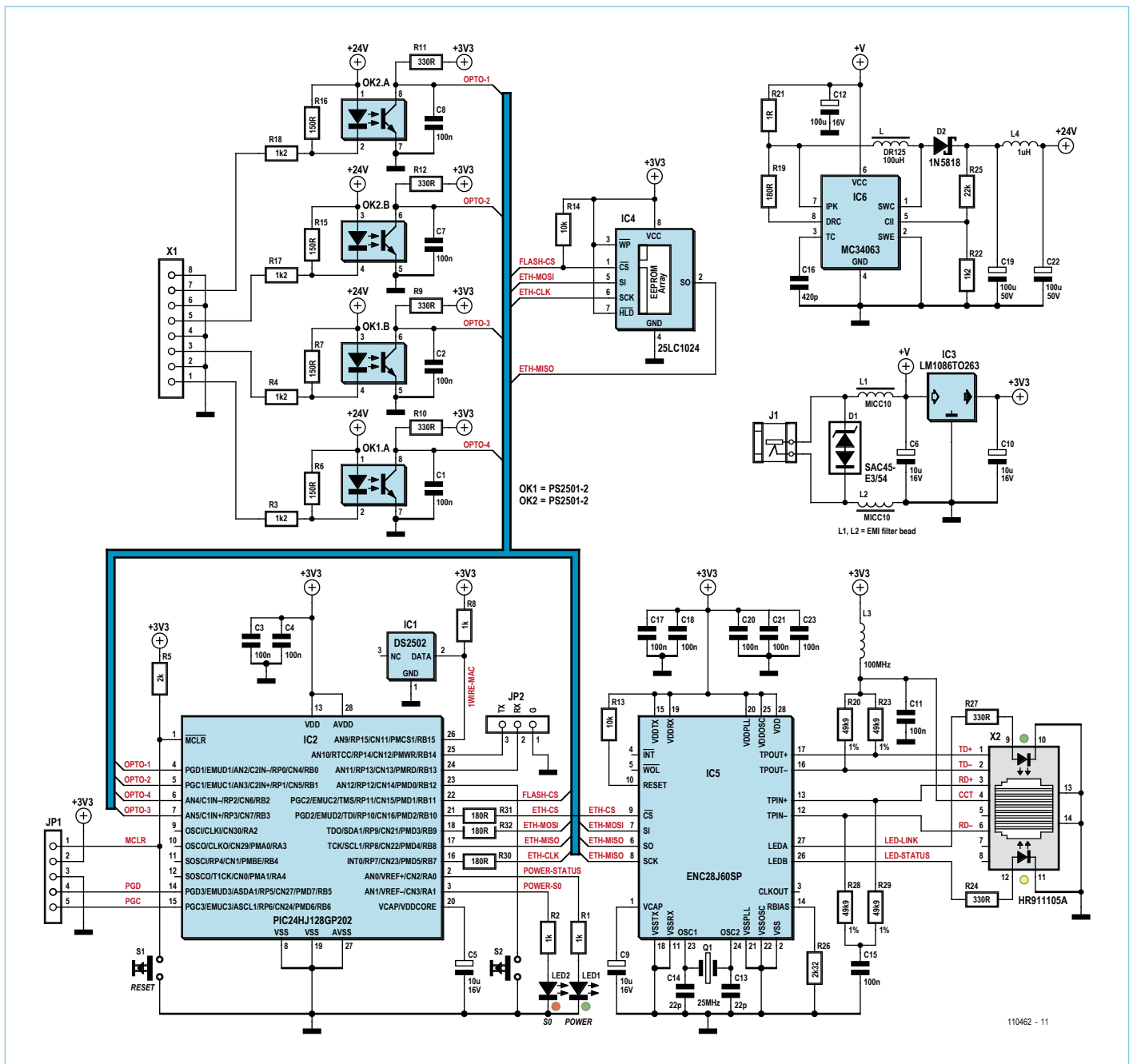


Figure 1. Schéma de l'interface Puls0 pour énérgimètres compatibles S0. Elle dispose de quatre entrées S0.



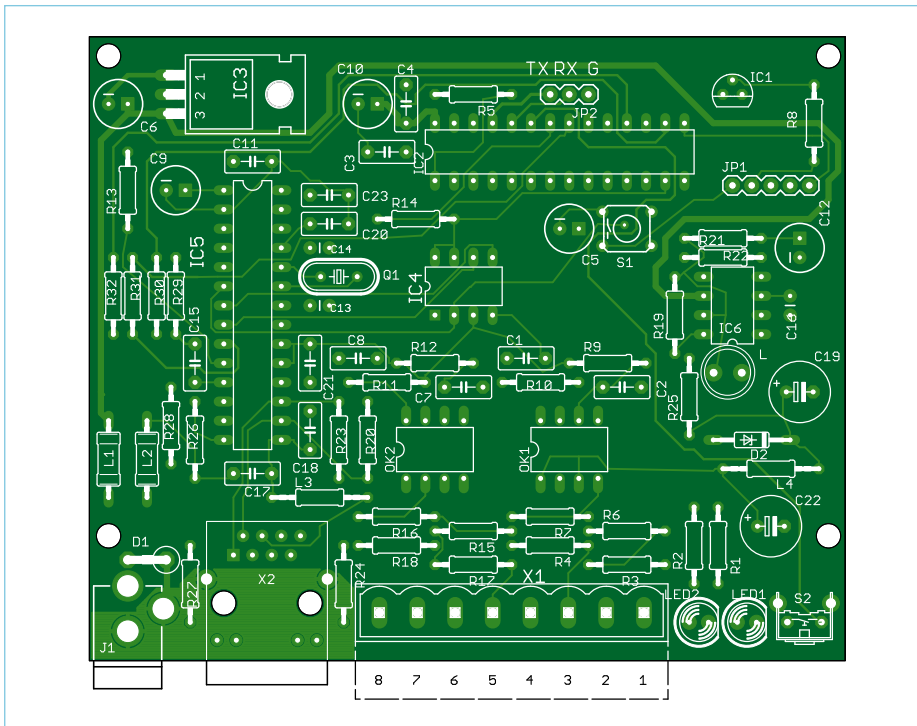


Figure 2. La platine de l'interface énérgimètre par internet. Remarquez qu'il n'y a que des composants à fils !

un expert de la programmation réseau pour l'utiliser. Vous avez le choix entre un service d'hébergement à prix démocratique ([www.emoncms.org](http://www.emoncms.org)), mais vous pouvez aussi l'héberger sur votre propre serveur ou chez n'importe quel hébergeur qui propose PHP et MySQL.

### Interface de l'énergimètre Puls0

L'interfaçage du Puls0 requiert quelques conditions préalables :

- quatre entrées S0 ;
- une alimentation 5 V ;
- une connexion Ethernet ;
- une version installée de l'EmonCMS d'OpenEnergyMonitor ;
- un témoin visuel des impulsions reçues.

La **figure 1** schématise le circuit de l'interface. Avant tout, disons que tous les composants sont à fils, pas de tracas pour les bricoleurs aux gros doigts et mauvais yeux, une catégorie dont je fais partie. Tous nos prototypes ont été manufacturés sur les platines d'un fabricant spécialisé.

L'interface Puls0 se compose des blocs suivants : une alimentation, une mémoire, un microcontrôleur, une interface S0 et une interface Ethernet.

**L'alimentation** autour d'IC3 convertit la tension d'entrée vers les 3,3 V pour la

mémoire, le microcontrôleur et l'interface Ethernet. Le plus gourmand d'entre eux, le circuit Ethernet, demande jusqu'à 500 mA. Le régulateur LM1086-3.3 effectue la conversion. Il réclame un radiateur.

La **mémoire** locale consiste en une EEPROM sérielle SPI de Microchip, IC4. Elle contient le logiciel local ainsi que les pages internet dont se sert l'interface utilisateur. Toutes ces pages ainsi que les graphiques sont dans une EEPROM de 1 024 Kb. Elle serait un peu étriquée pour une décoration animée, mais nous nous contenterons d'un modèle décent, intelligemment conçu en HTML, JavaScript et CSS.

Le **microcontrôleur** est un PIC24HJ128GP202 à 16 bits, 40 MIPS de Microchip (IC2). Il possède 128 Ko de mémoire flash et 8 Ko de RAM. Sa vitesse suffira largement à assurer la fluidité du serveur web ainsi que les fonctions réseau.

L'**interface S0** autour de OK2A/B et OK1A/B exige une plus haute tension d'environ 24 V qui lui est fournie par un convertisseur élévateur cc/cc (IC6). Ce convertisseur va puiser son énergie directement sur l'adaptateur de l'interface. Le convertisseur est le très répandu MC34063. Un photocoupleur isole du microcontrôleur chaque port de l'interface S0.

L'**interface Ethernet** se base sur un contrôleur ENC28J60 de Microchip (IC5). On lui adjoint un connecteur Ethernet à magnétique intégrée. Il est crucial de s'en tenir au modèle indiqué (X2) parce que le ENC28J60 est très ombrageux à propos du connecteur utilisé. N'importe quel connecteur à magnétique ne fera pas l'affaire. En outre, certains connecteurs disponibles sur le marché ne sont pas compatibles broche à broche.

### Assemblage et essais

Le tracé de la platine pour l'interface Puls0 est à la **figure 2**. Aucune consigne particulière pour assembler ce circuit. Il est cependant plus facile de souder les différents blocs fonctionnels par étape, en commençant par l'alimentation. Vérifiez ensuite que les puces dans chaque module reçoivent la tension adéquate et qu'il n'y a de court-circuit nulle part.

Accordez une attention particulière au régulateur de tension et à son radiateur. Leur assemblage demande une certaine précision pour éviter le court-circuit.

Une fois la tension vérifiée sur chaque puce, assurez-vous que le contrôleur Ethernet reçoit également la tension qui lui convient sur Vcap (broche 14). La bonne valeur se situe aux alentours de 2,5 V. Si c'est le cas, branchez un câble Ethernet raccordé à un réseau fonctionnel en passant par un commutateur réseau (*switch*). La LED sur le connecteur devrait commencer à clignoter. Les ultimes vérifications faites, votre interface est prête à recevoir son logiciel.

### Programmer l'interface Puls0

Sans logiciel, l'interface Puls0 ne sert pas à grand-chose. On la programme en deux étapes. Téléchargeons d'abord le micrologiciel du contrôleur, puis nous placerons les pages internet dans l'EEPROM. Vous trouverez gratuitement tous les composants logiciels pour ce projet et les données *Eagle board design* sur le site d'Elektor [2].

Vous pouvez programmer le microcontrôleur PIC24HJ128GP202 avec un PICKIT3 de Microchip, par exemple. N'oubliez pas d'alimenter le circuit via une source externe pendant la programmation.

L'interface S0 a besoin de trouver ses pages web de configuration à l'intérieur de l'EEPROM de 1024 Kb pour fonctionner correctement. Le micrologiciel de l'interface énérgimètre est capable de les télécharger de façon autonome, grâce à ses fonctions réseau internes. Pour enclencher la procédure, vous vous rendez à l'adresse : <http://192.168.X.X/mpfsupload>.

## Utilisation de l'interface énérgimètre Puls0

Lors du premier démarrage, branchez la sortie S0 de l'énérgimètre à l'entrée S0-1 de l'interface Puls0 comme illustré à la **figure 3**. Un simple câble à deux fils suffira puisqu'il ne conduit qu'une tension de 24 V<sub>CC</sub>. L'interface S0 travaille avec des câbles dont la longueur peut aller jusqu'à 20 m.

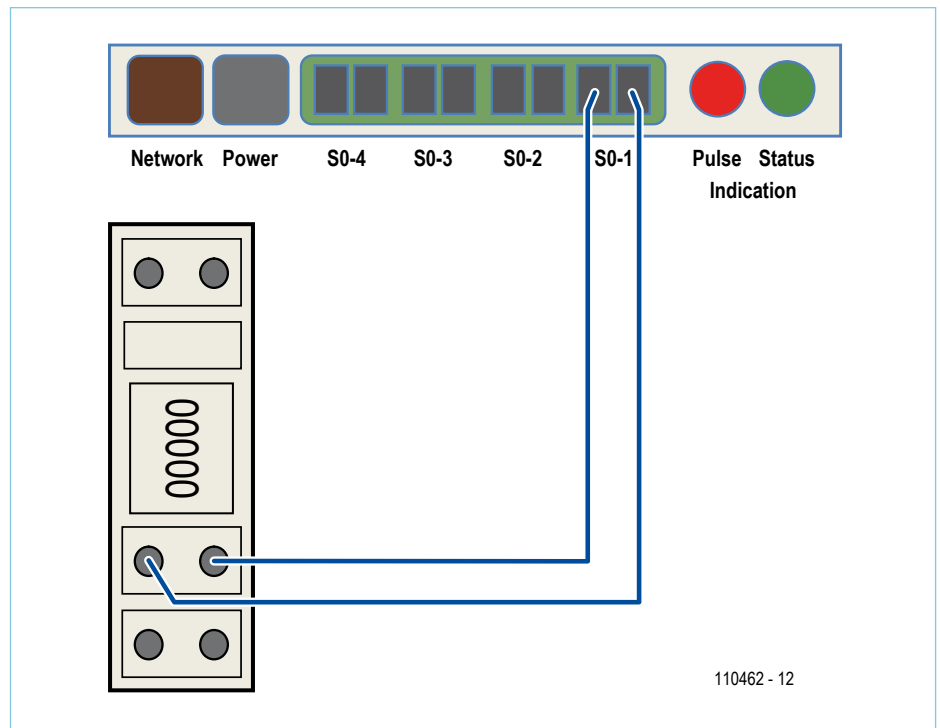


Figure 3. Un simple câble à deux fils suffit pour connecter l'interface à l'énérgimètre S0.

Branchez ensuite le port Ethernet de l'interface Puls0 au reste du réseau, via un *switch* ou un routeur. Puis, raccordez l'interface Puls0 à son alimentation de 5 volts. La LED verte située à l'arrière du boîtier devrait commencer à clignoter. Chaque fois que l'interface détecte une impulsion, la LED rouge s'allume brièvement. Si ce n'est pas le cas, essayez d'inverser la polarité du câble S0. En effet, il est possible qu'une diode à la sortie de l'interface S0 bloque la transmission des impulsions.

## Localiser l'interface internet

La configuration du Puls0 se réalise entièrement par le biais de son interface internet. Reste à connaître l'adresse IP derrière laquelle se trouve cette interface. Pour la dénicher, il existe deux techniques : utiliser un scanner d'IP qui balaye le réseau systématiquement ; repérer la bonne adresse dans la liste d'attributions DHCP de votre routeur.

## Configuration de l'interface Puls0

Dès que vous avez atteint l'interface de configuration du Puls0, il ne vous reste plus qu'à configurer le serveur EmonCMS openEnergyMonitor, comme le montre la **figure 4**.

Le site internet [openenergymonitor](http://openenergymonitor.org) explique en détail tous les paramètres disponibles pour le système d'enregistrement de données de l'EmonCMS.

## Tant qu'on y est

Bien que ce projet vise spécifiquement la plateforme openEnergyMonitor, on pourrait modifier le micrologiciel pour utiliser d'autres services d'enregistrement de données en ligne. Quant aux spécificités électroniques, il semble difficile de s'en écarter beaucoup sans envisager de passer à des composants montés en surface.

(110462 – version française : Robert Grignard)

## Liens

- [1] [www.openenergymonitor.org](http://www.openenergymonitor.org)
- [2] [www.elektor.fr/110462](http://www.elektor.fr/110462)

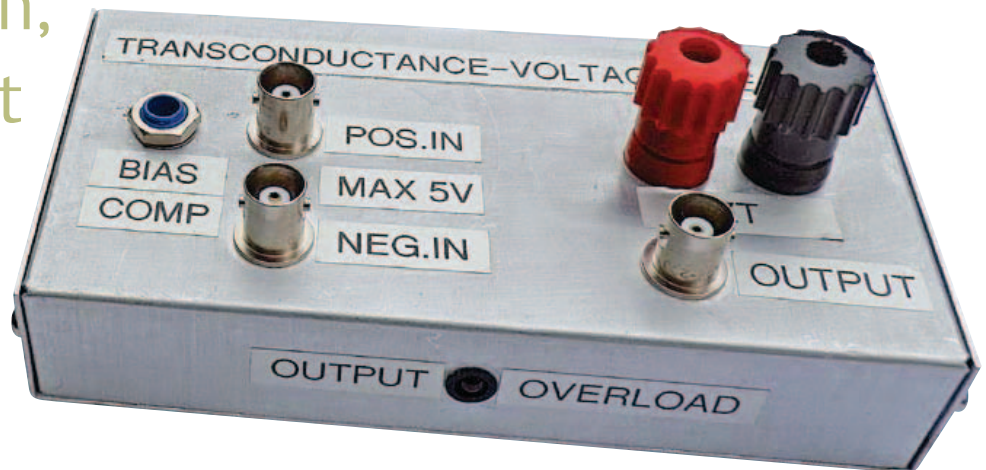


Figure 4. Configuration du serveur OpenEnergyMonitor.

# amplificateur de mesure à transconductance

Entrée en tension, sortie en courant

Un amplificateur à transconductance est un amplificateur différentiel particulier dont le courant de sortie est une fonction de la différence de tension entre les entrées. Le circuit intégré AD844 d'Analog Devices permet d'obtenir simplement ce comportement dans un circuit. Un tel amplificateur à transconductance constitue un circuit d'entrée polyvalent pour diverses applications de mesure, comme le montre cet article.



Wim de Jager (Pays-Bas)

L'amplificateur opérationnel AD844 est conçu initialement pour des applications avec rétro-action en courant, mais sa structure particulière offre d'autres possibilités en boucle ouverte dans le domaine des sources commandées. Ainsi, cet amplificateur opérationnel permet de réaliser simplement une source de courant commandée (transconductance) qui ouvre nombre de possibilités dans le domaine de la mesure. Cet article décrit le fonctionnement de l'AD844, la conception et la construction d'un préamplificateur avec ce circuit intégré, ainsi que divers exemples d'applications.

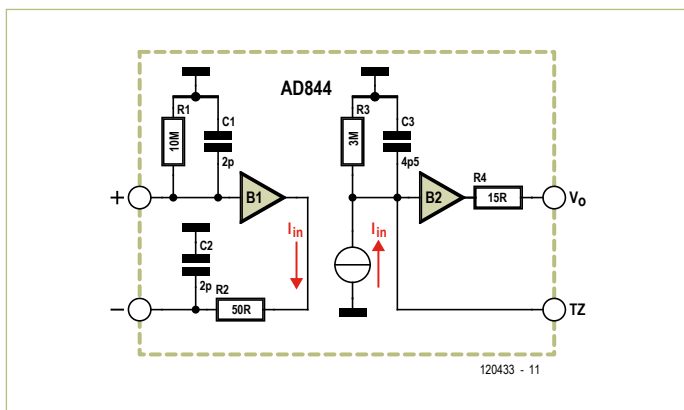


Figure 1 - Schéma de principe de l'AD844.

## Principe de l'AD844

La figure 1 représente le modèle simplifié de l'AD844. Le tampon B1 présente une entrée positive et une entrée négative. Le courant  $I_{in}$  imposé au circuit d'entrée est également disponible sur la borne TZ, la sortie du convertisseur U/I. Le courant  $I_{in}$  du préamplificateur est destiné à piloter des composants ou réseaux externes. La tension sur TZ est disponible à travers le tampon B2 sur la borne de sortie en tension. Les résistances R1 à R4 et les condensateurs C1 à C3 sont une image des écarts entre le comportement idéal et celui de l'amplification opérationnel. Les valeurs données proviennent de la documentation d'Analog Devices [1].

Si nous connectons une source de tension  $V_{in}$  aux entrées + et -, la pente de transfert  $I_{in}/V_{in} = 1/R2 = 20 \text{ mA/V}$ . Si la sortie TZ n'est pas chargée,  $A_o = R3/R2 = 60\ 000$ . La bande passante est fixée par  $R3//C3$ , soit approximativement  $f_{3dB} \approx 12 \text{ kHz}$ .

## Schéma du préamplificateur

Le schéma du préamplificateur (fig. 2) comporte un réglage de la compensation du courant de polarisation, un circuit écrêteur pour la protection de l'entrée positive (POS.IN) et un détecteur de surcharge pour la tension de sortie  $V_o$ . Dans toutes les applications de cet article, l'entrée négative (NEG.IN) est reliée à la masse. Un signal d'entrée sur POS.IN est transféré par R2 sur l'entrée + de l'AD844. La tension maximale admissible entre les bornes d'entrée est de  $5 V_{CC}$ , la protection est assurée par le circuit constitué par D1 à D4, R9 et R10. Les diodes D1 et D2 sont des 1N4148 au silicium. Les diodes Schottky ont été écartées à cause de leur courant de fuite relativement élevé, lequel aurait provoqué un décalage en continu indésirable. Si la résistance R2 est un modèle 1/4 W, la tension d'en-



trée maximale sera de  $20 V_{CC}$ . On peut éventuellement monter une résistance à couche métallique de  $1\text{ k}\Omega/2\text{ W}$  pour porter à  $50 V_{CC}$  la tension maximale admissible. Des valeurs supérieures de R2 (tout comme les résistances à fils) ont un effet néfaste sur le comportement en haute fréquence. Dans le cas de couplage par condensateur, le circuit écrêteur est très important pour éviter à l'AD844 les dégâts que provoquerait la pointe de courant lors de la charge du condensateur. La transconductance choisie est de  $1\text{ mA/V}$ , d'où la valeur de  $R3 = 1\text{ k}\Omega$ . La résistance RT4 compense l'effet de  $R_i$ . Avec une valeur de  $1\text{ k}\Omega$ , l'intensité est de  $5\text{ mA}$  sous  $5 V_{CC}$ . C'est le maximum selon les spécifications de l'AD844. Avec ces valeurs, l'utilisation de l'AD844 est optimale.

La résistance d'entrée est déterminée principalement par R1. Le courant de polarisation de l'entrée + de l'amplificateur opérationnel produit aux bornes de R1, quand la broche POS.IN est ouverte, une tension d'environ  $10\text{ mV}$ . Comme la tension d'offset de l'amplificateur opérationnel est très faible, il peut être utile pour des applications données (couplage en alternatif ou forte résistance de source) de compenser l'effet du courant de polarisation. Comme le sens du courant de polarisation peut changer du fait des tolérances dans les étages différentiels, le circuit de compensation autour de P1 offre une plage de réglage d'environ  $-23\text{ mV}$  à  $+23\text{ mV}$ . Les résistances R7 et R8 garantissent une faible résistance entre le curseur de P1 et la masse. Quand l'entrée POS.IN est attaquée par une source à faible impédance de sortie, le réglage est inutile. Ce réglage n'est pas destiné à compenser l'offset propre de l'AD844..

La borne TZ est la sortie de la source de courant commandée et en même temps l'entrée du tampon interne B2 de la figure 1. On peut connecter à TZ un composant ou réseau quelconque. Grâce à l'absence de rétro-action, il n'y a aucune limitation liée à la stabilité. L'impédance de sortie  $Z_{TZ}$  provoque une erreur de mesure, mais son influence est négligeable dans la plupart des applications. Le tampon B2 délivre la tension sur TZ,  $V_o$ .

Le réseau de diodes D5 à D10 constitue un détecteur de surcharge. La tension  $V_o$  est redressée par le pont D5 à D8 (les diodes Schottky ont été choisies pour leur faible tension de seuil et leurs bonnes caractéristiques en haute fréquence) et ce signal est appliqué à l'assemblage en série de la diode zener D9 et de la LED D10. Un faible éclairage de la LED signale une distorsion d'environ  $0,5\%$  due au dépassement de l'excursion maximale en sortie ; un éclairage fort signale un blocage du signal. Les filtres R11/C2 et R12/C3 sont importants pour la stabilité de l'AD844. Ils sont recommandés par Analog Devices.

### Alimentation

L'alimentation (figure 3) passe par un bloc secteur. La tension secondaire est de  $12\text{ V}_{CA}$  sous  $500\text{ mA}$ . À faible charge, comme dans notre cas, la tension secondaire monte à quelque  $15\text{ V}_{CA}$ , ce qui suffit pour cette application. Pour produire une tension symétrique de  $\pm 15\text{ V}$  avec un seul enroulement secondaire, on applique un redressement mono-alternance. Il n'y a pas d'inconvénient avec un condensateur de lissage relativement gros. Les régulateurs de tension IC1 et IC2 fournissent des tensions d'alimentation stables et se protègent contre les surcharges et les courts-circuits (environ  $100\text{ mA}$ ).

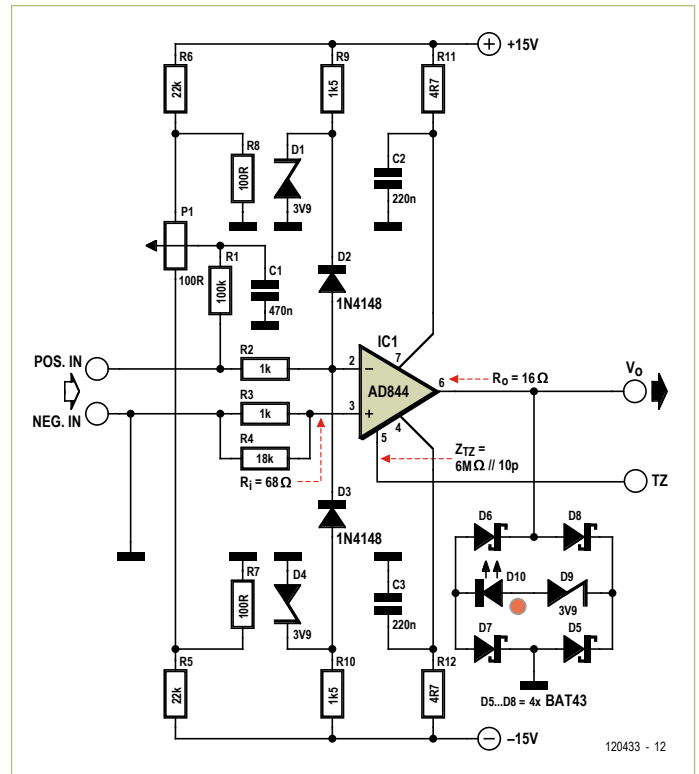


Figure 2 - Schéma complet de l'amplificateur de mesure à AD844.

### Réalisation

Pour la construction, voir figure 4, on utilise un coffret fermé en aluminium avec une plaque d'isolant cuivré. La couche de cuivre étamé est utilisée comme plan de masse pour souder les connexions de masse. Pour les entrées et sorties, le choix s'est porté sur des prises BNC, sur des douilles banane pour le raccordement de TZ.

Le support DIL de l'AD844 est fixé sur un petit circuit imprimé qu'on peut aussi réaliser soi-même. Les diodes D5 à D8 sont montées entre les broches du support. Le potentiomètre P1 est un modèle

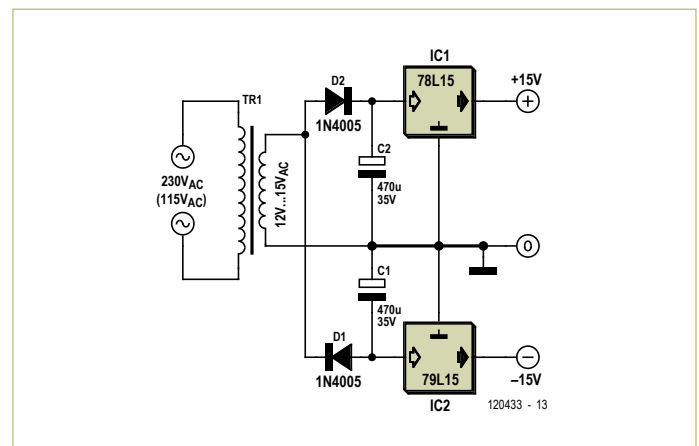


Figure 3 - L'alimentation adaptée fait appel à un bloc secteur. Le redressement mono-alternance donne deux tensions symétriques à partir d'un unique secondaire.

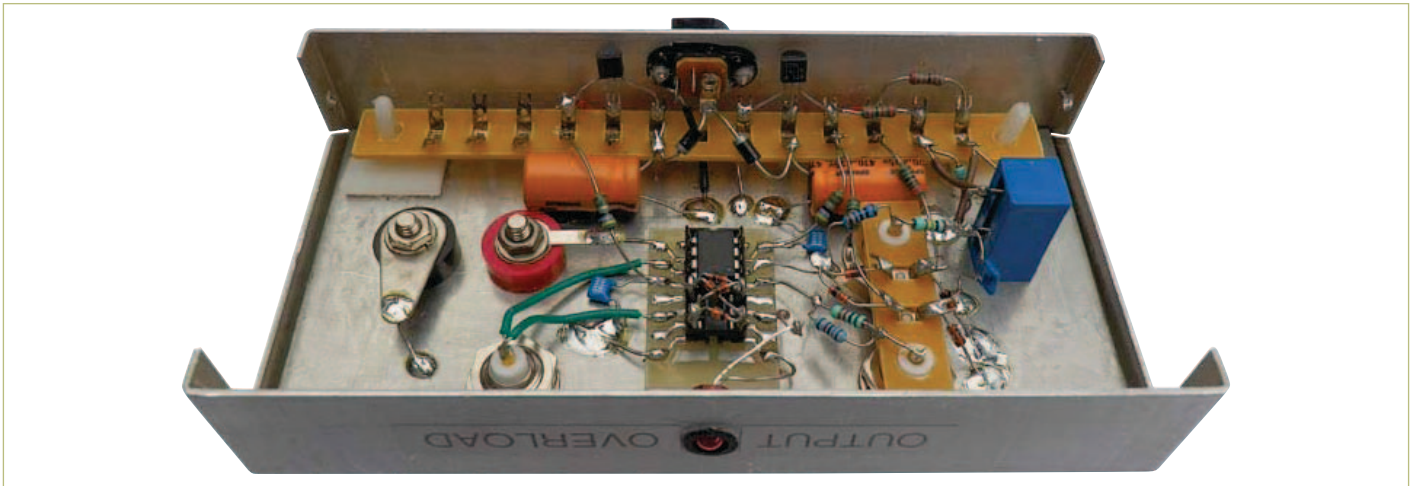


Figure 4 - Les entrailles du prototype de l'auteur.

cermet à 15 tours, ce qui permet un réglage précis de la compensation du courant de polarisation.

Les transistors bipolaires complémentaires de l'AD844 ont une fréquence de coupure élevée, ce qui rend nécessaire un agencement soigné pour éviter les oscillations parasites. Le plan de masse combiné à des connexions courtes est la meilleure garantie de bonnes caractéristiques et de la stabilité en HF.

### Mesures sur le préamplificateur

L'effet de l'offset interne de l'AD844 se mesure avec un micro-ampèremètre connecté à TZ, avec les bornes d'entrée court-circuitées. La valeur est de 0,1  $\mu$ A, elle est très basse en partie grâce au calibrage initial au laser. Un réglage d'offset facultatif, décrit dans la documentation de l'AD844, permet de réduire encore cette valeur. Avec l'entrée POS.IN ouverte, le courant de polarisation produit aux bornes de R1 une tension de 10 mV, avec comme conséquence un courant de 10  $\mu$ A à la sortie TZ. Cette valeur est très supérieure à celle de l'offset et elle justifie la présence du circuit de compensation

décrit plus haut. Il permet de ramener l'offset à moins de 0,1  $\mu$ A. Le comportement en alternatif sans charge à la sortie TZ est mesuré avec un signal d'entrée de 0,5 mV à 500 Hz. Le gain est de 6000. Si  $1 \text{ mA/V} \times R_{TZ} = 6000$ ,  $R_{TZ} = 6 \text{ M}\Omega$ . La fréquence de coupure (-3 dB) se situe à 2,5 kHz. Puisque  $\omega \times R_{TZ} \times C_{TZ} = 1$ ,  $C_{TZ} = 10 \text{ pF}$ . La valeur mesurée de  $R_i$  est de 68  $\Omega$ . Pour  $R_i$ , nous trouvons 16  $\Omega$ .

### Mesures sur les condensateurs et inductances

Connectez un condensateur sur TZ et un générateur sinusoïdal sur POS.IN, réglez l'amplitude d'entrée à 500 mV et réglez la fréquence (de préférence de haut en bas pour éviter des saturations inutiles) jusqu'à une valeur telle que la tension sur TZ soit également de 500 mV. À cette fréquence, le gain est 1 et la valeur de C peut se calculer suivant  $R = 1/\omega C$ , ou encore  $C = 1/\omega R$ .

Exemple : avec un condensateur donné, le gain unitaire est obtenu à 16 kHz. Comme  $R = R3 = 1 \text{ k}\Omega$ , le calcul donne  $C = 10 \text{ nF}$ . Un LCR-mètre *Voltcraft* donne une valeur de 10,47 nF.

Pour la mesure des inductances, une tension du générateur de

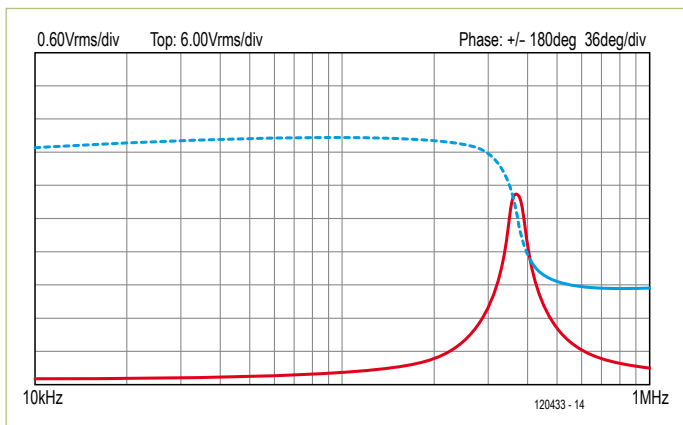


Figure 5 - Mesure de l'impédance d'une bobine à l'aide d'un oscilloscope USB avec générateur de fonctions incorporé. Les 0,6 V/div correspondent à 6 k $\Omega$ /div, le maximum de 6 V correspond à 60 k $\Omega$ .

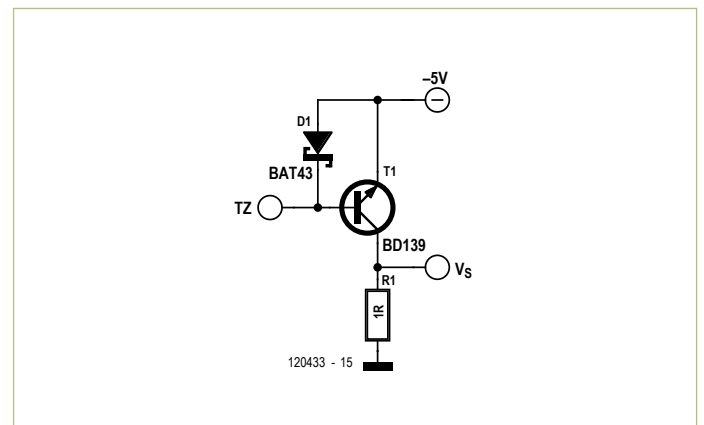


Figure 6 - Schéma du dispositif de mesure des transistors NPN.

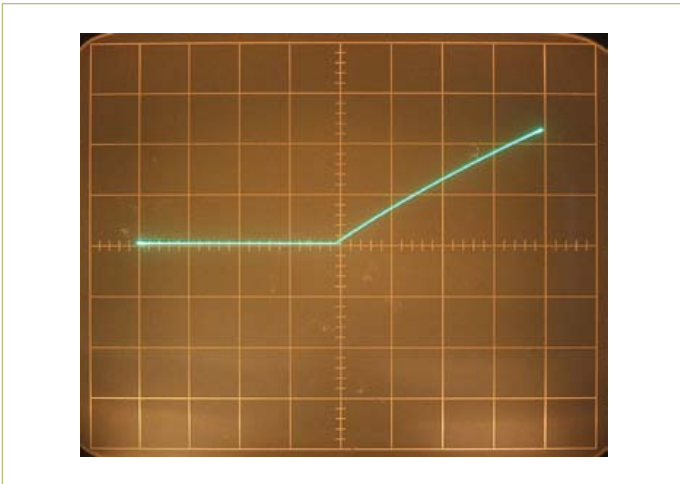


Figure 7 - Cette courbe s'affiche à l'écran de l'oscilloscope lors de la mesure du circuit de la figure 6. Horizontal : 1 mA/div ; vertical : 0,2 A/div.

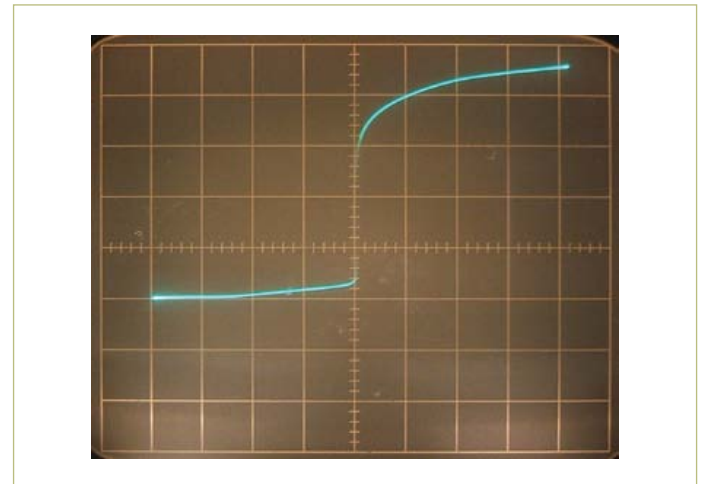


Figure 8 - Courbe relevée lors de la mesure d'une diode zener. Horizontal : 0,5 mA/div ; vertical : 1 V/div.

500 mV est recommandée aussi. Dans ce cas, on évite les saturations inutiles en faisant varier la fréquence de bas en haut. La mesure d'une bobine sur un noyau torique donne un gain unitaire pour 56 kHz. Comme  $\omega L = R$  et  $L = R/\omega$ ,  $L = 2,84$  mH. Le LCR-mètre affirme 2,86 mH.

Il est possible aussi de calculer la capacité  $C_p$  du bobinage à partir de la fréquence à laquelle se produit l'auto-oscillation. Pour cette bobine, la fréquence est de 360 kHz. Suivant  $C_p = 1/\omega_0^2 L$ , la capacité calculée est 69 pF. Pour obtenir une valeur correcte, il faut soustraire la capacité de sortie  $C_{TZ} = 10$  pF, la capacité réelle  $C_p$  est donc 59 pF. Lors de la résonance propre, l'impédance est égale à la résistance de perte en parallèle, laquelle peut être déterminée par la mesure du gain. La valeur du gain ressort à 38, d'où  $R_p = 38 \times R_3 = 38 \times 1$  k $\Omega = 38$  k $\Omega$ .

### Mesure d'impédance avec le Velleman PCSGU250

L'oscilloscope USB PCSGU250 de *Velleman* comporte un générateur de fonction intégré. Combiné avec le logiciel livré, il peut être utilisé comme analyseur de circuit. Ce dispositif permet de tracer des diagrammes de Bode qui représentent l'amplitude et la phase de la fonction de transfert d'un circuit électronique, en fonction de la fréquence. Le convertisseur U/I du préamplificateur permet de mesurer l'amplitude et la phase d'une impédance. Ce dispositif de mesure, parce qu'il mesure aussi la phase, s'appelle impédancemètre vectoriel. La **figure 5** représente la caractéristique d'amplitude et de phase de la bobine utilisée précédemment. L'angle de phase est d'environ 90° jusque vers 200 kHz, ce qui n'est pas étonnant pour une bobine. La résonance propre se manifeste à 370 kHz, ce qui permet de déterminer la résistance de perte en parallèle. Puisque l'amplitude du générateur est de 100 mV, la graduation de l'axe d'amplitude (Y) est de 6 k $\Omega$ /V. La valeur de  $R_p$  mesurée ici est 34 k $\Omega$ .

### Mesures sur les semi-conducteurs

Le pilotage en courant est utilisable aussi pour relever les caractéristiques de semi-conducteurs tels que les transistors bipolaires. Un dispositif de mesure pour un transistor NPN est proposé par la **figure 6**. Le fait d'alimenter l'émetteur sous -5 V permet de référencer la résistance de mesure R1 à la masse. La tension aux bornes de cette résistance (1  $\Omega$ ) est une image du courant de collecteur.

La borne TZ du préamplificateur est connectée à la base. L'attaque de la base est de 1 mA/V. La diode Schottky D1 protège T1 contre les tensions inverses excessives. L'oscilloscope affiche la courbe  $\beta$ , voir la figure 7, avec la tension sur R1 en ordonnée et la tension sur POS.IN en abscisse. La sensibilité verticale est fixée à 0,2 V/div, ce qui correspond au 0,2 A/div du courant de collecteur. La sensibilité horizontale est de 1 V/div, pour 1 A/div de courant de base. Comme le signal sur Vs est négatif, on choisit le mode *inverse* (disponible sur certains oscilloscopes) pour afficher un résultat positif. On peut déduire de la courbe que pour  $I_c = 400$  mA le gain en courant  $\beta$  vaut 118. Un deuxième exemple, **figure 8**, s'applique à une diode zener. L'échelle horizontale est de 0,5 V/div, ce qui correspond à 0,5 mA/div pour le courant à travers la zener. L'échelle verticale est de 1 V/div, la tension aux bornes de la zener. Pour un courant de 2 mA, la tension de zener est égale à 3,6 V, conformément aux données du fabricant.

### Préamplificateur

Un préamplificateur à couplage en continu est facile à réaliser avec une résistance de charge  $R_L$  connectée à ZT. Exemple : avec  $R_L = 10$  k $\Omega$ , le gain est de  $R_L/R_3 = 10$ . La bande passante se calcule selon  $R_L/C_{TZ}$  et vaut 1,6 MHz. Les mesures confirment ces valeurs. Le préamplificateur peut être utilisé par exemple pour augmenter la sensibilité d'un appareil de mesure.

### Tampon

Le tampon de sortie peut aussi être utilisé séparément comme suiveur de tension, avec l'entrée sur TZ et POS.IN à la masse. La bande passante est d'environ 10 MHz. L'impédance d'entrée est égale à  $Z_{TZ}$ , soit 6 M $\Omega$ //10 pF et la résistance de sortie  $R_o = 16$   $\Omega$ . La tension de sortie maximale est de 5  $V_{eff}$  avec un THD de 0,7% (1 kHz). Le courant de sortie maximal est de 80 mA $_{CC}$  et le circuit est protégé contre les courts-circuits.

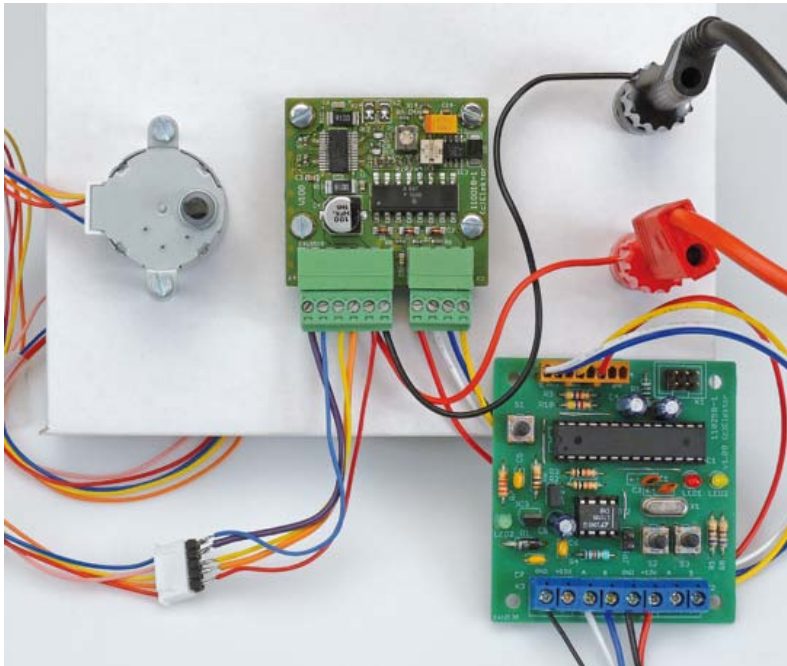
(120433 – version française : Jean-Paul Brodier)

### Liens

[1] [www.analog.com](http://www.analog.com)



# bibliothèque C pour ElektorBus



## accélérer l'accès à votre propre micrologiciel

Il y a juste un an, nous présentions un logiciel qui vous permet de réaliser sur notre ElektorBus, dare-dare et sans souci, la commande par PC d'une application personnelle. Élargissons maintenant cette méthodologie du *vite fait, bien fait* au

micrologiciel des nœuds à microcontrôleur. Grâce à cette bibliothèque C pour ElektorBus, il n'y a plus à s'emm... brouiller avec les protocoles de bus et ses bits. En plus, la couche dédiée de matériel ouvre la voie au libre choix du contrôleur... et d'intéressantes perspectives par-delà le bus lui-même.

### Jens Nickel (Elektor)

Pour des projets complexes, arrivé au stade du logiciel, quel plaisir d'en trouver un qui fait tout simplement ce qu'on lui demande ! C'était déjà le cas avec ElektorBus. Dans la série d'articles [1] qui lui ont été consacrés, nous avons dû commencer par développer les protocoles, dont nous n'avons chargé dans le contrôleur que la partie du micrologiciel indispensable aux tests des nouvelles fonctions.

Avec le temps, les exigences ont gonflé. Pour différents types d'appareils reliés au bus, comme la platine d'installation et les nœuds expérimentaux avec photocellule ou relais, on a dû chaque fois développer un nouveau logiciel. Bien que la structure de base reste la même, il fallait laborieusement chercher le bout de code à adapter, pour peut-être expédier une valeur numérique au lieu de la rece-

voir. Il fallait une étude approfondie des spécifications du protocole pour savoir, au moment de fixer définitivement le code des octets d'un message, comment exciter un relais ou le laisser retomber.

### Nombreux avantages

C'est précisément dans de tels cas qu'une bibliothèque peut se révéler une aide précieuse. Elle masque la structure de base invariable du micrologiciel derrière quelques appels de fonctions, si bien que le code de l'application proprement dite reste court et limpide. Elle calcule automatiquement le nombre d'octets nécessaires qui devront être envoyés sur le bus, de sorte que le concepteur ne doive plus s'occuper un à un des bits du protocole. Et finalement, elle rend le code de l'application indépendant du matériel, donc autant des cartes de nœuds sur le bus que des microcontrôleurs utilisés. Encore et toujours, nous recevons des demandes visant à assimiler d'autres contrôleurs, comme le PIC. Le langage C, par bon-

## Produits et services Elektor

- Noeud expérimental (circuit imprimé 110258-1 ou jeu de trois circuits imprimés 110258-1C3)
- Convertisseur USB/RS485 (monté et essayé 110258-91)
- Pilote de moteur pas à pas (circuit imprimé 110018-1)

- Logiciels gratuits à télécharger

Vous pouvez commander les produits ou charger les fichiers gratuits à l'adresse : [www.elektor.fr/120582](http://www.elektor.fr/120582)



heur, dispose d'une syntaxe normalisée et le compilateur accepte les différents types de puce. On peut ainsi rédiger du code d'application (**fig. 1**) vraiment indépendant de la plateforme et portable d'une platine à l'autre.

## Appelons la bibliothèque

Assez de théorie ! Intéressons-nous aux lignes de code suivantes :

```
TransmitValue(OwnAddress, MasterAddress, 1, 0,
TestLEDStatus);
TransmitValue(OwnAddress, MasterAddress, 2, 0,
SensorValueRaw);
```

Avec elles, nous avons composé un message ElektorBus qui transporte deux valeurs numériques (ce genre d'unités d'information s'appellent *Parts*) sur deux canaux différents, l'un représente l'état de TestLED et l'autre, une valeur de mesure comprise entre -1 023 et +1 023.

Au moyen de :

```
SendParts();
```

nous envoyons ces *Parts* ensemble dans un message sur le bus. Sans nous soucier de la place que doivent occuper les bits dans les octets de notre message, qui en compte 16. Si vous avez reconnu cet appel de fonction, c'est que vous avez suivi attentivement notre série sur l'ElektorBus. En novembre 2011, nous avons présenté une bibliothèque Javascript utilisable lors du développement d'une commande de bus personnelle [2]. À l'époque, nous avons délibérément opté pour une interface de bus utilisateur basée sur HTML et Javascript et donc utilisable indifféremment sur PC comme sur tout appareil Android.

Au fur et à mesure, je me suis orienté, pour les fonctions de la nouvelle bibliothèque, sur celles contenues dans la bibliothèque Javascript orientée JSBus. Pour certaines d'entre elles, je dois la réussite au fait que Javascript utilise une syntaxe comparable à celle du C. Mais j'ai dû tenir compte de l'exigüité de la mémoire sur le microcontrôleur. Par exemple, la bibliothèque C ne dispose pas de la possibilité de rassembler des *Parts* pour différents destinataires et qui seraient codées automatiquement dans plusieurs messages. Le message à envoyer ne peut pas contenir plus de quatre *Parts*, que la bibliothèque stocke dans une mémoire tampon d'émission sous forme de tableau.

## Maintenant, c'est elle qui nous rappelle !

Il y a une mémoire similaire pour les *Parts* reçues, elle est remplie par la bibliothèque si le message reçu est destiné à un nœud. Ensuite, la bibliothèque appelle pour chaque *Part* la fonction :

```
void ProcessPart(struct Part part)
{
...
}
```

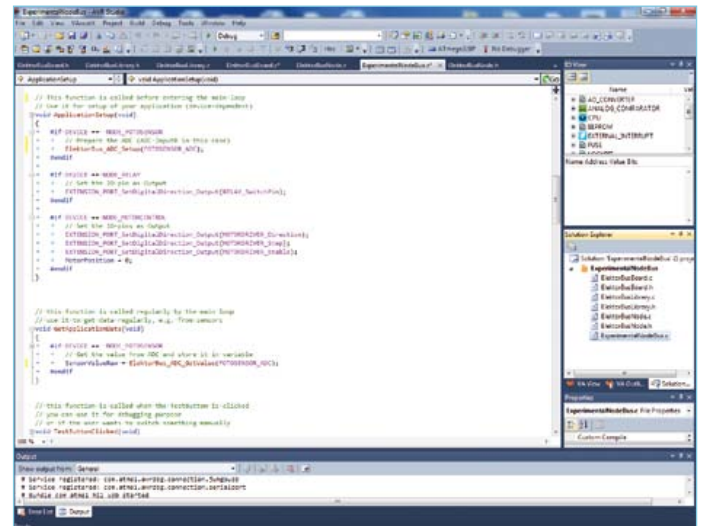


Figure 1. Dorénavant, toute application de l'ElektorBus pourra être développée indépendamment de la carte de nœud et du microcontrôleur employé (ici en AVR Studio 5).

Celle-ci doit nécessairement avoir été définie dans le fichier principal du code source du bus et est maintenant réservée totalement à la fonction principale et à l'application elle-même. Le concepteur peut alors remplir la fonction `ProcessPart(...)` de son code propre qui réagit à des *Parts* reçues, lesquelles peuvent être des consignes pour actuateur ou toute autre instruction du maître du bus. Le concepteur a accès à toutes les propriétés des *Parts*, par exemple une valeur numérique transférée, au moyen d'une expression telle que `part.Numvalue` (cf. l'encadré *Parts*).

Un exemple d'un tel code d'application serait :

```
void ProcessPart(struct Part part)
{
if ((part.Channel == 0) && (part.Parttype ==
PARTTYPE_VALUE2))
{
SwitchTestLED(part.Numvalue);
SwitchRelay(part.Numvalue);
}
}
```

Selon la valeur reçue, qui ici ne peut être que 0 ou 1, la TestLED sur la platine sera allumée ou éteinte. On peut aussi activer ou relâcher un relais. La différence entre les fonctions `SwitchTestLED(...)` et `SwitchRelay(...)` est que la première est fournie par la bibliothèque (toute carte ElektorBus dispose d'une LED de test). La seconde, en revanche, est spécifique de l'application et le concepteur devra la rédiger lui-même (nous y reviendrons).

**Listage 1. La boucle principale de chaque micrologiciel de l'ElektorBus.**

```
int main(void)
{
    ElektorBusLibrarySetup();
    ApplicationSetup();

    while(1)
    {
        GetApplicationData();
        ElektorBusEngine();
    }
}
```

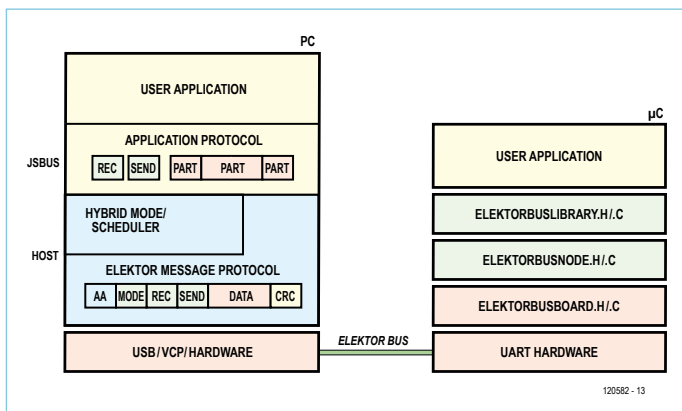


Figure 2. Du logiciel pour la *Rapid Application Development*. En même temps, l'ElektorBus-C-Library est compétente pour deux couches de protocole, message et application. S'y ajoute une couche de matériel.

Il y a encore d'autres fonctions, selon les circonstances, que l'on peut appeler de la bibliothèque. Pour chacune d'elles, il faut donner une définition dans le fichier principal ou dans un autre fichier du projet qui contient la bibliothèque C de l'ElektorBus. Vous pouvez remplir le corps de la fonction avec votre propre code, mais ce n'est pas nécessaire.

La fonction suivante est toujours appelée quand on pousse sur le bouton de test :

```
void TestButtonClicked(void)
{
    ToggleTestLED();
}
```

Dans cet exemple, nous faisons basculer chaque fois la TestLED ; une fonction correspondante est aussi disponible dans la bibliothèque. Quelques lignes de code suffisent à réaliser une possibilité de déboguer simplement pour vérifier au moins si le logiciel du nœud est actif.

L'encadré *Fonctions principales* vous donne un aperçu des fonctions implémentées jusqu'ici dans la bibliothèque.

**Machinerie du bus**

Rien n'a changé dans le processus de base du micrologiciel. D'abord, on initialise le matériel et les paramètres du nœud, l'adresse par exemple. Puis on lance une boucle sans fin dans laquelle le nœud a d'abord l'occasion d'accomplir des tâches spécifiques de l'application, comme de lire les valeurs des capteurs sur le convertisseur A/N. Puis la machine du bus (*engine*) intervient. Elle répond d'abord aux messages entrants, y compris ceux du superviseur du bus. Si le nœud est supervisé (*scheduled*), il a pour instruction de transmettre et si le superviseur l'y autorise, le *SendFlag* est levé pour qu'il envoie immédiatement un message. Autrement, le nœud ne l'enverra que comme participant au *FreeBus* dans la phase de *FreeBus* et seulement s'il a quelque chose à envoyer (*SendEventFlag == TRUE*). L'algorithme qui attend la confirmation du message par le destinataire et, le cas échéant, répète l'envoi (parce qu'en phase *FreeBus*, les collisions sont possibles), est lui aussi resté le même (pour le concept, voyez sur [3]).

Tout cela est désormais encapsulé dans la bibliothèque, le code n'est pas important pour le concepteur d'application. Tout ce qu'il doit savoir, c'est que le *SendEventFlag* est levé quand il appelle la fonction *SendParts()* ; alors le message sera envoyé immédiatement après le début de la prochaine *FreeBusPhase*. Pour un nœud supervisé, destiné à être interrogé régulièrement, un autre mécanisme est en vigueur. Quand le message du superviseur arrive pour dire qu'il est temps d'émettre, il appelle d'abord de la bibliothèque la fonction

```
void PreparePartsForScheduledMessage()
{
    ...
}
```

dont la définition doit également figurer dans le code de l'application. Alors, le nœud peut assembler les *Parts* appropriées (comme des valeurs d'un capteur) en temps réel :

```
void PreparePartsForScheduledMessage()
{
    TransmitValue(OwnAddress, MasterAddress, 0, 0,
    SensorValueRaw);
}
```

Ici, un appel de *SendParts()* est inutile.

Puisque la bibliothèque contient la partie récurrente du micrologi-



ciel, la boucle principale en est allégée d'autant, comme vous pouvez le voir dans le **listage 1**. Les noms des appels de fonction parlent d'eux-mêmes. Les fonctions `ApplicationSetup()` et `GetApplicationData()`, c'est au concepteur de les mettre en œuvre. On pourrait par exemple envisager l'initialisation d'un convertisseur A/N et la lecture régulière des valeurs calculées.

## Fonctions matérielles

Pour initialiser le convertisseur A/N (ADC), on écrit :

```
void ApplicationSetup()
{
    ElektorBus_ADC_Setup(0);
}
```

La fonction `ElektorBus_ADC_Setup(0)` initialise le CAN (canal 0). Les instructions qui suivent sont spécifiques du contrôleur, certaines valeurs doivent s'écrire dans des registres déterminés du contrôleur. Ces fonctions sont classées dans une petite bibliothèque séparée, dans les fichiers `ElektorBusBoard.h` et `ElektorBusBoard.c`. Le code source de la bibliothèque dédiée à l'ElektorBus se trouve lui dans les fichiers `ElektorBusLibrary.h` et `ElektorBusLibrary.c` (cf. **fig. 2** à ce sujet).

Pourquoi cette séparation ? Eh bien, si un lecteur décide de porter son application d'une plateforme sur une autre, avec un autre contrôleur ou d'un brochage différent, il peut emporter non seulement le fichier principal du code de l'application, mais aussi les fichiers de la bibliothèque `ElektorBusLibrary.h` et `.c` et ne devra refaire l'implémentation que des fonctions dans `ElektorBusBoard.c` (il peut même espérer qu'un autre concepteur l'ait déjà fait pour chacune des plateformes). La **directive #define** est une particularité très intéressante du C. On en trouve une cascade dans le fichier `ElektorBusBoard.h` et par exemple :

```
#define LED_PORT_OUT PORTD
#define TESTLED_PIN 4
```

Avant la compilation proprement dite du code source, un **préprocesseur** recherche dans le code des expressions comme `TESTLED_PIN` et remplace toutes les occurrences suivantes par le nombre 4. Si le concepteur prend la peine de toujours l'utiliser pour des expressions définies et abstraites, le code source de l'application de bus, de même que la bibliothèque du bus, pourront rester inchangés, quel que soit le numéro de la broche de port où raccorder la TestLED, dans cet exemple. Alors, quand on portera l'application sur une autre plateforme, il n'y aura plus qu'à adapter les directives `#define`.

## Spécifications de la carte

Une telle bibliothèque matérielle de bus n'a de sens que si l'on envisage de porter certaines fonctions dont chaque carte d'ElektorBus devra disposer. Après la TestLED déjà mentionnée et le bouton correspondant, je propose d'ajouter :

### Listage 2. Commutation d'un relais avec du code indépendant du matériel.

```
void SwitchRelay(unsigned char relayStatus)
{
    if (relayStatus == 0)
    {
        // Set pin to low
        EXTENSION_PORT_SetDigital_Low(EXTENSION_
DIGITAL0_PIN);
    }
    else
    {
        // Set pin to high
        EXTENSION_PORT_SetDigital_High(EXTENSION_
DIGITAL0_PIN);
    }
}
```

- jusqu'à 7 autres LED rassemblées sur un groupe de broches de port d'un registre accessible nommé `LED_PORT` (LED1, LED2, ...; LED0 étant la TestLED),
- jusqu'à 7 autres boutons à un `BUTTON_PORT` (Button1, Button2, ...; Button0 étant le TestButton),
- un groupe de broches comprenant les broches RX/TX ainsi que deux broches d'E/S pour les fonctions `DriverEnable` et `ReceiverDisable` de la liaison RS485 (`ELEKTORBUS_PORT`),
- un groupe de broches réservées pour y raccorder de futurs développements matériels, par exemple sur un connecteur (`EXTENSION_PORT`). Ce port pourrait accueillir jusqu'à huit E/S numériques ou jusqu'à huit entrées analogiques, au choix.

Pour toutes ces fonctions, des instructions indépendantes du matériel sont définies dans `ElektorBusBoard.h`. Dans le code de l'application, par exemple pour commuter un relais par la fonction spécifique de l'application `SwitchRelay()` il est préférable de l'utiliser, comme dans le **listage 2**.

Si vous y trouvez l'instruction

```
EXTENSION_PORT_SetDigital_Low(EXTENSION_DIGITAL0_PIN)
```

c'est ce qu'on appelle une **macro** qui est définie également dans `ElektorBusLibrary.h`. Elle provoque la mise au niveau bas de la broche d'E/S numérique 0 (et la chute du relais). Comme le relais n'est probablement pas relié systématiquement à la broche 0 du port d'extension (sur le nœud expérimental, c'est sur PC0), il faudrait encore écrire dans le fichier principal une directive `#define` :

```
#define RELAY_SwitchPin EXTENSION_DIGITAL0_PIN
```

On peut alors écrire :

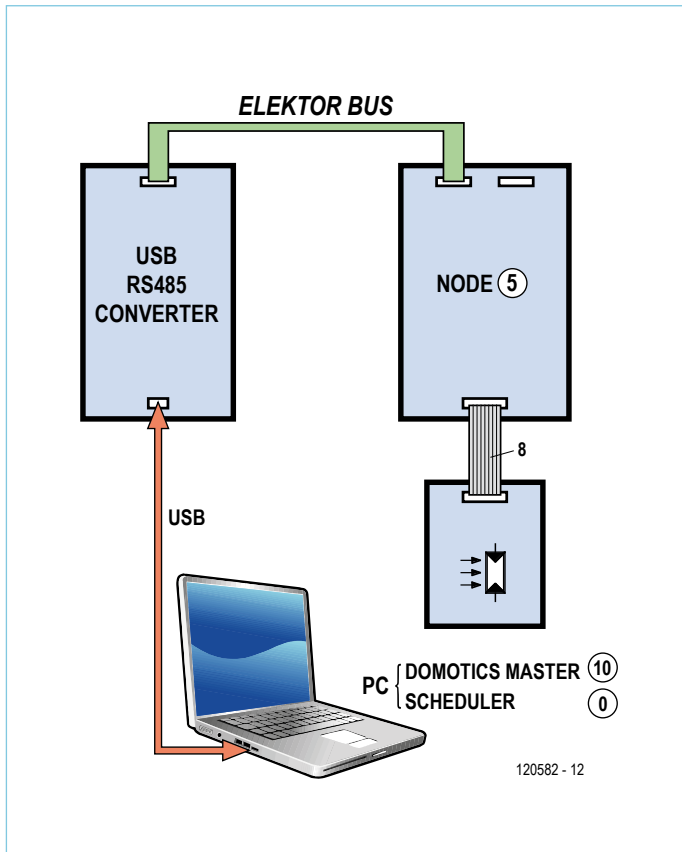


Figure 3. Pour réaliser notre exemple d'application, il suffit d'avoir un convertisseur USB/RS485 et un nœud expérimental équipé d'un ATmega328.

```
EXTENSION_PORT_SetDigital_Low(RELAY_SwitchPin);
```

Si le relais, lors d'un développement ultérieur, migre vers une autre broche de port, il n'y aura qu'à modifier en ce sens la directive #define ci-dessus.

### Fichier de noeud

Tout en haut du texte, nous avons pris la ligne de code :

```
TransmitValue(OwnAddress, MasterAddress, 1, 0, TestLEDStatus);
```

pour donner, avec la variable OwnAddress, l'adresse du nœud. La bibliothèque du nœud doit aussi connaître cette adresse pour pouvoir déterminer quels messages sont destinés au nœud. Lors du lancement du logiciel, la bibliothèque du bus peut obtenir les paramètres du nœud au moyen de fonctions qui sont reprises dans une autre paire de fichiers : ElektorBusNode.h et .c. À ce moment, l'adresse du nœud est fixée dans la fonction correspondante :

```
unsigned char GetNodeAddress()
{
    return 5;
}
```

Au fil des développements futurs de ce projet, nous ajouterons certainement l'adressage dynamique des nœuds.

### Exemple de logiciel

Tradition oblige, cet article sur le bus propose évidemment un logiciel d'exemple [4]. Le projet pour AVR Studio 5 reprend tous les fichiers cités ci-dessus, en plus d'un exemple d'application. À vrai

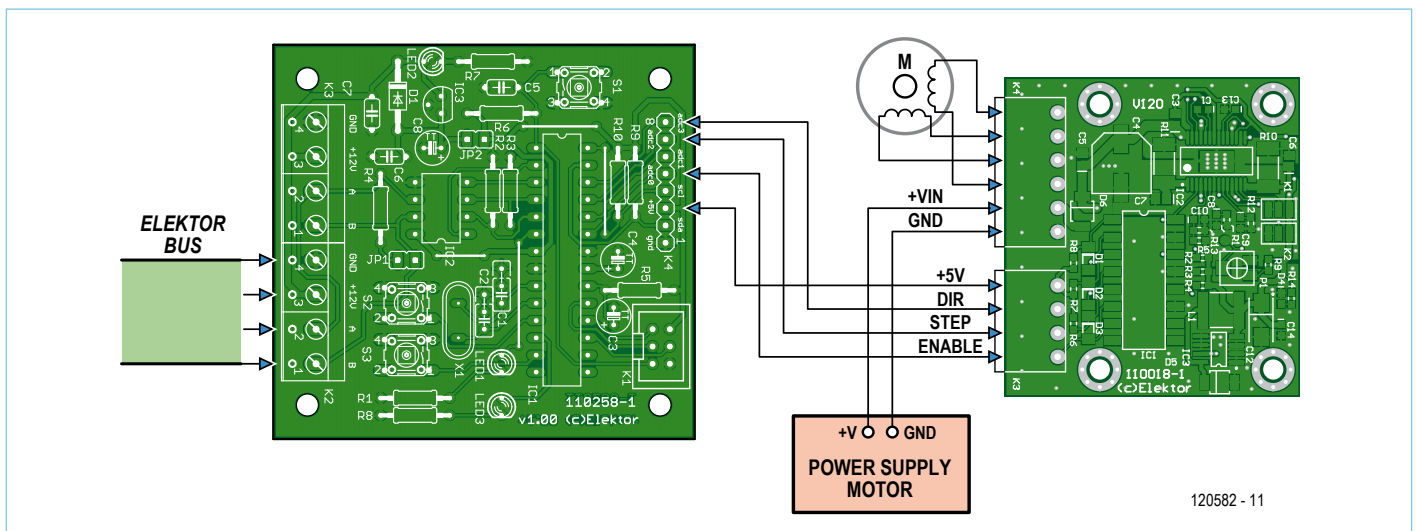
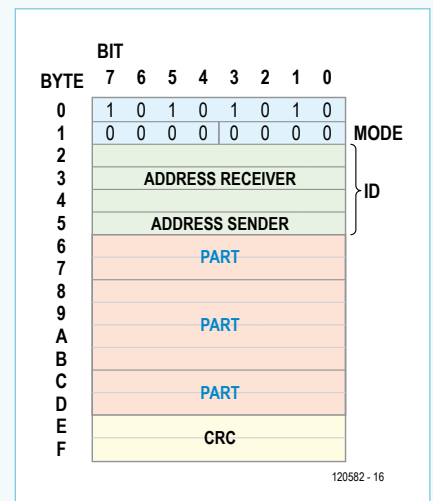


Figure 4. Pour le troisième exemple d'application, on relie un nœud expérimental à la platine de moteur pas à pas d'Elektor de juin 2012. Nous avons utilisé un petit moteur pas à pas de Ming Jong [9].

## Parts

Dans les huit octets de données d'un message ElektorBus, on peut transmettre, selon le protocole d'application, jusqu'à quatre *Parts* (unités d'information), p. ex. des valeurs de mesure ou de consigne, des informations de limite ou des commandes pour changer d'unité ou d'échelle. Dans l'ElektorBus-C-Library, une *Part* est représentée par une structure de données et identifiée par le mot-clé `struct`. Pour chaque *Part* reçue, la bibliothèque appelle la fonction `void ProcessPart(struct Part part)` et lui passe une semblable structure appelée `part`. Le concepteur de l'application a accès au moyen d'une instruction comme `part.Numvalue` à toutes les propriétés des *Parts* reçues :

Valid	code de contrôle ok ? (pas encore implémenté)
Sender	adresse de l'émetteur
Receiver	adresse du récepteur
Channel	numéro de canal
Setflag	consigne ou valeur de mesure ?
Ackflag	AcknowledgeMessage ou message original (drapeau dans une application)
Mode	octet de mode du message (avec Acknowledge-Flag dans un message)
Parttype	type de <i>Part</i> , selon des constantes définies : <code>PARTTYPE_VALUE2</code> , <code>PARTTYPE_VALUE4</code> , <code>PARTTYPE_VALUEFLOAT</code> , <code>PARTTYPE_LIMIT</code> , <code>PARTTYPE_SCALE</code> , <code>PARTTYPE_INTERVAL</code> .
Numvalue	valeur numérique transmise (p.ex. -1023 à +1023 pour <code>PARTTYPE_VALUE2</code> )
Limit	0 = ok, 1 = en dessous de la consigne, 2 = dépassement de consigne
Quantity	grandeur physique (0 à 127, cf. [3])
Unit	unité (0 à 3, cf. [3])
Scale	puissance de dix pour l'échelle de grandeur (-15 à +15)
Interval	intervalle de l'unité d'échelle pour rapports de capteurs (cf. [3])
Preset	réservé



dire, il y en a trois que vous pouvez choisir par une directive `#define` placée tout en haut du fichier `ElektorBusNode.h` :

```
#define DEVICE NODE_FOTOSENSOR
```

Au démarrage du programme, le nœud se voit attribuer une adresse adéquate (5, 6 ou 7), et on détermine aussi s'il sera supervisé ou pas. Bien sûr, on peut adapter l'adresse dans le fichier `ElektorBusNode.c` selon les circonstances. Les fichiers liés au matériel `ElektorBusBoard.h` et `.c` sont réduits en rapport avec la petite taille de la carte de nœud expérimental. Malgré tout, le fichier hex ne peut pas se loger dans la mémoire flash d'un ATmega88, il faut donc le remplacer par un ATmega328 dont le brochage lui est compatible et qui coûte à peine plus cher. Comme la bibliothèque du bus devrait être à la base de nouveaux développements, disposer de 32 Ko de flash serait une nouvelle exigence pour la carte d'ElektorBus, 64 Ko ou plus serait encore préférable. Mais pour cet exemple, un seul nœud expérimental suffit. Nous aurons aussi besoin d'un convertisseur USB/RS485 (cf. **fig. 3**).

## Un peu de commande de moteur

Pour essayer le logiciel, il faut une photorésistance ou un relais à brancher sur le port d'extension du nœud expérimental, par le connecteur K4 (5 V, GND, PC0 à PC6). Le schéma de la photocellule (un potentiomètre ferait aussi l'affaire) et de l'extension à relais, vous les trouverez dans le 8<sup>e</sup> épisode de la série sur le bus [5]. Comme gâterie, nous y avons ajouté une troisième application, une commande simple de moteur pas à pas. Pour cela, on se sert de la platine *ad hoc* d'Elektor [6] [7] que l'on relie au nœud expérimental (**fig. 4**). Il s'agit d'ailleurs d'une petite anticipation sur un article sur l'ElektorBus à paraître, dans lequel sera présentée une platine pour moteur pas à pas compatible avec le bus.

Dans le téléchargement [4] se trouve aussi un dossier `UIBus` avec un fichier HTML très simple pour la commande par PC de l'ensemble. Il suffit de le faire glisser sur le bureau et de lancer `ElektorBusBrowser.exe` [8]. Faisons d'abord démarrer le superviseur avec le bouton approprié (**fig. 5**). Si le micrologiciel d'origine est chargé dans l'ATmega328, nous devrions voir les valeurs du capteur. Appuyer sur le bouton de test de la carte doit aussi successivement allumer





Figure 5. Une interface utilisateur HTML très simple pour tester nos trois exemples d'application.

et éteindre la LED rouge. L'état de la LED est reflété par le bouton radio sur le PC.

Si NODE\_RELAY est sélectionné dans le micrologiciel et qu'un relais est connecté, on peut activer et faire chuter le relais. La transmission au nœud s'opère aussi manuellement par le bouton de test.

En revanche, si la directive dans le micrologiciel est #define DEVICE\_NODE\_MOTORCONTROL, alors vous allez, du PC, faire avancer puis reculer le moteur de 1 000 pas.

### Dans la boule de cristal

Il reste du chemin à parcourir, la bibliothèque du bus en est à sa première version et n'englobe encore que les fonctions indispensables.

Parmi d'autres, il nous faudrait des fonctions d'intervalle, pour que les nœuds envoient périodiquement les valeurs des capteurs et, dans le même genre, des fonctions qui permettent l'initialisation de temporisateurs.

En attendant, nous avons là une belle petite bibliothèque pour commander diverses fonctions d'un contrôleur sans nous préoccuper du type de puce en service. Pour savoir si cette ébauche fonctionne réellement, nous l'avons testée sur un ATmega256, actuellement implanté sur la future carte Elektor. De quoi ouvrir une perspective vers une sorte de « bibliothèque de micrologiciels Elektor » à venir, donc pas uniquement dédiée à l'ElektorBus. Une idée fructueuse si elle permettait à d'autres électroniciens de porter le projet d'Elektor sur leur contrôleur préféré...

De tout ceci et plus encore, vous pouvez dès à présent être tenu informé sur [www.elektor-projects.com](http://www.elektor-projects.com).

(120582 – version française : Robert Grignard)

### Liens

- [1] [www.elektor.com/elektorbus](http://www.elektor.com/elektorbus)
- [2] [www.elektor.fr/110517](http://www.elektor.fr/110517)
- [3] [www.elektor.com/Uploads/2012/2/ElektorBusReference.pdf](http://www.elektor.com/Uploads/2012/2/ElektorBusReference.pdf)
- [4] [www.elektor.fr/120582](http://www.elektor.fr/120582)
- [5] [www.elektor.fr/110428](http://www.elektor.fr/110428)
- [6] [www.elektor.fr/110018](http://www.elektor.fr/110018)
- [7] <http://rch.it/schrittmotor>
- [8] [www.elektor.fr/110708](http://www.elektor.fr/110708)
- [9] [www.mingjong.net/index.php?action=prod\\_detail&p\\_id=88&lang=en](http://www.mingjong.net/index.php?action=prod_detail&p_id=88&lang=en)

## Fonctions principales de l'ElektorBus-C-Library

### Fonctions et variables dans ElektorBusLibrary.c

```
void SetValue(sender, receiver, channel, mode, setvalue)
```

ajoute à la Part de la mémoire émettrice une autre Part définissant une valeur de consigne pour un actuateur.

```
void TransmitValue(sender, receiver, channel, mode, currentvalue)
```

ajoute à la Part de la mémoire émettrice une autre Part représentant la valeur actuelle d'un capteur ou d'un actuateur.

```
void SendParts()
```

code toutes les Parts inscrites dans la mémoire de Parts émettrice pour en faire un message et le transmettre.

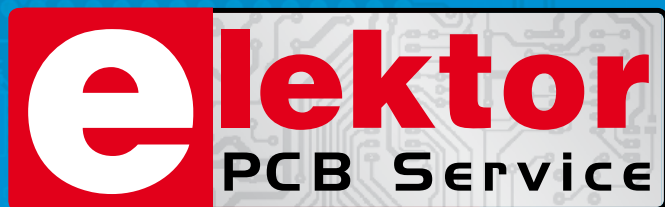
```
void SwitchTestLED(unsigned char ledStatus)
```

allume la TestLED (ledStatus = 1) ou l'éteint (ledStatus = 0).

```
void ToggleTestLED()
```

fait basculer la TestLED.

----- suite p.64



Présentation par Eurocircuits  
La référence Européenne  
pour les C.I. Prototypes et  
petites séries.

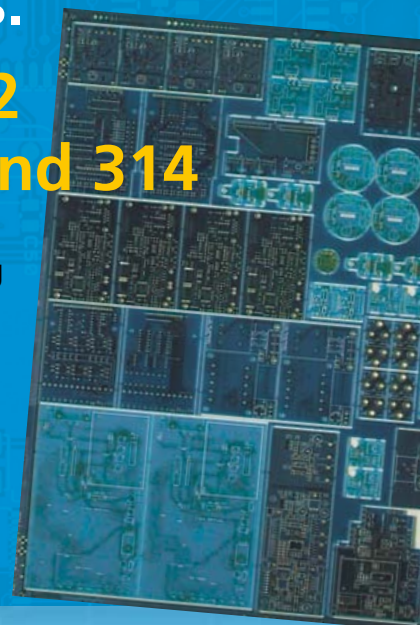
Venez nous visiter à **electronica 2012**  
du **13 au 16 novembre, Hall A5 – stand 314**

Réduisez vos frais de CI grâce à l'accès facile en ligne des services pooling

- Sans frais de dossier et d'outillages
- Sans minimum de commande à partir d'une pièce
- Commande en ligne instantanée sans règlement par avance

Nouveaux outils pour mettre vos dossiers en production plus vite

- PCB Visualizer contrôle vos fichiers immédiatement en ligne
- Conseils au conception pour la fabrication



**PCB proto** – *service rapide à prix très attractifs dédié aux bureaux d'études*

- 1 ou 2 C.I. en 2, 3, 5 ou 7 jours ouvrés
- Contrôle complet du dossier, finition comprenant 2 vernis épargne verts et une sérigraphie blanche – Isolements/conducteurs  $\geq 150\mu\text{m}$
- 1 C.I. 100 x 80mm en 7 jours - 2 faces 46,49 € - 4 couches 94,41 €
- 2 C.I. 100 x 80mm en 7 jours - 2 faces 36,47 € pièce - 4 couches 73,89 € pièce

*Prix avec TVA Française de 19,6% transport non compris*

**STANDARD pool** – *la plus large gamme d'options en pooling d'Eurocircuits*

- 1 à 8 couches isolements/conducteurs  $\geq 150\mu\text{m}$
- A partir de 2 jours ouvrés

**TECH pool** – *la force du pooling au service des C.I. haute densité*

- 2 à 8 couches isolements/conducteurs  $\geq 100\mu\text{m}$
- A partir de 4 jours ouvrés

**IMS pool** – *C.I. à haute dissipation thermique pour application de puissance (LED, conversion d'énergie, ...)*

- C.I. 1 face sur semelle métallique (SMI) – Isolements/conducteurs  $\geq 200\mu\text{m}$
- A partir de 3 jours ouvrés

**On demand** – *C.I. spéciaux minces et rigides*

- 0 à 16 couches isolements/conducteurs  $\geq 90\mu\text{m}$
- A partir de 2 jours ouvrés

[www.elektorPCBservice.com](http://www.elektorPCBservice.com)

TestLEDStatus

variable représentant l'état de la TestLED (allumée = 1, éteinte = 0)

-----

TestButtonToggleStatus

chaque action sur TestButtons fait basculer cette variable entre 0 et 1.

-----

OwnAddress

l'adresse du noeud.

-----

MasterAddress

l'adresse du nœud maître du bus.

-----

FreeBusPriority

indique quel message et après combien de *FreeBusPhases* le nœud répètera un message manqué.

-----

IsScheduled

indique si le nœud est supervisé (régulièrement interrogé) ou non (= 1 ou = 0).

-----

NodeDevice

un nombre qui identifie le type d'appareil pour différencier p.ex. cellule photo = 1, de relais = 2.

## Fonctions obligatoires dans le fichier principal (avec ou sans code) :

void ApplicationSetup()

appelée avant le début de la communication sur le bus, permet d'initialiser par exemple le convertisseur A/N ou le matériel connecté.

-----

void GetApplicationData()

appelée périodiquement, par exemple pour interroger les capteurs.

-----

void TestButtonClicked()

appelée après une action sur le TestButton.

-----

void ProcessPart(struct Part part)

après la réception d'un message à décoder en *Parts*, jusqu'à quatre, on appelle la routine pour chacune des *Parts* ; l'application peut aussi réagir aux commandes du maître ; pour les variables, on a accès à toutes les propriétés de la *Part* (cf. encadré *Parts*).

-----

void PreparePartsForScheduledMessage()

un nœud supervisé est régulièrement invité par le superviseur à transmettre au moyen d'un message, auquel cas la bibliothèque appelle alors cette fonction ; le code de l'application peut dès lors rassembler les *Parts* qui seront envoyées, des valeurs de mesure d'un capteur, par exemple.

## Fonction principales, macros et define de la couche matériel :

void ElektorBus\_ADC\_Setup(unsigned char ADCchannel)  
initialise le convertisseur A/N (une seule conversion, tension de référence = celle d'alimentation du contrôleur.

-----

unsigned short ElektorBus\_ADC\_GetValue(unsigned char ADCchannel)

force le convertisseur A/N à lire une valeur de tension et à la renvoyer.

-----

EXTENSION\_PORT\_SetDigitalDirection\_Output(Pin)

convertit une broche numérique du port d'extension en sortie ; Pin est le numéro de la broche physique parmi celles du port d'extension (par exemple Pin=0 sur PORT C=>PC0).

-----

EXTENSION\_PORT\_SetDigitalDirection\_Input(Pin)

convertit une broche numérique du port d'extension en entrée.

-----

EXTENSION\_PORT\_SetDigital\_High(Pin)

fait passer une broche numérique du port d'extension au niveau haut.

-----

EXTENSION\_PORT\_SetDigital\_Low(Pin)

fait passer une broche numérique du port d'extension au niveau bas.

-----

EXTENSION\_PORT\_Digital\_PinStatus(Pin)

demande l'état d'une broche numérique du port d'extension (==1 haut, ==0 bas).

-----

EXTENSION\_DIGITAL0\_PIN

EXTENSION\_DIGITAL1\_PIN

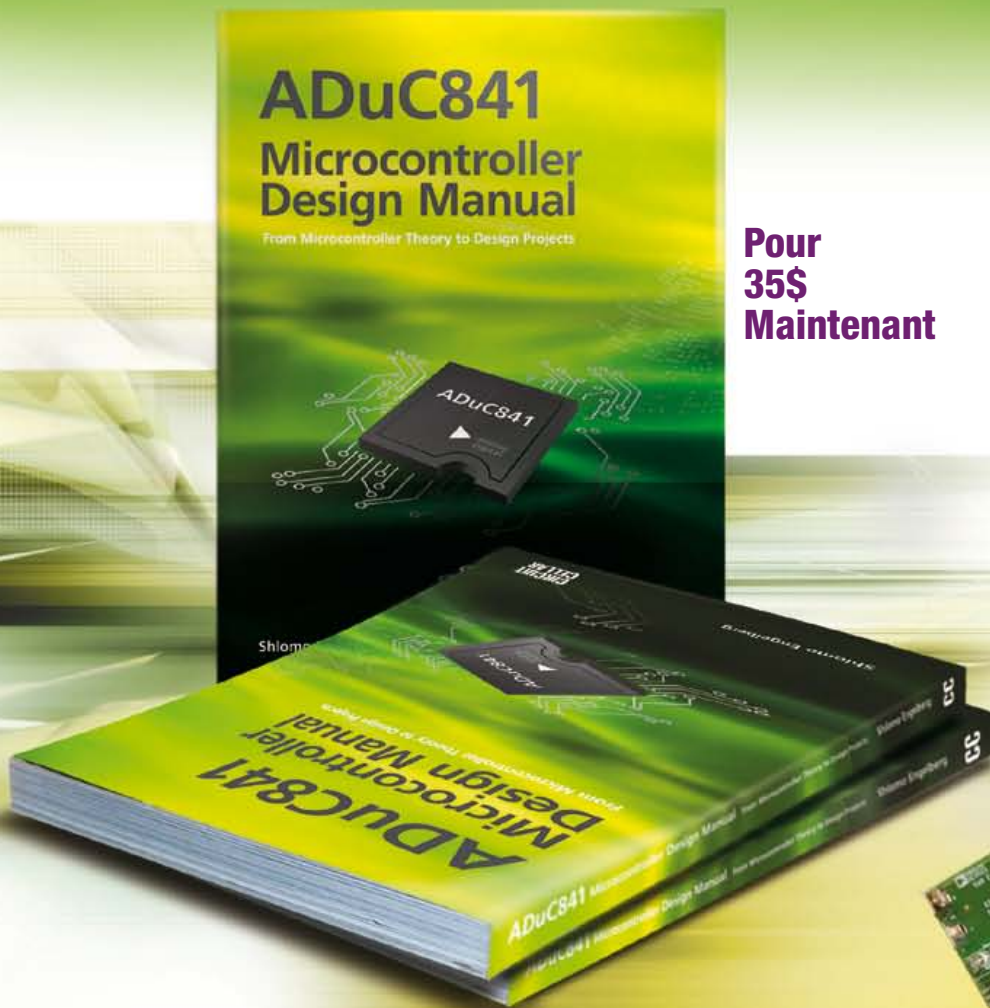
Au lieu de chiffres, mieux vaut utiliser pour Pin cette instruction. Cela permet de porter le code très facilement sur une autre carte avec un brochage différent.



# ADuC841 Microcontroller Design Manual:

De la théorie des microcontrôleurs  
à la conception de projets

Si vous avez toujours souhaité concevoir et programmer sur la base du microcontrôleur ADuC841, ou sur celle d'autres microcontrôleurs de la famille 8051, voici le livre qu'il vous faut. Grâce à des travaux pratiques d'initiation et de perfectionnement, vous serez bientôt en mesure de maîtriser les nombreuses façons d'utiliser un microcontrôleur. Parfait pour les universitaires !



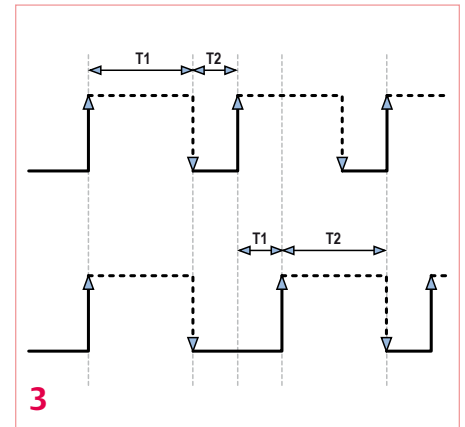
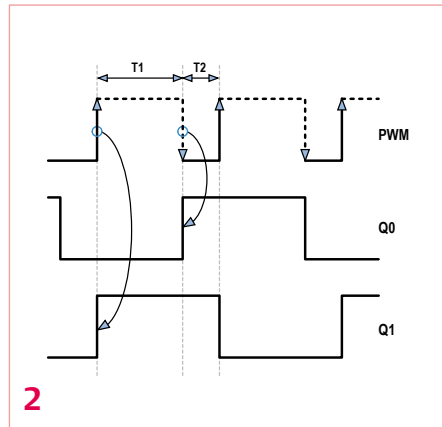
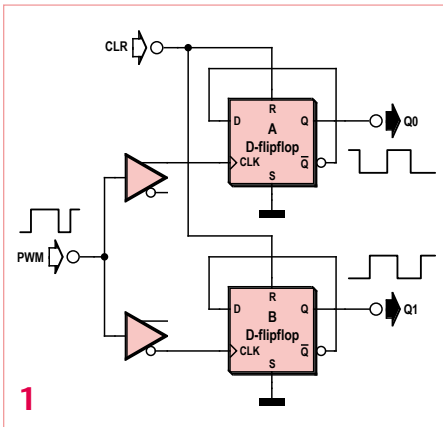
**Pour  
35\$  
Maintenant**

**Achetez-le aujourd'hui !**

**[www.cc-webshop.com](http://www.cc-webshop.com)**

# déphaseur commandé en MLI

Stephen Bernhoeft  
(Royaume-Uni)



Il existe des situations où faire varier le déphasage entre deux signaux carrés peut être bien utile. Sur la **figure 1**, les bascules D 'A' et 'B' transforment le signal MLI en signal carré de manière classique. Cependant, la bascule 'A' est cadencée par le front montant d'un signal MLI alors que la bascule 'B' l'est par son front descendant. Ainsi, le rapport cyclique du signal MLI détermine le

déphasage entre les sorties des deux bascules (**fig. 2**). Des déphasages de 0 ou  $2\pi$  ne seront cependant pas possibles. On pourrait faire la même chose en remplaçant A et B par des blocs ayant des sorties dites en quadrature. En prenant Q0 sur la sortie « 1 » (*in-phase*) du bloc A, et Q2 sur la sortie « Q » (*quadrature*) du bloc B, une différence de phase de  $\pi$  se transformera

en un déphasage de  $2\pi$ . Bien entendu, il ne sera pas possible d'obtenir un déphasage de exactement  $\pi/2$  (c.-à-d. en quadrature). Si le signal MLI est commandé par un  $\mu\text{C}$ , il est possible de choisir le sens du déphasage en inversant le sens (p. ex. niveau bas d'abord) de la MLI (**fig 3**).

(120193 – version française : Kévin PETIT)

Publicité



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

après le succès phénoménal du livre de Rémy Mallard voici enfin votre

## kit d'initiation

avec plaque d'expérimentation et pince coupante

avec vidéo didactique pour l'assemblage




livre + kit : **le cadeau idéal** pour un(e) débutant(e)

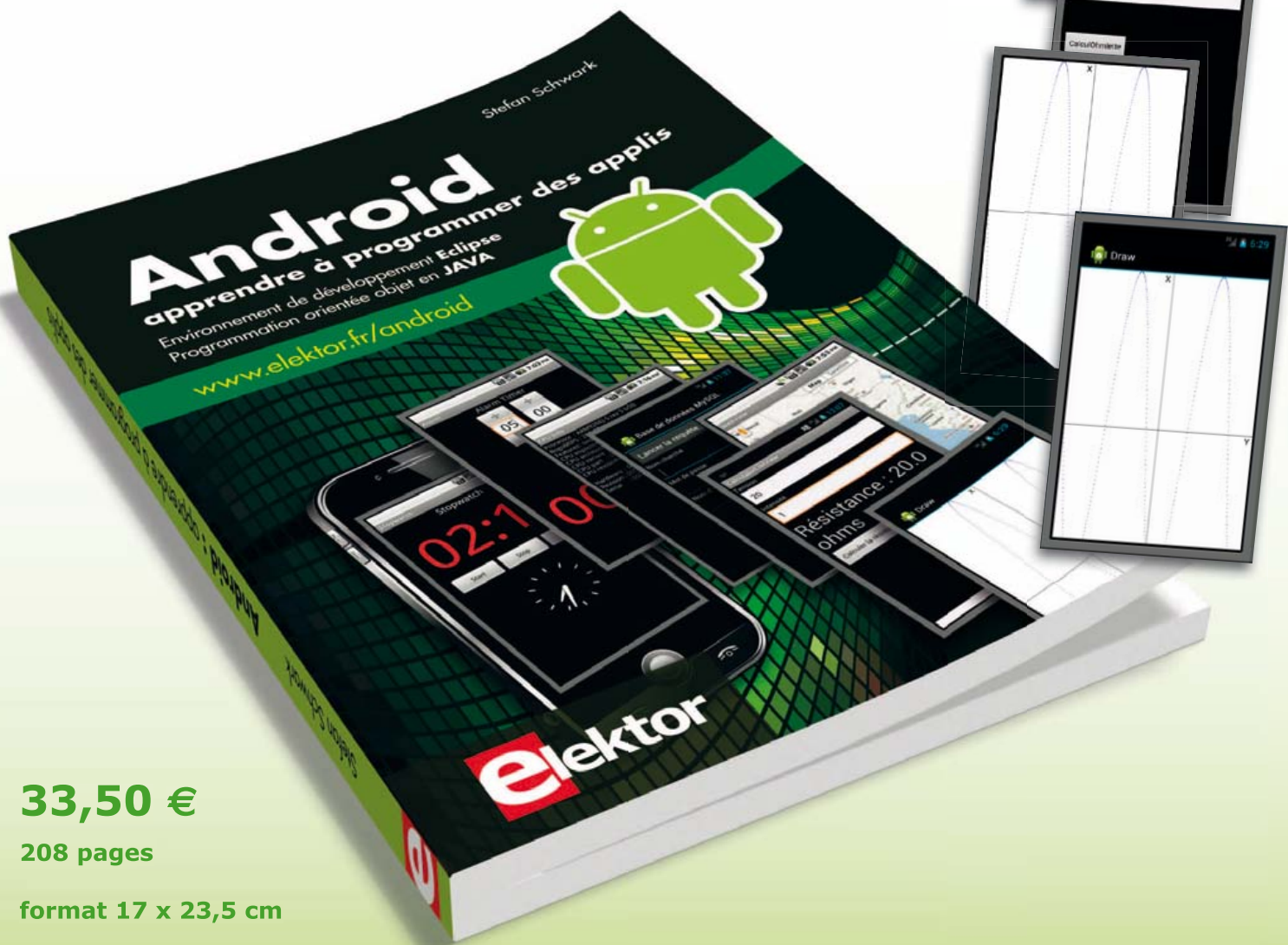




## Télécharger des applications existantes n'est pas le seul moyen de **rendre votre téléphone tactile encore plus intelligent !**

Voici un nouveau livre, publié par Elektor, pour apprendre à construire des applications sous Android et à tirer le meilleur de ce système d'exploitation.

C'est un ouvrage d'initiation, avec des exemples simples, variés et concrets, qui montre de façon progressive comment la combinaison de briques de code permet de créer toutes sortes d'applis dans l'environnement de développement Eclipse :  
calculatrice simple, interrogation des capteurs, exploitation des données GPS, communication par l'internet etc.



**33,50 €**

208 pages

format 17 x 23,5 cm

ISBN 978-2-86661-187-3

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

 **elektor**



# embarquez Linux ! (5)

## E/S, CAN, PWM, LAN & serveur web

La carte Linux-Elektor offre de nombreuses possibilités pour le développement d'applications. L'une d'entre elles est de lire des signaux analogiques et numériques au travers d'une connexion réseau, et de se servir d'un serveur web léger pour afficher les informations obtenues. Ici nous commanderons à distance la LED de la carte, et nous afficherons son état sur une page web créée de façon dynamique.

Benedikt Sauter [1]

Vous avez été nombreux à nous faire part de vos expériences avec le noyau et le chargeur de démarrage, mais aussi avec le système de fichiers et la carte SD (voir l'**encadré** « Image de la carte SD », si d'aventure vous aviez eu quelques... mésaventures). Nous voici arrivés en douceur au volet consacré à la construction d'une première « véritable » application. Puisque dans le monde de l'embarqué Linux la plupart des applications commandent des machines et collectent des données, nous allons d'abord

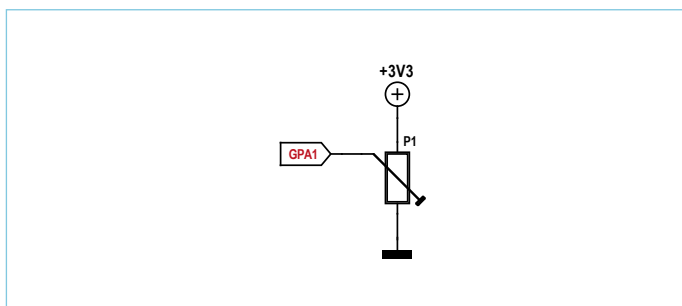


Figure 1. Schéma de connexion du potentiomètre.

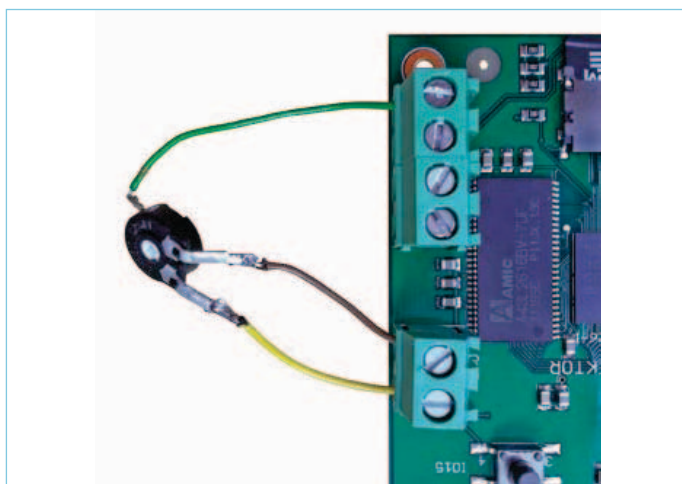


Figure 2. Le bornier relié à l'entrée analogique facilite le branchement.

voir comment lire et produire des signaux analogiques et numériques. Une connexion réseau nous fournira ensuite un accès distant à la carte, ouvrant ainsi la possibilité de la commander depuis une page web.

### Broches d'E/S numériques

Nous avons vu dans le deuxième volet comment allumer et éteindre la LED1 de la carte, la LED reliée à la broche GPIO3 du processeur. Les broches du port GPIO peuvent être utilisées en entrée ou en sortie, mais également servir d'entrées pour les interruptions. La procédure d'initialisation de ces broches vous est sans doute familière :

1. Activation de la broche en tant que broche GPIO.
2. Initialisation de la direction des données (entrée ou sortie).
3. Écriture d'une valeur, ou lecture du niveau de signal appliqué.

Sous Linux, nous pouvons nous servir des pilotes de périphérique pour dialoguer avec les broches GPIO depuis une console. Plaçons-nous d'abord dans le répertoire qui nous permettra de communiquer avec le pilote GPIO :

```
cd /sys/class/gpio
```

Définissons ensuite la broche reliée à la LED comme broche GPIO (voir schéma du lien [2]) :

```
echo 3 > export
```

Faisons de même avec la broche reliée au bouton-poussoir :

```
echo 15 > export
```

Configurons ensuite la broche de la LED en tant que sortie, celle du poussoir en tant qu'entrée :

```
cd gpio3
echo "out" > direction
```

```
cd ../gpio15
echo "in" > direction
```

Nous pouvons maintenant allumer ou éteindre la LED en écrivant l'état de sa sortie dans le fichier *value* :

```
cd ../gpio3
echo 1 > value
```

```
echo 0 > value
```

Le fichier (virtuel) *value* contient l'état du poussoir. La commande *cat* affiche son contenu :

```
cd ../gpio15
cat value
```

Le processus est simple, vous devriez à présent savoir également commuter le relais de la carte. La broche GPIO18 à laquelle il est relié doit bien sûr être configurée en tant que sortie.

### Convertisseur analogique-numérique

Le LPC3131 dispose de quatre entrées analogiques, chacune dotée d'une résolution maximale de 10 bits. Les résultats peuvent donc prendre toute valeur comprise entre 0 et 1023 (ou 0 et 0x3FF en hexa). La tension de référence utilisée est 3,3 V, soit la tension d'alimentation du port d'E/S auquel sont reliées ces broches.

On accède aux valeurs converties comme on accède à l'état du poussoir. Le convertisseur A/N a son propre pilote, mais celui-ci ne peut fournir les valeurs que d'un seul canal. Nous devons donc d'abord fixer le canal à interroger.

Le plus simple est de tester le comportement du convertisseur A/N à l'aide d'un potentiomètre d'ajustement. Reliez-le directement à l'alimentation de 3,3 V (**figure 1**), et branchez le curseur sur le bornier auquel est reliée la broche GPA1.

Votre montage devrait ressembler à celui de la **figure 2**. Nous pouvons maintenant initialiser le convertisseur et nous servir des commandes de la **figure 3** pour afficher les valeurs une par une. Entrer toujours la même commande est fastidieux, le programme *watch* est heureusement là pour exécuter une même commande à intervalles réguliers. Sa syntaxe est *watch -n intervalle\_en\_s commande*. Un intervalle d'une seconde convient :

```
watch -n 1 cat /dev/lpc313x_adc
```

Utilisez Ctrl-C pour mettre fin à son exécution. Comme GPA1, les canaux A/N GPA0 et GPA3 sont accessibles depuis le connecteur J5 de la carte Linux-Elektor (la broche GPA2 ne lui est pas reliée).

Suivez le montage de la **figure 4** pour protéger (dans une certaine limite) l'entrée du convertisseur A/N contre les surintensités et les surtensions : une résistance de 10 kΩ en série, et une zener de 3,3 V reliée à la masse.

### Produire un signal MLI

La modulation de largeur d'impulsion (MLI, PWM en anglais) est utilisée dans de nombreux domaines : production de tensions analogiques, commande de servomoteurs, audio, etc [3]. Sur la

```
root@gnublin:~# echo 1 > /dev/lpc313x_adc
set res to 10
root@gnublin:~# cat /dev/lpc313x_adc
0x21f
root@gnublin:~# cat /dev/lpc313x_adc
0x11d
```

Figure 3. Initialisation du CAN et affichage de deux valeurs.

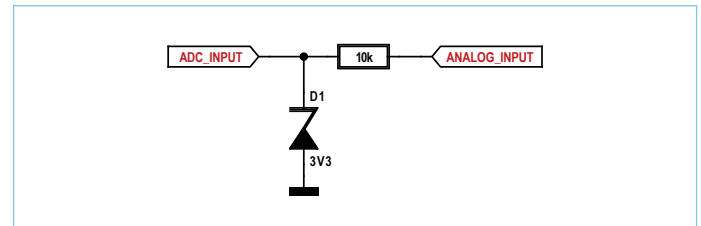


Figure 4. Comment protéger l'entrée du CAN.

carte Linux-Elektor, la sortie PWM du microcontrôleur est reliée au connecteur J5. Un moyen simple de tester cette sortie est de lui connecter un oscilloscope. En mode PWM, le contrôleur incrémente un compteur à 12 bits. Dès qu'une certaine valeur de référence est atteinte, la broche PWM passe du niveau haut au niveau bas (le compteur est remis à zéro en cas de dépassement de capacité, et la broche est en même temps replacée au niveau haut). La valeur de référence peut être choisie parmi n'importe quelle valeur représentable sur 12 bits, soit n'importe quelle valeur comprise entre

#### Listage 1 : PWM.

```
#include <stdio.h>
#include <stdlib.h>

#ifndef abs
#define abs(x) ((x) < 0 ? -(x) : (x))
#endif

int pwm(int value) {
    FILE* f = fopen("/dev/lpc313x_pwm", "wb");
    fputc(value & 0xff, f);
    fputc((value >> 8) & 0xff, f);
    fclose(f);
}

int main() {
    int value = 0;
    int b;

    while(1) {
        b = abs(63 - 2*value);
        pwm(b * b);
        value = (value + 1) % 64;
        usleep(1000);
    }
}
```

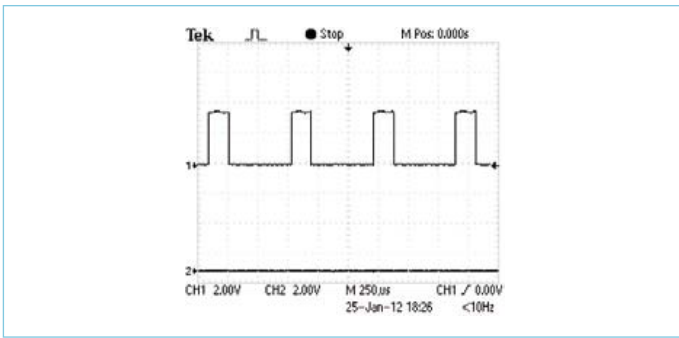


Figure 5. Sortie PWM avec une valeur de référence de 1000 pour le compteur.

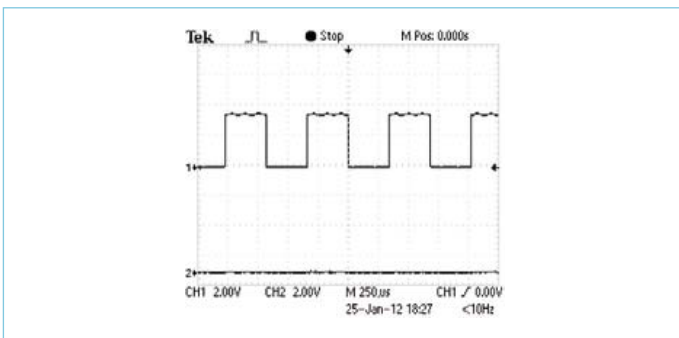


Figure 6. MLI avec rapport cyclique de 50 %.

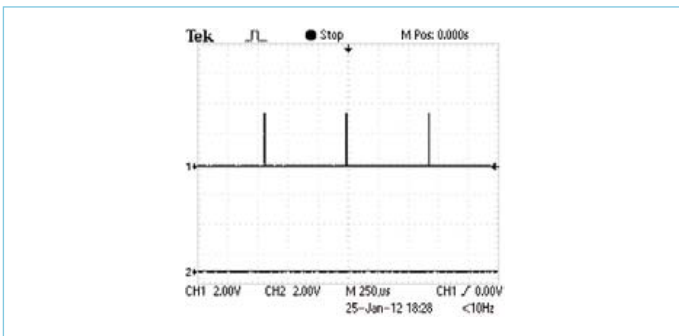


Figure 7. MLI avec rapport cyclique de 1 %.

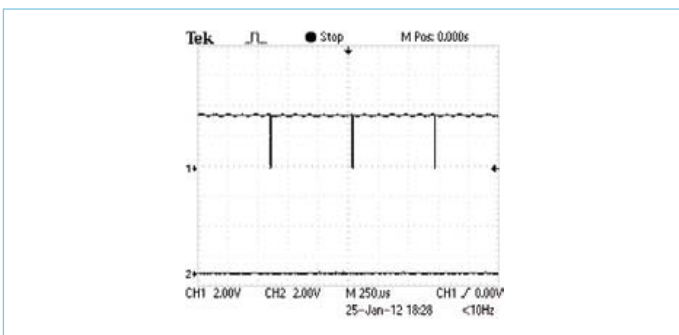


Figure 8. MLI avec rapport cyclique de 99 %.

0 et 4095. Le signal MLI passe immédiatement au niveau bas avec une valeur de référence de 0. À une valeur de 2000 correspond un rapport cyclique de 50 %.

Contrairement aux pilotes du convertisseur A/N et d'E/S, le pilote MLI a besoin d'une entrée binaire. Nous ne pouvons donc pas nous servir des commandes `echo` et `cat`, car elles interpréteraient ces entrées binaires comme des caractères (ASCII). Nous avons donc besoin du soutien d'un petit programme.

Il ne nous a guère fallu de temps pour l'écrire (en C) sur la carte. Il est reproduit dans le **listage 1**, et est déjà présent sous le nom `pwm.c` dans le répertoire d'utilisateur, le répertoire dans lequel vous vous retrouvez après vous être connecté. Le code de ce fichier doit juste être modifié pour contenir le nom du fichier de périphérique. Ouvrons-le à l'aide de l'éditeur `nano` présent sur la carte :

```
nano pwm.c
```

Servez-vous des touches de direction pour vous placer sur la ligne :

```
FILE* f = fopen("/dev/pwm", "wb");
```

Changez-la en :

```
FILE* f = fopen("/dev/lpc313x_pwm", "wb");
```

Enregistrez le fichier avec `Ctrl-o`, et quittez `nano` avec `Ctrl-x`. Le code peut-être compilé sur PC, ou directement sur la carte puisqu'un compilateur `y` est présent :

```
gcc -o pwm pwm.c
```

Lancez le programme une fois sa compilation terminée (ce qui peut prendre un certain temps) :

```
./pwm
```

L'oscilloscope montre comment varie graduellement le rapport cyclique, c'est-à-dire comment varient les durées des pauses et des impulsions.

Pour obtenir un signal possédant un certain rapport « durée d'une impulsion » / « durée d'une pause », on peut utiliser le script Python `pwm.py` placé dans le répertoire de l'utilisateur. Appelons d'abord l'interpréteur Python :

```
python
```

Nous sommes d'entrée en mode interactif, donc nous chargeons de suite le module PWM (une bibliothèque de fonctions écrites en Python) :

```
import pwm
```



## Image de la carte SD

Se retrouver dans une impasse lors des expérimentations avec la carte n'a rien d'improbable. Infaillible échappatoire, la restauration de la carte SD dans son état d'origine. Pour cela vous devez d'abord télécharger son image depuis la page du lien [8] (Téléchargements, Image de la carte SD, 120180-12.zip).

Décompressez ensuite l'archive :

```
unzip 120180-12.zip
```

Patientez jusqu'à ce que le message suivant apparaisse :

```
Archive: ../120180-12.zip inflating: Elektor_Linux_Board - Build_New_SD_Card.txt inflating: gnuvlin.img
```

Insérez dans un PC ou un lecteur de cartes la carte SD à écrire. Le système la monte automatiquement, mais puisque nous voulons y écrire une copie conforme de l'image, nous devons d'abord la démonter de façon manuelle.

Le plus simple est d'ouvrir une console et de lancer la commande `dmesg` après avoir connecté le lecteur de cartes. La sortie de `dmesg` devrait ressembler à ceci :

```
[ 1069.427374] sdf: sdf1 sdf2 [ 1069.430857] sd
5:0:0:0: [sdf] No Caching mode page present [
1069.430863] sd 5:0:0:0: [sdf] Assuming drive cache:
write through [ 1069.430868] sd 5:0:0:0: [sdf] Attached
SCSI removable disk [ 1070.002620] EXT2-fs (sdf1):
warning: mounting unchecked fs, running e2fsck is
recommended
```

Parmi les dernières lignes figure le nom de périphérique attribué par le noyau à la carte SD, ici `sdf`. Démontez-la avec la commande `umount` :

```
umount /dev/sdf1
```

Vous devez remplacer `sdf1` par le nom de périphérique de votre carte (plus précisément par le nom de périphérique de sa première partition, ce qu'indique ici le 1 de `sdf`).

L'image téléchargée peut maintenant être écrite sur la carte SD :

```
sudo dd if=gnuvin.img of=/dev/sdf
```

`sdf` est ici le nom de la carte en tant que périphérique de type bloc.

L'écriture complète peut prendre 10 minutes : pas de panique !

Les fonctions du module peuvent dès lors être appelées. L'une d'entre elles prend comme argument la valeur de référence du compteur PWM :

```
pwm.pwm_raw(1000)
```

La **figure 5** montre sa sortie à l'oscilloscope.

La **figure 6** montre le signal obtenu en passant 50 à la fonction `pwm()`, fonction qui attend un rapport cyclique exprimé en pour cent :

```
pwm.pwm(50)
```

Les instructions :

```
pwm.pwm(1)
pwm.pwm(99)
```

produisent des signaux dont les rapports impulsion/pause corres-

pondent respectivement à des rapports cycliques de 1 et de 99 % (**fig. 8** et **9**).

L'interpréteur Python se ferme avec `Ctrl-d`.

Le démarrage de Python demande un peu de patience, mais la réponse aux commandes est prompte une fois l'interpréteur chargé.

### Interface réseau

Nous avons vu dans le précédent volet comment utiliser un adaptateur USB/série avec la carte. L'adaptateur USB que nous allons maintenant installer servira à relier la carte Linux à un réseau Ethernet. Nous avons opté pour un adaptateur USB/LAN standard (**fig. 9**). Même si la plupart recourent aux mêmes puces, les modèles du commerce sont nombreux, donc vous avez le choix. Voici la référence du nôtre si vous préférez jouer la carte de la sécurité : D-Link DUB-E100 [4].

Dans le dernier numéro, nous avons compilé en dur le pilote de l'adaptateur USB/UART. Ici nous allons le compiler en tant que module du noyau, c'est-à-dire en tant que module chargeable en cours d'exécution. Le système de fichiers contient différents pilotes d'adaptateurs réseau. Chargeons celui du D-Link :

Publicité



**PCBs  
Muuuch Cheaper...**

**No-frills policy**

**17,22 EURO\***

5 pcs, 100 mm x 100 mm  
\*per piece, incl. VAT (23%)  
+ shipping costs e. g. Germany 10,89 EURO



**www.jackaltac.com**



Figure 9. Un adaptateur USB/LAN pour transformer la carte en interface réseau.

```
modprobe asix
```

La **figure 10** reproduit la sortie de cette commande. Trois pilotes sont présents dans le système de fichiers :

```
asix
pegasus
net1080
```

Si vous utilisez un adaptateur différent du nôtre et hésitez sur le choix du pilote, vous pouvez les charger les uns après les autres. Lorsque la commande :

```
ifconfig -a
```

indique une interface *eth0*, le réseau est prêt à transmettre des données.

```
root@gnublin:/lib/modules/2.6.33/kernel/drivers/net/usb# usb 1-1: new high speed USB device using lpc-ehci and address 2
usb 1-1: New USB device found, idVendor=2001, idProduct=3c05
usb 1-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-1: Product: DUB-E100
usb 1-1: Manufacturer: D-Link Corporation
usb 1-1: SerialNumber: 000001

root@gnublin:/lib/modules/2.6.33/kernel/drivers/net/usb# modprobe asix
eth0: register 'asix' at usb-lpc-ehci.0-1, ASIX AX88772 USB 2.0 Ethernet, fc:75:16:cf:6b:7f
usbcore: registered new interface driver asix
root@gnublin:/lib/modules/2.6.33/kernel/drivers/net/usb#
```

Figure 10. Reconnaissance automatique de l'adaptateur USB/LAN.

Si aucun des trois pilotes ne convient, vous pouvez en ajouter d'autres dans le noyau à l'aide de l'outil de configuration. Vous les trouverez dans le menu *Device Drivers -> Network device support -> USB Network Adapters*. Vous pourrez soit compiler le pilote dans le noyau (relisez [2] si vous avez oublié la procédure), soit compiler le module, le copier dans le système de fichiers, et plus tard le charger au besoin.

L'ensemble des modules se compile avec :

```
make modules
```

Leur installation sur la carte se fait avec la commande ci-dessous, en remplaçant */mnt* par le chemin d'accès à la carte SD :

```
make modules_install INSTALL_MOD_PATH=/mnt
```

L'adaptateur réseau une fois reconnu, on peut lui attribuer une adresse IP temporaire. Avant cela, par mesure de sécurité, il est préférable de s'assurer que l'adresse IP choisie est libre en se servant d'un PC relié au même réseau. On le fera en lançant un *ping* depuis une console Linux (la commande vaut pour Windows) :

```
ping 192.168.0.7
```

L'adresse passée en argument peut être choisie si aucun périphérique ne répond (*0 received*) :

```
2 packets transmitted, 0 received, 100% packet loss, time 1006ms
```

L'attribution de l'adresse se fait sur la carte Linux avec :

```
ifconfig eth0 192.168.0.7
```

Un nouveau *ping* depuis le PC devrait maintenant recevoir une réponse positive (**fig. 11**). Autre moyen d'obtenir une adresse de réseau, son attribution automatique par un serveur DHCP (**fig. 12**).

Le pilote sera chargé automatiquement à chaque démarrage si vous ajoutez son nom au fichier */etc/modules*. Ce fichier contient en effet la liste des modules que Linux charge automatiquement au démarrage.

```
sauter@pc:~$ ping 192.168.0.7
PING 192.168.0.7 (192.168.0.7) 56(84) bytes of data.
64 bytes from 192.168.0.7: icmp_req=1 ttl=64 time=1.24 ms
64 bytes from 192.168.0.7: icmp_req=2 ttl=64 time=1.50 ms
--- 192.168.0.7 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.248/1.375/1.502/0.127 ms
```

Figure 11. Réponse du ping.

L'adresse IP est à enregistrer dans le fichier `/etc/network/interfaces`. Ce fichier est déjà présent dans notre système de fichiers. Il suffit de l'ouvrir avec un éditeur de texte et d'y ajouter l'adresse IP correspondant à l'entrée `eth0`.

Après démarrage, la carte Linux-Elektor sera désormais toujours accessible dans le réseau interne.

### Serveur web

Grâce à cette connexion réseau, nous allons démarrer un serveur web léger, puis afficher une page web de démonstration dans un navigateur. Le répertoire de l'utilisateur `root` contient un script chargé de lancer le logiciel de serveur `lighttpd` :

```
root@gnublin:~# ./lighttpd-init.sh
```

Syntax OK

```
root@gnublin:~#
```

Entrez ensuite dans la barre d'adresse d'un navigateur l'adresse IP précédemment attribuée. Une page web vous envoie son bonjour (**fig. 13**).

La mission type d'un serveur web est d'afficher des pages HTML statiques. Pour afficher par exemple l'état d'une LED dans un navigateur, le serveur doit toutefois être en mesure d'assembler la page HTML de façon dynamique, puisqu'il doit prendre en compte l'état de la LED. Nous avons besoin pour cela d'une interface entre le serveur et un programme externe capable de reconnaître l'état de la LED et de créer la page web en conséquence.

L'interface la plus simple à cet égard est l'interface CGI (*Common Gateway Interface*), par laquelle le serveur peut exécuter quasiment n'importe quel programme. Seule condition, que le programme en question fonctionne en ligne de commande, c'est-à-dire puisse être lancé depuis la console (avec d'éventuels paramètres). Ce programme auxiliaire doit en outre écrire lui-même le code de la page HTML. Ici conviendra aussi bien un script shell Linux qu'un programme écrit en C, Python, ou encore PHP.

### Éteindre la LED depuis un navigateur

Illustrons les possibilités de l'interface CGI avec une petite application qui commande la LED à distance. Le script de cette application appellera l'interpréteur de commandes (la console). Nous devons tout d'abord configurer l'interface CGI dans le serveur web.

Ouvrez le fichier `/etc/lighttpd/modules.conf`, et à l'aide d'un éditeur de texte (nano ou vi p. ex.) remplacez la ligne :

```
#include "conf.d/cgi.conf"
```

par :

```
root@gnublin:~# udhcpc
udhcpc (v1.17.3) started
Setting IP address 0.0.0.0 on eth0
Sending discover...
Sending select for 192.168.0.182...
Lease of 192.168.0.182 obtained, lease time 864000
Setting IP address 192.168.0.182 on eth0
Deleting routers
Adding router 192.168.0.1
Recreating /etc/resolv.conf
Adding DNS server 192.168.0.1
root@gnublin:~#
```

Figure 12. Attribution d'une adresse IP par serveur DHCP.



Figure 13. La page de test du serveur web.

```
include "conf.d/cgi.conf"
```

Dans le fichier `/etc/lighttpd/conf.d/cgi.conf`, remplacez de même la ligne :

```
#alias.url += ( "/cgi-bin" => server_root + "/cgi-bin" )
```

par :

```
alias.url += ( "/cgi-bin" => var.server_root + "/cgi-bin" )
```

Le serveur web sait désormais que les fichiers du répertoire `/cgi/bin`





Figure 14. Commuter la LED depuis un navigateur.

sont des programmes exécutables (et non pas des pages HTML à envoyer au navigateur).

Toujours dans ce même fichier, nous devons ajouter les scripts shell à la liste des langages reconnus comme programmes CGI. Repérez le bloc suivant :

```
cgi.assign = ( ".pl" => "/usr/bin/perl",
               ".cgi" => "/usr/bin/perl",
               ".rb" => "/usr/bin/ruby",
               ".erb" => "/usr/bin/eruby",
               ".py" => "/usr/bin/python" )
```

Complétez-le avec la ligne «.sh» => «/bin/sh», :

```
cgi.assign = ( ".pl" => "/usr/bin/perl",
               ".cgi" => "/usr/bin/perl",
               ".rb" => "/usr/bin/ruby",
               ".sh" => "/bin/sh",
               ".erb" => "/usr/bin/eruby",
               ".py" => "/usr/bin/python" )
```

Créons un dossier pour les programmes CGI :

```
mkdir -p /srv/www/htdocs/cgi-bin
```

Copiez enfin le script du **listage 2** dans le fichier *example.sh*, que vous ouvrirez p. ex. avec :

```
nano /srv/www/htdocs/cgi-bin/example.sh
```

Vous pouvez également télécharger ce script depuis [5] si vous avez le doigt lent ou fatigué.

Configurons la broche et la direction des données en tant que sortie, afin que le serveur web puisse commander la LED. Soit, depuis la console :

```
echo 3 > /sys/class/gpio/export
echo out > /sys/class/gpio/gpio3/direction
```

Le serveur web ne travaille jamais en tant que *root* pour des raisons de sécurité, aussi devons-nous lui attribuer temporairement les droits qui lui permettront d'accéder à la LED :

```
chown lighttpd:lighttpd /sys/class/gpio/gpio3/value
```

Chaque utilisateur présent dans le système peut désormais accéder à la LED. Ce n'est certes pas très optimal du point de vue de la sécu-

rité, mais une attribution plus ciblée des droits nous entraînerait trop loin. Référez-vous au lien [6] pour plus de détails.

Le serveur a en outre besoin d'un dossier pour enregistrer les journaux (les fichiers « log »), ainsi que des droits d'accès appropriés :

```
mkdir /var/log/lighttpd
chown -R lighttpd:lighttpd /var/log/lighttpd
```

Tout est en place pour lancer le serveur :

```
root@gnublin:~# /etc/init.d/lighttpd restart
```

Vous devriez voir le message :

```
Syntax OK
```

### Listage 2 : Script CGI pour l'écriture de la page web.

```
#!/bin/sh

if [ "$REQUEST_METHOD" == "POST" ]
then
  if [ 'cat /sys/class/gpio/gpio3/value' == 1 ]
  then
    echo 0 > /sys/class/gpio/gpio3/value
  else
    echo 1 > /sys/class/gpio/gpio3/value
  fi
fi

echo "Content-Type: text/html; charset=utf-8"
echo ""
echo "<html>"
echo " <head>"
echo "   <title>Webserver CGI Port 3 (LED)</title>"
echo " </head>"
echo " <body>"
echo "   <h1>Control-Panel CGI Port 3</h1>"
if [ 'cat /sys/class/gpio/gpio3/value' == 1 ]
then
  echo "   Port: On"
else
  echo "   Port: Off"
fi
echo "   <br><br>"
echo "   <form action=\"/cgi-bin/example.sh \" method = \"POST\">"
echo "     <input type=\"submit\" value=\"Click\">"
echo "   </form>"
echo " </body>"
echo "</html>"
```

Dans un navigateur, entrez l'adresse IP précédemment attribuée. La page qui s'affiche indique l'état de la LED (fig. 14).

Nous avons écrit une commande déclenchée à l'aide d'un mini-formulaire HTML, mini car il n'est constitué que d'un seul bouton de validation. Le navigateur transmet habituellement les données du formulaire au serveur web lorsque l'utilisateur appuie sur un tel bouton. Ici le mécanisme de transmission ne sert qu'à demander au serveur d'exécuter une nouvelle fois le script `/cgi-bin/example.sh`. Le script commute la LED1 de la carte, puis construit la nouvelle page HTML avec le message d'état approprié.

### Perspectives

Dans le prochain volet, nous utiliserons plus de fonctions de la carte grâce à une interface HTML plus complexe. Une interface utilisateur reste bien sûr insuffisante pour exploiter toutes les possibilités de la carte, donc nous apporterons un peu d'intelligence à l'ensemble à l'aide d'un petit programme qui tournera en tâche de fond. Parlons aussi d'ores et déjà du dernier épisode de la série (à paraître

en janvier 2013). Son contenu sera particulier, en ce sens que nous vous invitons à en choisir le ou les thèmes. Grosse envie ou petit caprice, faites-vous entendre sur la page web du lien [7] !

(120182)

### Liens

- [1] [sauter@embedded-projects.net](mailto:sauter@embedded-projects.net)
- [2] [www.elektor.fr/120181](http://www.elektor.fr/120181)
- [3] [http://fr.wikipedia.org/wiki/Modulation\\_de\\_largeur\\_d\\_impulsion](http://fr.wikipedia.org/wiki/Modulation_de_largeur_d_impulsion)
- [4] <http://shop.embedded-projects.net/gnublin>
- [5] [www.elektor.fr/120182](http://www.elektor.fr/120182)
- [6] [http://en.gnublin.org/index.php/Permissions\\_GPIO](http://en.gnublin.org/index.php/Permissions_GPIO)
- [7] [www.elektor.fr/linux-feedback](http://www.elektor.fr/linux-feedback)
- [8] [www.elektor.fr/120180](http://www.elektor.fr/120180)

Publicité

## Design your own PC Visual Processing and Recognition System in C#

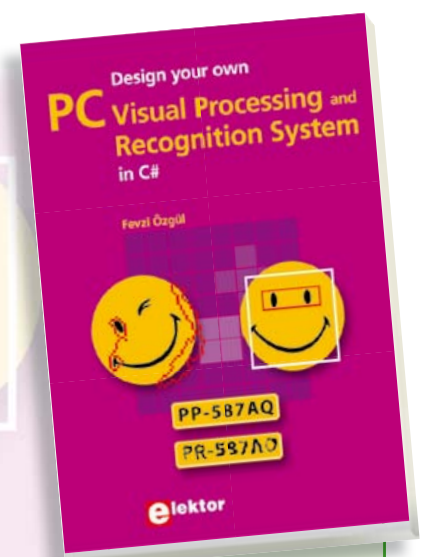
### ➔ Nouveau livre d'Elektor en anglais

Reconnaissance d'objet, détection et identification de visage, reconnaissance optique de caractères, détection de présence et de mouvement. Voilà les sujets chauds, liés au traitement de l'image. L'apparition de nombreuses bibliothèques libres (*open source*) de traitement de l'image permet maintenant d'inclure dans les applications.NET des fonctions complexes d'imagerie.

Ce livre s'adresse aux ingénieurs, scientifiques et amateurs éclairés, compétents en programmation et intéressés par les techniques de traitement de l'image sur PC.

Il utilise le langage de programmation orienté objet C# de Microsoft, avec des exemples pratiques et utiles, pour vous permettre de développer des logiciels de traitement de l'image de grande qualité.

L'ouvrage commence par une revue détaillée des principes de base du traitement de l'image. Il se poursuit par la présentation et l'étude de deux bibliothèques open source de traitement de l'image très raffinées : *AForge.NET* et *Emgu.CV* ; ces bibliothèques sont téléchargeables gratuitement et mises en œuvre dans l'environnement Microsoft Visual Studio.



307 pages  
Format 17 x 23,5 cm  
ISBN 978-1-907920-09-7  
39,95 €

Pour commander en ligne et bénéficier d'une remise spéciale de 5% :

[www.elektor.fr/e-choppe](http://www.elektor.fr/e-choppe)

# 3 mois de stage au labo

Il y a quelques semaines, François-Xavier Maurille, étudiant en 2<sup>e</sup> année d'École d'Ingénieurs en Électronique et Informatique à l'ESEO Angers bouclait un stage de 3 mois au labo d'Elektor aux Pays-Bas. Des circonstances fortuites ont fait que je ne l'avais pas rencontré au début de son séjour. D'autres hasards de la vie quotidienne ont fait que nos chemins ne se sont plus croisés à des moments où nous aurions été disponibles en même temps l'un et l'autre. Finalement, ce n'est que lorsqu'il m'a parlé par hasard, à la cantine, de son rapport de stage, trois jours avant de rentrer en Vendée, que nous avons fait connaissance pour de bon. Rapidement il est apparu que nous partagions le goût de l'électronique au service de la musique. Aussitôt j'ai souhaité en savoir plus sur lui et j'ai commencé par lire son compte-rendu.

François-Xavier Maurille & Denis Meyer

**DM :** *À la lecture de votre rapport, j'ai été frappé par la qualité du texte. La plupart des électroniciens, quand ils écrivent, ne paraissent pas se soucier de ce qui se passera dans l'esprit de ceux qui les liront. Si leurs montages comportaient les mêmes fautes de grammaire ou d'orthographe et autant de défauts de syntaxe que leur prose, ces circuits ne fonctionneraient jamais. Or vous, vous savez écrire... Est-ce le fruit de votre formation scolaire ou la manifestation d'un goût particulier pour la langue et l'écriture ?*

**FXM :** [Surpris] Au risque de décevoir un homme de métier, je ne suis en aucun cas attiré par l'écriture, c'est plutôt quelque chose que je redoute. Je suis de la génération de Français qui écrit avec beaucoup (trop) de fautes. Même jusqu'en BAC +4 ma formation inclut du soutien en français.

**DM :** *Eh bien, ce soutien est plus efficace chez vous que chez d'autres. Parlons de votre stage à l'étranger. Ça n'a plus rien d'exotique en 2012, même pour un étudiant français, Pourquoi avez-vous choisi non seulement un coin de province des Pays-Bas, mais en plus une maison d'édition ?*

**FXM :** Je connaissais Elektor par ses projets audio. J'ai par exemple agrémenté une vieille guitare avec un préamplificateur basé sur un article d'Elektor. J'ai l'habitude de lire le magazine au centre de doc de mon École. J'ai découvert en début d'année qu'Elektor disposait d'un labo, et cette année je voulais faire mon stage technique à l'étranger. Alors, sans trop espérer, j'ai tenté le coup (Je pensais à l'époque qu'Elektor était en Angleterre...), mon e-mail est passé d'ordinateur en ordinateur avant de recevoir une réponse positive.

**DM :** *Chez Elektor, vous n'êtes ni dans l'industrie ni dans le secteur auquel vos études vous destinent. Comment est l'ambiance au labo d'Elektor, en deux mots ?*

**FXM :** Deux mots ? « Détendue » et « efficace ». L'équipe est vraiment accueillante et c'est plaisant de travailler dans cette ambiance.

**DM :** *Quels ont été les sujets et les travaux principaux de votre stage ?*

**FXM :** C'était assez vague au départ : « étude et conception de montages électroniques au sein du laboratoire d'Elektor, en vue d'une publication ».

**DM :** *... mais encore ?*

**FXM :** J'ai surtout travaillé sur une batterie électronique, au sens musical du terme (*boum boum tchac*). J'ai d'ailleurs commencé le prototype à partir de la carte Linux embarqué d'Elektor<sup>1</sup> à la demande du labo à la recherche d'applications intéressantes pour cette plateforme qui venait

d'être publiée. Finalement, j'ai poursuivi ce projet sans OS embarqué, avec un microcontrôleur PIC32, à cause de problèmes<sup>2</sup> de délais de lecture de données audio.

**DM :** *... qui ont fait l'objet d'un article dans Elektor de septembre 2012. Votre première contribution au magazine ?*

**FXM :** Oui, c'est dans la rubrique « Bruits de Labo ». Tout est allé plus vite que je ne pensais. J'ai même assuré le support technique pour quelques problèmes avec cette plateforme Linux, notamment au sujet des cartes SD où est stocké le système de fichiers Linux. C'est mon script de correction de ce système de fichiers qui a été mentionné dans la revue et qui est disponible sur le site d'Elektor.

**DM :** *... pour le plus grand bonheur des centaines d'utilisateurs de cette carte.*

**FXM :** Des centaines ? J'ai entendu qu'on en avait déjà vendu plus de 1000 ?

**DM :** *Oui, on a largement passé le cap du millier, mais beaucoup de lecteurs en ont acheté plusieurs. Difficile de faire un compte précis en termes d'utilisateurs. En tout cas, c'est un grand succès, auquel vous avez contribué.*

**FXM :** Je n'ai pas fait grand chose, c'est pas moi qui ai conçu la carte quand même ! On m'a demandé aussi d'effectuer les tests de vitesse de lecture/écriture dans la RAM de cette carte Linux, après un changement de composant dans une nouvelle version de la carte. Et j'ai préparé la documentation de la mise en place du système Linux. Tout ça est disponible sur le site d'Elektor. Et la documentation technique de ma batterie se trouve sur le site [www.elektor-projects.com](http://www.elektor-projects.com)

**DM :** *Ah, bien, c'est précisément la fonction de ce site créé récemment pour permettre à la communauté de suivre les projets en cours et d'y participer.*

**FXM :** Ma batterie ne sera donc pas publiée dans le magazine lui-même ?

**DM :** *Vous avez sans doute constaté que la rédaction d'un magazine, c'est comme le désert, plus on avance, plus l'horizon recule ?*

**FXM :** [éclat de rire] C'est ça. J'ai compris que vous faites pas mal de

<sup>1</sup> La carte *embarquez Linux !* fait l'objet d'une série d'articles depuis le numéro de mai 2012. Elle est en vente sur le site d'Elektor.

<sup>2</sup> « GPIO access on Elektor Linux board », « SD card correction script », Elektor septembre 2012 N°429 (rubrique « E-Lab »)







compromis avec le calendrier de publication. Les échéances sont pourtant là pour être respectées, non ?

**DM :** *En théorie peut-être. D'après mon expérience, dans une maison d'édition, elles semblent faites pour être reportées.*

**FXM :** Au début de mon stage, la contrainte principale m'a paru en effet la date de publication des projets. Il faut s'y prendre à temps pour leur finalisation et pour la préparation d'une documentation technique, qui elle doit être aussi riche que possible en vue de l'écriture des articles. Une autre difficulté a résidé pour moi dans le fait d'avoir à travailler sur des plateformes que je n'avais jamais utilisées ni même vues au cours de ma formation, notamment pour cette toute nouvelle carte Linux embarqué d'Elektor. Quand je suis arrivé, il n'y avait même pas encore de support technique.

**DM :** *Vous vous en êtes bien tiré. Avez-vous fini par trouver une application pour cette plateforme dont le nom exact est « embarquez Linux ! » ?*

**FXM :** Bien sûr, j'ai proposé un jeu de pendu, que j'ai baptisé *Hang Tux*, présenté et documenté lui aussi sur le site [www.elektor-projects.com](http://www.elektor-projects.com).

**DM :** *Nous y reviendrons. Parlons encore du stage. Trois mois, c'est vite passé. Il y a sans doute eu des temps forts, mais peut-être aussi des passages à vide ?*

**FXM :** Pas trop. De juin à août 2012, je n'ai pas chômé, entre la présentation de mes différents projets à l'équipe à mon arrivée, puis les tests et la prise en main de la carte Linux embarqué d'Elektor, l'écriture du script de correction de carte mémoire SD, la réalisation du projet *Tux Pad*, du projet *PIC32 Drum Pad* avec conception de la carte électronique, la programmation du jeu *Hang Tux* pour la carte Linux, la finalisation du code de mes deux projets et la rédaction de leur documentation technique, la réception et le test des cartes électroniques... [après quelques secondes de réflexion] ah, oui, j'allais oublier la validation du bon fonctionnement du TAPIR pour la RFID. En fait, vu que je prenais le train tous les jours, Thijs m'avait simplement demandé de tester le TAPIR pour la RFID dans la gare.

**DM :** *Je suppose que le projet *Tux Pad*, c'est la version abandonnée de la batterie sur la carte Linux. En tout cas, c'est ce que j'appelle un stage intensif. Avez-vous l'impression que c'est la cadence normale au labo d'Elektor ?*

**FXM :** Oui c'est ça, *Tux Pad*, ça sonnait bien pourtant... mais je me suis

consolé avec le simplissime *Hang Tux*. Cadence de production normale, je ne sais pas trop, mais de ce que j'en ai vu en 3 mois, les projets s'enchaînent assez vite pour chacun dans le labo.

**DM :** *Dans votre rapport, vous notez qu'Elektor a été créé en 1965 aux Pays-Bas, et en 1978 en français. C'est loin ça. D'autre part, le siège de la société est un imposant château. Le poids de tant d'histoire est-il perceptible ? Comment voyez-vous l'adaptation d'Elektor au monde en train de changer ?*

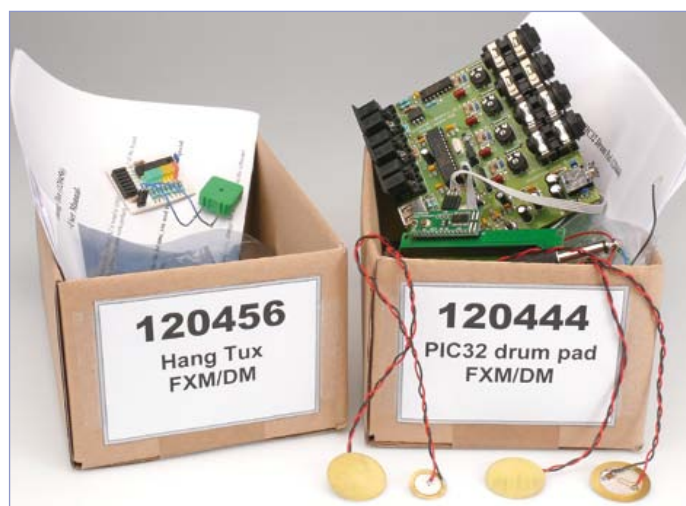
**FXM :** Je ne suis pas sûr de bien comprendre le « poids de l'histoire » ... mais la première fois que je suis descendu au sous-sol, je suis tombé nez à nez avec un chevalier en armure en standby près de l'escalier. Oh la trouille, j'étais pas très fier. Après on s'habitue vite au fait d'aller travailler dans un château tous les matins ! Je n'ai pas énormément de recul sur l'évolution d'Elektor, mais voyant tous les projets basés sur des microcontrôleurs, je pense qu'Elektor est loin d'être dépassé aujourd'hui.

**DM :** *Vous parlez aussi de l'influence d'Elektor sur l'orientation professionnelle de bon nombre de jeunes. C'était vrai quand l'électronique n'était pas la branche industrielle qu'elle est devenue avec l'explosion de la micro-informatique, mais aujourd'hui ? Elektor peut-il encore inspirer des jeunes dans leur choix ?*

**FXM :** Certainement. En arrivant ici, j'ai découvert qu'Elektor, outre sa rédaction et son labo, s'appuyait aussi sur une vaste communauté d'auteurs de par le monde, qui coopèrent librement aux publications. C'est stimulant.

**DM :** *Et l'une des principales tâches du labo est donc de tester la viabilité des projets envoyés par ces auteurs. Qu'en est-il de vos méthodes de travail ?*

**FXM :** La démarche que j'ai suivie pour ma batterie est en partie celle utilisée au sein de mon école, l'ESEO d'Angers. J'ai commencé par des tests unitaires pour chaque module. C'est ainsi que je me suis rendu compte de la difficulté du projet initial de batterie électronique sur la carte *embarquez Linux* ! Ensuite, sur le projet PIC32, j'avais à disposition un kit de développement *Microstick II* pour prendre en main les différents périphériques. J'ai alors pu écrire des fonctions pour chaque périphérique et faire le choix des composants en conséquence. Au début du mois d'août, la conception de la carte était finie. Après réception et assemblage de la carte, j'ai pu réutiliser



mes fonctions pour vérifier le bon fonctionnement du proto et travailler sur la version finale du code.

**DM :** Vous utilisez des sons de batterie échantillonnés ?

**FXM :** Oui, stockés sur une clef USB. Une fois la clé connectée, les échantillons sont joués plus ou moins fort selon la force de frappe des pads de la batterie.

**DM :** Combien y a-t-il d'entrées ?

**FXM :** Quatre seulement, c'est donc une petite « batterie », mais le principe de fonctionnement est le même quel que soit leur nombre.

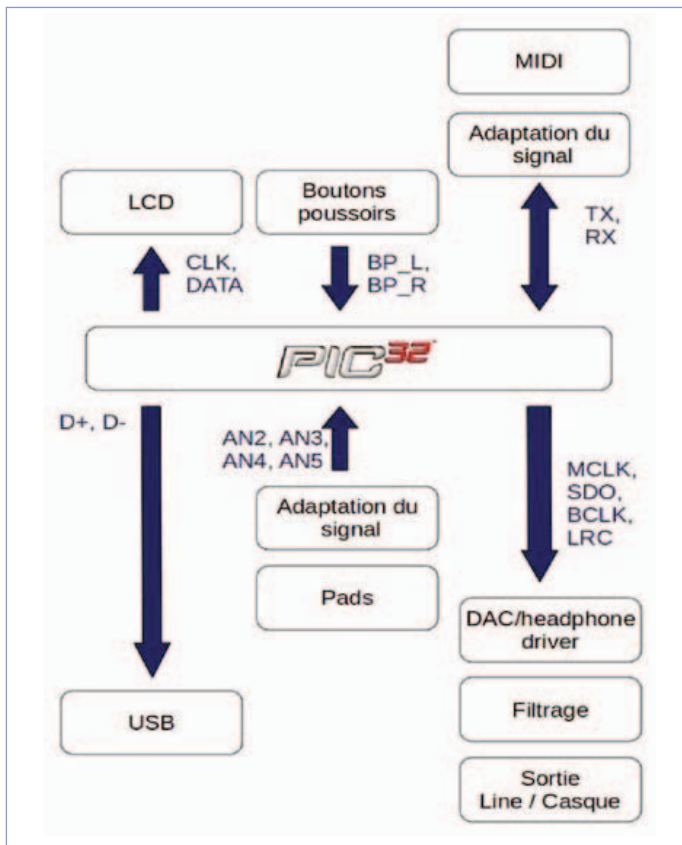
**DM :** Mais les contraintes ne sont pas les mêmes.

**FXM :** Non, pas quand on joue en temps réel. Il faut toujours surveiller la latence des signaux, sous peine de déformer le son. L'échange de données avec la clé USB est complexe. J'ai eu beaucoup de choses à apprendre ou approfondir.

**DM :** Faute de pouvoir rentrer ici dans les détails, nous pouvons reproduire l'architecture du projet « PIC32 Drum Pad ».

**Deux mots sur le choix du micro ?**

**FXM :** La nouveauté du PIC32MX250 (apparu en 2011) présente un intérêt particulier pour Elektor, puisqu'il n'y a pas encore beaucoup de projets qui l'utilisent. La version PIC32MX128b dispose de tous les périphériques nécessaires dans un boîtier DIP28, petit et facilement utilisable. Les périphériques sont le convertisseur A/N pour la capture du signal délivré par les pads, l'PS/SPI pour l'envoi des données audio au convertisseur N/A, l'USB en mode *host* permettant d'utiliser les données sur une clef USB, l'UART pour



l'envoi et la réception de données MIDI et le GPIO pour l'utilisation d'un écran LCD et de boutons poussoirs.

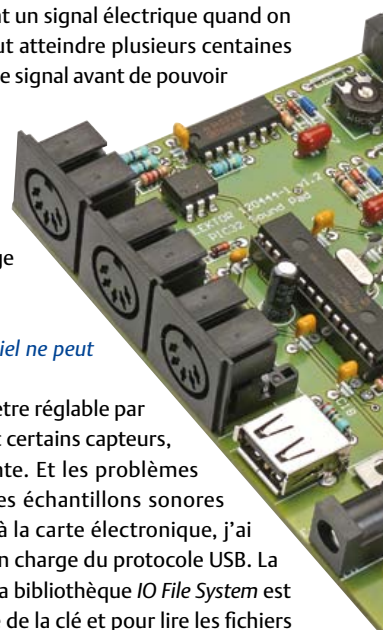
**DM :** Cela vous embêterait-il de revenir sur les moments... disons forts ?

**FXM :** Il y a l'embarras du choix. La gestion des périphériques les uns par rapport aux autres, par exemple, et donc la gestion à bon escient des interruptions du  $\mu C$ . Au premier abord, il paraît évident que la priorité absolue doit être accordée à l'envoi de données sonores ; cependant, stopper la communication lors d'un échange de données avec la clef USB plante le  $\mu C$ . J'ai compris que la priorité doit donc être accordée au périphérique USB, puis à l'envoi des données sonores au DAC. D'un autre côté il faut aussi que le temps de latence entre la frappe sur la *pad* et le déclenchement du son soit le plus faible possible.

Je pense à un autre détail, à propos des capsules piézoélectriques utilisées comme capteurs, parce qu'elles produisent un signal électrique quand on les déforme. Cette tension alternative peut atteindre plusieurs centaines de volts crête à crête. Il faut donc adapter ce signal avant de pouvoir l'exploiter correctement. La capacité nécessaire pour ne pas trop amortir le signal ne permet pas toujours de le lisser parfaitement. Dans certains cas, lorsque l'on frappe la capsule piézoélectrique, le signal peut être vraiment difforme. Le filtrage du signal est donc vraiment délicat.

**DM :** L'analogique garde ses droits, le logiciel ne peut visiblement pas tout faire.

**FXM :** J'ai tout de même rajouté un paramètre réglable par l'utilisateur dans le logiciel parce que, avec certains capteurs, l'amplitude du signal filtré est insuffisante. Et les problèmes de logiciel ne manquent pas. Pour lire les échantillons sonores directement sur une clef USB connectée à la carte électronique, j'ai utilisé la pile USB *Microchip* pour la prise en charge du protocole USB. La carte fonctionne alors comme hôte USB. La bibliothèque *IO File System* est utilisée pour naviguer dans l'arborescence de la clé et pour lire les fichiers







audio. Au début j'ai eu quelques problèmes pour détecter la clef USB, en utilisant l'oscillateur interne du  $\mu$ C qui ne permettait pas une utilisation normale de l'USB. La solution a été de mettre un quartz de 8 MHz, et ainsi de pouvoir configurer convenablement les PLL.

**DM :** *Donnez-nous un autre exemple de difficulté de la programmation ?*

**FXM :** La lecture et l'écriture des données audio doivent se faire dans un seul et même tableau, cependant il ne faut pas écraser les données avant de les avoir lues. Le pointeur d'écriture doit toujours être derrière le pointeur de lecture. Ces deux pointeurs n'avancent donc pas à la même vitesse, ce qui n'est pas propice à la bonne reproduction d'un signal musical !

**DM :** *Nous aurons l'occasion de revenir sur les solutions quand nous publierons l'article. Comment se présente l'interface ?*

**FXM :** Un bel écran LCD alphanumérique de 4 x 20 caractères et deux boutons poussoirs pour changer de mode.

**DM :** *J'ai remarqué que pour certains réglages vous utilisez les pads de la batterie elle-même, pour incrémenter ou décrémenter des paramètres par exemple, c'est astucieux. Et votre jeu du pendu, il doit être astucieux aussi celui-là ?*

**FXM :** La plus fine astuce, c'est peut-être d'avoir recyclé une étude plus ancienne. Ce projet n'était pas le sujet principal du stage, mais en plus on m'a tellement rebattu les oreilles au sujet du sacro-saint planning à respecter que j'ai cherché à faire court, sans possibilité d'approfondir...

**DM :** *... [se gratte la gorge, gêné]... Et le cahier de charges de ce jeu du pendu ?*

**FXM :** L'application doit illustrer l'utilisation des périphériques de la carte Linux sans trop consommer de courant. Le joueur doit deviner un mot en faisant le moins possible d'erreurs. Plusieurs niveaux

de difficulté sont possibles, le jeu doit pouvoir se jouer avec des listes de mots dans toutes les langues courantes. Pour la machine elle-même, c'est le langage C, compilé et exécuté sur la cible embarquée Linux !

**DM :** *Et pour l'affichage ?*

**FXM :** L'idée de départ était de dessiner en LED, sur une carte d'extension, la forme du pingouin Linux (*Tux*). Pour limiter la consommation, je me suis contenté d'un VU-mètre qui compte les erreurs, et j'ai essayé de dessiner mon pingouin en caractères ASCII pour un affichage via le terminal.

**DM :** *Et pour les différentes langues ?*

**FXM :** Des fichiers de type « francais.dict ». L'utilisateur peut créer un dictionnaire simplement en écrivant une liste de mots dans un fichier de ce type, séparés par des retours chariot, puis choisir n'importe quel dictionnaire placé dans le dossier *dictionary* à la racine du jeu.

**DM :** *À propos de langues, Elektor ressemble à une tour de Babel. Tout le monde y baragouine au moins deux, souvent trois, parfois quatre langues. Comment s'est passée la communication ?*

**FXM :** Un peu dur au début, je ne cache pas que j'avais quelques difficultés en anglais, mais j'étais là en partie pour m'améliorer. D'une manière générale, les échanges ont été très formateurs, grâce aussi à la langue. Ici, la langue habituelle est le néerlandais, mais tout le monde parle anglais, notamment lors des réunions du laboratoire où je devais exposer mes avancées sur mes projets. Le dialogue avec les rédacteurs pour la préparation des articles est également très instructif, car il s'agit de leur expliquer les principes de fonctionnement ou les problèmes techniques d'une manière compréhensible par quelqu'un d'externe au projet.

**DM :** *Quel est le bilan de votre stage ?*

**FXM :** Autant Linux embarqué que le PIC32 et les modules et protocoles utilisés tels que l'USB, l'I<sup>2</sup>S, le MIDI, étaient pour ainsi dire nouveaux pour moi. Ce qui a constitué à la fois une contrainte forte et une excellente occasion d'apprendre tout en faisant. Par rapport aux projets menés dans le cadre de ma formation, j'ai pu exploiter plus en profondeur les capacités de ce type de microcontrôleur.

**DM :** *Et par rapport à Elektor ?*

En fait, jamais je n'aurais pu imaginer il y a quelques mois, en lisant les magazines d'Elektor, y voir publiés certains de mes projets, ni même de travailler dans l'équipe du laboratoire. D'ailleurs, durant les dernières semaines, un nouveau stagiaire est arrivé et j'ai trouvé très enrichissant de pouvoir à mon tour répondre à quelques-unes de ses questions à propos de programmation de systèmes embarqués.

Au final, je garderai un très bon souvenir de ces trois mois dans les murs d'Elektor. Je remercie Wisse Hettinga qui a rendu ce stage possible ainsi que toute l'équipe pour son accueil, et plus particulièrement mes collègues du labo avec qui j'espère rester en contact.

**DM :** *Merci à vous pour cet entretien et à bientôt peut-être pour la publication de vos circuits dans Elektor.*

120609



# générateur AM-HF TF801D/1 de Marconi Instruments

Le TF801D/1 de l'anglais Marconi Instruments est un générateur HF (10 MHz à 470 MHz), produit aux environs de 1960, pour un prix relativement élevé à l'époque, ne serait-ce qu'à cause de la mécanique coûteuse.

Stephan Germann (Suisse)

J'ai obtenu ce générateur, il y a une quinzaine d'années, « par dessus le marché » pour l'achat d'un analyseur de spectre d'occasion. Depuis lors, il a fait son office dans mon atelier pour la construction ou la réparation d'émetteurs-récepteurs amateurs. Les caractéristiques électriques sont très bonnes, même selon les critères actuels. Le seul point faible serait la stabilité en fréquence, qui ne supporte pas la comparaison avec un générateur DDS (synthèse numérique directe), cela va de soi. Avec un peu de patience et de doigté, pourtant, on peut vérifier un récepteur BLU pour la bande des 2 m.

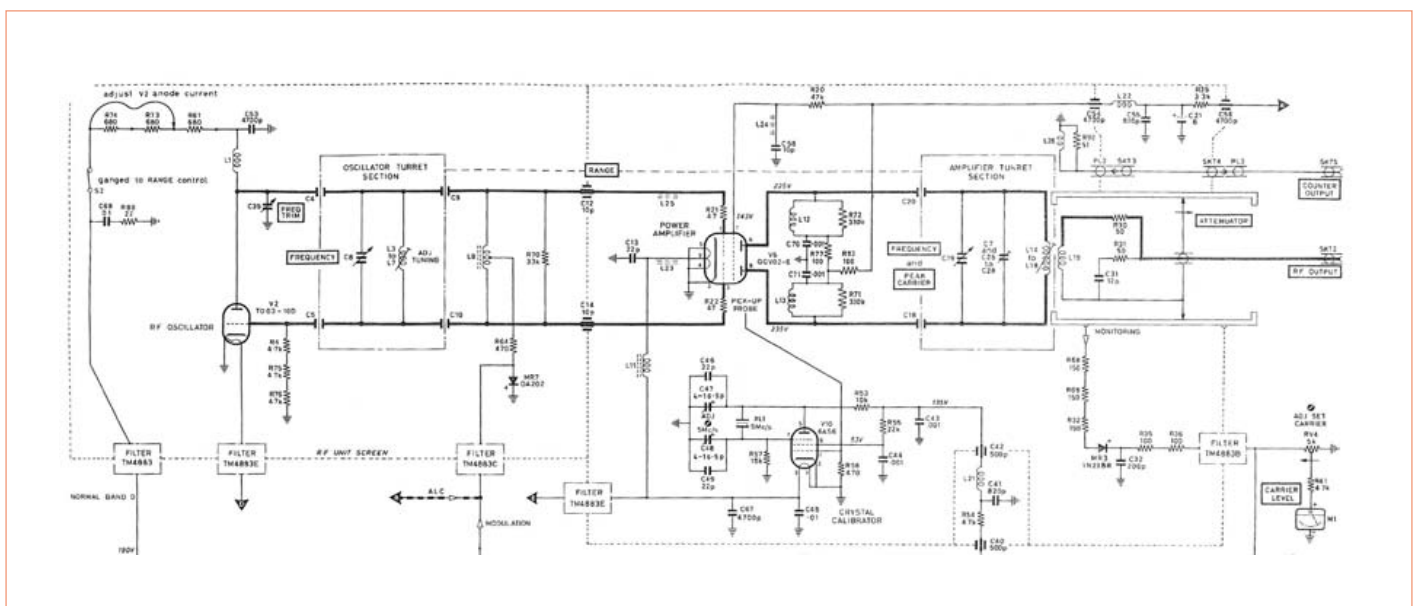
## Utilisation

J'utilise cet appareil principalement pour les mesures sur des postes de radio-amateurs, pour le dépannage, pour la mesure d'antennes, filtres, réseaux d'adaptation de construction maison. La plage de fréquences étendue sans trou couvre toutes les bandes amateurs de 30 m à 70 cm, en plus de toutes les FI courantes. La très grande plage de tensions de sortie de 100 nV à 1 V permet de tester aussi bien des récepteurs sensibles que des étages de puissance antenne ou des ponts de mesure d'impédance passifs. Grâce aux tubes, l'appareil est très robuste électriquement. La précision de la tension de sortie et l'absence totale de fuite HF n'en finissent pas de m'étonner. En plus, la manipulation est très confortable, et bien plus simple que celle des appareils actuels avec leurs menus et boutons à fonctions

multiples. Chaque fonction a son bouton et il n'y a pas de fonction inutile.

Exception faite d'un condensateur chimique défectueux, il n'y a pas eu de réparation jusqu'ici, ce qui montre l'excellente qualité des composants de l'époque. Ainsi, les potentiomètres sont de type bobiné et étanches. L'affichage de profondeur de modulation et de la tension HF se fait sur deux galvanomètres séparés. Les organes de commande sont largement espacés et les boutons d'une taille confortable. L'échelle de fréquence est grande et assez précise pour qu'on puisse obtenir une valeur avec un écart inférieur à 100 kHz, jusque dans la bande UHF, sans autre appareil de mesure. La transmission est très précise et il y a un réglage fin en plus.

La construction est extrêmement solide. Cela se reconnaît d'abord au poids, supérieur à 30 kg. Sur un appareil anglais, aucun filetage n'est en pas métrique : ne perdre aucune vis lors d'un démontage ! Le robuste châssis interne reçoit le modulateur, le circuit de commande et l'alimentation. La partie HF est logée dans une double enceinte en tôle de cuivre argentée, étanche à la HF. L'enveloppe externe ne contient que la mécanique d'accord et les filtres HF pour amener la tension de commande et les alimentations. Le circuit HF proprement dit se trouve dans l'enceinte interne, étanche à la HF, divisée en deux parties pour l'oscillateur et l'étage de sortie. Grâce à cette conception, la HF n'apparaît effectivement que sur les bornes de sortie. Elle est indétectable à tout autre endroit, ce qui est une belle réussite si on pense que l'oscillateur et l'étage de sortie délivrent une puissance de quelques centaines de milliwatts.





Le circuit lui-même, modeste, comporte dix tubes, quelques diodes à semi-conducteurs et un transistor au germanium. L'alimentation comporte un transformateur, un redresseur double alternance à tubes, une stabilisation par tube à gaz, des tubes de régulation et un tube de puissance audio comme organe de réglage. La tension de chauffage du tube oscillateur est stabilisée par un amplificateur magnétique.

Le signal HF est produit « en direct », c'est-à-dire que l'oscillateur oscille à la fréquence désirée et que la double tétrode qui suit sert de tampon HF.

L'oscillateur est constitué d'une triode UHF et de cinq bobines commutables pour les cinq bandes de fréquence. L'accord se fait par condensateur variable. Curiosité : l'anode de la triode est munie d'un radiateur.

Le signal de l'oscillateur attaque directement l'étage *push-pull* symétrique de sortie, équipé d'une double tétrode UHF. Ce tube fonctionne en classe A : maximum de gain et minimum de distor-

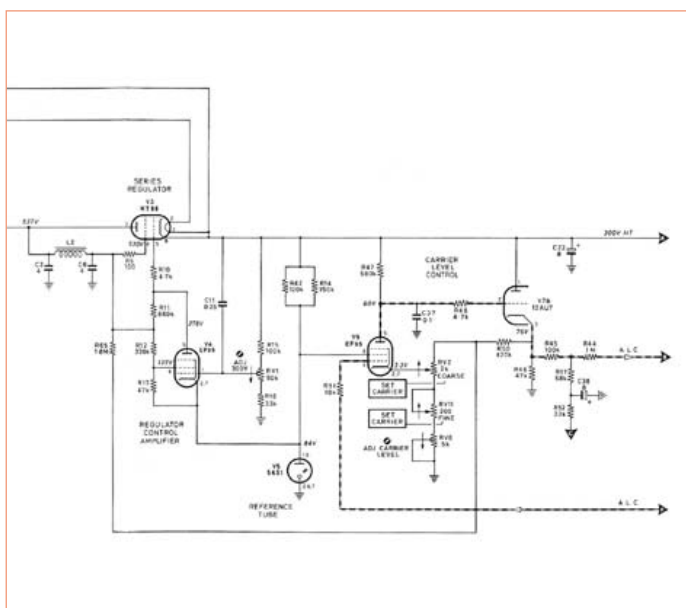
sion. Le niveau continu des grilles de commande règle le niveau de sortie, par la commande du gain, et permet la modulation d'amplitude en cas de besoin. La régulation du gain maintient la tension de sortie indépendante de la charge et de la fréquence, ce qui est très pratique pour la mesure et l'accord de filtres.

L'étage de sortie comporte aussi, dans le circuit d'anode, cinq bobines commutables. Avec un autre condensateur variable, elles constituent un circuit oscillant réglable, accordé avec l'oscillateur. Pour corriger les écarts, on peut décaler un peu le réglage du condensateur du circuit de sortie par rapport à celui de l'oscillateur ; une conception pas vraiment simple mécaniquement. Grâce à l'amplification linéaire de l'étage de sortie et au circuit accordé de sortie, il n'y a quasiment pas d'harmoniques. Le signal est transmis par couplage inductif à travers l'atténuateur, sur une plage unique calibrée de 135 dB, avec un réseau de résistances qui présente une impédance de sortie de 50 Ω, indépendante de la fréquence.

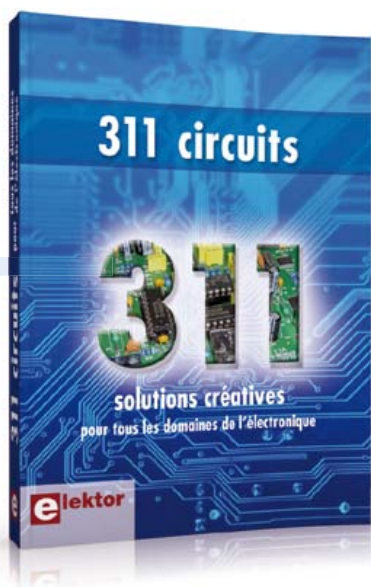
Le commutateur des cinq bandes de fréquence mérite d'être examiné. Contrairement à ce qu'on attend, il ne suffit pas de le tourner d'une graduation à l'autre pour passer à la bande suivante, mais qu'il faut effectuer une révolution complète en plus. Le commutateur lui-même fonctionne sans contact, la connexion avec les bobines interchangeable est capacitive. Cela présente l'avantage d'éliminer les problèmes de contact, ce qui est très important pour le facteur de qualité du circuit oscillant.

En plus du mode d'emploi, le manuel décrit l'entretien et les procédures de réparation éventuelles, autant mécaniques (remplacement des câbles) qu'électriques. Appréciez ce truc de dépannage : « Cherchez d'abord des résistances carbonisées pour localiser la partie défectueuse du circuit. »

(120169 – version française : Jean-Paul Brodier)



Rétronique est une rubrique mensuelle sur les pages glorieuses et jaunies de l'électronique, avec aussi des montages de légende décrits dans Elektor. Si vous avez des suggestions de sujets à traiter, merci de les adresser à [redaction@elektor.fr](mailto:redaction@elektor.fr) en indiquant : Rétronique Elektor.



### Solutions créatives pour tous les domaines de l'électronique

#### 311 circuits

Cet ouvrage est un trésor : il réunit 311 schémas d'électronique analogique, logique ou numérique, des programmes, des liens vers des sites internet, des tableaux de caractéristiques de composants et des dessins de circuit imprimé. Il est le onzième volume de la collection « 300 circuits » (301... 302... 303... 304... 305... 306... 307... 308... 309... 310... 311 circuits). Ses deux tables des matières alphabétique et thématique vous permettent de trouver rapidement et facilement parmi les 311 articles proposés ceux qui répondront à vos besoins. Ces articles viennent des numéros doubles récents de la revue Elektor, publiés chaque année en été, et appelés numéros Hors-Gabarit, par allusion à leur contenu exceptionnellement riche. Ils forment un véritable catalogue d'idées, de trouvailles et d'astuces. C'est une source d'inspiration inépuisable, et à partir de laquelle chacun élaborera ses propres variantes qu'il combinera ensuite à sa guise avec d'autres circuits. Tous les domaines familiers et usuels de l'électronique sont abordés : alimentations, régulateurs et chargeurs • audio & vidéo • communication • hautes fréquences • informatique • jeux & modélisme • maison & automobile • mesure & test • processeur & contrôleur • robots et leurs accessoires.

448 pages • ISBN 978-2-86661-184-2 • 36,00 €



### Rémy Mallard présente

#### L'électronique pour les débutants

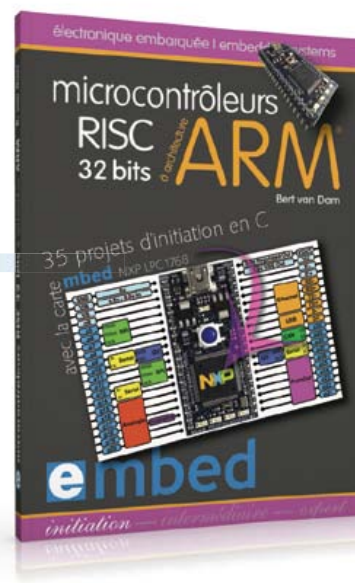
Par où commencer pour débiter en électronique ? Vais-je m'égarer en explorant l'internet, qui regorge de schémas, mais sont-ils fiables ? Me faut-il un livre avec des montages simples ou plutôt un livre sur les composants ? Après trente ans de pratique, l'auteur de ce livre, resté l'éternel débutant qui réalisait lui-même son premier montage dès l'âge de dix ans, partage ici sa soif toujours vive d'apprendre. Fin pédagogue, il guide les débutants et répond aux questions que trop de livres laissent en suspens : « Quel type de fer à souder acheter ? »... « Un multimètre à 5 € peut-il suffire ? »... « Un oscilloscope est-il indispensable ? ».

352 pages • ISBN 978-2-86661-180-4 • 42,50 €

#### 35 projets d'initiation en C avec la carte mbed NXP LPC 17688

#### Microcontrôleurs RISC 32 bits à architecture ARM

La plate-forme mbed et son microcontrôleur ARM, le NXP LPC1768, sont conçus pour l'informatique en nuage ou cloud computing qui révolutionne le développement de logiciel : aucune installation de logiciel spécifique, il suffit d'un navigateur et d'un port USB. Vous programmerez et stockerez vos résultats



sur l'internet, et y accédez depuis n'importe quel PC, où que vous vous trouviez. Dans ce livre, il est question aussi bien du langage C, des bibliothèques mbed, d'exemples de programmes que du traitement de signaux analogiques, de capteurs, de moteurs etc.

232 pages • ISBN 978-2-86661-178-1 • 40,00 €

### sécurité – confort – économies

#### Domotique

La domotique, c'est l'électronique et l'informatique appliquées au logement. Elle améliore la vie quotidienne au moyen de dispositifs électriques et électroniques. Il peut s'agir aussi bien de motoriser et télécommander par exemple la porte du garage ou bien les volets, que réguler le chauffage, programmer les appareils électroménagers, simuler une présence, commander l'alarme à distance, arroser automatiquement, exploiter un réseau multimédia, etc. Ce livre montre comment la domotique gère le fonctionnement des appareils et dispositifs électriques de la maison.

256 pages • ISBN 978-2-86661-182-8 • 33,00 €

#### Une carte compacte et bon marché qui vous initiera tout en douceur !

#### Embarquez Linux !

Linux est partout, même dans certaines machines à café. Souvent, l'électronicien tenté d'adopter ce





système d'exploitation est arrêté par sa complexité et par le prix des cartes de développement. Voici Linux pour les électroniciens, sous la forme d'une carte compacte et bon marché qui vous initiera tout en douceur !

Carte Linux Elektor (montée et testée)  
Réf. : 120026-91 • 64,95 €

**Tous les articles de 2011 sur DVD-ROM**  
**DVD Elektor 2011**

Ce DVD-ROM réunit tous les articles d'ELEKTOR, le mensuel d'électronique et de micro-informatique appliquées, parus au cours de l'année 2011. Il contient non seulement le texte des articles ainsi que les schémas, mais aussi tous les dessins des circuits imprimés, sous forme de fichiers à haute résolution. Ceci permet à l'utilisateur de modifier à sa guise les dessins existants à l'aide d'un programme adéquat. Dès lors, rien ne s'oppose plus à l'exportation des documents vers un autre format à la convenance de l'utilisateur.

ISBN 978-90-5381-276-1 • 27,50 €

**Initiation et maîtrise par l'expérimentation**

**50 applications des microcontrôleurs PIC**

Voici 50 projets instructifs et utiles pour vous initier



au langage de programmation JAL et maîtriser les microcontrôleurs PIC16 et 18, avec des techniques universelles comme la commande de relais, ou le traitement des signaux émis par divers capteurs (y compris par exemple un codeur rotatif), la communication avec les bus I<sup>2</sup>C, SPI, RS232, USB, les afficheurs à 7 segments et même le bus CAN. C'est un ouvrage récréatif et pédagogique : assemblez et utilisez les projets proposés. Les explications claires, les schémas et les photographies vous feront découvrir une activité enrichissante.

394 pages • ISBN 978-2-86661-177-4 • 45,00 €

**23 projets ludiques et instructifs à construire soi-même**

**Intelligence artificielle**

Ce livre ne traite pas de théories abstraites, mais de pratique. Il s'adresse à vous, passionné d'électronique et de micro-informatique appliquée, et propose des circuits et des programmes simples. Expérimentez le biomimétisme sur vos propres robots mobiles, construits avec des pièces de Lego ! Il y est donc question de microcontrôleurs PIC, de programmation, de capteurs, de moteurs... mais aussi de morpions, de fourmis, de gnous, de vers, et d'autres bestioles « naturellement géniales ».

238 pages • ISBN 978-2-86661-179-8 • 43,50 €



**Kit (circuits imprimés & composants)**  
**TAPIR – Détecteur ultrasensible d'électromag**

Fin limier de la pollution électromagnétique qu'il traque et rend audible sur casque, le TAPIR est (aussi) un beau projet à construire, un kit avec tout ce qu'il faut, même le boîtier, qui est fait de quatre circuits imprimés ingénieusement assemblés. Le TAPIR, sigle de Totally Archaic but Practical Interceptor of Radiation (totalement archaïque mais pratique intercepteur de radiations), détecte et localise (en la faisant entendre) toute source de champ électrique E ou, avec l'antenne appropriée, de champ magnétique H.

Réf. : 120354-71 • 14,95 €

**Informations complémentaires et gamme complète sur :**

[www.elektor.fr/e-choppe](http://www.elektor.fr/e-choppe)

Elektor/Publitronic S.A.R.L.  
1, rue de la Haye  
BP 12910 – 95731 Roissy CDG Cedex  
Tél. : +33(0)1.49.19.26.19  
Fax : +33(0)1.49.19.22.37  
@ : ventes@elektor.fr



# Hexadoku

## Casse-tête pour électroniciens

Que vous soyez un habitué ou un joueur occasionnel, une grille hexadoku mensuelle ne devrait vous faire que du bien. Ça dégrasse les méninges... si on en a. C'est tout le mal que nous vous souhaitons. Remplissez la grille selon les règles, envoyez-nous votre solution, et vous gagnerez peut-être l'un des quatre chèques-cadeaux Elektor mis en jeu. Après ça, vous serez d'aplomb pour reprendre vos cogitations électroniques.

Les instructions de ce jeu sont simples. Une grille Hexadoku est composée de chiffres du système hexadécimal, de 0 à F. Du tout cuit pour les électroniciens et les programmeurs ! 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. Si vous trouvez la solution de ce casse-tête, vous pouvez gagner un chèque-cadeau. Inutile de nous envoyer toute la grille, il suffit de nous envoyer la **série de chiffres** sur fond grisé.

### Participez et gagnez !

Nous tirerons au sort l'une des réponses internationales correctes qui nous seront parvenues dans les délais ; son auteur recevra un chèque-cadeau Elektor d'une valeur de 100 €. Nous offrons en outre 3 chèques-cadeaux Elektor d'une valeur de 50 € chacun.

À vos crayons !

### Où envoyer ?

Envoyez votre réponse (les chiffres sur fond grisé) avec vos coordonnées par courriel, télécopie ou courrier avant le 1<sup>er</sup> décembre 2012 à :

Elektor c/o Regus Roissy CDG – Le Dôme – 1, rue de la Haye

BP 12910 – 95731 Roissy CDG

Courriel : hexadoku@elektor.fr

### Les gagnants

La solution de la grille de septembre 411 est **3F126**

Le gagnant du **chèque-cadeau Elektor** d'une valeur de **100 €** est : William Neumann, de Leduc, Canada.

Les **3 chèques-cadeaux Elektor** d'une valeur de **50 €** chacun vont à : Yves Printems, de Valbonne (France),

Jean-Louis Vidaud, de Clamart (France) et Eugene Stemple, de Willcox (États-Unis d'Amérique).

Bravo à tous et félicitations aux gagnants !

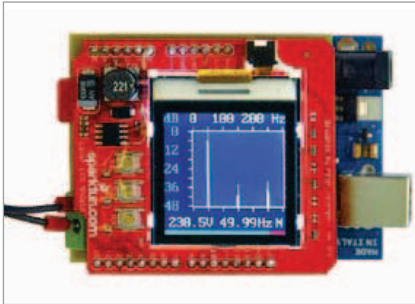
	E		A	6		9	D		4		2	7	8	C
1		2	9			B	7	C			8		3	
	B					2			3			A		
5	8		C			3			0	A	9			1
							7			4				5
B				4	5						7			8
	5	7	1		C				D	0		9		
2	6						B			C	E		D	F
6	7			8				2		E	F	C	3	
		4	D					8		3			2	7
3					4								6	B
	C		5	0		6	2	4	7					E
A		0	F		D		6	3						
9	1					C		A		2			F	B
8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

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

Tout recours est exclu de même que le sont, de ce jeu, les personnels d'Elektor International Media B.V. et leur famille. Un seul gagnant par foyer.







### analyseur de tension du réseau électrique à base d'Arduino

Les applications d'Arduino se suivent mais ne se ressemblent pas toutes. En voici une qui surprendra ceux qui considèrent cette carte avec condescendance. Ce projet vous montre comment tirer parti d'un microcontrôleur à 8 bits et met aussi en lumière ses limites. Ce n'est ni plus ni moins qu'un analyseur de spectre qui donne une image détaillée de la tension fournie par votre réseau local. Une transformée de Fourier en décompose le spectre harmonique et l'affiche. Au passage l'auteur montre bien comment le microcontrôleur est mis à contribution, de même que les outils de programmation (gratuits) nécessaires pour mener à bien un tel projet.



### interface USB-I/O universelle

Tous les appareils ont maintenant leur prise USB (même certains réveils). À tel point que la plupart des ordinateurs n'ont plus d'autre interface que ça. Ne serait-il pas pratique qu'une interface aussi universelle permette aussi de commander et de commuter toutes sortes de fonctions simples ? C'est ce que fait le fameux câble USB-TTL conçu par FTDI auquel est associé ici un petit circuit imprimé qui tient dans un connecteur DB25, avec un microcontrôleur R8C. Ce dernier assure d'un côté la communication avec le PC et commande de l'autre pas moins de 24 broches d'entrée-sortie que l'utilisateur configurera à sa guise pour toutes les applications possibles et imaginables.



### réveil multifonctionnel

Êtes-vous vraiment satisfait du réveil-matin numérique qui le matin vous tire de votre sommeil ? Faute des grives dont nous rêvons, nous sommes légion à nous contenter de merles qui ne nous procurent guère de satisfaction réelle. Il était temps que quelqu'un s'attelle à l'étude d'un nouvel appareil, avec toutes les fonctions souhaitables, jamais réunies sur un même modèle à notre connaissance. Ce réveil multifonctionnel est construit sur un microcontrôleur d'Atmel (AT89C5131). Vous retrouverez bien sûr les fonctions classiques, mais d'autres aussi que vous n'avez jamais vues. Cet appareil a même une connexion USB, pour communiquer, avec un PC, bien sûr !

*Sous réserve de modifications. Le numéro de décembre paraîtra le 20 novembre*

**Pour vous abonner : [www.elektor.fr/abo](http://www.elektor.fr/abo), c'est plus rapide et moins cher.**

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

## Elektor en ligne

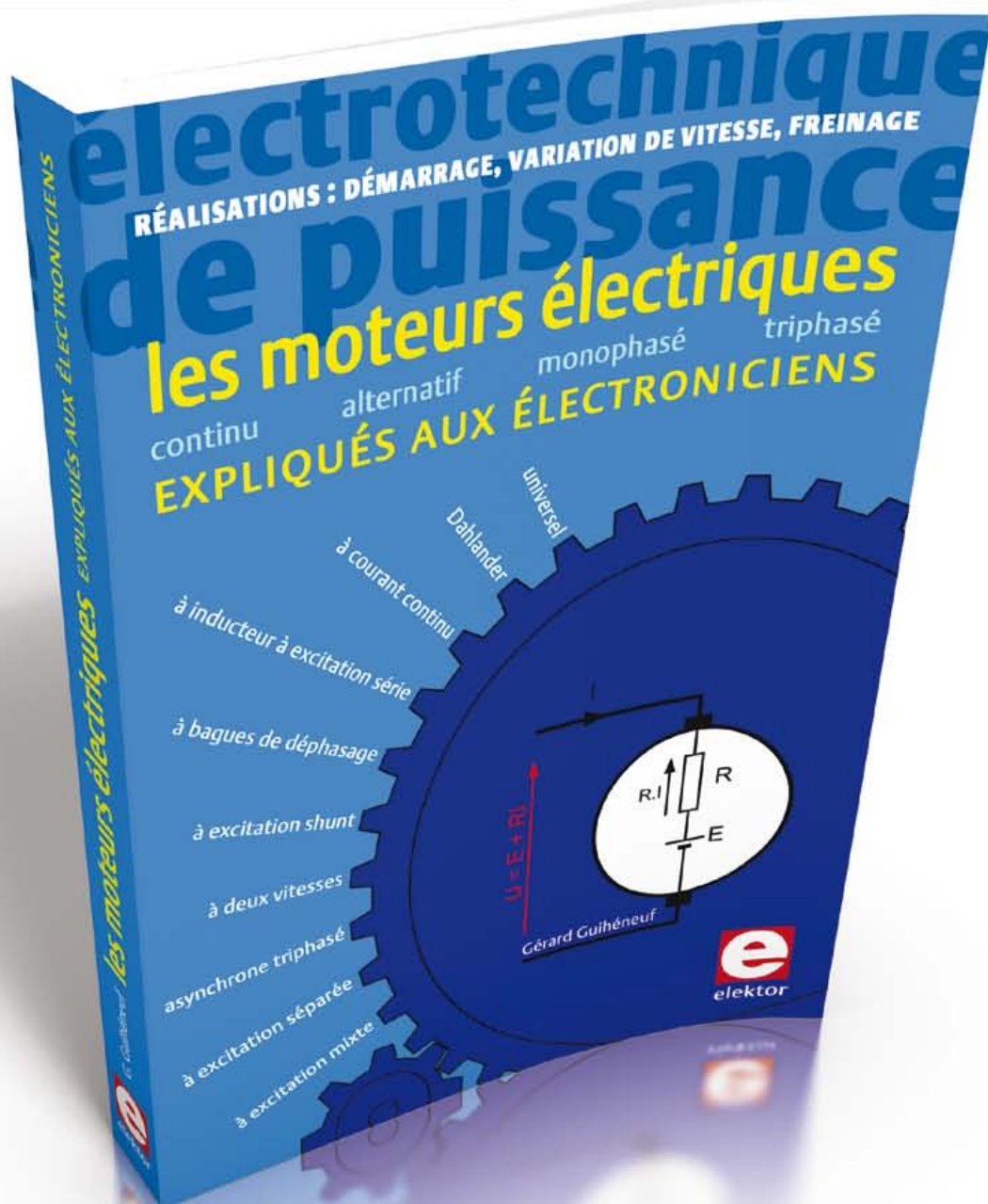
Sur le site d'Elektor, vous trouvez tous les articles publiés depuis 2000, sous forme de fichiers PDF téléchargeables individuellement, certains gratuitement, d'autres moyennant un modeste paiement forfaitaire. Un résumé de l'article donne une idée du contenu avant de le télécharger, de même que la liste des composants (le cas échéant). Le site propose également les autres ressources liées à chaque article : code source, liens, circuits imprimés, et les corrections ainsi que les mises à jour s'il y en a. L'e-shoppe d'Elektor propose de nombreux produits : CD-ROM, DVD, kits, modules assemblés, appareils & instruments, E-blocks, livres. Sans oublier l'indispensable et puissante fonction de recherche.

### Également sur le site d'Elektor :

- des nouvelles sur le monde de l'électronique
- un forum des lecteurs
- téléchargement de logiciel et de circuits imprimés
- des offres temporaires très avantageuses
- Foire Aux Questions



« Comme son titre l'annonce, les électroniciens découvriront dans ce livre le monde des moteurs électriques. Les électrotechniciens qui le liront seront surpris de découvrir qu'électronique n'est pas obligatoirement synonyme de complexité. »



Ce livre en trois chapitres offre une information accessible et digeste : constitution, fonctionnement, caractéristiques, domaines d'utilisation, pour proposer aussi des réalisations électroniques simples et concrètes.

Le premier détaille les principes de variation de la vitesse des moteurs à courant continu : conversion alt./continu (redresseurs commandés par des thyristors) et conversion CC (hacheurs à transistors IGBT). Mise en pratique avec des variateurs pour mini-perceuse et pour train miniature.

Les moteurs à alimentation alternative monophasée (à induction, à bagues de déphasage, universel) du 2<sup>e</sup> chap. font appel à une électronique de puissance. La puissance des réalisations proposées s'exprime en kW : démarreur à contacteur statique à deux points de commande pour moteur asynchrone monophasé à induction et variateur de vitesse pour moteur universel.

Reste le moteur électrique le plus utilisé dans l'industrie : le moteur asynchrone triphasé et ses différents principes de démarrage, de variation de vitesse et de freinage : démarreur électromécanique à contacteurs, démarreur-ralentisseur, convertisseur de fréquence ou couplage des pôles pour la variation de vitesse, moteur frein, freinage par injection de courant... Construisez le démarreur inverseur statique pour moteur asynchrone triphasé et découvrez la proximité entre électronique et électrotechnique : des portes NON-OU alimentées sous 12V commandent le sens de rotation d'un moteur de 1,5 kW alimenté sous 400 V en triphasé.

les moteurs électriques expliqués aux électroniciens - Gérard Guihéneuf

isbn 978-2-86661-188-0 | 320 pages | 37,50 €

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

nouveau livre



# WIRELESS POWER : CHARGING INNOVATION



L'intégration de l'énergie sans fil en toute simplicité, avec les modules d'évaluation et l'éventail de solutions bqTESLA de Texas Instruments et Würth. Ces solutions permettent aux ingénieurs de conception de facilement accélérer l'intégration des technologies de l'énergie sans fil dans l'électronique grand public, tels que les téléphones intelligents, les appareils photo numériques, lecteurs MP3, et les applications d'infrastructure tels que les meubles et les voitures. Nous vous aiderons à franchir la ligne d'arrivée en un temps record de conception avec

une large gamme de modules d'évaluation à la fois sur l'émetteur et récepteur pour vous aider à réduire le cycle de conception de solutions d'alimentation sans fil. Que ce soit la mise en œuvre d'alimentation sans fil au sein d'une conception existante, ou l'ajouter à un nouveau design, nous avons les outils, le soutien et l'expertise pour vous aider – coupez le cordon! Faites votre propre kit en sélectionnant un émetteur/récepteur TI et le module Charging-Coils correspondant proposé par Würth Elektronik.



element14

[element14.com/wireless-power-solution](http://element14.com/wireless-power-solution)