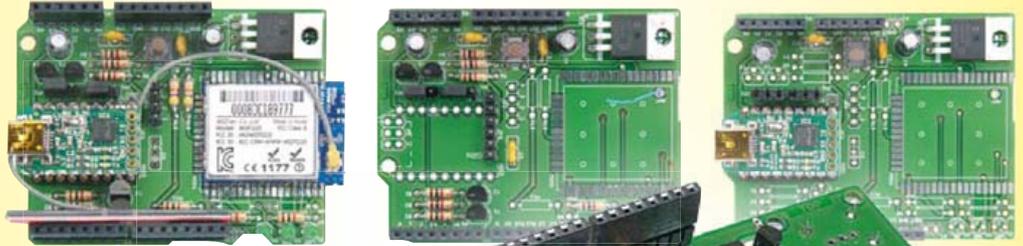


# e lektor

www.ektor.fr



## Arduino et Platino dans le nuages

Wi-Fi/Bluetooth/USB pour s'envoyer en l'air



- +** Linux embarqué autour du noyau
- +** initiation à l'électronique le préamplificateur BF
- +** AndroCAR véhicule piloté par un téléphone Android

## détecteur de rayonnement revisité

petites modifications grandes améliorations



L 19624 -412- F: 7,20 €



**elektor sur papier**

le plaisir de la lecture classique



**elektor à l'écran**

le plaisir de la lecture sur les nouveaux supports : PC, portable ou tablette



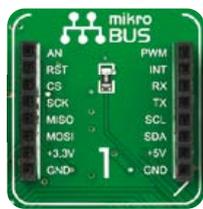
**elektor PLUS**

le plaisir ultime de lire partout

**Profitez plus de votre magazine grâce à la formule avantageuse elektor PLUS !**

**Abonnez-vous ou changez de formule maintenant : [www.elektor.fr/abo](http://www.elektor.fr/abo)**

# un SUPPORT pour 30 cartes click



Il vous suffit de placer la carte Click™ dont vous avez besoin dans le nouveau support mikroBUS™ innovant

et elle est immédiatement fonctionnelle. Jamais encore il n'a été aussi facile d'ajouter une nouvelle fonctionnalité à votre système.



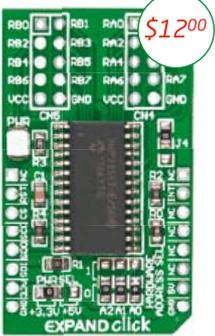
\$49<sup>00</sup>



\$18<sup>00</sup>



\$17<sup>90</sup>



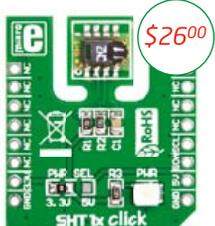
\$12<sup>00</sup>



\$10<sup>00</sup>



\$16<sup>90</sup>



\$26<sup>00</sup>



\$25<sup>90</sup>



\$9<sup>00</sup>



\$16<sup>00</sup>



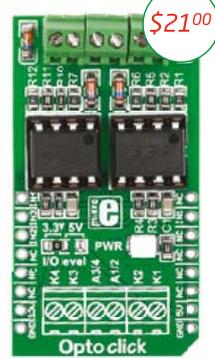
\$45<sup>00</sup>



\$21<sup>00</sup>



\$41<sup>90</sup>



\$21<sup>00</sup>



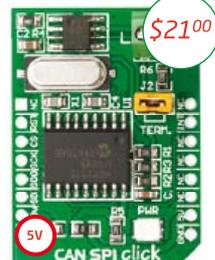
\$29<sup>00</sup>



\$21<sup>90</sup>



\$8<sup>00</sup>



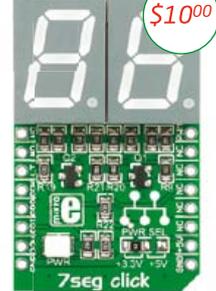
\$21<sup>00</sup>



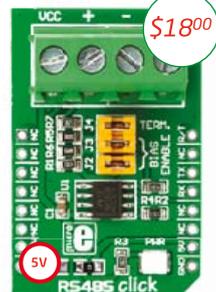
\$29<sup>00</sup>



\$13<sup>90</sup>



\$10<sup>00</sup>



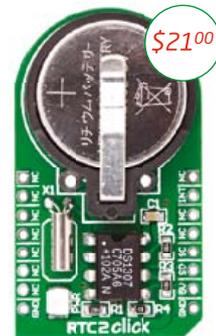
\$18<sup>00</sup>



\$13<sup>90</sup>



\$18<sup>00</sup>



\$21<sup>00</sup>



\$44<sup>00</sup>



\$19<sup>00</sup>



\$24<sup>00</sup>



\$21<sup>00</sup>



\$24<sup>00</sup>

## La tension monte

Ce qui frappe dans la communication avec nos abonnés, c'est leur fidélité. « *J'ai tous les numéros d'Elektor* » est une formule magique, souvent lancée par nos interlocuteurs, à laquelle nous ne sommes pas indifférents. Nous en tirons même fierté. En dépit des différences linguistiques, nous formons avec nos lecteurs du monde entier une communauté unique. Une communauté qui date de bien avant que ce mot fasse son apparition sur l'internet. Cependant, en pratique, cette communauté se résumait *grosso modo* jusqu'ici à l'envoi d'onze numéros par an (plus un DVD-ROM annuel) et d'un bulletin de réabonnement suivi de votre chèque. Rien que l'on puisse qualifier de relation intime.

Le magazine imprimé joue forcément un rôle clé, mais les membres d'une communauté moderne ne sont-ils pas en droit d'attendre davantage ? C'est pourquoi nous avons lancé notre grand labo virtuel, le site [www.elektor-projects.com](http://www.elektor-projects.com).

Ce qui revient à ouvrir grand les portes de la cuisine, pour encourager la créativité des électroniciens et les inviter à participer à l'élaboration des menus.

Pour ElektorHebdo, notre lettre d'info gratuite, nous concoctons aussi des améliorations, toutes commandées par cet appel : **Get Elektorized!** c'est-à-dire **elektorisez-vous**, pour plus d'interaction, plus de projets, plus de découvertes, davantage d'avantages aux membres de la communauté. Bientôt notre présence en ligne sera renforcée, les rédactions travaillent déjà sur une nouvelle formule du magazine. Les publications numériques seront développées, mais le papier ne disparaîtra pas.

Avec de tels projets, comme dans tout circuit, quand les charges augmentent, les différences de potentiel en font autant : la tension monte chez Elektor comme aux bornes d'un gigantesque condensateur, et parfois ça barde : des arcs électriques se forment ici ou là. Au labo notamment, qui reste au cœur de notre modèle unique au monde.

Bientôt, de simple abonné que vous êtes, vous deviendrez membre d'une communauté.

Denis Meyer

PS : Auriez-vous d'ailleurs une idée de nom à donner à cette communauté ? Et par quoi remplacer le mot *membre* ?

# elektor

### 6 de nous à vous : le réseau Elektor

Informations pratiques & légales

### 8 infos & actualités

Nouveaux produits & initiatives

### 12 retour aux sources (8)

*lire ci-contre*

### 16 détecteur de rayonnement revisité

Quelques astucieuses modifications matérielles et logicielles permettent d'améliorer le détecteur de rayonnement d'Elektor que vous êtes (fort) nombreux à utiliser.

### 22 Arduino et Platino dans les nuages

*lire ci-contre*

### 28 mettez le cap sur Arduino

La communication par GSM / GPRS rend possible de nouveaux concepts tels que l'IdO (pour internet des objets) ou le M2M (pour machine à machine).

### 34 accéléromètre DIPLômé

Accessoire d'adaptation pour puce d'accéléromètre à une carte de prototypage ordinaire.

### 36 compteur horaire

Pour connaître le temps pendant lequel un appareil électrique a été en marche et a consommé du courant du secteur.

### 40 transformateurs de courant

Faites les vôtres avec des composants faciles à trouver. À vous les mesures de courant en toute sécurité et avec isolation.

### 43 les masses du bus RS-485

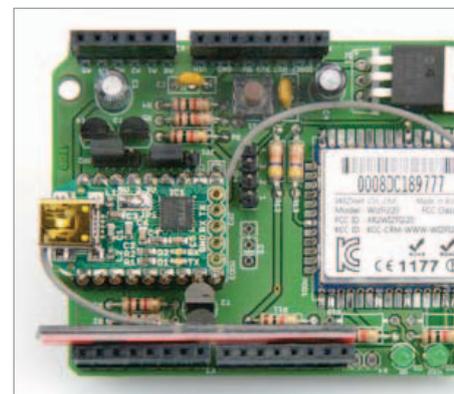
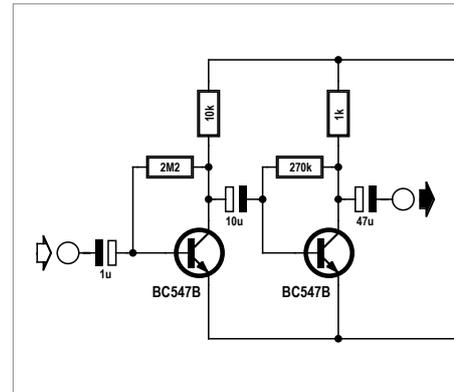
Sur un bus, le potentiel des points de masse est censé être le même partout et nul de préférence.

### 44 Parkinson touche aussi les AVR

Connaissez-vous le syndrome de l'AVR (aussi appelé tremblante du bouton) ?

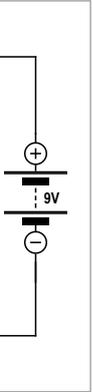
### 44 e-pollution !

Black-out total sur les signaux radio horaires au labo d'Elektor



# SOMMAIRE

35<sup>e</sup> année  
octobre 2012  
n° 412



## 12 retour aux sources (8) le préamplificateur BF

L'un des principaux domaines d'emploi du transistor est l'amplification de signaux sonores dits signaux basses fréquences (BF). Peu importe que le son provienne d'un microphone, d'un fichier mp3 ou d'un circuit radio, il faut lui donner du volume, avec un amplificateur que l'on souhaite aussi linéaire que possible !



## 22 Arduino et Platino dans les nuages Wifi/Bluetooth/USB pour mieux s'envoyer en l'air

La carte d'extension décrite permet de relier tout et n'importe quoi, avec ou sans fil. C'est peut-être un peu exagéré, mais cette carte est belle et bien polyvalente. Équipée d'un module Wifi, elle permettra de connecter un appareil à l'internet. Dotée d'un module Bluetooth, elle discutera avec ses congénères. Et pour les inconditionnels du câble, il y a un pont série vers USB.

## 62 embarquez Linux ! (4) Autour du noyau

Le système d'exploitation GNU/Linux fournit toutes les interfaces pour périphériques et applications nécessaires aux ordinateurs ou serveurs modernes. Le monde de l'embarqué n'est pas en reste : que ce soit dans le domaine des réseaux, de l'USB ou encore de Bluetooth, le programmeur qui adopte Linux s'épargnera les coûteux développements de ses propres solutions en C, un avantage linuxien parmi bien d'autres !



## 72 AndroCAR véhicule piloté par l'inclinaison de votre téléphone

Pilotez un véhicule à distance et sans fil en inclinant votre téléphone Android. Celui-ci contient des accéléromètres et un module WiFi qu'il est très facile d'exploiter.



45  **votre composant préféré**  
myfavoritepart@elektor.com est l'adresse à laquelle Elektor vous convie à adresser votre témoignage sur votre composant préféré.

46  **signaux carrés avec MLI et pavé tactile**  
Impressionnante gamme de fréquences de 250 Hz à 60 MHz.

50  **radio logicielle avec AVR (6)**  
Ce 6<sup>e</sup> épisode confirme que les microcontrôleurs AVR conviennent au traitement numérique du signal.

56  **pratique de la commande des moteurs pas-à-pas (3)**  
Cette *histoire de couples* décrit un dynamomètre simple qui vous permettra de caractériser le comportement de moteurs pas-à-pas.

62  **embarquez Linux ! (4)**  
En orbite autour du noyau.

72  **AndroCAR véhicule piloté par l'inclinaison de votre téléphone**  
*lire ci-contre*

78  **Rétronique : Pansanitor (1928)**  
Stimulation électrique musculaire d'un autre temps.

84  **Hexadoku**  
Jouez, participez et gagnez !

85  **récup' d'énergie**  
Comment alimenter un capteur dont l'éloignement est trop grand pour une ligne d'alimentation directe ? Et si on récupérait l'énergie sur place pour la fournir au capteur, juste le temps d'une mesure ?

86  **avant-première**  
Pendant que vous lisez ce numéro, nous préparons les suivants. Vos idées, vos suggestions, vos schémas et vos circuits 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°412 octobre 2012  
ISSN 0181-7450 Dépôt légal : septembre 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

274932

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> . . . . . 11		MikroElektronika <a href="http://www.mikroe.com">www.mikroe.com</a> . . . . . 3
	CES <a href="http://www.cesweb.org">www.cesweb.org</a> . . . . . 33		National Instruments <a href="http://www.ni.com/academic/f">www.ni.com/academic/f</a> . . . . . 88
	Eurocircuits <a href="http://www.elektorpcbservice.com">www.elektorpcbservice.com</a> . . . . . 81		Reichelt <a href="http://www.reichelt.fr">www.reichelt.fr</a> . . . . . 21
	Jackaltac <a href="http://www.jackaltac.com">www.jackaltac.com</a> . . . . . 41		Schaeffer AG <a href="http://www.schaeffer-ag.de">www.schaeffer-ag.de</a> . . . . . 35

### 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

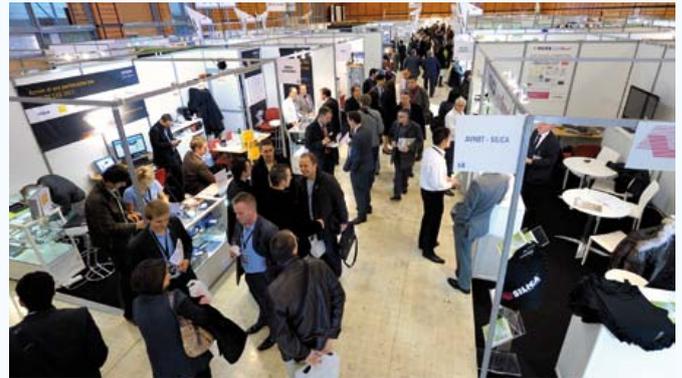
Distribué en France par M.L.P. et en Belgique par A.M.P.

**ForumLED Europe 2012 – 21 et 22 novembre 2012 – Paris, Grande Halle de la Villette**



**1 salon + 1 congrès international sur la technologie LED**

En seulement trois éditions, le **Congrès/Expo ForumLED Europe** s'est imposé comme l'événement n° 1 dédié à la technologie LED en Europe. L'objectif de cet événement est de réunir des experts et des professionnels du monde entier et de comprendre les tendances des marchés des LED, de découvrir les nouveaux secteurs d'applications des LED et de bénéficier des dernières informations technologiques et marketing. En associant une exposition et des conférences de haut niveau, *ForumLED Europe* a su convaincre un public international (Asie, Europe, Amérique du Nord, Afrique...) à la recherche des dernières informations technologiques, de solutions innovantes et des futures applications des LED.



**2 jours de conférences**

Ces conférences offrent aux acteurs du marché l'occasion d'écouter les meilleurs spécialistes internationaux faire l'état des lieux du marché, des évolutions réglementaires et des avancées technologiques de la LED. Les interventions alterneront comptes-rendus de travaux de recherche scientifique et retours d'expérience dans tous les domaines qui font polémique. En vue d'adapter les LED aux défis futurs, les experts mèneront des conférences de haut niveau afin d'apporter les réponses aux questions relatives aux avancées actuelles et à venir. Différents conférenciers issus d'institutions publiques ou à la tête de sociétés en pointe dans ce secteur sont invités à présenter leur point de vue et leurs connaissances dans ce domaine.

**2 espaces d'exposition**

Avec près de 120 exposants de qualité, ForumLED 2012 Europe sera une occasion unique à la fois de découvrir des applications et des solutions techniques nouvelles et de lier des contacts avec de futurs partenaires. Pour répondre à la demande des professionnels, fabricants et utilisateurs, ForumLED Europe innove avec deux espaces d'exposition complémentaires qui rassemblent plus de 120 sociétés.

**ForumLED Technologies : technologies et services de l'industrie de la LED**

Fabricants et distributeurs de :

- LED / OLED
- Modules LED
- Lampes et tubes LED
- Composants (optiques, électroniques, thermiques...)
- Outils de production pour l'industrie de la LED
- Systèmes d'alimentation, de pilotage et de contrôle commande
- Équipement de tests et mesures
- Circuits imprimés et de cartes électroniques

- Bureaux d'études technique et industrielles
- Concepteurs et distributeurs de solutions informatiques pour l'industrie de la LED
- Laboratoires et organismes de qualification et de certification

**ForumLED Lighting Products : NOUVEAUTE 2012 : systèmes et matériels d'éclairage à LED**

- Fabricants et distributeurs d'éclairage extérieur LED (Éclairage public fonctionnel – public d'ambiance – architectural)
- Fabricants et distributeurs d'éclairage intérieur LED (Éclairage de commerces et CHR – industriel – tertiaire/établissements de santé – résidentiel – de sécurité)
- Fabricants et distributeurs d'éclairage événementiel et scénique
- Concepteurs et distributeurs de solutions informatiques pour l'éclairage
- Fabricants et distributeurs d'autres applications LED/OLED

**Public ciblé et visiteurs qualifiés**

Plus de 4000 décideurs de provenance internationale seront présents en 2012 pour rencontrer les exposants :

- Fabricants de matériels d'éclairage, de systèmes d'éclairage pour les transports, d'écrans d'affichage, de matériels de signalisation, d'autres applications (industrielles, militaires, mobiles...)
- Architectes, prescripteurs, concepteurs, entreprises d'installation et de maintenance électrique, entreprises du bâtiment (promoteurs, constructeurs...), gestionnaires de bâtiments publics et privés, gestionnaires de lieux de vente et d'hospitalité (magasins spécialisés, centres commerciaux, franchises, hôtels, restaurants...), collectivités locales (techniciens, élus...)

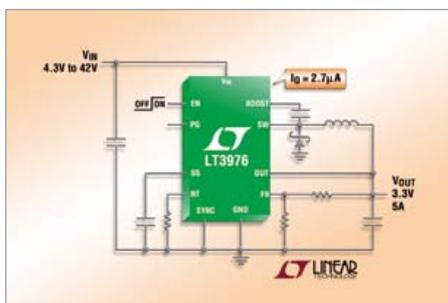
Pour plus d'informations ou votre badge d'accès :

[www.forumled.com](http://www.forumled.com) – [info@forumled.com](mailto:info@forumled.com) – +33 (0)4.37.40.31.63

## Convertisseurs DC / DC abaisseurs à très faible courant de repos



Linear Technology annonce le LT3975, un régulateur à découpage, abaisseur qui, sous 42 V à l'entrée, débite jusqu'à 2,5 A mais ne consomme que 2,7  $\mu$ A au repos. De même, le LT3976 fonctionne sous 40 V à l'entrée, et débite jusqu'à 5 A, avec seulement 3,3  $\mu$ A au repos. Cette faible intensité est obtenue grâce au fonctionnement en rafale (*Burst Mode*) des LT3975 et LT3976 qui conviennent donc aux applications, par exemple dans l'automobile, nécessitant un fonctionnement ininterrompu et une autonomie sur batteries optimale. La conception unique de ces composants permet de maintenir la tension de déchet à seulement 500 mV lorsque la tension d'entrée tombe en dessous de la tension de sortie programmée, une fonction nécessaire dans les applications soumises à des conditions de démarrage à froid pour l'automobile. La fréquence de commutation programmable par l'utilisateur de 200 kHz à 2 MHz permet au concepteur d'optimiser le rendement tout en évitant les bandes de fréquences critiques, sensibles au bruit. La combinaison de leur boîtier MSOP de 16 broches, à performances thermiques renforcées, et de la fréquence de découpage élevée qui autorise l'emploi de condensateurs et d'inductances externes de petite taille, conduit à une empreinte compacte et un bon rendement thermique.



Le fonctionnement en mode rafale, à faible ondulation, maintient un rendement élevé aux faibles intensités avec une tension d'ondulation inférieure à 15 mV<sub>crête-à-crête</sub>. Des techniques de conception spéciales, associées à un processus de gravure des transistors haute tension, permettent un rendement élevé sur une large gamme de tensions d'entrée, et leur topologie en mode courant autorise une réponse rapide aux transitoires et une excellente stabilité de boucle. Toutes les versions sont disponibles en stock.

### Résumé des caractéristiques : LT3975 & LT3976

- Très faible courant de repos :  $I_{\text{repos}} = 2,7 \mu\text{A}$  à  $V_{\text{entrée}} = 12 \text{ V}$  jusqu'à  $V_{\text{sortie}} = 3,3 \text{ V}$  (LT3975) et  $I_{\text{repos}} = 3,3 \mu\text{A}$  (LT3976)
- Fonctionnement en *Burst Mode* à faible ondulation, ondulation < 15 mV<sub>crête-à-crête</sub>
- Gamme étendue de la tension d'entrée : fonctionnement de 4,3 V à 42 V (LT3975) et 4,3 V à 40 V (LT3976)
- Intensité de sortie : 2,5 A max. (LT3975)
- Intensité de sortie : 5 A max. (LT3976)
- Performances excellentes au démarrage et faible tension de déchet
- Fréquence de commutation réglable : de 200 kHz à 2 MHz
- Synchronisation possible entre 200 kHz et 2 MHz
- Blocage précis programmable en cas de sous-tension
- Faible courant d'arrêt :  $I_{\text{repos}} = 700 \text{ nA}$
- Indicateur de tension établie
- Possibilité de démarrage progressif
- Protection par arrêt thermique
- Limitation de l'intensité par repliement de caractéristique
- Commutateur à saturation : résistance à l'état passant de 75 mohms
- Petits boîtiers MSOP de 16 broches, à performances thermiques renforcées

(120175-1)

## Le premier transcepteur vectoriel de signaux RF

Le VST (transcepteur de signaux vectoriel) RF de National Instruments PXIe-5644R, premier instrument conçu de façon logicielle, combine un générateur vectoriel et un analyseur vectoriel avec un FPGA programmable par l'utilisateur, dans un seul et unique instrument modulaire PXI. Les ingénieurs peuvent transformer le trans-

cepteur vectoriel de signaux en un nouvel instrument ou enrichir ses fonctionnalités existantes en utilisant le logiciel de conception de systèmes NI LabVIEW.

Le nouveau VST est idéal pour tester les dernières normes de communication cellulaires et sans fil comme les 802.11ac et LTE.



### Caractéristiques

- Couverture fréquentielle jusqu'à 6 GHz et 80 MHz de bande passante RF instantanée
- Performances maximales pour tester les toutes dernières normes de communication sans fil comme les 802.11ac et LTE
- Mesures 10 fois plus rapides qu'avec des solutions classiques
- Peut remplacer plusieurs instruments traditionnels pour une fraction du prix et de l'emplacement occupé
- Repose sur la technologie FPGA programmable avec LabVIEW
- Supporte facilement des configurations MIMO (multiple input, multiple output) ou le test parallèle dans un seul et unique châssis PXI

[www.ni.com/vst/f](http://www.ni.com/vst/f)

## Nouvelle gamme de parafoudres photovoltaïques

Ces parafoudres Finder, de type 2 - configuration en Y, (**série 7P**) conformes aux normes UTE C 61-740-51 et prEN 50539-11, sont conçus exclusivement pour

des systèmes photovoltaïques de 700 V CC, 1000 V CC et 1200 V CC. Selon le type, ils sont équipés de 3 varistors ou 2 varistors + 1 éclateur et possèdent chacun un contact de sortie permettant la signalisation à distance de l'état du varistor et le contrôle visuel en façade de l'appareil.  
Les modules sont débrochables.



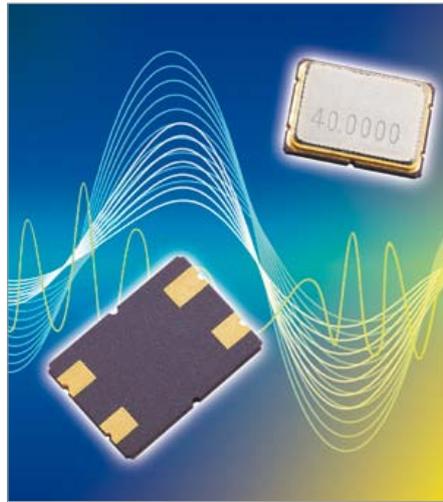
### Caractéristiques

- Tension photovoltaïque (Uoc stc): de 700 à 1200 V CC
- Intensité nominale (In) : 20 kA
- Intensité maxi (Imax) : 40 kA
- Niveau de protection par module (Up) : de 1,2 à 1,8 kV selon le type
- Tenue au courant de court circuit : 100 A – 200 V CC
- Montage sur rail 35 mm

[www.finder.f](http://www.finder.f)

## Mini quartz CMS bon marché pour modules sans fil

Le marché vertical des modules radio sub 1 GHz de tous les fabricants poursuit sa forte croissance. Le spécialiste des quartz et oscillateurs PETERMANN-TECHNIK de Kaufering propose à des prix très avantageux le MINI-SMD-QUARTZ LOW COST en boîtier céramique 3,2 x 2,5 mm avec des tolérances de fréquences très réduites, ainsi qu'une assistance au développement.



Ces quartz CMS fabriqués en salles blanches peuvent être fournis dans une plage de fréquence fondamentale de 12 à 60 MHz avec des tolérances de fréquence à partir de +-8 ppm. Des plages de température de service de 0/+40 °C à -40/+125 °C sont possibles. Le quartz de la série SMD03025/4 peut naturellement être certifié d'après la norme relative à l'automobile AECQ200. Pour garantir à ses clients des délais de commercialisation réduits, les spécialistes de PETERMANN-TECHNIK leur proposent une assistance au moyen d'analyses d'application et de circuits, de développements et de simulations de circuits. Des circuits de référence sont disponibles.

[www.petermann-technik.de/Quarze](http://www.petermann-technik.de/Quarze)

## Contrôleur double à diodes idéales en remplacement de deux diodes Schottky

Linear Technology propose un double circuit de commande à diodes idéales, de 0 V à 18 V, qui remplace deux diodes Schottky de forte puissance, et permet la réalisation de nombreuses alimentations à circuits OU, à faibles pertes, avec un minimum de perturbations pour les tensions. Le LTC4353 régule la tension directe d'un MOSFET canal N, externe, pour assurer un débit de courant égal entre alimentations, dans les applications à diodes à fonction OU. Dans les systèmes à basse tension, une

commutation lente entre des contrôleurs résulte en une chute de tension pendant le passage d'une alimentation à l'autre. La mise en marche rapide du LTC4353, inférieure à 1 µs, assure une commutation souple d'une ligne à l'autre, sans oscillation. Si une alimentation fait défaut ou est court-circuitée, l'arrêt rapide atténue le risque lié aux possibles courants inverses. En plus d'une alimentation redondante à fonction OU, le LTC4353 s'apparente à une diode de remplacement, excellente dans les applications où, pendant une défaillance momentanée de la tension d'alimentation la charge est isolée afin de maintenir la tension de sortie pendant une courte durée.

Le LTC4353 présente deux entrées permettant de valider les alimentations correspondantes. Avec les deux entrées de validation à l'état bas, le LTC4353 consomme seulement 75 µA par alimentation. Les sorties d'état individuelles indiquent si les MOSFET sont en marche ou à l'arrêt. Le contrôleur fonctionne avec des tensions de 2,9 V à 18 V et offre, en plus, un contrôle souple des tensions jusqu'à 0 V par l'adjonction d'une alimentation externe.



### Caractéristiques

- Remplace, à moindre pertes, les diodes Schottky de puissance dans les applications de plusieurs alimentations ou d'alimentation à maintien de tension
- Commande des MOSFET canal N, externes, pour un courant de forte intensité
- Gamme de tensions d'alimentations à circuits OU ou à maintien de tension, de 0 V à 18 V
- Entrées de validation
- Sorties d'état de marche des MOSFET
- Disponible en boîtiers MSOP de 16 broches et DFN (4 mm x 3 mm)

[www.linear.com/product/LTC4353](http://www.linear.com/product/LTC4353)



L'ORIGINAL DEPUIS 1994  
**PCB-POOL**<sup>®</sup>  
 Beta LAYOUT

## Prototypes PCB et Petites Séries

**Pochoir gratuit  
 avec chaque commande  
 "Prototype"**

**Evaluation**  
 Notez 5 commandes et recevez  
 un code de réduction de 10%

Appel Gratuit : FR 0800 90 33 30  
 sales@pcb-pool.com

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

**Beta**  
 LAYOUT  
 create : electronics

PCB-POOL est la marque déposée de Beta LAYOUT GmbH



**eSTORE**<sup>®</sup>  
 Beta LAYOUT

## Développer, assembler, souder



**€ 36,50\***

**Arduino Mega (ATMega 1280-16AU)  
 compatible**

**Reflow-Controller**



**€ 129,00\***

**LED Blinker 6 LEDs  
 Kit CMS**



**€ 6,00\***

**Le grand  
 Beta-Reflow-Kit**



**€ 129,00\***

**Trousse à outils  
 « Extended »**



**€ 149,00\***

\* TVA comprise, frais d'expédition en sus

[www.beta-eSTORE.com](http://www.beta-eSTORE.com)

**Beta**  
 LAYOUT  
 create : electronics

# retour aux sources (8)

## le préamplificateur BF

L'un des principaux domaines d'emploi du transistor est l'amplification de signaux sonores. Nous sommes là dans le cadre de la basse fréquence (BF). Peu importe que le son provienne d'un microphone ou d'un circuit radio, il faut le faire passer du murmure à la clameur !

### Burkhard Kainka (Allemagne)

Voici comment fonctionne un amplificateur linéaire (et on le souhaite aussi linéaire que possible !) : on module, en l'augmentant et le diminuant, un faible courant de base par un courant BF pour que le courant de collecteur contienne, amplifié, le signal BF correspondant. Le montage à émetteur commun est le plus couramment utilisé comme amplificateur. Son nom dérive du fait que l'émetteur est relié à la masse commune du circuit. La tension de base et celle de collecteur varient par rapport au potentiel de masse, tandis que la tension d'émetteur reste toujours la même, 0 V par exemple. Dans les schémas, il arrive que l'on ne représente pas la ligne continue de masse, qui est la plupart du temps le pôle négatif de l'alimentation, on utilise alors le symbole de la masse.

Un amplificateur devrait donc avoir la plage dynamique la plus large possible avant que la distorsion ne se manifeste, que le gain s'inscrive dans la partie uniforme de la courbe avant le genou. Pour y arriver, il

faudrait que le courant de repos du collecteur se situe à mi-hauteur de la courbe de courant accessible, compte tenu de la résistance de collecteur et de la tension d'alimentation. Dans ces conditions, l'intensité pourra varier autant dans un sens que dans l'autre, sans rencontrer de limitation. Mais le réglage du juste courant de polarisation au moyen d'une résistance de base n'est pas une sinécure.

Considérons la **figure 1**, nous y avons choisi au mieux les composants et, par hasard, le transistor BC547B présente très exactement un gain en courant de 300 fois. Le courant de base vaut  $I_B = U / R = (9 \text{ V} - 0,6 \text{ V}) / 560 \text{ k}\Omega = 15 \mu\text{A}$ . Du coup, le courant de collecteur fait  $I_C = I_B * \beta = 15 \mu\text{A} * 300 = 4,5 \text{ mA}$ . La chute de tension sur la résistance de collecteur s'élève alors à  $U = I * R = 4,5 \text{ mA} * 1 \text{ k}\Omega = 4,5 \text{ V}$ . Ce qu'il reste des 9 V, donc 4,5 V également, se trouve encore entre émetteur et collecteur. C'est le cas idéal, puisque la sortie peut ainsi offrir un débattement égal dans les deux sens. La tension de collecteur peut alors fluctuer

entre 0 V et 9 V et par exemple transmettre une pure sinusoïde sous la tension maximale sans déformation. L'amplification de tension est ici de plus de 100 fois.

Mais si vous construisez ce circuit avec différents transistors, vous n'obtiendrez pas chaque fois le même résultat, parce que le gain en courant peut varier beaucoup. Dans le pire des cas, le gain d'un transistor peut atteindre 600, ce qui fait que le transistor serait déjà en conductivité totale. Pas d'espoir de faible distorsion du signal de sortie. Mais il existe un remède approprié : la contre-réaction.

### La contre-réaction

La rétroaction la plus simple consiste à relier la résistance de base, non plus à la tension d'alimentation, mais au collecteur (**figure 2**). La formule empirique est  $R_B = R_C * \beta$ , dans laquelle  $\beta$  est la moyenne du gain en courant voulu. C'est ainsi que pour gain de 300 fois et une résistance de collecteur de 1 k $\Omega$ , on utilise une résistance de base de 300 k $\Omega$  environ.

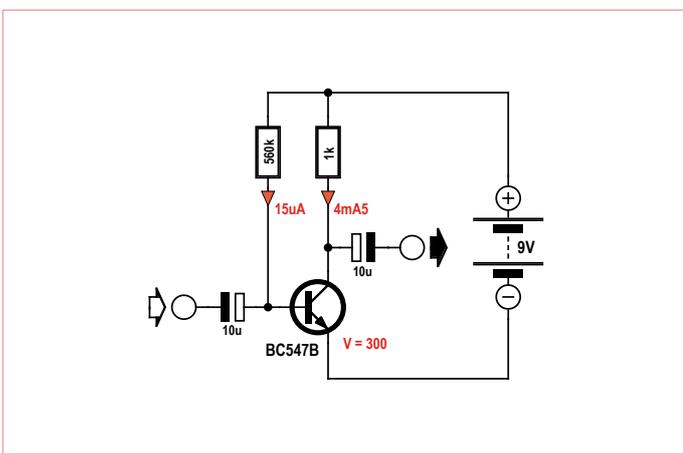


Figure 1. Réglage du point de fonctionnement pour une amplification de 300 fois.

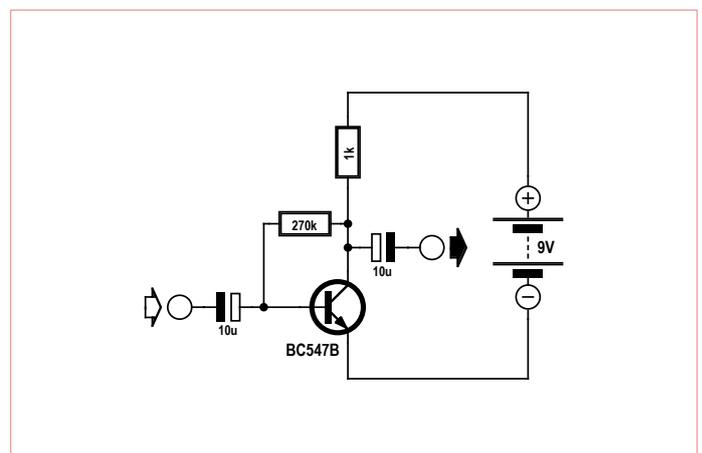
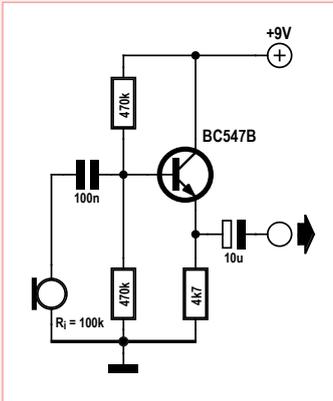


Figure 2. Réglage du point de fonctionnement par la rétroaction.

## Le collecteur commun



Le montage en collecteur commun, aussi appelé émetteur suiveur (*emitter follower*), signifie que c'est le collecteur du transistor qui fonctionne sous tension constante. La sortie de l'amplificateur se fait sur l'émetteur. Tout changement de la tension d'entrée produit automatiquement la même variation de tension sur l'émetteur, même un

très faible écart se répercute sur le courant de l'émetteur. La tension d'émetteur reste toujours à quelque 0,6 V sous celle de la base.

Une variation de la tension de base de 1 V oblige donc la tension d'émetteur à changer de pratiquement 1 V. La valeur précise pourrait être de 0,99 V, du fait que la variation du courant de repos au collecteur nécessite une déviation de la tension entre base et émetteur de 10 mV. Le gain en tension est donc proche de 1, ce qui veut dire que la tension d'entrée n'est pas amplifiée, mais bien le courant d'entrée. L'avantage du collecteur commun réside dans sa grande résistance ou impédance d'entrée. Alors que celle d'un émetteur commun avoisine 1 kΩ, celle d'un collecteur commun atteint, voire dépasse 100 kΩ. On peut de la sorte y brancher directement un microphone ou une cellule de lecture à cristal, par exemple. Ce circuit est capable de fonctionner avec un transducteur piézo-céramique simple comme microphone, mais également pour capter les vibrations corporelles : on peut ainsi, par exemple, écouter les battements du cœur.

De cette manière, le circuit se montre nettement plus conciliant à l'égard des différences de gain en courant. Un facteur d'amplification plus grand entraîne un plus fort courant de collecteur et donc une plus grande chute de tension sur la résistance de collecteur. La tension collecteur/émetteur diminue en même temps et en conséquence celle sur la résistance de base, causant une diminution du courant de base. En fin de compte, un facteur d'amplification plus grand détermine un courant de base proportionnellement plus réduit et le courant de collecteur augmente moins. On parle ici de contre-réaction parce que l'augmentation du courant de collecteur est contrecarrée. La contre-réaction conduit à un gain inférieur, mais aussi à une réduction de la distorsion. Ce qui compte, c'est que n'importe quel transistor NPN sorti d'un fond de tiroir fonctionnera toujours convenablement dans ce montage. En termes

plus choisis, l'ingénieur vous expliquera que le circuit se comporte avec davantage de tolérance par rapport à la dispersion de caractéristiques.

Un réglage plus précis du point de fonctionnement est plus exigeant. On gratifie alors le transistor d'une résistance d'émetteur (**figure 3**). Un diviseur de tension fixe le niveau sur la base, à condition que le courant qui y circule soit au moins dix fois supérieur à celui de polarisation de la base pour éviter toute dérive quand celui-ci fluctue. La tension d'émetteur se stabilise à cette même valeur, moins la tension entre base et émetteur. Grâce à quoi le courant d'émetteur est stable, celui de collecteur aussi, par la même occasion.

Dans l'exemple choisi, la base est portée au potentiel de 3 V. La tension d'émet-

teur s'ajuste donc à 2,4 V. Avec une résistance d'émetteur de 1 kΩ, un courant de 2,4 mA y circule. Aux bornes de la résistance de collecteur, qui vaut 1 kΩ, on trouve aussi 2,4 V. La tension de collecteur s'établit ainsi à 6,6 V, à distance de 4,2 V de l'émetteur.

Le circuit présente un fort taux de contre-réaction en tension. Une petite variation de la tension sur l'émetteur se répercute directement sur la tension de base à émetteur et, comme la caractéristique de base est raide, sur une grande modification des courants de collecteur et d'émetteur. Puisque seules de très petites variations de la tension base-émetteur sont nécessaires, l'émetteur reste continuellement à une tension à peu près 0,6 V en dessous de celle de la base. Le fonctionnement est donc similaire à celui d'un collecteur commun (voyez l'encadré).

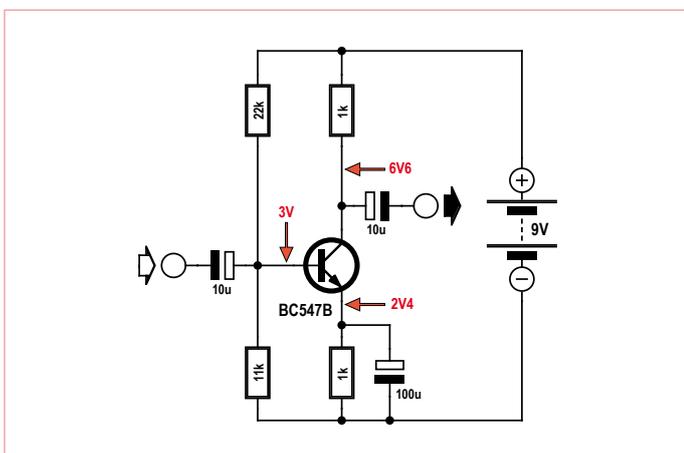


Figure 3. Stabilisation du point de fonctionnement.

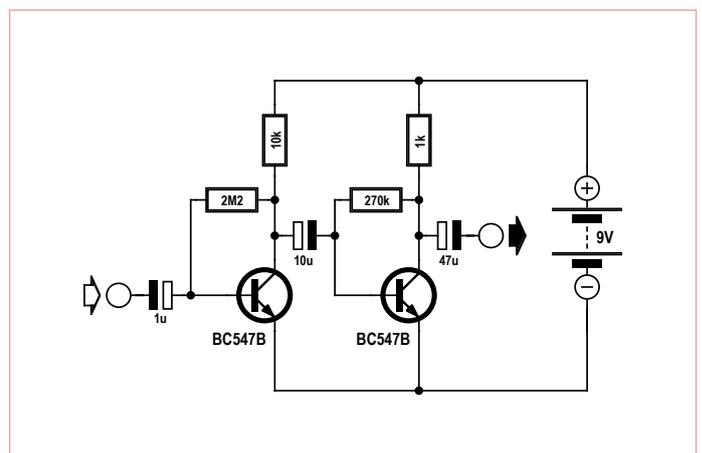
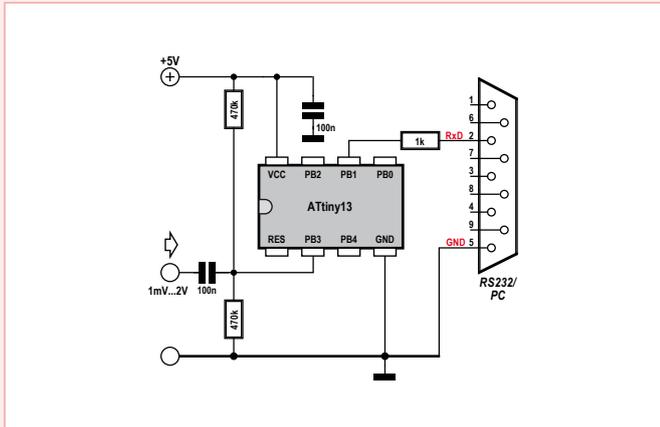


Figure 4. Un amplificateur BF à deux étages.

## Un millivoltmètre BF

Un millivoltmètre BF est l'accessoire tout trouvé pour examiner les amplificateurs présentés ici. En principe, le convertisseur A/N de l'ATtiny13 ne mesure que des tensions continues. Mais en haussant la tension moyenne à 2,5 V, il est possible d'analyser des tensions alternatives par des mesures rapides. Les résultats sont envoyés en série au PC et peuvent être visualisés à l'aide d'un programme de terminal.



Le principe du programme de mesure est simple. D'abord, un système de détermination de la moyenne calcule la valeur médiane. Suit alors une série rapide de nombreuses mesures élémentaires dont on soustrait la valeur médiane pour en tirer la valeur absolue. En fait, l'échelon de tension le plus petit que le convertisseur A/N peut mesurer est de 5 mV. Mais en calculant la moyenne sur de nombreuses mesures indépendantes, on peut malgré tout évaluer des valeurs plus petites que le millivolt. Dans la boucle de mesure effective, on procède à 2 780 mesures que l'on traite et additionne. Les points de mesure sont toujours asynchrones par rapport au signal à mesurer. Mais le grand nombre de mesures et leur caractère aléatoire font que le résultat est valable tant que le signal d'entrée reste dans la plage de 50 Hz à 50 kHz.

Mais d'où sort ce mystérieux 2 780 ? Il provient de la tension de référence, précisément 5 V, et anticipe la différence entre une moyenne arithmétique et la mesure en valeur efficace réelle (rms, moyenne géométrique), si bien que le résultat s'indique en mVeff, en valeur efficace vraie dans le cas d'un signal sinusoïdal. Pour déterminer des valeurs efficaces précises, il faudrait calculer le carré de la tension, puis après, extraire la racine carrée de la moyenne. C'est au-dessus des forces d'un ATtiny13. Aussi prend-on ici la moyenne arithmé-

tique de la tension absolue. Elle est environ 10 % plus basse, plus précisément c'est la racine de 2 divisée par pi/2, soit 0,9003. Un pas du CAN représente  $5\,000\text{ mV} / 1\,023 = 4,8876\text{ mV}$ . De plus, il faut multiplier par 8 la valeur mesurée puis la diviser par 4 096, ce qui revient à la diviser par 512. Pour un résultat final correct en mV, nous devons donc multiplier par  $512 * 4,8876 / 0,9003 = 2\,780$ . Le plus simple, c'est d'additionner 2 780 valeurs de mesure.

Le résultat est étonnamment précis. Même des valeurs de 1 ou 2 mVeff s'affichent de manière stable !

```
'Millivoltmeter 1 mVeff ... 2000 mVeff
$regfile = "attiny13.dat"
$crystal = 1200000
$hwstack = 8
$swstack = 4
$framesize = 4
```

```
Dim U1 As Integer
Dim U2 As Integer
Dim U3 As Long
Dim N As Integer
```

```
Config Adc = Single , Prescaler = Auto
Start Adc
Open "comb.1:9600,8,n,1,INVERTED" For Output As #1
```

```
Do
    U2 = 0
    For N = 1 To 64
        U1 = Getadc(3)
        U2 = U2 + U1
    Next N
    Shift U2 , Right , 3 ' /8
    U3 = 0 ' Zéro
    For N = 1 To 2780
        U1 = Getadc(3)
        Shift U1 , Left , 3 ' *8
        U1 = U1 - U2
        U1 = Abs(U1)
        U3 = U3 + U1
    Next N
    Shift U3 , Right , 12 ' / 4096
    Print #1 , U3
```

```
Loop
```

```
End
```

Une puissante contre-réaction entraîne en principe un très faible gain en tension. Aussi, pour conserver une amplification suffisante des signaux BF, on ajoute dans le circuit d'émetteur un condensateur qui annule l'effet de la rétroaction sur les courants alternatifs. Sa valeur détermine la fréquence la plus basse qui peut être transmise

et doit donc être choisie assez grande.

### Et avec deux étages ?

Si le gain d'un transistor ne vous suffit pas, mettez-en plusieurs. La **figure 4** montre une version à deux étages couplés par RC. Dans la plupart des cas, une manière simple de réaliser la contre-réaction au

moyen d'une résistance entre base et collecteur suffit à fixer un point de fonctionnement adéquat. Le circuit illustre une autre règle de conception d'un amplificateur : de la gauche vers la droite, l'impédance du circuit devient de plus en plus faible. On obtient ainsi une haute impédance d'entrée et une faible résistance en

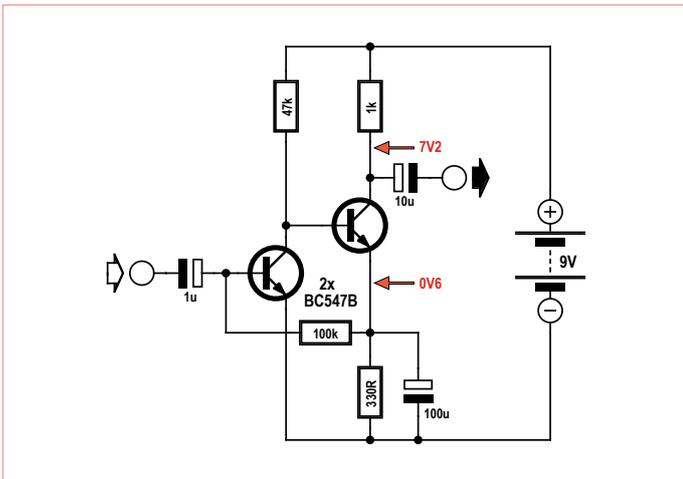


Figure 5. Un amplificateur BF à deux étages couplés en continu.

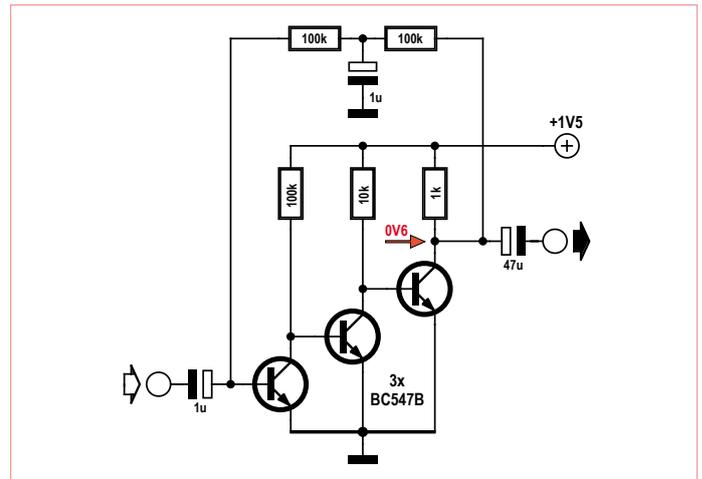


Figure 6. Un amplificateur à trois étages à couplage direct.

sortie. On n'arrivera pas encore à émouvoir un haut-parleur, mais on pourrait fort bien s'en tirer avec un écouteur.

### Le couplage d'étages en continu

Il est possible de réaliser à moindres frais des amplificateurs BF à plusieurs étages en remplaçant entre eux le couplage à condensateur par un couplage direct, en continu. La même résistance sert de charge du collecteur du premier et de polarisation de base du suivant (figure 5). On peut souvent trouver un moyen simple de fixer le point de fonctionnement en fournissant aux deux transistors un courant de base convenable. La figure 5 représente un amplificateur à deux étages à couplage direct. Le courant de base du premier est prélevé sur l'émetteur de second. On crée ainsi une rétroaction et une stabilisation du point de fonctionnement. Sur une résistance de base relativement petite, de 100 kΩ, la chute de tension est négligeable et donc la tension d'émetteur du second transistor doit avoisiner 0,6 V. La résistance d'émetteur de 330 Ω définit ainsi les courants d'émetteur et de collecteur du second étage qui, en prime, sont indépendants de la tension d'alimentation.

La contre-réaction ne doit pas occasionner de diminution du gain en alternatif. Un condensateur d'émetteur veille à ce que seul le courant continu soit renvoyé. En pratique, l'amplificateur présente une fréquence de coupure inférieure que l'on peut encore repousser en augmentant la capacité du condensateur d'émetteur.

### Et maintenant, trois étages !

C'est encore plus facile avec un amplificateur à trois étages, parce qu'alors, on peut

pratiquer une rétroaction directe de la sortie vers l'entrée (figure 6). En fait, chaque étage procure une rotation de phase de 180 degrés. Avec deux étages, on serait dans une situation de rétroaction comme sur une bascule. Avec un troisième étage, on obtient une contre-réaction qui stabilise le point de fonctionnement.

La figure 6 présente un circuit qui s'accommode parfaitement d'une alimentation basse, à partir de 1 V environ. Tout comme à la figure 2, on voit ici qu'un plus grand gain des transistors conduit à une augmentation du courant et donc de la tension aux bornes de la résistance de collecteur du troisième transistor. La tension entre collecteur et émetteur diminue et donc aussi celle sur les deux résistances de 100 kΩ de rétroaction. Au bout du compte, comme on l'a vu précédemment, on note une diminution du courant de base.

La tension de collecteur à émetteur ne fait ici qu'à peu près 0,6 V, aussi le gain en BF de chaque étage est-il relativement faible, mais il est largement compensé par l'adaptation du nombre d'étages.

La ligne de contre-réaction du circuit comporte un filtre passe-bas avec condensateur vers la masse, en découplage (ou *bypass*), qui élimine la haute fréquence de la boucle. Il est indispensable quand on veut un grand gain en BF.

Ce circuit convient bien comme préamplificateur sensible pour microphone ou comme ampli audio dans des récepteurs simples. On peut lui adapter directement un casque d'écoute qui prend alors la place de la résistance de collecteur du troisième étage. Il a

d'ailleurs été utilisé dans le projet de détecteur de l'e-smog, Le flair du Tapir, paru dans le numéro d'été 2012 d'Elektor ([www.elektor.fr/120354](http://www.elektor.fr/120354)).

(120008 – version française : Robert Grignard)

## Jeu-concours

### La solution du jeu de juin était le code CDG.

#### Explication.

#### Solution 1 :

La chute de tension sur la LED est d'environ 1,8 V,  $U_{CE}$  du transistor est env. 0,1 V. Il reste 7,1 V. 1 kΩ et 470 Ω forment un diviseur de tension de 3,1 à 1.

$$7,1 \text{ V} / 3,1 = 2,3 \text{ V (Solution C).}$$

#### Solution 2 :

Pour que le transistor commence à conduire, il faut que sa tension de base soit d'environ 0,6 V plus haute que celle de l'émetteur, donc 2,3 V + 0,6 V = 2,9 V. La bonne réponse est D.

#### Solution 3 :

Si le transistor de droite est complètement bloqué et celui de gauche saturé, on trouve env.  $9 \text{ V} / 11 = 0,8 \text{ V}$  sur la résistance d'émetteur. À cela s'ajoute la tension de base à émetteur de 0,6 V que le transistor de gauche utilise pour atteindre une conduction suffisante. Le niveau de commutation inférieur se situe à env. 1,4 V. C'est la réponse G.

# détecteur de rayonnement revisité

petites modifications, grande amélioration

Burkhard Kainka (Allemagne)

Vous êtes nombreux à utiliser le détecteur de rayonnement d'Elektor, un compteur particulièrement adapté aux mesures de longue durée et à l'analyse d'échantillons faiblement radioactifs. Son comportement n'étant pas figé dans le marbre mais dans le cuivre et le silicium, nous allons voir qu'il suffit de quelques modifications matérielles et logicielles pour l'améliorer.



Nous vous avons présenté en novembre 2011 une version améliorée du détecteur de rayonnement Elektor [3]. L'instrument, disponible en kit de montage [2], est piloté par un microcontrôleur ATmega88 doté d'un chargeur de démarrage. Nous verrons au travers d'exemples que ce chargeur et de simples changements du programme de base permettent d'obtenir de l'instrument une réponse spécifique. Nous commencerons toutefois par dégainer notre fer à souder pour procéder à quelques modifications matérielles du détecteur.

## Étalonnage

On nous a souvent demandé s'il était possible d'étalonner le détecteur en *microsievert* par seconde ( $\mu\text{Sv/h}$ ). La réponse n'est pas évidente puisqu'elle dépend de nombreux facteurs, dont le plus important est le blindage utilisé.

Nous avons pu constater en pratique à quel point il était délicat de rendre à la fois le logement du capteur totalement opaque, et d'avoir un rayonnement le moins absorbé

possible. Deux stratégies sont possibles : placer le capteur et l'échantillon dans une boîte en fer-blanc mise à la masse, ou envelopper l'ensemble de la platine dans une feuille d'aluminium, elle aussi reliée à la masse. Certains lecteurs ont observé un effet microphonique avec cette dernière méthode, la photodiode et la feuille d'aluminium formant en effet une sorte de microphone électrostatique. Une solution est de coller la feuille directement sur le capteur, ou de le recouvrir d'une feuille de cuivre autocollante de 0,035 mm d'épaisseur, ce qui de surcroît facilitera le soudage de la liaison de masse.

Nous avons effectué une mesure avec un blindage en feuille de cuivre et une BPW34. La source du rayonnement était un vieux manchon à incandescence au thorium dans son emballage de papier. Les résultats des mesures ne dépendaient que peu de la distance, puisqu'une source de grande surface produit un champ de rayonnement homogène à son voisinage. Le détecteur Elektor a

mesuré 40 impulsions par minute à 1 cm de la source. À la même distance, un compteur de Geiger professionnel équipé d'un tube compteur pour rayons  $\beta$  et  $\gamma$  a mesuré 800 impulsions par minute, ou  $20 \mu\text{Sv/h}$ . D'où l'équivalence (non homologuée) :

$$1 \text{ imp/min} = 0,5 \mu\text{Sv/h}$$

Si la mesure est faite sur une durée suffisante, l'instrument permet en principe de mesurer la dose reçue de n'importe quel échantillon. Il pourrait aussi, pourquoi pas, mettre en évidence une augmentation de l'activité solaire. La période de mesure de notre détecteur est ici toutefois beaucoup plus longue que celle nécessaire à un compteur Geiger, ce qui peut s'avérer gênant. Vous pourrez par exemple vous assurer que le niveau de radioactivité de votre salade est sans risque, mais après mesure celle-ci risque de ne paraître fraîche et savoureuse qu'aux lapins.

Notre détecteur se montre de fait plus à son avantage dans l'analyse de petits échan-

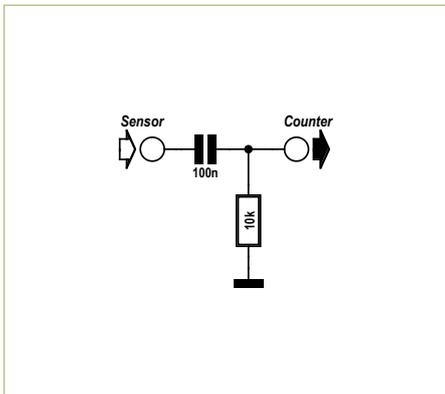


Figure 1. Couplage CA.

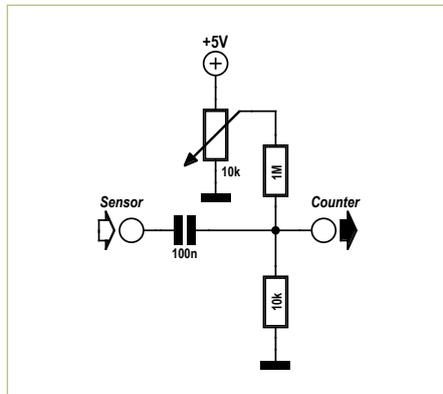


Figure 2. Réglage du seuil de réponse.

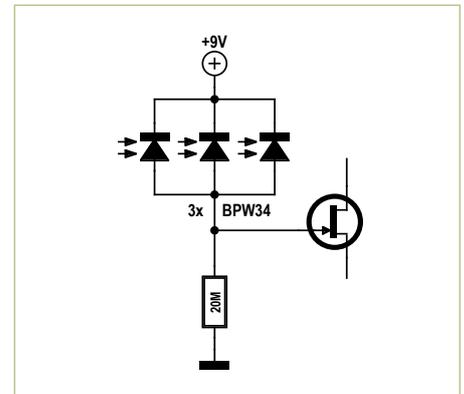


Figure 3. Trois BPW34 en parallèle.

tillons faiblement radioactifs que dans la mesure du rayonnement ambiant. Il est même à cet égard supérieur à un compteur de Geiger, puisque son petit capteur permet de mieux distinguer un signal de la radioactivité naturelle.

## Couplage CA

Nous pouvons également diminuer la sensibilité à la lumière de façon électronique en réalisant un couplage CA entre le capteur et le compteur. Pour cela nous avons besoin d'un condensateur de 100 nF et d'une résistance de 10 kΩ (fig. 1). Le circuit agit alors comme un filtre passe-haut doté d'une fréquence de coupure de 160 Hz. Peu importe donc qu'un flux lumineux uniforme atteigne maintenant le capteur et décale peu à peu le point de fonctionnement de l'amplificateur du capteur. Même le léger scintillement à 100 Hz typique d'une lampe à incandescence sera quelque peu atténué. Il n'existe par contre aucun

moyen de contrer le rayonnement d'un tube fluorescent, car celui-ci produit des ondes lumineuses à flancs raides. Avec ce filtre, le circuit se montre néanmoins globalement moins photosensible.

Ajouter un potentiomètre permet de modifier le seuil de déclenchement (fig. 2). Nous pouvons déjà le faire depuis le programme PC associé, mais certains lecteurs aimeraient se passer de la liaison PC, et également effectuer sans contrainte différentes mesures à l'extérieur. Quoi de plus compréhensible ? Trimbaleriez-vous par exemple votre coûteux portable au milieu d'une carrière pour en mesurer la radioactivité ? Avec le circuit présenté ici, il est possible de régler le niveau moyen jusqu'à 50 mV.

Le seuil de déclenchement est pré-réglé sur 10 dans le programme, soit environ 50 mV. Au démarrage le microcontrôleur mesure la tension moyenne à son entrée, puis lui ajoute cette valeur de seuil. Toutes les impulsions d'amplitude supérieure sont comptées. Avec le nouveau couplage CA, le contrôleur établirait le niveau moyen comme étant la moitié du niveau de bruit de l'entrée, pour arriver p. ex. à une valeur de 5. Le seuil de déclenchement vaudra alors 5. Le niveau de repos peut être augmenté jusqu'à 50 mV à l'aide du nouveau potentiomètre. D'abord positionné en butée basse, on le tournera lentement pour chercher la limite de bruit, déterminant ainsi le point de plus grande sensibilité. Appuyer sur le bouton d'initialisation lancera une nouvelle mesure avec ce paramétrage.

## Association de capteurs

La pièce principale du capteur est une unique diode PIN BPW34, mais rien ne nous empêche d'en câbler plusieurs en parallèle (fig. 3). Cette association est le meilleur moyen d'obtenir des valeurs de mesure vite exploitables. On pourra alors soit se contenter de compter les impulsions, soit afficher le spectre d'énergie à l'aide du programme d'analyse Alpha Gamma.

Une mesure à vide est nécessaire avant toute mesure réelle afin d'évaluer le niveau du rayonnement naturel. La figure 4 montre le résultat d'une telle mesure, effectuée avec trois BPW34 reliées en parallèle et une valeur de seuil de 5. Quarante impulsions d'énergies différentes furent enregistrées en une seconde. Ces valeurs à vide sont à soustraire des mesures suivantes dans le cas d'échantillons faiblement radioactifs.

Le chlorure de potassium, plus précisément le potassium 40 qu'il contient, montre une



Figure 4. Mesure à vide avec trois BPW34.

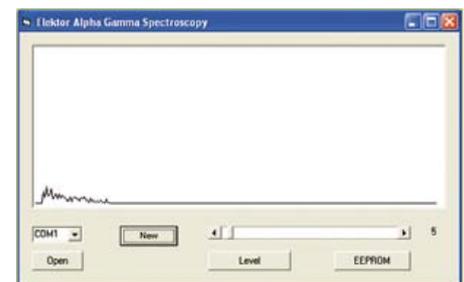


Figure 5. Rayonnement  $\beta$  du bicarbonate de potassium.

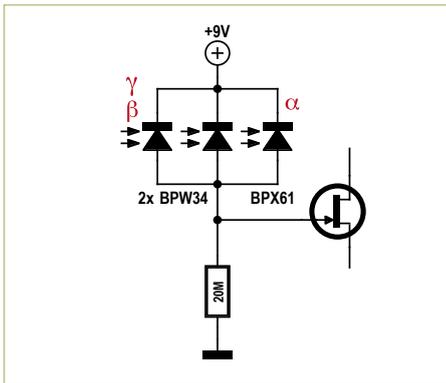


Figure 6. Association de capteurs.

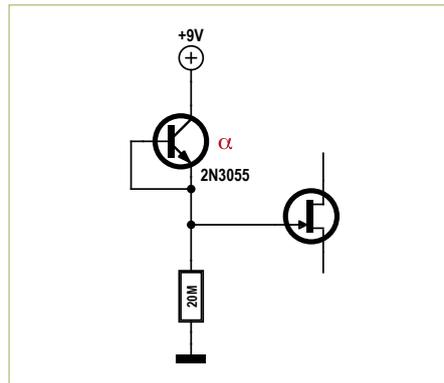


Figure 7. Un transistor de puissance comme capteur  $\alpha$ .

faible activité bêta. Plus faible encore est celle du bicarbonate de potassium (qui sert p. ex. d'additif alimentaire). Placé dans un petit sac en plastique près du capteur, il affichait au bout d'une heure 251 impulsions (fig. 5), soit environ 6 fois plus que le rayonnement ambiant.

Profitons-en pour citer aussi ces piles alcalines supposées ne jamais fuir. Lorsqu'elles le font malgré tout, elles libèrent de l'hydroxyde de potassium concentré qui réagit avec le  $\text{CO}_2$  de l'air pour donner du carbonate de potassium. Une mesure avec le détecteur Elektor nous a montré que ces résidus blancs pouvaient également servir d'échantillons de test pour la mesure des rayonnements faibles.

Ainsi équipé, doté d'une plus sensibilité, adapté aux impulsions de faible énergie, le détecteur peut donc aussi être utilisé pour les matériaux faiblement radioactifs. Mesurez par exemple les cendres de charbon de bois d'un barbecue. La radioactivité de sa potasse est mesurable, nous avons compté

environ 150 impulsions au bout d'une heure. Nous avons aussi mesuré la radioactivité des cendres d'une cigarette : 80 impulsions en une heure, soit le double du bruit de fond. Le principal responsable est probablement le potassium, même si le rôle du polonium n'est pas à exclure puisqu'il s'agit d'un sous-produit de la désintégration de l'uranium, un élément qui s'accumule dans la plante de tabac.

### Mesure alpha

Les particules alpha peuvent aussi être mesurées avec une BPX61 dépourvue de son hublot protecteur. On combinera avec profit les deux types de photodiodes, p. ex. deux BPW34 pour les mesures  $\beta$  et  $\gamma$ , et une BPX61 pour le rayonnement  $\alpha$  (fig. 6).

Pour vérifier vos résultats, mesurez d'abord la faible radioactivité d'un sel de potassium ou de potasse, et réglez en conséquence le seuil de déclenchement. Cette étape réussie, vous pourrez vous attaquer à d'autres analyses. La radioactivité de certains matériaux de construction est ainsi intéressante à étudier. Les rayons alpha sont caractérisés

sés par leur haute énergie. Vous observerez p. ex. un rayonnement notable en provenance des plaques de plâtre, car le gypse dont dérive le plâtre provient de la désulfuration des rejets gazeux des centrales à charbon, et le charbon peut contenir des traces d'uranium et de ses produits de désintégration.

### Un 2N3055 comme capteur alpha

Des particules alpha heurtant une jonction au silicium produisent un signal relativement puissant. Cet effet concerne les photodiodes, les diodes en général, et les transistors en particulier. La plupart des rayons alpha n'atteignent toutefois pas la jonction, car ils sont interceptés par le boîtier ou toute autre couche protectrice. Certains transistors de puissance en boîtier TO3 font à cet égard exception. Si l'on retire leur capot métallique, l'ensemble de la surface de la puce de silicium se retrouve entièrement exposée. C'est généralement le cas avec un 2N3055, mais d'autres ont une couche de protection supplémentaire. Comptez donc avant tout sur la chance pour vos essais. Nous en avons eu avec un nouveau transistor de chez ST (fig. 7). La puce est propre et forme un excellent capteur de rayons  $\alpha$ . Les rayons  $\beta$  et  $\gamma$  ne produisent a contrario aucun signal distinctif. La figure 8 montre le spectre d'un échantillon de pechblend. Les désintégrations des atomes d'américium 241 présents dans un détecteur de fumée produisaient quant à eux tant d'impulsions que le compteur ne pouvait les enregistrer toutes.

### Interface série

La partie de la platine associée au compteur dispose d'un connecteur tripolaire qui permet l'accès à l'interface série du  $\mu$ contrôleur. Les signaux RXD et TXD y sont à niveaux TTL. Le connecteur série Elektor Breakout-Board (BOB) [1] est compatible avec le FT232R, et avec lui vous pourrez donc relier platine et PC par USB (interface série virtuelle).

Vous pouvez également lui relier un autre appareil par interface RS-232 grâce à un circuit très simple (fig. 9) : ici deux transistors NPN suffisent pour inverser les niveaux. Cette adaptation de niveau permet d'exploiter le détecteur aussi bien via une inter-

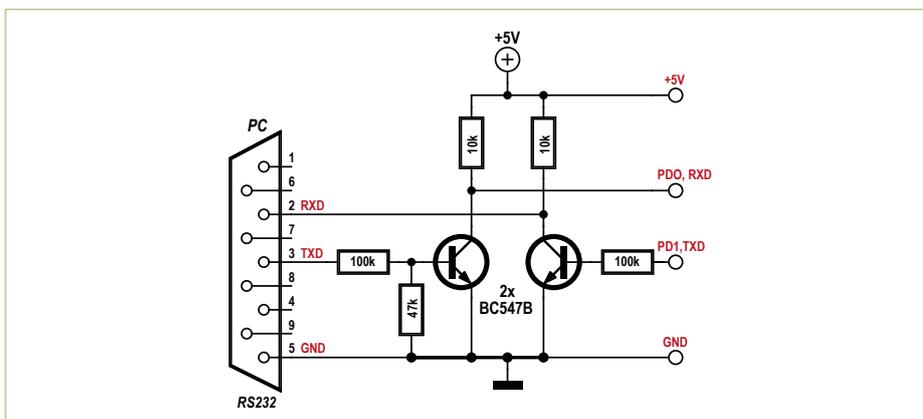


Figure 9. Adaptation de niveau RS-232.

face COM classique qu'avec un adaptateur USB/série ordinaire.

Le programme PC sert à mesurer des spectres d'énergie et à modifier le seuil de réponse. Autre emploi utile grâce à l'interface série : l'installation de mises à jour ou l'écriture de programmes dans le microcontrôleur.

## Charger un programme

Peut-être voudrez-vous modifier le fonctionnement du compteur. Rien de plus facile avec le chargeur de démarrage, et la version d'évaluation de Bascom (fig. 10) le permet aussi. Chargez d'abord le code source Basic, puis compilez-le après l'avoir éventuellement modifié.

Sous BASCOM-AVR, choisissez ensuite le programmeur approprié, ici MCS-Bootloader (fig. 11). Pensez à sélectionner les bons port COM et débit (19200 bauds).

Lancez le programmeur (= chargeur de démarrage) avec p. ex. le bouton vert symbolisant une platine : MCS-Bootloader va tenter de se connecter avec la puce.

La liaison s'établit rapidement après une initialisation ou un nouveau démarrage du contrôleur. Mettez donc hors puis sous tension la platine afin de lancer le chargement (fig. 12).

Une fois le processus terminé, le contrôleur

démontre immédiatement avec le nouveau programme.

## Modifier le code

Nul besoin d'être un expert en Basic pour modifier le programme puis écrire le nouveau micrologiciel dans le contrôleur. Quelques exemples :

### 1. Seuil de déclenchement constant (Listage 1)

Il est conseillé de toujours commencer avec une valeur de seuil de 5. Pour cela il suffit d'insérer la ligne « L = 5 » à l'endroit approprié du code. Pour omettre le message de démarrage « Counter », il nous a suffi de commenter la ligne correspondante avec un point-virgule. Et en décommentant la ligne « Um = 0 », nous affecterions 0 à la valeur moyenne de référence, 0 étant la valeur la mieux adaptée au circuit doté du filtre à découplage CA (fig. 1).

### 2. Mesure d'une distribution énergétique (Listage 2)

Ce code compte les impulsions en fonction de leur énergie. Trois compteurs sont présents : N pour l'ensemble des événements, N2 pour ceux de niveaux supérieurs à 40, et N3 pour les impulsions de niveaux supérieurs à 80. Pas besoin de PC ici, l'affichage des trois résultats sur une même ligne du

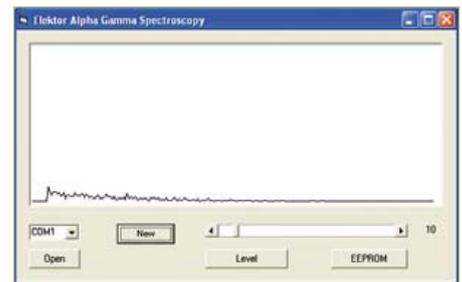


Figure 8. Mesure avec un 2N3055.

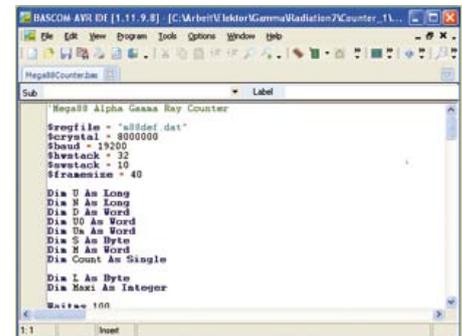


Figure 10. L'environnement de développement Bascom.

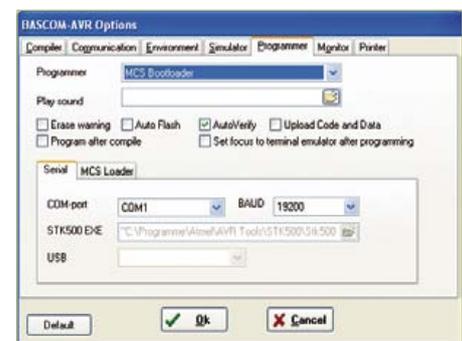


Figure 11. Sélection du programmeur.



Figure 12. Chargement en cours.

### Listage 1 : Counter\_1

```
'Print "Counter"
Locate 1 , 1
'Lcd "Counter"
'Waitms 500
Readeeprom L , 1
If L = 255 Then L = 5
U = 0
For N = 1 To 1000
    D = Getadc(0)
    U = U + D
Next N
U = U / 1000
Um = U
'Um = 0
U0 = Um + L
N = 0
```

### Listage 2 : Counter\_2

```
Do
    D = Getadc(0)
    If D > Maxi Then Maxi = D
Loop Until D < U0
Maxi = Maxi - Um
If Maxi > 255 Then Maxi = 255
N = N + 1
If Maxi > 40 Then N2 = N2 + 1
If Maxi > 80 Then N3 = N3 + 1

Locate 1 , 1
Lcd N
Lcd " "
Locate 1 , 8
Lcd N2
Locate 1 , 14
Lcd N3
```

LCD donne déjà une estimation de la distribution énergétique. C'est un des principaux avantages des capteurs à semi-conducteurs par rapport aux compteurs de Geiger.

### Des candidats radioactifs

On nous demande souvent quels objets ou matériaux mesurer pour vérifier le bon fonctionnement du détecteur. Le premier article suggérait déjà les minerais d'uranium, les aiguilles luminescentes, et le chlorure de potassium. Voici d'autres candidats :

L'ouraline, un verre fluorescent toujours utilisé, p. ex. pour la fabrication de perles. Les électrodes en tungstène thorié de type WT20, qui contiennent 2 % d'oxyde de thorium et présentent une radioactivité alpha, bêta et gamma faible.

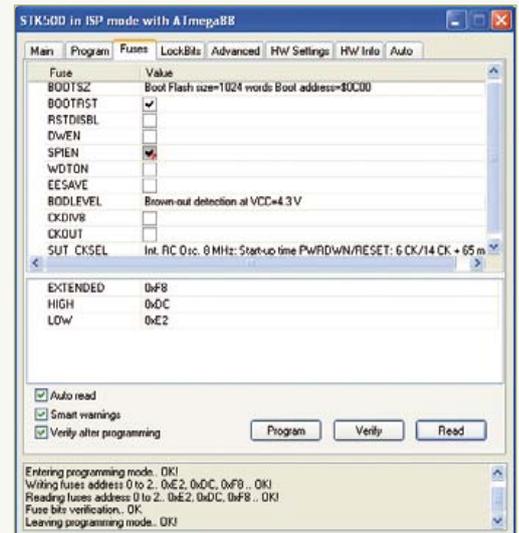
Les cathodes des tubes émetteurs de forte puissance et à chauffage direct, que cela dit on ne trouve pas à chaque coin de rue. Le tungstène et le thorium qu'ils contiennent pourront aussi être détectés dans le magnétron d'un micro-ondes. Sa cathode une fois démontée est un matériau d'essai efficace. Les manchons à incandescence des anciennes lampes à pétrole ou à gaz, qui contenaient des sels de thorium. Les nouveaux manchons n'en utilisent plus, mais si vous en dénicher un vieux vous posséderez certainement une puissante source de rayons, en particulier de rayons gamma pénétrants.

Les détecteurs de fumée ioniques, dont l'américium 241 représente une source de rayons alpha de 33 kBq. Courants aux États-Unis et en Grande-Bretagne, plus rares en Europe, ils représentent une source de rayons gamma relativement faible : une BPW34 ne détectera qu'une trentaine d'impulsions par minute. Une BPX61 à nu sera par contre un bon détecteur de rayons alpha, et délivrera plusieurs milliers d'impulsions par minute.

À votre tour maintenant ! Vous n'aurez que quelques modifications à apporter au détecteur pour étendre ses possibilités. Le kit [3] est facile à assembler puisqu'il ne contient que des composants traversants. La connexion au PC sera quant à elle sim-

## Programmer soi-même le chargeur de démarrage

Ceux qui ont gravé eux-mêmes le microcontrôleur l'ont en général fait sans le chargeur de démarrage (*bootloader*). Le chargeur peut être installé après coup. Vous trouverez sur la page Elektor associée à cet article un chargeur adapté à l'ATmega88, ainsi que les exemples écrits pour l'environnement BASCOM [2]. Le fichier compilé BootLoader88.hex doit être gravé dans la mémoire de l'ATmega à l'aide d'un programmeur ISP, p. ex. le STK500. Le programme Counter présent sera d'abord écrit en mémoire. Installez-le une nouvelle fois avec le chargeur de démarrage en fin de processus, pas avec un programmeur matériel, de façon à ce que le chargeur ne soit pas ré-effacé par la nouvelle programmation ISP. Les fusibles doivent au préalable être positionnés correctement (voir **figure**). Il importe de paramétrer et activer le secteur de boot avec une taille de 1024 mots. Une fois l'ensemble configuré, le programme Counter peut être flashé directement via l'interface série.



Quelques faits pour bien saisir le processus : lorsque le chargeur est appelé depuis l'environnement de développement, le PC envoie sans arrêt une série d'octets 123, et attend la même réponse du contrôleur connecté. Toutes les données de programme nécessaires sont alors envoyées via l'interface série et transférées dans la mémoire flash.

Côté microcontrôleur, après un nouveau démarrage ou une initialisation, le contrôleur lit d'abord le contenu du secteur de démarrage, puisque tels ont été paramétrés les fusibles. Le programme du chargeur qui y réside doit déterminer si un processus de démarrage est planifié. Si ce n'est pas le cas, le contrôleur se rend à l'adresse 0 et lance le micrologiciel présent. Si par contre il trouve un octet 123, il le renvoie et reçoit d'autres données qu'il transfère dans la mémoire de programme. Le nouveau programme est lancé depuis l'adresse 0 à la fin de cette programmation.

plifiée par l'utilisation du convertisseur USB/série BOB d'Elektor [1] [2].

(120381 – version française : Hervé Moreau)

### Liens :

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

[2] [www.elektor.fr/120381](http://www.elektor.fr/120381)

[3] [www.elektor.fr/110538](http://www.elektor.fr/110538)



- ✓ 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 !**

### Microcontrôleur

Composant



Fabriquée en Allemagne !

DIAMEX PROG-S  
**21,50**

#### Programmeur ISP multi-talents pour trois modes de programmation

Programme les microcontrôleurs favoris AVR, SMT32, NXP et LPC

#### Fonctionnalités :

- Fuse-bits • FLASH • EEPROM
- Écriture • Lecture lock-bits
- Lecture du registre OSCCAL
- Suppression mémoire

#### Programmeur ISP - USB

DIAMEX USB ISP  
**19,95**



Programmeur ISP à un prix extrêmement bon marché pour les processeurs ATMEL programmables à partir de l'interface SPI simple. Il se démarque par le processeur intégré doté d'un logiciel d'exploitation spécialement développé qui permet un cycle de programmation très rapide.

- AVR STK 500, AT Mega, AT Tiny, AT 90

DIAMEX ALL AVR  
**29,50**



DIAMEX ALL-AVR est adaptateur de programmation qui peut être employé dans le cadre de la programmation des contrôleurs AVR couramment utilisés d'ATMEL.

- AT 90, AT Mega, AT Tiny, ATX Mega

### Besoin de l'atelier

## Technique des mesures

Alimentation en courant électrique



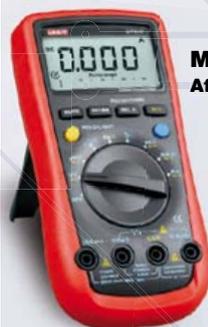
UTD 2052 CEL  
**299,00**

UNI-T

#### Oscilloscope à mémoire 2 canaux

Oscilloscope à mémoire présentant d'excellentes caractéristiques, doté d'un affichage couleur extrêmement grand avec éclairage en arrière plan et d'une interface USB pour l'export de résultats de mesures pour PC/ordinateur portable.

- Largeur de bande : 50 MHz
- Temps de montée : <7 ns
- Gamme échantillon : 1 C.C./s



UT 61E  
**79,95**

UNI-T

#### Multimètre numérique Affichage Comptage 22000

Appareil de mesure haute précision, barographe analogique et indication en cas de batterie faible.

- Mesure RMS
- Test de diodes
- Bip de passage
- Data-Hold • MIN/MAX
- Fonction REL
- Fonction Valeur de crête
- Interface USB
- Sécurité : Cat. II 600 V, Cat. III 300 V,



#### Détecteur de tension AC 90-1000 V AC

- Sécurité : EN 61010-1, CAT III 1000 V
- signal acoustique/visuel
- lampe de poche intégrée
- piles comprises



PEAKTECH 1030 **9,95**

### Technologie réseau et PC

## Connecteur

Technologie satellite et TV



NEUTRIK  
CONNECTING THE WORLD

#### Haut-parleur Neutrik

- pour les technologies professionnelles des hauts-parleurs/de la scène
- facilement verrouillable grâce à un système de verrouillage Quick-Lock
- protégé contre les contacts accidentels

#### Connecteur de câble avec borne à vis

NEUTRIK NL-2FC **3,15** 2 pôles  
NEUTRIK NL-4FC **5,40** 4 pôles

#### Prises encastrées - châssis

- pour connecteurs de câble 2 et 4 pôles
- Raccord Faston 4,8 mm

NEUTRIK NL-4MP **1,80**



#### Tous les connecteurs HF sous : <http://rch.it/4->

par exemple  
UG 88U-C59 **0,28** RG 59



#### Tous les connecteurs à broche <http://rch.it/50>

par exemple  
PFL 10 **0,08** 10 pôles



#### Tous les connecteurs D-Sub sous : <http://rch.it/51>

par exemple  
D-SUB BU 09 **0,12**  
D-SUB BU 25 **0,26**



#### Tous les capots entièrement métalliques D-Sub <http://rch.it/52>

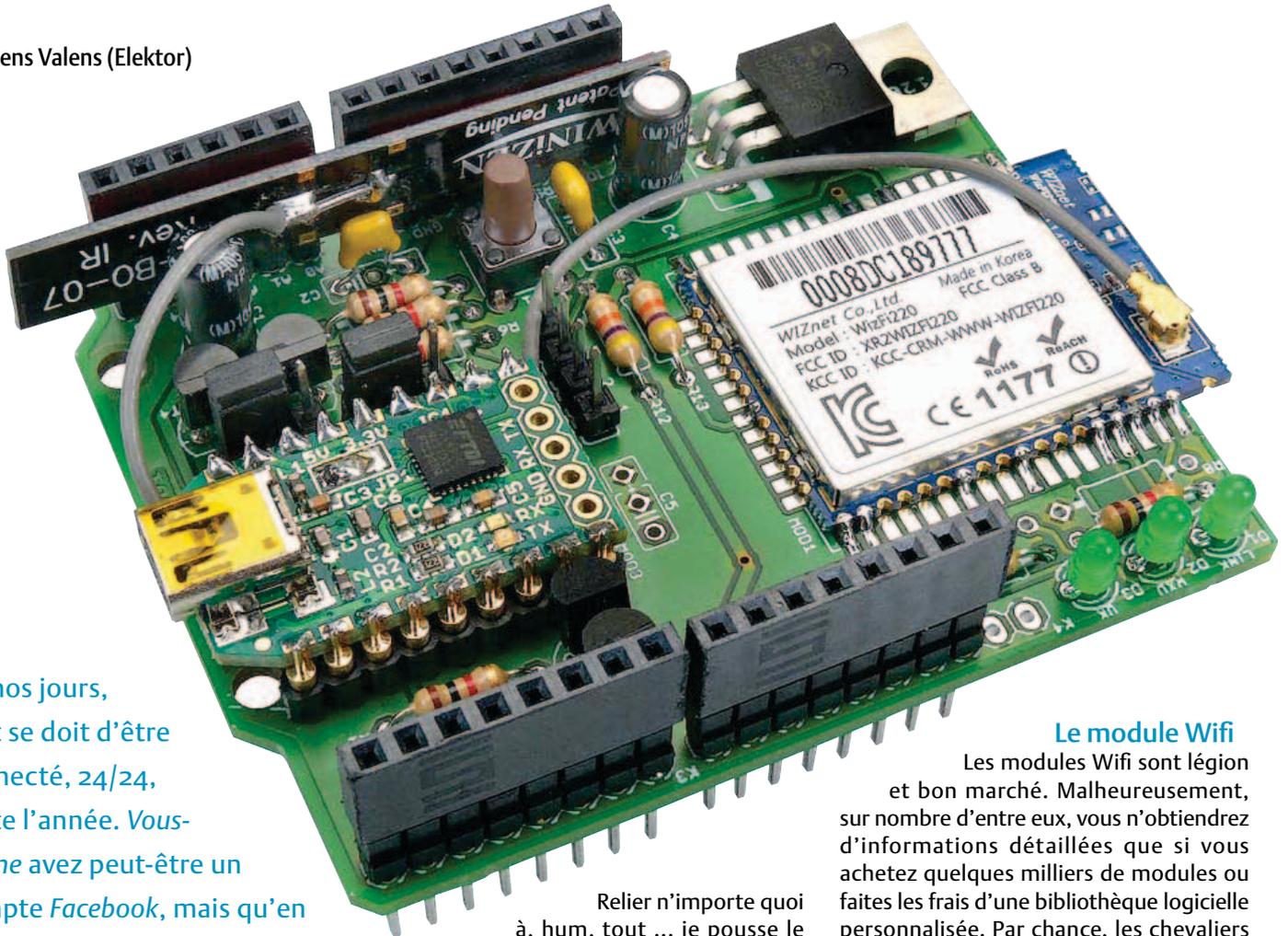
par exemple  
KAPPE EMV 09 **1,90**  
KAPPE EMV 25 **1,95**



# Arduino et Platino dans les nuages !

## shield Wifi/Bluetooth/USB pour mieux s'envoyer en l'air

Clemens Valens (Elektor)



De nos jours, tout se doit d'être connecté, 24/24, toute l'année. Vous-même avez peut-être un compte Facebook, mais qu'en est-il de votre oscilloscope ? Votre multimètre twitte-t-il assez ? Votre fer à souder est-il suffisamment branché ? Vous êtes peut-être un geek pas très bavard mais peut-être votre alimentation a-t-elle une folle envie de vie sociale. La carte d'extension décrite dans cet article permet de relier tout et n'importe quoi, avec ou sans fils.

Relier n'importe quoi à, hum, tout ... je pousse le bouchon un peu loin. J'ai la réputation d'exagérer, je ne le nie pas. Toujours est-il que la carte présentée ici est polyvalente. Équipée d'un module Wifi elle permettra de connecter un appareil à l'internet. Dotée d'un module Bluetooth, elle discutera avec ses congénères. Et pour les anciens, encore attachés aux câbles, il y a un pont série vers USB. Elle servira aussi de carte d'extension pour les systèmes à microcontrôleurs dotés de connecteurs du type Arduino (fig. 1), comme le Platino d'Elektor [1] (fig. 2), mais elle peut également être utilisée seule, comme un module sans-fil pour PC. Il n'est pas possible de monter simultanément les modules Wifi et Bluetooth sur la même carte.

### Le module Wifi

Les modules Wifi sont légion et bon marché. Malheureusement, sur nombre d'entre eux, vous n'obtiendrez d'informations détaillées que si vous achetez quelques milliers de modules ou faites les frais d'une bibliothèque logicielle personnalisée. Par chance, les chevaliers blancs de l'internet coréen de chez WIZnet sont moins cachotiers, et proposent des modules que l'on peut utiliser pour de vrai. J'ai réussi à mettre la main sur deux de leurs WizFi220, présentés dans cet article. Les modules WizFi2x0 (il existe un WizFi210) fonctionnent avec des points d'accès 802.11 b/g/n jusqu'à la vitesse de 11 Mbit/s (802.11b), et prennent en charge le WEP, le WPA, le WPA2-PSK et les Enterprise security standards (802.11i). Ils possèdent des fonctions vraiment sympa. L'une d'entre elles est le port série Wifi, remarquablement facile à utiliser. Une autre est leur capacité à se comporter comme des points d'accès (AP) minimalistes, permettant une connexion

directe aux tablettes et aux téléphones tactiles Android ou Apple.

Le module WizFi220 mesure 32 x 23,5 x 3 mm, possède une puce d'antenne intégrée et un connecteur Hirose U.FL pour les antennes externes. Le petit frère de ce module, compatible, est le WizFi210. Il consomme moins, mais ne délivre pas autant de puissance en sortie, ce qui réduit donc la portée. Les 49 pastilles de connexion fournissent suffisamment de points de soudure pour fixer le module solidement sur un PCB. C'est bien plus qu'il n'en faut aux modestes utilisateurs que nous sommes ; elles sont pour les gros clients. Pour les autres, un simple lien à deux fils suffit pour envoyer/recevoir les données et configurer le module.

Comme souvent avec de tels modules, le WizFi2x0 peut être commandé via un lien série à l'aide de commandes AT (ou commandes Hayes). Ces commandes sont composées de chaînes de caractères lisibles par un humain et commencent par « AT ». Le module se trouve en mode *commande* à la mise sous tension afin que vous puissiez le configurer. Une fois la configuration terminée, le module peut être basculé en mode *données* via la commande O(nline) : "ATO". Pour revenir au mode *commande*, il suffit d'envoyer trois caractères '+' à la suite soit "+++".

### La conception

Certaines des commandes de configuration, comme celles qui règlent le débit du port série, sont faites pour n'être émises qu'une seule fois, après quoi le module enregistre ces paramètres. Or, le fait que les commandes AT soient lisibles par un humain est si pratique non seulement pour la configuration initiale, mais aussi, par la suite, pour expérimenter en communiquant avec le module à l'aide d'un terminal série ; c'est la raison pour laquelle j'ai ajouté un convertisseur série vers USB. Comme beaucoup de lecteurs d'Elektor possèdent le fameux câble série/USB FTDI [2], j'ai fait en sorte qu'il soit utilisable avec ce module. D'un autre côté, le fait d'avoir un tel convertisseur directement sur la carte peut aussi être utile, surtout quand le système hôte ne possède pas de port série USB, ce qui est le cas de Platino. Dans le numéro de

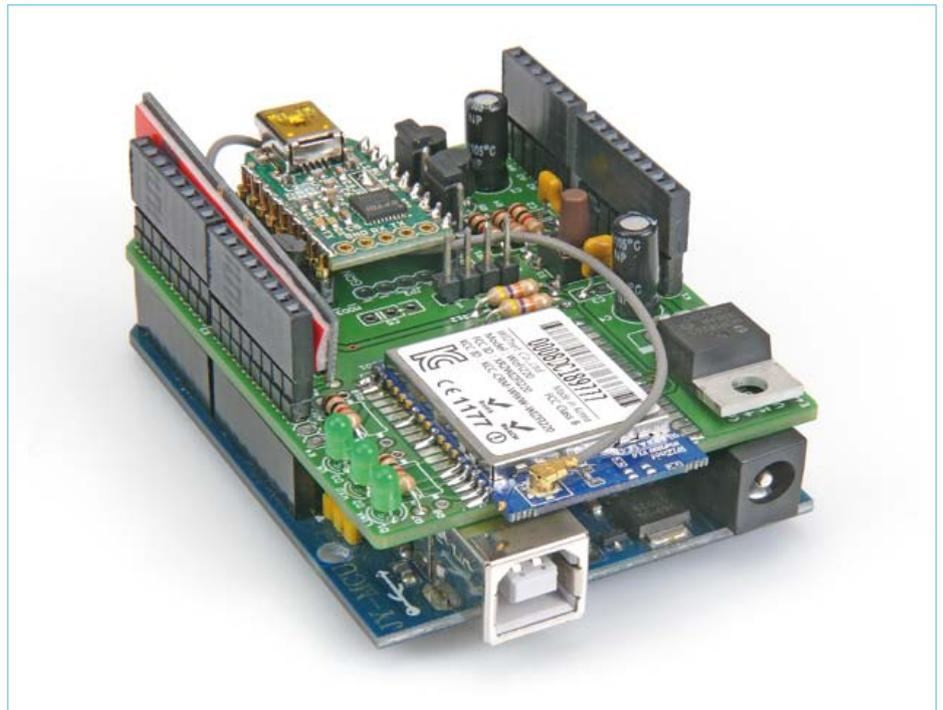


Figure 1. *Shield Wifi* complet monté sur un Arduino Uno. Remarquez l'absence de R8 et R10.

septembre 2011, nous vous présentions BOB [3], une passerelle USB/série très pratique et si populaire que j'ai jugé utile de l'incorporer également. Maintenant si j'arrive à trouver un câblage malin, il devrait être possible de relier le convertisseur USB soit au module Wifi, soit au système hôte ; voire de le déconnecter. Et c'est ainsi que

la flexibilité s'est insinuée dans mon projet. Le module Wifi doit être alimenté sous 3,3 V (et pas 5 V !). En général, le port série du module Wifi sera relié à un microcontrôleur, qui lui n'est pas forcément alimenté sous 3,3 V : avec Arduino ou Platino, c'est 5 V qu'il faut. L'adaptation des niveaux sur le port série est donc nécessaire. Le câble FTDI



Figure 2. *Shield Wifi* complet sur un Platino, doté également d'un afficheur alphanumérique à cristaux liquides 4x20.

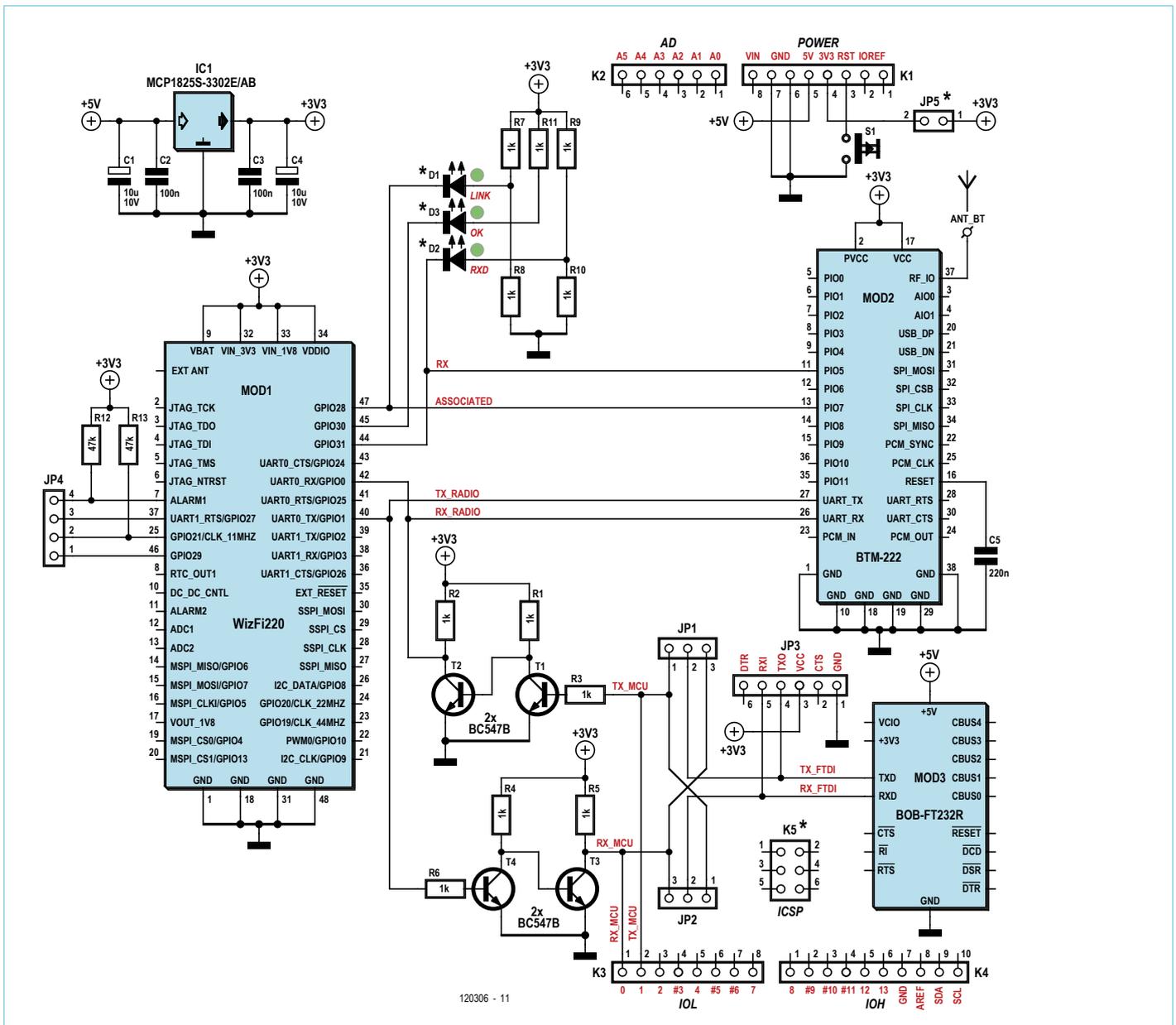


Figure 3. Schéma complet du *shield* Wifi/Bluetooth/USB. Référez-vous au texte pour connaître les composants à monter.

existe en deux versions, 5 V et 3,3 V, et une telle adaptation rendrait les deux utilisables. Pas de problème du côté du BOB : il prend en charge ces deux tensions.

Même si certains systèmes hôtes, comme Arduino, fournissent une alimentation sous 3,3 V, j'ai équipé la carte d'un régulateur de tension pour répondre, quelle que soit la configuration dans laquelle il est utilisé, aux besoins du module ; celui-ci peut en effet consommer jusqu'à 250 mA quand il est actif. Or, le régulateur LP2985-33 des cartes Arduino Uno ne fournit par exemple que 150 mA, ce qui est insuffisant. Quant à Platino, il ne fournit pas d'alimentation

3,3 V du tout. Un cavalier permet de sélectionner l'origine du 3,3 V : *shield* ou système hôte.

Remplacer « Wifi » par « Bluetooth » dans ce que je viens de dire, le problème est le même, car beaucoup de modules Bluetooth ressemblent au module Wifi : ils peuvent être commandés par des commandes AT sur un lien série, s'alimentent en 3,3 V, et possèdent des dimensions et besoins similaires. Il est facile d'ajouter du Bluetooth au système puisqu'il suffit de relier un module Bluetooth en parallèle avec le module Wifi. C'est ce que j'ai fait avec le module Rayson BTM220 que je connais

bien [4] et qui n'est vraiment pas cher. Nous l'avons déjà pas mal utilisé dans Elektor [5] [6].

Maintenant que vous savez comment j'ai raisonné en concevant ce système, regardons le schéma de plus près (fig. 3). Si vous avez capté ma prose, le schéma ne vous surprendra guère.

### L'assemblage

À gauche le module Wifi, à droite le Bluetooth. Vous voyez qu'ils partagent les lignes RX et TX du lien série, ce qui interdit de monter les deux ensemble. C'est soit Wifi, soit Bluetooth, jamais les deux. Ces

## LISTE DES COMPOSANTS

### Résistances

R1–R11 = 1 k $\Omega$  5% 0,25 W  
R12, R13 = 47 k $\Omega$  5% 0,25 W

### Condensateurs

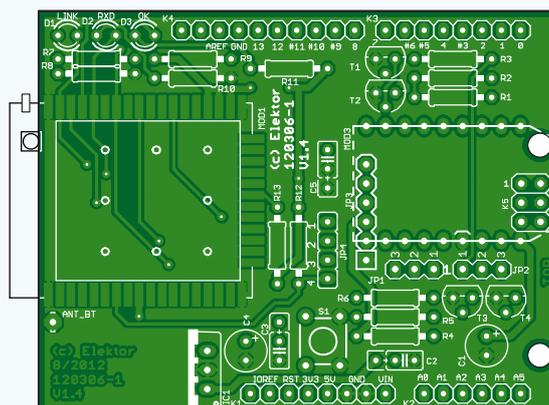
C2, C3 = 100 nF 50 V au pas de 5 mm  
C5 = 220 nF 50 V au pas de 5 mm  
C1, C4 = 10  $\mu$ F 35 V au pas de 2,5 mm

### Semi-conducteurs

D1, D2, D3 = LED vertes  $\varnothing$  3 mm  
T1–T4 = BC547C en boîtier TO-92  
IC1 = MCP1825S-3302E/AB, régulateur de tension 3,3 V, boîtier TO-220

### Divers

JP1 = barrette sécable 6 points au pas de 2,54 mm, droite  
JP3, JP4 = barrette sécable 3 points au pas de 2,54 mm, droite + 2 cavaliers  
JP2 = barrette sécable 4 points au pas



de 2,54 mm, droite  
K1, K3, K4 = connecteur empilables à 8 pattes au pas de 2,54 mm, verticaux  
K2 = connecteur empilables à 6 pattes au pas de 2,54 mm, verticaux  
K5 = connecteur empilables à 6 pattes (2 x 3) au pas de 2,54 mm, verticaux  
S1 = bouton poussoir, SPNO, traversant, 6 x 6 mm  
Mod3 = Adaptateur USB vers série BOB-FT232R, Elektor 110553-91, ou câble USB vers série, Elektor 080213-71 (5 V) ou 080213-71 (3,3 V)  
Mod2 = module Bluetooth Rayson BTM22x  
Mod1 = module Wifi WIZnet WizFi2x0 circuit imprimé 120306-1 [1]

modules partagent également deux LED (D1 et D2) ; D3 n'est utilisée que par le module Wifi. La surprise ici sera peut-être R8 et R10. Si j'avais mieux dessiné mon schéma, j'aurais moins de détails à expliquer. Rassurez-vous, ce n'est pas bien compliqué. Pour le Wifi, c'est le module qui draine le courant des LED, alors que pour le Bluetooth, le courant des LED est fourni par le module. Quand vous utilisez le module Bluetooth, il faut donc inverser le sens des LED. Dans ce cas, il faudra monter R8 et R10 et **ne pas monter** R7 et R9 (monter R11 et D3 est inutile à moins de relier la LED à une E/S libre sur un des connecteurs). Pour le module Wifi, la position des LED est celle du schéma et il ne faut **pas monter** R8 et R10 (en cas d'erreur, rien ne cramera), il ne faut que R7 et R9.

Vous voudrez peut-être utiliser des fonctions spéciales du module WizFi que je n'utilise pas, mais dont j'ai rendu les pattes correspondantes accessibles sur un connecteur séparé : J4. Afin de vous

permettre d'empiler les *shields* comme bon vous semble, j'ai décidé de ne pas câbler ces signaux sur les connecteurs d'extension Arduino K1 à K4. Si vous en avez besoin, câblez-les vous-même. La patte 25 (GPIO 21) permet de remettre le module dans sa configuration d'usine. En envoyant deux impulsions de niveau bas, le module passe en mode *Limited Access Point*, avec trois impulsions il repasse en mode *Ad hoc*. La patte 46 (GPIO 29) fournit un moyen matériel de passer du mode *commande* (état haut) au mode *données* (état bas). La patte 37 (GPIO 27) sert pour mettre à jour le micrologiciel du module (ce que vous ne ferez sans doute jamais). Il faut dans ce cas la tirer au niveau haut. Enfin, un niveau bas sur la patte 7 (Alarm1) permet de réveiller le module endormi.

Les adaptateurs de niveau (R1-3, T1-2 et R4-6, T3-4) se passent de commentaire ; ils ont déjà été utilisés dans Elektor. Les transistors T1 et T4 assurent la conversion de niveau, mais inversent le signal. T2

et T3 rattrapent le coup. Les valeurs des résistances ne sont pas critiques. J'ai choisi d'utiliser autant de résistances de 1 k $\Omega$  que possible.

Les lignes RX et TX sont reliées à JP1 et JP2. Le rôle de ces cavaliers à trois pattes n'est pas évident à saisir au premier coup d'œil. Suivez les fils et vous comprendrez. Avec un cavalier sur les pattes 1 et 2 de JP1, la patte TX du convertisseur USB série est reliée à l'entrée RX du module (Wifi ou Bluetooth). Avec un cavalier sur les pattes 2 et 3 de JP2, la patte RX du convertisseur est reliée à la sortie TX du module. C'est le mode de « configuration du module avec le PC ». Quand les cavaliers sont ainsi placés, le convertisseur USB série ne peut communiquer avec le système hôte. Mieux vaut en déconnecter la carte. Elle s'alimentera alors par le port USB.

Si vous placez un cavalier entre les pattes 2 et 3 de JP1 et entre les pattes 1 et 2 de JP2, le convertisseur USB/série peut être utilisé pour dialoguer avec le système hôte. Dans

## Produits et services Elektor

- Platine polyvalente pour montages à microcontrôleur(s) AVR (oct. 2011); circuit imprimé 100892-1
- Passerelle USB/série BOB-FT232R (sept. 2011); module 110553-91
- Câble Usb → Série TTL (juin 2008) 080213-71 (5 V) ou 080213-71 (3,3 V)

**Listing 1.**

Comment configurer le module WizFi2x0 en tant que serveur série à l'aide de commandes AT. Les commandes sont en **gras**, les réponses du module en *italique*. Les commentaires sont entre parenthèses (). Référez-vous au manuel du WizFi2x0 pour plus de commandes.

```

AT (réveil)
[OK]
AT+NSTAT=? (quel est ton état ?)
MAC=00:08:dc:18:97:76
WSTATE=NOT CONNECTED MODE=NONE
BSSID=00:00:00:00:00:00 SSID=»» CHANNEL=NONE SECURITY=NONE RSSI=0
IP addr=0.0.0.0 SubNet=0.0.0.0 Gateway=0.0.0.0 DNS1=0.0.0.0 DNS2=0.0.0.0
RxCount=0 TxCount=0
[OK]
AT+WPAPSK=germaine,»Philippe Noirette« (régler le SSID et la pass phrase)
Computing PSK from SSID and PassPhrase...

[OK]
AT+NDHCP=1 (demander une IP à un serveur DHCP)
[OK]
AT+WAUTO=0,germaine (se connecter automatiquement à germaine)
[OK]
AT+NAUTO=1,1,,8011 (paramétrer la connexion automatique : serveur, TCP, port 8011)
[OK]
ATA (démarrer la connexion automatique)
   IP SubNet Gateway
   192.168.2.7: 255.255.255.0: 192.168.2.1
[OK]

```

(vous pouvez maintenant vous connecter par telnet sur le module via le Wifi)

ce cas, il ne faudra pas monter le module sans-fil (ou les adaptateurs de niveau) qui se trouverai(en)t perturbé(s) par la communication. C'est le mode *shield* USB. Sans aucun cavalier, la carte sera en mode « sans-fil » et le convertisseur USB/série ne sert à rien. Vous pourriez toutefois le laisser et le câbler sur d'autres pattes des connecteurs d'extension. Cela peut par exemple servir dans un système où l'UART est émulée en logiciel sur l'Arduino ou sur un système possédant plus d'une UART. Pour finir, quelques remarques sur des détails. Le bouton *reset* a surtout été prévu pour Platino qui n'en dispose pas. Il est aussi utile avec un Arduino quand les *shields* rendent le bouton d'origine inaccessible. Cela dépend de la carte Arduino et de sa révision. K5 n'est pas relié. Je ne l'ai mis sur la carte que pour donner accès au signaux qu'il transporte dans le cas d'un hôte Arduino.

### Un peu de pratique

Pour transformer un schéma en *shield* ou en carte d'extension, il faudra un circuit imprimé (PCB). J'en ai conçu un dont vous téléchargerez les fichiers Eagle depuis la page de cet article sur notre site [7]. La taille du PCB est celle d'un Arduino Uno, toutefois sans la forme bizarre du petit côté de l'Arduino Uno. Tous les composants sont de type traversant ; seuls les modules Wifi et Bluetooth seront montés en surface. JP5 et le module Bluetooth se trouvent sur le côté cuivre ; les autres composants sont montés côté... composants. Si vous souhaitez monter la carte sur un Arduino ou un Platino et que vous utilisez pour K1 à K4 des connecteurs qui ne peuvent pas s'empiler, il faudra les monter côté cuivre. La position de JP1 et JP2 est telle que les cavaliers qu'ils supportent devront toujours être dans la même position, c.-à-d. les deux à gauche, les deux à droite ou aucun

des deux. En fait, vous pourriez laisser un cavalier sur JP2 au cas où vous voudriez espionner depuis le micro les échanges avec le module sans-fil. Cela peut être utile pour le dépannage.

JP3 possède quelques contacts en commun avec la passerelle série vers le BOB USB, l'idée étant d'utiliser un BOB ou un câble FTDI mais pas les deux.

Sur les systèmes où tout est en 3,3 V les adaptateurs de niveau et le régulateur de tension ne sont pas nécessaires. Dans ce cas, vous devrez utiliser deux fils de R3-JP1 à R2-T2 et de R6-MOD1 à R5-T3. La position des résistances R3 & R2 a été prévue pour faciliter l'installation de ces fils et permettre qu'ils soient courts. Vous devrez aussi court-circuiter JP5 de l'autre côté de la carte.

Comme précisé plus haut, l'orientation des LED dépend du module sans-fil. La sérigraphie du PCB correspond à l'utilisation d'un module Wifi. Dans le cas d'un module

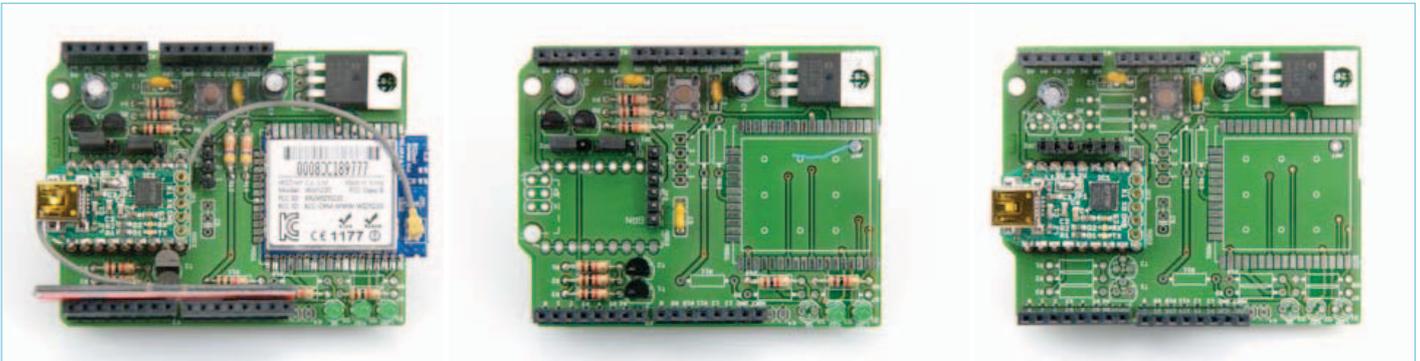


Figure 4. Les trois configurations principales les unes à côté des autres. Le module Bluetooth n'est pas visible (carte du milieu) puisqu'il est monté de l'autre côté du PCB. Vous remarquerez qu'il s'agit d'une première révision du PCB qui a la même forme qu'un Arduino Uno. Le PCB final est rectangulaire.

Bluetooth, il faudra monter les LED D1 et D2 « à l'envers ». Aussi, ne montez que R7 ou R8 et R9 ou R10. Le régulateur 3,3 V pourra être soudé ou rester droit.

Trois configurations standards sont possibles (fig. 4) :

- *shield* Wifi — pas besoin de BOB si vous possédez un câble FTDI, mais vous pourrez le monter quand même. JP3 sert à la connexion du câble FTDI. Montez les LED comme indiqué sur le PCB et ne montez ni R8 ni R10. Les adaptateurs de niveau seront nécessaires, sans doute aussi le régulateur 3,3 V, suivant le système hôte.
- *shield* Bluetooth — comme pour le *shield* Wifi sauf pour les LED. D3 n'a aucune utilité. Montez R8 et R10 au lieu de R7 et R9. Un bout de fil isolé de 31 mm fera une antenne convenable.
- *shield* adaptateur série vers USB — juste le BOB sans les adaptateurs de niveau. N'oubliez pas de mettre une goutte de soudure sur le cavalier JP1 du BOB pour choisir la tension, 5 V ou 3,3 V. Au choix, vous pourrez ajouter le régulateur 3,3 V et/ou le bouton *reset*. Les LED peuvent aussi être utiles mais il faudra les relier à un connecteur. Court-circuitez enfin les pattes 2 et 3 de JP1 et les pattes 1 et 2 de JP2 avec un cavalier ou un bout de fil.

### Testons, quand même !

Pour une utilisation en *shield* Wifi ou Bluetooth, reliez la carte à un port USB de votre PC à l'aide d'un câble FTDI ou de la passerelle BOB. Déterminez le nom du port série créé par votre OS (assurez-vous que les pilotes sont bien installés. Vous pourrez les trouver, ainsi que la documentation sur [www.ftdichip.com](http://www.ftdichip.com)) puis démarrez un émulateur de terminal. Le module WizFi220 utilise une vitesse par défaut de 115 200 bits/s. Pour le module BTM220, c'est 19 200. Les deux modules utilisent des mots de 8 bits, pas de bit de parité et aucun contrôle matériel du flux. Dans le terminal, tapez 'AT' puis appuyez sur Entrée. Si vous avez tout réglé correctement, le module vous répondra par '[OK]' ou 'OK'. Si ce test réussit, vous êtes tiré d'affaire. Référez-vous au **listing 1** pour un exemple d'utilisation du Wifi (n'oubliez pas d'adapter le SSID et la *passphrase* à votre réseau).

Pour une utilisation en adaptateur série vers USB, les étapes sont plus ou moins les mêmes à la différence près qu'il ne sert à rien de balancer des commandes AT depuis votre terminal, à moins bien sûr, que le système hôte n'y réponde. C'est à vous de fournir un système hôte capable de communiquer via une liaison série.

(120306 – version française : Kévin PETIT)

Merci à Joachim Wulbeck de WIZnet Europe GmbH ([www.wiznet.eu](http://www.wiznet.eu)) pour m'avoir fourni les modules WizFi220 et les antennes Wifi.

### Liens et références

- [1] **Platino** : [www.elektor.fr/100892](http://www.elektor.fr/100892)
- [2] **câble série USB vers TTL** : [www.elektor.fr/080213](http://www.elektor.fr/080213)
- [3] **adaptateur série vers USB BOB** : [www.elektor.fr/110553](http://www.elektor.fr/110553)
- [4] **expériences avec les modules Bluetooth Rayson** : <http://elektorembded.blogspot.com/2010/08/rayson-btm222-btm112-bluetooth-modules.html>
- [5] **Bluetooth sur l'ATM18** : [www.elektor.fr/080948](http://www.elektor.fr/080948)
- [6] **Bluetooth et OBD-2** : [www.elektor.fr/090918](http://www.elektor.fr/090918)
- [7] **shield Wi-Fi** : [www.elektor.fr/120306](http://www.elektor.fr/120306)
- [8] **shield Wi-Fi sur Elektor Projects** : <http://www.elektor-projects.com/project/wi-fi-bluetooth-usb-shield-for-arduino-platino.12252.html>

Publicité

**SCOPES** and more  
**HAMEG**<sup>®</sup>  
 Instruments  
 A Rohde & Schwarz Company

Great Value in  
**TEST & MEASUREMENT**

# mettez le cap sur Arduino

## 2: capteurs (très) distants par GSM / GPRS

Pour voir ce que la communication par GSM / GPRS peut apporter à vos projets d'électronicien, nous jonglerons ici avec de nouveaux concepts tels que l'IdO (pour *internet des objets*) ou le Màm (pour *machine à machine*). Je ne cache pas mon intention de vous épater. N'hésitez pas à réagir et à donner votre avis\*.

David Cuartielles (Espagne)

### D'abord les courses

Voici ce dont vous aurez besoin pour reproduire les expériences décrites dans cet article :

- une carte Arduino Uno
- un *shield* GSM Arduino (fabriqué en collaboration avec Telefonica R&D)
- une carte SIM pour le réseau GSM/GPRS
- un téléphone portable doté de sa propre carte SIM
- le *shield* TinkerKit
- des modules LED TinkerKit LED (x3)
- un module Slider TinkerKit

Les trois derniers ingrédients pourront être remplacés par une plaque d'essais garnie des composants adéquats. De mon côté, j'ai préféré opter pour du matériel tout fait pour me concentrer sur le fonctionnement de la bibliothèque GSM pour Arduino.

Maintenant, et avant de poursuivre votre lecture, je vous conseille fortement de télécharger tous les codes source que nous allons utiliser. A cause de leur longueur, je n'ai pu en représenter ici que des extraits.

### Màm et IdO : à vos souhaits !

En quelques mots, l'IdO est un paradigme informatique que sont en train d'explorer sur le tard beaucoup de laboratoires de recherche et d'universités de par le monde. L'idée qui en est à la base est que tout peut être relié à internet : de votre micro-ondes à votre voiture,



en passant par le verrou de votre porte et votre vélo. Tous ces objets devraient être capables de communiquer et s'échanger des données afin d'améliorer la vie quotidienne.

Ericsson, géant suédois de la communication, prévoit [1] 50 milliards de périphériques connectés en 2020. Cela veut dire que la plupart de nos appareils domestiques seraient reliés à l'internet : frigos et micro-ondes, réveils et TV, mais aussi l'ascenseur et la cuisinière, et tous les autres appareils qui nous entourent comme les voitures, les lampadaires, les caddies (ceux qu'on pousse, pas ceux sur lesquels on clique).

Le Màm regroupe l'ensemble des techniques de communication entre machines, qu'elles soient sans-fil, filaires ou hybrides. D'une certaine manière, le Màm rend l'IdO possible en offrant l'infrastructure dont les appareils ont besoin pour collecter des données et les transmettre sur le réseau. Certains voient le Màm comme un synonyme de la télémétrie, mais dernièrement le concept a muté et se réfère désormais plus aux communications sans fil et plus précisément encore à l'utilisation des réseaux de téléphonie mobile pour commander des appareils à distance.

Pour simplifier, j'utiliserai les acronymes IdO pour parler du lien entre les appareils et l'internet et Màm lorsque les données passent par le réseau de téléphonie mobile. Cette terminologie apparaît dans beaucoup d'articles, sans doute parce que le réseau le plus étendu aujourd'hui est le réseau GSM (*global system for mobile communications*).

Supposons que vous préparez la réalisation d'un appareil relié à l'internet, qui vous enverrait des données depuis n'importe quel point du globe ; à votre avis, quelle serait la meilleure méthode de communication ? La réponse est évidente, c'est la téléphonie mobile. Le GSM est un formidable moyen de contacter votre appareil presque partout dans le monde sans vous soucier du comment, un peu comme s'il était à la maison.

### Le *shield* GSM/GPRS Arduino

Nous autres, contributeurs à Arduino, cherchons à vous simplifier la construction de prototypes et l'apprentissage des technologies. Nous passons donc pas mal de temps à réfléchir aux meilleures

\* [redaction@elektor.fr](mailto:redaction@elektor.fr)

manières de vous convaincre de l'intérêt de l'IdO. L'une d'elles est le M2M pour lequel nous avons fait équipe avec le *Physical Internet Laboratory* de Telefonica, pour créer un *shield* Arduino officiel qui gère la communication par GSM/GPRS (ou 1G/2G). Nous avons également écrit une bibliothèque pour simplifier l'envoi de messages textuels (SMS), les appels vocaux et la publication de données sur un site. Le résultat est à sources ouvertes. Ce logiciel pourra aisément être adapté pour prendre en charge d'autres *shields* (utilisant potentiellement d'autres commandes AT ; voir plus loin). Enfin, le *shield* est livré avec une carte SIM, grâce à la générosité de Telefonica, qui vous permettra (Ndlr : en Espagne) de vous connecter au réseau, envoyer/recevoir des SMS, passer/recevoir des appels, et vous connecter à l'internet via le réseau GPRS.

*Remarque* : Le *shield* n'est pas couplé à la carte SIM fournie ; vous pourrez utiliser n'importe quelle autre carte SIM GSM/GPRS disponible dans votre pays. La SIM est optimisée pour les communications de données ; pour créer un appareil commandé par SMS, vous devrez sans doute utiliser une carte SIM fonctionnant sur votre réseau favori. Pour la carte fournie avec le *shield*, n'oubliez pas de suivre la procédure d'activation en ligne, selon les directives fournies.

### Commander un modem – les commandes AT

Les modems sont à l'origine de l'expansion d'internet. Ils alternent deux modes de fonctionnement : commandes et données. Afin d'automatiser leur fonctionnement, une entreprise appelée Hayes [2] a inventé en 1981 ce que l'on appelle désormais le jeu de commandes AT. Le mode commande permet d'effectuer des opérations sur le modem lui-même comme composer un numéro, changer la vitesse de communication, etc.

Pour la communication avec les périphériques via un port série, le jeu de commandes Hayes est devenu la norme de facto. Elle est toujours utilisée par beaucoup de périphériques. En mode commande, la communication commence par l'envoi de la chaîne «AT» (pour «attention»), suivie par d'autres chaînes représentant les commandes à exécuter. À l'inverse, en mode données, les données sont simplement transférées depuis ou vers l'internet à travers le modem.

Le *shield* GSM/GPRS possède un modem radio fabriqué par Quectel [3] qui répond aux commandes AT. Celles-ci sont pour la plupart spécifiques au modem qui équipe la carte, mais la bibliothèque qui commande le *shield* a été écrite de manière à faciliter l'adaptation à d'autres modems d'autres fabricants. Aussi cette bibliothèque masque-t-elle toute la complexité de fonctionnement. Par conséquent, nous ne parlerons plus de commandes AT dans le reste de l'article. Si toutefois vous utilisez la bibliothèque GSM en mode *debug*, il faudra vous frotter à ces commandes.

### Installer la bibliothèque GSM pour Arduino

Au moment où j'écris ces lignes, l'EDI Arduino a atteint la version 1.0.1 et n'embarque pas la bibliothèque GSM dont nous allons parler. Il vous faudra donc, si vous ne voyez pas apparaître « GSM3 » dans le menu « Sketch / Import Library », la télécharger depuis le site

Arduino officiel hébergeant ce projet [4] et l'installer en suivant les instructions de la page « Library ». En gros, l'ajout d'une nouvelle bibliothèque à l'EDI se fait en créant un dossier nommé 'libraries' dans votre *sketchbook* et y décompresser le fichier téléchargé depuis le site. Redémarrez ensuite l'EDI Arduino et la bibliothèque devrait être disponible depuis le menu mentionné.

En installant la bibliothèque, vous installerez non seulement le code permettant d'exécuter des commandes sur le modem, mais aussi une série d'exemples qui vous permettront de :

- vérifier que le modem fonctionne convenablement
- envoyer et recevoir des SMS
- passer et recevoir des appels vocaux
- ouvrir une connexion TCP/IP et échanger des données via internet, par le réseau GPRS
- poster et recueillir des informations de capteurs sur/ depuis Cosm (un service en ligne)
- recueillir et afficher des messages *twitter*

Rien de plus facile que d'accéder aux exemples : utilisez le menu pour vous rendre dans « File / Examples / GSM3 ».

### 3615 VERIF

Le plus sage est de commencer par vérifier que votre carte fonctionne correctement et que votre SIM est compatible avec le *shield*. Pour ce faire, utilisez l'exemple suivant : « File / Examples / GSM3 / Tools / TestModem ».

```
// bibliothèque
#include <GSM3ShieldV1ModemVerification.h>

// objet de vérification du modem
GSM3ShieldV1ModemVerification modemTest;

// variable IMEI
String IMEI = «»;

// messages pour le moniteur série
String oktext = «OK»;
String errortext = «ERROR»;

void setup()
{
  // initialisation de la communication série
  Serial.begin(9600);

  // lancement du test modem (reset et vérification de
  la réponse)
  Serial.print(«Starting modem test...»);
  modemTest.begin();
  Serial.println(oktext);
}
void loop()
```

```

{
  // obtention de l'IMEI du modem
  Serial.print("Checking IMEI...");
  IMEI = modemTest.getIMEI();

  // vérification de l'IMEI reçu
  if(IMEI != NULL)
  {
    Serial.println(oktext);
    // affichage de l'IMEI sur la liaison série
    Serial.println("Modem's IMEI: « + IMEI);
    // remise à zéro du modem et vérification du
    démarrage
    Serial.print("Reseting modem...");
    modemTest.begin();
    // re-vérification de l'IMEI
    if(modemTest.getIMEI() != NULL)
    {
      Serial.println(oktext);
      Serial.println("TEST COMPLETE!");
    }
    else
    {
      Serial.println(errortext);
    }
  }
  else
  {
    Serial.println(errortext);
  }
  while(true);
}

```

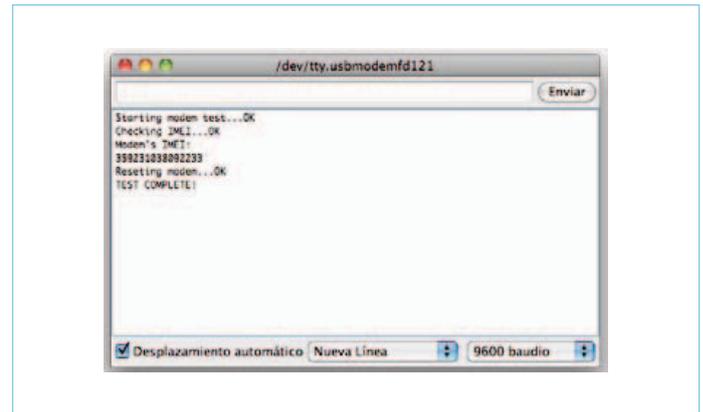


Figure 1. L'exemple TestModem a fonctionné : votre modem est opérationnel.

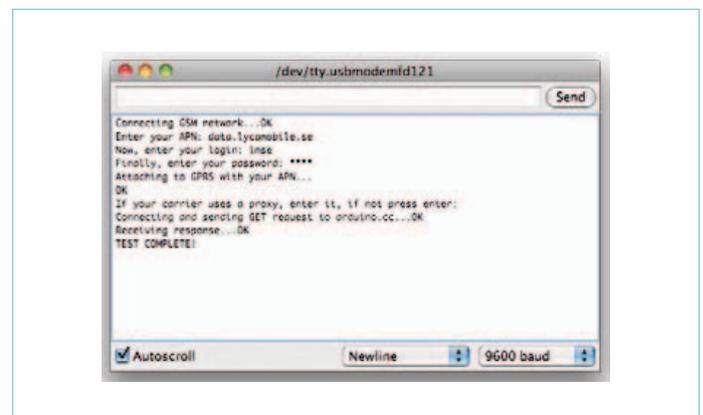


Figure 2. Configurez votre connexion GPRS avec le moniteur série.

Ce code d'exemple vous permettra de vérifier que votre Arduino détecte bien le modem radio. Il tentera également d'obtenir l'IMEI (*International Mobile Equipment Identity*) du modem sur votre *shield* GSM — il s'agit du numéro unique de chaque appareil capable de se connecter à un réseau GSM/GPRS/3G/4G. Par exemple, votre portable s'identifie via un IMEI. Chaque appareil sur le réseau GSM possède un identifiant unique. Si tout se passe bien, ce que vous verrez dans le moniteur série devrait ressembler à la **figure 1** après avoir chargé le code sur votre Arduino Uno.

Si votre modem fonctionne correctement, la prochaine chose à vérifier est la connexion au réseau GPRS. Voici le code d'exemple à utiliser : *File / Examples / GSM3 / Tools / TestGPRS*. Ce code ressemble beaucoup au précédent et n'a donc pas été reproduit ici. Il vous permettra de vérifier que la carte SIM autorise la connexion au réseau GPRS (celle fournie avec le *shield* le permet), mais aussi de vérifier vos réglages.

*Remarque* : ce *shield* utilise le GSM/GPRS et pas la 3G. Les cartes SIM GSM et 3G se ressemblent, mais rappelez-vous que certains opérateurs ne fournissent pas de services 2G (ce que nous appelons GPRS).

La connexion au réseau GPRS requiert un peu plus de configuration que l'envoi de SMS ou l'émission d'appels. Quatre paramètres doivent être configurés :

- *APN* ou *Access Point Name* : c'est le nom du domaine qui servira d'intermédiaire au *shield* pour atteindre l'internet. Ce

- paramètre dépend de votre opérateur.
- *login* : certains opérateurs utilisent un identifiant
- *password* : certains opérateurs utilisent un mot de passe
- *proxy* : c'est l'adresse d'un serveur qui relayera vos communications à l'intérieur du réseau de l'opérateur. Vous n'en aurez sans doute pas besoin.

Ce code d'exemple vous demandera de saisir les informations dont le *shield* a besoin pour se connecter au réseau.

*Conseil* : prenez soin d'activer l'option 'Newline' dans le menu en bas du moniteur de port série. Ce faisant, le système ajoutera un caractère de fin de ligne (EOL), lorsque vous cliquerez sur 'Send' ; c'est nécessaire pour que le modem puisse repérer la fin des données.

Si tout se passe bien, la sortie ressemblera à la **figure 2**. Ces messages signifient que votre *shield* est en mesure de se connecter à l'internet à l'aide de votre SIM. Même si vous ne pouviez pas vous connecter à l'internet, remarquez qu'il reste possible d'envoyer/recevoir des SMS et des appels.

## Envoyer des SMS

Avec le *shield* GSM, rien de plus facile que d'envoyer et recevoir des SMS. Ces quelques lignes de code permettent d'envoyer une chaîne de caractères à un numéro donné.

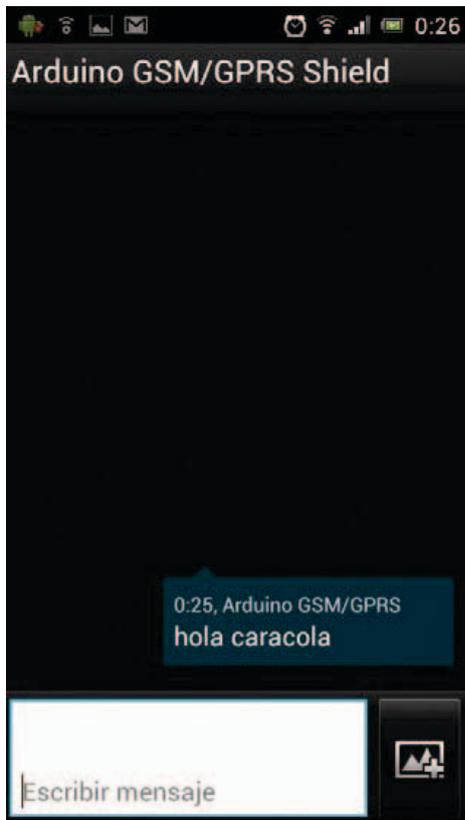


Figure 3. L'écran de mon portable après réception du SMS.

```
// envoi du message
sms.beginSMS(«0755442200»);
sms.print(«hola caracola»);
sms.endSMS();
```

Vous remarquerez que le numéro de téléphone est fictif ! Le résultat sur mon téléphone ressemble à la **figure 3**.

Les exemples fournis avec la bibliothèque couvrent à la fois l'envoi et la réception. Par exemple, le code *File / Examples / GSM3 / Tools / SendSMS* enverra un SMS depuis la carte interactivement via le moniteur de port série. Il vous demandera d'abord le numéro de téléphone puis le texte à envoyer. Essayez-le !

### Commande à distance par SMS

La commande à distance implique d'être capable de recevoir des messages, analyser leur contenu, et réagir aux différentes commandes. Nous allons pour l'exemple construire un petit prototype comprenant trois LED : une rouge, une verte et une bleue. Si l'on remplace les LED par des relais, il devient possible de commander n'importe quel type d'appareil à distance. Le programme lira le premier caractère du SMS et illuminera la diode correspondante : R pour rouge, G pour vert et B pour bleue. Ce programme est, comme vous l'aurez remarqué, très simple. Pas

d'analyse complexe du message ; il se contente de lire le premier caractère du SMS pour décider de ce qu'il devra faire. Après analyse, le SMS sera effacé de la mémoire à l'aide de la commande `sms.flush()`.

Voici un extrait du programme `DecodeSMS` que vous pourrez télécharger depuis [5].

```
// Y-a-t-il des SMS à lire ?
if (sms.available())
{
  Serial.println(«Message received from:»);

  // Obtention du numéro
  sms.remoteNumber(remoteNumber, 20);
  Serial.println(remoteNumber);

  // Voici un exemple de filtrage
  // Les messages commençant par un # seront écartés
  if (sms.peek()!='#')
  {
    Serial.println(«Discarded SMS»);
    sms.flush();
  }

  // Lecture et affichage du contenu du message
  int count = 0;
  while (c=sms.read()) {
    if (!count) {
      digitalWrite(pinR, LOW);
      digitalWrite(pinG, LOW);
      digitalWrite(pinB, LOW);

      switch (c) {
        case 'R':
          digitalWrite(pinR, HIGH);
          break;
        case 'G':
          digitalWrite(pinG, HIGH);
          break;
        case 'B':
          digitalWrite(pinB, HIGH);
          break;
      }
    }
    count++;
  }

  Serial.println(«\nEND OF MESSAGE»);

  // effacement du message de la mémoire du modem
  sms.flush();
  Serial.println(«MESSAGE DELETED»);
}
```

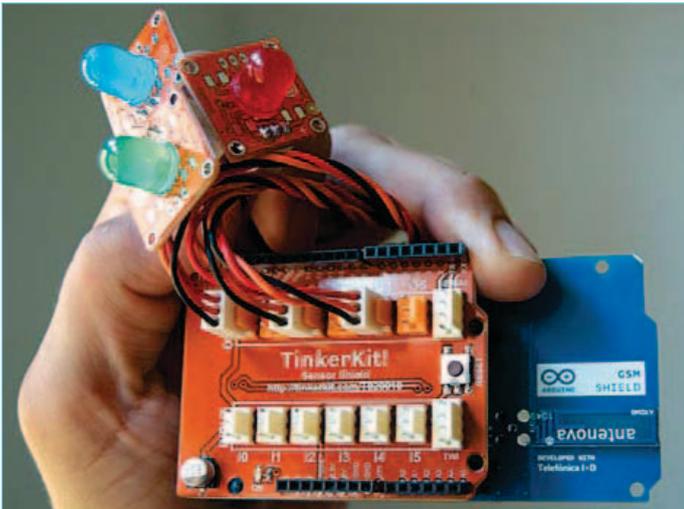


Figure 4. Arduino Uno + shield GSM/GPRS + TinkerKit shield + TinkerKit LED.

Les platines utilisées sont visibles en **figure 4**.

### Et par le GPRS ?

Essayons maintenant d'envoyer les données issues d'un capteur analogique (un curseur dans notre cas) vers un site web. Côté serveur, j'utilise Apache et un script PHP très simple qui, à partir des données envoyées par la carte, produira du HTML. Voici ce que j'aime appeler « une cellule mémoire en ligne ». Il s'agit d'un script PHP qui reçoit des données et les affiche. Vous pourrez accéder à cette même page depuis votre navigateur et regarder l'évolution des données affichées. Utilisez le même triplet APN/login/pass que lors des tests GPRS que nous avons faits un peu plus haut. Le code suivant envoie une valeur analogique à un serveur toutes les dix secondes :

```
void loop()
{
  client.connect(server, 80);
  Serial.println("sending data...");
  client.print("GET «);
  client.print(path);
  client.print(query);
  client.print(analogRead(A0)); // envoi de la
                                // valeur lue sur A0

  client.println(" HTTP/1.1");
  client.print("HOST: ");
  client.println(server);
  client.println();
  client.stop();
  delay(10000);
}
```

Il vous faudra posséder un serveur pour faire tourner le code PHP qui réceptionnera les données envoyées depuis votre Arduino. Le serveur devra être accessible publiquement si vous voulez que votre Arduino, votre navigateur, votre portable ou je-ne-sais-quel appareil connecté puisse y accéder.

**Important** : rappelez-vous que ce code n'est qu'une démo, il n'est en aucun cas sécurisé et vous devriez sans doute désactiver le script PHP après vos essais.

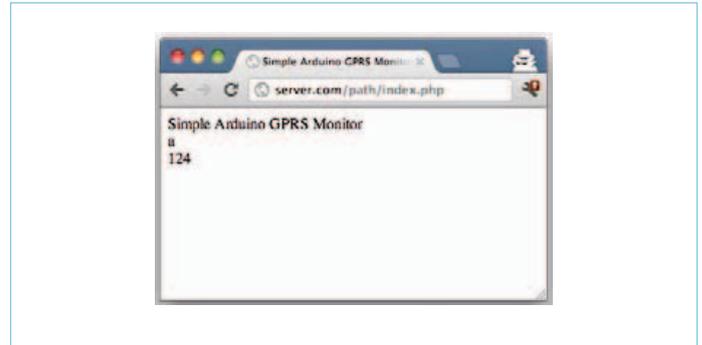


Figure 5. Les informations envoyées depuis le shield GSM dans votre navigateur.

Tout ce qu'il vous reste à faire, c'est décompresser le code de la cellule mémoire PHP que vous aurez téléchargé quelque part en dessous de la racine de votre serveur http. Assurez-vous que le fichier est doté de droits en lecture-écriture-exécution afin qu'il soit autorisé à s'écraser. Entrez ensuite l'URL de votre serveur (p ex. : <http://server.com/path/index.php>) dans votre navigateur et admirez les informations qui se mettent à jour toutes les 10 s. Comme sur la **figure 5**.

### Le mot de la fin

J'espère que vous aurez trouvé cette brève présentation de l'utilisation du GSM/GPRS à la fois enrichissante et amusante. Vous ai-je convaincu\* ? Mesurez-vous l'intérêt que présente pour nous, bidouilleurs invétérés, le fait de disposer d'un réseau sans fil omniprésent ? Si vos appareils ne consomment pas trop de données, vous pouvez d'ores et déjà, à travers le réseau de téléphonie mobile, commander toutes sortes de fonctions inattendues. Tant que vous avez de quoi alimenter votre carte, vous pouvez collecter et traiter des données presque partout.

(120506 – version française : Kévin PETIT)

### Liens

- [1] **Le mémo d'Ericsson sur les 50 milliards d'appareils connectés :** [www.ericsson.com/res/docs/whitepapers/wp-50-billions.pdf](http://www.ericsson.com/res/docs/whitepapers/wp-50-billions.pdf)
- [2] **A propos du jeu de commandes AT :** [http://en.wikipedia.org/wiki/Hayes\\_command\\_set](http://en.wikipedia.org/wiki/Hayes_command_set)
- [3] **Description des commandes AT de Quectel :** <http://datasphere.eu/en?t=/documentManager/sfdoc.file.supply&fileID=1285079825955>
- [4] **Ce site Arduino héberge les informations relatives au shield GSM :** <http://labs.arduino.cc/GPRS/Index>
- [5] **Tous les exemples et images :** [www.elektor.fr/120506](http://www.elektor.fr/120506)

### Remerciements

L'auteur remercie toute l'équipe du Physical Internet Laboratory de Telefonica R&D, pour son aide qui a rendu possible ce nouveau shield Arduino. Un merci tout particulier à F. J. Zorzano pour les nombreuses soirées de débogage.

\* [redaction@elektor.fr](mailto:redaction@elektor.fr)

*Le génie attire.  
Ressez l'attraction.*

DU MARDI 8 JANVIER AU VENDREDI 11 JANVIER 2013 \* LAS VEGAS, NEVADA \* ENREGISTREMENT SUR [CESWEB.ORG](http://CESWEB.ORG)



ENREGISTREZ-VOUS  
MAINTENANT



## Liste des composants

### Résistances (0603) :

R1 = 100 Ω  
R2, R3 = 2k2  
R4, R5 = 4k7

### Condensateurs :

C1, C2 = 100 nF (16 V, 0603)  
C3, C4 = 1 μF (16 V, tantale, SMC-A)

### Semi-conducteurs :

IC1 = MMA7455LR1 (LGA 14)  
T1...T4 = BSS138 (SOT-23)

D1 = BZV55-C3V0, 115 (SOC-80C)

### Divers :

K1, K2 = barrettes de broches 4 contacts (4x1)

convertissent les niveaux logiques des signaux SCL et SDA entre 3,3 V et 5 V de façon à ce que les broches I<sup>2</sup>C (SCL et SDA) soient compatibles avec 5 V. Les signaux I<sup>2</sup>C sont alors prêts à être utilisés sur les broches correspondantes de K2.

Relier la broche 7 d'IC1 (I2C\_EN) à la sortie 3,3 V place la puce en mode I<sup>2</sup>C. Les sorties pour les interruptions, broches 8 et 9, attaquent chacune les grilles des transistors

T3 et T4. Les signaux d'interruption respectifs apparaissent donc inversés en K1, mais les sorties n'en restent pas moins souples à utiliser (que ce soit p. ex. avec des contrôleurs fonctionnant en 3,3 V ou en 5 V). Le cavalier JMP1 (réalisé par soudure entre pastilles) positionne le premier bit de l'adresse I2C [1] du composant, une fonction utile lorsque plusieurs puces I<sup>2</sup>C sont en jeu.

L'écartement des broches des barrettes K1 et K2 correspond à l'espacement des cartes de prototypage. K1 et K2 sont séparées d'environ 15 mm, soit la distance standard entre lignes de broches d'un support DIP.

(090535 – version française : Hervé Moreau)

[1] [www.freescale.com/files/sensors/doc/data\\_sheet/MMA7455L.pdf](http://www.freescale.com/files/sensors/doc/data_sheet/MMA7455L.pdf)

## Produits & Services Elektor

• Platine Elektor équipée de composants CMS (090535-91).

Ce produit est disponible sur : [www.elektor.fr/090535](http://www.elektor.fr/090535)



## Elektor Electronic Toolbox



### Enfin une app utile pour les électroniciens, conçue par des électroniciens

La nouvelle application *Elektor Electronic Toolbox* répond aux questions des électroniciens et à leur besoin d'information rapide dans la vie quotidienne. 33 applications sont réunies sous un écran d'accueil commun et donnent accès à des banques de données pour les semi-conducteurs discrets (transistors bipolaires, FET, triacs, thyristors, diodes) ou intégrés. Pour retrouver en un éclair un composant et ses caractéristiques, il suffit de taper sa référence. Pas de connexion internet requise, toutes les informations sont en mémoire pour rien moins que 45.000 composants ! Une banque de données annexe donne le brochage d'une foule de connecteurs, notamment dans les domaines Audio & Vidéo, informatique et téléphonie. Une autre application fort utile permet de calculer la valeur des composants, dans les filtres, les diviseurs, les régulateurs, les étages à transistors, à amplificateurs opérationnels etc. D'autres font pour vous les conversions entre systèmes de numération, entre unités de grandeur, fréquences, longueurs d'ondes etc. Sans oublier l'inévitable code des couleurs et le tableau des symboles utilisés en électronique.

Votre nouvelle app *Elektor Electronic Toolbox* pour iPhone, iPod et iPad ne coûte que 4,99 €.



## FACES AVANT ET BOÎTIERS

Pièces unitaires et petites séries à prix avantageux.

A l'aide de notre logiciel – *Designer de Faces Avant\** – vous pouvez réaliser facilement votre face avant individuelle. **GRATUIT**: essayez-le! Pour plus de renseignements, n'hésitez pas à nous contacter, **des interlocuteurs français** attendent vos questions.

\*Vous en trouverez la dernière version sur notre site internet.

- Calcul des prix automatique
- Délai de livraison: entre 5 et 8 jours
- Si besoin est, service 24/24



Exemple de prix: 34,93 € majoré de la TVA/ des frais d'envoi

Schaeffer AG · Nahmitzer Damm 32 · D-12277 Berlin · Tel +49 (0)30 8 05 86 95-30  
Fax +49 (0)30 8 05 86 95-33 · Web [info.fr@schaefter-ag.de](mailto:info.fr@schaefter-ag.de) · [www.schaefter-ag.de](http://www.schaefter-ag.de)

# compteur horaire pour 68 ans au maximum



Vous est-il arrivé de vous demander combien de temps un appareil électrique a été en marche et a consommé du courant du secteur ? Le circuit simple présenté ici compte les secondes des périodes de marche et retient le temps cumulé dans l'EEPROM intégrée du microcontrôleur ATtiny 13. Pour connaître l'« âge » réel de votre matériel, il suffit de lire le contenu de l'EEPROM avec un programmeur ISP approprié ou le circuit de lecture *ad hoc*.

Vladimir Mitrovic (Croatie)

## Mouchard

On suppose que le matériel est alimenté par le secteur alternatif, pour permettre au microcontrôleur de compter les périodes de la tension alternative du secteur. Les bornes d'entrée seront raccordées derrière l'interrupteur (ou relais) marche-arrêt de l'appareil, ou au secondaire d'un transformateur d'alimentation. Toute tension supérieure à 4 V (alterna-

tif, 50 ou 60 Hz) conviendra si R1 est choisie correctement.

Venons-en au schéma du compteur d'impulsions : la zener D2 écrête la tension alternative en impulsions à peu près carrées de quelque 4 à 5 V crête à crête. Ces impulsions ne sont pas seulement l'horloge du compteur, mais aussi l'alimentation du circuit à travers D1/C1. Peu après l'application de l'alimentation (CA), la tension aux bornes de C1 atteint la valeur à partir de laquelle le microcontrôleur commence à exécuter

le programme qu'il a en mémoire. Au départ, le programme lit dans l'EEPROM le temps accumulé précédemment, puis il commence à compter les impulsions appliquées à l'entrée PB3 de l'ATtiny13. Chaque transition est reconnue, si bien qu'il y aura 100 ou 120 impulsions par seconde, suivant la fréquence du secteur (50 ou 60 Hz). Dès qu'une transition a été comptée, le microcontrôleur se met en sommeil (mode *Idle*). Ainsi, sa consommation moyenne reste en dessous de 0,5 mA.



## Stop ! Calculez d'abord R1

Le microcontrôleur a besoin de moins de 1 mA pour fonctionner correctement, et il faut au moins 1 mA à travers la zener D2 pour qu'elle joue son rôle de stabilisateur. Il faut donc un courant de 2 mA au moins à travers R1. Sa valeur maximale se calcule suivant la formule :

La constante 3,5 dans la formule représente la tension estimée de la zener D2. Sous une intensité aussi faible, attendez-vous

toujours à une tension de zener inférieure à la valeur nominale. Le tableau ci-dessous donne quelques valeurs expérimentales de R1. Vous pouvez toujours choisir une valeur plus faible, au prix d'une plus forte consommation d'énergie. Avec les valeurs du tableau, la consommation varie de 10 à 50 mW, suivant la tension d'entrée.

Si vous essayez différentes valeurs de R1, vérifiez que la tension aux bornes de C1 n'est pas inférieure à 3 V ni supérieure à 5 V.

R1 en fonction de la tension d'entrée.

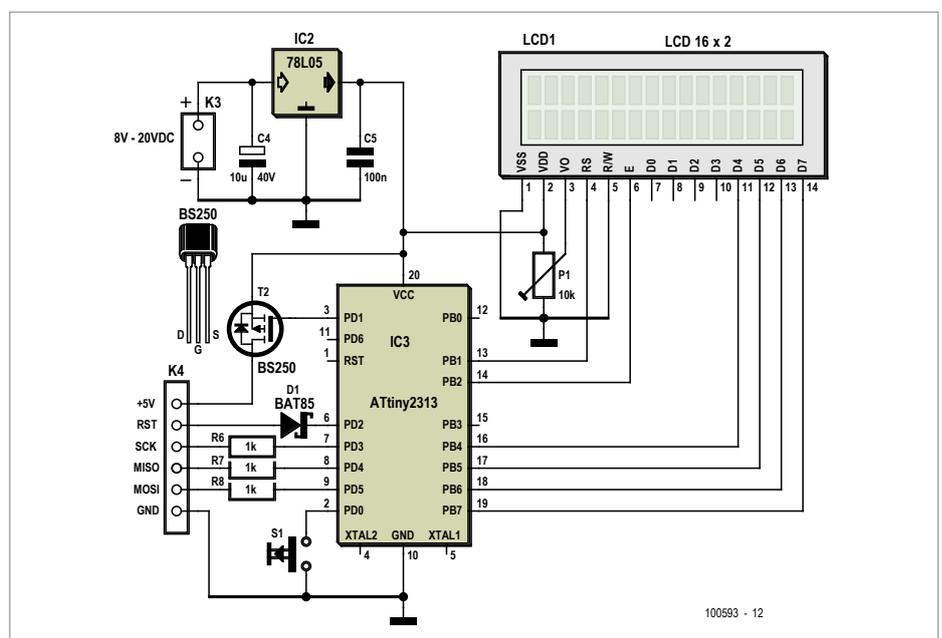
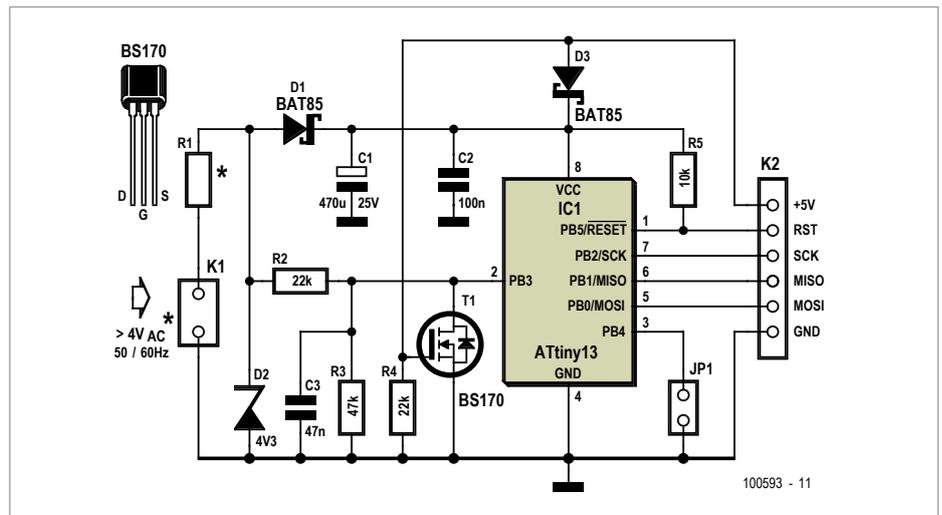
Tension d'entrée	R1
4 à 5 V CA	1 kΩ
5 à 6 V CA	1,5 kΩ
6 à 8 V CA	2,2 kΩ
8 à 10 V CA	3,3 kΩ
10 à 12 V CA	4,7 kΩ
12 à 15 V CA	6,8 kΩ
15 à 18 V CA	8,2 kΩ
18 à 22 V CA	10 kΩ

Simultanément, le compteur *Timer0* compte les impulsions d'horloge. Une fois démarré, il compte jusqu'au maximum et provoque une interruption lors du débordement. C'est ce qui se produirait après quelque 50 ms en l'absence d'impulsions d'entrée, mais les impulsions d'entrée ne sont pas seulement comptées : elles remettent périodiquement le compteur à zéro, ce qui l'empêche d'atteindre la valeur maximale. De ce fait, tant que les impulsions d'entrée seront présentes, le compteur *Timer0* ne débordera pas et l'interruption associée ne s'exécutera pas.

Quand la tension d'entrée disparaît, la tension aux bornes de C1 commence à décroître. La capacité de C1 est suffisante pour maintenir le microcontrôleur en marche pendant au moins une seconde. Aussitôt que la tension alternative est coupée, les impulsions sur PB3 disparaissent. Comme il n'est plus remis à zéro, le compteur *Timer0* provoque une interruption quelque 50 ms après la dernière transition sur PB3. L'interruption tire le microcontrôleur de son sommeil (*Idle*) et déclenche l'exécution de la routine d'interruption. C'est elle qui copie dans l'EEPROM le contenu du compteur de secondes. Ainsi, l'état du compteur est sauvegardé quand la tension d'alimentation tombe en dessous du seuil de sécurité. La procédure d'écriture dans l'EEPROM prend environ 10 ms, le microcontrôleur retourne au mode sommeil pour attendre l'un des événements suivants :

- si l'alimentation est rétablie dans un délai suffisamment bref — c'est-à-dire avant que la tension de C1 tombe notablement —, si elle est encore dans la plage de fonctionnement sûr, le microcontrôleur se réveillera et recommencera à compter les impulsions d'entrée dès qu'elles se présenteront.
- si la tension aux bornes de C1 tombe en dessous de 2,7 V, ce qui se produit quelques secondes après la coupure de l'alimentation, le circuit de détection de baisse de tension (*Brown Out Detection*) de l'ATtiny13 remettra le microcontrôleur à zéro.

Le circuit *BOD* devra être activé lors de la programmation, avec le seuil à 2,7 V (*BODLEVEL fuse 1 = 0*, *BODLEVEL fuse 0 =*



1). Cela garantit que le microcontrôleur s'arrête de fonctionner avant que la tension d'alimentation tombe sous le seuil de sécurité, où la RAM risquerait d'être altérée aussi. Quand il est activé, le circuit *BOD* surveille en permanence la tension *VCC* et la compare au seuil défini. Si la tension d'alimentation tombe sous 2,7 V, le microcontrôleur est remis à zéro et maintenu dans cet état tant que la tension n'est pas revenue au niveau sûr. Au démarrage après remise à zéro, le microcontrôleur lit la valeur accumulée et continue le comptage à partir de là.

Retour au circuit d'entrée : la valeur de R1 sera choisie en fonction de la tension d'entrée puisque son rôle est de limiter l'intensité du courant d'entrée — voir l'encadré à propos de R1.

### Circuit de lecture

Bien sûr, vous aurez envie de lire le compteur de temps en temps. Pour garder le circuit aussi simple que possible, aucun afficheur n'a été prévu au départ, les lectures se faisaient au moyen d'un programmeur ISP (*In System Programming*) à basse tension. Finalement, un outil spécial a été conçu et son schéma est ici. Le cœur du circuit est le  $\mu$ C ATtiny2313. Il produit tous les signaux de commande nécessaires pour lire le contenu de l'EEPROM du Mouchard, le convertir en une forme lisible et l'afficher sur un écran LCD alphanumérique à 2 lignes de 16 caractères. Si vous utilisez le circuit de lecture proposé, vous pouvez le connecter au Mouchard, que l'appareil dans lequel il est installé soit en marche ou non. Toutefois, soyez prudent si le Mouchard est connecté directement à la tension du secteur ! Si le mouchard est actif (maté-

## Stop encore ! Sécurité électrique !



Si le circuit est raccordé directement au secteur, remplacez R1 par l'assemblage en série de :

- une résistance de 2,2 k $\Omega$ /1 W et un condensateur de 100 nF pour un secteur 115 V/60 Hz, ou
- une résistance de 4,7 k $\Omega$ /1 W et un condensateur de 100 nF pour un secteur 230 V/50 Hz.

Dans les deux cas, le condensateur doit être prévu pour la connexion directe au secteur (p.ex. WIMA MKP-X2, WIMA MP3-X2, Epcos MKP X2 ou similaire).



riel en marche), le lecteur doit être mis en marche avant d'être connecté. Cela garantit un niveau défini sur les lignes de communication, pour ne pas perturber le fonctionnement du Mouchard. Si le lecteur n'est pas mis sous tension avant la connexion au Mouchard en marche, il risque de le remettre à zéro et de perdre le temps accumulé depuis la dernière mise sous tension. Si le Mouchard est inactif (appareil arrêté), l'état du lecteur au moment de la connexion est indifférent.

Pour lire et afficher le temps mesuré, appliquez simplement une pression brève sur le poussoir S1. Le lecteur appliquera d'abord la tension de +5 V, avec les conséquences suivantes sur le mouchard :

- si le Mouchard était inactif, le signal +5 V du lecteur alimentera le microcontrôleur ATtiny13 et rendra conducteur le transistor T1 du Mouchard, ce qui empêche l'exécution du programme de l'ATtiny13 ;
- si le Mouchard était déjà alimenté et actif, le signal +5 V du lecteur se contente de rendre T1 conducteur. Le court-circuit de l'entrée PB3 à la masse bloque les impulsions, ce qui force l'ATtiny13 à enregistrer l'état du compteur dans l'EEPROM.

En d'autres termes, quel que soit l'état préalable du Mouchard, le signal +5 V garantit que l'ATtiny13 est sous tension et que le contenu de son EEPROM est à jour. Peu après l'application du +5 V, le lecteur commence à communiquer avec le Mouchard et lit le contenu de son EEPROM. Dès la fin de la lecture, le lecteur coupe la ligne +5 V, traite les données et affiche le temps au format « jjjj:hh:mm:ss » (jours, heures, minutes,

secondes). N'oubliez tout de même pas que la précision générale dépend de la stabilité en fréquence du secteur et que les longues durées ne seront pas exactes à la seconde !

En plus de la lecture, le lecteur peut aussi effectuer la remise à zéro du compteur en cas de besoin. La procédure d'effacement

est possible aussi avec le programmeur ISP, il suffit d'effacer la mémoire EEPROM. Remarque : l'effacement de la mémoire la remplit de 1 binaires. Cela met le compteur à -1 au lieu de 0, mais une inexactitude de cet ordre est négligeable.

### Programmes

Deux programmes Bascom-AVR ont été écrits pour ce projet.

EE\_T\_on . bas est le programme de l'ATtiny13 du Mouchard. Commencez par configurer les fusibles CKSEL pour : oscillateur RC interne à 9,6 MHz, BODLEVEL comme indiqué. Effacez l'EEPROM lors de la programmation pour mettre le compteur à zéro avant la première utilisation.

EE\_T\_on\_reading\_tool . bas est le programme de l'ATtiny2313 du circuit lecteur. Configurez les fusibles CKSEL pour oscillateur RC interne calibré à 8 MHz.

Connectez-vous maintenant à l'internet ! Les deux programmes sont dans le fichier 100593-11 . zip à télécharger

[1]. C'est aussi l'endroit où commander des microcontrôleurs programmés pour ce projet. Pour les adeptes de Burn-aChip@Home : une version de démonstration gratuite de Bascom-AVR est disponible [2], et le catalogue Elektor propose aussi des livres (en anglais) sur ce merveilleux petit compilateur [3].

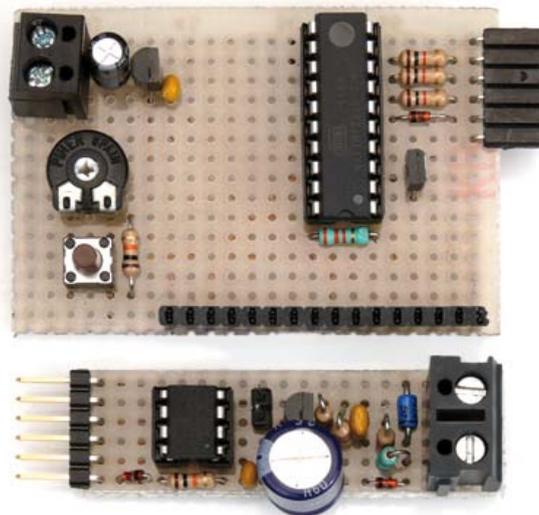
(100593 - version française : Jean-Paul Brodier)

### Liens

[1] [www.elektor.com/100593](http://www.elektor.com/100593)

[2] [www.mcselec.com](http://www.mcselec.com)

[3] [www.elektor.com/products/books.255. lynkx](http://www.elektor.com/products/books.255 lynkx)



ressemble à celle de la lecture, à ceci près que vous devrez maintenir la touche enfoncée pendant trois secondes pour effacer le compteur du  $\mu$ C ATtiny13.

Si vous lisez l'EEPROM du Mouchard avec un programmeur ISP ordinaire, n'oubliez pas de mettre hors tension l'appareil où il est installé, avant de raccorder le programmeur. Le compteur est enregistré à partir de l'adresse 0, sur 4 octets (type *Long*), *little endian*, ou *LSB first*, ou encore poids faible d'abord. Ce nombre représente le temps de marche mesuré en secondes. Il reste à effectuer quelques calculs pour le déchiffrer (le lecteur le fait pour vous). L'effacement du comp-

## Attention !



Le circuit entier est soumis à la tension du secteur, il est dangereux de le toucher. Ne travaillez jamais sur un appareil raccordé à la prise secteur. Le circuit doit être enfermé dans un coffret agréé qui empêche tout contact avec un composant. En cas de doute, demandez conseil à un électricien qualifié.



[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)

## Tubes audio anciens & récents

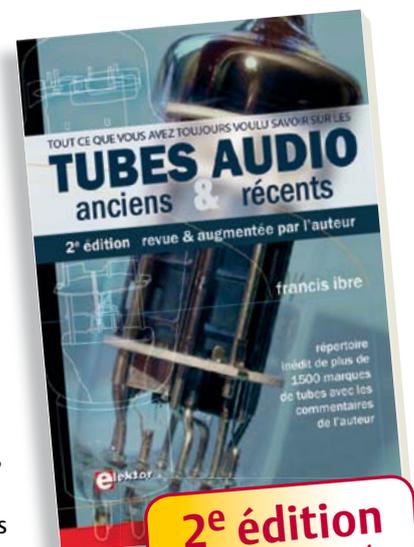
➔ Répertoire inédit de plus de 1500 marques de tubes, commenté par l'auteur

Il y a plus de cent ans, Lee de Forest eut la merveilleuse idée d'ajouter une troisième électrode, une grille, entre le filament et la plaque d'une lampe détectrice : l'Audion était né, et avec lui, débutait l'ère de l'électronique. Que de progrès depuis ! Et pourtant cette technologie vieille d'un siècle n'est pas encore obsolète : aujourd'hui, à l'heure des nanotechnologies et de la très haute intégration des circuits électroniques, Western-Electric fabrique toujours la triode 300B.

Quelle est donc cette magie des tubes de verre qui nous enchante encore ? Quels secrets les rendent irremplaçables à nos oreilles de mélomanes ? Rien que le savoir-faire d'hommes passionnés, transmis et enrichi sur plus de quatre générations.

Découvrez ici ce qui se cache au cœur des "lampes", à travers une centaine de photos et d'illustrations, un tour d'horizon des productions actuelles et un historique des fabricants anciens les plus connus.

Apprenez à reconnaître et à choisir les tubes NOS les plus réputés, grâce aux codes et à leurs particularités de construction.



**2<sup>e</sup> édition**  
revue & augmentée

175 pages  
Format 17 x 24 cm  
ISBN 978-2-86661-185-9  
43,50 €

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

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

# transformateurs de courant

Les transformateurs de courant professionnels utilisent souvent des noyaux toroïdaux et sont bobinés avec une grande précision. Cet article vous apprendra à faire les vôtres avec des composants faciles à trouver. À vous les mesures de courant en toute sécurité et avec isolation.

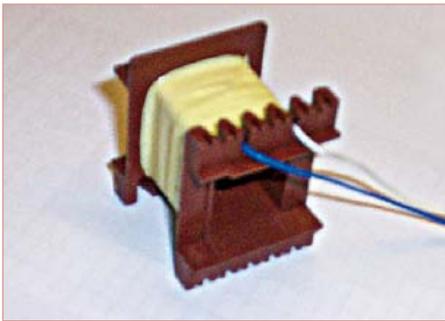


Figure 1. Un transformateur de courant, l'enroulement secondaire est en place.

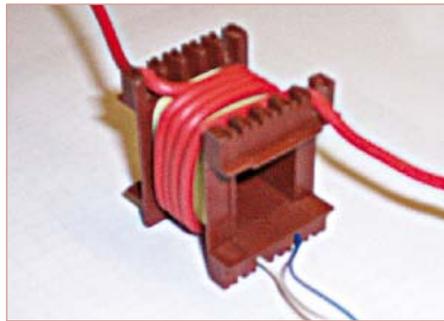


Figure 2. Avec l'enroulement primaire en plus.

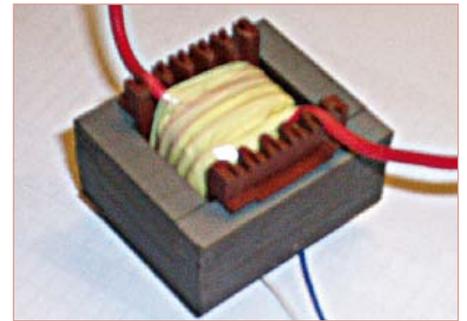


Figure 3. On ajoute les tôles et c'est fini.

## Ed Dinning (Royaume-Uni)

Les transformateurs de courant (TC), comme leur nom l'indique, transforment un fort courant en un plus petit (souvent) pour la mesure et le contrôle. Ils apportent également une isolation entre enroulements primaires et secondaires ; c'est pour cela qu'ils sont très utilisés dans les réseaux électriques. Par rapport aux capteurs à effet Hall et aux bobines de Rogowski, ils présentent l'avantage considérable d'être auto-alimen-

tés et de pouvoir alimenter des compteurs et relais de protection.

### Un peu de calcul

Lorsque l'on conçoit un transformateur, on commence habituellement par déterminer la charge. Celle-ci est en général inférieure à 10 VA pour la plupart des TC professionnels (les charges sont souvent exprimées en VA plutôt qu'en watts à cause des déphasages ; cela n'aura pas vraiment d'impact ici).

Nous allons attaquer un instrument de mesure ou un comparateur faisant partie

d'un circuit de protection, une charge de 0,5 VA sera suffisante. Nous supposons que le courant à surveiller sera compris entre 50 mA et 10 A, soit un rapport de 200 à 1. Avec une charge de 0,5 VA et un courant de 50 mA, un rapide calcul nous donne une tension de 10 V ( $U = P / I$ ) et une charge résistive de 200  $\Omega$  ( $R = U / I$ ).

Calculons maintenant le nombre de spires nécessaires au secondaire du transformateur. Pour cette application, nous allons utiliser un petit transformateur possédant un noyau d'une surface de, disons 300 à

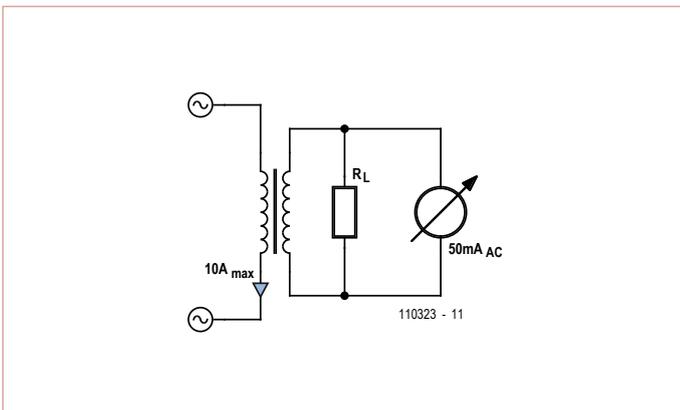


Figure 4. Circuit de test pour notre transformateur bobiné à la main.

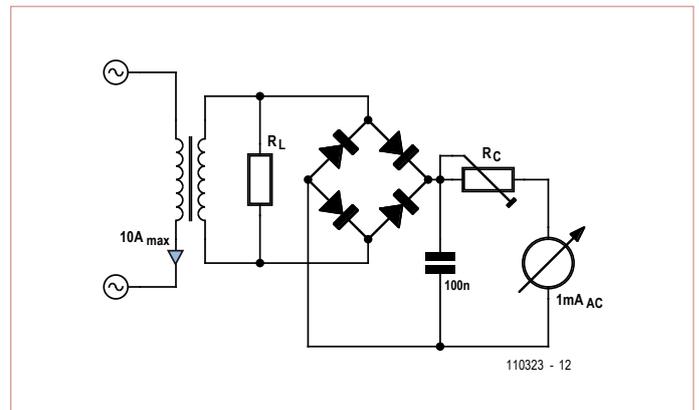


Figure 5. Un TC dans un circuit avec un galvanomètre à cadre mobile.

400 mm<sup>2</sup>. On peut les acquérir pour pas très cher, voire les récupérer sur de vieux appareils. Plus grande sera la surface du noyau et moins de spires il faudra.

La plupart des noyaux en tôle des transformateurs commerciaux sont classés selon un numéro EI. Nous utiliserons un EI48 (la grande dimension des tôles vaut 48 mm) d'une épaisseur de 20 mm. La branche centrale du noyau fait 16 mm de large, l'épaisseur étant de 20 mm, soit une surface de :

$$0,016 \times 0,02 = 0,00032 \text{ m}^2$$

Utilisons ensuite l'équation du transformateur :

$$N = E / (4,44 f B A_e)$$

où  $N$  est le nombre de spires,  $E$  la tension appliquée,  $f$  la fréquence de fonctionnement,  $B$  la densité du flux en Tesla et  $A_e$  la surface en coupe du noyau en mètres carrés.

La fréquence du secteur est de 50 Hz. Il est courant d'utiliser une densité de flux relativement faible de 0,2 T (contrairement à un transformateur de tension/puissance qui utilise une densité >1 T, suivant les matériaux du noyau). En remplaçant les paramètres par leurs valeurs dans l'équation, on obtient :

$$N = 10 / (4,44 \times 50 \times 0,2 \times 0,00032)$$

$$N = 704 \text{ spires}$$

Nous arrondirons à 700 spires. Nous savons qu'il nous faut un rapport de 200 pour 1 sur le nombre de spires, il nous faudra donc 3,5 spires au primaire. Ce n'est pas une valeur très pratique, nous utiliserons donc 800 spires au secondaire (ce qui nous donne une plus faible densité de flux et une meilleure précision) et donc 4 spires au primaire. Assez de théorie, enroulons du fil à présent !

### Bobinage du noyau

Le secondaire devra conduire 50 mA. On fait couramment circuler 3 A/mm<sup>2</sup> dans du fil de cuivre. En faisant le calcul à rebours, cela nous donne un diamètre de fil de 0,15 mm (environ), soit 0,16 mm en comptant le revêtement en email. La surface de bobinage sur le transformateur fait 20 mm de large et 6 mm de profondeur. Ce qui nous permet de bobiner 125 spires par couche et 6,4 couches (arrondi à 7) soit une épaisseur d'enroulement de 1,2 mm. Il restera donc 4,8 mm pour le primaire (**fig. 1**).

Vous remarquerez que le primaire est fait avec un fil de câble secteur, souple ou rigide, spécifié pour 10 A (**fig. 2**). L'isolation du primaire est assurée par l'isolant de ce fil. Quatre spires sont ajoutées par-dessus le secondaire, les enroulements sont maintenus en place par du ruban adhésif isolant (du ruban pour transformateur en polyester serait l'idéal). Le transformateur sera ensuite équipé de tôles en E et I, alternativement ; il n'y a quasiment pas d'entrefer (**fig. 3**).

Nous pouvons maintenant le tester avec le schéma de la **figure 4**. Vous remarquerez que la résistance de charge ( $R_L$ ) **doit** toujours être reliée, faute de quoi les enroulements peuvent entrer en court-circuit à cause de la très haute tension produite par un transformateur de courant fonctionnant en circuit ouvert. Un voltmètre numérique en mode tension alternative peut être utilisé pour mesurer la tension au secondaire. C'est le rapport entre son impédance interne et celle de la charge qui déterminera la déviation : plus l'impédance interne sera faible et mieux ça sera.

### Un deuxième exemple

Une utilisation plus pratique d'un TC est l'attaque d'un galvanomètre à cadre mobile à travers un redresseur, comme sur le schéma de la **figure 5**. La déviation maximale d'un

tel galvanomètre correspond souvent à 1 mA ; utilisé en parallèle avec la charge (traversée par 50 mA), il n'aura pas beaucoup d'impact sur la précision du système.

Comme le montre le schéma, le transformateur produit une tension maximale de 10 V aux bornes de la résistance de charge avec un courant de 10 A au primaire. En mettant cette tension à l'entrée d'un pont redresseur, on récupère une tension continue en sortie du pont.

Avec le galvanomètre de 1 mA relié à cette sortie continue, on peut le calibrer pour lire directement les ampères du primaire en utilisant une résistance série ( $R_C$ ). La loi d'Ohm nous dit que la résistance doit avoir une valeur de 10 kΩ moins la résistance interne du galvanomètre (typiquement 100 Ω ou moins). En utilisant une résistance ajustable de 10 kΩ, le galvanomètre peut être calibré précisément.

Les vibrations de l'aiguille du galvanomètre peuvent être atténuées par un condensateur en parallèle sur la sortie du pont. 0,1 μF devraient suffire. Une valeur plus forte augmentera l'inertie du galvanomètre qui affichera alors un maximum local.

Après une calibration précise, vous vous apercevrez que dans le bas de l'échelle cet appareil n'est pas linéaire. C'est la faute, en partie, aux pertes dues à la tension de seuil des diodes du pont redresseur. L'utilisation de diodes Schottky ou au germanium améliorera les choses.

Ce montage pourra aussi être étendu pour la mesure de courants hachés et sur les circuits audio ; l'utilisation d'un transformateur à noyau de ferrite est recommandée pour réduire les pertes sur les fréquences supérieures à 1 kHz. Avec de la ferrite, il faudra réduire la densité du flux à environ 25 mT pour maintenir la précision.

(120193 – version française : Kévin PETIT)



PCBs  
Muuuuch 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

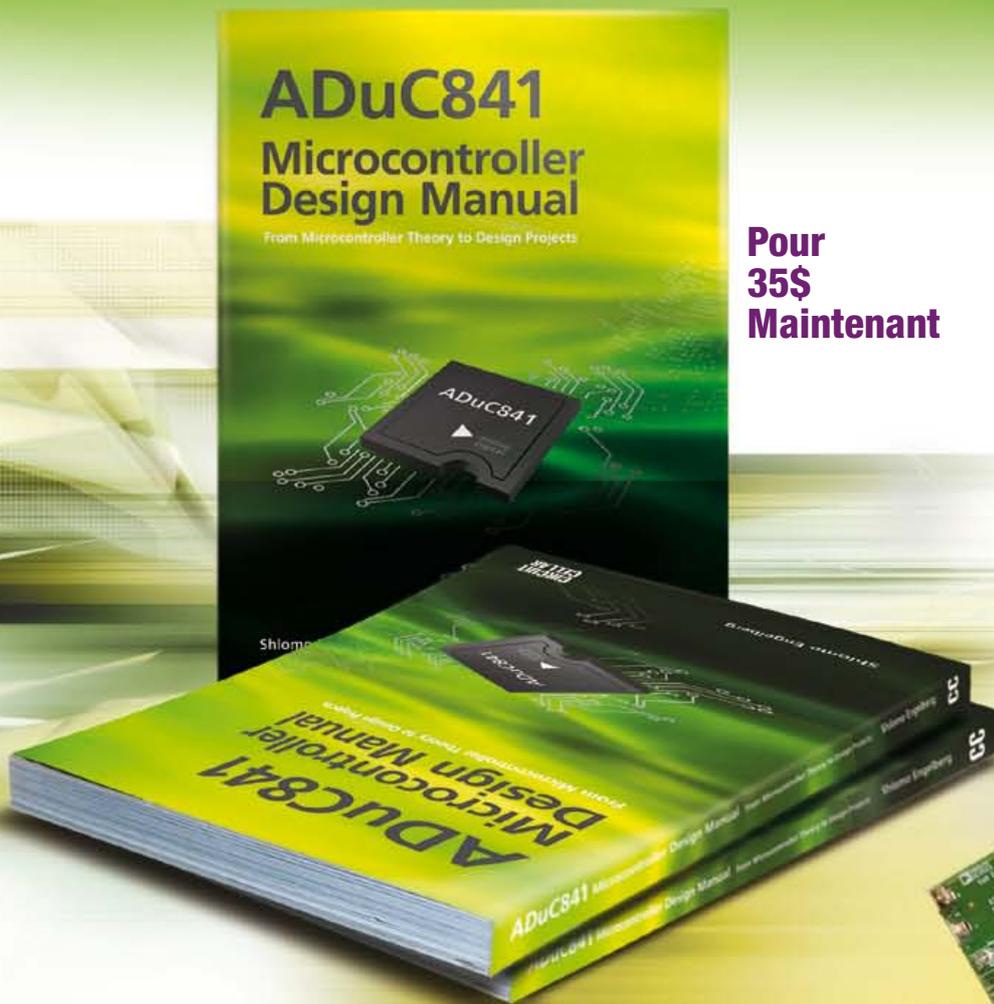
www.jackaltac.com



# 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)**

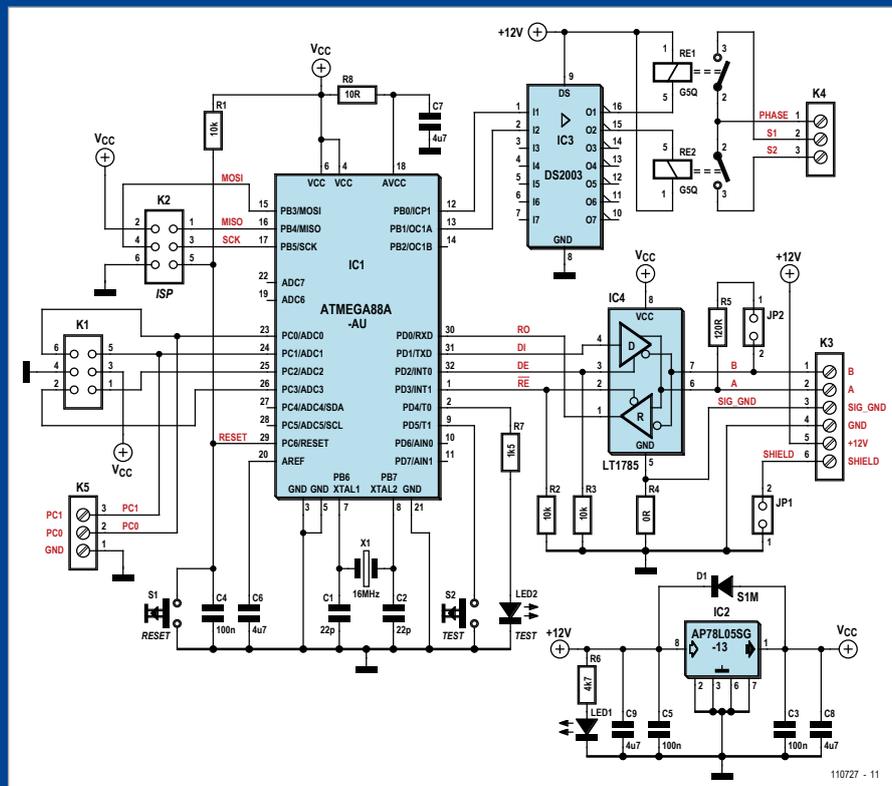
# les masses du bus RS-485

Jens Nickel (Elektor)

Fans de l'automatisation en général et d'ElektorBus en particulier, les nouvelles sont bonnes : bientôt le matériel de notre bus va s'enrichir, avec notamment une plus grande carte XMEGA flanquée d'un afficheur et d'un module serveur web utilisable comme central de commande. La bibliothèque *C-firmware* est en cours d'élaboration. Pour satisfaire votre curiosité, visitez [www.elektor-projects.com](http://www.elektor-projects.com) ! En attendant, voici quelques potins de machine à café. Lors de la conception de notre carte de *Commande de charge sous 230 V par RS-485* [1], nous avons prévu un accès supplémentaire à la connexion de terre du circuit de commande RS485 (SIG\_GND = signal ground). L'idée était de faire passer cette masse supplémentaire de nœud en nœud du bus RS485 parallèlement à la ligne de masse normale (GND). L'article cité indiquait ce qui suit :

*Lors de la commutation de charges, des perturbations locales polluent la masse du bus. Dans le pire des cas, des bits fantômes peuvent alors apparaître si les potentiels de masse des pilotes diffèrent trop fortement les uns des autres. Solution possible : relier les masses du signal et du bus en un seul point du bus. Pour les autres participants, les deux masses pourraient être séparées par une résistance de p. ex. 100 Ω.*

Un lecteur, Andreas Schoenberg, féru du bus RS-485, estime légitime notre souhait d'une masse de référence dont le potentiel serait rigoureusement le même pour tous les circuits de commande du bus. À ceci près, dit-il, que sur le LT1785 la tension en mode commun sur A/B doit se situer dans la plage entre -7 et +12 V (par rapport à la masse). Même si la différence de potentiel entre les masses de deux nœuds du bus atteignait 7 V, le bus fonctionnerait correctement ! Notre lecteur poursuit : « Dans une configuration statique, une telle différence est impossible. Si la différence de potentiel entre la ligne GND et un nœud débiteur de courant de forte intensité atteignait 7 V, la différence serait la même sur la ligne d'alimentation correspondante, soit 14 V en tout ! » Une valeur impossible puisque la tension d'alimentation du bus n'est que de 12 V. Et notre spécialiste de conclure : « Le comportement dynamique serait plutôt le suivant : quand p. ex. un relais du module est excité, cela peut se traduire par une chute de tension. Qu'il conviendrait de compenser localement avec des condensateurs sur les broches d'IC3. Je suis d'avis qu'avec un tampon local bien dimensionné, il n'y a pas à craindre sur l'ElektorBus de problème de différences de potentiel de masse, du moins pas de nature à perturber la communication. »



Si vous vous retrouvez dans une configuration particulière avec un nœud qui consommerait tant de courant (ou dont les connexions seraient mal dimensionnées pour une telle intensité), au point que GND se retrouverait à un potentiel plus élevé de 3 V que celui de SIG\_GND, d'autres problèmes pourraient survenir. Le LT1785 verrait sur sa broche V<sub>CC</sub> une tension supérieure de 5 V au potentiel GND local, de sorte que sa tension d'alimentation serait de 8 V.

La broche DE est forcée par R3 au potentiel de GND, que même la CPU ne peut abaisser davantage, faute de voir un autre potentiel de référence que celui-là. Pour l'émetteur-récepteur ce sont 3 V au-dessus de son propre potentiel de masse, SIG\_GND, soit un *niveau logique haut permanent*. Il serait impossible dès lors d'exercer une quelconque influence au moyen du logiciel sur l'état de la broche *Enable* du circuit !

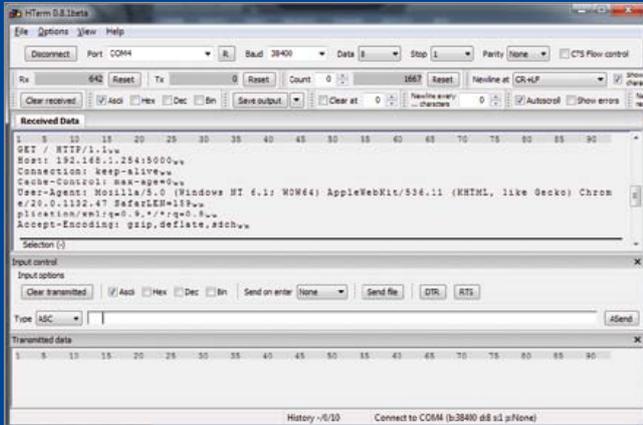
Assez de théorie. Nous aimerions savoir comment ça se passe en pratique, chez vous. Nous n'avons rien constaté de tel [2] mais si vous avez monté l'ElektorBus, vous avez peut-être eu des ratés liés à ce problème. Si oui, quels câbles avez-vous utilisés ? Adressez votre témoignage à [redaction@elektor.fr](mailto:redaction@elektor.fr)

(120501 – version française : Gerry Atrix)

[1] [www.elektor.fr/110727](http://www.elektor.fr/110727)

[2] [www.elektor.fr/120198](http://www.elektor.fr/120198)

# Parkinson touche aussi les AVR

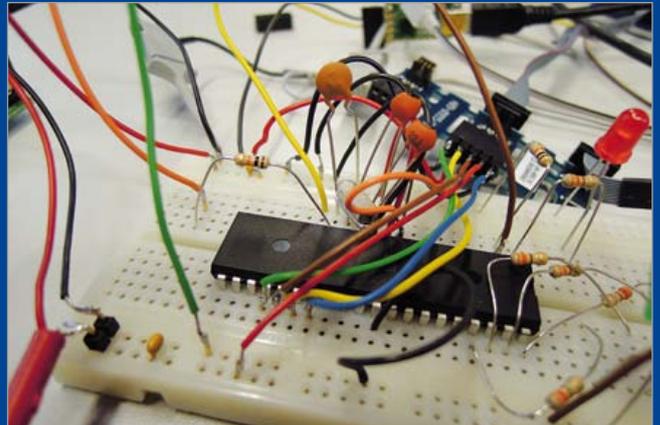


Koen Beckers (Elektor)

Connaissez-vous le syndrome de l'AVR qui, sans raison apparente, se remet à zéro brutalement lors d'une communication avec votre PC ? Cela m'est arrivé alors que je testais un ruban à LED Wifi de ma conception.

Après bien des changements de configuration et d'essais, j'ai fini par trouver que cela se produisait lorsque je tentais de faire recevoir au microcontrôleur un gros paquet de données par l'UART. Parfois tout va bien, mais le plus souvent l'AVR redémarre juste après le début de la transmission et, des fois, il se plante. Plus rarement, on dirait qu'il a la *tremblante du bouton* et redémarre sans cesse. Le seul moyen d'interrompre la boucle infernale est alors de couper l'alimentation.

Après consultation du Dr. Internet, j'ai appris qu'il s'agissait d'un problème courant avec les AVR. Par chance, la solution est simple : insérer dans le lien UART une résistance de 10 kΩ en



série et un condensateur de découplage de 100 pF, ce qui crée tout simplement un filtre passe-bas à environ 159 kHz. Mes problèmes affectaient la réception de données, j'ai donc inséré une résistance dans la ligne RX et ajouté un condensateur entre RX et la masse. Quelques lignes de code qui renvoyaient les données reçues sur RX à mon PC (voir la capture d'écran) ont confirmé que le circuit fonctionnait correctement. Désormais, toutes les données sont reçues, le  $\mu\text{C}$  ne plante plus.

Pour m'assurer que si le problème était maintenant résolu ce n'était pas par hasard parce que je venais de réécrire mon code de test, j'ai retiré le filtre RC. En relançant mon code de test sans rien y changer, l'AVR s'est immédiatement planté. Je mets le filtre RC, et tout fonctionne à nouveau parfaitement. Je suis plutôt confiant, assez en tout cas pour affirmer que ce simple filtre RC a résolu mon problème. Résoudra-t-il aussi les vôtres ?

(120386 – version française : Kévin PETIT)

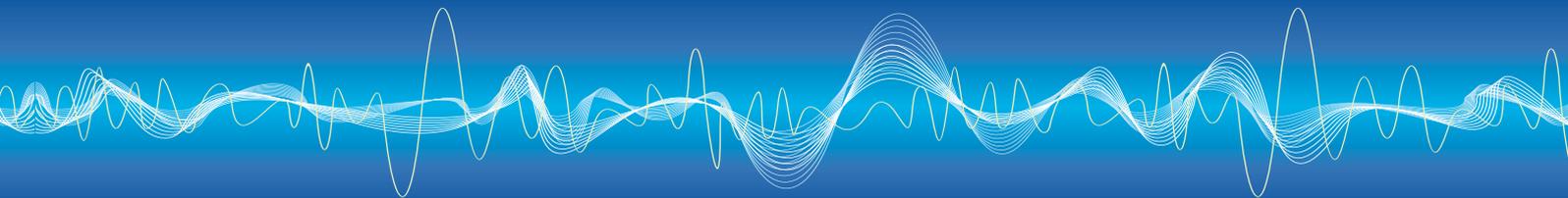
## e-pollution !

Thijs Beckers (Elektor)

En matière de mesure et de test, plus ça change, plus c'est la même chose, et les surprises sont assez rares au labo... et pourtant, il y en a, et même d'assez belles pour qu'on en parle ici. Récemment notre collègue Ton Giesberts testait un récepteur de signal horaire dans le cadre de notre série d'articles sur la radio logicielle avec AVR (commencée en mars 2012 [1]). Ton s'aperçoit qu'il ne reçoit pas le moindre signal radio ni sur sa paillasse au labo ni nulle part ailleurs dans le château d'Elektor. L'épaisseur des murs (environ 80 cm) est certes de nature à atténuer le signal d'entrer. Pourtant, même déportée au bout d'une perche en bois de 2 m tendue par la fenêtre, l'antenne ne captait toujours pas le moindre signal horaire.

Suspectant les routeurs WiFi, PC, portables, néons, blocs secteurs à découpage et téléphones DECT d'interférer avec le





signal radio, Ton s'est mis à arpenter le château pour déterminer l'endroit le moins bruyé à l'aide d'un récepteur *GO portable* (oui, c'est comme ça qu'on disait avant que les téléphones deviennent *portables*). Nulle trace dans tout le château de la porteuse du signal horaire britannique sur les grandes ondes (198 kHz). Tout n'était que bruit et *e-smog* (hé, n'avons-nous pas publié récemment un détecteur pour ça ? Mais si, c'est le TAPIR [2] !). Il a finalement fallu déporter la radio de plus de 5 m en dehors de la massive bâtisse pour que le signal puisse enfin être discerné.

De retour dans les murs — radio allumée — Ton a observé un pic d'interférences à proximité des néons, dont il est avéré qu'ils produisent pas mal de bruit, pile autour de 198 kHz. Ton, qui ne manque ni d'expérience ni de jugeote, est aussi très matinal ; il a réessayé (très) tôt un lundi matin, quand il est encore à peu près seul et que dans la plupart des bureaux les plafonniers sont encore éteints. En vain, car il reste bien trop de bruit causé par les PC, les routeurs WiFi, les téléphones DECT, les imprimantes et tous les autres appareils qui ne dorment jamais.

C'est officiel maintenant, nous ne pouvons pas tester d'équipement sensible dans le château. Prenons les choses du bon côté et considérons que tout circuit qui fonctionne dans notre labo

fonctionnera forcément partout... Pas mal comme épreuve du feu, non ?

Notre collègue Luc Lemmens a rencontré le même genre de problèmes en testant un montage à récepteur de signaux horaires (à paraître) qui refusait de fonctionner au labo. Il l'a donc emporté chez lui pour vérifier, et là, dès la mise sous tension de son prototype, tout à fonctionné comme prévu, et du premier coup !

Fort de cette expérience, Ton a lui aussi envoyé son prototype à un collaborateur externe pour qu'il le teste. Là aussi, ça a marché immédiatement. La morale de cette histoire : si vous testez un prototype qui reçoit des signaux radio, ne soyez pas trop surpris s'il ne fonctionne pas dans votre labo. C'est peut-être tout simplement un problème de pollution électromagnétique qui interfère avec le signal à recevoir.

(120504 – version française : Kévin PETIT)

## Liens

[1] [www.elektor.fr/100180](http://www.elektor.fr/100180)

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

# votre composant préféré

[myfavoritepart@elektor.com](mailto:myfavoritepart@elektor.com)

Thijs Beckers (Elektor)

Vous qui comme nous concevez des circuits électroniques, vous ne cessez de manipuler des composants, certains tout nouveaux, d'autres inchangés depuis des lustres. Comme nous, vous avez forcément un régulateur de tension préféré, une diode Schottky adorée, un microcontrôleur adulé, un type de condensateur ou même un tube pour lequel vous avez une prédilection marquée. Nous souhaiterions en savoir plus sur les préférences des lecteurs d'Elektor à qui nous demandons de nous expliquer pourquoi telle diode Gunn dont vous parlez tout le temps est votre préférée. Racontez-nous vos expériences

avec cette BRY39, décrivez votre rapport avec tel CI que vous préférez aux autres. Le LM729 a-t-il encore des secrets pour vous ? Pourquoi utilisez-vous encore un 6502 dans tous vos montages ?

La rédaction est prête à accueillir le tsunami de témoignages que vous allez adresser à [myfavoritepart@elektor.com](mailto:myfavoritepart@elektor.com). Surprenez nous, ne craignez pas les histoires percutantes au sujet de votre composant préféré.

(120503 – version française : Kévin PETIT)





**NOUVEAU**

Stefan Schwark

# Android

**apprendre à programmer des applis**

Environnement de développement **Eclipse**  
Programmation orientée objet en **JAVA**

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



**elektor**

**208 pages**

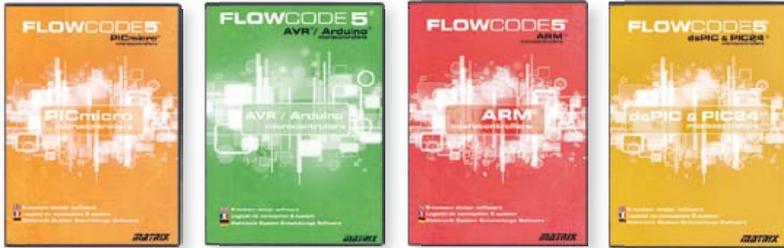
**format 17 x 23,5 cm**

**ISBN 978-2-86661-187-3**

**33,50 €**

# Flowcode 5 pour concev

## FLOWCODE5

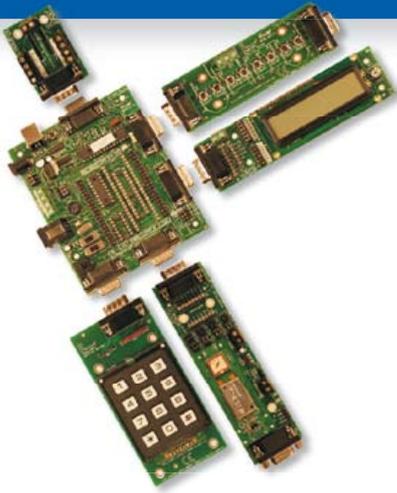


**Flowcode 5** est l'un des langages de programmation graphique pour microcontrôleurs (PIC, AVR, ARM et dsPIC/PIC24) les plus avancés au monde. Son avantage principal est de permettre la création de systèmes électroniques et robotiques complexes même si l'on manque encore d'expérience.

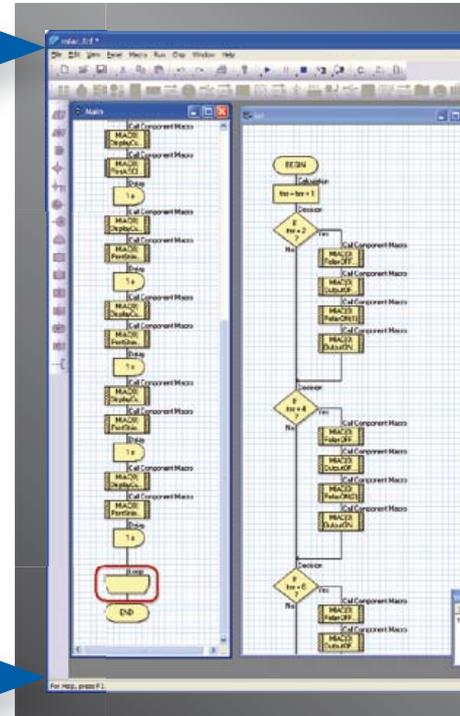
**Flowcode est utilisé :**

- 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'aux 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 (**Matrix Industrial Automotive Controller**) 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.

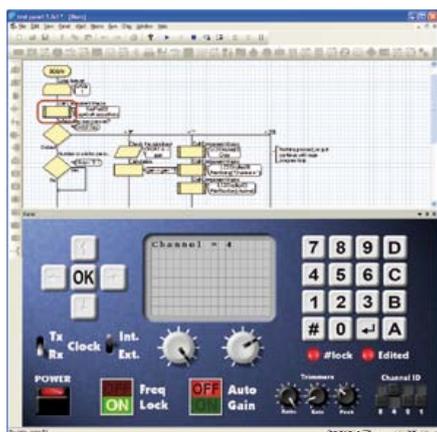
## Outil de débogage FlowKit

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.

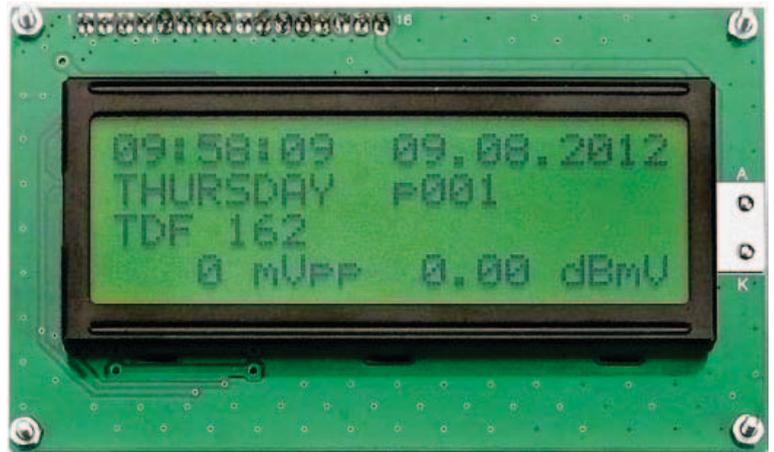


Retrouvez les E-blocks et leur documentation sur :  
**[www.elektor.fr/eblocks](http://www.elektor.fr/eblocks)**

# radio logicielle avec AVR (6)

## d'autres décodages sur la BBC

Les microcontrôleurs AVR conviennent parfaitement au traitement numérique du signal, c'est ce que cette série d'articles montre à suffisance. Cette fois-ci, pour terminer la série, examinons différentes manières de procéder au décodage. Comme d'habitude, avec l'accent mis sur la pratique. Attaquons-nous à l'extraction des données horaires des signaux de la BBC.



Martin Ossmann (Aix-la-Chapelle)

Dans le cinquième épisode [5], nous avons déjà décodé des signaux de référence du temps qui passe. Avec les filtres RII et CIC adapté, nous avons traduit les informations horaires de l'émetteur allemand DCF77, du britannique MSF et du français TDF162. Passons finalement au décodage de BBC198 et de l'ancien émetteur BBC648. Avec ces exemples, nous verrons d'autres techniques de décodage et comment les réaliser en pratique.

### La synchronisation de l'horloge de bits

La technique des communications dispose d'un ensemble de procédés pour la transmission synchrone des données. Elles circulent ainsi sans arrêt dans tout le système à un rythme déterminé. Dans le récepteur, on commence d'habitude par une démodulation. Le signal ainsi obtenu contient les bits recherchés.

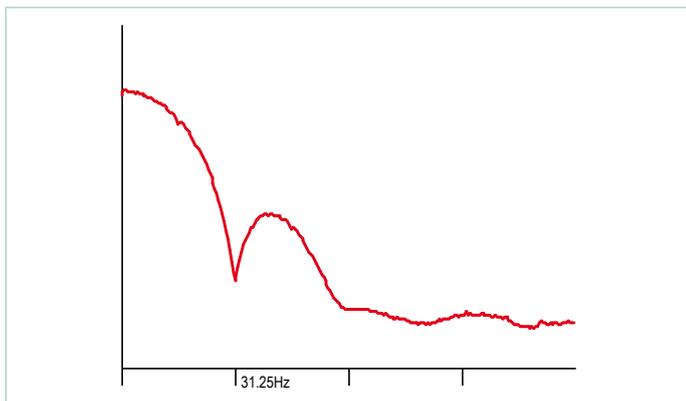


Figure 1. Spectre du signal de données à 31,25 Hz.

Pour faire nos expériences sur la synchronisation d'une horloge de bits, il nous faut le syntoniseur simple ou la platine de récepteur disponible chez Elektor, ainsi que le logiciel de réception EXP-RX-FM-125kHz-RDStlike-BitSync-V01.c, le générateur de signaux avec le circuit oscillant à 125 kHz plus le logiciel d'émission EXP-TX-FM-125kHz-RDStlike-V01.c qui permet de fabriquer des données modulées en fréquence sur 125 kHz au débit de 31,25 bit/s. Le signal d'entrée et l'horloge sont alors disponibles aux sorties du convertisseur N/A.

Pour récupérer l'horloge, on échantillonne le signal à une fréquence de  $31,25 * 16 = 500$  Hz. Ainsi, chaque bit compte 16 échantillons. On pourrait essayer de reconstituer le signal de données avec une boucle à phase asservie (PLL) au débit binaire de 31,25 Hz. Mais ça ne marche pas. La raison : on la trouve en examinant le spectre du signal à la **figure 1**. Il ne contient pas l'ombre d'une composante de signal à 31,25 Hz sur laquelle la PLL pourrait accrocher.

Cependant, quand on redresse le signal de données, le spectre,

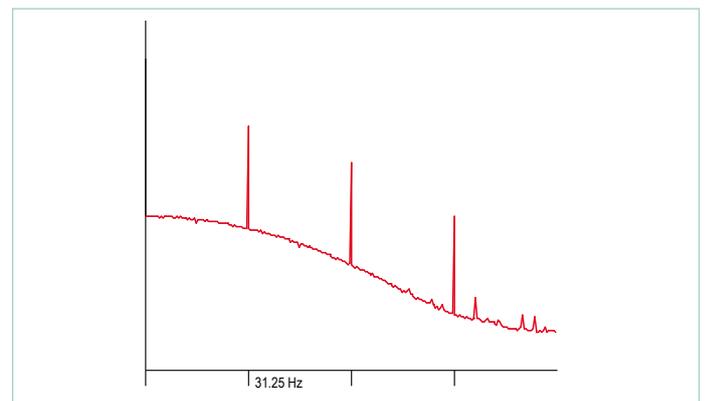


Figure 2. Spectre du signal de données après redressement.

comme on peut le voir à la **figure 2**, contient bien une raie à 31,25 Hz. Une PLL peut alors être asservie sur ce signal. La PLL de récupération des données est représentée schématiquement à la **figure 3**.

Cette PLL est similaire à celle utilisée précédemment pour récupérer la porteuse (cf. fig. 7 du quatrième épisode [4] de la série). Voici comment procéder pour régler l'horloge de bits en se référant au programme EXP-RX-FM-125kHz-RDSlike-BitSync-V01.c. La routine « do-Signal-Sample() » du **listage 1** est appelée à 16 fois la fréquence de l'horloge de bits. Elle incrémente la variable Clk16 et la remet à zéro quand le nombre 16 est atteint. La phase de ce signal en dent de scie est alors ajustée pour que la valeur médiane (Clk16 = 8) se trouve réellement au milieu des bits émis. La routine doBit-Sample détermine alors le bit qui sera ensuite traité. La commande de phase s'opère comme ceci : normalement, Clk16 compte périodiquement l'horloge de données jusqu'à 16. En fonction du signal de sortie du comparateur de phase, le compte peut s'arrêter à 15 ou se poursuivre jusqu'à 17 de manière à rattraper la phase.

Pour contrôler le fonctionnement correct de l'horloge de la PLL, on sort le signal démodulé sur DAC1. Sur DAC2, c'est le signal Clk16 qui est présent avec une pointe lorsque Clk16 = 8, le moment de l'échantillonnage. La **figure 4** le montre lors de la réception de la BBC sur 198 kHz (cf. **fig. 5**).

### Early-Late-Synchroniser

Pour récupérer le signal d'horloge, il n'y a pas que la PLL, nous connaissons aussi un *Early-Late-Gate-Synchroniser*. Qu'est-ce ? Coup de projecteur sur le diagramme de l'œil à la **figure 6** pour mieux comprendre. Avant l'heure, ce n'est pas l'heure ; après l'heure, ce n'est plus l'heure.

Ici aussi, il s'agit de faire coïncider le signal Clk16 avec l'horloge de bits pour que le signal de donnée tombe juste au milieu du bit. Pour y arriver, on mesure la valeur du signal en avance, au point E (*early*), et en retard au point L (*late*), par exemple lors des troisième (E) et douzième (L) coups de Clk16. Si les deux valeurs sont égales, les points E et L sont en symétrie avec le milieu. Si l'une des valeurs est plus grande que l'autre, l'œil est décentré et il faut, tôt ou tard, corriger la phase de Clk16.

Le code de programme correspondant est indiqué dans le **listage 2**. La différence entre la valeur au point E et celle au point L est additionnée dix fois dans *earlylateDelta*. Finalement, cette valeur sert à corriger la période de Clk16. On peut essayer le code pour ce synchronisateur avec le programme EXP-RX-FM-125kHz-RDSlike-BitSync-V01.c.

### Décodage de la modulation de phase sur BBC198

Nous en arrivons à la mise en pratique de la synchronisation de bit. Programmons le récepteur pour recevoir les données modulées en phase de la BBC sur 198 kHz.

La structure du signal est décrite précisément sur [7]. La phase de la porteuse à 198 kHz est modulée de  $\pm 22,5^\circ$ . Le débit binaire est de 25 bit/s et c'est le codage Manchester, dit aussi codage biphasé,

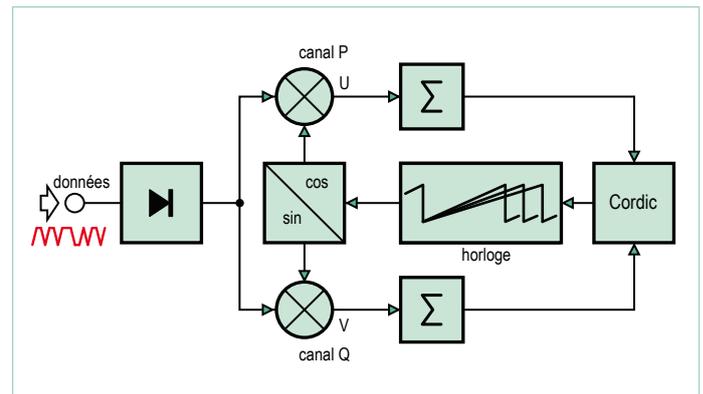


Figure 3. Diagramme fonctionnel de la boucle à phase asservie.

### Listage 1 : réglage de phase

```
void advanceCLK16() {
    Clk16++;
    if ( Clk16 >= ClockPeriod ) { Clk16=0 ; }
}

void doSignalSample(){
    int16_t Signal ;
    ATOMIC_BLOCK(ATOMIC_FORCEON) {
        Signal=FRQcicout ;
    }
    ClockPLL16(Signal) ;
    doBitSample(Signal) ;
    advanceCLK16() ;
}
```

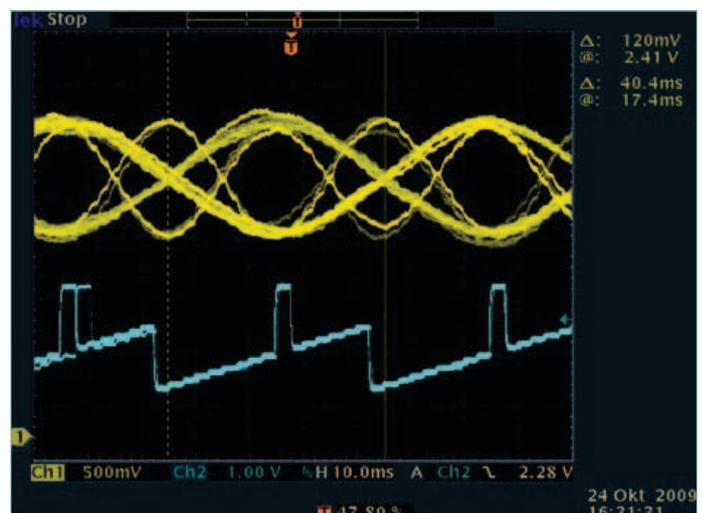


Figure 4. Le diagramme de l'œil de BBC198.

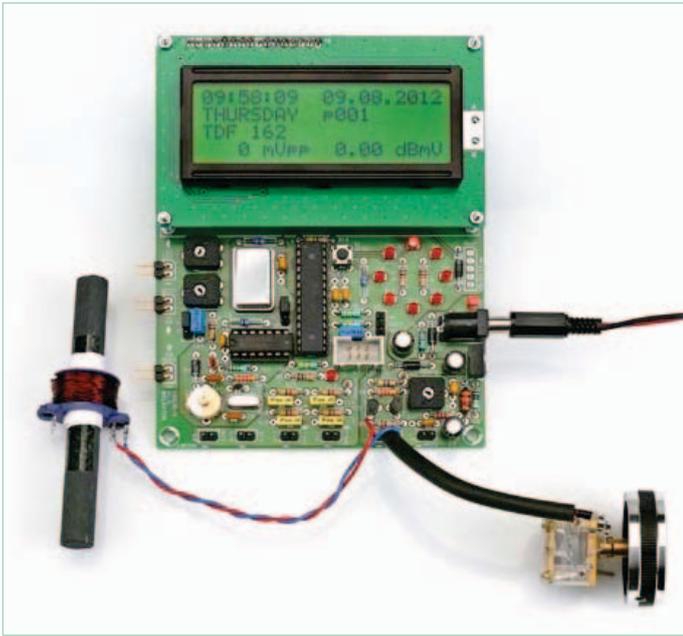


Figure 5. Le récepteur complet pour recevoir BBC198.

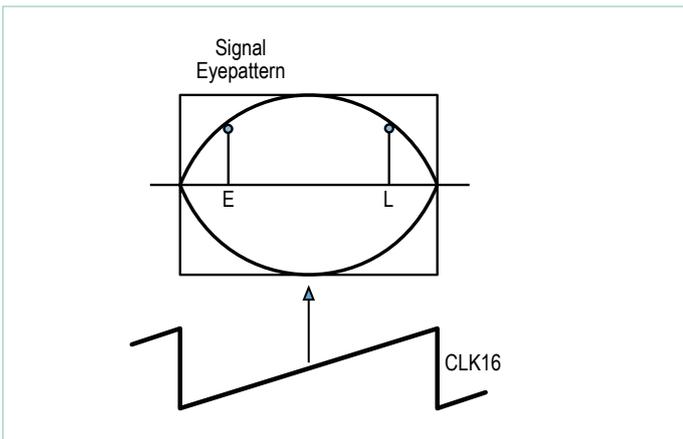


Figure 6. Le diagramme de l'œil du synchronisateur Early-Late.

**Listage 2 : synchronisateur Early-Late**

```

void earlyLateSynchronizer(int16_t Signal){
  if ( Clk16==3 ) { earlyValue=abs(Signal) ; }
  if ( Clk16==12 ) { lateValue=abs(Signal) ; }
  if ( Clk16==8 ) {
    earlyLateDelta += lateValue-earlyValue;
    ClkCnt++;
    ClockPeriod=16 ;
    if ( ClkCnt==10){
      ClockPeriod += earlyLateDelta/16 ;
      earlyLateDelta=0 ;
      ClkCnt=0 ;
    }
  }
}

```

avec filtrage, qui est en usage. De ce fait, la phase moyenne de la porteuse n'est pas modifiée et sa fréquence peut servir de référence très stable.

La conversion A/N s'effectue à une fréquence de  $20\,000 / 2\,500 = 8$  kHz. Conséquence de cet échantillonnage, la porteuse à 198 kHz est mélangée vers le bas sur  $198 - 25 * 8 = -2$  kHz. La fréquence intermédiaire est à  $-2$  kHz et est donc échantillonnée à 4 échantillons par période, comme le réclame le mélangeur P-Q. Le signe moins signifie ici qu'on travaille sur l'image miroir de la bande latérale. Après séparation des composantes P et Q, elles sont filtrées par un premier filtre CIC du second ordre puis subissent un sous-échantillonnage avec un facteur  $M_d = 16$ . Suit un autre filtre CIC passe-bas sans sous-échantillonnage. Les signaux filtrés P et Q vont alors permettre de déterminer l'amplitude et la phase instantanées par la méthode CORDIC.

Une possibilité de démodulation du signal serait d'utiliser une PLL pour récupérer la porteuse, puis de décoder le signal NRZ. Abréviation de *Non Return to Zero*, NRZ représente ici le signal de données d'origine avec lequel la modulation de phase a été opérée, précisément le signal utile que nous voulons récupérer. Pour le décodage, nous pouvons éviter la construction d'un circuit à PLL si nous considérons qu'une modulation de phase n'est rien d'autre qu'une modulation de fréquence. À l'observation de la **figure 7**, on peut adhérer à cette idée.

Le déroulement de la phase théorique est représenté sous la forme du signal  $p(t)$ . Comme il ne peut pas y avoir de saut de phase, mais seulement une lente variation continue pour économiser la bande passante, la courbe de phase réelle se présente plutôt comme le signal  $q(t)$ . L'échantillonnage du signal démodulé se produit à 400 Hz de manière à retrouver 16 échantillons par bit. Or, en technique de communication, la variation de la phase dans le temps correspond à la fréquence d'un signal. Par conséquent, la courbe de fréquence qui correspond à  $q(t)$  donne  $f(t)$  après dérivation en fonction du temps. Quand la phase s'accroît linéairement, elle produit une fréquence positive constante ; si la phase diminue linéairement dans le temps, elle produit une fréquence négative constante.

Par un nouveau filtrage passe-bas,  $f(t)$  produit le signal  $g(t)$ , lequel est facile à évaluer, il suffit de regarder la valeur au milieu du bit. Si elle est positive, la réception correspond à un 1. Si elle est négative, c'est un 0. De cette manière, on voit clairement comment le récepteur fonctionne.

Par différenciation des échantillons de phase issus de la méthode Cordic, on obtient la fréquence actuelle. On les passe de nouveau dans un filtre passe-bas CIC pour obtenir le signal  $g(t)$ . Ce signal approvisionne à son tour la récupération de l'horloge binaire. Il en résulte les formes d'ondes décrites à la figure 8 du deuxième article de la série [2].

La suite du traitement va s'occuper de la reconnaissance et du décodage des blocs de données. Nous verrons ensuite comment cela fonctionne. Au préalable, le **listage 3** montre les données reçues de la BBC sur 198 kHz.

### La synchronisation des blocs

Après l'extraction de la chaîne de bits du signal reçu, il ne reste plus qu'à repérer le début et la fin des blocs. C'est l'affaire de la synchronisation de blocs.

La plupart des protocoles de transmission par radio de données prévoient des blocs de longueur connue. Mais au moment de l'allumage du récepteur, comment savoir où ils commencent ? Il existe plusieurs méthodes. De temps en temps, un préambule reconnaissable est transmis, qui ne peut pas être confondu avec des données. C'est le cas par exemple avec les RFID du type EM4102. Tout préambule occupe de la place et réduit d'autant la charge utile. De nombreux protocoles se basent sur l'information de la correction d'erreur qui est déjà présente.

### La détection d'erreur

Beaucoup de systèmes de transmission de données par radio ajoutent un bloc de données constitué d'une série de bits qui peut être une somme (*checksum*) ou un mot de contrôle qui permet de détecter les erreurs de réception et éventuellement de les corriger. Comme on va le voir, ce complément d'information peut aussi servir à la synchronisation de blocs. Examinons d'abord avec l'émetteur de test à 125 kHz (logiciel EXP-TX-FM-125kHz-RDSLlike-V01.c) comment calculer le mot de contrôle. L'algorithme du **listage 4** est représenté à la **figure 8** comme schéma théorique.

Un registre à décalage appelé Syndrom, flanqué de portes OU exclusif (XOR) en rétroaction, calcule dix bits de contrôle. Les 16 bits de donnée sortent d'un registre à décalage et servent à contrôler les nouveaux bits qui passent dans le registre Syndrom. Le mot de contrôle de redondance cyclique CRC (*Cyclic Redundancy Check*) est calculé en 16 coups d'horloge. À ce mot, on ajoute encore un décalage d'une valeur de  $198_H$  pour faciliter la synchronisation. Les 10 bits de contrôle sont transmis à la suite des bits de données. Cette procédure est inspirée du système RDS [8].

Quand le récepteur sait où commence le nouveau bloc, Syndrom peut calculer, exactement comme l'émetteur l'a fait, le mot de contrôle sur les 16 bits qui arrivent. Il vérifie avec les dix bits suivants si les deux mots de contrôle sont identiques. Mais comment faire pour trouver le début d'un bloc ? À l'arrivée d'un nouveau bit, vérifier sur les 26 bits qui précèdent s'ils présentent une combinaison de 16 bits de donnée et 10 bits de contrôle sans erreur. Le test devrait être concluant à une distance de 26 bits. Mais comme il peut arriver que le compte soit bon sur une série quelconque de bits qui n'appartiennent pas à la même trame, il faut garder trace du rythme et rechercher si la correspondance se répète trois fois de suite à 26 bits d'intervalle. On est alors sûr d'avoir bien trouvé le début du bloc.

La **figure 9** vous montre schématiquement comment les bits de contrôle sont calculés en permanence. On peut considérer que le calcul avec Syndrom est aussi une sorte de filtrage. En réalité, le registre à décalage à rétroaction est exactement le même que celui de l'émetteur. Les bits qui entrent y sont traités et quand l'un d'eux sort du registre à  $10+16$  bits, il est également pris en charge dans Syndrom. Le logiciel se contente de vérifier si la valeur de  $198_H$ , qui a été ajoutée au résultat, est actuellement dans le registre Syndrom. Chaque fois que le logiciel y trouve cette valeur, il dit OK. Il y a aussi

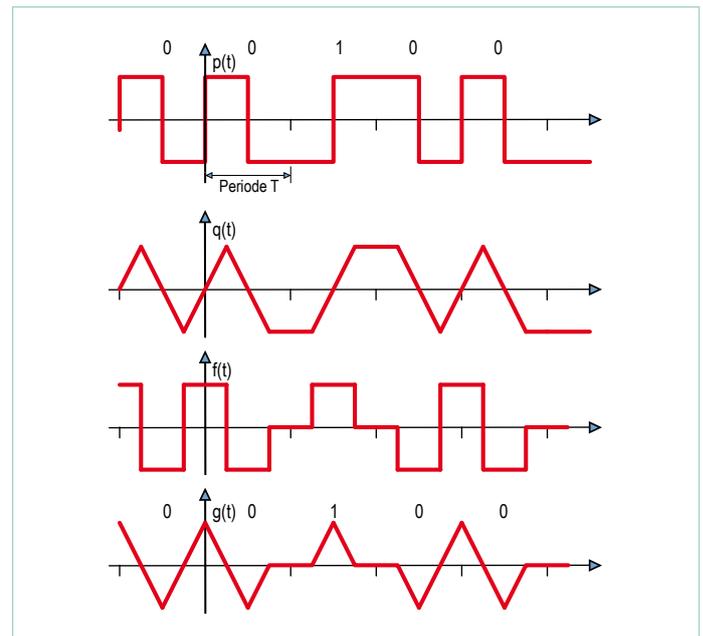


Figure 7. La modulation de phase comme modulation de fréquence.

#### Listage 3 : les données reçues de la BBC sur 198 kHz

```
+E EB11 0391
+0 AAAA AAAA
+E E8F1 0391
+E EB09 0391
+E EB49 0391
+0 AAAA AAAA
+E EB31 0391
+E EB41 0391
+0 5475 2980 time= 18:38 week=07 day=02 tuesday
+E EB29 0391
```

#### Listage 4 : calcul du CRC à l'émetteur.

```
SendBits(w,16) ;
Syndrom=0 ;
for (j=0 ; j<16 ; j++) {
    Syndrom=Syndrom<<1 ; // outgoing bit is at mask
    0x0400
    if (w & 0x8000) {
        Syndrom ^= 0x0400 ; // exor into outgoing bit
    }
    w=w<<1 ;
    if ( Syndrom & 0x0400 ) {
        Syndrom ^= 0x1B9 ;
    }
} ;
Syndrom=(Syndrom^ofs)<<6 ;
SendBits(Syndrom,10) ;
```

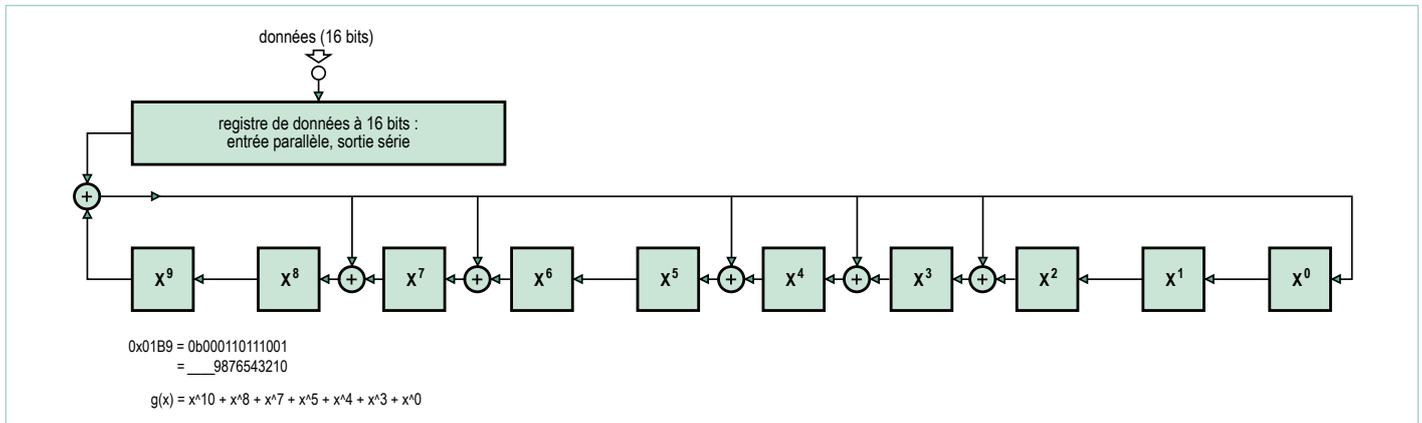


Figure 8. CRC1.

un compteur, du nom de localTime, qui tourne toujours de 0 à 25. Quand il est revenu à zéro, c'est qu'un bloc complet, bits de contrôle inclus, est passé.

Pour illustrer ce processus, j'ai programmé une variante du récepteur qui met en service une sortie MLI du convertisseur N/A pour lui faire générer une dent de scie, image de l'état du compteur localTime. La deuxième sortie MLI donne toujours un niveau haut quand le circuit de synchronisation détecte le OK. On en voit bien le rythme à la **figure 10**. Mais aussi le cas d'une fausse correspondance, un OK pour rien. Il faut dire que l'attente est longue avant d'en apercevoir une.

Si, après la période de synchronisation, vous recevez un OK au temps zéro, c'est que le bloc reçu est sans erreur, les données peuvent être transmises. Et notre récepteur est complet.

Voilà qui conclut l'explication de la synchronisation sur BBC198. Mais sachez que les paramètres du RDS (*Radio Data System*) ne sont pas les mêmes, ils constituent des blocs de 47 bits de données plus 13 bits de contrôle. Les données émises par la BBC sur 198 kHz le sont en blocs plus faciles à décoder en se référant à la documentation [7]. En particulier, l'information horaire est condensée en un seul bloc.

### Réception de BBC648 AMSS avec décodage

Au départ, j'avais prévu un dernier exemple avec la réception de la BBC sur 648 kHz. Pas de chance, entre-temps, cet émetteur a cessé

son activité, ce qui nous prive d'expérimentation pratique avec lui. Je vais cependant vous décrire en bref le procédé, parce qu'il permet de comprendre comment les données AMSS sont transmises. Il n'y a pas si longtemps que BBC648 a cessé d'émettre et une abondante littérature [10] est disponible. Notre syntoniseur simple échantillonne à  $20\,000 / 1\,875 = 10,666\dots$  kHz. Il en découle une fréquence intermédiaire de  $648 - 61 * 10,666\dots = -2,666\dots$  kHz qui est le quart de  $10,666\dots$  kHz. La modulation employée est biphase avec une modulation de phase de  $\pm 20^\circ$ .

Pour ce qui est de la modulation et du codage binaire, c'est (presque) les mêmes que sur 198 kHz, seuls les paramètres sont légèrement modifiés. Les blocs comptent toujours 36 bits utiles et 11 bits de contrôle CRC. Deux de ces blocs forment un groupe et un certain nombre de groupes donnent un bloc DRM SDC. Pour s'y retrouver, il faut jeter un œil dans la spécification DRM. Notre récepteur simple ne peut qu'en extraire et afficher le nom de la station, mais au moins il reçoit des données et fonctionne correctement. Le **listage 5** vous montre les données reçues. L'expression « 2/7+ » signifie que le deuxième groupe de sept a été reçu sans erreur. Quand on a reçu les sept groupes au complet, on décode les données.

### Conclusion

Nous voici à la fin de cette série d'articles sur le traitement du signal et la radio logicielle ou *Software Defined Radio* sur contrôleur AVR.

#### Listage 5 : BBC sur 648 kHz : données AMSS-SDC

```
[6 1]BBC WS [C B] [6 B] 0/7+ 1/7+ 2/7+ 3/7+ 4/7+ 5/7+ 6/7+ 7/7+!0007 ;
[6 1]BBC WS [C B] [6 B] 0/7+ 1/7+ 2/7+ 3/7+ 4/7+ 5/7+ 6/7+ 7/7+!0007 ;
[6 1]BBC WS [C B] [6 B] 0/7+ 1/7+ 2/7+ 3/7+ 4/7+ 5/7+ 6/7+ 7/7+!0007 ;
[6 1]BBC WS [C B] [6 B] 0/7+ 1/7+ 2/7+ 3/7+ 4/7+ 5/7+ 6/7+ 7/7+!0007 ;
[6 1]BBC WS [C B] [6 B] 0/7+ 1/7+ 2/7+ 3/7+ 4/7+ 5/7+ 6/7+ 7/7+!0007 ;
```

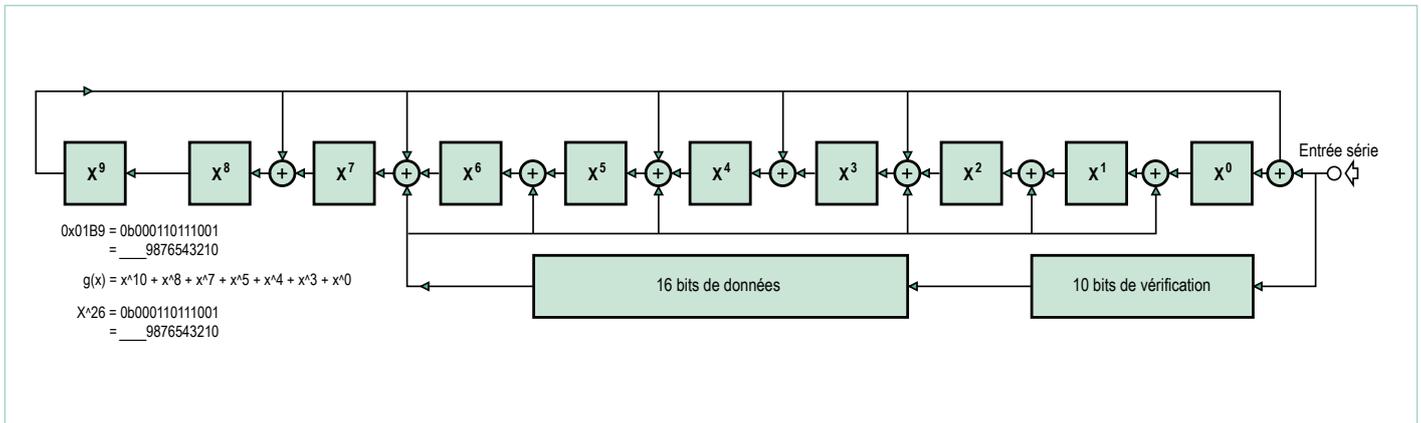


Figure 9. CRC2.

Ils ont permis de montrer qu'il est possible de réaliser beaucoup de choses avec ces microcontrôleurs simples. Sur les « vrais » récepteurs SDR, on utilise exactement les mêmes procédés que ceux qui ont été décrits ici, mais avec des syntoniseurs et des DSP nettement plus puissants.

(120392 – version française : Robert Grignard)

## Liens

- [1] [www.elektor.fr/100180](http://www.elektor.fr/100180)
- [2] [www.elektor.fr/100181](http://www.elektor.fr/100181)
- [3] [www.elektor.fr/100182](http://www.elektor.fr/100182)
- [4] [www.elektor.fr/120088](http://www.elektor.fr/120088)
- [5] [www.elektor.fr/120089](http://www.elektor.fr/120089)
- [6] [www.elektor.fr/120392](http://www.elektor.fr/120392)
- [7] <http://downloads.bbc.co.uk/rd/pubs/reports/1984-19.pdf>
- [8] <ftp://ftp.rds.org.uk/pub/acrobat/rbds1998.pdf>
- [9] [www.drm.org/](http://www.drm.org/)

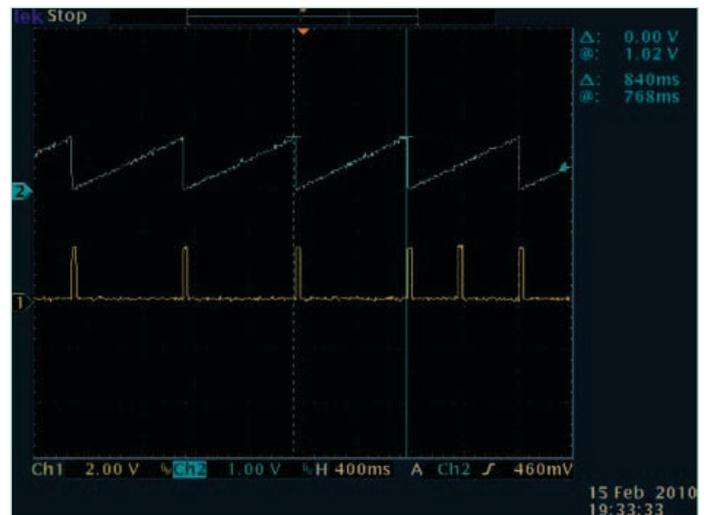


Figure 10. Le signal OK.

- [10] [www.ebu.ch/fr/technical/trev/trev\\_305-murphy.pdf](http://www.ebu.ch/fr/technical/trev/trev_305-murphy.pdf)  
[www.broadcastpapers.com/whitepapers/ABUBBCamss2006.pdf?CFID=16508900&CFTOKEN=dac28b1a87e54d77-47F9A337-9A30-F5E3-667FAB2A9EA27223](http://www.broadcastpapers.com/whitepapers/ABUBBCamss2006.pdf?CFID=16508900&CFTOKEN=dac28b1a87e54d77-47F9A337-9A30-F5E3-667FAB2A9EA27223)

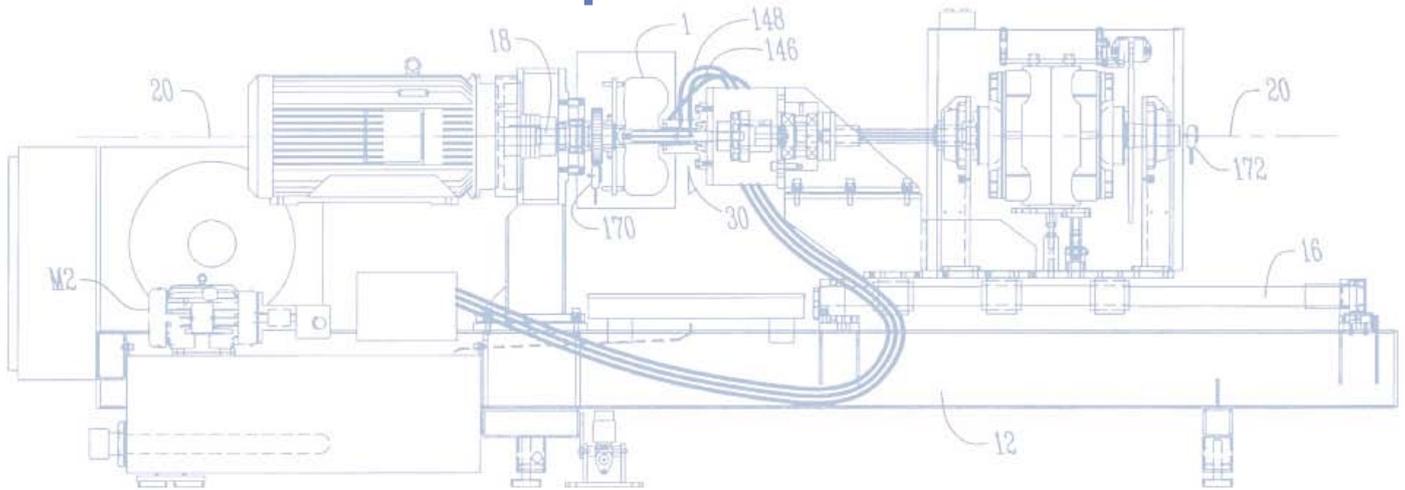
## Produits et services Elektor

- Générateur de signaux (module prêt à l'emploi : 100180-71)
- Récepteur universel (module prêt à l'emploi : 100181-71)
- Antenne ferrite active (module prêt à l'emploi : 100182-71)
- Ensemble des trois modules ci-dessus plus convertisseur USB/TTL BOBFT232 : 100182-72
- Convertisseur USB/TTL BOBFT232, câblé et testé : 110553-91
- Programmeur USB AVR, platine à CMS implantés plus les autres composants : 080083-71
- Téléchargement gratuit des logiciels (fichiers hexadécimal et de code source)

Retrouvez tous les produits ici : [www.elektor.fr/120392](http://www.elektor.fr/120392)

# pratique de la commande des moteurs pas-à-pas (3)

## histoires de couple



Cet article décrit un dynamomètre simple qui vous permettra de caractériser le comportement de moteurs pas-à-pas liés à différentes charges. Si sa précision n'est pas digne d'un équipement de laboratoire, il vous donnera quand même une bonne idée des performances réelles de vos moteurs.

Ed Nisley (Poughkeepsie)

$$\text{couple} = \frac{\text{puissance}}{\text{vitesse}}$$

Maintenant que nous avons vu dans les deux premiers volets de cet article comment un pilote par micro-pas régule le courant dans les enroulements d'un moteur pas-à-pas, il est temps d'examiner ce qui nous intéresse au final : le couple produit sur l'axe.

### Dynamomètre : principe et utilisation

L'idée d'utiliser un dynamomètre pour mesurer la puissance d'un moteur remonte à 1821, époque où Gaspard de Prony a inventé le frein de Prony : une balance à ressort fixée à un bras solidaire d'un frein à friction entourant l'axe du moteur. La balance à ressort indique la force de freinage qui, multipliée par la longueur du bras, donne le couple du moteur. En multipliant couple et vitesse on obtient la puissance du moteur : **puissance = couple × vitesse**

Resserrer un frein à friction sur l'axe d'un moteur fonctionne bien mieux sur les modèles dont la puissance se compte en kilowatts ou chevaux que sur les petits moteurs pas-à-pas que j'utilise. La force appliquée par un tel frein varie à cause des inévitables irrégularités sur la surface de freinage et je m'attends à ce que ces variations soient relativement grandes vis-à-vis du couple moyen d'un petit pas-à-pas.

La feuille de caractéristiques d'un moteur pas-à-pas donne son couple, plutôt que la puissance en sortie, il nous faudra donc transformer notre équation et changer d'approche. Le rapport de deux grandeurs facilement mesurables donne le couple :

Mesurer une vitesse ne requiert rien de plus qu'un capteur optique similaire à celui que j'ai utilisé le mois dernier et, comme nous transformons de l'énergie mécanique en énergie électrique, la puissance sera facile à mesurer. Un tableur se chargera des calculs : tout ce qu'il nous faut c'est des mesures.

C'est dans cet esprit que j'ai réalisé le dynamomètre de la **photo 1** constitué de deux moteurs pas-à-pas NEMA 17 reliés par un axe. Le petit, le même Minebea 17PM-J034 que dans l'article du mois dernier, est le moteur testé. Le gros, un ACT 17HS5426, est utilisé en générateur et dissipe son énergie dans les deux résistances de puissance à corps en aluminium visibles à gauche. Je possède un bel assortiment de ces résistances et il m'est donc facile de faire varier la puissance dissipée pour une tension donnée.

Sur la plaque à essais sans soudure se trouve une platine un pilote de moteurs pas-à-pas Pololu A4988 sous un petit dissipateur en aluminium, les interrupteurs DIP servent à choisir le type de micro-pas. Référez-vous à la figure 1 de l'article du mois dernier pour les détails de câblage ; la remarque sur l'inadéquation des plaques à essais pour les câblages de puissance est toujours d'actualité.

Les deux pinces crocodile juste en dessous de l'interrupteur DIP relient un générateur d'impulsions à l'entrée *Pas* du circuit. La fréquence des impulsions détermine donc la vitesse du moteur, vitesse que je peux faire varier très lentement afin d'éliminer l'influence des accélérations mécaniques sur les valeurs mesurées.

Le disque cranté qui se trouve à la jonction des deux axes interrompt le faisceau IR de la fourche optique visible en bas de la photo ; une résistance sur la plaque à essais permettra d'obtenir une impulsion par tour d'axe. Comme nous ne nous intéressons qu'aux tours complets du moteur, nous n'avons pas besoin du générateur de synchro à Arduino utilisé le mois dernier.

Mesurer la puissance en watts (en l'honneur de James Watt, qui a rendu utilisables les premières machines à vapeur) est peut-être naturel pour un ingénieur, mais nous pensons la vitesse en tours par minute (tr/min ou rpm) ou par seconde (tr/s), plutôt que dans l'unité SI qui est le radian par seconde (rad/s). Il faut donc convertir les tr/min en rad/s pour obtenir le couple du moteur en Newton mètres (N·m) :

$$N \cdot m = \frac{W}{rpm \times 2\pi / (60 \text{ s} / \text{min})}$$

La **figure 1** représente l'amplitude de crête de notre moteur-générateur 17HS5426 à vide en fonction de la vitesse de rotation de l'axe. La pente de cette droite étonnamment... droite nous permet de calculer la constante Kg du générateur, ici Kg = 16,7 mVc/(tr/min). La valeur de Kg dépend de la construction du générateur, vous devrez donc la mesurer pour chaque moteur que vous voudrez utiliser en générateur. Par exemple, pour un Minebea J034, j'ai mesuré Kg = 14 mVc/(tr/min).

À strictement parler, le moteur 17HS5426 est un alternateur : il produit une tension alternative. J'aurais pu utiliser un moteur à courant continu, mais j'ai préféré utiliser deux moteurs pas-à-pas pour faciliter l'assemblage et pouvoir intervertir facilement moteur et générateur.

Bien que la tension de sortie varie linéairement avec la vitesse de l'axe, la puissance dissipée dans la résistance de charge varie avec le carré de la vitesse de rotation :

$$W = \frac{E^2}{R} = \frac{(K_g \times rpm)^2}{R}$$

Aux très faibles vitesses, la tension de sortie est proche de zéro et c'est la résistance d'enroulement du générateur qui limite la puissance maximale disponible. Par chance, les moteurs pas-à-pas produisent un couple raisonnablement constant aux faibles vitesses, et l'extrapolation des résultats obtenus aux vitesses moyennes donnera des résultats utiles.

Faire fonctionner le dynamomètre mettra à l'épreuve vos talents de manipulateur. L'oscillogramme de la **photo 2** montre l'évolution de la tension aux bornes d'une résistance de 4 Ω à mesure que j'augmente lentement la vitesse jusqu'à l'arrêt du moteur. La tension de sortie vaut environ 3,25 Vc juste avant que le moteur J034 ne puisse plus suivre la charge croissante, que le rotor s'arrête et que la tension chute. Toutefois, celle-ci ne tombe pas à zéro, étant donné que le pilote de pas-à-pas continue de produire des courants dans les enroulements qui font vibrer l'axe autour de sa position d'arrêt. J'ai calculé la dissipation énergétique en utilisant la tension RMS,

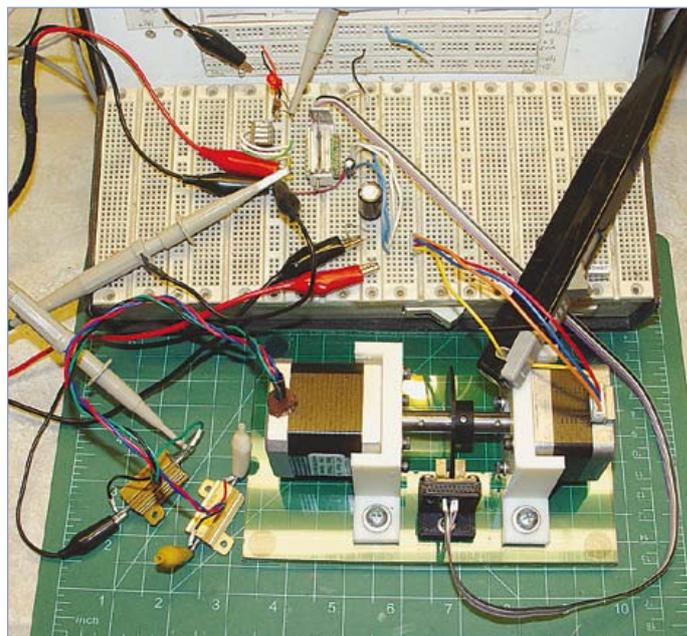


Photo 1. Une petite platine Pololu de pilotage de pas-à-pas dotée d'un A4988 fait tourner le moteur Minebea, une sonde de courant Tektronix surveillant son courant d'enroulement. Le moteur ACT, plus gros, fonctionne en génératrice et dissipe de l'énergie dans les deux résistances de puissance en aluminium. Le disque solidaire de l'axe, associé à une fourche optique, permet de produire une impulsion par tour pour déclencher l'oscilloscope.

plutôt que la valeur crête, afin de déterminer la puissance équivalente en courant continu et m'affranchir des petites distorsions du signal. L'oscilloscope calculant la valeur RMS directement à partir de la forme d'onde, le choix aura été facile. Cependant, le scope traite la totalité des données enregistrées et j'ai donc dû faire la mesure de telle manière que le blocage du moteur ne soit pas visible sur l'écran et n'affecte pas les résultats. Le calcul à la main pour les 3,25 Vc de la photo 2 donne 2,30 Vrms.

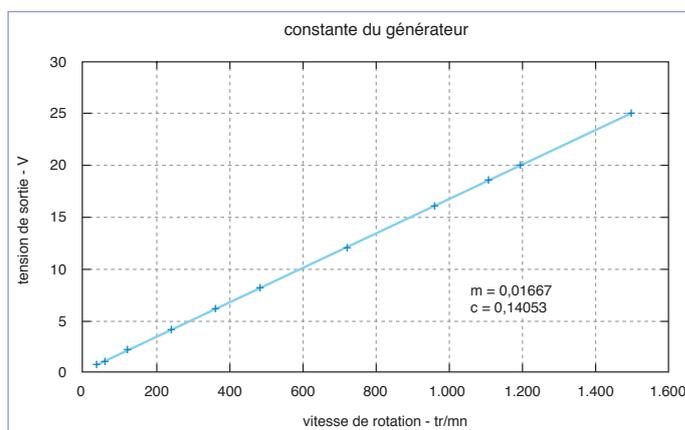


Figure 1. La constante Kg du gros moteur pas-à-pas utilisé comme générateur lorsqu'il est à vide est égale au coefficient directeur de la droite liant la tension de sortie à la vitesse de rotation. Le fait que la droite ne passe pas par zéro résulte de l'imperfection de mes mesures : la tension en sortie du moteur n'est pas de 140 mV lorsqu'il est à l'arrêt. Sans charge en sortie, le petit moteur pas-à-pas se bloque à environ 1500 tr/min.

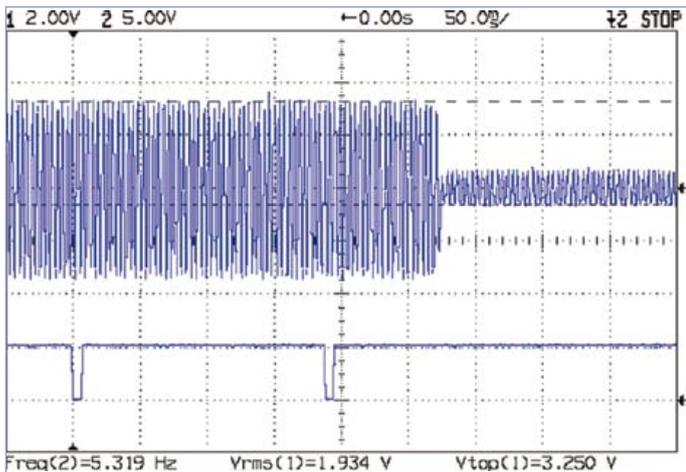


Photo 2. Le moteur se bloque lorsque la charge imposée par le générateur dépasse le couple produit par le pilote. La chute de tension rapide et l'absence des impulsions de synchronisation montrent que le rotor s'arrête quasi instantanément après le blocage du moteur ; seule une faible tension subsiste à cause de la vibration de l'axe du moteur.

Le courant dans la résistance de 4 Ω est :

$$575 \text{ mA} = \frac{2,30 \text{ V}}{4 \Omega}$$

Les enroulements du 17HS5426 possèdent une résistance de 1,1 Ω à ajouter aux 4,0 Ω de la résistance externe ; la résistance totale du circuit est donc de 5,1 Ω. La puissance pour un enroulement est donc : 1,7 W = (0,575 A)<sup>2</sup> × 5,1 Ω.

Étant donné que la tension et le courant des deux enroulements sont déphasés de 90°, le couple dépend du courant de pointe d'un seul enroulement, plus grand d'un facteur √2 que la valeur RMS. La puissance crête correspondante est deux fois plus grande que la valeur calculée à partir de la valeur RMS du courant ou de la tension. Quiz éclair : démontrez qu'une augmentation d'un facteur √2

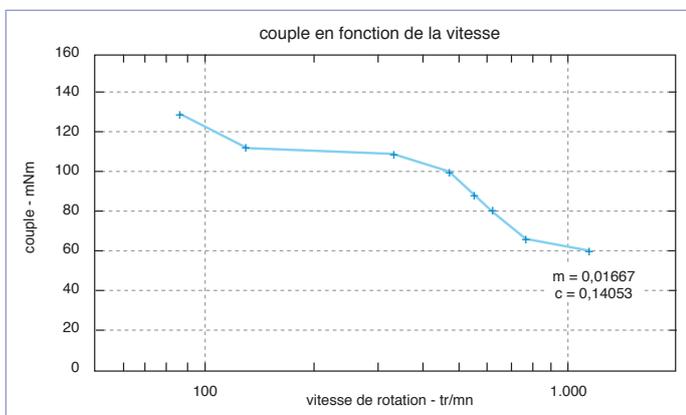


Figure 2. Ces données montrent le couple maximal du moteur 17PM-J034 de Minebea pour diverses vitesses en mode micro-pas 1/8 avec des pointes de courant à 1 A. L'échelle logarithmique sur l'axe des abscisses permet de représenter la grande étendue des valeurs de vitesse : des rapports de vitesse égaux ont la même taille.

du courant traversant une résistance double la puissance qu'elle dissipe.

Les impulsions de tour représentées sur la voie du bas de l'oscillogramme de la photo 2 correspondent à une vitesse de rotation de 5,32 tr/s = 319 tr/min = 33,4 rad/s, soit une fréquence en pas entiers de 1060 pas/s. Le couple du moteur est donc :

$$100 \text{ mN} \cdot \text{m} = \frac{2 \times 1,7 \text{ W}}{33,4 \text{ rad/s}}$$

Le moteur 17PM-J034 ne figure pas au catalogue Minebea, un moteur 17PM-J049 supposément similaire possède un couple de décrochage d'environ 190 mN·m à 300 tr/min sous son courant nominal de 0,7 A.

Cependant, la résistance d'enroulement de 6,2 Ω du moteur J049 dissipe 3 W sous son courant nominal. Les enroulements de 2,1 Ω de mon J034 suggèrent qu'un courant nominal de 1,2 A entraînerait la même dissipation énergétique. Par extrapolation, un courant de 1,2 A en sortie du pilote Pololu de mon prototype (au lieu de 1 A) produirait un couple d'environ 120 mN·m, mais vous vous apercevrez rapidement que de telles extrapolations ne tiennent pas la route.

Les données de Minebea supposent une alimentation sous 24 V plutôt que les 12 V de mon montage de test. Comme nous l'avons vu dans les deux volets précédents, une tension de commande supérieure produit des changements plus rapides du courant dans les enroulements, ainsi qu'une augmentation du couple en sortie. Je m'attends aussi à ce que les ingénieurs de Minebea aient optimisé les conditions de test pour montrer leurs moteurs de manière avantageuse.

J'ai beau ne pas connaître le couple attendu, je pense que les résultats donnent une bonne approximation. Abstraction faite des erreurs absolues, ce dynamomètre est capable de produire des données relatives pertinentes pour pas mal de conditions de pilotage et de charge.

Exercice : construisez un frein de Prony à friction et mesurez le couple d'un moteur dont vous connaissez les caractéristiques. Question bonus : donnez une estimation des erreurs possibles.

### Couple en fonction de la vitesse

Un pilote de moteur pas-à-pas idéal imposerait un courant constant dans les enroulements pour chaque micro-pas, quelles que soient la vitesse de rotation et donc la durée des micro-pas. Cela veut dire que le moteur fournit une puissance constante durant le micro-pas et, comme le couple est inversement lié à la vitesse pour une puissance donnée, que le couple décroît à mesure que la vitesse augmente.

Mais les désagréments de la pratique, dont l'incapacité du pilote à produire des courants de pas rectangulaires à cause des effets inductifs, font que le couple décroît de plus en plus vite à mesure que la vitesse des micro-pas augmente. Au-delà de quelques milliers de tours par minute, le moteur ne produit plus qu'une fraction de son couple nominal.

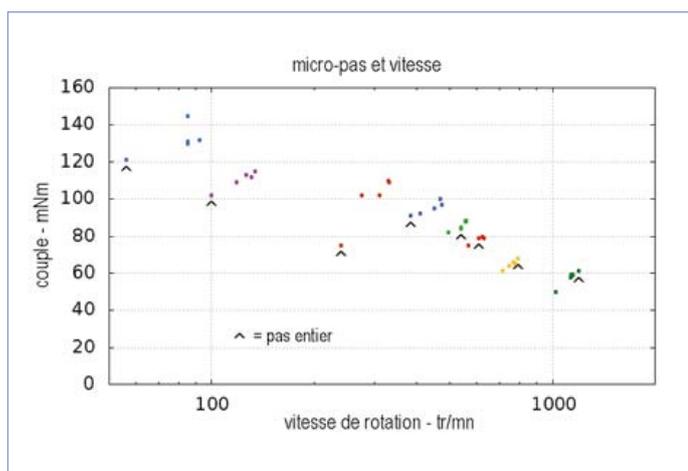


Figure 3. Chaque mode du A4988 d'Allegro produit une relation différente entre couple et vitesse pour une résistance de charge donnée. Le mode par pas entiers ne produit pas systématiquement plus de couple que les modes par micro-pas.

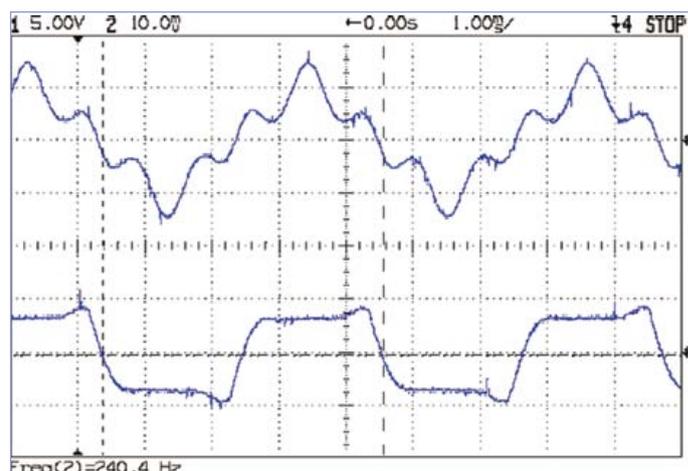


Photo 3. Le fonctionnement à 960 pas/s en mode par pas entiers produit une résonance mécanique à 1200 Hz. La courbe du haut montre la tension en sortie du générateur à vide, celle du bas représente le courant de commande du moteur à 1 A/div.

Inversement, jusqu'à des vitesses de quelques centaines de tours par minute, quand le pilote est encore capable de produire des courants de pas presque rectangulaires, le moteur produit un couple quasi-constant limité par la capacité de l'induit à convertir le courant électrique en champ magnétique, quelle que soit la vitesse. Les moteurs pas-à-pas ont donc tendance à être plus utiles dans les applications où le couple est relativement élevé et la vitesse basse, lorsque la vitesse de l'axe est comprise entre presque zéro et quelques centaines de tours par minute.

Comme nous venons de le voir, les données du dynamomètre donnent le couple du moteur en charge lorsque l'axe est quasi immobile. La figure 1 montre que la tension à vide du générateur dépend linéairement de la vitesse, la résistance doit donc varier pour imposer une certaine charge pour une vitesse donnée. Cela nécessiterait des rhéostats de puissance assortis, que je ne possède pas, j'ai donc choisi des paires assorties de résistances de puissance depuis mon tas et augmenté la vitesse jusqu'à ce que la tension croissante du générateur dissipe assez de puissance pour stopper le moteur. Les points de données ne seront donc pas espacés régulièrement à cause des variations de couple.

Le graphique de la **figure 2** montre le couple maximal disponible pour chaque vitesse en mode micro-pas 1/8 et lorsque le pilote applique un courant d'enroulement de 1 A. Le point le plus à gauche a été obtenu en mesurant le courant de court-circuit du générateur tandis que les autres l'on été en mesurant le courant à travers des résistances de puissance, comme décrit précédemment. Le point le plus à droite correspond à une paire de résistances de 50 Ω.

Le moteur produit un couple raisonnablement constant jusqu'à environ 500 tr/min avant qu'il ne commence à décroître avec la vitesse, comme prévu. Le dernier point de donnée paraît un peu trop haut pour s'inscrire dans le prolongement de la courbe, mais il s'agit de vraies mesures !

### Couple en fonction du courant d'enroulement

J'ai testé le moteur avec des courants d'enroulement de 1 A et 500 mA pour mettre à l'épreuve le mantra bien connu « le couple est proportionnel au courant » : un courant diminué de moitié devrait

produire moitié moins de couple. En mode micro-pas 1/8 et avec des résistances de charge de 1 Ω, les couples étaient 112 mN·m à 131 tr/min et 45 mN·m à 36 tr/min, les deux vitesses se trouvant dans la gamme « couple constant » du moteur.

Le rapport entre les couples est de 2.5:1 ou 1:0.40, à comparer aux chiffres attendus de 2:1 ou 1:0.5. Cependant, la vitesse du moteur varie presque d'un facteur quatre, ce qui affecte à la fois les formes d'onde de la commande en micro-pas et le couple. Rétrospectivement, j'aurais dû faire varier les résistances comme l'inverse du carré du courant afin que la vitesse d'arrêt du moteur soit plus ou moins la même.

Il est raisonnable d'utiliser la proportionnalité comme recette de cuisine, mais il ne faut pas s'attendre à des résultats exacts. Comme toujours, il faut mesurer les choses dans leurs conditions de fonctionnement.

### Couple en fonction du mode de micro-pas

Étant donné que le couple varie linéairement avec le courant d'enroulement, le courant réduit imposé par un pilote par micro-pas devrait diminuer le couple disponible sur le moteur. Le graphique de la **figure 3** montre le couple, moteur bloqué, pour tous les modes de pas de l'A4988 (par pas entiers et micro-pas 1/2, 1/4, 1/8, et 1/16) ; les couleurs correspondent aux huit différentes résistances de charge allant de zéro (court-circuit) à 50 Ω.

Un lambda met en valeur le mode par pas entiers pour chaque groupe de cinq points. Ce n'est qu'à partir d'environ 1000 tr/min que le mode par pas entiers produit un couple supérieur à celui obtenu avec les modes micro-pas et, au cas où vous vous poseriez la question : les erreurs de mesure ne sont pas significatives.

Vous pourrez examiner les données brutes en téléchargeant la feuille de calcul pour cet article.

### Résonance et micro-pas

La structure rigide de notre dynamomètre fait de métal et plastique fournit un terrain de jeu idéal pour expérimenter autour des résonances mécaniques. J'ai choisi le mode par pas entiers sur les interrupteurs DIP et réglé le rythme des pas jusqu'à être en plein dans la zone de résonance assourdissante de la **photo 3**.

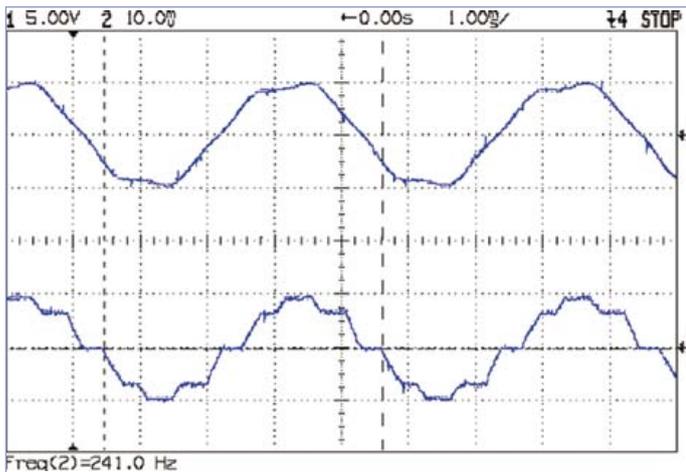


Photo 4. Le fonctionnement à 1920 pas/s en mode micro-pas 1/2 élimine la majeure partie de la résonance à 1200 Hz, même s'il subsiste une distorsion considérable de la tension de sortie du générateur à vide.

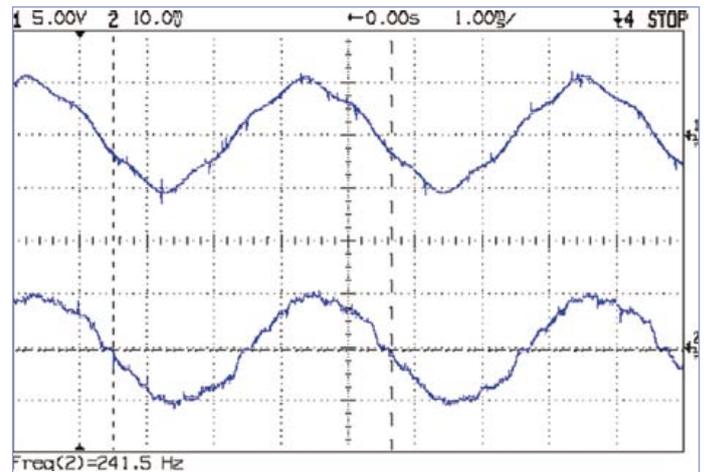


Photo 5. A 3840 pas/s en mode micro-pas 1/4, il ne reste plus grand-chose de la résonance mécanique à 1200 Hz sur la sortie du générateur visible sur la courbe du haut.

Le courant d'enroulement visible sur la courbe du bas possède une fréquence de 240 Hz soit 960 pas/s ou 4,8 tr/s à 200 pas/tr. Le courant du moteur, un signal carré avec un pic à 0,7 A représente ce que produit le A4988 en mode pas entiers : les deux enroulements sont stimulés à  $1/\sqrt{2}$  de la valeur crête de 1 A. Les petits sursauts qui apparaissent juste avant les transitions montrent que le A4988 ne commande pas complètement le courant, même s'il ne se comporte pas mal du tout face à la puissante résonance.

La tension en sortie du générateur à vide sur la courbe du haut montre les 5 oscillations du rotor qui se produisent lors des 4 pas qui composent chaque cycle du courant, ce qui impose une résonance à 1200 Hz sur le signal fondamental à 240 Hz. Les vibrations desserrent les écrous du manchon en acier qui relie les deux axes en à peu près une minute ; sur une installation permanente, les filetages doivent être bloqués avec une résine de freinage !

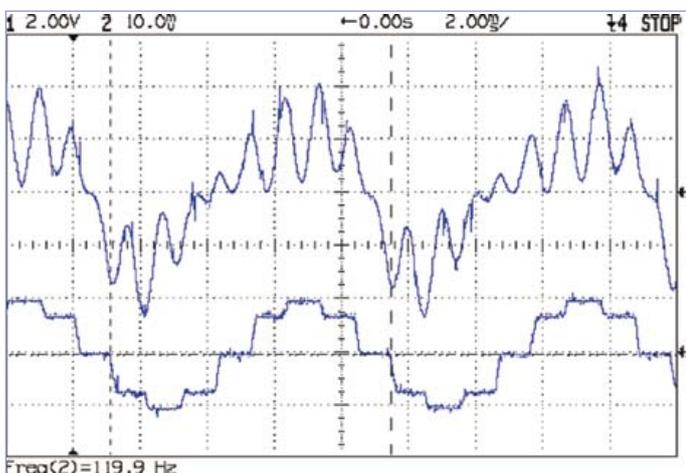


Photo 6. Le fonctionnement à 960 pas/s en mode micro-pas 1/2 déclenche la même résonance mécanique à 1200 Hz, mais selon des cycles irréguliers commandés par les transitions entre les micro-pas successifs.

Le passage en mode micro-pas 1/2 double la fréquence des pas à environ 1920 pas/s et la majeure partie de l'énergie ne se trouve plus sur la fréquence de résonance. La sortie du générateur, visible sur la courbe de la **photo 4** montre qu'une distorsion subsiste, mais le moteur produit beaucoup moins de bruit et de vibrations.

L'utilisation du mode micro-pas 1/4 produit des résultats encore meilleurs que vous pourrez admirer sur la **photo 5**. Une des conséquences des modes par micro-pas est l'augmentation de la fréquence de pas pour une vitesse de rotation donnée : produire des pas avec un timing précis entraîne une vraie difficulté de programmation des micrologiciels de commande. Si vous vous attellez à ce genre de tâche, utilisez des temporisateurs matériels pour produire les impulsions !

Enfin, vous verrez sur la **photo 6** que les mode par micro-pas ne résolvent pas tous les problèmes de résonance mécanique. A 960 pas/s et en mode micro-pas 1/2, il y a suffisamment d'énergie produite pour réveiller la résonance qui hurle à 1200 Hz, bien que dans ce cas la fréquence des cycles successifs varie entre un peu en dessous de 1000 Hz et environ 1300 Hz.

## On coupe le contact

Il est certainement possible d'améliorer le dynamomètre présenté ici, même si en l'état il fournit déjà des données utiles. Vous vous apercevrez que ce que l'on trouve sur les feuilles de caractéristiques est souvent différent de ce que l'on mesure, et que tous deux sont différents de ce dont vous avez réellement besoin.

Ces moteurs pas-à-pas qui vous paraissent numériques auront vite fait de vous faire plonger dans la mécanique avant même que votre projet ne sorte du labo !

(120360 – version française : Kévin PETIT)

# Le labyrinthe de l'électronicien perdu

L'électronicien égaré dans le numéro de juillet-août a retrouvé son chemin parmi les diodes, les zeners, les résistances, les bobines, les condensateurs et les boutons. À la question de savoir à laquelle des trois entrées du labyrinthe (p.64) il fallait appliquer une tension de 20 V pour que le courant arrive à la sortie, des centaines de réponses ont été faites, dont certaines soulèvent d'ailleurs une controverse. Selon l'auteur du labyrinthe, la réponse juste est l'entrée 2 et nous nous tiendrons à son avis.

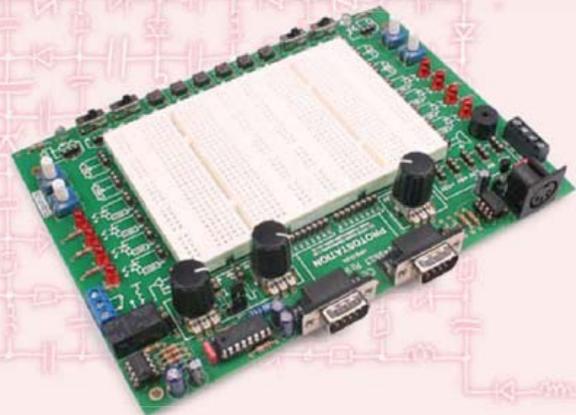
Comme il a non seulement donné la réponse exacte, mais aussi deviné (à un chouïa près) le nombre de réponses exactes reçues par Elektor, le gagnant :

**M. Evan Wasserman**

Lakewood, New Jersey, États-Unis d'Amérique

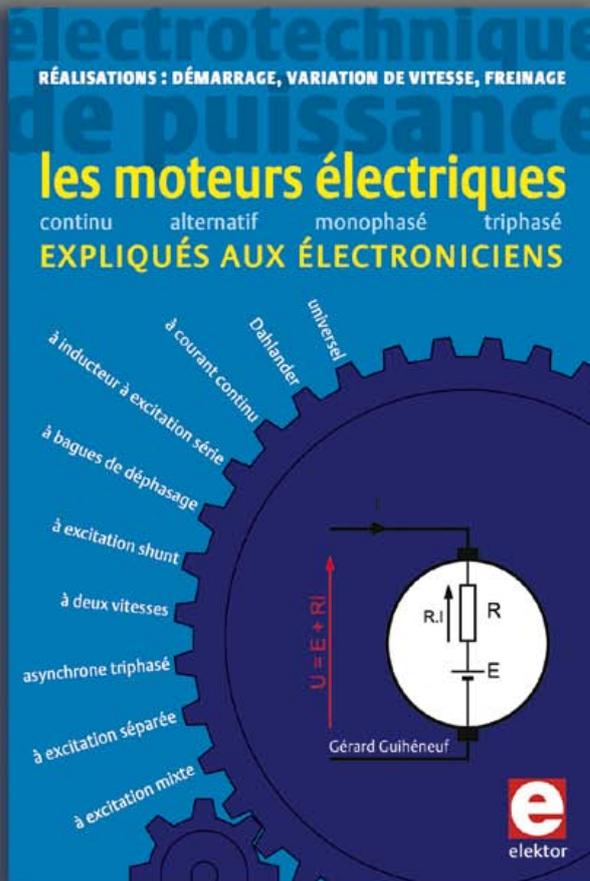
emporte le prix parrainé par **Matrix Multimedia** sous la forme d'une station **Electronic Workstation - Desktop** d'une valeur de **1000 €** accompagnée de la carte de prototypage **Protostation Advanced Breadboard** d'une valeur de **150 €**.

(120537)



Publicité

nouveau livre



Trois chapitres s'appuient sur une information accessible et digeste : constitution, fonctionnement, caractéristiques, domaines d'utilisation des différents types de moteurs, 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 continu/continu (hacheurs à transistors IGBT). Mise en pratique immédiate avec un variateur pour mini-perceuse et un variateur de vitesse 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 une tension triphasée 3 × 400 V.

978-2-86661-188-0 | 320 pages | 37,50 €

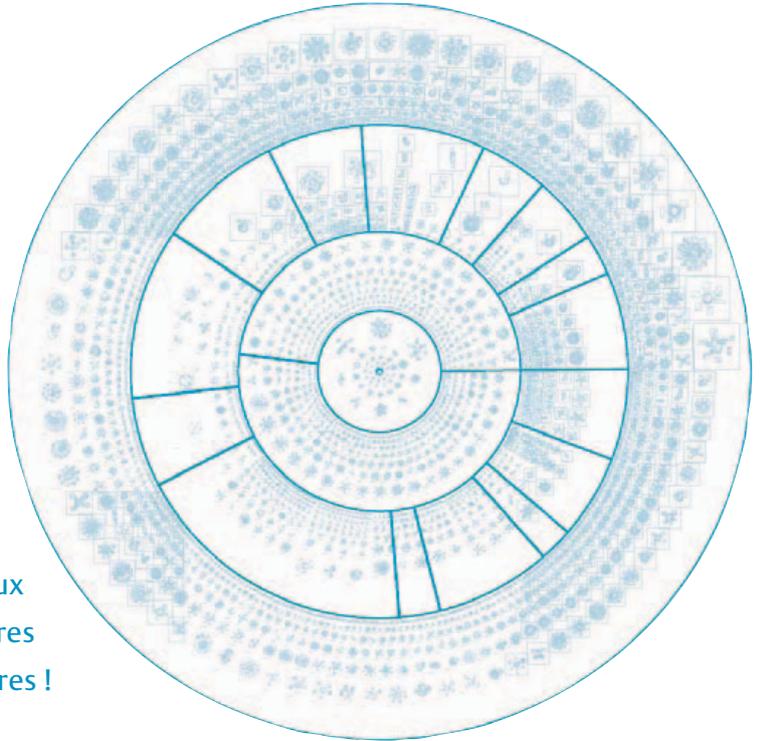
[www.elektor.fr/moteurs](http://www.elektor.fr/moteurs)

# embarquez Linux ! (4)

## Tout tourne autour du noyau

Benedikt Sauter [1]

Le système d'exploitation GNU/Linux fournit toutes les interfaces pour périphériques et applications nécessaires aux ordinateurs ou serveurs modernes. Le monde de l'embarqué n'est pas en reste : que ce soit dans le domaine des réseaux, de l'USB ou encore de Bluetooth, le programmeur qui adopte Linux s'épargnera les coûteux développements de ses propres solutions en C, un avantage linuxien parmi bien d'autres !



Dans l'épisode précédent, nous avons installé la chaîne d'outils, compilé noyau et chargeur de démarrage, et vu comment monter un support de données dans le système de fichiers. Soulevons maintenant le capot pour voir ce qu'il se passe au plus près du noyau, par exemple lors de la gestion des périphériques et des processus. Nous découvrirons la valeur ajoutée qu'apporte un système d'exploitation par rapport au classique micrologiciel implanté à même le matériel.

Qu'est-ce qu'un système d'exploitation ? Pour l'essentiel un ensemble de programmes permettant l'exécution en parallèle de différentes applications. La transition entre micrologiciel « normal » et véritable système d'exploitation est continue.

La base de l'édifice est une couche matérielle bien définie sur laquelle sont progressivement amenées des fonctions utiles (par exemple à la gestion de la mémoire). Même un petit ordonnanceur qui alloue de façon cyclique des temps de calcul à différents programmes s'apparente déjà à un système d'exploitation.

### Système d'exploitation type

Les livres qui abordent ce thème sont évidemment légion [2]. Nous pouvons nous contenter ici d'un survol des principaux composants d'un système d'exploitation :

- Gestion des périphériques
- Gestion des processus
- Gestion de la mémoire
- Gestion des fichiers
- Gestion des utilisateurs et de leurs droits

La gestion des périphériques est cruciale dans le monde de l'embarqué. Le système d'exploitation doit représenter le matériel et les périphériques de façon à ce qu'une application puisse y accéder simplement. Cet accès se fait à l'aide de pilotes. Le système d'exploitation gère également les cas d'accès simultanés à une carte SD, une interface réseau, etc.

Du point de vue d'un système d'exploitation, une application (tant que l'on n'opte pas pour un modèle de programmation plus élaboré) est un processus unique, autonome et exécutable. Un processus possède son propre espace d'adressage virtuel, où il ne peut interférer avec aucun autre processus. Les processus sont gérés de manière centralisée dans la table des processus, où l'état de chacun d'entre eux est représenté par un bloc de contrôle de processus [3]. Un tel bloc contient les zones de mémoire types dans lesquelles sont tamponnées les valeurs des registres ainsi que d'autres variables :

- Compteur ordinal
  - Registre du CPU
  - Pointeur de pile
  - État des fichiers ouverts
  - État du processus
- etc.

Le noyau crée un nouveau processus dès qu'une application est chargée. Un nouveau processus dérive toujours du processus en cours et hérite de ses droits d'utilisateurs. La commande :

```
ps -ejH
```

affiche l'arborescence des processus en cours. Le premier (identifié

par le PID 1) à être appelé par le noyau est toujours `/sbin/init`. Ce processus est du reste également le premier à être appelé par le noyau lors du démarrage du système.

Le développeur a en particulier besoin de connaître :

- les interfaces pour les pilotes de périphériques
- le système de fichiers pour les programmes et les fichiers
- la façon dont sont lancés et exécutés les programmes
- la façon dont sont partagées les bibliothèques entre les programmes
- l'accès aux interfaces d'entrée et de sortie
- le transfert de la mémoire de travail en mémoire non-volatile
- la gestion de la mémoire pour les application ou les processus
- la gestion des processus

Commençons par la gestion des périphériques.

## Configuration du noyau

Le lancement d'une application nécessite différentes interfaces ainsi que plusieurs blocs de processus propres au processeur utilisé. Grâce à la conception modulaire du noyau Linux, il est possible de n'inclure dans le noyau que certains pilotes et mécanismes, ce qui est laissé de côté pouvant toujours être chargé dynamiquement plus tard. Essayons de mieux comprendre ce mode de fonctionnement en nous rendant dans le répertoire des sources du noyau :

```
cd ElektorLinuxBoardDownload_20120509
```

```
cd linux-2.6.33-lpc3131x
```

Appelons l'outil de configuration du noyau :

```
make menuconfig
```

La **figure 1** montre son menu principal. Si rien n'apparaît, c'est sans doute qu'il vous manque la bibliothèque nécessaire à l'affichage des menus sous console. Lancez la commande suivante pour l'installer :

```
sudo apt-get install libncurses5-dev
```

Ce menu permet d'activer ou désactiver graphiquement n'importe quel pilote de périphérique ou caractéristique. Il n'est pas évident de prime abord de s'y retrouver, mais avec un peu de pratique la navigation devient de moins en moins tâtonnante. Nous pourrions aussi modifier la configuration du noyau en éditant directement le fichier `.config` (le point signifie qu'il s'agit d'un fichier caché) du répertoire des sources (`linux-2.6.33-lpc3131x`), mais l'outil `menuconfig` reste le moyen le plus pratique. Commençons par créer une copie de sécurité de la configuration actuelle :

```
cp .config config_backup
```

Les débutants se demandent souvent où trouver de la bonne docu-

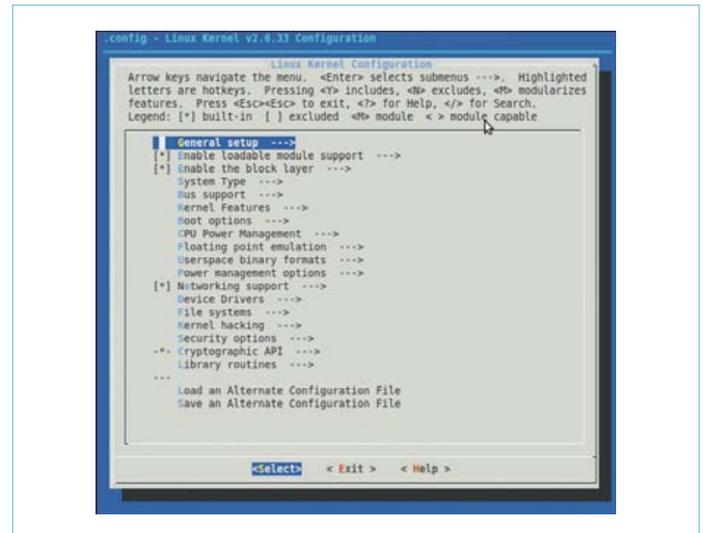


Figure 1. Menu principal de l'outil de configuration du noyau.

mentation. Le meilleur endroit pour se documenter sur le noyau est le noyau lui-même. Pour obtenir de l'aide depuis `menuconfig`, sélectionnez une entrée du menu et validez le bouton `Help`, ou appuyez directement sur la touche « ? ». Vous pouvez également fouiller dans les sous-dossiers thématiques du dossier `Documentation` du répertoire des sources.

Les différentes options de configuration sont regroupées par thème (**fig. 1**). La première concerne la configuration générale du noyau. Vous trouverez ensuite sans peine les menus relatifs aux pilotes, aux différents systèmes de fichiers, ou encore au cryptage et à la sécurité.

### General Setup

Le menu *General Setup* concerne les caractéristiques de base du noyau, p. ex. l'activation du mécanisme de pagination de la mémoire virtuelle (« swap »), ou encore le mode d'implantation des communications internes. On y trouve aussi les caractéristiques de l'image du noyau, des options d'optimisation pour les petits systèmes, la possibilité d'activer certaines statistiques, des outils, et bien d'autres options utiles, que l'on découvrira au fur et à mesure de son exploration du noyau Linux. Cela dit, rares sont ceux qui les connaissent et les comprennent toutes !

### Enable loadable module support

Ce qu'il y a de bien avec Linux, c'est qu'il est très simple de charger des modules en cours d'exécution. Une option marquée d'un astérisque signifie une option compilée dans le noyau, une option marquée par la lettre M signifie une option compilée en tant que module exécutable. On ne sera par exemple pas obligé de grossir inutilement le noyau d'une foultitude de pilotes si l'on souhaite utiliser un grand nombre de périphériques : on les compilera en tant que modules et on les chargera au besoin, de façon manuelle ou automatique.

## Nouvelle version de la carte Linux-Elektor

La vente de la carte Linux-Elektor s'est transformée en véritable ruée sur la banque ! Victimes de notre succès, nous avons épuisé le stock plus vite que prévu. Le délai de livraison de la mémoire DRAM en boîtier BGA étant malheureusement d'une vingtaine de semaines, nous avons décidé d'adopter un modèle TSOP54 à la place du BGA. La taille de cette nouvelle mémoire vive est identique à la précédente, mais nécessite 3,3 V au lieu de 1,8 V.

La photo du haut montre la carte originale et sa mémoire vive en boîtier BGA, celle du milieu montre la version avec le boîtier TSOP54. La version 1.1 de la carte, livrée depuis juillet dernier, n'est donc plus équipée du régulateur de tension de 1,8 V.

La version la plus récente de la carte est facilement reconnaissable à la couleur bleue de sa platine. Son schéma est reproduit ici.

### Enable the block layer

Le noyau distingue deux types de périphériques : le type bloc, avec lequel les échanges se font par ensemble de données appelés blocs, et le type caractère, qui communique octet par octet. Les disques durs et les cartes mémoire MMC ou SD sont des exemples de type bloc. Les consoles et les interfaces UART sont typiques des périphériques de type caractère. Désactiver l'option *block layer* économisera un peu d'espace mémoire si l'on n'a pas d'unité de type bloc (ce qui sera probablement le cas pour un noyau destiné à un système embarqué).

### System Type

Certaines caractéristiques du noyau dépendent directement du type de processeur. C'est ici que l'on activera ou désactivera p. ex. des unités de gestion de la mémoire ou des caches spécifiques, et que l'on pourra spécifier le type de son processeur. S'il n'est pas présent, il faudra chercher (ou écrire soi-même !) une rustine (*patch*) ou un paquet prenant en charge sa carte.

### Bus support

On activera ici les bus PC. Dans notre cas, seule l'option PCMCIA est présente.

### Kernel Features

On pourra déterminer ici l'interface binaire entre le système et les programmes, de même que des options telles que l'affichage du taux de charge du processeur à l'aide de la fréquence de clignotement d'une LED.

### Boot options

Dans le cas de la carte Linux-Elektor, la façon dont est démarré le noyau dépend essentiellement du chargeur de démarrage. L'interface primaire entre chargeur et noyau est une ligne de commande qui permet de passer certains paramètres au noyau. Ces paramètres sont évalués ultérieurement par le noyau ou d'autres programmes. Il est possible de spécifier ici une autre ligne de commande. On peut également compiler le noyau de façon à ce qu'il puisse être lancé directement depuis une mémoire flash.

### CPU Power Management

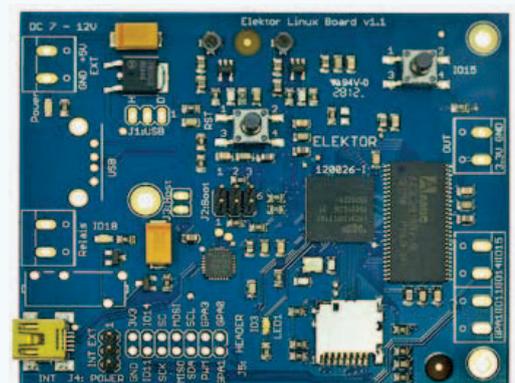
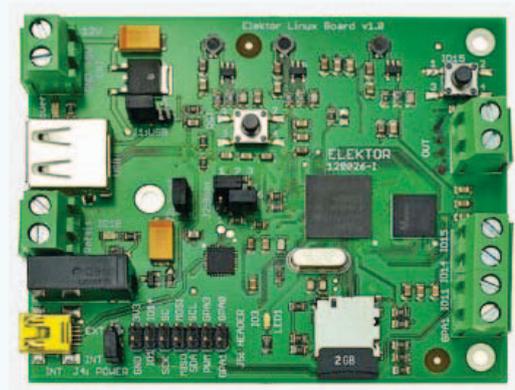
Les processeurs modernes prennent de mieux en mieux en charge la gestion de l'alimentation. Ces options doivent être implantées dans le noyau afin de pouvoir être exploitées par les applications.

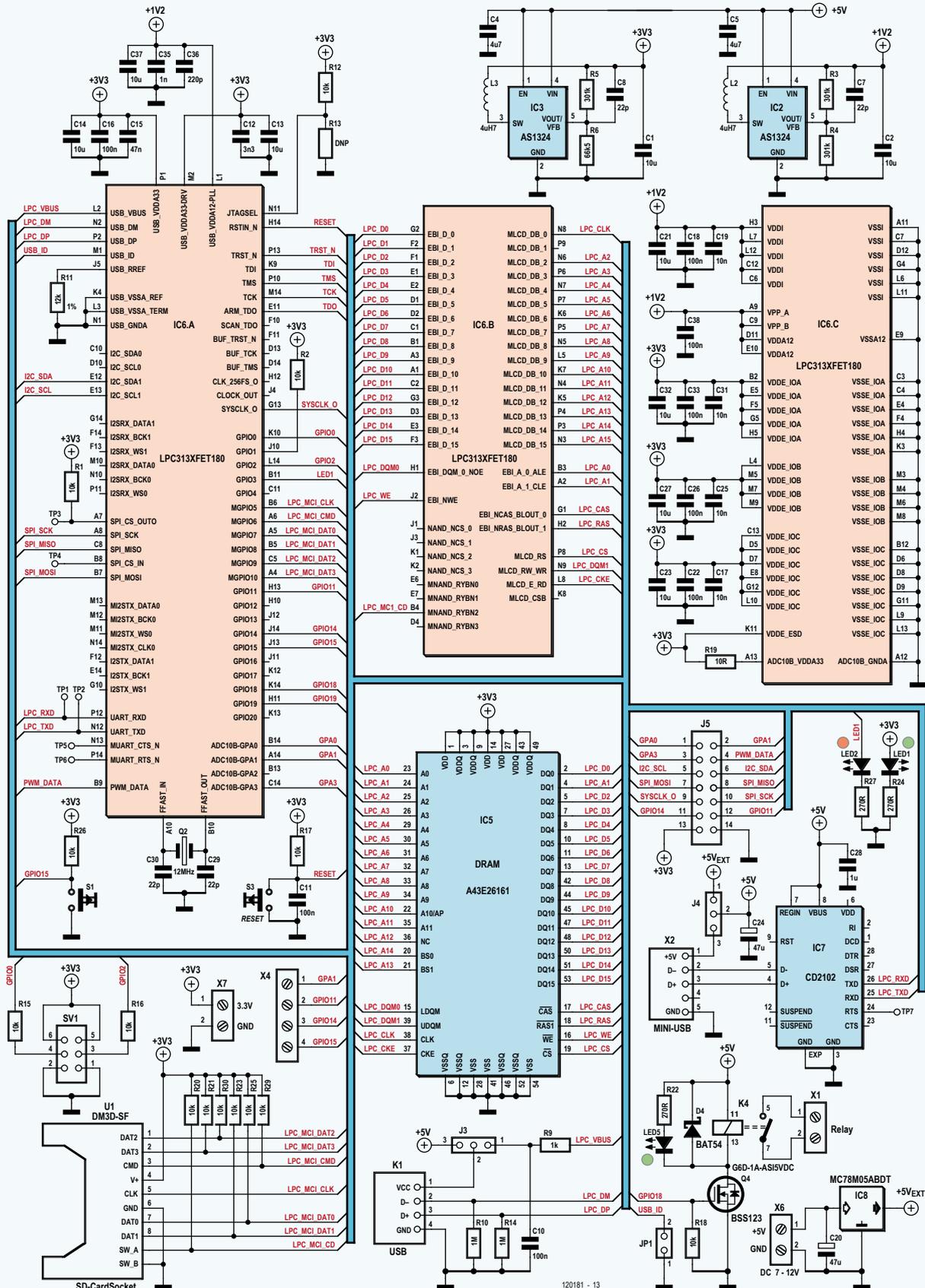
### Floating point emulation

On décidera ici si le noyau est lui-même capable d'effectuer des calculs en virgule flottante, ou s'il peut recourir à une unité matérielle. Au moins une des deux possibilités doit être choisie pour permettre le lancement des programmes.

### Userspace binary formats

Les programmes et les applications sont compilés avec la chaîne d'outils (cf. les précédents articles). Sous Linux la chaîne produit en général un fichier *.elf* (*Executable and Linkable Format*, l'équivalent





120181 - 13

du .exe de Windows) ou un fichier *a.out*. Le noyau est capable de lire ces formats, donc de lancer ces exécutables.

Astuce : sous Linux, la commande `file` affiche le type du fichier qui lui est passé en argument, et dans le cas d'un exécutable précise s'il s'agit d'un exécutable statique ou dynamique. Appliquée au fichier `vmlinux` du dossier *Linux* dans lequel on se rendra avec `cd`, la commande :

```
file vmlinux
```

affiche :

```
vmlinux: ELF 32-bit LSB executable, ARM, version 1 (SYSV), statically linked, not stripped
```

Il s'agit donc d'un exécutable statique pour processeurs ARM.

### Power management options

Le processeur n'est pas le seul à fournir une gestion de l'alimentation, certains périphériques comme la mémoire de travail peuvent eux aussi être mis en veille. Le système d'exploitation place au préalable les données importantes en mémoire vive, puis les récupère au réveil afin de restaurer son état antérieur. On réglera ici le comportement souhaité.

### Networking support

Nous arrivons aux menus qui intéressent plus particulièrement les développeurs que nous sommes. Celui-ci permet de gérer l'ensemble des piles logicielles (ici les bibliothèques de protocoles) pour les interfaces de communication. Ethernet, TCP/IP, CAN, infrarouge, radio-amateur, Bluetooth, sans-fil, etc., toutes ces options peuvent être activées, désactivées ou configurées ici.

### Device Drivers

Les pilotes de périphériques sont bien sûr d'un grand intérêt pour nous. Ce menu n'en est pas avare : ils vont du simple pilote pour périphérique d'E/S au pilote écrit pour le plus actuel des bus à haut débit pour serveurs. Côté périphériques USB, I<sup>2</sup>C et SPI, on en trouvera un grand nombre dans les sources du noyau (à télécharger depuis [4]).

### File systems

Le rôle du système de fichiers est primordial dans deux situations. Dans la première le système d'exploitation doit être placé dans un système de fichiers (sur la carte SD) de façon à ce que le noyau puisse lire les données qui lui sont vitales.

Dans la seconde, le système d'exploitation doit fournir aux applications de l'espace disque pour les fichiers et les répertoires.

### Kernel hacking

Les options de ce menu servent avant tout au développement du noyau.

### Security options

Linux fournit différents mécanismes de sécurité. Le plus connu est le système par droits d'accès, un unique utilisateur étant défini comme administrateur système (superutilisateur ou *root*), les autres étant répartis en simples utilisateurs ou groupes d'utilisateurs. Un degré de protection supplémentaire peut être apporté grâce aux modules TPM (*Trusted Platform Module*), ou encore aux modèles de sécurité internes.

### Cryptographic API

Pour activer les fonctions de cryptage du noyau.

### Library routines

Ce menu contient entre autres fonctions des méthodes de calcul des sommes de contrôles.

Le dernier menu, *Save an Alternate Configuration File*, sert à enregistrer la configuration du noyau nouvellement définie sous un nom différent. Le fichier peut ensuite être sauvegardé dans le répertoire des sources du noyau.

La compilation du noyau (`make zImage`) et sa copie sur la carte SD ont été décrites dans l'article précédent [4]. L'utilisation de nouveaux modules dans le noyau requiert leur compilation et leur installation. La compilation se fait par appel à la commande `make` :

```
make modules
```

La copie des modules sur la carte SD peut se faire de deux façons : soit en insérant dans la commande ci-dessous le chemin d'accès à la carte SD, soit en installant les modules dans le répertoire `/tmp` et en copiant manuellement sur la carte, en tant que *root* ou avec `sudo`, le nouveau dossier créé. Les privilèges d'administrateur ne sont pas vraiment indispensables pour la copie, mais en écrivant ainsi les données sur le système de fichiers de la carte SD, le noyau pourra les utiliser plus tard en tant que *root*.

```
make INSTALL_MOD_PATH=/tmp modules_install
```

On se place ensuite dans le répertoire `/tmp` pour copier le dossier *lib* sur la carte SD :

```
cd /tmp
sudo cp -R lib /
media/86b3be7-00f3-45e0-832e-1f48c2c3065e
```

### Interfaces

Linux est né dans le monde des PC, où les principales interfaces sont IDE, SATA, PCI, AGP, etc. Le monde de l'embarqué recourt quant à lui aux habituelles interfaces pour microcontrôleurs, et bien sûr à l'USB. Voici en résumé celles présentes sur notre carte :

- USB (hôte et périphérique)
- UART (comme console système, mais peut aussi servir

Tableau 1. Quelques fichiers de périphériques types du répertoire /dev/

Périphérique	Fichiers dans /dev/	Mode d'accès
Disque dur IDE	/dev/hda, /dev/hdb, ...	bloc
Disque dur SCSI	/dev/sda, /dev/sdb, ...	bloc
Interface RS-232	/dev/ttyS0, /dev/ttyS1, ...	caractère
Imprimante	/dev/lp0, /dev/lp1	caractère
Webcam	/dev/video0, /dev/video1	bloc
Souris	/dev/input/mice	caractère
RAM	/dev/ram	bloc

d'interface pour d'autres périphériques)

- I<sup>2</sup>C (maître)
- SPI (maître)

Le processeur fournit en outre des fonctions matérielles comme :

- GPIO
- Convertisseur A/N
- Sortie PWM

Nous aborderons les détails de ces interfaces au fur et à mesure de leur utilisation.

## Fichiers de périphérique

Le dialogue avec un périphérique depuis une application nécessite la présence d'une interface pour les pilotes. Les modes d'accès aux pilotes diffèrent selon les systèmes d'exploitation. Linux a hérité de l'architecture UNIX, et UNIX repose sur un concept simple : chaque périphérique est représenté par un fichier, et manipulable en tant que tel. L'idée est qu'aucun outil spécial ne doit être nécessaire pour piloter un périphérique, que l'on doit pouvoir les commander facilement (depuis la console) avec de courts programmes effectuant les habituelles opérations de lecture et d'écriture sur les fichiers. Nous avons illustré ce mécanisme dans le deuxième article en montrant comment accéder aux broches du port GPIO pour faire clignoter la LED rouge de la carte, le tout depuis la ligne de commande [5].

Deux commandes simples permettent déjà d'accomplir beaucoup de choses : `cat` et `echo`. La commande `cat` affiche le contenu d'un fichier, `echo` permet d'écrire dans un fichier en redirigeant sa sortie dans ledit fichier.

La commande `echo` permet ainsi d'écrire rapidement des données sur un disque dur (directement, pas via le système de fichiers) :

```
echo 'Données sur le disque dur' > /dev/sda
```

Afficher « Salut les gars » sur l'interface série n'est pas plus compliqué :

```
echo 'Salut les gars' > /dev/ttyS0
```

Attention : si vous souhaitez tester l'écriture de données directement sur un disque dur ou une carte mémoire, utilisez impérativement un support de stockage inutilisé !

Les fichiers de périphérique se distinguent par leurs types : bloc ou caractère. La communication avec un disque dur se fait par exemple toujours en mode bloc, à l'inverse d'une interface RS-232 comme

un terminal classique, qui elle ne peut recevoir et traiter que des caractères. Le **tableau 1** liste quelques fichiers de périphérique typiques d'un système Linux ou Unix, et précise leur mode d'accès.

Vous vous demandez sans doute quel est le mécanisme à l'oeuvre derrière `echo` et `cat` – comment un pilote affiche ou capture des données. Rappelez-vous que sous Linux tout est fichier, donc qu'il est possible d'accéder à leur contenu en lecture ou en écriture. Ce sont précisément ces fonctions que doit offrir un pilote : `Read`, `Write`, à laquelle s'ajoute `Ioctl`. `Read` et `Write` sont les classiques fonctions de lecture et d'écriture de fichiers, `Ioctl` est un appel système qui permet d'effectuer des opérations sur un périphérique. Fichiers et périphériques sont manipulés de la même façon : ils sont ouverts et un descripteur de fichier (un entier qui les identifie) leur est attribué. Les opérations se font alors sur ces descripteurs. Un descripteur sera par exemple passé à la fonction `Read` pour une lecture.

## Noyau et espace utilisateur

Les systèmes d'exploitation classiques – dont fait partie Linux – sont la plupart du temps organisés de façon à ce que les zones de mémoire pour le noyau et les applications soient séparées. Le développeur qui travaille très près du système devra donc se souvenir qu'une application n'a aucun moyen d'accéder à la zone de mémoire du noyau. Linux comprend un espace noyau et un espace utilisateur, et un pilote de périphérique est en principe toujours un programme exécuté dans l'espace noyau. Comment dans ce cas récupérer les données d'une interface ? Grâce à plusieurs fonctions du noyau, dont `copy_to_user` [6], qui copie des blocs de données de l'espace noyau à l'espace utilisateur. Un pilote appellera ainsi cette fonction en arrière-plan pour accéder en lecture à un fichier de périphérique.

## Des pilotes bien fichés

Nous savons maintenant comment un pilote travaille dans le noyau, mais, pour reprendre l'exemple d'un pilote pour l'interface série, d'où le noyau tient-il qu'avec `/dev/ttyS0` il doit s'adresser à la console ?

Aux périphériques sont associés deux numéros, dits majeur et mineur, identifiant respectivement le type du périphérique et le périphérique lui-même. Cette correspondance unique est décrite dans le fichier `devices.txt` présent dans les sources du noyau. Le numéro majeur indique à quel pilote correspond le périphérique associé. Pour voir ceux qui correspondent aux pilotes actuellement chargés en mémoire, nous pouvons lancer :

```
cat /proc/devices
```

```

root@gnublin:~# cat /proc/devices
Character devices:
 1 mem
 4 /dev/vc/0
 4 tty
 4 ttys
 5 /dev/tty
 5 /dev/console
 5 /dev/ptmx
 7 vcs
10 misc
13 input
21 sg
89 i2c
128 ptm
136 pts
153 spi
180 usb
189 usb_device
252 usbmon
253 lpc313x_pwm
254 lpc313x_adc

Block devices:
 1 ramdisk
259 blkext
 7 loop
 8 sd
65 sd
66 sd
67 sd
68 sd
69 sd
70 sd
71 sd
128 sd
129 sd
130 sd
131 sd
132 sd
133 sd
134 sd
135 sd
179 mmc
    
```

Figure 2. Liste des pilotes chargés.

La sortie affiche les numéros majeurs, regroupés par mode bloc et caractère (fig. 2).

Si nous nous plaçons dans le répertoire /dev et listons son contenu avec la commande `ls -l` (fig. 3), nous voyons qu'il contient de nombreux fichiers spéciaux représentant les périphériques du sys-

```

root@gnublin:~# cd /dev
root@gnublin:/dev# ls -l
total 1
crw----- 1 root root  5, 1 Sep 27 06:02 console
prw----- 1 root root  0 Sep 27 06:02 initctl
srw-rw-rw- 1 root root  0 Sep 27 06:02 log
crw-r--r-- 1 root root 254, 0 Sep 26 03:57 lpc313x_adc
crw-r--r-- 1 root root 253, 0 Sep 26 04:21 lpc313x_pwm
lrwxrwxrwx 1 root root 12 Jan 1 1970 mtab -> /proc/mounts
crw-r--r-- 1 root root  1, 3 Dec 22 2011 null
crw-r--r-- 1 root root 89, 1 Sep 26 03:06 pca9555
crw-rw-rw- 1 root root  5, 2 Jan 1 1970 ptmx
drwxr-xr-x 2 root root  0 Jan 1 1970 pts
crw-r--r-- 1 root root  1, 9 Sep 26 04:28 random
brw-r--r-- 1 root root  8, 0 Sep 27 05:52 sda
brw-r--r-- 1 root root  8, 1 Sep 27 05:52 sda1
crw-rw-rw- 1 root root  5, 0 Jan 1 1970 tty
-rw-r--r-- 1 root root 24 Jan 1 1970 tty0
crw----- 1 root root  4, 64 Sep 27 06:03 ttyS0
crw-r--r-- 1 root root  1, 9 Sep 26 04:30 urandom
    
```

Figure 3. Ces fichiers représentent nos périphériques.

tème. Le premier caractère de la ligne indique le mode d'accès du périphérique : **c** pour caractère, **b** pour bloc. Les caractères suivants sont les droits d'accès (dont nous reparlerons plus tard). Viennent ensuite les noms de l'utilisateur et du groupe auquel appartient le fichier puis, séparés par une virgule, les numéros majeur et mineur. Les numéros majeurs sont 1,4,5,8,89,253 et 254 ; les mineurs 0,1,2,3,9,12,24 et 64.

Si l'on accède à un fichier de périphérique depuis l'espace utilisateur, le système saura quel pilote utiliser en consultant la correspondance entre numéro majeur et pilote telle que définie dans le fichier /proc/devices.

Nous nous servirons plus tard de l'outil `mknod` [7] pour créer de tels fichiers de périphérique.

### Piloter un convertisseur USB/série

Il est temps de passer à la pratique ! Voyons comment piloter (par USB) un simple convertisseur USB/série depuis la carte Linux-Elektor (fig. 4). Nous allons successivement :

- Activer le pilote dans le noyau
- Compiler le noyau
- Copier le noyau sur la carte SD
- Vérifier si le pilote est activé
- Créer un fichier de périphérique
- Tester le périphérique

#### Activation des pilotes dans le noyau

Retournons dans le répertoire des sources du noyau pour lancer l'outil de configuration du noyau :

```
cd ElektorLinuxBoardDownload_20120509/
```

```
cd linux-2.6.33-lpc313x/
```

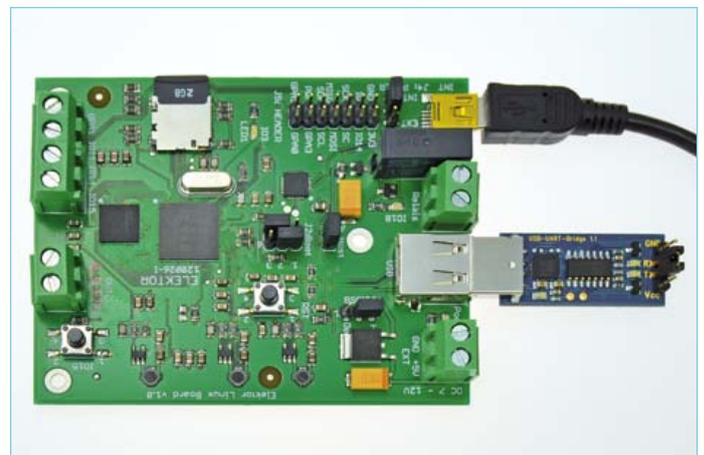


Figure 4. Le convertisseur USB/Série CP2102 sur notre carte Linux-Elektor.

```
./set.h
```

```
make menuconfig
```

Sélectionnez d'abord l'entrée *Device Drivers* (fig. 5). Ce menu contient un sous-menu *USB support* (fig. 6). La capture d'écran le montre marqué d'un astérisque, mais la configuration par défaut est un M. La barre d'espace permet de sélectionner trois options : compilé en tant que module (M), compilé dans le noyau (\*), ou exclu (choix laissé vide). Choisissez la compilation en dur (\*) pour ne pas avoir à manipuler les modules. Dans le menu *USB Serial Converter support*, sélectionnez également \* (fig. 7) à la place du M présent par défaut (fig. 8). Appuyez sur la touche d'entrée pour entrer dans le sous-menu et choisir les pilotes (fig. 9). Comme il est probable que vous soyez nombreux à utiliser une puce de chez FTDI ou un microcontrôleur CP210x de chez Silabs, nous avons opté ici pour une compilation en dur (\*) de ces deux pilotes. Activez enfin *Exit* jusqu'à ce que l'outil vous demande de confirmer la sauvegarde de la nouvelle configuration.

### Compilation du noyau

Nous voici prêts à compiler le noyau :

```
make zImage
```

La figure 10 reproduit un extrait de la sortie de cette commande.

### Copie du noyau sur la carte SD

Insérez la carte SD de la platine dans un lecteur de cartes SD, et reliez le lecteur au PC. La commande suivante copie le noyau sur la carte SD :

```
sudo cp arch/arm/boot/zImage /  
media/386b3be7-00f3-45e0-832e-1f48c2c3065e/
```

Comme nous l'avons expliqué dans l'article précédent [4], il est préférable de démonter manuellement le système de fichiers après la copie. De cette façon, nous serons certains que tous les blocs de données ont bien été transférés sur la carte :

```
umount /media/386b3be7-00f3-45e0-832e-1f48c2c3065e
```

### Vérification de la présence des pilotes

Avant de démarrer la carte Linux-Elektor, veillez à ce que le cavalier JP1 soit bien inséré ! C'est lui qui au démarrage force le mode hôte de l'interface USB/série. Le périphérique relié au connecteur USB de la carte Linux-Elektor doit en outre être alimenté en courant, et pour cela le cavalier 3 doit être en position 3-2.

La présence des deux nouveaux pilotes est visible durant l'amorçage :

```
cp210x: v0.09:Silicon Labs CP210x RS232 serial  
adaptor driver
```

```
elektor 10-2012
```

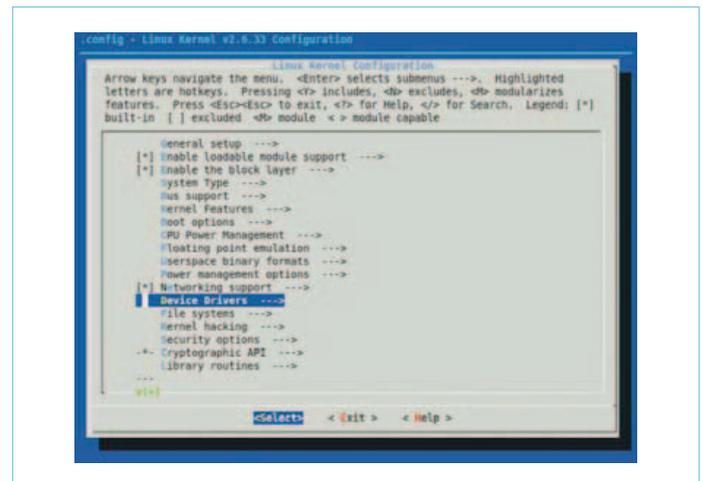


Figure 5. Menu *Device Drivers*.

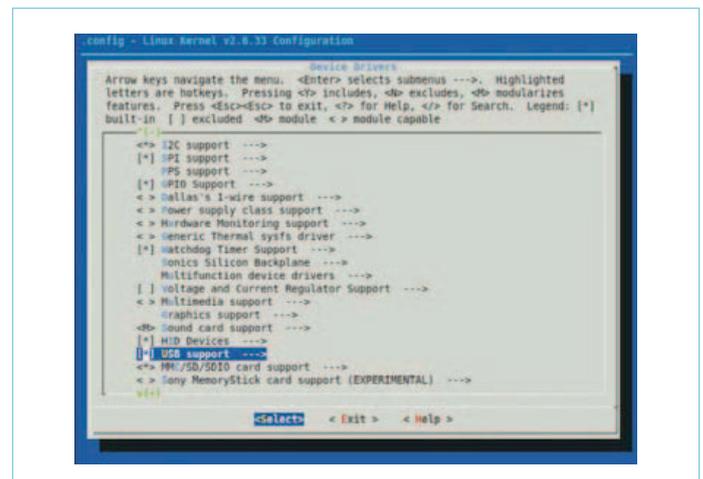


Figure 6. Menu *USB support*.

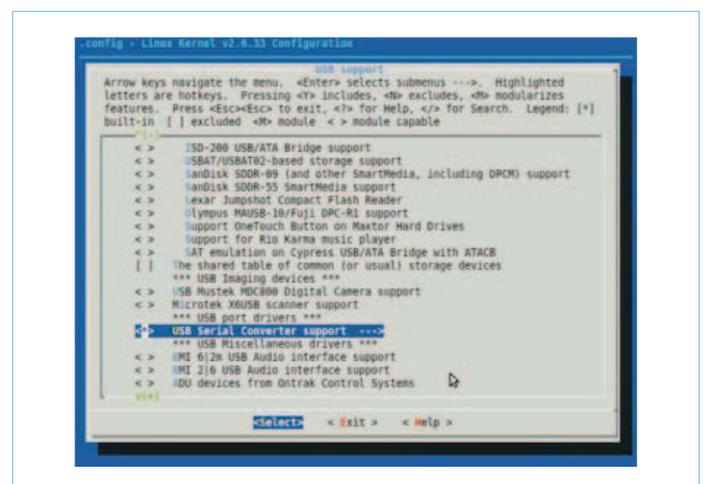


Figure 7. *USB Serial Converter support* inclus dans le noyau (\*).

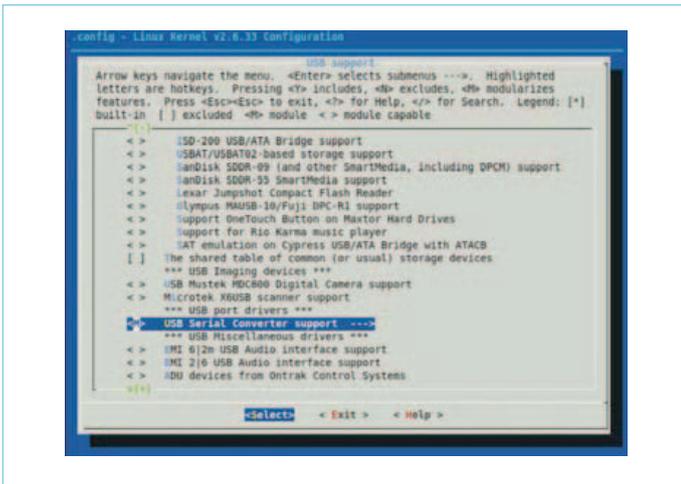


Figure 8. USB Serial Converter support en tant que module chargeable (M).

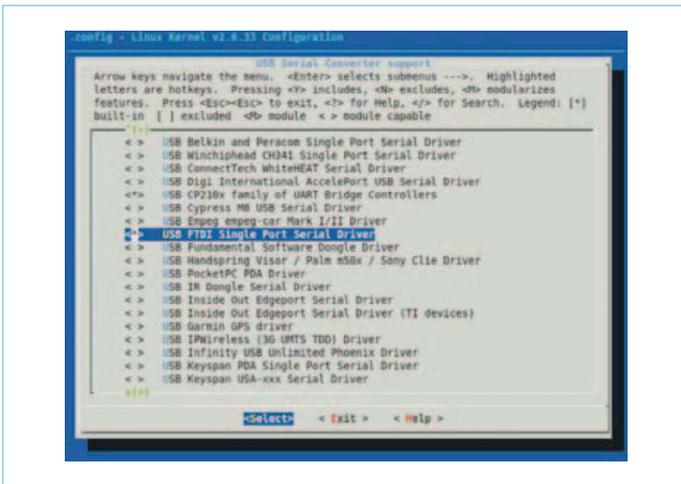


Figure 9. Menu de sélection de l'adaptateur USB/série utilisé.

USB Serial support registered for FTDI USB Serial Device

### Création du fichier de périphérique

La sortie de la commande `cat /proc/devices` montre la présence d'une nouvelle entrée : `188 ttyUSB`. Créons un fichier de périphérique pour ce numéro majeur :

```
mknod /dev/ttyUSB0 c 188 0
```

### Test du périphérique

La connexion de l'adaptateur USB/série à la carte déclenche la sortie suivante :

```
usb 1-1: New USB device found, idVendor=10c4, idProduct=ea60
```

```
usb 1-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
```

```
usb 1-1: Product: CP2102 USB to UART Bridge Controller
```

```
usb 1-1: Manufacturer: Silicon Labs
```

```
usb 1-1: SerialNumber: 0001
```

```
cp210x 1-1:1.0: cp210x converter detected
```

```
usb 1-1: reset full speed USB device using lpc-ehci and address 2
```

```
usb 1-1: cp210x converter now attached to ttyUSB0
```

La dernière ligne prouve que la création du fichier de périphérique a réussi. Le noyau affecte le bon fichier de périphérique au périphérique nouvellement reconnu.

Pour tester le pilote, nous pouvons nous servir de *microcom*, un petit émulateur de terminal contenu dans notre système d'exploitation Linux. Établisons un contact électrique (cavalier) entre les broches RX et TX de l'adaptateur USB/série, et appelons *microcom* avec les paramètres suivants :

```
microcom -s 9600 /dev/ttyUSB0
```

Entrons quelques lettres : elles sont envoyées à l'adaptateur, et de là retournées à la carte. Elles devraient s'afficher si le pilote est actif. La sortie restera par contre muette si l'on rompt la liaison entre RX et TX de l'adaptateur USB/série, preuve définitive que le pilote de périphérique fonctionne. Quittez *microcom* avec Ctrl-x.

Vous pouvez utiliser de la même façon des adaptateurs LAN/USB et WLAN/USB, des cartes son USB, et bien d'autres périphériques USB.

## Weblinks

[1] [sauter@embedded-projects.net](mailto:sauter@embedded-projects.net)

[2] [http://en.wikipedia.org/wiki/Operating\\_Systems:\\_Design\\_and\\_Implementation](http://en.wikipedia.org/wiki/Operating_Systems:_Design_and_Implementation)

[3] <http://de.wikipedia.org/wiki/Prozessstabelle>

[4] [www.elektor.de/120180](http://www.elektor.de/120180)

[5] [www.elektor.de/120146](http://www.elektor.de/120146)

[6] [www.gnugeneration.com/mirrors/kernel-api/r4299.html](http://www.gnugeneration.com/mirrors/kernel-api/r4299.html)

[7] <http://linuxwiki.de/mknod>

```

CC      arch/arm/boot/compressed/misc.o
In file included from arch/arm/boot/compressed/../../../../lib/decompress_inflate.c:10:0,
      from arch/arm/boot/compressed/misc.c:248:
arch/arm/boot/compressed/../../../../lib/zlib_inflate/inflate.c: In function 'zlib fixedtables':
arch/arm/boot/compressed/../../../../lib/zlib_inflate/inflate.c:100:1: warning: the frame size of 2176 bytes is larger than 1024 bytes
SHIPPED arch/arm/boot/compressed/lib1funcs.S
AS      arch/arm/boot/compressed/lib1funcs.o
LD      arch/arm/boot/compressed/vmlinux
OBJCOPY arch/arm/boot/zImage
Kernel: arch/arm/boot/zImage is ready

```

Figure 10. Sortie de la console lors de la compilation du noyau.

Dans le prochain article nous partirons à la découverte d'autres périphériques et d'autres interfaces, et bien sûr nous jouerons avec !

(120181 – version française : Hervé Moreau)

## Liens

- [1] [sauter@embedded-projects.net](mailto:sauter@embedded-projects.net)
- [2] [http://en.wikipedia.org/wiki/Operating\\_Systems:\\_Design\\_and\\_Implementation](http://en.wikipedia.org/wiki/Operating_Systems:_Design_and_Implementation)
- [3] [http://fr.wikipedia.org/wiki/Process\\_Control\\_Block](http://fr.wikipedia.org/wiki/Process_Control_Block)
- [4] [www.elektor.fr/120180](http://www.elektor.fr/120180)
- [5] [www.elektor.fr/120146](http://www.elektor.fr/120146)
- [6] [www.gnugeneration.com/mirrors/kernel-api/r4299.html](http://www.gnugeneration.com/mirrors/kernel-api/r4299.html)
- [7] <http://manpages.ubuntu.com/manpages/lucid/fr/man1/mknod.1.html>

Publicité

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

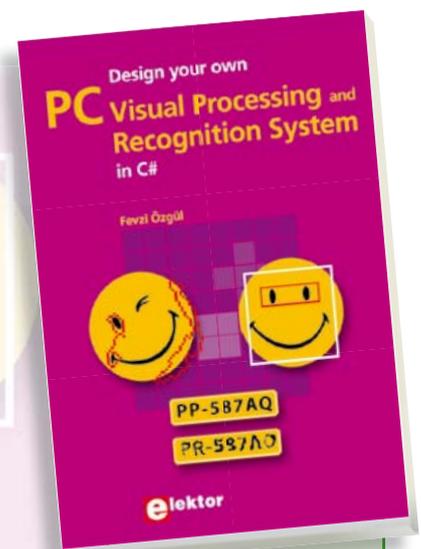
### ➔ Nouveau livre d'Elektor en anglais

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)

# AndroCAR

## véhicule piloté par l'inclinaison de votre téléphone

Benoit Pierret, Nicolas Stouls et Yann Ricotti (INSA Lyon)

Pour piloter ce prototype, il suffit d'incliner votre téléphone Android. Celui-ci contient des accéléromètres et un module WiFi qu'il est très facile d'exploiter. Ce montage peut être réalisé très simplement et rapidement grâce à l'E-Block WiFi EBo6g. À partir de cet article, vous pourrez télécommander avec votre téléphone tactile et, selon vos besoins, un véhicule de la taille d'une boîte d'allumettes ou faire du transport humain !

Dans notre article publié dans le numéro de septembre 2012 d'Elektor [1], nous avons vu comment piloter facilement une LED à distance grâce à un terminal Android. Cette démonstration simple ouvre le champ à de nombreuses possibilités. En voici une qui ne manque pas d'allure. Et de simplicité ! Ce qui nous laisse le loisir de nous intéresser de près à la communication WiFi : notre réseau envoie-t-il correctement les données (côté *smartphone*) et les interprète-t-il correctement (côté électronique embarquée) ?

### Fabrication de la mécanique

La partie mécanique, inexistante le mois dernier sur l'AndroLED, est cette fois assez développée. La solution présentée pour la fabrication a le mérite de la rapidité. À vous de l'adapter selon vos envies et vos besoins. Vous aurez le choix entre un micro-véhicule de 5 cm, un robot de combat ou un modèle capable de transporter un humain.

**Le châssis.** Pour un montage très simple des E-Blocks, il vous faut une plaque avec des perçages espacés de 20 mm, comme le support d'E-Blocks BP232. Une version plus économique pourra être faite de contreplaqué de 5 mm percé au bon entraxe.

**Les moteurs.** Comme nous utilisons des moteurs à courant continu, ceux-ci devront être compatibles en courant et en tension avec la carte de pilotage des moteurs. Pour préparer cet article, nous avons utilisé des moteurs EMG30, mais leur équerre de fixation standard n'est pas au pas de 20 mm. Il faut soit les repercer soit n'utiliser qu'une seule vis. Si vous utilisez un module Sabertooth 2 x 25 A, le moteur et l'alimentation devront avoir une tension comprise entre 6 V et 24 V.

Avec la carte *Sabertooth* 2 x 25 A, nous avons testé les modèles de moteurs suivants :

- EMG30 – 4,2 W à 12 V – codeur incrémental
- MFA 944D1041 – 1,25 W à 12 V
- MY1016Z3 – 350 W à 24 V

**La roulette.** La solution proposée consiste à utiliser une roulette avec un seul trou de fixation (fig. 1). Vous pourrez simplifier à l'ex-



trême en faisant frotter une vis au sol, mais c'est moins élégant et cela ne conviendra pas pour les surfaces rugueuses.

### Branchements électroniques

La solution présentée est très simple : le module WiFi EB069 et une carte de puissance moteur *Sabertooth*. Sur la photo, c'est le module 2 x 25 A (fig. 2), mais la version 2 x 5 A suffit pour de petits moteurs. Il n'y a aucun microcontrôleur à programmer. Le secret ? Les modules WiFi et *Sabertooth* communiquent selon le protocole RS232 : il suffit de les brancher et de les configurer correctement (fig. 2). Reliez vos deux moteurs et l'alimentation à la carte moteur, côté puissance. Dans le cas de l'EMG30, 8 piles ou accu LR6 suffisent. N'essayez surtout pas d'inverser le sens de marche des moteurs en inversant l'alimentation. Vous verriez un joli feu d'artifice, et peut-être même un gros trou dans le circuit imprimé comme ce fut le cas quand nos étudiants s'y sont risqués.

Côté commande, il faut connecter le signal RS232 : RX de l'EB069 sur S1 du *Sabertooth*. Pour cela, si vous utilisez des fils de câblage mono-brin, vous pouvez les enficher directement dans la borne RX du support tulipe du EB069 (fil jaune sur la figure 2). Le cavalier de configuration doit alors être placé sur PATCH. Enfin, il faut mettre TRIG à 5 V pour que les données WiFi soient renvoyées sur le RS232 (fil rouge), et ne pas oublier les masses.

Reste l'alimentation : pas de problème si vous utilisez une carte *Sabertooth* 2 x 25 A V2 et de petits moteurs EMG30, elle peut fournir jusque 1 A sous 5 V. Il suffit dans ce cas de connecter la sortie 5 V du *Sabertooth* aux entrées +14 V et +V du module WiFi. Si vous utilisez des moteurs plus puissants (350 W), la chute de tension au démarrage ou au freinage peut perturber l'alimentation du WiFi. Pour cela, nous vous conseillons d'utiliser une alimentation séparée pour la carte EB069, p. ex. avec 4 piles LR6. Il y a certainement

mieux : toutes les propositions qui permettraient de travailler avec une unique alimentation sont les bienvenues, si vous avez des idées, adressez-les à [redaction@elektor.fr](mailto:redaction@elektor.fr). L'intensité du courant fourni sous 5 V par le petit module *Sabertooth 2 x 5 A* est insuffisante. Prenez quatre de vos huit piles LR6 pour ça.

Si vous n'envisagez pas de passer à des moteurs de forte puissance, il est plus économique et plus simple d'utiliser uniquement des E-Blocks : tous les connecteurs sont en subD9 et sur toutes les cartes l'écart entre les trous de fixation est un multiple de 20 mm. Dans ce cas, il vous faudra la carte multi-programmateur EB006 utilisée dans l'article du mois dernier ainsi que la platine de commande de moteur EB022. Un programme Flowcode de quelques cases sera nécessaire, sur un PIC muni de deux sorties PWM (16F887 ou 16F877 p. ex.). Dans vos archives Elektor vous trouverez de quoi vous fabriquer aussi circuit de puissance adapté à vos besoins.

### Configuration des modules

Côté module WiFi EB069, vous n'avez rien à faire si votre dernier montage est l'AndroLED publié en septembre [I]. Sinon, il faudra le configurer en UDP sur l'adresse 192.168.1.254, port 5000, avec un débit de 38400 bauds pour le port RS232. Si l'adresse est toujours 192.168.1.254, tapez-la dans votre navigateur préféré pour faire les modifications, sinon, il faudra faire initialiser la carte en mettant le cavalier sur PATCH et en reliant la broche RST à 5 V pendant plus de 3 s. Si le module WiFi n'est toujours pas entré en communication avec vous, il est urgent de retrouver votre numéro de septembre enfoui sous vos montages électroniques.

Côté carte moteur, il suffit de configurer les micro-interrupteurs pour être en mode *simplified serial* à 38400 bauds (voir la notice du module *Sabertooth* [II]). Si l'anglais n'est pas votre point fort, recopiez simplement la configuration de la **figure 3**.

À ce stade du développement, votre prototype est opérationnel et exécutera toute commande reçue en WiFi. Passons donc à la réalisation d'une télécommande... sous Android tant qu'à faire.

### Le programme Android

Vous pouvez évidemment vous contenter de la solution la plus simple : télécharger directement sur votre téléphone tactile l'application toute faite [III]. Ce serait dommage. Elle est simple à réaliser, ne vous privez pas du plaisir de la modifier. Lisez donc la suite et faites votre propre version !

Veillez aussi vous reporter à l'article AndroLED, car nous ne décrivons ici que les nouveaux éléments en commençant par le plus facile :

#### Émission des commandes

Reprenez votre programme AndroLED [I] et modifiez simplement la méthode d'envoi des messages pour qu'elle prenne deux paramètres en entrée (*leftMotor* et *rightMotor*). Il suffit ensuite de placer ces deux valeurs dans le paquet UDP et de les envoyer (**listing 1**). Attention au piège : n'oubliez pas d'ajouter la permission d'accès au réseau dans votre *Manifest*, comme décrit dans l'article précédent.



Figure 1 : Une seule vis de fixation suffit pour fixer cette roulette.

### Utilisation des capteurs

Attaquons maintenant la nouveauté de cet article, qui consiste à capturer les informations d'orientation de la plateforme. Il est important de comprendre que ce n'est pas le programme qui va lire l'information lorsqu'elle est disponible. C'est la plateforme qui prévient tous les programmes intéressés lorsqu'une nouvelle information est disponible. Il faut donc *primo* répondre à l'interface d'écoute, *secundo* écrire l'action à réaliser lorsqu'une information est disponible et *tertio* enregistrer le programme auprès de la plateforme pour être averti.

Pour répondre à l'interface d'écoute, il suffit de déclarer que la classe implémente l'interface `SensorEventListener` (**listing 2**), et de

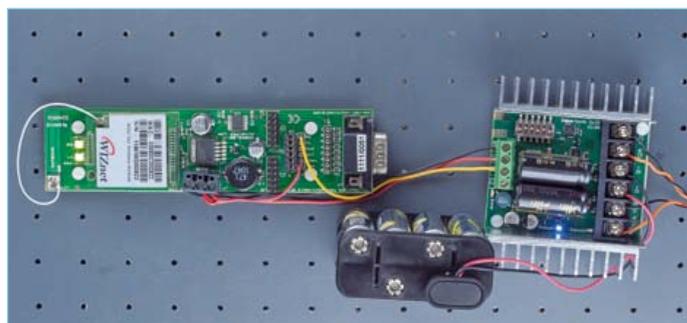


Figure 2 : Modules sous tension



Figure 3 : Configuration des micro-interrupteurs.

### Listing 1 : émission de la commande

```
private void sendCommand(int leftMotor, int rightMotor) {
    try {
        // Packet data contains both commands
        byte[] data = new byte[2];
        data[0] = (byte) leftMotor;
        data[1] = (byte) rightMotor;

        // Sending packet
        DatagramPacket packet = new DatagramPacket(data, 2, address, PORT);
        UDP socket.send(packet);
    } catch (Exception e) {
        // If error: show an error message
        Toast.makeText(this, "Sending error.", Toast.LENGTH_LONG).show();
    }
}
```

### Listing 2 : respect de l'interface d'écoute des capteurs

```
public class DroidcarActivity extends Activity implements SensorEventListener {
```

### Listing 3 : méthodes d'écoute des valeurs des capteurs

```
public void onAccuracyChanged(Sensor sensor, int accuracy) { /* Nothing to do */ }
public void onSensorChanged(SensorEvent event) {
    // Read sensors data
    int pitch = (int) event.values[SensorManager.DATA_Y];
    int roll = (int) event.values[SensorManager.DATA_Z];

    // Translation of degrees into motors command
    int leftMotor = pitch * 2 - (roll / 2) + 64;
    int rightMotor = pitch * 2 + (roll / 2) + 192;

    // Checking limits
    if (leftMotor < 1) leftMotor = 1;
    if (leftMotor > 127) leftMotor = 127;
    if (rightMotor < 129) rightMotor = 129;
    if (rightMotor > 255) rightMotor = 255;

    // Sending commands
    sendCommand(leftMotor, rightMotor);
}
```

### Listing 4 : enregistrement et désenregistrement de l'application auprès du gestionnaire de capteurs

```
/** Variable for association with Android sensors */
private SensorManager sm;

protected void onResume() {
    super.onResume();
    // Registration of this class to be called at each sensor event.
    sm = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
    sm.registerListener(this, sm.getDefaultSensor(Sensor.TYPE_ORIENTATION),
        SensorManager.SENSOR_DELAY_NORMAL);
}

protected void onPause() {
    super.onPause();
    // Remove orientation listener
    sm.unregisterListener(this, sm.getDefaultSensor(Sensor.TYPE_ORIENTATION));

    // Stop motors
    sendCommand(64, 192);
}
```

**Listing 5 : action du bouton de sécurité, à mettre dans la méthode onCreate**

```
// Management of the button
btn = (Button) findViewById(R.id.btn);
btn.setOnClickListener(new OnClickListener() {
    public boolean onTouch(View v, MotionEvent event) {
        switch (event.getAction()) {
            case MotionEvent.ACTION_DOWN:
                stop = false;
                btn.setTextColor(Color.GREEN);
                return true;
            case MotionEvent.ACTION_UP:
                stop = true;
                btn.setTextColor(Color.BLACK);
                return true;
        }
        return false;
    }
});
```

fournir les deux méthodes déclarées dans cette interface (**listing 3**) : l'une sera appelée chaque fois que la sensibilité des capteurs est modifiée (`onAccuracyChanged`), tandis que l'autre sera appelée chaque fois qu'une nouvelle valeur des capteurs est disponible (`onSensorChanged`). Dans notre cas, seule cette dernière méthode aura un corps. Son corps aura comme action de lire l'événement transmis, qui contient l'état des capteurs, de convertir ces valeurs en commandes moteur et enfin d'envoyer ces valeurs au prototype. Pour réaliser correctement cette action, il faut comprendre ce que représente chaque valeur et connaître son domaine de définition. L'état des capteurs est une valeur en degrés. Elle varie de  $-180$  à  $+180^\circ$  pour le tangage (*pitch*) (rotation avant/arrière) et de  $-90$  à  $90$  pour le roulis (*roll*) (rotation gauche/droite). Les commandes d'avance et de recul des moteurs sont codées sur 8 bits (la commande proprement dite sur 7 bits, le bit de poids fort étant la sélection du moteur). Comme chaque moteur peut tourner dans les deux sens, leur origine est fixée au milieu de leur intervalle de commande. Ainsi, pour le moteur 1, les commandes 1, 64 et 127 correspondent respectivement à « recul maximum », « stop » et « avance maximum », tandis que, pour le moteur 2, les valeurs de commande extrêmes sont 129, 192 et 255.

Notons que les deux axes de rotation de la tablette utilisée par nous correspondent à une direction et une avance, et non pas à deux commandes de moteur. Il faut donc réaliser la conversion de l'un en l'autre. Une solution relativement simple consiste à dire que la commande de moteur gauche est la somme des deux états des capteurs, tandis que la commande de moteur droite est la différence entre les deux états de capteurs. Ainsi, un virage correspond à l'accélération d'un moteur en même temps qu'au ralentissement de l'autre. Enfin, pour terminer cette partie sur la récupération des informations des capteurs, il ne manque qu'une chose : faire en sorte d'être averti lorsqu'une nouvelle valeur arrive. Pour cela, il suffit de récupérer un pointeur vers le gestionnaire de capteurs et de s'enregistrer auprès de lui en précisant le taux de rafraîchissement souhaité. Cependant attention : lorsque l'on s'enregistre sur un capteur, l'application reçoit des notifications tout le temps, même si elle est en pause (masquée par une autre application). Il est donc important de considérer l'enregistrement et le désenregistrement auprès des capteurs. Pour cela, il suffit de ne pas mettre le code dans la méthode `onCreate`, qui n'est exécutée qu'une seule et unique fois par exécution, mais dans les méthodes `onResume` et `onPause` (**listing 4**).

**Pour finir le programme...**

À ce niveau du programme : ça marche ! Vous pouvez déjà le tester et commander votre prototype. Vous constaterez qu'il est très difficile d'arrêter le prototype, car l'horizontale parfaite est impossible à atteindre et à tenir. Vous aurez également des problèmes de rotation de l'écran qui permute les axes (sur certaines plateformes). Nous allons donc d'une part ajouter un bouton qui n'autorise l'émission des commandes que tant qu'il est pressé et, d'autre part, bloquer la rotation de l'écran.

Pour ce faire, déclarez une variable globale `stop` de type booléen et ajoutez un bouton dans votre interface graphique. Le rôle de ce bouton sera de configurer l'état de `stop`. Il suffira ensuite de lire l'état de `stop` avant chaque émission pour savoir si elle doit avoir lieu ou pas. Commençons par identifier notre bouton en lui donnant le doux nom de `btn` et récupérons ensuite un pointeur vers `btn` depuis le programme Java, pour y associer une action pour l'événement `onTouch`. Cet événement permet d'être averti du début et de la fin d'une pression sur le bouton. Ainsi, il sera possible de dire « tant que le bouton est appuyé, des commandes sont envoyées ». Le **listing 5** décrit l'action de ce bouton, à déclarer dans la méthode `onCreate`. Pour finir, il ne reste plus qu'à prendre en compte le booléen `stop` dans les différentes parties de notre programme. Tout d'abord, dans la méthode `sendCommand`, qui ne doit émettre la commande demandée que si `stop` est *faux*. Dans le cas contraire, la commande à envoyer est 0 (tout arrêter). Enfin, dans la méthode `onPause`, il peut être intéressant de forcer `stop` à *vrai*, pour forcer l'arrêt du prototype lorsque le programme de commande est masqué. Et voilà ! Ainsi modifié, votre programme Android permet de commander le prototype en toute sécurité.

Il ne reste plus que le « petit » détail de la rotation de l'écran. L'application étant inutilisable sur certaines plateformes sans ce blocage (à cause du changement d'axes en cas de rotation), il vous est vivement conseillé d'ajouter l'attribut `screenOrientation` dans le *Manifest* de votre application, comme montré dans le **listing 6**. Maintenant, place aux tests !

**Test du véhicule ou du programme Android**

Au moment de la mise sous tension, c'est soit le bonheur de la réussite immédiate, soit, trop souvent, la déception : « Ça marche pas ». Si cela vous arrive, ne vous découragez surtout pas. Voyons comment se sortir de cette situation.

Côté *Sabertooth*, si la LED bleue *status1* est allumée, le module est sous tension.

## Réalisations à l'INSA de Lyon à partir de cette base

À l'INSA de Lyon, en deuxième année (bac+2), les élèves ingénieurs réalisent un prototype, en groupes de 24 élèves (30 groupes), à partir d'une électronique standard à base d'E-blocks. Le groupe choisit soit de travailler avec cette base électronique, éventuellement modifiée, soit de ne pas s'en servir du tout.

En 2011-2012, un véhicule était proposé. Capable de transporter un étudiant, il devait passer sous un tunnel de 53 cm de haut. Une course entre les différents véhicules s'est déroulée le 21 juin[VI].

Le groupe 42 a choisi de respecter le standard. La **figure 5** montre le prototype pendant la course. Les moteurs EMG30 de 4 W ont été remplacés par deux moteurs MY1016Z3 de 350 W. Les deux batteries de 7Ah fournissent la puissance et les quatre piles LR06 alimentent la commande. On reconnaît la carte moteur et la carte WiFi EB006 de l'AndroCAR. Grâce au 18F26K22 programmé sous Flowcode, le véhicule répond par WiFi aux ordres du terminal Android. La position et la vitesse du véhicule sont connues grâce à l'E-Block GPS, un pilotage local du véhicule est possible grâce au manche et il est possible d'afficher certains paramètres localement grâce à l'E-Block afficheur LCD.

Le véhicule pouvait être commandé par une tablette Android, mais le groupe 42 a trouvé les quatre pilotes plus performants à bord du véhicule qu'à distance. Ceux-ci ont donc pris place à bord pendant la

course.

Le groupe EURINSA2 a légèrement modifié la structure électronique du véhicule standard : moteur sur la roue arrière et moteur pour la direction des roues avant, potentiomètre pour l'angle des roues avant. Le programme Flowcode embarqué a été beaucoup retouché, car une régulation PID de la commande d'angle des roues avant est nécessaire.

Les autres innovations des EURINSA2 sont l'embarquement d'une webcam et d'un ordinateur sur *Scorpio*. Celui-ci peut ainsi suivre le ballon rose promené devant sa caméra. Voici en bref comment fonctionne le prototype :

La caméra envoie le flux vidéo vers *Matlab* sur l'ordinateur portable. *Matlab* calcule la position et la taille du ballon et envoie ces données au programme Java sur l'ordinateur. Java calcule l'angle des roues et la vitesse nécessaire au suivi du ballon et les envoie par WiFi à l'électronique semi-standard. Le circuit semi-standard effectue les actions demandées et retourne par WiFi les réponses aux ordres reçus.

Le groupe SCAN71 a choisi de réaliser un gyropode baptisé *Clémobile*[VII], inspiré du *Wheelie*[VIII] d'Elektor dont l'électronique a été modifiée. Un accéléromètre et un gyroscope sont bien sûr incontournables.

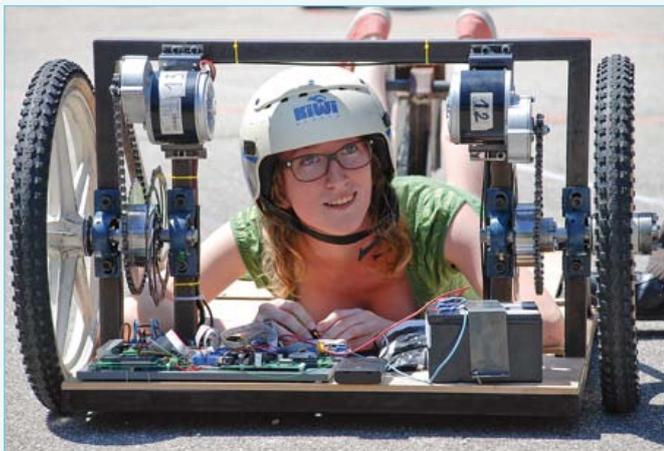


Figure 5 : Version du groupe 42. L'électronique embarquée est strictement identique au modèle réduit.



Figure 6 : Scorpio du groupe Eurinsa2. Les mêmes E-Blocks sur la même platine que le groupe 42 sont embarqués, mais ici une webcam permet de suivre le ballon rose! Les données sont traitées par l'ordinateur embarqué puis sont envoyées vers le véhicule par WiFi.

Côté EB069, la LED power doit d'abord s'allumer : le module est sous puissance. 5 s plus tard, la LED wireless s'allume, s'éteint brièvement 10 s plus tard, pour enfin se rallumer. Vous devez alors voir le réseau WLAN du module EB069 sur n'importe quel appareil équipé : téléphone Android, ordinateur portable...

Une fois ces trois LED allumées, vous devez pouvoir configurer votre module EB069 selon la procédure expliquée un peu plus tôt dans le texte.

Si, arrivé ici, votre ensemble ne fonctionne toujours pas, comment faire pour savoir si cela provient de votre programme ou de votre électronique locale ?

Pour cela, il faut tester les commandes réseau pour savoir si elles

réagissent correctement ou non, en utilisant Network Stuff[IV] téléchargé sur un ordinateur sous Windows équipé de WiFi.

Testons la réception par le prototype : remplir les champs UDP client tels qu'indiqués sur la **figure 4** puis cliquer sur Start. La fenêtre du bas apparaît. Cocher *Send hexa data* et *Receive hexa data* puis taper 1 dans le champ de saisie puis valider. Le moteur 1 devrait reculer à pleine vitesse. Ça ne fonctionne pas ? Deuxième chance : taper FF. Le moteur 2 devrait avancer à pleine vitesse. Si ce n'est pas le cas, cherchez l'erreur côté véhicule (configuration de la carte WiFi ou branchements). À cette occasion, vous pouvez, si nécessaire, remonter les moteurs pour qu'ils tournent dans le bon sens.

Vous n'avez pas d'ordinateur portable WiFi mais un ordinateur fixe

Flowcode permet de réaliser rapidement le filtre complémentaire et la régulation PID de stabilisation. Comme ces tâches en temps réel sont lourdes pour un PIC à 8 bits, la platine pour PIC EB006 a été remplacée par une platine pour dsPIC EB064 qui fait p. ex. le calcul en virgule flottante en un seul cycle d'horloge. La migration de Flowcode vers Flowcode dsPIC a été obtenue par un simple copié/collé. Dans cette configuration, le µC embarqué effectue facilement toutes les tâches demandées.

Clémobile fonctionne selon 3 modes : avec un pilote à bord comme *Wheelie*, mais aussi sans pilote, soit stabilisé comme on peut le voir sur la **figure 7**, soit couché sur une roulette pour pouvoir passer sous le tunnel de la compétition.

Les véhicules des groupes 42, SCAN71 et EURINSA2 (et ceux de cinq autres groupes) peuvent être pilotés par le même programme Android standard très proche de ce qui est présenté dans cet article, car ils peuvent être dirigés par les mêmes commandes UDP standardisées. Autour de ce standard, chaque groupe a développé ses propres solutions mécaniques, électroniques et informatiques. À votre tour d'imaginer vos propres réalisations à partir des bases présentées dans ces deux articles.

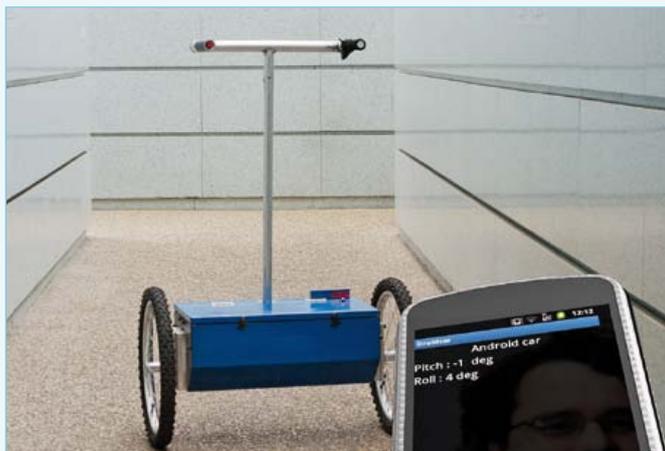


Figure 7 : Le même programme Android permet de piloter à distance un gyrocoptère en équilibre sans pilote. Une version améliorée du *Wheelie* d'Elektor. Réalisation du groupe Scan 71. Ce véhicule fonctionne évidemment aussi avec une personne à bord !

sans WiFi ? Mais vous avez forcément un téléphone Android ! Nous avons développé pour les étudiants de l'INSA l'application *Packet Injection* qui permet d'envoyer des *packets* UDP. Nous la mettons à disposition gratuitement sur *Google Play* pour les lecteurs d'Elektor[V].

Testons l'émission par le téléphone : modifier l'adresse IP dans le programme Android pour qu'elle corresponde à celle de votre ordinateur portable et utiliser le programme *Netstuff*. Vous pouvez configurer *Netstuff* en réception, pour voir s'il reçoit les commandes. Les Linuxiens utiliseront *Netcat*, inclus dans la plupart des distributions.

(120365)

**Listing 6 : blocage de l'écran en mode portrait, dans le Manifest.**

```
>
activity android:label="@string/app_name"
          android:name=".DroidcarActivity"
          android:screenOrientation="portr
ait"
>
```

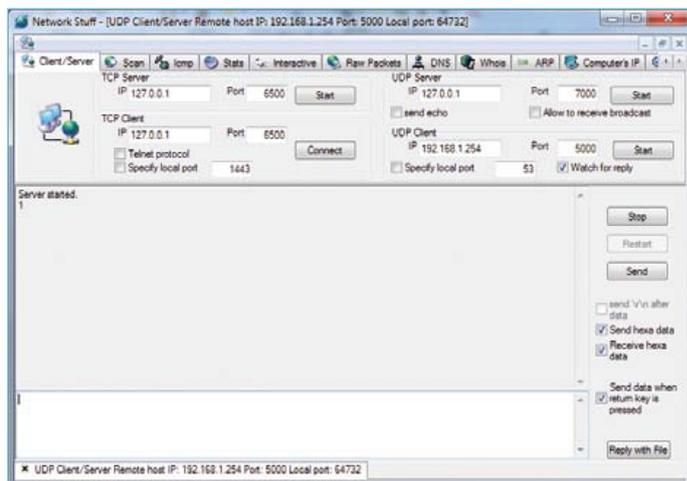


Figure 4 : Connexion en UDP sous Network Stuff

**Liste des composants**

- 1 x carte WiFi EB069
- 1 x platine BP232
- 1 x Sabertooth 2 x 25 A V2.00
- 2 x moteur EMG30
- 2 x étrier de montage EMG30
- 1 x paire de roues DRG001
- 1 x coupleur pour 8 piles AA
- 1 x connecteur 9 V

**Fournisseurs possibles pour la France : Elektor et Gotronic.**

- I <http://www.elektor.fr/120364>
- II <http://www.dimensionengineering.com>
- III [www.elektor.fr/120365](http://www.elektor.fr/120365)
- IV <http://jacquelin.potier.free.fr/networkstuff/>
- V <https://play.google.com/store/apps/details?id=ot.sembo.packetinjection>
- VI <https://www.youtube.com/watch?v=pSoaecDOZUM>
- VII <http://mofosegway.free.fr/>
- VIII <https://www.elektor.fr/outils/wheelie.987268.lynx>

**L'équipe de rédaction**

Benoit PIERRET (enseignant) et Nicolas STOULS (enseignant-chercheur) ont fabriqué en 2009 leur premier prototype piloté par téléphone Android : un lance-balle de tennis. L'intérêt pédagogique de la structure électronique de cette réalisation a permis de faire des projets transversaux entre plusieurs disciplines. En 2011, ils ont ouvert avec Yann Ricotti (enseignant), un nouveau cours : « Systèmes embarqués : du robot au téléphone ». Tous les enseignements cités ont lieu en deuxième année de l'INSA de Lyon avant spécialisation des élèves dans une discipline.

# Pansanitor (1928)

Karl-Ludwig Butte (Allemagne)



Les appareils électriques de stimulation musculaire détendent les muscles ; ils soulagent douleurs, névralgies, et autres affections au moyen de courants alternatifs à basse ou haute fréquence. Produire ces courants à l'aide des semi-conducteurs modernes, c'est du gâteau, mais il y a 80 ans, à l'époque où les transistors – je ne parle même pas des circuits intégrés et microcontrôleurs – n'étaient pas des composants courants, c'était une autre affaire. Saisissons l'occasion d'examiner le savoir-faire (presque) oublié de cette époque, grâce au *Pansanitor*, un appareil de stimulation musculaire de 1928.

## Stimulation musculaire : utilisation et fonctionnement

«Le rayonnement émanant de courants à haute tension et haute fréquence est, tout comme les hautes tensions fournies par un électrophore, supposé avoir un effet apaisant sur le système nerveux en général et sur les nerfs endoloris en particulier.» C'est ce qu'on peut lire dans le livre *High frequencies for the sick and the healthy: a medical companion* [1], qui cite le *Gynaecology Handbook* [2] du Prof. Opitz. Toutefois, il ne s'agit pas là de la seule application de l'électrothérapie. Grâce aux appareils comme le *Pansanitor* utilisés pour traiter le métabolisme, les organes respiratoires, ou la peau et les cheveux, le patient retrouve rapidement la santé... c'est du moins ce que les auteurs du livre voulaient vous faire croire.

Des appareils de stimulation musculaire sont toujours utilisés par la médecine moderne, par exemple dans les cas de myasthénie ou de troubles de la circulation, et plus généralement contre la douleur. Le rayonnement thermique est utilisé dans de tels cas, il pénètre plus profondément que, par exemple, la lumière infrarouge.

## Un peu d'histoire

L'appareil que nous autopsions date de 1928, l'année où Amelia Earhart est devenue la première femme à traverser l'Atlantique, où Alexander Fleming a découvert la pénicilline, et où Coca-Cola est devenu le premier parrain des jeux olympiques. Trois ans plus tôt, le physicien Julius Edgar Lilienfeld déposait le premier brevet décrivant le principe de fonctionnement du transistor [3].

La technologie de base qui a ouvert la voie à l'explosion des appareils de stimulation musculaire a été développée par Nikola Tesla dans les dernières années du 19<sup>e</sup> siècle. Sous le titre *High frequency oscillators for electro-therapeutic and other purposes* [4] il publie les résultats de ses recherches dans *The Electrical Engineer* en 1898. À partir d'une bobine d'allumage ordinaire, il décrit, plus du point de vue d'un ingénieur que d'un médecin, le principe de ses dispositifs expérimentaux et leurs applications. Une bobine d'allumage est constituée simplement d'un transformateur, d'un condensateur et d'un rupteur. Pour économiser sur les matériaux, Tesla câblait deux condensateurs en parallèle afin de réduire la différence de potentiel entre les connexions côté secondaire, ce qui avait pour inconvénient une réduction de la fréquence du courant alternatif, compensée par l'ajout d'une bobine aujourd'hui dite de Tesla (fig. 4 de [4]). En guise d'électrodes, Tesla faisait le vide dans des ampoules de verre pour les remplir d'un gaz qui produisait une lueur violette, ce qui a donné la fameuse *violet wand*, dite « baguette magique violette » en français ou *viola vergo* en esperanto, très prisée également comme accessoire de stimulation sexuelle. La forme des ampoules de verre a été adaptée aux symptômes ou aux parties du corps à traiter (fig. 1).

# stimulation électrique musculaire d'un autre temps

## La peau de Pansanitor

Le Pansanitor ainsi que ses accessoires, câbles et supports d'électrodes accompagnés d'une impressionnante gamme d'électrodes en verre, était fourni dans une mallette de transport pratique, doublée d'un somptueux tissu violet (fig. 1). Le module de commande, chargé de produire les courants de stimulation, est logé dans un boîtier en bois noir, derrière une face avant en plastique couleur ivoire avec des inscriptions pour les deux boutons rotatifs et les prises (fig. 2), avec également, en grandes lettres, le logo de l'appareil, et le numéro de modèle. Aucune mention du fabricant. J'ai eu beau chercher cette information, notamment sur internet, sans succès jusqu'à présent.

Sur la gauche, quatre broches disposées en T permettent de relier l'appareil au secteur, la broche du milieu à gauche pour le neutre, les trois autres permettant d'utiliser au choix une tension alternative de 110 V, 150 V ou 220 V. Cet appareil était donc conçu de toute évidence pour l'exportation, puisqu'il fonctionne avec différentes tensions de réseau. Il incombait au client de choisir la broche adéquate pour son pays. En revanche, il n'y a ni protection contre les surtensions ni même de fusible, seulement des capuchons de plastique noir sur les broches inutilisées.

Le fil du cordon secteur de cet exemplaire a été remplacé par son précédent propriétaire, mais les prises en porcelaine d'origine sont toujours là et en bon état (fig. 3).

Parmi les deux gros boutons rotatifs noirs, celui de gauche sert d'interrupteur de mise sous tension et celui de droite commande la tension de sortie de l'appareil (et donc l'intensité de la stimulation).

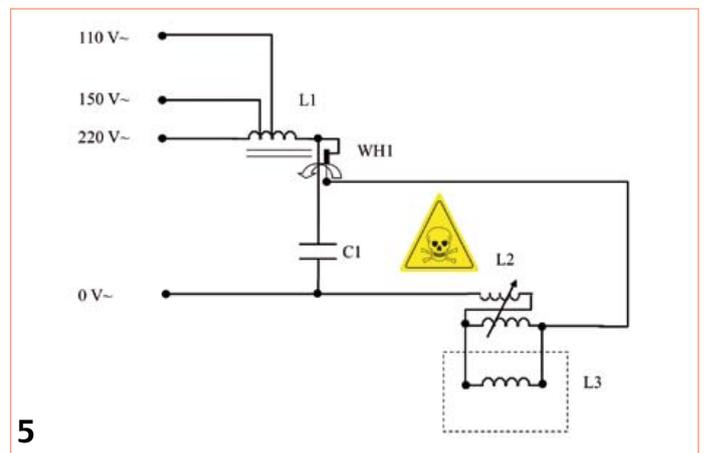
Les électrodes de verre sont alimentées via les deux embases au centre à l'aide des câbles du support d'électrodes, qui servait aussi de manche (fig. 4). Craignant pour ma sécurité, je n'ai pas osé mettre l'appareil sous tension !

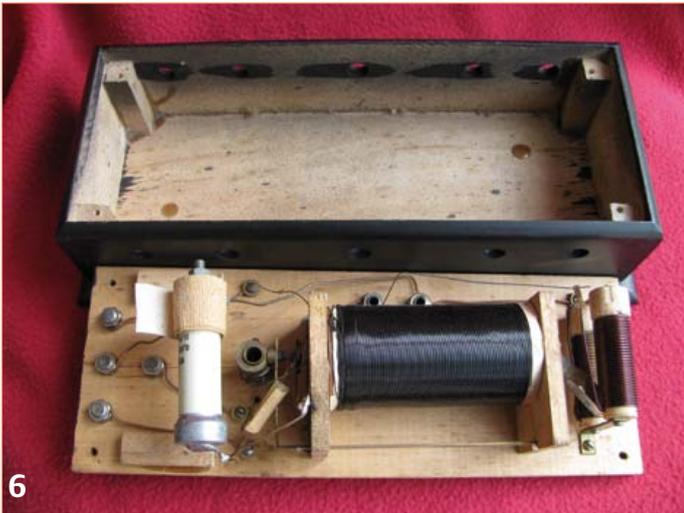
## Les tripes de Pansanitor

Vous trouverez en **figure 5** le schéma du Pansanitor. Jetez un œil aux entrailles de la bête avec la **figure 6**. La bobine L1 possède deux prises supplémentaires, pour l'utilisation des différentes tensions d'alimentation. Cela implique que tout est porté au potentiel du secteur ! L1, accompagnée du contact mécanique WH1, forme le rupteur ; conjointement avec une came sur le bouton de mise sous tension, cette partie commande l'alimentation globale de l'appareil (fig. 7). Lorsque l'interrupteur principal est passant et la tension du secteur présente, un champ magnétique est créé par L1, ce qui attire le contacteur WH1. Le circuit s'ouvre alors et la grande force contre-électromotrice en provenance de la bobine commence à charger C1 (25 nF). Ce condensateur est prévu (il serait plus exact de dire « était prévu ») pour fonctionner jusqu'à 2 kV. Vu l'âge du composant, mieux vaut ne plus s'y fier !

Avec l'interruption de la tension d'alimentation, le contact WH1 reprend sa position d'origine, entraînant à nouveau le passage du courant ; et c'est reparti pour un tour.

La haute tension ainsi produite est ensuite élevée par le transformateur de Tesla comprenant une bobine primaire (L2) et une bobine





secondaire (L3). L2 est constituée de deux enroulements reliés en série et montés l'un au-dessus de l'autre (fig. 8). Une paire de ressorts, qui assurent le contact entre les deux enroulements, peut être déplacée à l'aide du bouton de commande de l'intensité, ce qui permet de faire varier la tension en sortie : finalement, ces contacts ne sont que des prises intermédiaires réglables sur les enroulements, un peu comme sur un potentiomètre bobiné.

La bobine du secondaire L3 (fig. 4) se trouve dans la partie mobile de l'appareil dans laquelle peuvent être disposées les différentes électrodes de verre ; elle est reliée à l'appareil via un jeu de prises et embases. L3 ne comporte que dix tours d'un fil à deux conducteurs (fig. 9).

La totalité du circuit est montée à l'arrière du couvercle de l'appareil. Difficile de faire plus simple et plus élégant.

## Transformateur de Tesla inclus !

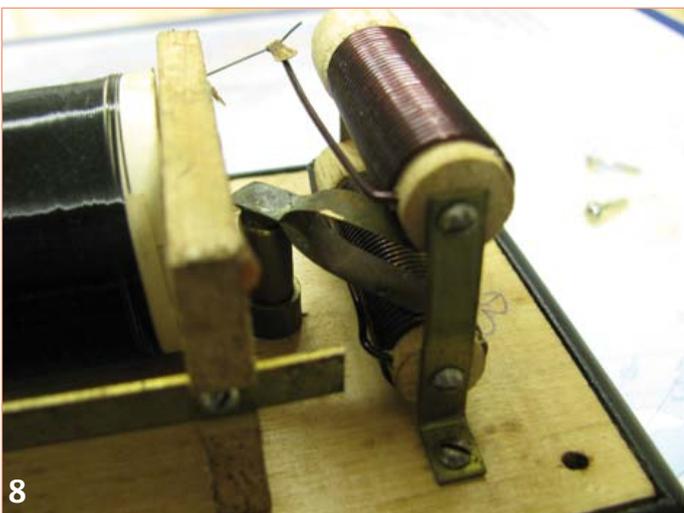
Tout comme le microcontrôleur trône dans la majorité des circuits modernes, la bobine de Tesla est au cœur du Pansanitor de 1928. Les  $\mu\text{C}$  font partie de la vie courante de nombreux lecteurs d'Elektor pour qui ils sont devenus des composants banals, dépourvus de tout mystère. La bobine de Tesla garde en revanche un caractère presque mystique, de nature à capter l'attention de l'électronicien même le plus blasé. La personnalité charismatique de son inventeur, Nikola Tesla, y est pour quelque chose, de même que l'aura qui entoure la tour de Wardencllyffe, appelée aussi « tour Tesla » [5], malheureu-

sement démolie en 1917. On retrouve facilement sur internet des images saisissantes de décharges électriques énormes [6]. Cependant, ce qui dans le transformateur de Tesla fascine le plus est sans doute le fait que l'amplitude de la tension produite dépende non seulement du rapport entre le nombre de tours des enroulements primaire et secondaire, mais aussi de la résonance entre les deux bobines [7].

(120177 – version française : Kévin PETIT)

## Liens

- [1] Thüringer Verlagsanstalt und Druckerei GmbH: Jena, 1930; [www.electrotherapymuseum.com/2005/HF/index.htm](http://www.electrotherapymuseum.com/2005/HF/index.htm)
- [2] Bergmann: Munich, 1922; page 354
- [3] <http://fr.wikipedia.org/wiki/Transistor>
- [4] The Electrical Engineer, volume XXVI, number 550, November 17 1898; [www.tfcbooks.com/tesla/1898-11-17.htm](http://www.tfcbooks.com/tesla/1898-11-17.htm)
- [5] [http://en.wikipedia.org/wiki/Wardencllyffe\\_Tower](http://en.wikipedia.org/wiki/Wardencllyffe_Tower)
- [6] [http://fr.wikipedia.org/wiki/Fichier:Lightning\\_simulator\\_questacon02.jpg](http://fr.wikipedia.org/wiki/Fichier:Lightning_simulator_questacon02.jpg)
- [7] [http://fr.wikipedia.org/wiki/Bobine\\_Tesla](http://fr.wikipedia.org/wiki/Bobine_Tesla)



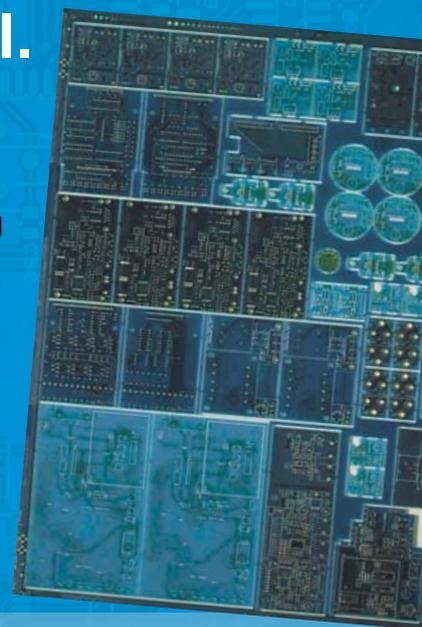
## La référence Européenne pour les C.I. Prototypes et petites séries.

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 isolations/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*

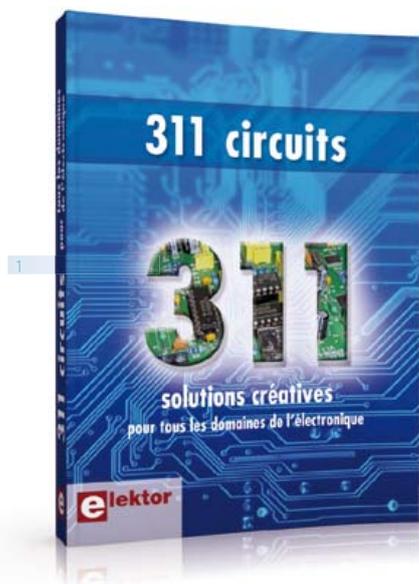
- 2 à 8 couches isolations/conducteurs  $\geq 100\mu\text{m}$
- A partir de 4 jours ouvrés

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

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

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

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



### Solutions créatives pour tous les domaines de l'électronique

#### 1 311 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

#### 2 L'électronique pour les débutants

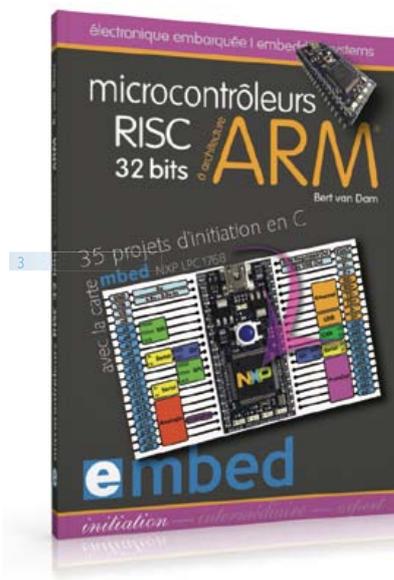
Par où commencer pour débiter en électronique ? Vais-je m'égarer en explorant l'internet, qui regorge de schémas, mais sont-ils fiables ? Me faut-il un livre avec des montages simples ou plutôt un livre sur les composants ? Après trente ans de pratique, l'auteur de ce livre, resté l'éternel débutant 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 €

#### 3 35 projets d'initiation en C avec la carte mbed NXP LPC 17688

#### 3 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

#### 4 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 !

#### 5 Embarquez Linux !

Linux est partout, même dans certaines machines à café. Souvent, l'électronicien tenté d'adopter ce



4

5

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 €

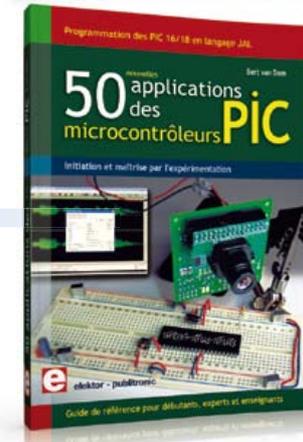
**Initiation et maîtrise par l'expérimentation**

**50 applications des microcontrôleurs PIC**

Voici 50 projets instructifs et utiles pour vous initier

6

7



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 €

8

9



**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





# récup' d'énergie

Raymond Vermeulen (Elektron)

Les capteurs, c'est comme les cousins : il y en a de proches qu'on voit tout le temps, d'autres moins souvent. Imaginez un tel capteur, éloigné, dont les nouvelles sont plus rares, mais pas moins intéressantes. Comment l'alimenter ? L'éloignement est trop grand pour une ligne d'alimentation directe. Une pile exigerait un entretien régulier. Et si on récupérait l'énergie sur place ? Les capteurs modernes se distinguent par leur faible consommation, avec un mode de veille à consommation ultra faible comme celui des petits microcontrôleurs. De sorte qu'un simple condensateur pourrait fournir au capteur l'énergie nécessaire le temps d'une mesure. Le reste du temps, ce condensateur tampon peut se recharger pendant que le circuit du capteur est mis en sommeil et ne consomme rien du tout. Je propose ici deux circuits intégrés de *Linear Technology* spécialisés dans cette fonction de gestion de faibles quantités d'énergie, conçus l'un pour être associé plutôt à un élément Peltier, et l'autre pour parasiter un élément piézo-électrique. Je l'utiliserai peut-être pour alimenter à partir d'une source d'énergie récupérée le détecteur d'éclairs présenté dans cette rubrique le mois dernier.

## LTC3108

Le LTC3108 est un convertisseur et un circuit de gestion d'énergie. Il est conçu pour fonctionner avec des sources de tensions ultra faibles, de l'ordre de 20 mV seulement (!) ; un élément Peltier p. ex. qui fournira une telle tension même en présence d'un faible gradient de température. Dans sa feuille de caractéristiques, le fabricant mentionne d'autres sources possibles. Le principe est celui du convertisseur élévateur à tension de sortie réglable. Le circuit est doté d'un régulateur 2,2 V à faible gradient de tension et présente une sortie  $V_{out}$  commutable, pour alimenter brièvement un circuit auxiliaire qui peut rester totalement inactif le reste du temps (pas de mode de veille). Un composant analogue est le bq25504 de *Texas Instrument*.

LTC3108

<http://cds.linear.com/docs/Datasheet/3108fb.pdf>

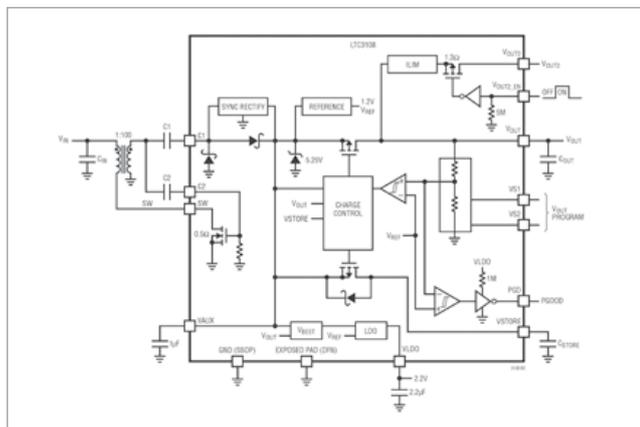


Figure 1. Synoptique des fonctions du LTC3108.

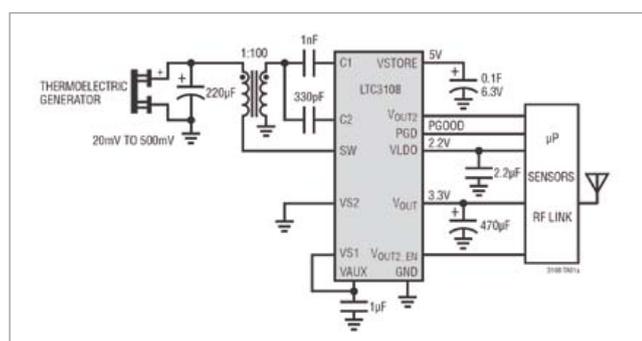


Figure 2. Application avec élément Peltier.

## LTC3588-1

Le LTC3588-1 a été conçu pour récupérer l'énergie à l'aide d'un élément piézo-électrique, mais ça fonctionne aussi avec deux plaques de métal à proximité d'un tube luminescent. Sa feuille de caractéristiques donne d'autres astuces de ce genre. Sur un élément piézo-électrique, la différence de potentiel peut atteindre des valeurs élevées que le régulateur limite à 20 V tandis qu'un hacheur l'abaisse à la valeur souhaitée.

L'élément piézo-électrique recommandé dans la fiche technique est si onéreux que cela vaudrait la peine d'essayer avec un simple buzzer piézo. Si ça marche, l'énergie suffira-t-elle ? Il y a de l'expérimentation dans l'air ! Si vous avez des choses à dire sur le sujet, n'hésitez pas : [redaction@elektor.fr](mailto:redaction@elektor.fr)

LTC3588-1

<http://cds.linear.com/docs/Datasheet/35881fa.pdf>

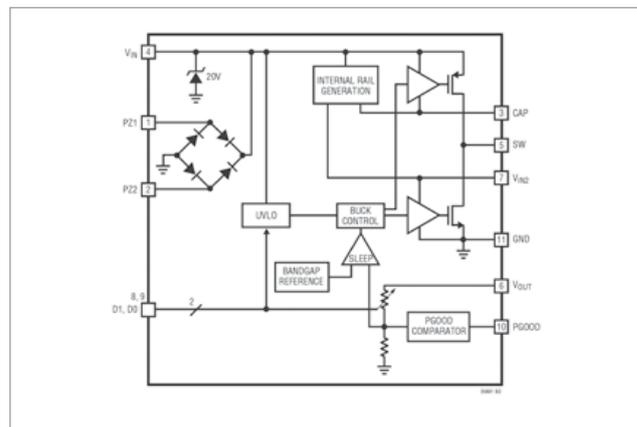


Figure 3. Synoptique des fonctions du LTC3588-1.

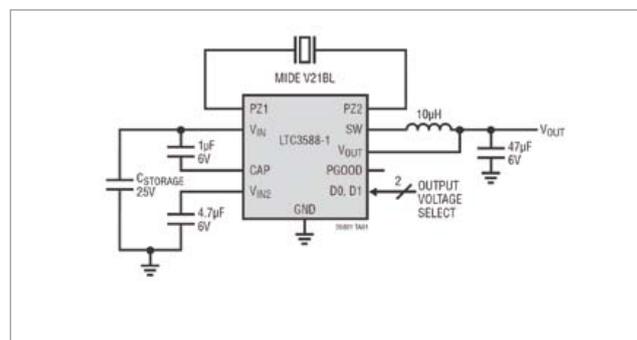
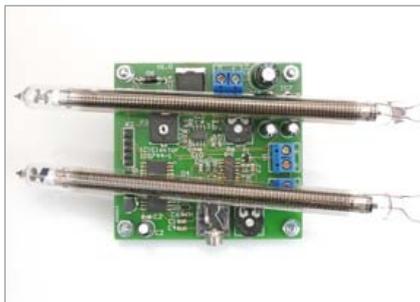


Figure 4. Application avec un élément piézo-électrique.



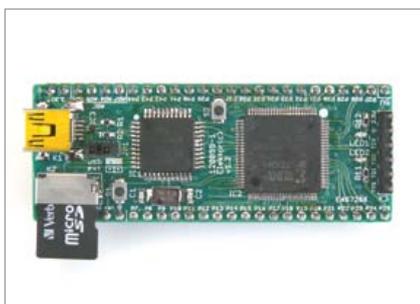
### VU-mètre à Nixie

L'engouement de nombreux électroniciens pour les tubes Nixie n'est pas étranger à l'abondance surprenante de cette production de thermomètres, d'horloges et d'hygromètres variés. Cette fois ce sera un VU-mètre. Celui-ci a été imaginé et conçu pour orner un amplificateur... à tubes, bien sûr. Il produit des effets lumineux spectaculaires, mais n'est pas calibré pour donner une indication rigoureusement exacte du signal de sortie. Au coeur du circuit, un Cypress PSoC de type CY8C27443, prend la mesure du niveau sur les deux canaux audio et commande les deux Nixie russes de type IN-9.



### « ... il n'y a pas que le 7805 dans la vie d'un électronicien... »

Le 7805 est sans conteste l'un des circuits intégrés les plus utilisés, certainement dans le domaine des régulateurs de tension. Avez-vous jamais songé à la quantité astronomique d'énergie dissipée en chaleur par tous ces soutiers de la régulation ? Que personne n'ait encore songé à le remplacer par un régulateur à découpage compatible broche à broche ! Au labo d'Elektor nous avons conçu un petit circuit imprimé avec un convertisseur TPS62150 dont la fonction de hacheur série procure un bien meilleur rendement ; alimenté entre 5,5 et 17 V, il délivre jusqu'à 1 A sous 5 V comme il se doit pour se substituer au 7805. Il suffira de changer quelques valeurs de résistances pour obtenir d'autres valeurs de tension régulée.



### carte FPGA

La route qui mène au succès en passant par les FPGA est assez sinueuse et semée d'embûches. Ces circuits intéressent, fascinent même, mais ils effraient aussi. Elektor a décidé de remettre les choses à plat avec une nouvelle carte FPGA facile, pratique, extensible. Le microcontrôleur embarqué assure la communication avec le PC via un port USB. La configuration du FPGA se trouve sur une carte de mémoire micro-SD (également sur la carte). Sous Windows, il est facile de transférer un fichier de configuration vers la carte SD ; lors de l'initialisation, ce fichier de configuration est écrit dans le FPGA.

*Sous réserve de modifications. Le numéro d'octobre paraîtra le 18 septembre*

**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-shop 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



**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!**





# Au service du génie

National Instruments met à la disposition des étudiants, le matériel et le logiciel dont ils ont besoin pour développer leur expérience, aller au-delà de la théorie, et réaliser l'importance du rôle de l'ingénieur dans la société.

>> Découvrez comment NI supporte la prochaine génération d'innovateurs, en visitant [ni.com/academic/f](http://ni.com/academic/f)

01 57 66 24 24