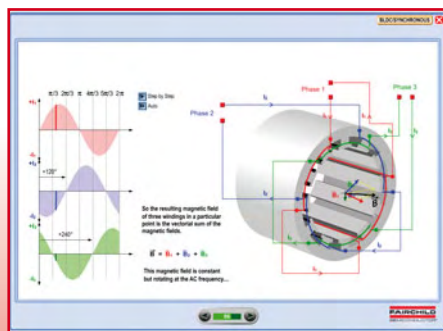


# Traceur de courbes de transistors sur USB

# La CAO gratuite

## outils de conception en ligne

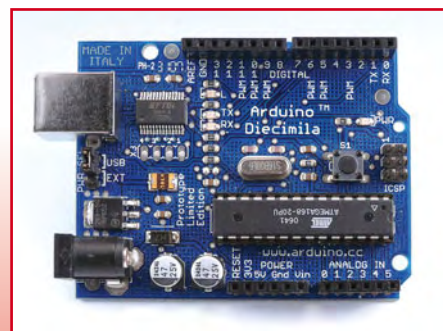


# Décodeur stéréo pour la mini radio FM



# Arduino

## les microcontrôleurs pour les nuls



L 19624 -368- F: 6,50 €

(B) 7,10 € • (CH) 12,20 FS • CAN 8,95 \$CAN • DOM surface 7,50 € • DOM avion 9,45 €  
Réu/A 9.40 € • Cal/S 850 CFP • Cal/A 1450 CFP • And 6.50 €





Quad, console, manga, rat, portable, MP3...  
vous cherchez des idées de cadeaux pour vos  
enfants ou vos petits-enfants ?

# inoculez-leur le virus

de l'électronique avant qu'il soit trop tard

- Collection complète des 58 numéros du mensuel d'initiation à l'électronique publiés par Elektor d'avril 1988 à sept. 1993
- Livret de 16 pages en couleur
- Vidéo d'initiation avec les personnages de la BD Rési & Transi

elex revient sur dvd-rom

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

Bonus : vidéo pour débutants

NIVEAU INITIATION



PUBLITRONIC ELEKTOR

## elex

explorez l'électronique

Collection complète au format PDF  
58 numéros du magazine d'électronique

Montages et rubriques d'initiation  
logique sans hic, analogique anti-choc





« ... entre Visual Basic  
à la fac et le C avec Elektor,  
vous hésiteriez, vous ?  
Moi pas ! »

Fabrice T., 20 ans, BAC+2



Avantages  
aux abonnés  
étudiants :

**-30%**

**Elektor, ma (p)référence en électronique**

**Prenez de l'avance,  
prenez un abonnement !**

**Les avantages exclusifs de l'étudiant :**

- 30% d'économie sur le prix de vente au numéro
- jusqu'à **40% de remise** sur certains produits d'Elektor
- collection complète, livraison ponctuelle à domicile
- toujours à jour, toujours une longueur d'avance



**www.elektor.fr/abo • Tél. 01 49 19 26 19**

**elektor**  
electronics worldwide

## Êtes-vous écolo ?

« Ah oui ! » me dites vous. « Bien sûr ! » Vous faites attention à votre consommation d'énergie, vous éteignez les lumières dans les pièces inoccupées, vous ne gaspillez pas d'eau, vous roulez moins vite et vous réalisez des montages à faible consommation avec des composants de récupération. Et pas seulement parce que c'est la crise et que les banques s'effondrent autour de nous, mais aussi parce que vous trouvez que c'est bien de faire attention à l'environnement. Du coup, vous avez faites vos courses de Noël par Internet et vous achetez vos composants, livres et DVD sur eBay, au lieu de vous déplacer en voiture.

Mais saviez-vous que déjà en 2005 on estimait la consommation en électricité de tous les serveurs Internet du monde à 180 TWh (térawattheure), soit 1,2% de la consommation mondiale d'électricité ? Quelqu'un chez Sun a estimé que chaque transaction eBay produise 55 gr de CO<sub>2</sub> et chaque recherche sur Google en produise 6,8 gr. C'était il y a quatre ans et Internet a continué à se développer ; nous pouvons donc supposer que ces chiffres ont augmenté. Alors, la prochaine fois que vous envoyez par courriel ce fichier hyper rigolo à tous vos contacts, pensez à la quantité de CO<sub>2</sub> que cela produise. Et aussi quand vous faites vos courses sur eBay et vous faites venir de Chine un sachet de résistances CMS. Êtes-vous vraiment écolo ?

Et que fait Elektor ? Eh bien, à part vous en parler, pas grand-chose. C'est vrai, il y a une astuce dans ce numéro pour faire des oscillateurs à quartz faible consommation, mais c'est tout, nous brûlons les calories comme d'habitude. Alors, achetez donc vos composants dont vous aurez besoin pour réaliser les montages de ce numéro chez votre revendeur du coin, c'est mieux pour l'environnement. En plus, vous faites vivre l'économie locale, ce qui aidera à combattre la crise, et vous pouvez raconter vos blagues aux autres clients : vos nouveaux ami(e)s.

Source : [blogs.sun.com/rolfk/entry/your\\_co2\\_footprint\\_when\\_using](http://blogs.sun.com/rolfk/entry/your_co2_footprint_when_using)

Clemens Valens  
Rédacteur en Chef

# Traceur de courbes pour transistors, sur U

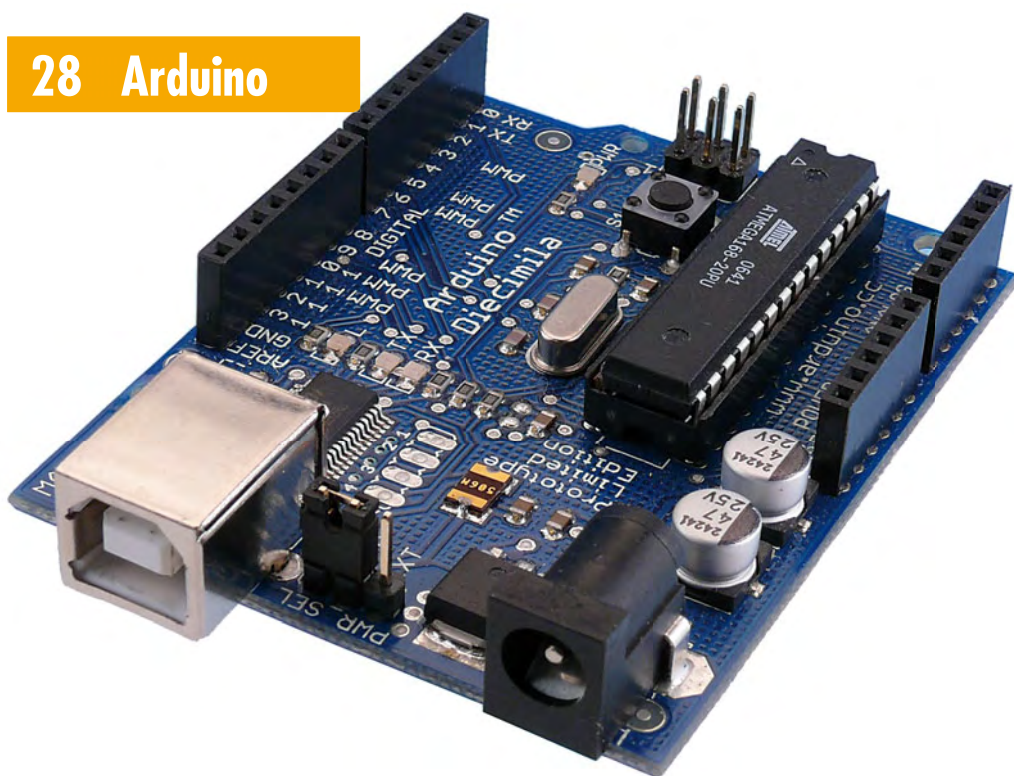
## 64

Ce montage pratique permet de tracer les courbes I/U caractéristiques de transistors NPN/PNP, N/P-JFETS et N/P-MOSFET. Au cœur du montage se trouve un microcontrôleur R8C/13, qui envoie les données mesurées par une liaison USB vers une application Windows écrite en Visual Basic.

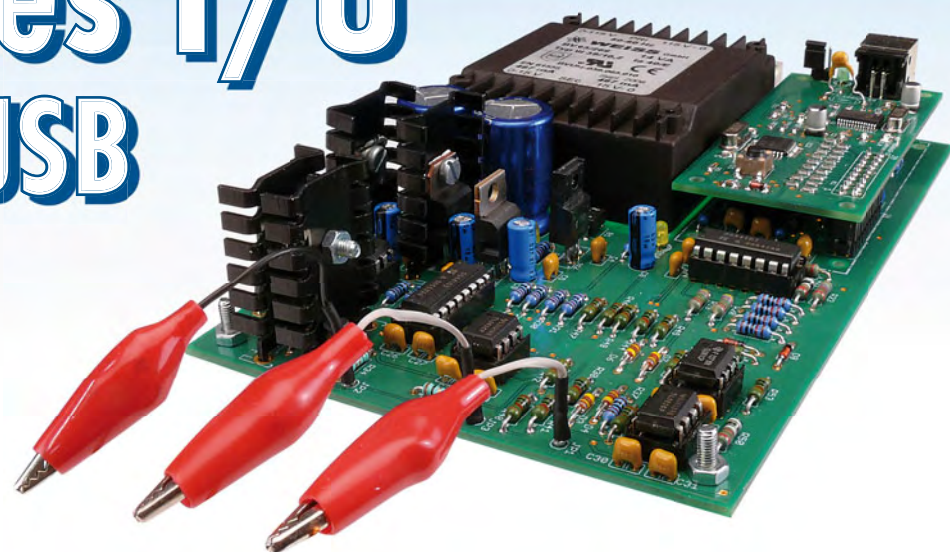
## Les microcontrôleurs pour les nuls

Il paraît qu'Arduino est un prénom italien, mais quand on fait une recherche sur Internet on trouve surtout des dizaines de références touchant à l'électronique et la programmation. En plus, ces références ont souvent un rapport avec l'art. L'électronique et de l'art, voilà un sujet intéressant et qui mérite d'être creusé. Alors, c'est quoi exactement, Arduino ?

## 28 Arduino







## pratique

- 22** ATM18 - Triangulation optique
- 28** Arduino  
Les microcontrôleurs pour les nuls
- 34** Cours BASCOM-AVR 5
- 38** Pont USB-I<sup>2</sup>C
- 42** Décodeur d'éclairage de voiture
- 47** Emetteur de guitare tout simple
- 56** Hygromètre programmable
- 64** Traceur de courbes I/U pour transistor, sur USB
- 72** Détecteur d'ultrason
- 76** Le décodeur stéréo

## technique

- 16** La CAO gratuite
- 27** Economie de courant sur les oscillateurs à quartz
- 46** Pilote de LED de puissance PR4101 à gradateur
- 48** Programmer en C les FPGA
- 52** C plus que C (1)
- 71** Sauve qui peut !

## info & marché

- 6** Ours
- 8** Courrier
- 10** Infos & Actualités
- 62** Elektor Live !
- 63** Kit de développement radio 2,4 GHz Decibit
- 84** Avant-première

## récréation

- 75** Rétronique
- 79** Hexadoku

## 48 Programmer en C les FPGA



On décrivait communément les FPGA dans des langages spéciaux, d'apprentissage difficiles, comme le VHDL. Pourtant, ça marche aussi en C ! Comparé à un programme séquentiel d'un microcontrôleur, une considérable accélération est possible. Nous allons montrer, à l'aide de l'environnement de développement Altium Designer, comment s'opère la conception d'un FPGA en langage C.

## 56 PICBASIC

### Hygromètre programmable

La gamme PICBASIC du coréen Comfile Technology compte huit PIC programmables en langage BASIC. Le PICBASIC-3B utilise un PIC16C876 et quelques composants discrets. C'est le composant idéal pour débiter dans la programmation des microcontrôleurs sans assembleur ni langage C.





# elektor international media

Elektor International Media propose une plateforme multimédia et interactive destinée à tous les électroniciens.

Du professionnel passionné par son métier à l'amateur aux ambitions professionnelles.

Du débutant à l'expert, de l'étudiant au professeur. Information, éducation, inspiration et récréation.

Analogique et numérique. Théorie et pratique.

**English**  
**German**  
**Dutch**  
**French**  
**Chinese**

**Italian**  
**Spanish**  
**Swedish**  
**Finnish**

**32<sup>ème</sup> année, n°368 février 2009**

ISSN 0181-7450 Dépôt légal : janvier 2009 CPPAP n° en cours

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

Numéro de compte : 002-007-97-026

IBAN : FR76 1873 9000 0100 2007 9702 603

BIC : ABNAFRPP

Monnaie : Euro - Branche ABN AMRO : Paris, France

*Elektor désire être une source d'inspiration pour ses lecteurs, les intéresser à l'électronique, par la description de projets à faire soi-même, et les tenir au courant des développements en électronique et en micro-informatique.*

Elektor paraît 11 fois, le numéro de juillet/août est un numéro double.  
Il existe, sous le nom Elektor, des éditions anglaises, allemande, espagnole, française et néerlandaise. Elektor est vendu dans plus de 50 pays.

*Conformément à la loi "Informatique et Liberté", vous bénéficiez d'un droit d'accès et de rectification des données vous concernant. Sauf refus écrit de votre part auprès du service abonnement, ces informations pourront être utilisées par des tiers.*

Rédacteur en chef international : Wisse Hettinga

Rédacteur en chef France : Clemens Valens ([redaction@elektor.fr](mailto:redaction@elektor.fr))

Maquette et graphisme : Giel Dols, Mart Schroijen

Rédaction internationale : Harry Baggen, Thijs Beckers, Jan Buiting, Ernst Krempelsauer, Jens Nickel

Secrétariat de rédaction : Hedwig Hennekens

Rédaction technique : Antoine Authier (chef labo), Ton Giesberts, Luc Lemmens, Daniel Rodrigues, Jan Visser, Christian Vossen



# Four à refusion CMS d'Elektor

**NOUVEAU**

➡ **Elektor démocratise la soudure par refusion**



**elektor**  
CHOPPE

- Outil professionnel pour l'amateur et le pro
- Idéal pour bureaux d'études, écoles, universités, PME et particuliers
- Notice en français
- Utilisation facile grâce aux menus
- Sélectionné, testé et certifié par Elektor
- Service après-vente assuré par Elektor
- Vidéo de démonstration et téléchargements gratuits sur [www.elektor.fr/four\\_cms](http://www.elektor.fr/four_cms)

#### Spécifications :

Surface de platine effective : 28,0 × 28,0 cm  
 Consommation max. : 1650 W  
 Alimentation : 230 V<sub>AC</sub>  
 Dimensions : 41,8 × 37,2 × 25,0 cm  
 Poids net : 16,7 kg

Prix : 1429 €\*  
 Référence : 080663-91

\*hors frais de port

Informations complémentaires et commandes :

**[www.elektor.fr/four\\_cms](http://www.elektor.fr/four_cms)**

Directeur/éditeur : Paul Snakkers

Responsable du marketing : Carlo van Nistelrooy

Administration des ventes : [ventes@elektor.fr](mailto:ventes@elektor.fr)

Publicité : SL Régie - Sophie Lallonder  
 12, allée des Crételles - 37300 Joué-Lès-Tours  
 Tél : 02.47.38.24.60 - Fax : 02.90.80.12.22  
 E-mail : [sophie.lallonder@wanadoo.fr](mailto:sophie.lallonder@wanadoo.fr)

DROITS D'AUTEUR : © 2009 Elektor International Media B.V.

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

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

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

Elektor est édité par Elektor International Media B.V.  
 Siège social : Allée 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.



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

Hexadoku

Depuis sa parution en janvier 2006, je participe au jeu Hexadoku avec plaisir et j'examine aussi les résultats, espérant figurer un jour parmi les lauréats. Même si tout recours est exclu, ce que je conçois, je tiens à faire remarquer que certains gagnants le sont un peu trop répétitivement : sur le n°366 de décembre 2008, trois gagnants sur quatre ont déjà été cités, voir le site :

[www.microfer.fr/geryko/](http://www.microfer.fr/geryko/)  
Si certains lecteurs pouvaient s'abstenir et avoir un minimum de moralité, voire de politesse, j'en serais très heureux. Une autre solution consisterait peut être à exclure les joueurs ayant déjà gagné ? Faut-il envoyer des centaines de réponses pour espérer figurer parmi les gagnants ?

Géry SZCZEPANSKI

Merci pour votre coup de gueule. Le tirage au sort est fait par ordinateur et les gagnants à répétition ont tout simplement eu de la chance. Continuez surtout à nous envoyer vos solutions puisque, pour citer le loto : 100% des gagnants ont tenté leur chance.

Transformateur Electronique (n°366)

Fidèle lecteur depuis le n°1, je me permets de vous apporter des informations complémentaires sur l'article « Transfo

Electronique », page 82 et suivantes du n°366. Ces transfos (en fait comme vous le dites bien) sont des alims à découpage (ou plutôt, pour être parfaitement correct, des auto-oscillateurs de puissance) et la majorité de ceux-ci sont basés sur la note d'application AN528 de



STMicroelectronics. Même si, sur certaines réalisations récentes, les transistors bipolaires sont remplacés par des MOSFET, le principe reste le même. Cette note d'application explique bien le fonctionnement et est très didactique. Mais, la chose la plus intéressante, si l'on regarde bien le schéma est de pouvoir transformer ces transfos en transfo-variateur. En effet on voit que l'oscillateur est déclenché par un diac avec un réseau R-C. Si l'on ajoute en série avec cette résistance R un potentiomètre d'une valeur égale à environ dix fois la valeur de R, on obtient un variateur de 0 à 100%. Il faut être un petit peu

connaisseur en électronique quand même (mais je pense que c'est le cas de la majorité de lecteurs) pour repérer cette résistance, dont on doit lever une patte. Mais en s'aidant du schéma général de la note d'application on la trouve facilement. Elle vaut en général 100 kΩ à 220 kΩ selon les modèles et le condensateur associé. Surtout ne pas l'enlever : elle sert de butée quand le potentiomètre est à zéro. Si on l'enlève, quand le potentiomètre passe en dessous de la valeur de la résistance d'origine, c'est le flash assuré. Ceci est très intéressant si ce potentiomètre est couplé à un interrupteur ; cela permet un allumage progressif, et augmente donc largement la durée de vie des spots (ils ne seront plus soumis à un très fort courant lors de l'allumage quand le filament est froid), cela augmente le confort par exemple dans une salle de bains le matin (on ne prends

plus les spots pleine puissance au réveil, pas toujours agréable). Mais surtout, pour le bricoleur c'est aussi très intéressant, car si, comme vous le suggérez, on redresse et filtre la sortie on obtient une alim variable de 0 V à 15 V de forte puissance pour pas cher. Les précautions d'usage relatives aux tensions secteur s'appliquent bien évidemment car une des bornes du potentiomètre est reliée au secteur redressé. Donc potentiomètre axe plastique, corps plastique si possible.

Electronic Transformer for a 12V Halogen Lamp, Fichera, P. et Scollo, R., AN528/0999, STMicroelectronics, 1999.

[www.st.com/stonline/products/literature/an/3707.pdf](http://www.st.com/stonline/products/literature/an/3707.pdf)  
Alain CAILLARD

Merci beaucoup pour cette excellente contribution !

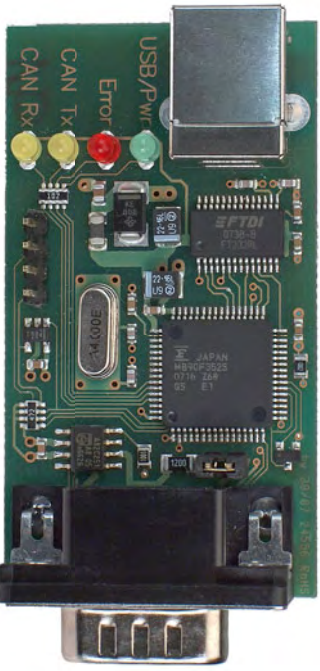
Tiny-CAN en anglais (n°363)

J'ai dernièrement acheté le kit adaptateur USB->CAN compact. J'ai monté le kit et installé le logiciel et tout fonctionne parfaitement. Par contre j'ai une question : le logiciel Tiny-CAN que j'ai téléchargé sur votre site est en allemand, est-il possible de l'obtenir ou de le configurer en anglais ? J'ai essayé de télécharger le logiciel directement sur [www.mhs-elektronik.de](http://www.mhs-elektronik.de) mais il est également en allemand. Sur les photos de votre article, le logiciel est en anglais !

Philippe MONNARD

D'après l'auteur le logiciel détecte automatiquement la langue à utiliser. Par défaut la langue est allemande, mais si l'on possède un PC avec Windows en anglais, le logiciel sera en anglais. Pour ceux qui n'ont pas de Windows en anglais, il propose la solution suivante :

- 1. Ouvrez l'Invite de commandes
- 2. Rendez-vous dans le répertoire Tiny\_CAN\CAN\_VIEW\windows

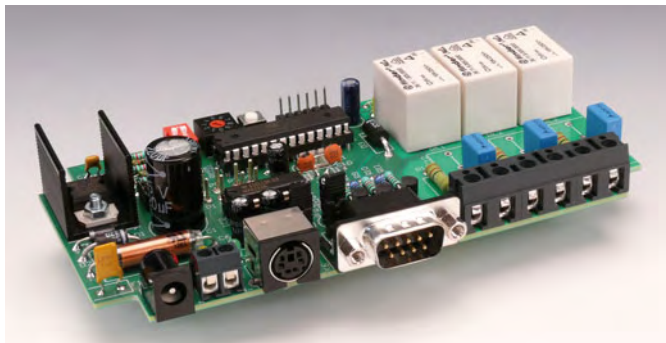




3. Tapez « Set LANG=en » suivi bien sûr par Entrée  
4. Tapez « can\_view », suivi également par Entrée  
Cette procédure a été validée sous Windows Vista.

## Télécommande par téléphone portable (n°365)

Depuis quelques temps déjà je pilote mon système d'alarme domestique à dis-



tance avec un téléphone portable (Siemens S35) et un PIC16F84. L'alarme est activé et désactivé par des appels téléphoniques. Pour m'avertir des événements, le système m'envoie un texto (ce qui coûte de l'argent) ou il laisse sonner le téléphone pendant 15 secondes (gratuit). Au début j'utilisais un câble de données de téléphone portable et un MAX232, mais je me disais qu'il devrait il exister des solutions plus simples. Le téléphone Siemens fonctionne avec des signaux d'environ 3 V, suffisants pour être reconnu comme niveau haut par un microcontrôleur. J'utilise une diode zener pour limiter à 3 V le signal 5 V du contrôleur. La communication entre le mobile et le microcontrôleur marche ainsi parfaitement, sans MAX232 ou sans câble de données (qui contient de toute façon un MAX232). Je charge également la batterie du téléphone avec une alimentation de 7 V branchée sur la broche 3 du connecteur du téléphone (connecteur Lumberg). Ainsi le système fonctionne correctement, même pendant mes absences

prolongées. Sur Internet on trouve plusieurs sites avec des descriptions détaillées des commandes AT (y compris les commandes Siemens), par exemple, [www.nobbi.com](http://www.nobbi.com). J'y ai trouvé PDU Spy, un petit outil qui rend le codage et le décodage des textos très facile. Les branchements des connecteurs de Nokia et Siemens y sont aussi décrits.

**Maik BUSSE**

## Les Pépites du Forum

### Programmer l'Explorer 16 sans outil externe (n°343)

Ca y est, suivez le lien : [forum.microchip.com/tm.aspx?m=338210&page=1&key=??](http://forum.microchip.com/tm.aspx?m=338210&page=1&key=??) ou fait un recherche sur **rafalw** dans [forum.microchip.com](http://forum.microchip.com) pour trouver le progiciel qui transforme le PIC18F4550 de la carte Explorer 16 en véritable PICKIT 2. N'oubliez pas de télécharger également le logiciel PICKIT2V2 disponible dans la même discussion. La version HEX à charger est : PK2\_Explorer16\_V0210007-BL.hex. Ensuite lancer Pickit2V2.exe et charger l'operating system du Pickit2. Une fois la mise à jour faite, il est possible de connecter la carte Explorer 16 directement au PC par le câble USB et d'utiliser le menu *Pickit2 Debugger* à partir de MPLAB ou l'application externe Pickit2V2. Cela

permet de se passer complètement d'un outil externe (ICD2, ICD3 ou REALICE). Super boulot de la communauté des PIC !  
**Jean Baptiste et RISC**

*Effectivement, ça marche !*

Toute la discussion se trouve ici : [www.elektor.fr/forum/forum/realisations-publiees-par-elektor/explorer-16-et-usb.366807.lynkx](http://www.elektor.fr/forum/forum/realisations-publiees-par-elektor/explorer-16-et-usb.366807.lynkx)

### Système d'exploitation temps réel pour Explorer 16 (n°343)

FreeRtos est un système d'exploitation temps réel bien adapté pour les microcontrôleurs. C'est un système libre fournit avec les codes sources. Il prend en compte les PIC18, PIC24 et dsPIC, ainsi que d'autres familles de microcontrôleurs. Pour ceux qui veulent s'aventurer : [www.freertos.org](http://www.freertos.org) Il est fourni avec un code exemple PIC24 et dsPIC fonctionnant avec l'Explorer 16. Bon développement.  
**meastro2**

*Et ça marche aussi !*

Toute la discussion se trouve ici : [www.elektor.fr/forum/forum/realisations-publiees-par-elektor/rtos-pic24-dspic.163213.lynkx](http://www.elektor.fr/forum/forum/realisations-publiees-par-elektor/rtos-pic24-dspic.163213.lynkx)

### DigiButler en ligne (n°358 & n°359)

J'ai branché mon DigiButler sur Internet, permettant ainsi les lecteurs d'Elektor de voir que cela marche vraiment. La carte a été programmée pour lire plusieurs températures, celle du processeur ColdFire y compris, et pour surveiller la lumière de ma cuisine.

Aller donc vérifier si la lumière de la cuisine est bien éteinte (vous pouvez le faire à ma place si besoin) ou s'il gèle chez moi. Rendez-vous sur : <http://81.56.186.109:59999>  
nom d'utilisateur : user  
mot de passe : 1234567  
Soyez indulgents si les temps de réponse ne sont pas à la hauteur d'un site conventionnel, ce montage est principalement prévu pour dialoguer avec un seul interlocuteur. Il est possible que vous soyez nombreux sur la ligne et comme chaque liaison génère une interrogation chaque seconde imaginez la charge. En outre il cohabite sur le réseau local avec quelques autres ordinateurs dont un serveur WEB (<http://www.ymasquelier.net>) qui disposent de priorités supérieures à mes yeux. Je remercie déjà ceux qui l'ont visité.  
**Yves MASQUELIER**



*Mais Yves, comment se fait-il qu'il n'y ait jamais de la vaisselle chez vous ?*

Le DigiButler a son propre forum : [http://www.elektor.fr/forum/forum/realisations-publiees-par-elektor/digibutler-\(04-05-2008\)-n-o-u-v-e-a-u-l.689499.lynkx](http://www.elektor.fr/forum/forum/realisations-publiees-par-elektor/digibutler-(04-05-2008)-n-o-u-v-e-a-u-l.689499.lynkx)

## Règles du jeu

- Publication de la correspondance de lecteurs à la discrétion du rédacteur en chef
- Les points de vue et opinions exprimées par les correspondants ne sont pas nécessairement ceux du rédacteur en chef ou de l'éditeur.
- La correspondance pourra, le cas échéant, être traduite ou éditée en longueur, clarté et style.
- En cas de réponse à COURRIER, veuillez s.v.p. indiquer le numéro concerné.
- Veuillez s.v.p. adresser votre correspondance : [redaction@elektor.fr](mailto:redaction@elektor.fr) ou

Elektor — le rédacteur en chef c/o Regus Roissy CDG  
1, rue de la Haye BP 12910 95731 Roissy CDG Cedex France

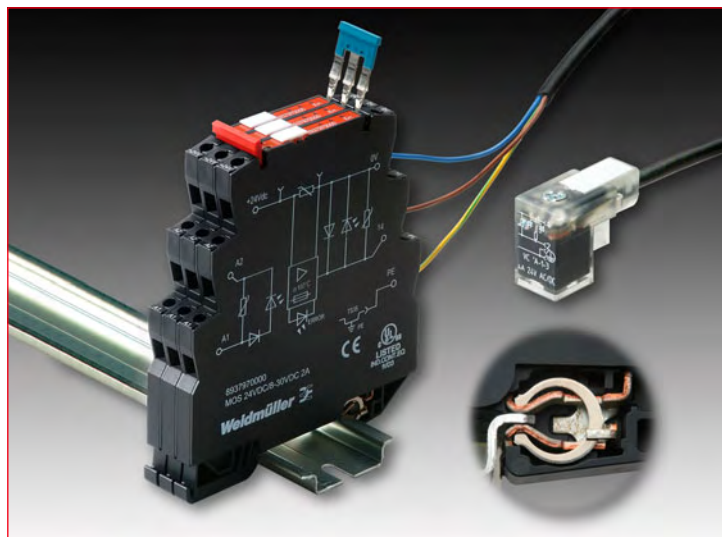
## Nouveaux relais statiques

Les nouveaux relais statiques MICROOPTO de Weidmüller bénéficient de nombreuses protections contre les surtensions et les perturbations transitoires qui leur garantissent un fonctionnement sûr et sans dérangement, ainsi qu'une grande longévité. Ils s'utilisent dans de très nombreuses applications industrielles et de procédés et représentent notamment la solution idéale pour la commutation de freins moteur et de contacteurs, le traitement des signaux rapides, le pilotage des électrovannes.

Les relais statiques classiques sont équipés d'un opto-coupleur pour l'isolation électrique et d'un amplificateur de commutation. Lorsqu'il s'agit de commuter des charges élevées, il est nécessaire d'amplifier la commutation. La solution : les nouveaux relais statiques de la gamme MICROOPTO de Weidmüller.

Les relais de la gamme MICROOPTO se distinguent par une protection intégrale : ils dis-

posent de pièces mécaniques leur confère fiabilité et longévité. Très compacts (6 mm de large),



posent d'une sortie résistante aux courts-circuits et sont universellement protégés, côté entrée, contre les surtensions. L'absen-

ce de pièces mécaniques leur confère fiabilité et longévité. Très compacts (6 mm de large), les MICROOPTO sont équipés d'un connecteur à visser acceptant des conducteurs jusqu'à 4 mm<sup>2</sup> de diamètre. Ils se

montent sur rail DIN TS 35. Des connexions transversales enfichables et une mise à la terre directement sur le module permettent de réduire considérablement les temps d'installation et de mise en service.

Les MICROOPTO se caractérisent par des temps de réponse très courts (jusqu'à 10 µs) et une insensibilité aux chocs et aux vibrations. Le matériau utilisé pour leur fabrication, le WEMID, est un thermoplastique spécialement conçu par Weidmüller qui supporte des températures de 120°C et offre une résistance au feu élevée (inflammabilité de classe V0 selon la norme UL 94). Ces relais sont équipés de LED signalant l'état du module. Ils sont approuvés CE et sont conformes aux standards EN 50178, IEC 62314 et UL508.

(080855-16)

[www.weidmuller.fr](http://www.weidmuller.fr)

## Le nouveau Toughbook doté d'un processeur Intel Atom

Panasonic a présenté le premier assistant clinique mobile (MCA) au monde doté d'un processeur Intel Atom. Le nouveau CF-H1 MCA fonctionne sur la base de la plateforme de référence Intel Mobile Clinical Assistant et a été spécialement conçu pour les médecins cliniciens. Le CF-H1, qui ne pèse que 1,5 kg, est un poids plume parmi les ordinateurs mobiles.

Il résiste aux éclaboussures, aux chocs et aux chutes, se désinfecte facilement et répond aux exigences maximales des environnements cliniques existants en matière de mobilité et de gestion de réseau. Il permet aux médecins et au personnel soignant d'accéder directement aux dossiers de l'hôpital lorsqu'ils se trouvent dans la chambre des patients et de consulter des informations précises sur l'évolution de la maladie et les médicaments prescrits aux patients.

Doté de batteries innovantes conçues par Panasonic et du processeur Intel Atom à faible

consommation, le CF-H1 affiche une autonomie de six heures, une performance hors du commun pour sa catégorie. De plus, ses

deux batteries, qu'il est possible d'échanger au cours du fonctionnement de l'appareil, garantissent une mobilité permanente,

sans période d'interruption.

Le CF-H1 se désinfecte facilement, ce qui réduit le risque d'infection auprès des patients. Le MCA est le premier Toughbook de sa catégorie sans ventilateur, ce qui limite le risque de transmission de germes par le biais de l'appareil. Ce dernier dispose également d'une surface sans rugosité, de touches encastrées et d'un écran LCD sans renforcement. De plus, aucun port d'accès n'est directement accessible.

Pour la surface extérieure, Panasonic a mis au point un matériau spécial qui résiste aux nettoyages fréquents à l'alcool. Le fabricant a également développé un logiciel spécial qui rappelle à l'utilisateur de désinfecter l'appareil à des intervalles réguliers et enregistre automatiquement dans les dossiers actuels de l'hôpital que la désinfection a été effectuée.

(080855-20)

[www.toughbook.eu](http://www.toughbook.eu)



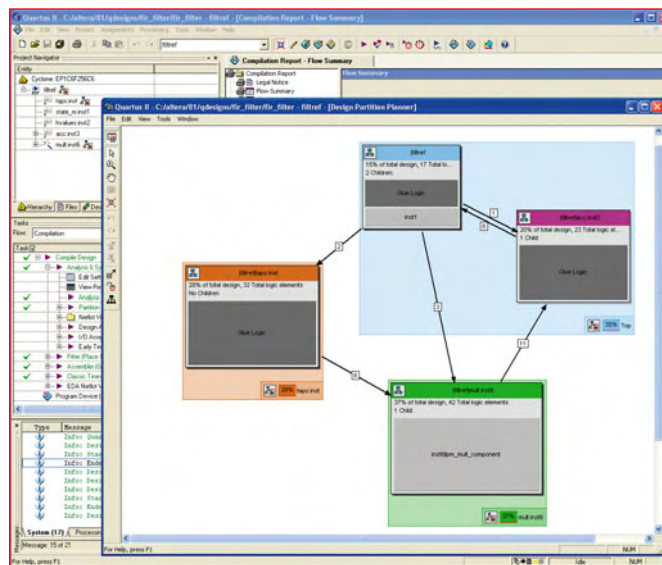


## Quartus II 8.1 disponible

Altera a dévoilé la version 8.1 de son logiciel de développement FPGA Quartus II. Cette nouvelle version permet toujours d'avoir des temps de compilation divisés par trois pour les FPGA haut de gamme par rapport aux plus proches concurrents, selon des tests internes. Les fonctions de productivité ont été améliorées permettant aux équipes de conception d'accélérer la clôture des délais et de puissance, d'abaisser les coûts de R&D et de raccourcir le délai de mise sur le marché. Le planificateur de partition du projet, présenté dans la version précédente de Quartus II, fournit désormais un partitionnement automatisé, permettant à davantage de concepteurs de s'appuyer sur les bénéfices en productivité de la compilation incrémentale. Avec Quartus II, il n'est plus nécessaire maintenant de modifier manuellement les gated clocks, en convertissant automatiquement les gated clocks en logique fonction-

nant de la même façon et prise en charge par l'architecture FPGA. La version 8.1 apporte des bro-

en charge les transceivers à 8,5 Gbps, le LVDS 1,6 Gbps et la mémoire DDR à 400 MHz.



ches et une nouvelle vitesse du Stratix IV dans un boîtier économique. Ce logiciel prend

Pour les concepteurs souhaitant mettre en œuvre un ASIC HardCopy, le logiciel Quartus II four-

nit un support initial aux ASIC HardCopy IV.

La nouvelle version de Quartus II prend désormais en charge les systèmes d'exploitation Red Hat Enterprise Linux 5 et CentOS 4/5 (32 bits/64 bits). L'interface de simulation externe a été améliorée. Un nouveau conseiller de broches aide pendant la création de broches et d'interfaces avec des cartes du marché. Il y a bien sûr des nouveaux cœurs IP et mégafonctions tel que le DSP, de la mémoire et des protocoles. Le moteur de synthèse physique a été optimisé permettant un gain de 20% en moyenne du timing critique par rapport à la version précédente.

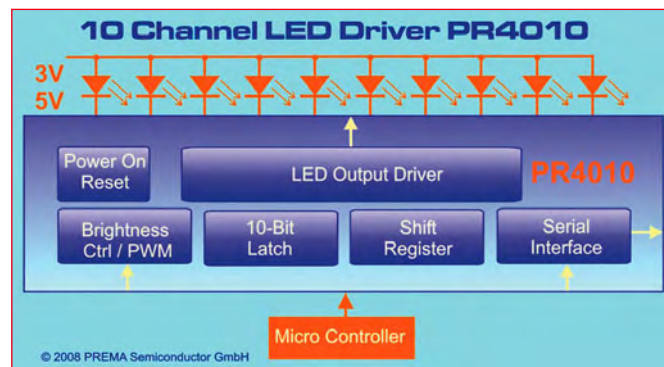
La version 8.1 de Quartus II Subscription Edition et la version web gratuite sont disponibles dès maintenant par téléchargement sur [www.altera.com/download](http://www.altera.com/download).

(080855-12)

[www.altera.com/quartus2](http://www.altera.com/quartus2)

## Pilote de LED à dix canaux

Le fabricant allemand PREMA Semiconductor a présenté un nouveau pilote de LED à dix voies. Le PR4010 peut piloter individuellement jusqu'à dix LED de n'importe quelle couleur. Si besoin, il est possible de connecter plusieurs PR4010 en série pour piloter plus de dix LED en utilisant une liaison série. De cette façon il est par exemple possible de piloter 75 LED d'un afficheur matrice de points de 15x5 points avec seulement deux PR4010 et cinq transistors additionnels. Dans cette configuration quinze sorties des deux puces sont utilisées pour piloter les quinze colonnes de l'afficheur, tandis que les cinq sorties restantes, renforcées avec un transistor PNP, mul-



tiplexent les cinq lignes de l'afficheur. D'autres applications sont des afficheurs à barre ou des afficheurs de statut. Chaque une des dix sorties du composant peut absorber jusqu'à

10 mA. L'interface série avec entrée à verrouillage permet le pilotage de chaque LED individuellement par un microcontrôleur externe. La variation de courant entre les voies ne dépasse

pas le  $\pm 5\%$ , ce qui garantit une luminosité uniforme des LED. Le courant des LED, comme leur luminosité, est ajusté avec une résistance externe. En plus, il est possible de le contrôler avec un signal MLP sur l'entrée « blank », qui permet l'activation/désactivation de toutes les voies indépendamment du port série. Le PR4010 a été conçu pour une alimentation de 3 V à 5,5 V.

Le PR4010 est logé dans un boîtier SSOP20 et des échantillons sont désormais disponibles sur demande.

(080855-10)

[www.prema.com/Application/pr4010\\_e.html](http://www.prema.com/Application/pr4010_e.html)

## Une famille de CAN sigma-delta à temps continu

Analog Devices annonce la famille de convertisseurs analogique/numérique (CAN) CTSD (continuous-time sigma-delta ou, en français, sigma-delta en temps

continu) qui offre le bruit le plus bas et la bande passante la plus large du marché. Les convertisseurs CTSD 16 bits AD9261 et AD9262 et le modulateur CTSD

AD9267 allient un bruit très bas et une haute gamme dynamique à une bande passante qui va jusqu'à 10 MHz. La nouvelle famille de CAN at-

teint ses performances uniques de vitesse, de précision et de bande passante en incorporant une technique révolutionnaire de convertisseur CTSD. Cette derniè-



re est idéale pour les infrastructures sans fil, les équipements médicaux et tout autre équipement hautes performances qui néces-

sité à la fois une résolution de données sans compromis et une large bande passante.

L'AD926x utilise les principes du sur-échantillonnage, de la mise en forme du bruit et des caractéristiques d'entrée particulières à son architecture pour atteindre de hauts niveaux de performances avec une plus grande facilité d'usage. La structure d'entrée résistive calme diminue les besoins de l'amplificateur pilote alors que le filtre en temps continu sur-échantillonné d'ordre supérieur atténue les signaux hors bande. Cela réduit les besoins

en filtres de large bande de base et autres circuits de conditionnement du signal.

Les composants AD9261 et AD9262 intègrent un multiplicateur d'horloge, des filtres de décimation et des convertisseurs de taux d'échantillonnage. Ce haut niveau d'intégration permet d'obtenir des débits modulables de sortie de données compris entre 30 MSPS et 160 MSPS. L'AD9267 qui propose uniquement le coeur de modulateur hautes performances de 640 MSPS et le multiplicateur d'horloge, envoie les données haute vitesse

se directement vers la sortie. Les concepteurs peuvent alors décharger le composant des fonctions de traitement du signal en les envoyant vers un FPGA ou un autre processeur. La consommation d'énergie des nouveaux convertisseurs CTSD va de 150 mW par canal à 350 mW par canal.

La famille de CAN CTSD AD926x est déjà échantillonnée. Elle sera disponible en volumes de production en avril 2009.

(081045-1)

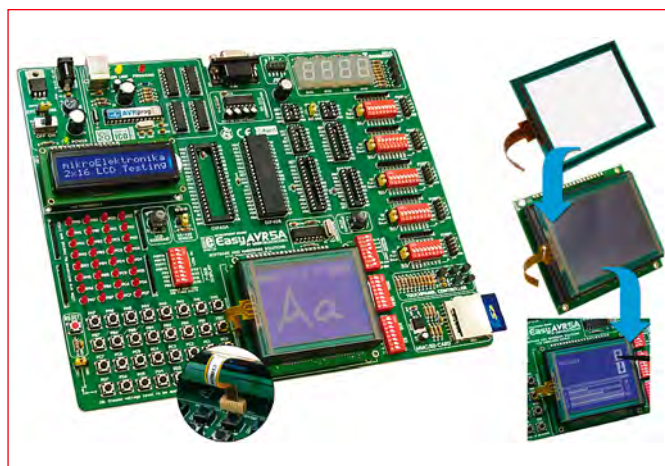
[www.analog.com](http://www.analog.com)

## Nouvel outil de développement pour microcontrôleurs AVR

mikroElektronika a récemment introduit un nouvel outil de développement pour les microcontrôleurs AVR d'Atmel.

Le EasyAVR 5A, avec supports pour les AVR en boîtier DIP de 8, 14, 20, 28 et 40 broches, est un environnement simple à utiliser qui permet aux concepteurs d'essayer toute type de montage. La nouvelle carte est fournie avec tout ce dont on a besoin pour apprendre, expérimenter, concevoir et programmer les microcontrôleurs AVR. Le EasyAVR 5A comprend un contrôleur pour écran tactile qui permet d'ajouter facilement au montage une interface homme machine (IHM).

Comme son prédécesseur, la carte possède un programmeur



embarqué ultra rapide USB 2.0. La carte a beaucoup de possibilités et d'outils. Elle possède, entre autres, des boutons pous-

soir permettant de modifier le niveau logique de chaque broche individuellement (chaque broche a, aux choix, une résistance de

rappel vers VCC ou la masse), des LED pour visualiser l'état de chaque broche, une interface pour afficheur LCD alphanumérique et graphique, un connecteur JTAG, un port RS-232, un support pour un capteur de température DS18B20 ou encore un afficheur LED 7 segments. La carte peut être alimentée par le port USB ou par une alimentation externe et possède un support pour carte MMC/SD et des connecteurs IDC10 pour rajouter des extensions au système.

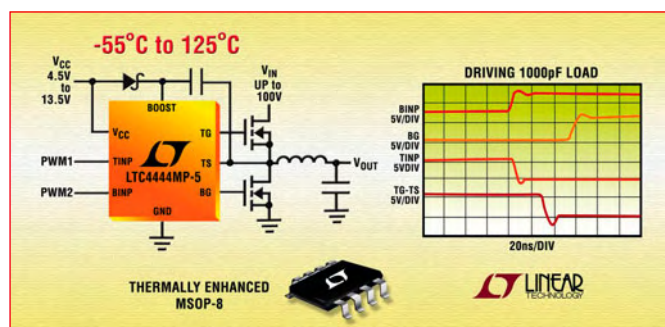
Toutes les options de la carte sont illustrées par des exemples de programmation pour les compilateurs mikroC PRO, mikroPascal et mikroBasic pour AVR.

(081045-9)

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

## Pilote de MOSFET canal N, 100 V, synchrone et rapide

Linear Technology annonce une nouvelle version du LTC4444-5, de grande fiabilité de classe MP, un pilote de MOSFET, rapide, synchrone, à tension d'entrée élevée (100 V), conçu pour piloter les MOSFET canal N de puissance haut et bas, dans des topologies de convertisseurs à redressement synchrone. Ce pilote, associé à des MOSFET de puissance et à un des nombreux contrôleurs DC/DC de Linear Technology, constitue un régulateur synchrone complet, de rendement élevé. Le LTC4444MP-5 fonctionne et est testé pour une température



de jonction, de -55°C à 125°C, alors que la gamme de températures de la version de classe I est de -40°C à 85°C.

Ce pilote puissant peut fournir jusqu'à 1,4 A, avec une impédance de tirage à la masse de 1,5 Ω pour piloter le MOSFET

supérieur et jusqu'à 1,75 A avec une impédance de 0,75 Ω pour le MOSFET inférieur, ce qui le rend idéal pour le pilotage de transistors de forte intensité et de capacité de porte élevée. Le LTC4444MP-5 peut également piloter plusieurs transistors de puissance en parallèle, pour des applications de plus forte intensité. Les pertes de commutation sont réduites grâce au temps de montée rapide de 8 ns, et de descente de 5 ns du MOSFET supérieur, et au temps de montée de 6 ns et de descente de 3 ns du MOSFET inférieur, lorsque la charge pré-



sente une capacité de 1000 pF. Une protection est intégrée pour réduire le temps mort afin d'empêcher les deux MOSFET, supérieur et inférieur, d'être conducteurs simultanément.

Le LTC4444MP-5 est configuré pour deux entrées d'alimentations indépendantes. Le signal logique d'entrée côté haut est amené au niveau de l'alimentation de tirage de façon interne,

qui peut atteindre 114 V au-dessus de la masse. Bien plus, ce composant pilote les grilles des deux MOSFET, supérieur et inférieur, sur une gamme de tensions de 4,5 V à 13,5 V.

Le LTC4444MP-5 est présenté en boîtier MSOP-8, à performances thermiques renforcées.

(081045-4)

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

## Pinces multimètres TRMS

Combinant les fonctionnalités d'un multimètre TRMS complet et d'une pince ampèremétrique, les MX 675 & MX 670 de METRIX possèdent des fonctions essentielles nécessaires à tout électricien. Elles couvrent notamment les besoins de la maintenance électrique et électrotechnique, des installateurs et des distributeurs d'électricité.

Dotées de bi-afficheur, elles permettent de visualiser deux mesures simultanément, telles que la tension et l'intensité. Côté sécurité, elles répondent aux à la norme IEC 61010, 600 V CAT IV & 1000 V CAT III.

La forme de ces pinces a été étudiée afin d'offrir une excellente maniabilité lors des mesures. La sélection des mesures s'effectue simplement via le commutateur de façon nette et précise.

Côté affichage, ces pinces offrent un atout supplémentaire : un bi-

afficheur de 10 000 points ! L'utilisateur peut visualiser instantanément les valeurs de deux mesures distinctes, telles que l'intensité et la tension. Un rétroéclairage bleu offre une excellente lisibilité quelles que soient les conditions.

L'utilisateur dispose d'outils robustes et fiables qui couvrent la totalité des applications de mesures électriques auxquelles il est confronté sur le terrain.

Côté multimètre, les MX 675 et MX 670 mesurent les tensions (CA et CC), les résistances et permettent de contrôler les diodes ou la continuité. En outre, elles déterminent aussi la fréquence des tensions et courants. La mesure de température est disponible avec le capteur thermocouple K, couvrant une plage allant de -40°C à +1200°C.

Les MX 675 et MX 670 sont équipées d'une touche HOLD

pour geler l'affichage et faciliter la lecture, quelles que soient les conditions d'accès ou de luminosité. À cela elles ajoutent une fonction « Peak », capable de capturer des signaux de 1 ms, les fonctions « Min », « Max », et une fonction « Zero » pour des mesures différentielles. Cette dernière fonction sera d'ailleurs particulièrement appréciée pour compenser la résistance des cordons, et lors des mesures d'intensités continues.

Leurs performances techniques ne sont pas leur seul point fort, loin de là. Leur prix, aussi, constitue un de leurs atouts ; d'autant qu'il comprend, outre la pince, les cordons de mesure, le capteur thermocouple K, les piles, et une sacoche de transport.

(081045-2)

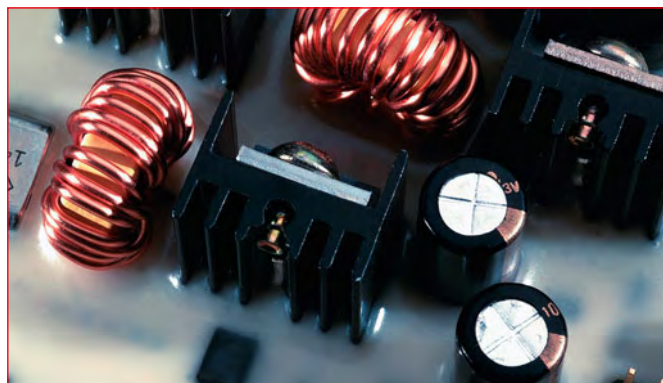
[www.chauvin-arnoux.com](http://www.chauvin-arnoux.com)



## Nouvelles résines époxy souples mono et bicomposantes

Syneo enrichi sa gamme de résines époxy souples mono et bicomposantes avec trois nouvelles références : Monopox AD 288, Katiobond KB 554 et trois versions de Duopox (AD 821, AD 850 et CR 804). Ces résines dernière génération sont idéales pour les collages mécaniques et électroniques.

Les références Duopox, résines d'enrobage époxydes bicomposantes sont utilisées pour enrober des composants électroniques. Elles sont exemptes de nonylphénol. Leur durcissement à basse température préserve les éléments électroniques sensibles. Ces adhésifs présentent une fiabilité élevée surtout avec des composants volumineux ou sen-



sibles à la température ; le collage est stable à long terme même dans des conditions climatiques extrêmes. Il protège les composants électroniques contre les vibrations, la poussière et l'humidi-

té. Ces nouvelles résines Duopox présentent de bonnes propriétés de remplissage et grâce à leur très basse viscosité elles éliminent quasiment toutes les bulles dans le moulage.

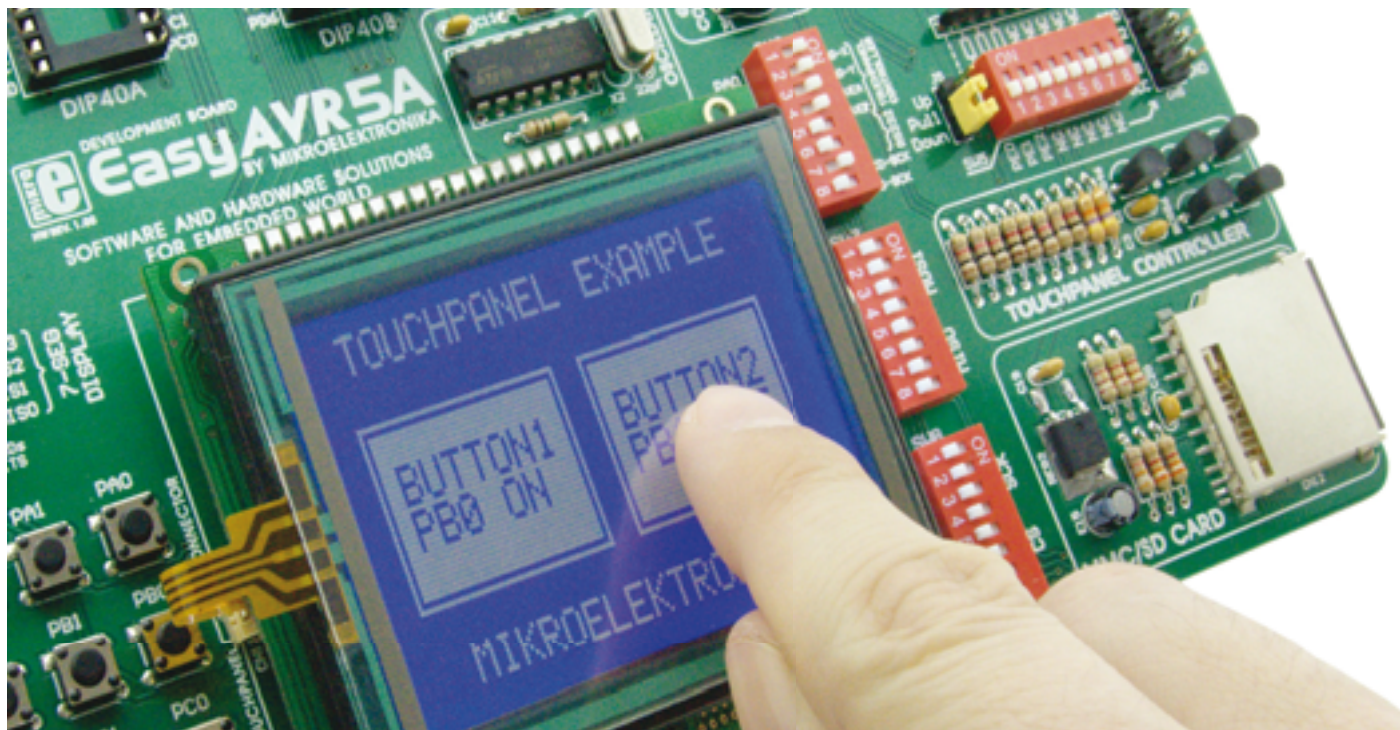
La résine monocomposante Monopox, à durcissement en température comprise entre 100 et 150°C, possède une elongation à la rupture de 200%. Enfin un époxy souple pour le collage de composants avec compensation des contraintes.

La résine époxy monocomposante Katiobond est photo activable aux UV ou à la lumière visible sur plage de longueur d'onde de 400 à 550 nm. Les temps de cycle courts adaptés à la production en série, la vaste plage d'élasticité et la grande pureté ionique de ce type de colle respectent les exigences du secteur de la microélectronique.

(081045-5)

[www.syneo.net](http://www.syneo.net)

# Bon. Maintenant il vous faut un... écran tactile



Par Dusan Mihajlovic  
mikroElektronika Département Matériel

Voulez-vous une interface intuitive et simple pour votre nouveau projet ? Si la réponse est OUI, un LCD graphique avec dalle tactile est le meilleur choix, parce que ensemble ils forment un écran tactile (LCD graphique + dalle tactile = écran tactile). Ainsi, avec un poignier de composants, vous pouvez réaliser un appareil attractif et facile à utiliser.

Qu'est qu'un écran tactile ? Un écran tactile consiste en une dalle tactile autocollante fixée sur un LCD graphique. Elle est si sensible que même une légère pression suffit pour faire varier le signal de sortie. Il existe plusieurs types de dalles tactiles. Le plus simple est la dalle tactile résistive dont nous allons vous parler ici.

## Principe de fonctionnement

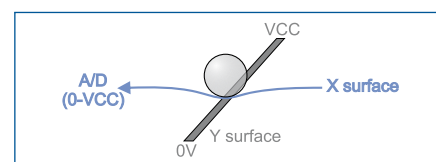
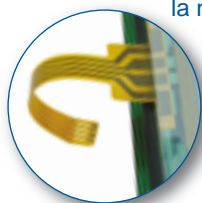
Une dalle tactile résistive consiste en deux films transparents en « sandwich » avec une couche résistive sur les cotés intérieures des films. La résistance de ces couches ne dépasse en général pas le 1 k $\Omega$ . Les autres cotés des films ont des contacts pour connecter une nappe. L'algorithme pour déterminer la position de la pression se décompose en deux étapes. La première étape est la détermination de la coordonnée X et la deuxième est la détermination de la coordonnée Y de la position. Pour déterminer la coordonnée X, il faut connecter le contact de gauche de la surface X à la masse et le contact de droit à l'alimentation. Un diviseur de tension se crée quand on appuie sur l'écran tactile. La valeur du diviseur est disponible sur le contact inférieur de la surface Y. La tension va de

0 V jusqu'à la valeur de l'alimentation et dépend de la coordonnée X. Si le point de pression est plus près du contact de gauche de la surface X, la tension sera plus près de 0 V. Pour déterminer la coordonnée Y, il faut connecter le contact inférieur de la surface Y à la masse et le contact supérieur à l'alimentation et prélever la tension sur le contact de gauche de la surface X.

## Connecter un microcontrôleur

Pour connecter la dalle tactile à un microcontrôleur, il est nécessaire de réaliser un pilote de dalle tactile. Ce montage permet au microcontrôleur de relier les bons contacts de la dalle tactile à la masse et à l'alimentation (comme décrit plus haut) pour ensuite déterminer les coordonnées X et Y (voir schéma 1). Le contact inférieur de la surface Y et le contact de gauche de la surface X

Détail de la nappe



Détermination de la coordonnée X

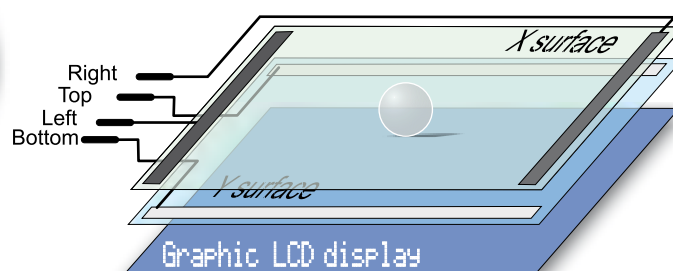
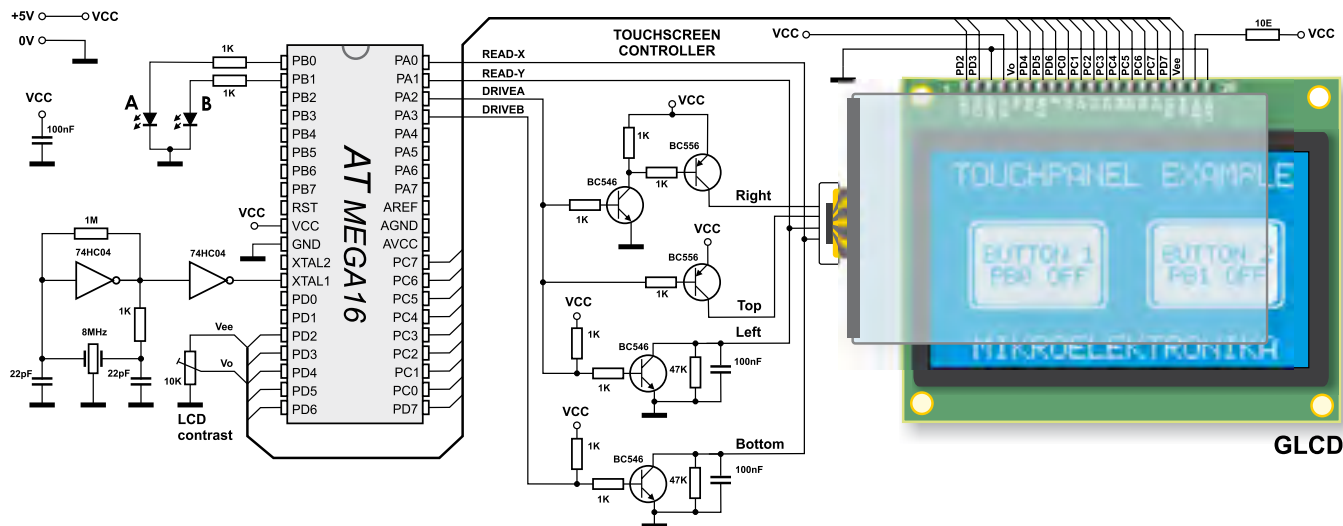


Figure 1. Structure interne de la dalle tactile



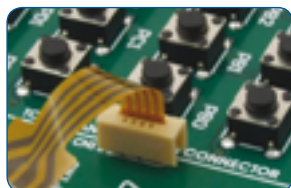


sont reliés au convertisseur A/N du microcontrôleur. Les coordonnées X et Y sont déterminés en mesurant les tensions présentes sur ces contacts. Le logiciel doit être capable d'afficher un menu sur le LCD graphique, de contrôler le pilote de la dalle tactile et de lire les valeurs du convertisseur A/N, qui représentent les coordonnées X et Y de la position.

Une fois les coordonnées connues, il est possible de décider ce que le microcontrôleur doit faire. Regardez l'exemple 1. Il montre comment activer et désactiver avec un afficheur et une dalle tactile deux broches du microcontrôleur connectées aux deux LED A et B.



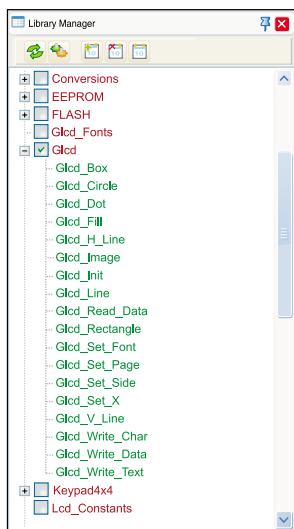
Le connecteur de nappe embarqué avant...



... et après la connexion de l'écran tactile

Vu que la surface de la dalle tactile est légèrement plus grande que la surface du LCD, au cas vous auriez besoin d'une précision plus importante pour déterminer les coordonnées, il est nécessaire de calibrer la dalle tactile avec le logiciel.

## Fonctions utilisées dans ce programme



ADC\_Read() Lire valeur analogique  
Delay\_ms() Retard

<b>Glcd_Box()</b>	Dessiner un rectangle opaque*
<b>Glcd_Circle()</b>	Dessiner un cercle
<b>Glcd_Dot()</b>	Afficher un point
<b>Glcd_Fill()</b>	Effacer/Remplir l'écran*
<b>Glcd_H_Line()</b>	dessiner une ligne horizontale
<b>Glcd_Image()</b>	Importer une image
<b>Glcd_Init()</b>	Initialiser l'afficheur LCD*
<b>Glcd_Line()</b>	Dessiner une ligne
<b>Glcd_Read_Data()</b>	Lire des données du LCD
<b>Glcd_Rectangle()</b>	Dessiner un rectangle*
<b>Glcd_Set_Font()</b>	Choisir une police*
<b>Glcd_Set_Page()</b>	Choisir une page
<b>Glcd_Set_Side()</b>	Choisir la coté de l'afficheur
<b>Glcd_Set_X()</b>	Déterminer la coordonnée X
<b>Glcd_V_Line()</b>	Dessiner une ligne verticale
<b>Glcd_Write_Char()</b>	Écrire un caractère
<b>Glcd_Write_Data()</b>	Écrire des données
<b>Glcd_Write_Text()</b>	Écrire du texte*

\* les fonctions de la librairie Glcd

\* les fonctions de la librairie Glcd

**mikroC PRO pour AVR®** Éditeur de librairie avec des librairies prêtes à l'emploi, comme Ethernet, CAN, SD/MMC, etc.

### Exemple 1: Programme montrant le fonctionnement de l'écran tactile

```

// GLCD module connections
char GLCD_DataPort at PORTC;

sbit GLCD_CS1 at PORTD.B2;
sbit GLCD_CS2 at PORTD.B3;
sbit GLCD_RS at PORTD.B4;
sbit GLCD_RW at PORTD.B5;
sbit GLCD_EN at PORTD.B6;
sbit GLCD_RST at PORTD.B7;

char GLCD_DataPort_Direction at DDRC;

sbit GLCD_CS1_Direction at DDRD.B2;
sbit GLCD_CS2_Direction at DDRD.B3;
sbit GLCD_RS_Direction at DDRD.B4;
sbit GLCD_RW_Direction at DDRD.B5;
sbit GLCD_EN_Direction at DDRD.B6;
sbit GLCD_RST_Direction at DDRD.B7;

// End Glcd module connections

sbit DRIVE_A at PORTA.B2;
sbit DRIVE_B at PORTA.B3;

sbit DRIVE_A_Direction at DDRA.B2;
sbit DRIVE_B_Direction at DDRAB3;

// Touch Panel module connections
// End Touch Panel module connections

long x_coord, y_coord, x_coord128, y_coord64;

// scaled x-y position

unsigned int GetX() {

    //reading X

    DRIVE_A = 1;
    // DRIVEA = 1 (LEFT drive on, RIGHT drive on, TOP drive off)
    DRIVE_B = 0;
    // DRIBEB = 0 (BOTTOM drive off)
    Delay_ms(5);
    return ADC_Read(0);
    // READ-X (BOTTOM)

}

unsigned int GetY() {

    //reading Y

    DRIVE_A = 0;
    // DRIVEA = 0 (LEFT drive off, RIGHT drive off, TOP drive on)
    DRIVE_B = 1;
    // DRIBEB = 1 (BOTTOM drive on)
    Delay_ms(5);
    return ADC_Read(1);
    // READ-X (LEFT)

}

void main() {

    DRIVE_A_Direction = 1;
    // Set DRIVE_A pin as output
    DRIVE_B_Direction = 1;
    // Set DRIVE_B pin as output
    PORTB.B0 = 0;
    PORTB.B1 = 0;
    // Set PB0 pin as output (Default value 0)
    PORTB.B1 = 0;
    PORTB.B1 = 0;
    // Set PB1 pin as output (Default value 0)

    Glcd_Init();
    // Initialize GLCD
    Glcd_Fill(0);
    // Clear GLCD
    Glcd_Set_Font(font5x7, 5, 7, 32);
    // Choose font,

    Glcd_Write_Text("TOUCHPANEL EXAMPLE","10,0,1);
    Glcd_Write_Text("MIKROELEKTRONIKA","17,7,1);

    Glcd_Rectangle(8,16,60,48,1);
    //Display Buttons on GLCD:
    Glcd_Rectangle(68,16,120,48,1);
    Glcd_Box(10,18,58,46,1);
    Glcd_Box(70,18,118,46,1);
    Glcd_Write_Text("BUTTON1","14,3,0);
    Glcd_Write_Text("PB0 OFF","14,4,0);
    Glcd_Write_Text("BUTTON2","74,3,0);
    Glcd_Write_Text("PB1 OFF","74,4,0);

    while (1) {

        // read X-Y and convert it to 128x64 space

        x_coord = GetX();
        y_coord = GetY();
        x_coord128 = (x_coord * 128) / 1024;
        y_coord64 = 64 - ((y_coord * 64) / 1024);

        //if BUTTON1 is selected

        if ((x_coord128 >= 10) && (x_coord128 <= 58) && (y_coord64 >= 18) && (y_coord64 <= 46)) {
            if(PORTB.B0 == 0) {
                PORTB.B0 = 1;
                Glcd_Write_Text("PB0 ON","14,4,0);
            }
            else {
                PORTB.B0 = 0;
                Glcd_Write_Text("PB0 OFF","14,4,0);
            }
        }

        //if BUTTON2 is selected

        if ((x_coord128 >= 70) && (x_coord128 <= 118) && (y_coord64 >= 18) && (y_coord64 <= 46)) {
            if(PORTB.B1 == 0) {
                PORTB.B1 = 1;
                Glcd_Write_Text("PB1 ON","74,4,0);
            }
            else {
                PORTB.B1 = 0;
                Glcd_Write_Text("PB1 OFF","74,4,0);
            }
        }

        Delay_ms(100);
    }
}

```

Écrit dans le  
compilateur  
mikroC  
for

Écrit dans le  
compilateur

**mikroC PRO  
for AVR**

**NOTE:**

Les codes source pour microcontrôleurs AVR de cet exemple écrit en C, BASIC et PASCAL ainsi que tous les programmes écrits pour les microcontrôleurs PIC® et dsPIC® sont disponibles sur notre site Internet [www.mikroe.com/en/article/](http://www.mikroe.com/en/article/)

# La CAO gratuite outils de conception en ligne

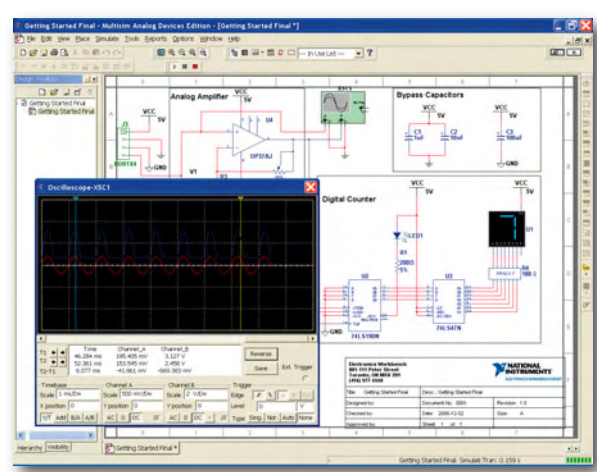
Clemens Valens

Plusieurs fabricants de composants électroniques proposent maintenant des outils de conception en ligne ou à télécharger gratuitement. Nous nous sommes enregistrés - étape souvent obligatoire - sur une dizaine de sites Internet pour voir ce que l'on nous propose et à quoi tout cela ressemble.

## Analog Devices

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

Toute une panoplie d'outils de conception interactive dont le nom commence avec **ADIsim** est proposée sur le site d'Analog Devices. Il y a par exemple **ADIsim-Power**, un simulateur pour alimentations, ou convertisseurs CC/CC pour être plus exact. Analog Devices offre aussi un calculateur de pont de résistances, une aide pour dimensionner des amplificateurs à diodes photo et un outil pour calculer les filtres analogiques. Il y a également des outils à télécharger gratuitement comme **SRD Design Studio**, un outil pour simuler les systèmes radio à courte distance ou encore **Multisim pour Analog Devices**, un monstre de 183 Mo (V 10.0.1). Après lancement de ce programme nous avons chargé le projet *Getting Started* dont on ne parle nulle part, mais qui se trouve au fond du répertoire où le programme a été installé. Cet exemple montre un compteur numérique à un chiffre avec comme horloge la sortie d'un amplificateur qui amplifie un signal de 1 kHz. Quand on appuie sur le bouton *Run*, en haut à droite de l'écran, la simulation démarre. Un double clic sur le pictogramme de

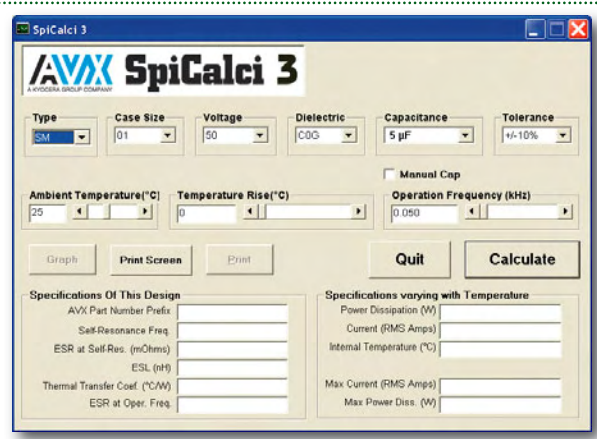


l'oscilloscope ouvre une fenêtre montrant un oscilloscope numérique à deux voies. C'est très joli. À part des modèles de certains composants courants, Multisim ne connaît que des composants d'Analog Devices et il n'est pas possible de les modifier ou d'en créer de nouveaux.

## AVX

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

AVX propose plusieurs outils pour sélectionner ses condensateurs. Nous avons essayé **SpiCalci 3.0**, un outil pour dimensionner des condensateurs pour alimentations à découpage. Malheureusement, l'outil est dépourvu d'aide et il faut bien connaître les condensateurs d'AVX. Après la sélection d'un type de condensateur, sa tension de travail, le diélectrique, etc. le programme montre la fréquence de résonance du condensateur et la dissipation pour une température choisie. Il est aussi possible de visualiser la courbe de l'impédance et l'ESR.

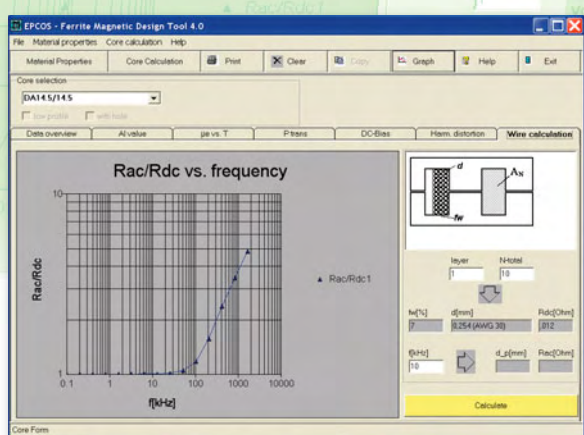




## Epcos

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

Ce fabricant de composants passifs propose plusieurs outils dont **AlCap** pour dimensionner des condensateurs électrolytiques en aluminium ou encore le **Ferrite Magnetic Design Tool**. Cet outil sert à évaluer les matériaux magnétiques d'Epcos et à faire certains calculs liés aux inductances et bobines.

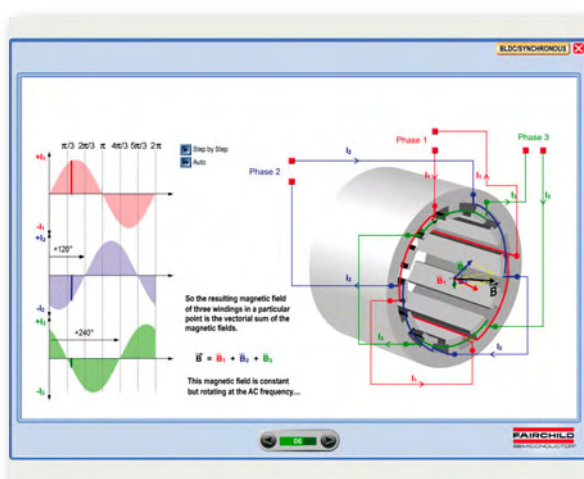


## Fairchild

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

**FETBench** est l'outil de conception en ligne de Fairchild. L'outil consiste en trois parties :

- **Device Analysis** permet l'évaluation d'un composant avec des graphes sur mesure ;
  - **Application Analysis** est un simulateur d'alimentations à découpage ;
  - **Thermal Analysis** permet de spécifier une platine et d'y placer des composants actifs de chez Fairchild. Ensuite l'outil montre un diagramme coloré avec la distribution des températures sur la surface de la platine.
- Mais il y a mieux chez Fairchild : **SPM Tool**, un très beau logiciel qui non seulement est capable de simuler les pilotes de moteurs, mais qui contient aussi toute la théorie des moteurs sous forme d'animations interactives. À ne pas manquer ! Avant de pouvoir télécharger le logiciel, un nom d'utilisateur et un mot de passe sont demandés. Ce sont les mêmes que ceux utilisés pendant l'enregistrement.



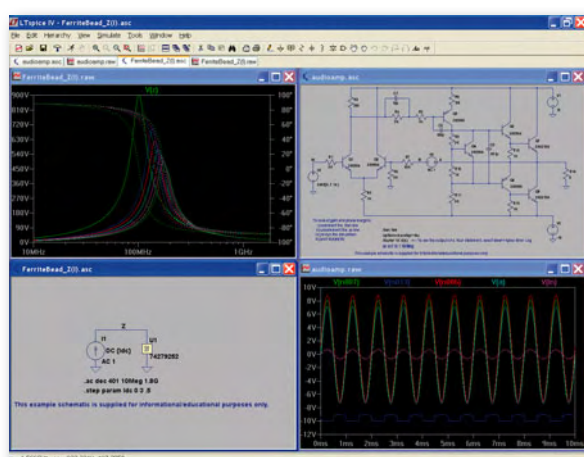
Pareil pour le **PFC Toolkit**, un outil pour explorer les techniques de correction de déphasage.

## Linear Technologie

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

**LTSpice IV** est actuellement le nom du simulateur SPICE de Linear Technology. L'outil a été lancé il y a quelques années sous le nom **SwitcherCAD** et servait surtout à simuler des alimentations à découpage. Les utilisateurs ont rapidement découvert que l'outil est en fait un SPICE amélioré qui se prête parfaitement à d'autres simulations. Utiliser LTSpice est assez facile et intuitif. Un schéma est vite dessiné et sa simulation est vite lancée. Le graphisme n'est peut-être pas au top, mais l'outil est très puissant et mis à jour très régulièrement.

À part de LTSpice, le site propose également **FilterCAD** et **BodeCAD** et quelques outils pour évaluer certains composants de Linear Technology.



## Maxim

www.maxim-ic.com

Maxim ne propose pas d'outils de conception en ligne très sophistiqués, mais il y a quand même une vingtaine de calculateurs utiles disponibles. Un certain nombre est dédié à un composant Maxim, mais pas tous et il y en a plusieurs qui peuvent vous rendre service. L'intéressant est que Maxim donne des références et des explications détaillées qui permettent de creuser la théorie mise en pratique par le calculateur.

Le **Power-Supply Cookbook** permet pour chaque régulateur dans le catalogue de Maxim de télécharger un fichier PDF contenant un schéma électrique et la liste des composants pour plusieurs configurations de la puce. Il est aussi possible de télécharger tous les fichiers en un coup sous forme de PDF de 9 Mo, ce qui fait gagner du temps. C'est un peu comme les *data-book* de l'époque avec plein de idées de montages.

**MAXIM** 25 Years of Engineering Success

ENGLISH 简体中文 日本語 繁體中文

WHAT'S NEW PRODUCTS SOLUTIONS DESIGN APPNOTES SUPPORT BUY COMPANY MEMBERS

Maxim > Tools, Models, and Software > Calculators

### Data Conversion Calculator

Tools, Models, and Software Notice

SPICE Models

IBIS Models

BSDL Files

CAD Files

Buttons Software

Power-Supply Cookbook

Wireless RF S-Parameters

Reference Designs (Gerber Files)

EV Kit Software

Calculators

Other Software and Tools

Enter ENOB: 5 bits **Convert** SINAD: 37.88 dB

Enter SINAD: 40 dB **Convert** ENOB: 5.4 bits

Enter SNR: 80 dB **Convert** Resolution (N): 14 bits

Enter Resolution (N): 14 bits **Convert** SNR: 86.0 dB

Enter SNR plus: 80 dB **Calculate** SINAD: 40.00 dB

Enter THD: -40 dB

Below is a list that describes what each conversion on the calculator means.

**decibel (dB)** A unit of relative amplitude defined on a logarithmic scale. For voltage values, dB is given by  $20\log(V_u/V_r)$ . For power, it is  $10\log(P_u/P_r)$ . When dB is referenced to a carrier signal, it is termed dBc; likewise, dB referenced to 1mW is termed dBm. For dBm, the load resistance must be known for the specification to determine the voltage or current equivalence (that is, 1mW into 50Ω).

**effective number of bits (ENOB)** The measured performance (in bits) of an analog-to-digital converter (ADC) with respect to the input frequency ( $f_{in}$ ). As  $f_{in}$  increases, overall noise (particularly the distortion components) also increases, thereby reducing the ENOB and SINAD. See also signal-to-noise and distortion ratio (SINAD). ENOB is related to SINAD by the following equation:

$$ENOB = \frac{SINAD - 1.76}{6.02}$$

**resolution** When an analog signal is digitized, it is represented by a finite number of discrete voltage levels. The resolution is the number of discrete levels that are used to represent the signal. To more accurately replicate the analog signal, the resolution must be increased. Resolution is usually defined in bits. Using converters with higher resolutions will reduce the quantization error.

**RMS** See root mean square (RMS).

**root mean square (RMS)** The effective value or effective DC value that an AC signal represents. For a sine wave, the RMS value is 0.707 times the peak value, or 0.354 times the peak-to-peak value.

**SFDR** See spurious-free dynamic range (SFDR).

**signal-to-noise and distortion ratio (SINAD)** The RMS value of the sine wave  $f_{in}$  (input sine wave for an ADC, reconstructed output sine wave for a DAC/DAC) to the RMS value of the noise of the converter from DC to the Nyquist frequency, including harmonic content. It is typically expressed in decibels. See also root mean square (RMS) and total harmonic distortion.

$$SINAD = 20\log_{10} \frac{\text{Signal (volts, RMS)}}{\text{Noise + Harmonics (volts, RMS)}}$$

**signal-to-noise ratio (SNR)** The RMS value of the sine wave  $f_{in}$  (input sine wave for an ADC, reconstructed output sine wave for a DAC) to the RMS value of the noise of the converter from DC to Nyquist frequency, excluding noise at DC and harmonic distortion content. It is typically expressed in decibels. See also root mean square (RMS).

$$SNR = 20\log_{10} \frac{\text{Signal (volts, RMS)}}{\text{Noise (volts, RMS)}}$$

The ideal theoretical minimum conversion noise is caused by quantization noise error only and results directly from the data converters resolution

$$(N): SNR = (6.02N + 1.76)dB$$

**spurious-free dynamic range (SFDR)** The ratio of the RMS value of the sine wave  $f_{in}$  (input sine wave for an ADC, reconstructed output sine wave for a DAC) to the RMS value of the peak spur observed in the frequency domain. It is typically expressed in decibels. SFDR is important in certain communication applications that require maximizing the dynamic range of the converter.

**total harmonic distortion (THD)** The RMS value of the distortion appearing at multiples (harmonics) of the input (or output for a DAC) frequency to the RMS value of the input (or output) sine wave. Only harmonics within the Nyquist limit are included in the measurement. It is typically expressed in decibels:

$$THD = 20\log \frac{\sqrt{V_2^2 + V_3^2 + \dots + V_N^2}}{V_1}$$

**FFT PLOT** ( $f_{in} = 7.2MHz$ )

## Microchip

www.microchip.com

**Mindi** est le simulateur analogique de Microchip basé sur (X)SPICE et développé par SIMetrix. Mindi est fourni avec beaucoup de modèles de composants, nous en avons compté environ 4500, ce qui rend la saisie de schéma très confortable. Après avoir sélectionné un composant, il est possible de modifier ses paramètres, ce qui se résume dans la plupart des cas à choisir un des modèles fournis.

Un autre outil de conception gratuit de Microchip est **FilterLab2**. Cet outil permet de simuler des filtres de type Chebychev, Butterworth et Bessel. Dans un environnement assez convivial on choisit son filtre et le logiciel dessine les courbes caractéristiques. Il propose également une implémentation du filtre, basée sur des sections Sallen & Key ou MFB (à contre-réaction multiple). Plusieurs façons de concevoir le filtre sont proposées. Quand on est satisfait des performances théoriques du filtre on dispose toute de suite d'un schéma électrique et d'un modèle SPICE.

## Murata

www.murata.com

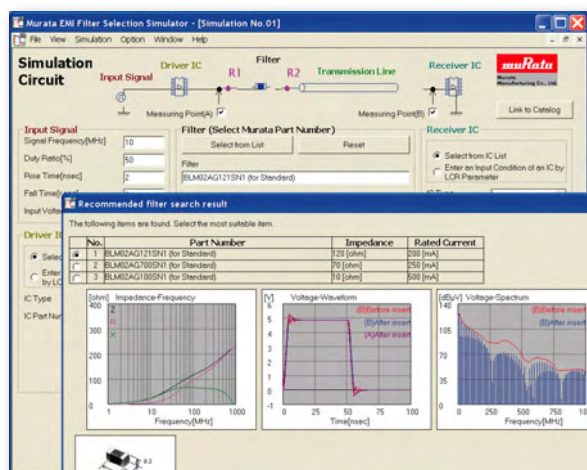
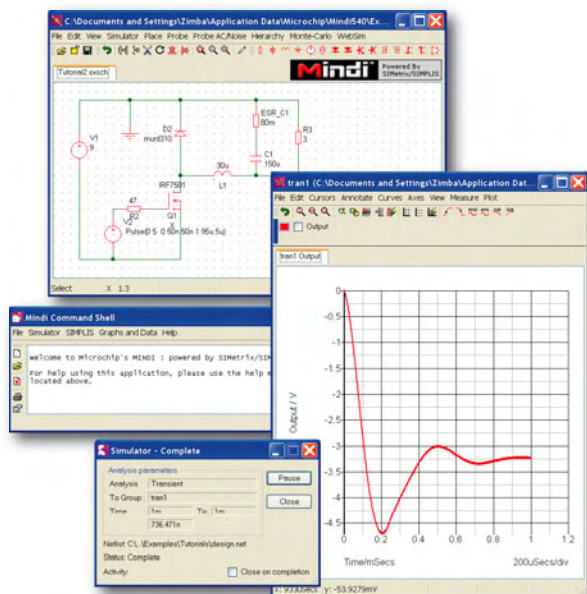
Chez Murata cinq outils de conception sont à télécharger, pour des condensateurs, des filtres CEM et des thermistances. Nous avons essayé le simulateur de filtre CEM, le **Murata EMI Filter Selection Simulator** (version 3.6.0). Cet outil permet de dimensionner des filtres CEM pour des lignes de transmission. Plusieurs configurations sont supportées et l'utilisateur peut paramétrer le circuit en détail. Une fois les paramètres saisis, un clic sur le bouton *Start Simulation* suffit pour avoir accès aux résultats sous forme de diagrammes. Bref, un bel outil et facile à utiliser.

## National Semiconductor

www.national.com

National Semiconductor propose depuis quelques années déjà son simulateur **WEBENCH**, d'abord pour simuler les alimentations, mais aujourd'hui l'outil aide aussi avec la conception des filtres, des boucles à phase asservie, des pilotes de LED et même des amplificateurs audio. En moins de dix minutes nous avons par exemple conçu et simulé un CAN avec filtre antirepliement. Pensez à automatiser les fenêtres intruses (*pop-up* en anglais), sinon vous ne voyez pas tous les résultats des simulations.

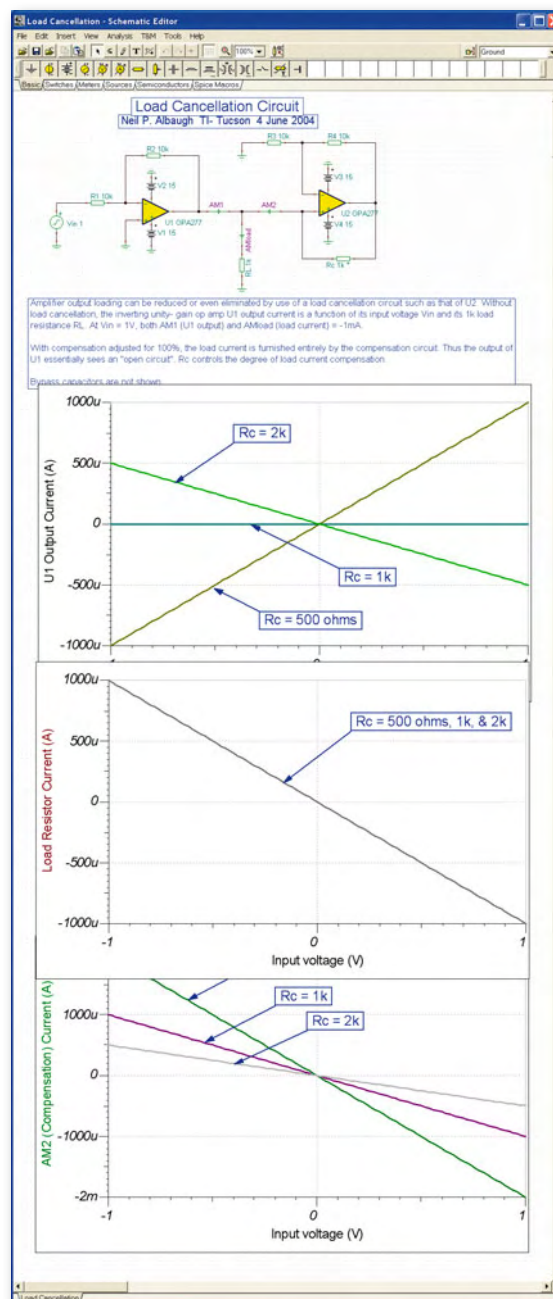




## Texas Instruments

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

**TINA-TI** est le simulateur gratuit à télécharger chez Texas Instruments et capable d'utiliser des modèles SPICE. TINA-TI est une partie de TINA tout court, un outil de CAO complète avec saisie de schéma, dessin de circuit imprimé et simulateur.



**SwitcherPro** est le simulateur d'alimentations à découpage de Texas Instruments. Depuis quelques mois ce simulateur est aussi disponible en ligne. Comme d'habitude il faut d'abord s'enregistrer avant de pouvoir l'utiliser. Une fois connecté au simulateur, il est possible de démarrer une simulation de deux façons : à partir d'un composant ou à partir de spécifications.

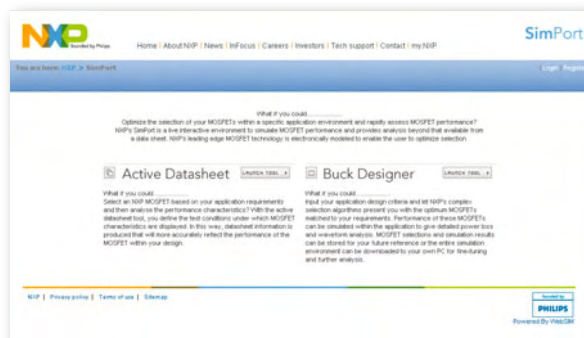
## NXP

www.nxp.com

**SimPort** regroupe les simulateurs en ligne de NXP, développés, comme Minda de Microchip, par SIMetrix/SIMPLIS. En fait, il n'y en a que deux : **Active Datasheet** et **Buck Designer**. Avant qu'il soit possible de les utiliser, il faut, comme partout, s'enregistrer. Rien de spécial, me disiez-vous, mais il faut avoir des bons yeux. Si vous n'êtes pas encore enregistré, il faut suivre le lien à droite en haut de la page qui permet de choisir entre les deux simulateurs.

**Active Datasheet** permet de sélectionner un MOSFET de NXP et de générer vous-même les courbes que l'on trouve habituellement dans une fiche technique. La différence avec une fiche technique est que c'est à vous de spécifier les valeurs pour lesquelles les courbes sont générées.

**Buck Designer** est un outil pour simuler un abaisseur de



tension. L'outil vous propose un schéma fonctionnel et un schéma électrique détaillé et il génère des diagrammes dans lesquels il est possible d'agrandir des zones et de mesurer certains paramètres.

## STMicroelectronics

www.st.com

Des outils à télécharger et des outils en ligne chez ST. Pour les outils en ligne, un simulateur d'amplificateur audio et un simulateur d'alimentations à découpage, il faut d'abord installer l'environnement *runtime* de Java (JRE). Ensuite les outils fonctionnent... plus ou moins, du moins sur l'ordinateur de test.

Par contre, **ST Lighting Designer** est un outil Flash pour dimensionner des néons qui vaut le détour. Un système à dû passer du temps sur l'interface graphique - on aime ou on n'aime pas - et l'outil fonctionne à merveille. Une fois que l'on a compris qu'il est impossible d'avoir une solution avec toutes les options activées, l'outil sort le schéma, la liste des composants et des graphes. Ensuite il est possible de peaufiner pleins de paramètres.

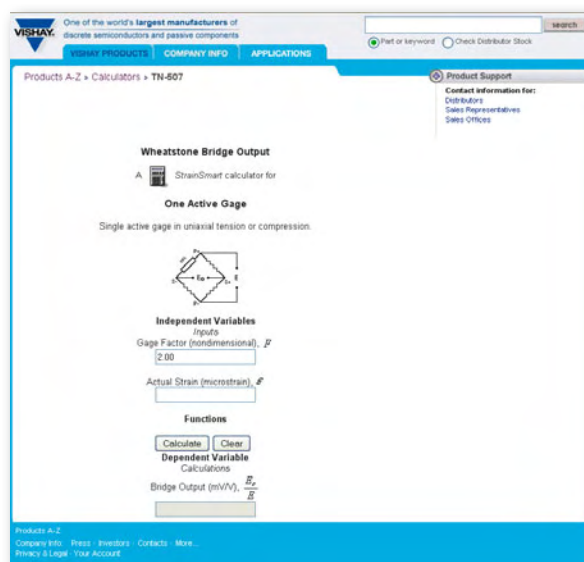


## Vishay

www.vishay.com

**ThermaSim** est un simulateur thermique pour circuit imprimé. Son usage n'est pas très intuitif, mais après quelques efforts on arrive à faire avaler nos paramètres par l'outil, après quoi il envoie les résultats par courriel, ce qui est assez original, mais pas très rapide. Et quand le message arrive enfin, on découvre qu'il y a eu un problème et que la simulation a échoué. Merci de la corriger et de la relancer, dit le mél. Ouais, t'as raison...

En dehors ThermaSim, Vishay propose également quelques calculateurs basiques, comme par exemple le calculateur **TN-507** pour jauges de déformation.





Deux outils sont disponibles sur le site de Würth : **WE Inductor Selector 1.0** et **WE Flex Designer 2.0**. Ce dernier sert à sélectionner un transformateur pour alimentation à découpage de type *flyback*. L'outil n'est pas joli joli, du moins sur le PC de test, puisque toutes les cases et boutons étaient décalés, néanmoins un simple clic sur le bouton *Calculate* suffisait pour générer une référence de transformateur.



## Enfin

Nous sommes en 2009. Tous les fabricants de composants électroniques proposent des outils de conception gratuits... Tous ? Non ! Quelques fabricants résistent encore et toujours à cette tendance. Sur les sites de Freescale, Infineon, Hitachi, Toshiba et NEC par exemple nous n'avons rien trouvé du tout qui mérite d'être mentionné ici.

Malgré le nombre d'outils de conception disponibles sur le web, on peut s'interroger sur leur utilité. Certes, certains sont bien faits, certains sont jolis, mais est-ce que l'on a vraiment besoin d'autant de simulateurs d'alimentations à découpage ?

(080874-1)

Publicité

**PCB-POOL®**  
**Prix très concurrentiels pour les PCB prototypes**  
**1 EUROCARD**  
 + **Outillage**  
 + **Photoplots**  
 + **TVA**

**€49,-**

\*Ce prix ne comprend pas les frais de port.

**Appel Gratuit**  
 0800-903 230

**ROHS/WEEE conforme**

**Sans Plomb**

**Beta**

**Calculez votre devis immédiatement en ligne**  
**Outillage /Set-up inclus**  
**Aucun montant minimum**  
**Livraison ponctuelle garantie**  
**Garantie de qualité ISO 9001**

**WWW.PCB-POOL.COM**

**Schaeffer** AG

## 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: 32,50 € majoré de la TVA/ des frais d'envoi

Schaeffer AG · Nahmitzer Damm 32 · D-12277 Berlin · Tel +49 (0)30 805 86 95-30  
 Fax +49 (0)30 805 86 95-33 · Web info.fr@schaeffer-ag.de · www.schaeffer-ag.de

# Voir avec un recul suffisant

## Triangulation optique avec l'ATM18

Udo Jürß et Wolfgang Rudolph



**Certains de nous ont le compas dans l'oeil et la plupart sont en mesure d'estimer, sans instruments, les distances qui les séparent des objets, assez en tout cas, pour ne pas les heurter. Comment faire pour qu'un robot puisse, au moins, en faire autant ? Nous allons ici envisager divers procédés et décrire un système de télémétrie avec un capteur infrarouge et le système ATM18.**

La façon dont nous pouvons apprécier l'éloignement d'un objet dépend de nombreux facteurs. La visibilité de cet objet en est un. La connaissance de ses dimensions ou de celles d'objets voisins, en est un autre. Il est rare que nous puissions donner de la distance une évaluation aussi précise que nos instruments de mesure, même les plus frustes, sans être favorisé par le hasard ou une très grande habitude. Nous nous déplaçons pourtant avec une adresse stupéfiante dans notre milieu. Il suffit, pour s'en convaincre, de considérer les personnes qui n'en ont pas encore la faculté (bébés) ou celles qui l'ont perdue.

Comment faire pour qu'un robot sélectionne au mieux son accélération et sa vitesse lorsqu'il est éloigné de 80 à 150 cm de l'objet vers lequel il se dirige ? Nous avons examiné d'un peu plus près les différents procédés de détermination des distances (télémétrie) et avons retenu les trois suivants qui nous semblent les plus répandus :

1. Mesures de temps de propagation et de déphasage d'ondes radioélectriques ;
2. Mesures utilisant la lumière (également laser) ;
3. Télémétrie aux ultrasons.

Disons un mot sur le dernier procédé, utilisé dans le monde animal, par lequel nous allons faire un petit

détour : les Chiroptères ou chauves-souris, par exemple, utilisent diverses techniques d'émission ultrasonore à fréquences fixes et variables, telles que la fréquence des ondes réfléchies par des objets immobiles soit fixe. Les résultats sont calculés si rapidement que ces petits acrobates aériens peuvent se déplacer avec une indescriptible virtuosité en pleine obscurité dans d'étroits boyaux, localiser et prendre de petits insectes en volant rapidement.

Nos sonars à ultrasons artificiels imitent ce procédé mais sont incomparablement plus primitifs. Le monde animal a sur nos techniques une avance que nous ne sommes pas prêts de rattraper.

Tous les procédés ont leurs avantages et leurs inconvénients. Les ultrasons sont très sensibles aux réflexions et aux caractéristiques physiques de l'air. Si nous utilisons les ondes radioélectriques, la détermination des temps de propagation demande une électronique d'analyse des plus rapides.

Il existe d'autres procédés de télémétrie utilisant la lumière, l'un d'eux, d'utilisation assez courante, est appelé triangulation optique [1].

### Goniométrie

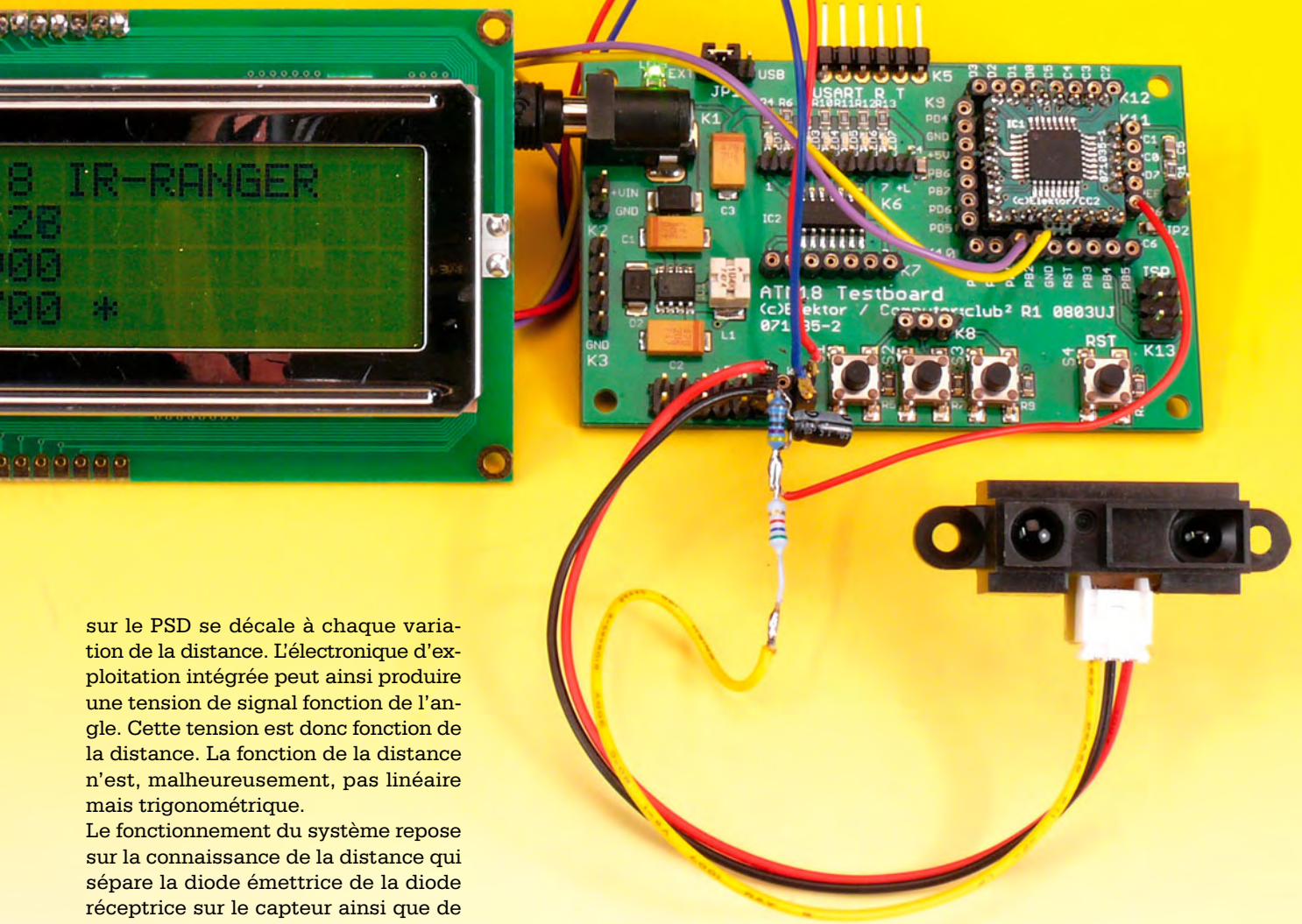
Nous n'allons pas mesurer ici le temps de propagation des ondes lumineuses mais l'angle que fait le faisceau lumi-

neux émis avec le faisceau réfléchi. Dans le domaine professionnel, lorsqu'une très grande précision est exigée, il n'est possible de prendre qu'une diode laser comme émetteur. Si la précision imposée n'est pas extrêmement grande, une diode normale suffit pour la mesure de distances relativement courtes.

Le principe de la triangulation optique est expliqué sur la **figure 1**. Le capteur comporte sur son côté gauche une LED, l'émetteur. Grâce à son optique de précision, elle émet un faisceau lumineux très étroit. Celui-ci est réfléchi par la cible. Une partie du faisceau réfléchi revient alors sur côté récepteur du capteur (à droite sur la figure). L'angle que fait le signal réfléchi varie avec la distance entre le capteur et la cible. Derrière l'optique du récepteur se cache un détecteur de position (« Position Sensitive Detector », PSD), un dispositif à transfert de charge (CCD) d'une seule ligne. Le faisceau lumineux réfléchi est focalisé par l'optique du capteur de façon à rencontrer le plus petit nombre de récepteurs photosensibles possible. Cette condition permet la localisation. Si, maintenant, la distance à l'objet varie, l'angle de la lumière incidente varie également et une autre région du photodétecteur est éclairée (**figure 2**).

Le principe est ainsi clair : le spot lumineux que le faisceau réfléchi produit





sur le PSD se décale à chaque variation de la distance. L'électronique d'exploitation intégrée peut ainsi produire une tension de signal fonction de l'angle. Cette tension est donc fonction de la distance. La fonction de la distance n'est, malheureusement, pas linéaire mais trigonométrique.

Le fonctionnement du système repose sur la connaissance de la distance qui sépare la diode émettrice de la diode réceptrice sur le capteur ainsi que de l'angle  $\alpha$ . L'électronique d'exploitation l'obtient indirectement par l'intermédiaire de la position du PSD éclairée. L'électronique intégrée du capteur produit alors une tension que l'on peut prélever à la sortie du capteur. Ajoutons que ce procédé ne convient qu'à la mesure de faibles distances, de quelques mètres, puisque sa sensibilité dépend de la distance entre émetteur et récepteur qui sont logés tous deux dans le petit boîtier.

Nous disposons d'une tension à partir de laquelle nous voulons déterminer la distance. Il nous faut la calculer. La trigonométrie nous permet d'obtenir la distance  $x - x_0$  à partir de la distance mesurée  $x' - x_0'$  :

$$\tan \delta = \frac{x' - x_0'}{f} \rightarrow \tan \alpha = \frac{x_0}{D}$$

$$x = D \cdot \tan(\alpha + \delta) = D \cdot \frac{\tan \alpha + \tan \delta}{1 - \tan \alpha \cdot \tan \delta}$$

$$x = D \cdot \frac{\frac{x_0}{D} + \frac{x' - x_0'}{f}}{1 - \frac{x_0}{D} \cdot \frac{x' - x_0'}{f}}$$

Une fois obtenue la dernière expression de  $x$ , nous devons, en tout cas, nous ren-

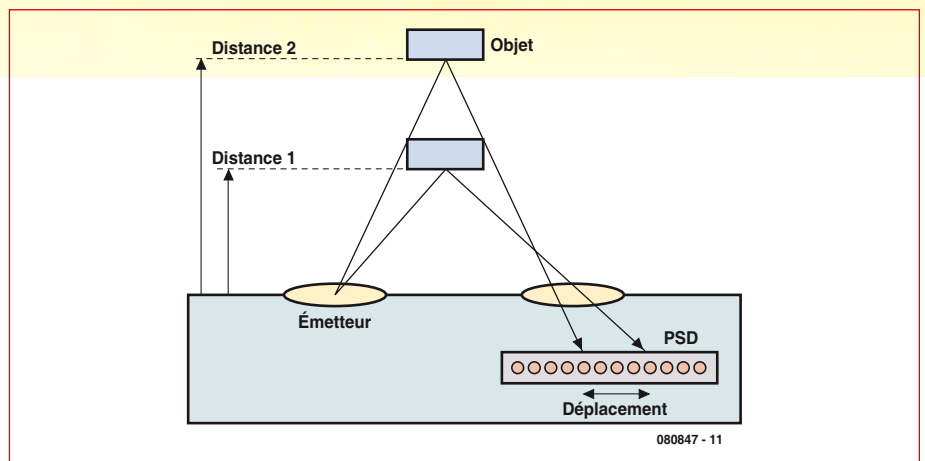


Figure 1. Télémétrie à triangulation optique.

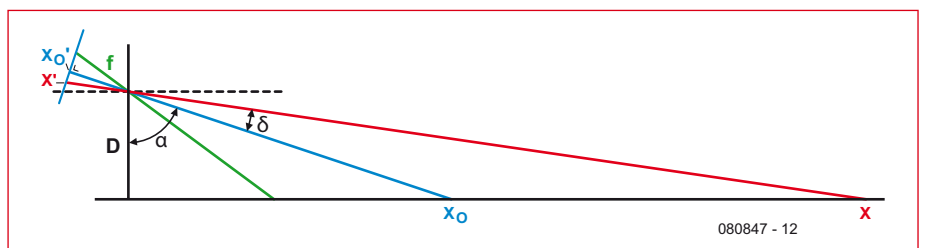


Figure 2. La trigonométrie permet de déterminer la distance  $x - x_0$  à partir de la distance mesurée  $x' - x_0'$ .

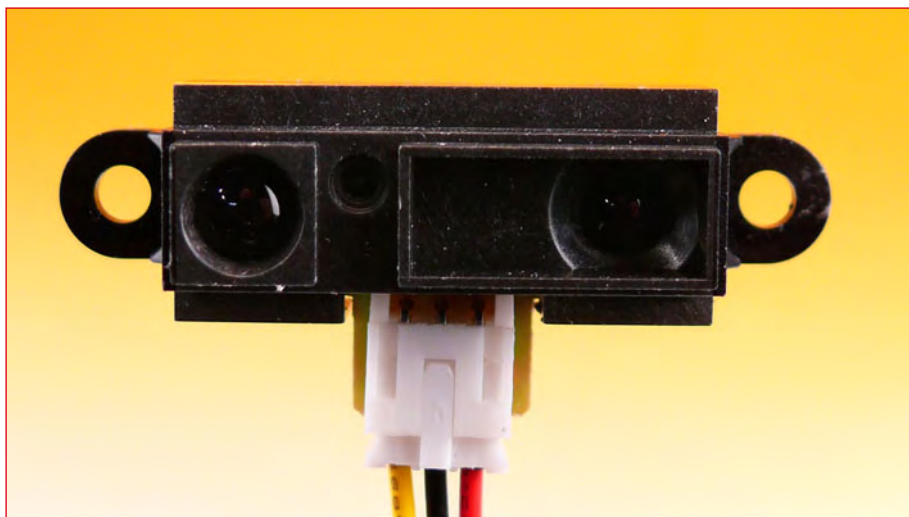


Figure 3. Capteur de distance infrarouge de Sharp.

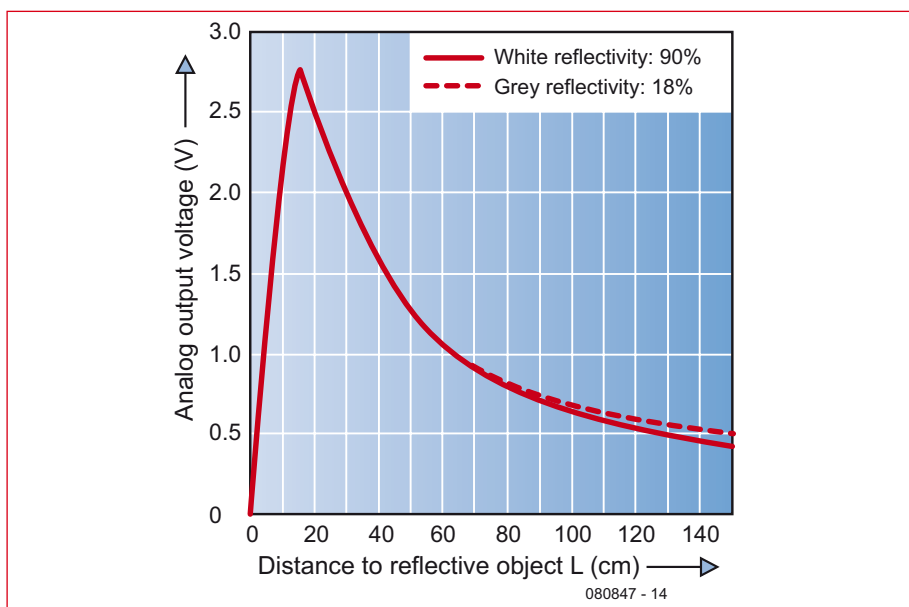


Figure 4. La relation entre la tension de sortie et la distance n'est pas linéaire.

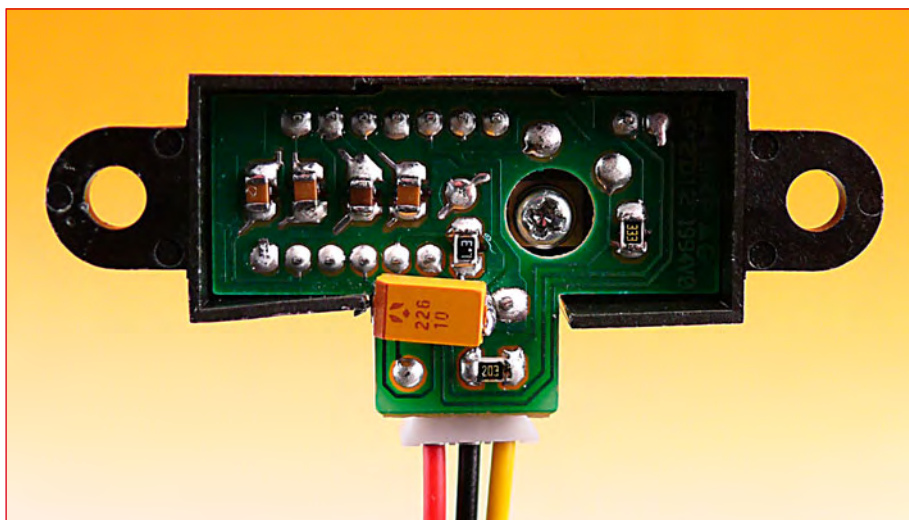


Figure 5. Le capteur a besoin d'un condensateur électrolytique CMS de découplage.

dre à l'évidence: si nous voulions mesurer en continu la distance à un objet quelconque, notre petit contrôleur de 8 bits serait vite surmené par les calculs. Il existe toutefois d'autres méthodes moins gourmandes en puissance de calcul pour déterminer la distance à partir de la tension de sortie du capteur. Laissons cela, nous avons assez de théorie pour notre projet d'aujourd'hui. Passons à la mise en oeuvre.

## Pratique des IRDMS

Ce projet décrit l'application de capteurs de distance infrarouge du fabricant japonais Sharp [2]. Ils circulent sous le terme générique IRDMS (Infrared Distance Measurement Sensor), capteurs infrarouge de mesure de distance. Il en existe de deux sortes. Certains ont des sorties numériques avec un comparateur interne pour une distance définie [3], les autres ont des sorties analogiques. Notre projet n'utilise que des capteurs à sorties analogiques dont le **tableau 1** recense les modèles pour différentes distances. Pour nos expériences, nous avons retenu le capteur GP2Y0A02YK0F pour 20 cm à 150 cm (**figure 3**). Il aurait été cependant possible d'utiliser n'importe quel autre modèle du tableau 1 et le meilleur capteur est celui qui répond le mieux à une application donnée.

Comme le montre la **figure 4**, la tension de sortie des capteurs est très peu linéaire. Il n'est pas possible d'affecter directement la tension à la distance sans linéarisation. Nos premiers essais ne le demandent cependant pas, comme nous l'avons dit précédemment.

En principe, nous n'avons besoin que d'appliquer la tension de service de 5 V pour obtenir le signal du capteur sur une plage qui peut aller jusqu'à 2,7 V environ. La diode IR émettrice fonctionne par éclairs infralumineux intenses et brefs. Ces impulsions ont pour effet de charger chaque fois fortement l'alimentation. Il est, en conséquence, recommandé de câbler un condensateur électrolytique de protection de la tension de fonctionnement à proximité du capteur. La lumière, dans le domaine du proche infrarouge, est juste à peine visible à l'oeil nu dans l'obscurité (elle est, en revanche, très nette sur l'écran de contrôle d'un appareil photo numérique).

La ligne CCD interne est dotée d'une centaine de points actifs. La tension du capteur varie par petits sauts d'en-



viron 20 mV. Une ondulation résiduelle du même ordre de grandeur se superpose au signal, la présence d'un filtre passe-bas est donc tout à fait justifiée. Le convertisseur analogique-numérique de 10 bits du Mega88 atteint déjà une résolution de l'ordre de 5 mV avec la référence externe de 5 V. Cette résolution est, en principe, suffisante pour l'application. Le programme en C préparé met toutefois en service la référence interne de 1,1 V du contrôleur. Avec elle, la résolution descend à 1 mV. Attention ! Ne posez pas le cavalier REF JP2 ! La plage de mesure doit encore être ajustée. Cet ajustage est confié à un diviseur de tension de 4,7 k $\Omega$  et 6,8 k $\Omega$ . La tension mesurée par le contrôleur s'étend ainsi jusqu'à 2,4 V. L'ajout d'un condensateur de 1  $\mu$ F au diviseur de tension nous permet d'obtenir notre filtre passe-bas. Les frais de matériels sont très modestes. Outre les deux résistances du divi-

Tableau 1	
Capteurs IR de distance de Sharp à sortie analogique pour différentes portées:	
Désignation	Portée [cm]
GP2D120XJ00F	4...30
GP2D12J0000F	10...80
GP2D15J0000F	10...80
GP2Y0A02YK0F	20...150
GP2Y0A710K0F	100...500

seur de tension, nous n'avons besoin que de deux condensateurs. Nous devons d'abord souder un condensateur de 10  $\mu$ F à 100  $\mu$ F au plus près du capteur. Vous pouvez toujours nous demander pourquoi Sharp ne l'a pas fait directement sur la platine du capteur. Désolé, mais nous n'en savons rien. On voit, sur la **figure 5**, cette « rustine », un condensateur CMS implanté directement sur la platine. Si vous n'avez pas envie d'intervenir directement sur la platine du capteur, vous pouvez souder un petit condensateur électrolytique de l'« extérieur », donc sur le connecteur, mais, comme toujours, avec des connexions aussi courtes que possible. On implante ensuite le diviseur de tension avec filtre passe-bas. Le diviseur se compose, comme nous l'avons dit, de deux résistances, une de 4,7 k $\Omega$  et une de 6,8 k $\Omega$  (autant que possible avec une tolérance d'au plus 1%). Le condensateur de 1  $\mu$ F est câblé en parallèle à la résistance de 4,7 k $\Omega$  (voir **figure 6**). Le point milieu du diviseur de tension est relié à l'entrée AD6. Vous trouverez sur la **figure 7** le câblage de la carte

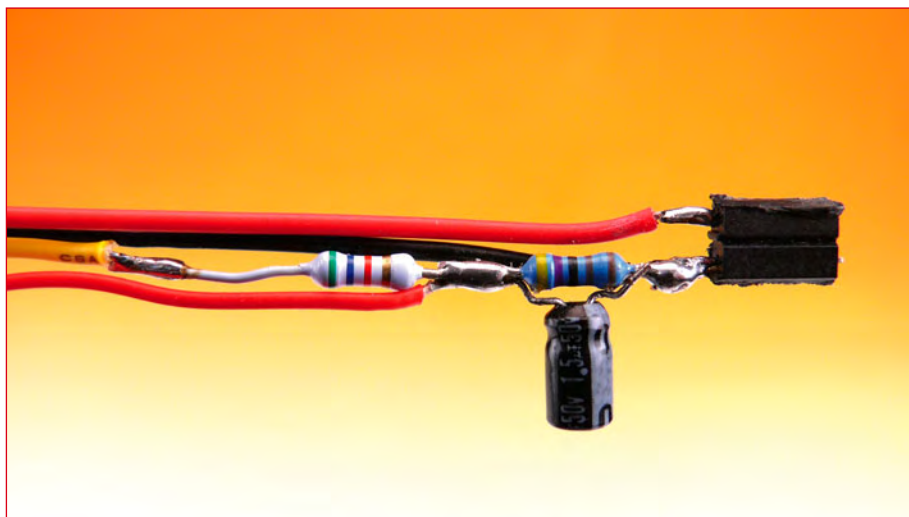


Figure 6. Deux résistances et un condensateur électrolytique forment un diviseur de tension avec filtre passe-bas.

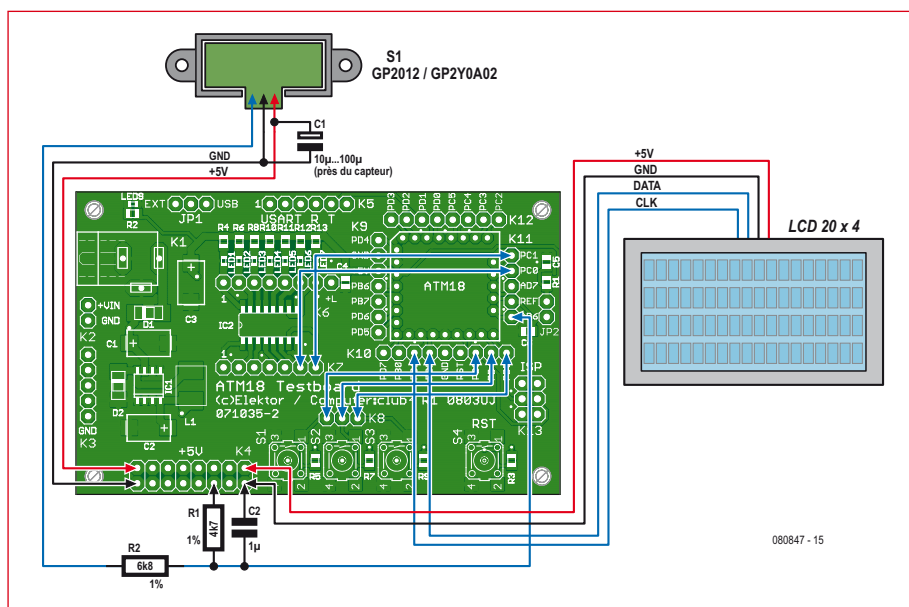


Figure 7. Vue d'ensemble de toutes les liaisons.

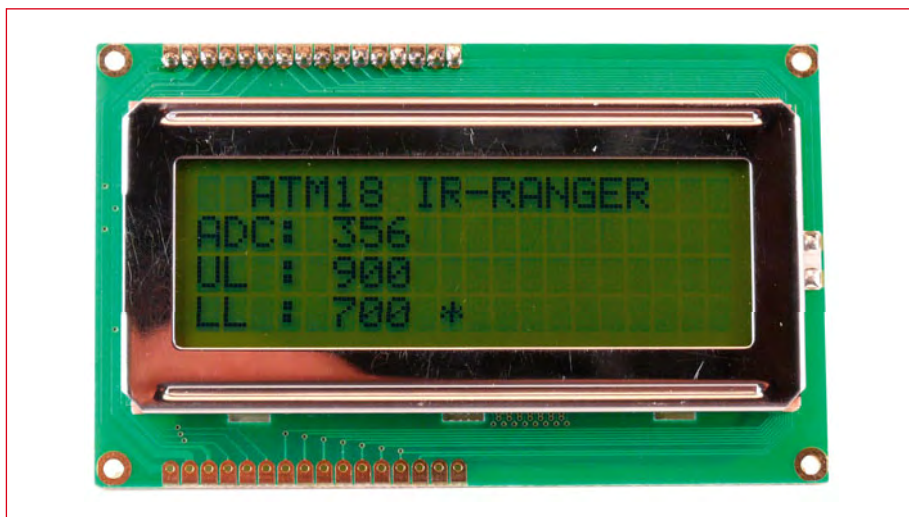


Figure 8. Affichage des valeurs du capteur.

## Listage 1

### Calcul de distance en BASCOM

```
Sub Calculate_s
  D = Getadc(6)
  U = D/1023 * (4.7+5.6)/4.7
  U = D * 5
  U = D * 1.1
  U = U * 10.3
  U = U / 4.7
  U = U / 1023
  Print U

  S = (0.008271 + 939.6 * Us) / (1 - 3.398 * Us + 7.339 * Us * Us)

  If U > 0.4 Then
    S1 = 930.6 * U
    S1 = S1 + 0.008271
    S3 = U * U
    S3 = 17.339 * S3
    S2 = 3.398 * U
    S2 = 1 - S2
    S2 = S2 + S3
    S = S1 / S2
  Else
    S = 0
  End If
  Print S
End Sub

End
```

d'essai. Les touches S1, S2 et S3 sont connectées à PB3, PB4 et PB5. Chacune des broches PC0 et PC1 est reliée à une entrée quelconque de l'ULN 2003. Nous pouvons maintenant passer au

## Logiciel

Le fonctionnement du programme en C de ce projet (ATM18\_IRDMS\_GP2xxx, à télécharger sur [www.elektor.fr](http://www.elektor.fr)) est également très facile à comprendre. Deux valeurs sont définies. Ces valeurs sont contrôlées par le programme et les points de commutation sont signalés par les LED. Quand le LCD est connecté, il affiche la valeur du convertisseur analogique-numérique (ADC : xxxx), le point de commutation haut (UL : xxxx) et le point de commutation bas (LL : xxxx, voir la **figure 8**).

La touche S1 (gauche) définit la valeur courante du capteur comme « lower\_limit », S3 (touche de droite) comme « upper\_limit ». Un appui sur S2 (touche du milieu) redonne aux « upper\_limit » et « lower\_limit » leurs valeurs standard. Une fois les valeurs limites définies, il est possible de déplacer à peu près n'importe quel objet dans la plage de détection. L'afficheur indique la valeur de la tension délivrée par le capteur. Quand les valeurs limites sont atteintes, les LED correspondantes de la

carte d'essai s'allument. Il est ainsi possible, par exemple, de piloter un robot de telle façon qu'il ne se cogne nulle part et évite les obstacles. Le programme peut également adapter dynamiquement les paramètres de distance à la vitesse de déplacement.

Fournissons encore une petite aide pour vos propres applications. Vous pouvez déterminer l'éloignement avec une précision suffisante par un calcul simple :

$$\text{Distance} = \frac{0,008271 + 939,6 \times U_s}{1 - 3,398 \times U_s + 17,339 \times U_s \times U_s}$$

où  $U_s$  est la tension de signal du capteur. Elle est comprise entre 2,5 V (pour une distance de 20 cm) et 0,45 V (pour une distance de 150 cm).

Le temps de calcul est naturellement beaucoup plus court à partir de cette relation qu'à partir de celle que nous avons donnée plus haut. Elle est en outre tout à fait digeste pour BASCOM.

Si, à l'avenir, un projet demande une mesure de distance avec conversion en longueur, nous utiliserons la méthode la plus rapide, à savoir, des tableaux. Nous écrirons dans un tableau les tensions de sortie du capteur sur toute la plage de distance et le programme n'aura plus qu'à les lire.

Restons-en pour l'instant à la technique présentée de façon à laisser un objet mobile décider, comme la chauve-souris avec son appareil de localisation à ultrasons, quelle action déclencher. À l'approche d'un obstacle, les moteurs peuvent s'arrêter et, si l'objet continuait à s'approcher, la marche arrière s'enclencherait. Les chauves-souris procèdent, bien sûr, autrement dans cette situation !

## Commande d'une lampe en BASCOM

Le programme BASCOM (à télécharger sur [www.elektor.fr](http://www.elektor.fr)) n'utilise pas seulement le capteur pour afficher la distance mesurée, mais également pour une commande simple de lampes. Il s'agit ici d'une lampe de bureau qui est

commandée par tous les bits du port B. On peut, par exemple, utiliser le circuit de puissance ULN 2003 de la carte pour attaquer un relais. Le capteur de distance est orienté vers le poste de travail. Dès que qu'une personne s'approche du bureau, la lampe s'allume automatiquement. Quand cette personne quitte son poste, la lampe s'éteint après une temporisation de 100 secondes.

Le système surveille en outre les mouvements de la personne. Lorsqu'elle travaille normalement, elle n'est pas tout à fait immobile et l'amplitude de ses mouvements est toujours d'un peu plus de 3 cm. Si aucun mouvement n'est détecté, la personne s'est sans doute endormie. Pour respecter son sommeil, la commande éteint la lampe. Qu'un chef vienne à passer, et la lampe se rallume, mais le chef a sans doute déjà réveillé son subordonné...

La fonction Calculate\_s (**listage 1**) effectue une mesure et convertit le résultat en une distance s en centimètres. Il est nécessaire de décomposer le calcul, dans BASCOM, puisqu'il n'est pas possible de tout écrire sur une seule longue ligne avec des parenthèses. Les mesures de tension tiennent compte du diviseur de tension de 6,8 kΩ / 4,7 kΩ ainsi que de la référence de tension interne de 1,1 V. La distance déterminée est, en outre, communiquée à 9600 bauds au PC. On voit que la mesure de distance atteint une précision relativement bonne de l'ordre de 10%. Lorsqu'aucun objet n'est dans le champ de vision, la sortie est à zéro. Les lignes correspondant à une variante de connexion du capteur sans diviseur de tension, avec utilisation obligatoire de la tension de fonctionnement de 5 V comme référence, sont en commentaires.

(080847-1)

## Littérature et liens :

- [1] Télémétrie sans contacts, Elektor 10/2001
- [2] [www.sharpsma.com](http://www.sharpsma.com)
- [3] Télémétrie par infrarouge, Elektor 5/2002

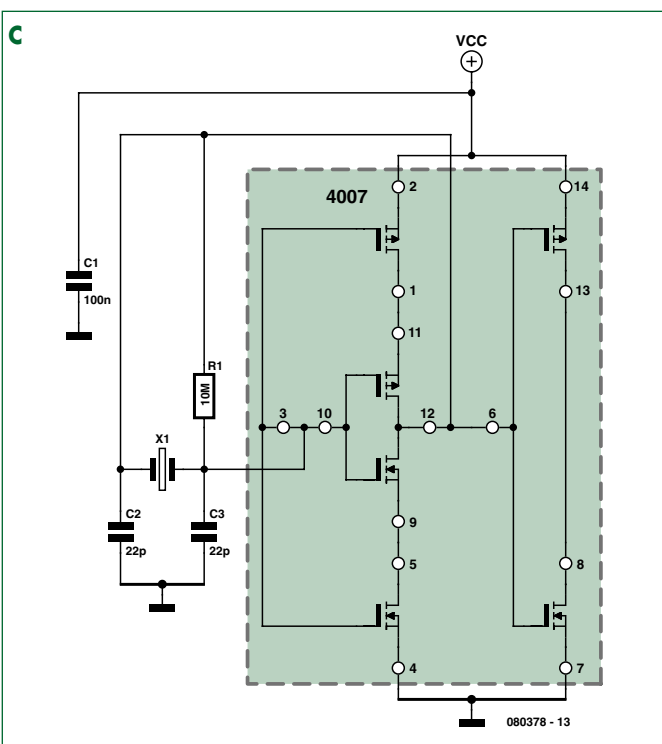
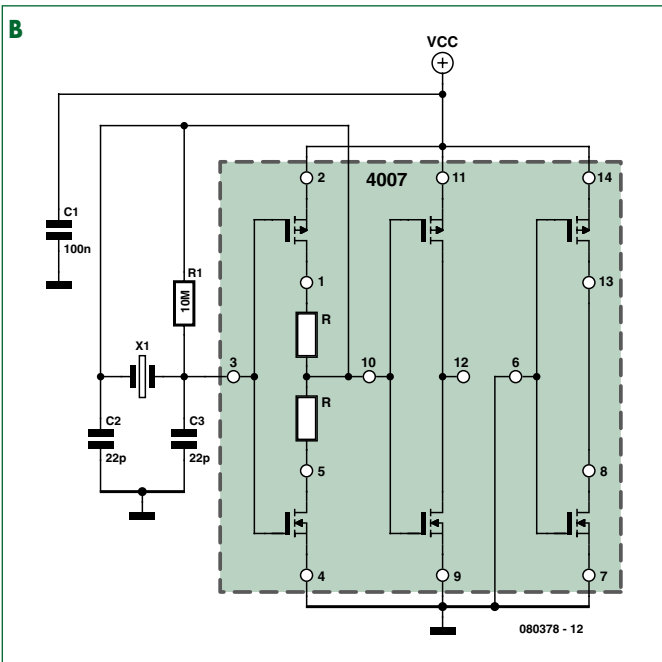
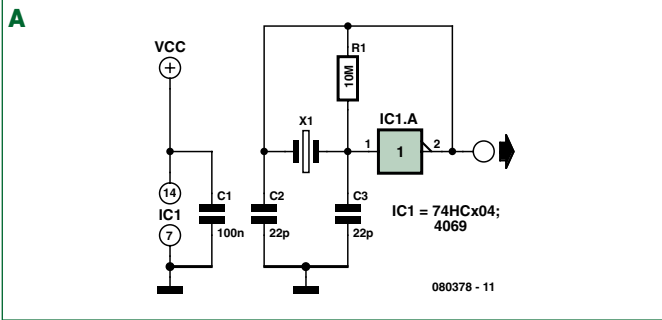
## Le projet ATM18 de Computer:club<sup>2</sup>

ATM18 est un projet mené en commun par Elektor et le Computer:club<sup>2</sup> ([www.cczwei.de](http://www.cczwei.de)) en collaboration avec Udo Jürss, le responsable du développement de [www.microdrones.de](http://www.microdrones.de). Wolfgang Rudolph de Computer:club<sup>2</sup> présente tous les mois les dernières évolutions et applications du système ATM18 lors d'une émission de la NRW-TV (Düsseldorf), CC2-tv.

CC2-tv est émise par NRW-TV comme programme du câble NRW et en flux direct sur Internet ([www.nrw.tv/home/cc2](http://www.nrw.tv/home/cc2)). CC2-tv est baladodiffusée par [www.cczwei.de](http://www.cczwei.de) et, quelques jours plus tard, également par [sevenload.de](http://sevenload.de).



# Economie de courant sur les oscillateurs à quartz



**Rainer Reusch**

La consommation d'un circuit numérique CMOS des séries 74HC et 4000 en fonctionnement statique est pratiquement négligeable. Prenez un inverseur, sur un 74HC04 ou un 4069, par exemple, et vous n'aurez guère à vous casser la tête ou la tirelire pour fabriquer un oscillateur à quartz (**montage A**). Sa consommation monte toutefois sensiblement. Pas de quoi réunir une conférence sur le réchauffement climatique, puisqu'il ne s'agit que de quelques milliampères. Ils jouent quand même un rôle si l'alimentation vient d'une pile ou d'un petit accumulateur. Il est donc légitime de se demander d'où vient l'augmentation de consommation et si les différentes familles logiques présentent, sur ce point, des différences notables. Nous savons que la fréquence de fonctionnement joue sur l'augmentation de consommation mais son rôle n'est que partiel. L'augmentation du courant de fonctionnement tient également au fait que l'inverseur CMOS est mal utilisé, comme amplificateur linéaire, dans l'oscillateur. En entrée et en sortie, nous n'avons, en moyenne que la moitié environ de la tension d'alimentation. Cela signifie que les deux transistors de l'étage complémentaire conduisent. De là à l'idée de construire un étage complémentaire en composants discrets et de réduire la consommation par des résistances, il n'y a pas loin. Le circuit CMOS 4007 contient trois de ces étages et, sur deux des trois, les broches de drain ne sont pas interconnectées. Le **montage B** montre comment cet oscillateur est construit. Comme le signal de sortie est plutôt si-

nusoïdal que rectangulaire, nous mettrons un deuxième étage en service pour l'améliorer. L'idée des résistances de drain peut se creuser encore. On peut ainsi, au lieu de résistances de drain, mettre en service un étage complémentaire de plus. Le **montage C** montre comment le réaliser. La consommation de tous les montages, alimentés sous 5 volts, a été mesurée dans le cas d'un quartz de 4 MHz et d'un quartz de 16 MHz. Pour le montage à 4007, nous avons essayé différentes résistances de drain. Le **tableau** présente les résultats. Remarquons que les circuits de la série 4000 sont déjà proprement surmenés à 16 MHz sous 5 volts et cela peut donner lieu à des problèmes de démarrage des oscillations. Les meilleurs résultats sont ceux du MC14069UB (qui correspond au 4069UB), avec toutefois des problèmes de démarrage à 16 MHz! Les montages à 4007 se détachent également bien. De plus grandes résistances de drain provoquent une diminution de la consommation mais l'effet n'est pas aussi grand qu'on l'espérait quand même. Il n'est, en outre, pas possible d'augmenter indéfiniment les résistances auxquelles la fréquence du quartz et la tension d'alimentation imposent des limites. Le montage C conduit au même résultat que le montage B avec des résistances de 1 k $\Omega$ . L'influence de la température est relativement modeste. Pour conclure, une remarque encore : les valeurs relevées dans le tableau ne sont données qu'à titre indicatif, elles dépendent des caractéristiques des exemplaires qui sont sujettes à dispersions !

(080378-1)

Tableau			
Montage	Circuit / résistance ( $\Omega$ )	4 MHz	16 MHz
A	74HC04	1,47mA	4,9mA
A	74HCU04	1,99mA	3,5mA
A	74HCT04	1,50mA	5,2mA
A	4069UBE	0,56mA	1,4mA
A	MC14069UB	0,37mA	0,7mA *
B	4007 / 0	0,95mA	1,8mA
B	4007 / 100	0,93mA	1,7mA
B	4007 / 470	0,86mA	1,4mA
B	4007 / 1000	0,79mA	(n'oscille pas)
C	4007	0,79mA	(n'oscille pas)

\*) avec des problèmes de démarrage

# Les microcontrôleurs

## ...Arduino pour les initiés

Clemens Valens

**Il paraît qu'Arduino est un prénom italien, mais quand on fait une recherche sur Internet on trouve surtout des dizaines de références touchant à l'électronique et la programmation. En plus, ces références ont souvent un rapport avec l'art. L'électronique et de l'art, voilà un sujet intéressant et qui mérite d'être creusé. Alors, c'est quoi exactement, Arduino ?**

À première vue, Arduino [1] est une petite carte microcontrôleur avec un port USB (**figure 1**) et il en existe plusieurs modèles. Il y a même des cartes en forme de pâquerettes, les *Lilypad*, prévues pour les applications *wearable*, c'est-à-dire, à intégrer dans les vêtements. La carte Arduino se programme dans un langage proche de C avec des outils Open Source disponibles pour Windows, MAC et Linux. Le matériel est *Open* aussi et tout le monde peut faire son propre Arduino, les schémas électriques et les typons étant disponibles gratuitement sur Internet. Les Arduino sont beaucoup utilisés par des artistes qui ont besoin de l'électronique dans leurs créations.

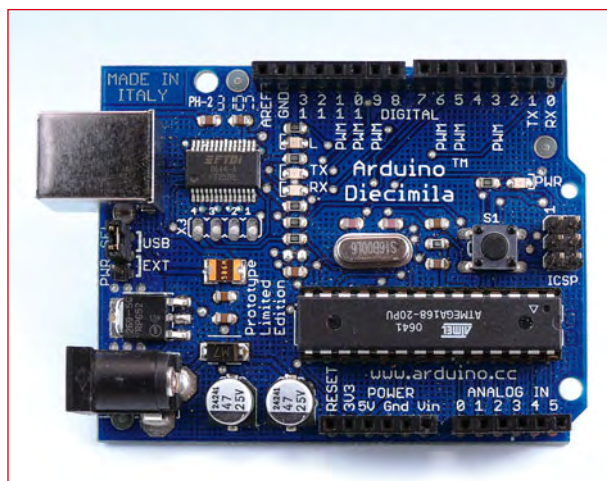


Figure 1.  
Une carte Arduino  
Diecimila. La nouvelle carte  
Duemilanove est quasi  
identique. Ces cartes sont  
faciles à trouver et pas  
chères.

Quand on y regarde de plus près, un Arduino n'est pas exactement une carte microcontrôleur. En fait, un Arduino est tout simplement un microcontrôleur 8 bits de chez Atmel, un ATmega8 pour les premiers Arduino et actuellement plutôt un ATmega168. Ce microcontrôleur est chargé avec un programme d'amorçage, appelé *bootloader*, qui permet de charger une application dans le contrôleur par un port série, sans écraser le bootloader. Les ordinateurs modernes n'ayant plus de ports série, on utilise souvent un port USB. Le tout devient Arduino quand on décide de dédier certaines broches du contrôleur à certaines fonctions, puisque

cela permet l'utilisation de l'environnement de développement Arduino pour écrire et compiler les applications avant de les charger dans le contrôleur.

Les applications, baptisées *sketch* (croquis en français), sont écrites dans un langage qui ressemble beaucoup à du C. Pas étonnant, c'est du C, mais avec des fonctions en plus. L'ensemble des fonctions présentées comme le langage pour Arduino forme une couche d'abstraction du matériel qui permet de programmer le contrôleur sans avoir besoin de se plonger dans les entrailles du processeur. Le langage possède tout ce dont l'on a besoin pour la plupart des applications. En gros il y a des fonctions pour les entrées/sorties numériques et analogiques, quelques fonctions mathématiques de base, des fonctions de gestion de temps (délais) et quelques fonctions pour la communication par port série - asynchrone (UART) et synchrone (SPI).

Les fonctions E/S numériques permettent de manipuler les niveaux logiques des broches, de les lire et de les écrire. Il y a aussi une fonction spéciale qui permet de mesurer la durée d'une impulsion. Avec les fonctions E/S analogiques il est possible de lire des tensions et de générer des signaux PWM. Beaucoup d'applications ne demandent pas plus et c'est exactement ça qui fait la force d'Arduino. Nul besoin d'aller farfouiller dans les registres et la fiche technique du contrôleur pour faire marcher une sortie PWM ou un compteur, le « sale boulot » a déjà été fait.

Si ces fonctions ne suffisent pas, il est tout à fait possible de programmer à un niveau plus bas et, comme dans le C standard, il est également possible de rajouter des bibliothèques avec ses propres fonctions. Mais attention, si on plonge dans les ténèbres du langage de programmation Arduino, on risque de perdre la compatibilité avec la communauté Arduino.

La communauté Arduino ? Arduino est déjà un microcontrôleur, mais aussi un environnement de développement et également un langage de programmation et maintenant c'est une communauté ? Oui. En fait, Arduino est plutôt une philosophie dont le but est de vulgariser la technique pour la rendre accessible aux artistes. Arduino est une suite logique de projets Processing [2] et Wiring [3]. Processing est un langage de programmation multimédia et Wiring est un environnement de développement pour l'électronique artistique. Mais on commence à se disperser, reportez-vous à l'encadré sur les origines d'Arduino si vous voulez en savoir plus.



# rs pour les nuls...



## Elektorino

La programmation facile et gratuite nous intéresse. En plus, l'électronique impliquée à l'air simple, quoi donc de plus logique que de monter notre propre système compatible Arduino ? Et bien, c'est exactement ce que l'on va faire ! Notre point de départ est la carte Arduino Serial basique. L'ordinateur de bureau que j'utilise quotidiennement possède encore un port série, mais pour les malheureux propriétaires d'un ordinateur sans, on va utiliser le câble USB-TTL [4]. De toute façon, nous aurons besoin d'une interface TTL quelconque, puisque notre Arduino à nous n'aura qu'un port série en TTL.

Notre processeur sera un ATmega168 que nous ferons fonctionner à 16 MHz pour éviter un piège. Car même si le contrôleur peut fonctionner jusqu'à 20 MHz, le bootloader standard suppose une cadence de 16 MHz. Cela est modifiable bien sûr si l'on est prêt à se plonger dans le bootloader, mais pour l'instant nous voulons juste une carte Arduino qui fonctionne.

Pour finir notre Arduino il suffit de rajouter une LED, un bouton poussoir de reset, quelques résistances et condensateurs et deux connecteurs : l'un pour le port série et l'autre pour la programmation du bootloader. Ce dernier est nécessaire pour pouvoir programmer notre Arduino la première fois, quand le contrôleur est encore vierge. Plus tard, quand l'application est finie et le bootloader n'est plus nécessaire, ce connecteur peut encore servir pour programmer le contrôleur directement avec l'application. Cela fait gagner de la mémoire.

La LED a plusieurs fonctions. Vu que la LED existe sur plusieurs types d'Arduino, beaucoup de sketches l'utilisent. De même pour le bootloader, qui la fait clignoter au démarrage.

Voilà le schéma électrique de notre Arduino fini (**figure 2**). Grâce à la simplicité du schéma, nous pouvons le réaliser sur une carte d'expérimentation.

Malheureusement nous n'avons pas le droit d'appeler notre belle réalisation Arduino, seulement des cartes approuvées par la communauté Arduino ont le droit à cette appellation. C'est pour cela qu'un deuxième mouvement s'est créé, Freeduino [5], qui permet la libre utilisation de l'appellation Freeduino pour les cartes Arduino faites maison. Mais bon, ce n'est qu'un nom, appelons donc le notre Elektorino.

## Mise en œuvre

Avant qu'il soit possible de charger un sketch dans Elektorino, il faut charger le der. C'est là où les choses se compliquent, puisqu'il y a deux bootloaders Arduino officiels, dont la seule différence, apparemment, est la façon dont le sketch est démarré après chargement. Cela se fait par un reset du contrôleur dont l'environnement Arduino peut se charger si la carte a été équipée pour et si elle possède le bon bootloader. Nous n'avons rien prévu à ce niveau là et il nous faut donc le bootloader basique pour cartes dites NG, Nouvelle Génération.

Mais il y a encore une autre option : un troisième bootloader nommé ADABOOST [6]. Ce bootloader, une version

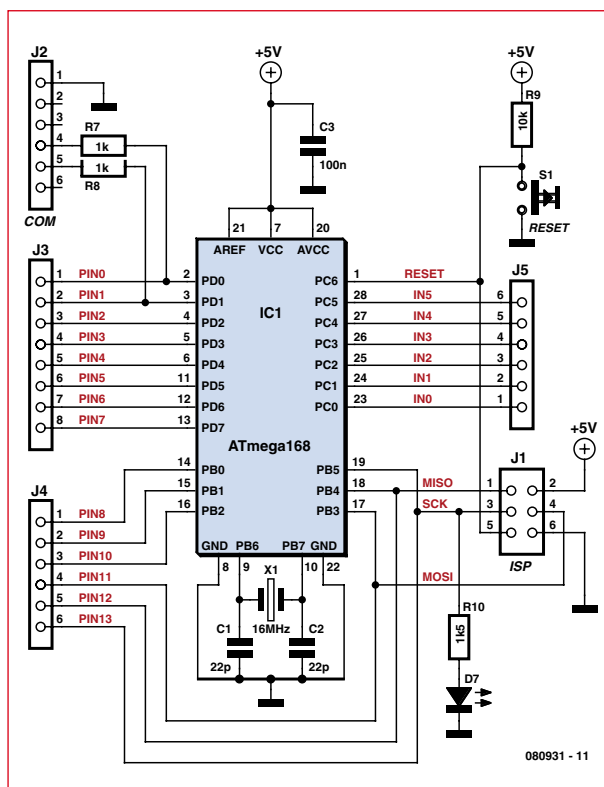


Figure 2.  
Le schéma électrique d'Elektorino. Pas compliqué du tout. Les connecteurs pour les pin ne sont pas nécessaires, ils servent plutôt comme référence. Notez que AREF est relié au VCC

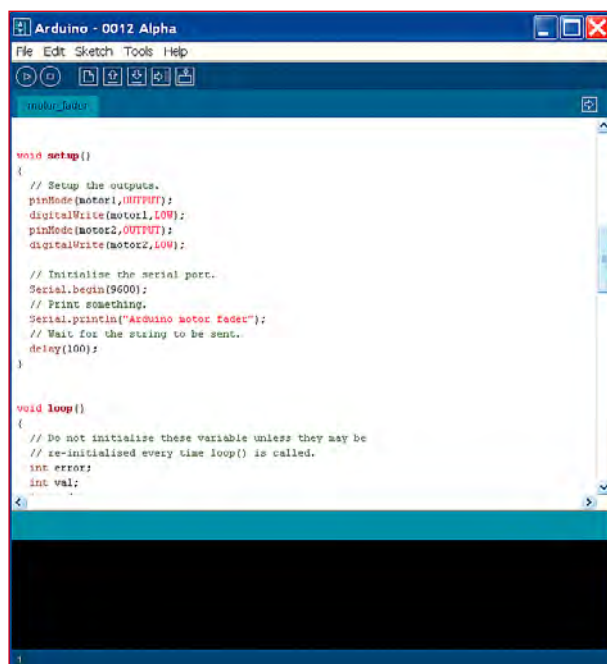


Figure 3.  
L'environnement de développement Arduino.

améliorée des bootloaders officiels, gère différemment le reset et les délais de démarrage. Dans un premier temps j'ai travaillé un peu avec le bootloader NG avant de le remplacer par ADABOOT. Les deux fonctionnent parfaitement mais, finalement, j'ai gardé ADABOOT parce qu'il fait clignoter la LED durant le chargement du sketch et parce qu'il est plus confortable.

Reportez vous à l'encadré pour savoir comment charger le bootloader dans le contrôleur.

## Hello world

Quand on a réussi à charger le bootloader, il est temps de voir si Elektorino arrive à communiquer avec l'environne-

ment Arduino et s'il est possible de charger un sketch. Installons donc Arduino. Je l'ai fait seulement sous Windows XP et c'était extrêmement facile. Il suffisait de télécharger un gros fichier zippé et de le décompresser quelque part sur le disque dur.

Après lancement de l'environnement Arduino (arduino.exe) on se retrouve avec une fenêtre comme celle de la **figure 3**. Allez dans le menu *Tools*, puis *Board* et sélectionnez la carte Arduino utilisée. Bien sûr, la notre n'est pas listée, mais une carte NG avec ATmega168 fait l'affaire.

Il faut aussi sélectionner le port série qui sera utilisé pour programmer le microcontrôleur. Allez dans le menu *Tools*, puis *Serial Port* et sélectionnez le bon port. Si vous voulez utiliser un port série USB vérifiez d'abord que les pilotes sont bien présents.

L'environnement Arduino vient avec un petit sketch pour vérifier si une carte fonctionne, *Blink*, et nous pouvons l'utiliser, puisque nous avons câblé la LED. La procédure est simple :

- Chargez le sketch ; il se trouve dans *File > Sketchbook > Examples > Digital > Blink*.

- Compilez le sketch en appuyant sur le bouton *Verify/Compile* ; cela ne prend que quelques secondes et fini (normalement) avec un message de succès.

- Chargez le sketch dans Elektorino ; appuyez d'abord brièvement sur le bouton *Reset*, puis cliquez sur *Upload* pour démarrer le chargement du programme. Si tout va bien et si vous utilisez ADABOOT vous verrez après un bref délai la LED se mettre à clignoter aléatoirement, normal, c'est le signe que le transfert est en cours. Après environ cinq secondes (dépendant de la taille du sketch) le programme est chargé et le contrôleur est redémarré (la façon exacte comment le programme est lancé dépend du bootloader). Si la LED clignote maintenant avec une fréquence de 0,5 Hz tout s'est bien passé. Elektorino fonctionne ! S'il ne se passe rien, essayez de réinitialiser Elektorino.

## Une vraie application

C'est bien d'avoir un environnement de développement Arduino qui fonctionne à merveille, mais sans véritable

## Charger le bootloader...

... n'est pas si difficile que ça, pourvu que vous ayez toutes les informations à votre disposition. Pour vous épargner des heures de navigation sur Internet, les voici résumées en quelques lignes.

D'abord, il vous faut un programmeur. Il y a plusieurs possibilités, par exemple celui publié dans le numéro double 2008 [7], ou un autre programmeur « compatible SK200 », facile à faire et dont le schéma est disponible sur le site de PonyProg [8]. Sur le site d'Arduino [1] on parle d'encore un autre programmeur pour port parallèle très simple et utilisable directement depuis l'environnement Arduino. Je l'ai essayé et ainsi bousillé un contrôleur... je me suis donc rabattu sur un modèle SK200 que je possédais déjà.

Ensuite, choisissez votre bootloader. Je vous conseille ADABOOT [6], mais la version NG, disponible sur le site d'Arduino, marche aussi parfaitement.

Charger le bootloader dans le contrôleur se fait par exemple avec AVRDUDE [9], fourni avec l'environnement Arduino. AVRDUDE est un outil UNIX typique - à la base c'est un outil de FreeBSD - avec beaucoup d'options incompréhensibles. Parce qu'il est très facile de se tromper, voici les commandes qui vont bien (copier le bootloader dans le répertoire où se trouve l'exécutable d'AVRDUDE) :

```
avrdude -p m168 -c pony-stk200 -V -e -U lock:w:0x3F:m -U hfuse:w:0xDF:m -U lfuse:w:0xFF:m -U efuse:w:0x0:m
avrdude -p m168 -c pony-stk200 -V -D -U flash:w:ATmegaBOOT_168_ng.hex
avrdude -p m168 -c pony-stk200 -V -U lock:w:0x0F:m
```

Si vous utilisez un autre programmeur, remplacez pony-stk200 par la valeur adaptée. Vérifier aussi le nom de votre bootloader.

Il y a trois commandes qui, en gros, déverrouillent la mémoire, chargent le programme, positionnent les fusibles et finalement verrouillent la mémoire. Reportez-vous à la notice d'AVRDUDE si vous voulez savoir exactement ce qui se passe (âmes sensibles s'abstenir). Le verrouillage de la mémoire sert à éviter que le chargement d'un sketch écrase accidentellement le bootloader.

Un bon site sur l'AVR, mais en anglais, est le site de Lady Ada [10].



application ce n'est pas très intéressant. J'avais encore en ma possession une dizaine de potentiomètres rectilignes motorisés et il était temps de les utiliser. Pourquoi pas avec Elektorino ? Elektorino a des entrées analogique et des sorties PWM, tout ce qu'il faut pour piloter un moteur. Je vous propose donc un pilote pour potentiomètre (rectiligne) motorisé. Notez que ce montage est utilisable avec toutes les cartes Arduino basé sur le ATmega168.

Le potentiomètre en question (**figure 4**) consiste en un potentiomètre rectiligne, un petit moteur et un petit assemblage de quelques roulettes, des ressorts et un bout de fil, permettant au moteur de déplacer le curseur dans les deux sens. Cet assemblage fait tourner le moteur dans le vide quand le curseur ne peut pas bouger, aux extrémités par exemple. Mis à part le texte *10K B* imprimé sur le potentiomètre, je ne connaissais aucune caractéristique technique, mais quelques expériences ont montré que le moteur tournait à une bonne vitesse s'il était alimenté sous 12 V. Sa consommation dans ce cas est d'environ 200 mA.

Le *B* imprimé sur le potentiomètre laisse supposer qu'il s'agit d'un modèle logarithmique, comme souvent, mais après vérification mes potentiomètres s'avéraient être linéaires. Comme pilote pour le moteur j'ai choisi un double pont en H astucieusement modifié avec seulement deux lignes de commande et trois états : gauche, droit et freinage, juste ce dont nous avons besoin (**figure 5**). Normalement, deux commandes permettent quatre états, mais dans ce montage les états 00 et 11 sont identiques. Un régulateur 5 V s'est glissé dans le schéma pour pouvoir alimenter l'ensemble du contrôleur et le moteur sous 12 V. Les transistors sont tous des NPN et ceux qui forment le pont doivent supporter 200 mA sans broncher. J'ai utilisé des BD139 dans mon prototype.

Le potentiomètre est monté comme simple diviseur de tension. En mesurant la tension présente sur le curseur nous savons où il se trouve (pour peu que le potentiomètre soit linéaire).

Il faut connecter les commandes du pilote de moteur à des sorties numériques capables de fournir un signal PWM. Un Arduino basé sur un ATmega168 en possède six, un ATmega8 que trois. Le curseur du potentiomètre peut quant à lui être connecté à n'importe quelle entrée analogique, in0 dans notre cas.

## Le sketch

Maintenant que nous avons connecté un pilote de moteur à Elektorino (**figure 6**), il est temps de s'occuper du logiciel. Vous allez voir, le sketch final sera étonnamment simple, grâce à la puissance d'Arduino.

Un sketch basique se compose de deux fonctions : *setup()* et *loop()*, qui sont appelées par la couche d'un niveau plus bas. Dans *setup*, appelée une fois au démarrage, on met tout ce qui touche à l'initialisation du système, les entrées/sorties et la vitesse du port série par exemple.

99,9% des programmes embarqués passent probablement leur vie dans une boucle. C'est donc pour ça que dans Arduino cette boucle est déjà implémentée sous forme de la fonction *loop*. La fonction *loop* est appelée périodiquement et peut être considérée comme le *main* d'Arduino. Il est important de comprendre que *loop* est exactement comme n'importe quelle fonction en C, même si elle à l'air d'une fonction spéciale. Ses variables locales sont donc réinitialisées à chaque appel et il faut déclarer globalement les variables qui doivent garder leurs valeurs entre les différents appels de *loop* (ou comme *static* pour ceux qui se connaissent un peu en C).

Le *setup* de notre sketch ne contient pas grand-chose. Les

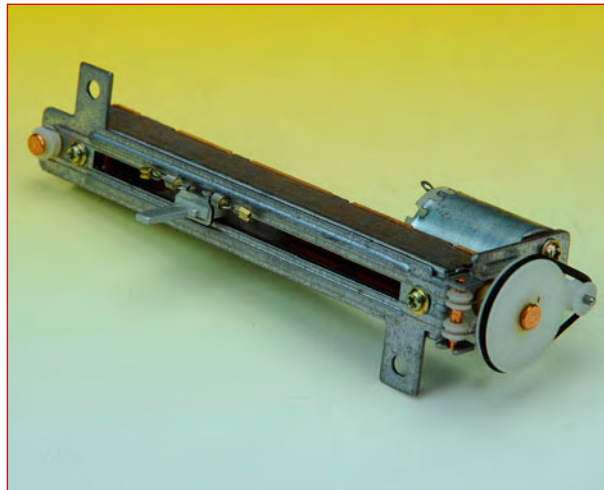


Figure 4.  
Un potentiomètre rectiligne motorisé de marque inconnue.

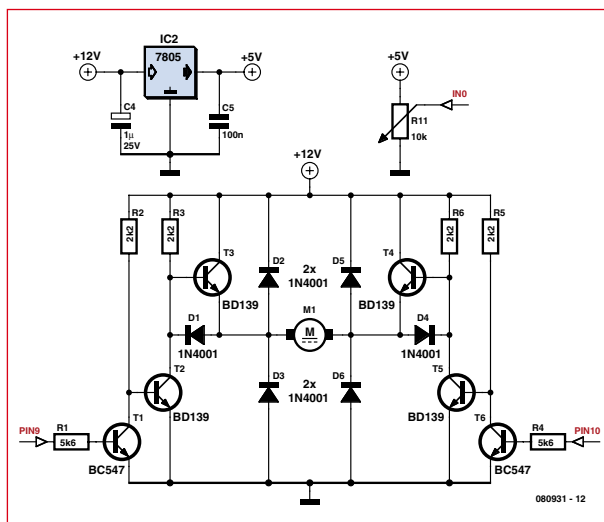


Figure 5.  
Le double pont en H modifié et à trois états. Les labels référencent des pin, pas des broches du contrôleur !

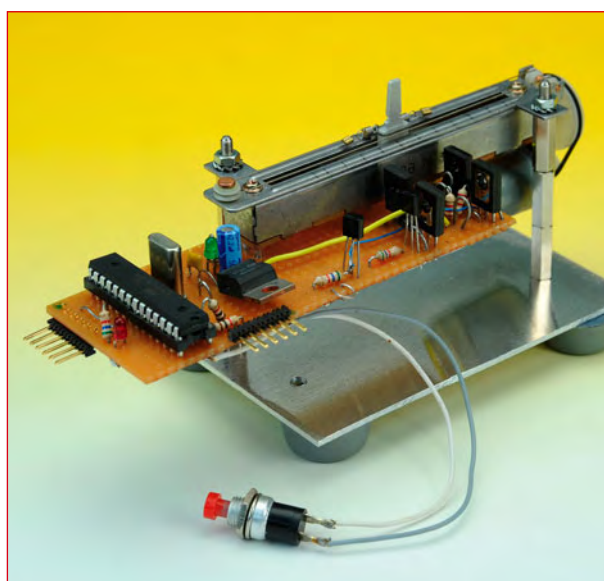


Figure 6.  
Le prototype de Elektorino : l'ATmega168 à gauche et le double pont en H à droite pour piloter le moteur.

pins d'Arduino sont des entrées par défaut, il suffit donc d'initialiser les deux sorties. Nous allons utiliser le port série pour piloter notre montage et pour cela il faut l'initialiser. Grâce aux simplifications apportées par Arduino il suffit

## Processing, Wiring et Arduino

**Processing** [2] est un langage et un environnement de programmation Open Source pour la programmation d'images, d'animations et d'interactions. Le projet, à l'initiative de Ben Fry et Casey Reas, est basé sur des idées développées par le Aesthetics and Computation Group du MIT Media Lab. Processing a été créé pour enseigner les fondements de la programmation dans un contexte visuel et pour servir comme carnet de croquis ou outil de production professionnel de logiciel. Processing tourne sous GNU/Linux, Mac OS X, et Windows.

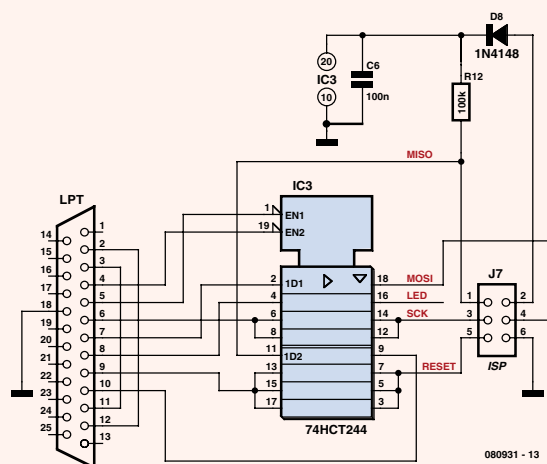
Plusieurs livres au sujet de Processing ont déjà été écrits.

**Wiring** [3] est, tout comme Arduino, un environnement de programmation avec carte microcontrôleur pour explorer les arts électroniques, l'enseignement de la programmation et le prototypage rapide. Wiring, programmé en Processing, est une initiative de Hernando Barragán et a été conçue à l'Interaction Design Institute Ivrea (IDII) en Italie.

**Arduino** [1] est une plateforme de prototypage électronique rapide Open Source. Arduino vise les artistes, les stylistes, les passionnés et ceux qui sont intéressés par la création d'objets ou d'environnements interactifs. Créé par Massimo Banzi, Gianluca Martino, David Cuartielles et David Mellis, Arduino utilise un langage de programmation basé sur Processing. Arduino peut être considéré comme une simplification de Wiring.

de saisir la vitesse de communication - 9 600 bauds dans notre cas.

Le programmeur compatible STK200 utilisé par l'auteur. Il existe des programmeurs plus simples, une question de goût.



Déplacer le curseur du potentiomètre se fait dans *loop*. Le principe est très simple : si la tension mesurée sur le *pin* d'entrée est différente de la tension souhaitée, il faut déplacer le curseur dans le sens qui fait diminuer cette différence. En réalité c'est un peu plus compliqué. Déjà il y a le problème de la direction, mais plus important est le problème de l'inertie. Une fois que le curseur est en mouvement, il a besoin d'un peu de temps pour s'arrêter complètement. Il est donc facile de dépasser la position voulue si on freine

trop tard. En cas de dépassement il faut faire revenir le curseur avec le même risque de dépassement et ainsi de suite. Ce système peut même commencer à osciller.

Pour éviter ces problèmes nous faisons appel à un régulateur *Proportionnel Différentiel* (PD). Dans ce genre de régulateur le système atteint sa valeur finale sans dépassement en ajustant en continu le signal de correction à la différence restant à corriger. Donc, au début d'un réglage, quand l'erreur est maximale, le signal de correction est fort. Puis, quand l'écart commence à diminuer, le signal de correction diminue aussi et le système ralentit.

Le signal de correction est composé de deux parties, un signal proportionnel à l'erreur (P) et un signal proportionnel à la diminution d'erreur (D). Avec un tel système bien réglé il est possible de déplacer le curseur rapidement sans dépasser la valeur cible.

Dans le sketch (**listing 1**) on retrouve le régulateur PD dans la fonction *loop*. Nous mesurons d'abord la tension présente sur le curseur. La valeur cible est soustraite de la valeur mesurée pour obtenir l'erreur à corriger. À partir de cette valeur nous calculons les deux composants P et D du signal de correction. Le composant P est l'erreur multipliée par la constante  $K_p$ , le composant D est obtenu en multipliant la différence de l'erreur actuelle et l'erreur précédente par la constante  $K_d$ . Les valeurs des constantes ont été déterminées par expérimentation et vous pouvez les modifier pour voir leur influence sur le réglage. C'est très instructif.

Les deux composants P et D sont combinés et le résultat est adapté à la plage des valeurs utilisables. Le curseur du potentiomètre ne bouge pas pour les valeurs en dessous de 50 et la valeur maximale pour le signal PWM est de 255. Ensuite nous regardons si l'erreur est assez petite pour pouvoir arrêter le moteur. Il faut faire cette comparaison pour les deux directions du curseur. Nous laissons une petite marge d'erreur, puisque la perfection est peut-être trop ambitieuse.

## AVR ISP par USB

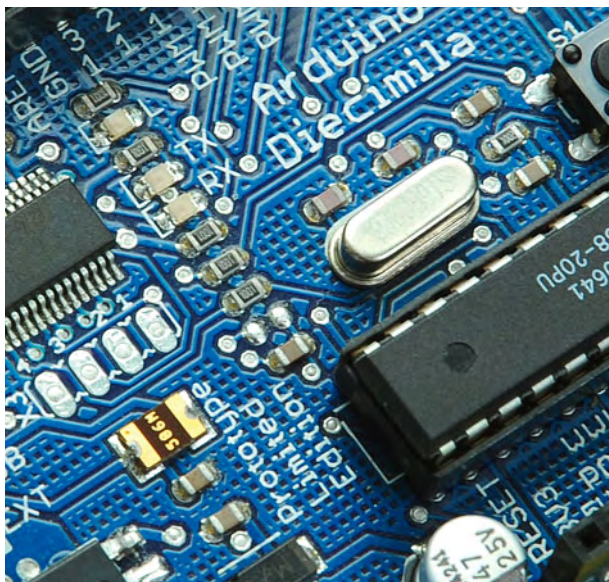
Si l'on utilise la puce FT232R comme interface USB et si l'on a accès à toutes ses broches, il est possible d'utiliser cette puce pour charger le bootloader dans le microcontrôleur sans avoir besoin d'un programmeur ISP spécial ! La puce FT232R possède un bus nommé CBUS avec fonction bit bang qui permet de manipuler les broches associées individuellement. Un certain monsieur Suz du Japon a écrit un petit logiciel qui exploite cette possibilité et que l'on peut télécharger gratuitement. Son outil *avrdude-serj-tag* ne fonctionne que sous Windows et son site Internet est malheureusement en japonais ([suz-avr.sblo.jp/article/4438871.html](http://suz-avr.sblo.jp/article/4438871.html)). Toutefois un de ses compatriotes explique gentiment et en détail sur son site, en anglais, comment programmer un Arduino avec cet outil. Reportez vous à la référence [11] pour tous les détails.



Quand l'erreur est assez petite nous interdisons des corrections supplémentaires pour libérer le curseur ; nous supposons que le système ne dépassera jamais la valeur cible. Ainsi il est possible de déplacer le curseur manuellement, sans que le système essaye de le remettre à sa place. (C'est qui, le plus fort ?)

Une fois le curseur libéré, le système commence à envoyer périodiquement (10 Hz) la position du curseur par le port série. L'entrée du port série est également scrutée et dès que quatre caractères ont été reçus, ils sont transformés en valeur cible pour le curseur et le régulateur PD est réactivé pour le déplacer vers sa nouvelle position. Il n'y a pas de vérification de format pour la valeur reçue, le système imposant l'existence d'une valeur de quatre chiffres en ASCII comprise entre 0000 et 1023. Pour limiter les erreurs, la valeur cible obtenue est bornée entre 3 et 1020, ce qui limite des problèmes d'activité continue aux extrémités. Le port série n'est pas utilisé pendant que le moteur est actif, car cela produirait des interférences, résultant en des positions incorrectes et même des oscillations. Je n'ai pas fait l'effort de comprendre pourquoi : à vous de jouer !

(080931-I)



## Références

- [1] [arduino.cc](http://arduino.cc)
- [2] [www.processing.org](http://www.processing.org)
- [3] [wiring.org.co](http://wiring.org.co)
- [4] [www.elektor.fr/usb-ttl](http://www.elektor.fr/usb-ttl)
- [5] [www.freeduino.org](http://www.freeduino.org)
- [6] [nearspacevermont.org/TheShoppe/freeduino/ADABOOT.shtml](http://nearspacevermont.org/TheShoppe/freeduino/ADABOOT.shtml)
- [7] SimpleProg – ISP pour AVR, Elektor, juillet/août 2008
- [8] [www.lancos.com/prog.html](http://www.lancos.com/prog.html)
- [9] [www.bsddhome.com/avrdude](http://www.bsddhome.com/avrdude)
- [10] [www.ladyada.net/learn/avr/index.html](http://www.ladyada.net/learn/avr/index.html)
- [11] [www.geocities.jp/arduino\\_diecimila/bootloader/index\\_en.html](http://www.geocities.jp/arduino_diecimila/bootloader/index_en.html)
- [12] Getting Started with Arduino, Banzi, Massimo, O'Reilly, 2008
- [13] Making Things Talk, Igoe, Tom, O'Reilly, 2007
- [14] La carte Arduino Duemilanove est vendue en France par Alyasoft Education.

## Listing 1. Trop fastoche !

```
void loop()
{
    int error;
    int val;
    int spd;
    float spd_p, spd_d;

    // Lire tension du curseur
    val = analogRead(slider);

    // Calcule erreur.
    error = val - target;

    // Calcule composant proportionnel P.
    // Utiliser valeur absolue car 2 directions.
    spd_p = abs(error)*Kp;

    // Calcule composant différentiel D.
    spd_d = (last_error-error)*Kd;
    last_error = error;

    // Mélanger P & D.
    spd = int(spd_p+spd_d);

    // Ne pas dépasser les limites.
    spd = constrain(spd,0,255);
    // Compenser friction.
    if (spd<50) spd += 50;

    if (error<-1 && stop==0)
    {
        // Vers valeur max ("gauche").
        digitalWrite(motor2,LOW);
        analogWrite(motor1,spd);
    }
    else if (error>1 && stop==0)
    {
        // Vers valeur min ("droite").
        digitalWrite(motor1,LOW);
        analogWrite(motor2,spd);
    }
    else
    {
        // Couper le moteur
        digitalWrite(motor1,LOW);
        digitalWrite(motor2,LOW);
        stop = 1;

        // Transmettre la position du curseur.
        Serial.println(val);
        delay(100);

        // 4 caractères forment une nouvelle valeur cible.
        if (Serial.available()>=4)
        {
            target = Serial.read() - '0'; // Mille.
            target = Serial.read() - '0' + target*10; // Cent.
            target = Serial.read() - '0' + target*10; // Dix.
            target = Serial.read() - '0' + target*10; // Un.
            constrain(target,1,1022);
            // Démarrer le moteur
            stop = 0;
        }
    }
}
```

# Cours BASCOM-AVR

## (5) Mémoire, interrogation de commutateurs et gestion du temps

Burkhard Kainka

**On ne dispose jamais de trop de mémoire et de puissance de calcul. Mais ce sont précisément les ressources des microcontrôleurs qui sont souvent « un peu justes ». Il faut donc en tirer le meilleur parti possible. Cela s'applique particulièrement aux boucles de programme exécutées sans interruption.**

Il est toujours important de savoir à quoi le contrôleur passe son temps de calcul. Il suffit par exemple d'une petite boucle discrète d'interrogation de touches ou poussoirs pour dévorer l'intégralité du temps de calcul. Il importe donc de développer un concept basé sur l'utilisation optimale du temps de calcul. Cette partie du cours BASCOM illustre quelques possibilités de programmation optimisée.

### RAM et EEPROM

L'ATmega88 possède 1024 octets de RAM et 512 octets d'EEPROM. C'est un bon début ! BASCOM utilise la RAM

pour les variables et autres piles. Que reste-t-il ? Testons en enregistrant des données dans un tableau. La variable A(500) est un tableau. Il s'agit donc de 500 octets A(1) à A(500). Attention : A(0) n'existe pas. Le petit programme de test dans **Listage 1** écrit des données ascendantes en mémoire. Elles sont lues dans une seconde boucle et envoyées au PC.

Le fichier Memory.rpt, qui indique le degré d'occupation mémoire, permet de déterminer à quel point on s'approche dangereusement de la limite. Il s'agit d'un fichier texte qui peut être ouvert avec Notepad. Il contient la taille mémoire, la position exacte de toutes les variables et bien plus encore.

Test 2 illustre les possibilités de l'EEPROM qui, contrairement à la RAM, conserve les données lors de la mise hors tension. Les données sont mémorisées de façon permanente par Writeeeprom, Variable, Adresse memoire et lues par Readeeprom, Variable, Adresse memoire. Toutes les adresses d'une EEPROM effacée ne contiennent que FF (255). On peut donc lire un octet et déterminer si son contenu a déjà été modifié. Dans Test 2 (**Listage 2**), 512 octets sont écrits et relus.

### Interrogation de touches

Les appareils autonomes sont souvent équipés de touches de commande qui doivent être interrogées et évaluées selon des règles exactement définies. Quoi de plus simple que d'interroger des touches ? Cette tâche, apparemment anodine, recèle toutefois un certain nombre de pièges. L'un d'entre eux résulte de l'ignorance dans laquelle on se trouve quant à la fréquence et à la durée de la pression sur la touche. Il faudrait donc interroger continuellement la touche utilisée. L'introduction de fonctions par touches dans un programme déjà opérationnel ne constitue pas exactement un jeu d'enfant. Il vaut mieux partir des touches pour réaliser pas à pas les fonctions désirées.

Le rebondissement fréquent des touches constitue un autre piège. Les rebondissements d'une touche pressée se chargent de simuler plusieurs événements. L'interrogation des touches sur la base d'une temporisation ne constitue donc

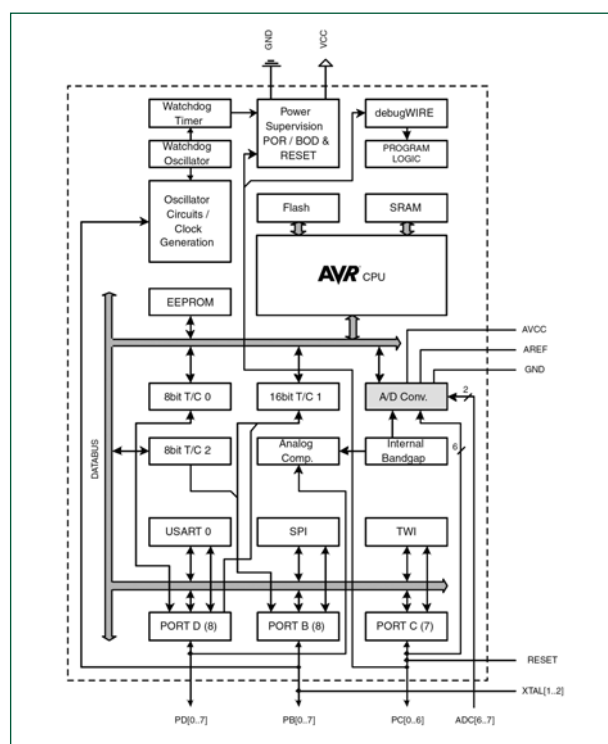


Figure 1.  
Schéma fonctionnel de  
l'ATmega88.





## Listage 5 Une double bascule bistable

```
Test5:
Do
  If S1 = 0 Then
    If Out1 = 0 Then
      Out1 = 1
    Else
      Out1 = 0
    End If
    Waitms 10
  End If
Do
Loop Until S1 = 1
If S2 = 0 Then
  If Out2 = 0 Then
    Out2 = 1
  Else
    Out2 = 0
  End If
  Waitms 10
End If
Do
Loop Until S2 = 1
Waitms 100
Loop
```

## Listage 6

### Un double compteur d'événements

```
Test6:
Dim Count1 As Word
Dim Count2 As Word
Do
  If S1 = 0 Then
    Count1 = Count1 + 1
    Print «Count1 »;
    Print Count1
    Waitms 50
  Do
    Loop Until S1 = 1
  End If
  If S2 = 0 Then
    Count2 = Count2 + 1
    Print «Count2 »;
    Print Count2
    Waitms 50
  Do
    Loop Until S2 = 1
  End If
Loop
```

## Listage 7 Interrogation des touches par interruption

```
Test7:
Dim Ticks As Byte
Dim Sw1 As Byte
Dim Sw2 As Byte
Dim Sw3 As Byte
Dim Sw4 As Byte
Dim Pwm1 As Integer
Dim Pwm1old As Integer
Dim Ledtimer As Byte
Dim Ledblink As Byte
```

```
Ledblink = 1
Enable Timer0
Enable Interrupts
Cls
Lcd 0
```

```
Do
  If Ticks = 1 Then Out1 = 1
  If Ticks = 5 Then Out1 = 0
Loop
```

```
Timer0isr:
  Ticks = Ticks + 1
  If Ticks = 1 Then
    If S1 = 0 Then Sw1 = Sw1 + 1 Else Sw1 = 0
    If Sw1 > 100 Then Sw1 = 100
    If S2 = 0 Then Sw2 = Sw2 + 1 Else Sw2 = 0
    If Sw2 > 100 Then Sw2 = 100
    If S3 = 0 Then Sw3 = Sw3 + 1 Else Sw3 = 0
    If Sw3 > 100 Then Sw3 = 100
  End If
  If Ticks = 2 Then
    If Sw1 = 3 Then
```

```
      Pwm1 = Pwm1 + 1
      If Pwm1 > 1023 Then Pwm1 = 1023
    End If
    If Sw1 = 100 Then
      Pwm1 = Pwm1 + 1
      If Pwm1 > 1023 Then Pwm1 = 1023
    End If
    If Sw2 = 3 Then
      Pwm1 = Pwm1 - 1
      If Pwm1 < 0 Then Pwm1 = 0
    End If
    If Sw2 = 100 Then
      Pwm1 = Pwm1 - 1
      If Pwm1 < 0 Then Pwm1 = 0
    End If
    If Pwm1 <> Pwm1old Then
      Print Pwm1
    End If
    Pwm1a = Pwm1
    Pwm1old = Pwm1
  End If
  If Ticks = 3 Then
    If Sw3 = 3 Then
      If Ledblink = 1 Then
        Ledblink = 0
      Else
        Ledblink = 1
      End If
    End If
  End If
  If Ticks = 4 Then
    Ledtimer = Ledtimer + 1
    If Ledtimer > 100 Then Ledtimer = 0
    If Ledtimer = 1 Then
      If Ledblink = 1 Then Out2 = 1
    End If
    If Ledtimer = 50 Then Out2 = 0
  End If
  If Ticks = 10 Then Ticks = 0
Return
```



sur la touche pour aboutir finalement à un état aléatoire. Le déroulement du Test 5 permet aussi de commander 2 compteurs. Les pressions sur les touches S1 et S2 sont comptées. Le PC est informé de chaque modification d'un compteur. Le problème de la charge à 100 % du contrôleur n'est toujours pas résolu.

### Interrogation des touches par interruption du temporisateur

Tous les exemples d'interrogation de touches présentés jusqu'ici, s'ils en illustrent bien le principe, ne gèrent pas efficacement le temps de calcul. Quand il faudra gérer plusieurs touches et d'autres fonctions, il sera grand temps de trouver une méthode de gestion du temps plus performante. Test 7 (**Listage 7**) illustre une solution plus efficace. Les touches sont interrogées à l'arrière-plan par un sous-programme d'interruption de temporisation. Le programme principal peut se consacrer accessoirement à d'autres tâches.

Une variable Sw1 à Sw3 est associée à chaque touche S1 à S3. La variable associée à une touche est nulle tant que cette touche n'est pas pressée. La valeur de la variable croît continuellement jusqu'à 100 quand la touche est pressée. On peut donc déterminer à tout moment si une touche a été pressée et, si tel est le cas, pendant combien de temps. Pour évaluer une pression particulière sur une touche on teste si sa valeur, par exemple 3, est associée au déclenchement d'un événement. La valeur 100 correspond à une pression prolongée.

Le sous-programme de temporisation contient un second compteur prévu pour les petites unités de temps : les

« Ticks ». La variable Ticks varie de 1 à 10. Les touches ne sont donc interrogées qu'à chaque dixième appel (lorsque Ticks=1). Les problèmes de rebond sont donc accessoirement résolus et le programme est encore assez rapide pour ne rater aucun événement.

Les événements peuvent être évalués à des instants différents. Dans l'exemple, une valeur PWM est modifiée lorsque le temps Ticks = 2 et le clignotement d'une DEL est activé ou désactivé lorsque le temps Ticks = 3. Ticks = 4 gère le clignotement proprement dit. Ce partage du temps de calcul donne une impression de simultanéité à l'utilisateur, mais le contrôleur dispose encore de grandes réserves. Le programme principal commande la sortie 1. Il l'active pendant 5 Ticks et la désactive pendant 5 Ticks. Une DEL raccordée clignote si rapidement qu'on n'observe qu'une diminution de la luminosité. La luminosité de la DEL ne varie pas, indiquant que le rythme de fonctionnement du programme est constant.

La touche de commande de la sortie MLP possède deux modes dans cet exemple : une courte pression et une pression prolongée. Une courte pression permet de modifier la valeur d'une unité. Une pression prolongée provoque un changement continu du compteur. Il est donc plus facile d'atteindre une valeur particulière.

(080853-I, trad. Softcraft)

### Téléchargements :

Vous trouverez sur [www.elektor.fr](http://www.elektor.fr) une page web se référant à chaque partie de cours et permettant de télécharger le logiciel nécessaire.

## Voir votre montage publié !

### Elektor est, mois après mois, à la recherche d'auteurs/concepteurs techniques freelance

Si vous

- \* avez conçu un montage innovant, ou original sous quelque angle que ce soit, et que vous aimeriez voir publié dans le magazine d'électronique le plus vendu d'Europe,
- \* possédez une expérience de conception de montages électroniques supérieure à la moyenne,
- \* avez une certaine expérience d'écriture de logiciels ayant trait à l'électronique,
- \* possédez le don d'agrémenter votre circuit d'un texte d'explication bien bâti en accentuant les originalités techniques,
- \* disposez d'un PC, d'E-mail et avez accès à Internet, ceci en vue d'une communication efficace avec les ingénieurs de notre labo,

alors, n'hésitez pas à nous contacter pour des plus d'infos sur les possibilités excitantes de voir vos projets publiés à intervalle plus ou moins régulier. N'ayez crainte, vous ne seriez pas le premier.

**Elektor – Clemens Valens, rédacteur en chef**

**Télécopie-: +31 46 4 378 161 – E-mail-: [redaction@elektor.fr](mailto:redaction@elektor.fr)**



# Pont USB-I<sup>2</sup>C

## L'I<sup>2</sup>C par le 1-Wire

Fons Janssen (Maxim Benelux)

**Les ordinateurs récents n'ont plus de ports parallèle ou série intégrés. Et bien que ces ports voient leur usage baisser en pratique, pour les concepteurs, un PC dépourvu de ces ports pose un réel problème. Afin de résoudre au moins une partie de ce problème, nous vous présentons une interface pour le bus I<sup>2</sup>C sur port USB. Cela fonctionne à merveille dans l'environnement LabVIEW.**

### Aperçu rapide du projet

- Design simple et compact
- Compatible avec Windows XP et Vista
- Compatible avec LabVIEW 7 ou plus (pour le support .Net)
- Alimenté par port USB
- Des échantillons de tous les CI peuvent être demandés à Maxim

Parfois on veut simplement que son PC communique directement avec du matériel via une interface I<sup>2</sup>C. Il y a plusieurs manières d'y arriver, mais toutes n'offrent pas les fonctionnalités désirées. Par exemple, il est possible d'utiliser le port parallèle afin d'émuler un maître I<sup>2</sup>C (voir [1]). Un problème de cette méthode est que les PC récents ne possèdent plus de port parallèle (connu aussi comme « LPT », « Centronics » ou « imprimante »). Un autre problème est que les versions actuelles de Windows comme XP et Vista n'autorisent pas un accès facile à ces vieux ports, rendant leur utilisation pénible s'ils sont intégrés au PC.

### Passez à l'USB

Une approche, espérons-le pérenne, serait d'utiliser le bus USB. Cependant, cela nécessite un pilote complexe et très probablement un microcontrôleur, lequel demandera un micrologiciel adéquat. Ce n'est pas quelque chose que l'on fait en une après-midi. Beaucoup

de solutions commerciales sont là mais leur prix est dissuasif.

La solution suivante peut être mise en œuvre rapidement et est utilisable avec beaucoup de langages (LabVIEW, Visual Basic, etc.). Elle utilise la technique 1-Wire de Maxim.

### Le matériel

Le schéma de l'interface est visible en **Figure 1**. Le circuit est relié au PC par le connecteur USB K1. L'alimentation venant du bus USB est filtrée par L1 et

L2. Cela fonctionnerait probablement sans ces inductances, mais l'auteur préfère écarter tout problème potentiel. Les lignes de données D+ et D- du bus USB sont connectées à IC2, le DS2490S+ via R2 et R3. Ce CI contient un esclave USB complet et s'occupe du processus d'énumération USB. La résistance de rappel R1 indique que IC2 est un esclave USB de type *full-speed* (c.-à-d. 12 Mbit/s).

IC2 contient également un maître 1-Wire, ainsi n'importe quel périphérique esclave peut être adressé depuis

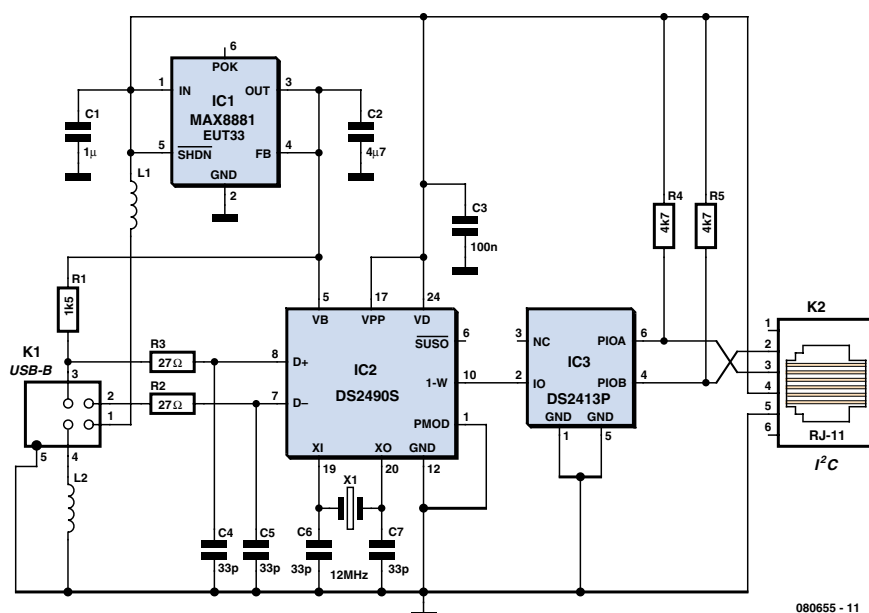
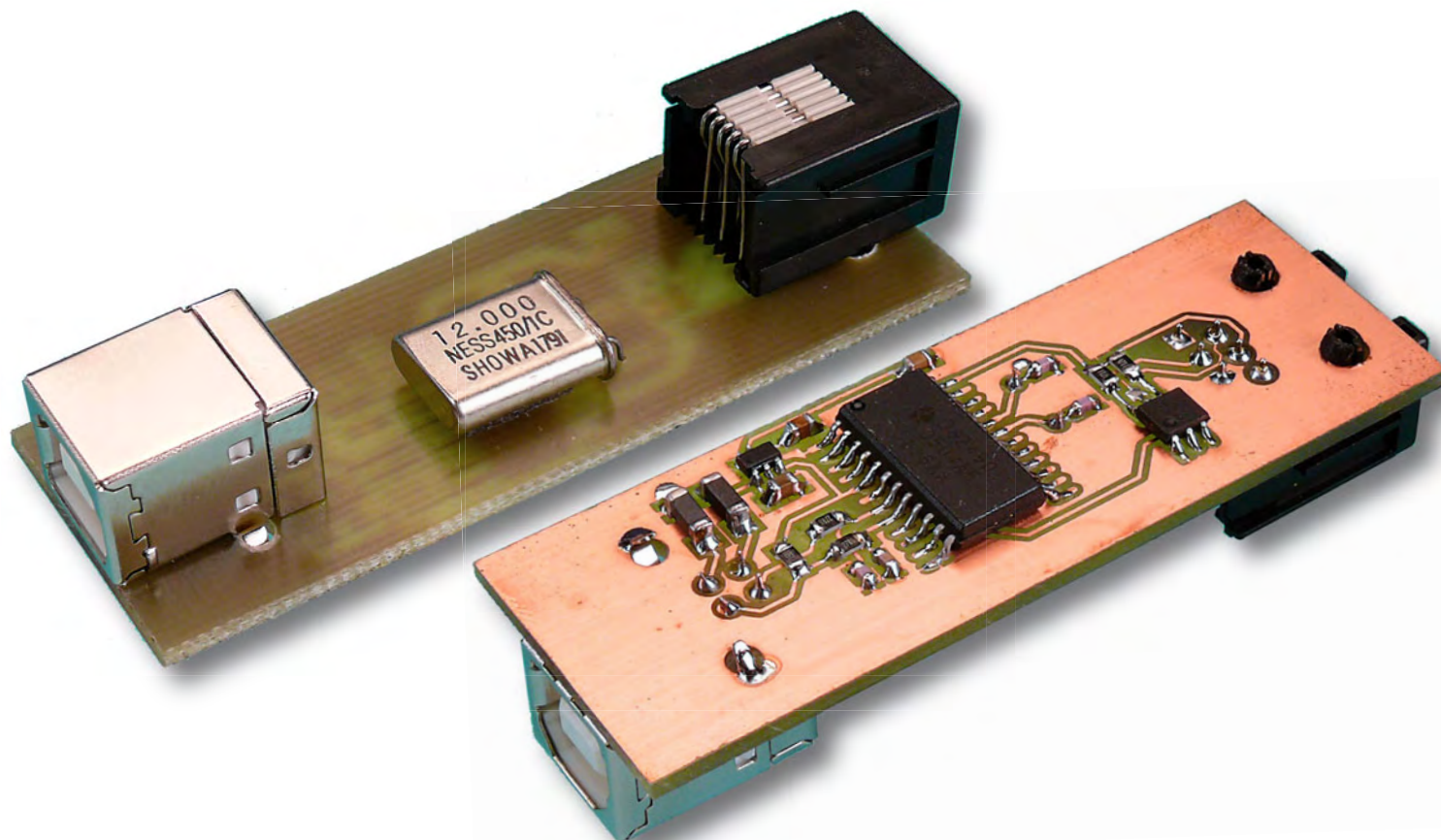


Figure 1. Ce circuit permet un bus I<sup>2</sup>C sur un port USB d'ordinateur.





le PC. Le seul esclave 1-Wire de notre circuit est IC3 (un DS2413 de Maxim), qui nous fournit deux ports bidirectionnels à collecteur ouvert (PIOA & PIOB). R4 et R5 sont les résistances de rappel de ces deux ports. C'est exactement le matériel nécessaire pour un maître I<sup>2</sup>C. Il y a un typon (voir **Figure 2**) téléchargeable gratuitement sur le site web d'Elektor [2] pour réalisation maison. Sinon, les lecteurs désireux peuvent commander un circuit imprimé chez ThePCBShop.com [3].

## Le logiciel

Sur des microcontrôleurs qui ne possèdent pas de maître I<sup>2</sup>C matériel, l'émulation d'un tel maître est une méthode éprouvée. Avec cette même technique, un maître I<sup>2</sup>C logiciel peut être créé sur cette plateforme.

Le logiciel doit pouvoir contrôler l'état des deux broches pour pouvoir générer les signaux I<sup>2</sup>C. Par exemple, pour créer une condition de *start* (début des échanges), SCL doit être à l'état haut, alors que SDA passe de l'état haut à l'état bas. Pour ce faire, SCL (PIOA) et SDA (PIOB) doivent être à l'état haut, après quoi SDA est mise à l'état bas. Donc, si le logiciel a la possibilité de contrôler PIOA et PIOB, il peut devenir un maître I<sup>2</sup>C.

Afin de changer les sorties PIO, IC3 doit d'abord être adressé par le maître 1-Wire (fonctions ROM). Dans notre cas, il n'y a qu'un esclave d'où l'utilisation de la commande *Skip ROM* (0xCC ; une liste complète des commandes

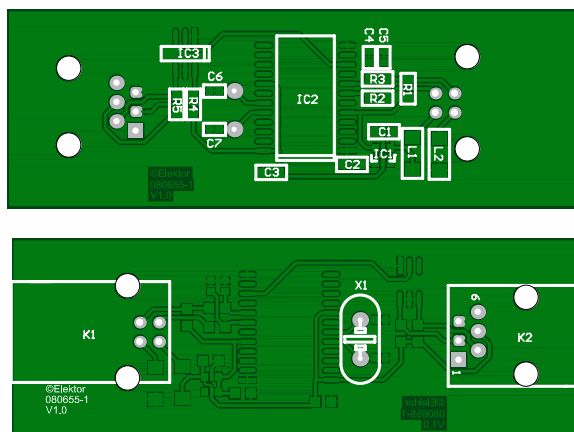


Figure 2. Le tracé est très ordonné. Le typon peut être téléchargé sur [www.elektor.fr](http://www.elektor.fr).

pour plusieurs circuits se trouve en [4]) qui permet d'éviter la complexe procédure d'adressage. Puis la commande

0x5A est envoyée pour autoriser l'écriture vers les verrous des sorties PIO (le jeu de commandes et les graphes

## NOMENCLATURE

### Résistances

(toutes en format 0805)  
R1 = 1kΩ5  
R2, R3 = 27 Ω  
R4, R5 = 4kΩ7

### Condensateurs

C1 = 1 μF (0805)  
C2 = 4μF7 (0805)  
C3 = 100 nF (0805)  
C4 à C7 = 33 pF (0603)

### Inductances

L1, L2 = BLM21PG221SN1J (Murata)  
(220 Ω à 100 MHz)

### Semi-conducteurs

IC1 = MAX8881EUT33+ (Maxim)  
IC2 = DS2490S+ (Maxim)  
IC3 = DS2413P+ (Maxim)

### Divers

K1 = Connecteur USB, montage sur circuit imprimé, type B  
K2 = connecteur RJ-11, montage sur circuit imprimé  
X1 = quartz 12 MHz  
Circuit imprimé, réf. 080655-1, sur [www.ThePCBShop.com](http://www.ThePCBShop.com)

de fonctionnement sont dans la fiche technique du DS2413 [5]). Il y a un registre 8 bits dont les deux bits de poids faible sont connectés à PIOA et PIOB (les six autres n'étant pas utilisés). Pour changer leur état, les données sont envoyées deux fois : une fois normalement et une fois en logique inversée. Le DS2413 acquitte la réception en retournant 0xAA au maître puis l'état des sorties. La séquence peut être répé-

Tableau 1. Génération d'une commande I2C start		
Mode du Maître	Donnée	Commentaires
TX	(Reset)	Reset du bus
RX	(Presence )	Le DS2413 est présent
TX	0xCC	Skip ROM
TX	0x5A	Écriture du registre PIO
TX	0xFF	SCL=1 & SDA=1
TX	0x00	Données inversées
RX	0xAA	Données reçues
RX	0x0F	Retour de l'état des sorties
TX	0xFD	SCL=1 & SDA=0
TX	0x02	Données inversées
RX	0xAA	Données reçues
RX	0x2D	Retour de l'état des sorties

montrons comment un maître I2C peut être implémenté dans LabVIEW, en utilisant .NET. Cela fonctionne seulement si les pilotes 1-Wire sont installés avec le support .NET. Aussi, il est nécessaire d'installer « Microsoft .NET framework » et « Visual J# redistributable package » d'abord (voir [6]).

Les instruments virtuels

L'instrument virtuel (IV)

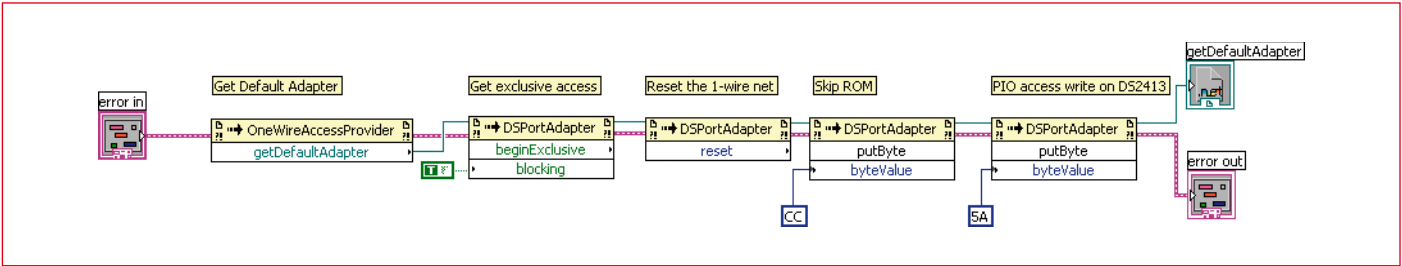


Figure 3. L'instrument Virtuel « I2C Initialize » initialise le DS2413, après quoi il est prêt à recevoir des données d'E/S.

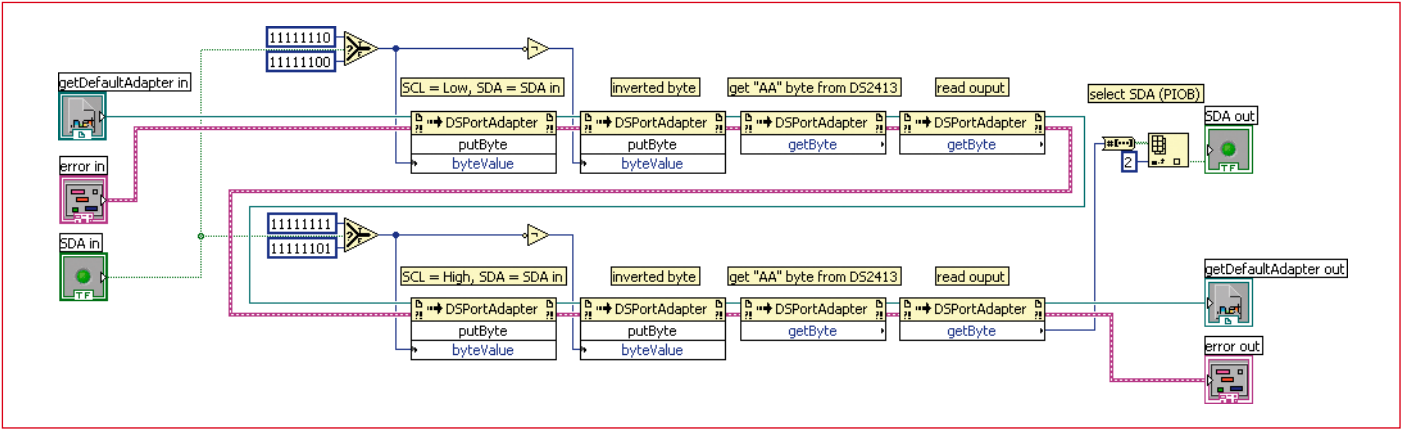


Figure 4. Le VI « I2C clock » fait transiter SCL de l'état bas à l'état haut. SDA peut être forcé à l'état haut ou bas afin d'effectuer une écriture « I2C write » ou lu lors d'une lecture « I2C read ».

tée, ou le maître peut stopper la communication par un reset. Un total de quatre octets est transféré pour chaque changement des sorties (si les commandes Skip ROM et 0x5A ne sont pas prises en compte). Pour générer la commande I2C start, la séquence du Tableau 1 doit être exécutée. Pour d'autres événements, des séquences similaires peuvent être utilisées (stop, envoi d'octet, acquittement, etc.).

Dot Net

Maxim offre des pilotes pour le DS2490 (DS9490) ainsi qu'un kit de dévelop-

pement (SDK) 1-Wire pour Windows. Ce SDK supporte la plateforme .NET de Microsoft. À titre d'exemple, nous

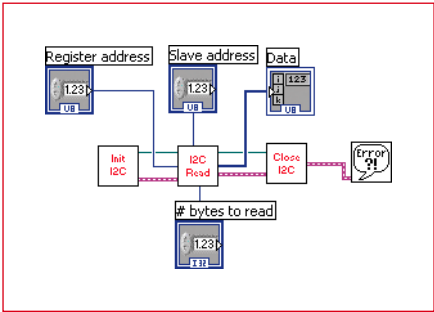


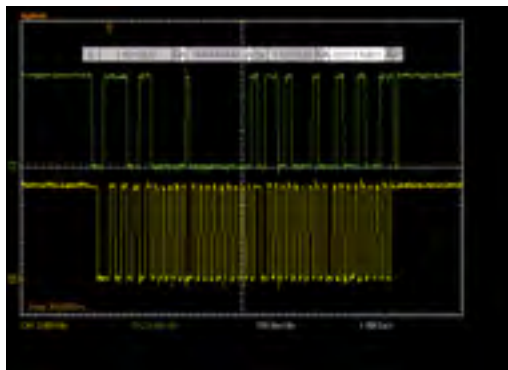
Figure 5. Exemple : comment lire un registre.

LabVIEW « I2C Initialize » (la Figure 3 en montre le schéma bloc) initialise le DS2490 et donne à LabVIEW l'accès exclusif au bus 1-Wire afin d'empêcher d'autres applications d'y accéder. Puis, le maître 1-Wire procède au reset du bus et envoie la commande Skip ROM (0xCC) suivie d'une commande d'écriture des PIO (0x5A). Le DS2413 est maintenant prêt à recevoir des données pour les PIO. L'IV « I2C clock » (Figure 4) change l'état de SCL de bas vers haut ; SDA peut alors être mise à l'état haut ou bas, l'IV renvoie enfin l'état de SDA. Un octet complet peut être écrit en appelant cette fonction huit fois d'affilé (une fois par



bit). Si l'esclave doit renvoyer des données, SDA est placée à l'état haut par le maître. L'esclave peut alors imposer un état bas s'il désire transmettre un '0' (la sortie PIOB est de type collecteur ouvert). Puisque le DS2413 renvoie automatiquement l'état des sorties mises à jour, aucune action de lecture n'est nécessaire (voir les IV « Send I<sup>2</sup>C byte » et « Get I<sup>2</sup>C byte » parmi les téléchargements supplémentaires [2]). Les IV « I<sup>2</sup>C start », « I<sup>2</sup>C stop », et « I<sup>2</sup>C acknowledge » — aussi en téléchargement — utilisent la même structure pour générer les signaux SDA et SCL adéquats. En combinant ces IV, il est possible de programmer une séquence de lecture ou d'écriture I<sup>2</sup>C complète.

Sur la **Figure 5** nous voyons un exemple de session de communication, dans laquelle le registre 0 d'un DS1337 (horloge temps réel I<sup>2</sup>C) est lu. Le DS1337 répond 0x39, c'est le contenu



**Figure 6.** Lecture du registre 0 du DS1337. S=start, A=acquiescement, Sr=start répété, et P=Stop. Les échanges de maître à esclave apparaissent grisés, les autres, blancs.

du registre comptant les secondes. Si l'on choisit le registre 0x00, l'adresse d'esclave 0xD0 et que l'on lit un octet, les signaux générés par le circuit sont semblables à ceux de la **Figure 6**. A cause du nombre important de transmissions qu'il faut sur le bus 1-Wire pour générer les signaux I<sup>2</sup>C, la

vitesse à laquelle ils sont générés est très limitée (de l'ordre de 20 bits/s). Cependant, la solution est fonctionnelle. Les IV utilisés sur les figures peuvent être téléchargés sur le site d'Elektor. Des échantillons de tous les CI utilisés sont disponibles via le service échantillons de Maxim. Il est à noter que le DS2490 n'est pas recommandé pour de nouveaux designs, mais que les échantillons resteront disponibles quelques temps.

(080655-1, trad. Kévin PETIT)

## Liens Internet

- [1] [www.maxim-ic.com/appnotes.cfm/an\\_pk/3230](http://www.maxim-ic.com/appnotes/cfm/an_pk/3230)
- [2] [www.elektor.fr/080655](http://www.elektor.fr/080655)
- [3] [www.thepcbshop.com](http://www.thepcbshop.com)
- [4] [owfs.sourceforge.net/family.html](http://owfs.sourceforge.net/family.html)
- [5] [datasheets.maxim-ic.com/en/ds/DS2413.pdf](http://datasheets.maxim-ic.com/en/ds/DS2413.pdf)
- [6] [pdfserv.maxim-ic.com/en/an/AN155.pdf](http://pdfserv.maxim-ic.com/en/an/AN155.pdf), pages 16-17.

Publicité

# Kit d'évaluation professionnel pour ARM

**NOUVEAU**

**Faites une économie de 135 € !**

- Solution idéale pour aborder, à peu de frais, le développement à base de processeur ARM.
- Idéal aussi pour réaliser rapidement des prototypes de systèmes électroniques à base de noyau ARM 7, le contrôleur extrêmement rapide d'Atmel.
- Ensemble compatible avec le reste des modules de la famille E-blocks.

**elektor**  
SHOP



**Offre exclusive sous**  
**[www.elektor.fr/arm-kit](http://www.elektor.fr/arm-kit)**  
**et nulle part ailleurs !**  
**Commandez et faites des économies !**

Voici comment économiser 135 € par rapport à la vente au détail. Le kit d'évaluation professionnel comprend le matériel et le logiciel suivants :

Flowcode Professional for ARM (CD-ROM)	179,95 €
Programmeur ARM	133,95 €
Platine LCD	28,95 €
Platine à LED	21,95 €
Platine à commutateurs	21,95 €
Alimentation	17,85 €
Câble USB	5,40 €

**Valeur totale 410,00 €**

**Prix spécial pour les lecteurs d'Elektor 275,00 €**

**franco de port !**

Commandez sur **[www.elektor.fr/arm-kit](http://www.elektor.fr/arm-kit)**

# Décodeur d'éclairage

## Compatible Märklin et programmable

Graham Guthrie

Parmi les sujets du modélisme ferroviaire qui n'ont jamais satisfait totalement les amateurs figure l'éclairage des voitures. Jusqu'à présent. La commande d'éclairage décrite dans cet article est la combinaison d'une barrette à LED CMS et d'un PIC12F683 pour la compatibilité avec le système Märklin. Les LED permettent l'adaptation à différentes couleurs en même temps que trois longueurs de platine différentes permettent au projet d'équiper tous les types de voitures courants. L'adresse du contrôleur peut lui être assignée sans démontage de la voiture.

La commande de l'éclairage des voitures miniatures a toujours été compliquée, parce que le décodeur était difficile à cacher et que son adresse n'était pas facile à fixer ou à changer.

L'auteur a relevé le défi de concevoir une solution qui réponde à cette liste de critères :

- Coût intéressant par rapport aux modèles du commerce.
- Adaptation facile à différentes couleurs.
- Commande individuelle de l'éclairage des voitures.
- Commande des feux de queue.
- Changement d'adresse des décodeurs sans ouverture des voitures.
- Ligne d'alimentation unique pour relier les voitures.
- Élimination du clignotement dû au déséquilibre du signal de voie Märklin.
- Différentes longueurs pour équiper tous les types de voitures courants.

Le résultat est une barrette d'éclairage de voiture avec son contrôleur incorporé. La platine utilise un microcontrôleur PIC12F683 à 8 broches pour décoder le signal Märklin et commander l'intensité des LED. Ce microcontrôleur contient un modulateur de largeur d'impulsion (PWM) et se trouve à bas prix, ce qui garde à l'ensemble un prix abordable.

### LED, couleurs et taille des platines

En principe, toute LED CMS (montage en surface) peut convenir. Au moment de les acheter, assurez-vous qu'elles sont spécifiées pour au moins 100 mcd (millicandelas) avec un angle de vue de 110° au moins. Les LED blanches en boîtier 0805 avec un débit de 140 mcd et un angle de vue de 140° fonctionnent très bien. Pour un éclairage jaune (jaunâtre), les LED en boîtier 1206 sont aussi un bon choix. Expérimentez avec différents types jusqu'à trouver celui qui donne la couleur la plus convenable à votre goût.

La platine est disponible en trois longueurs pour se loger dans différents types de voitures. La platine courte a une longueur de 110 mm et quatre LED d'éclairage ; la moyenne mesure 190 mm et loge huit LED ; enfin la longue mesure 230 mm et porte dix LED. Toutes les platines sont en simple face et tous les composants sont montés en surface.

### Schéma

Comme le montre la **figure 1**, la tension d'alimentation du système Märklin arrive par les rails sur le pont de diodes D1 à D4 via PC1 (Märklin ROUGE) et PC5 (Märklin BRUN). La sortie du pont est shuntée par une diode zener de 27 V, D2, destinée à protéger le régulateur des pointes de tension excé-

sives qui pourraient se présenter au long du réseau de voies. Le condensateur C3 de 22  $\mu$ F agit comme réservoir pour éviter que le microcontrôleur soit remis à zéro par des périodes courtes de manque de tension. Cette configuration de l'alimentation élimine aussi le clignotement des LED dû au déséquilibre du signal de voie Märklin. La tension continue brute de 16 à 19 V en sortie du pont alimente le régulateur 5 V IC1, qui alimente à son tour le microcontrôleur PIC et le détecteur à effet Hall TLE4917, IC3.

Le détecteur à effet Hall présente une sortie basse quand il détecte un aimant à proximité. Par logiciel, il est utilisé pour démarrer la procédure de réglage de l'adresse de la platine, décrite plus loin. La sortie GP2 du microcontrôleur PIC pilote un transistor FET monté en surface (T2) de type MMBF170 pour allumer et éteindre les LED d'éclairage. Une deuxième sortie (GP4) et un deuxième FET (T1) commandent deux LED externes qui forment les feux de queue de la voiture. Les feux de queue seront connectés entre les bornes « A » (anode) et « C » (cathode) et munis de leur résistance de limitation d'intensité (par exemple 2,2 k $\Omega$ ). Si vous n'avez pas besoin de feux de queue, T1 et R9 ne seront pas montés sur la platine. Il faut noter que la figure 1 montre le schéma générique du projet. Il existe trois options de construction, à vous de choisir.



# e de voiture

## le avec un aimant !

### Caractéristiques en bref

- Compatible Märklin
- Microcontrôleur PIC12F683
- Logiciel gratuit
- Commande de luminosité en Modulation de largeur d'impulsion (PWM)
- Trois tailles de platines disponibles pour 4, 8 ou 10 LED
- Adaptable à différentes couleurs
- Changement d'adresse de décodeur « au vol »

La platine courte utilise seulement les LED D6, D8, D10 et D15.

La platine moyenne utilise seulement les LED D6, D8, D10, D15, D7, D9, D11 et D13.

La platine longue utilise toutes les LED, c'est-à-dire D6, D8, D10, D15, D7, D9, D11, D13, D12 et D14.

### Microcontrôleur PIC

Désolé si ce qui suit paraît un peu condescendant pour les virtuoses du microcontrôleur, mais le PIC12F683 utilisé dans ce projet est (1) un composant à monter en surface et (2) doit être programmé avant de pouvoir s'acquitter d'une tâche quelconque. Aussi devrez-vous soit l'acheter programmé à la boutique Elektor (e-choppe, référence **080689-41**), soit vous procurer un composant vierge chez votre fournisseur préféré et le programmer vous-même avec votre matériel et le programme fourni gratuitement sur le site Elektor (fichier compacté **080689-11.zip**) sous la forme de source en assembleur et de fichier hex. Ce logiciel a été développé avec l'environnement intégré (IDE) MPLAB de Microchip, version 3.14.

### Construction

Tous les composants sont à montage en surface et ceux qui ne sont pas

habités à les manipuler (ou à les récupérer dans l'aspirateur) seront bien inspirés de demander l'aide d'autres

modélistes, dans un club ou sur les forums d'Elektor. De même, pour les achats de composants, le nombre fait

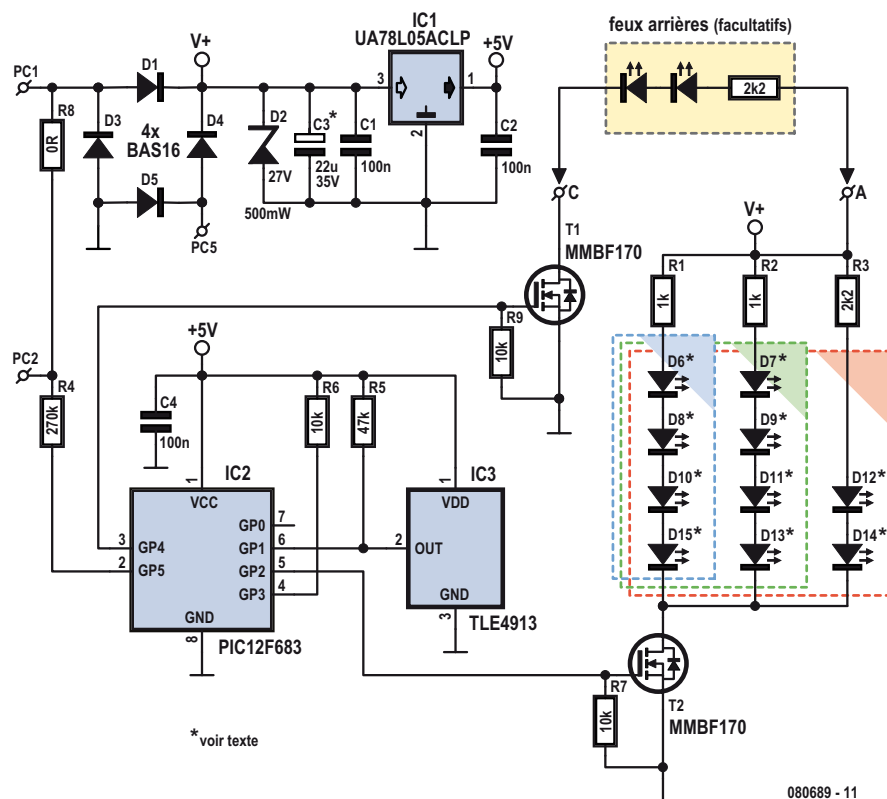


Figure 1 – Schéma de la commande d'éclairage de voiture. Il s'agit de rien de plus qu'un micro PIC et d'une série de LED commandées en modulation de largeur d'impulsion (PWM).



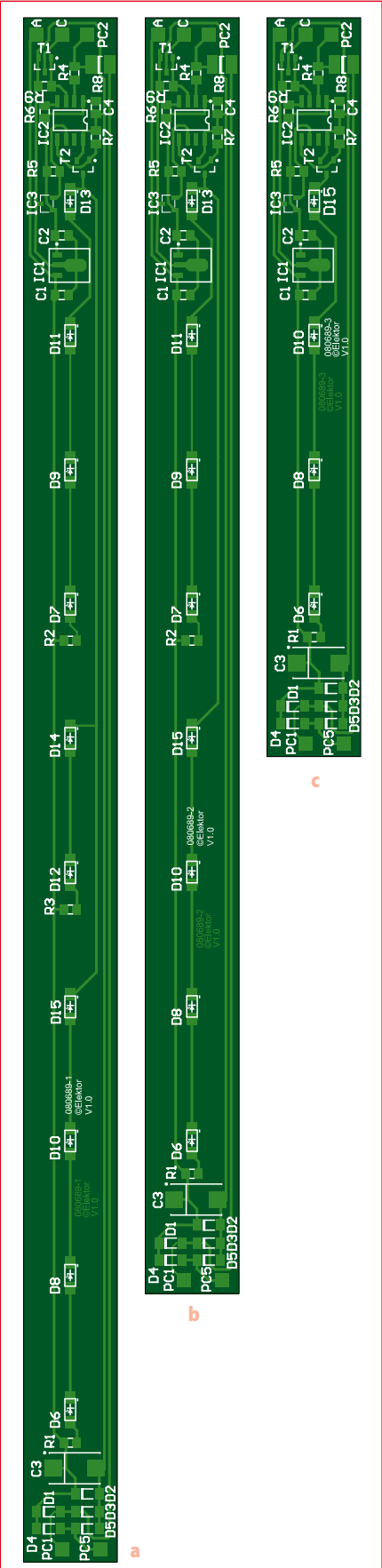


Figure 2 – La platine de commande est déclinée en trois longueurs : longue (a), moyenne (b) et courte (c) pour équiper toutes les sortes de voitures. Le dessin des trois platines est disponible gratuitement sur le site ouëbe d'Elektor.

la force. Elektor s'acquitte de sa part en proposant des kits avec circuit imprimé et microcontrôleur. Voyez l'e-choppe sur le site.

Le plan d'implantation est donné par les figures 2a (platine longue), 2b (moyenne) et 2c (courte). Les trois tailles sont disponibles à la pièce, prêtes à l'emploi, sur l'e-choppe (références 080689-1, -2, -3).

Au laboratoire Elektor, on a confirmé que le câblage de ces cartes est faisable avec une main sûre et un fer à pointe fine, sans aucun outil spécial pour les CMS. C'est probablement lié au fait que les composants sont largement espacés sur la platine. De toute façon, vous avez intérêt à commencer par le microcontrôleur programmé.

Encore quelques remarques. La taille et l'espacement des îlots de cuivre des LED permettent d'utiliser une large gamme de composants, parmi lesquels les 0805 et 1206.

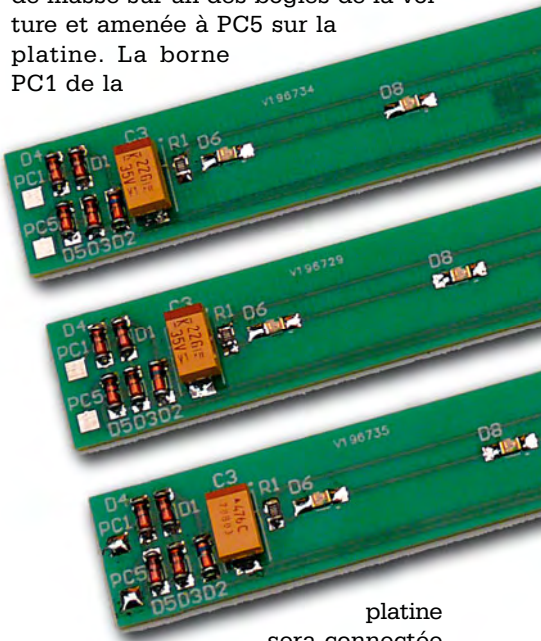
Assurez-vous de connaître, vérifier et vous faire confirmer la polarité du condensateur électro-chimique C3 avant de le monter sur la platine en toute certitude. Les condensateurs chimiques de nos prototypes ont le pôle positif (+) repéré par un trait rouge. La capacité de C3 peut être augmentée si vous voulez que l'éclairage fonctionne plus longtemps après la disparition de la tension sur les rails. Vous pouvez même envisager de le remplacer par une supercapa Goldcap.

Pour ce qui est des diodes CMS, en cas de doute mesurez la polarité ou demandez à votre fournisseur. Il est recommandé de tester la platine avant de l'installer dans la voiture. On le fera facilement en la connectant temporairement aux rails par des fils. Tous nos remerciements à M. Henk Prince

pour avoir testé les prototypes sur son réseau personnel.

Installation dans la voiture

La barrette lumineuse sera fixée au plafond à l'intérieur de la voiture avec un ruban adhésif double face. La connexion de masse (brun chez Märklin) sera reliée aux ressorts de prise de tension de masse sur un des bogies de la voiture et amenée à PC5 sur la platine. La borne PC1 de la



platine sera connectée au fil rouge du sabot central ou au couplage conducteur au bout de la voiture. La borne PC2 doit amener le courant d'un bout de la voiture au couplage conducteur à l'autre bout pour alimenter la voiture suivante en tension positive.

Définition de l'adresse de voiture

Quand on a compris que chaque élément du système Märklin – qu'il

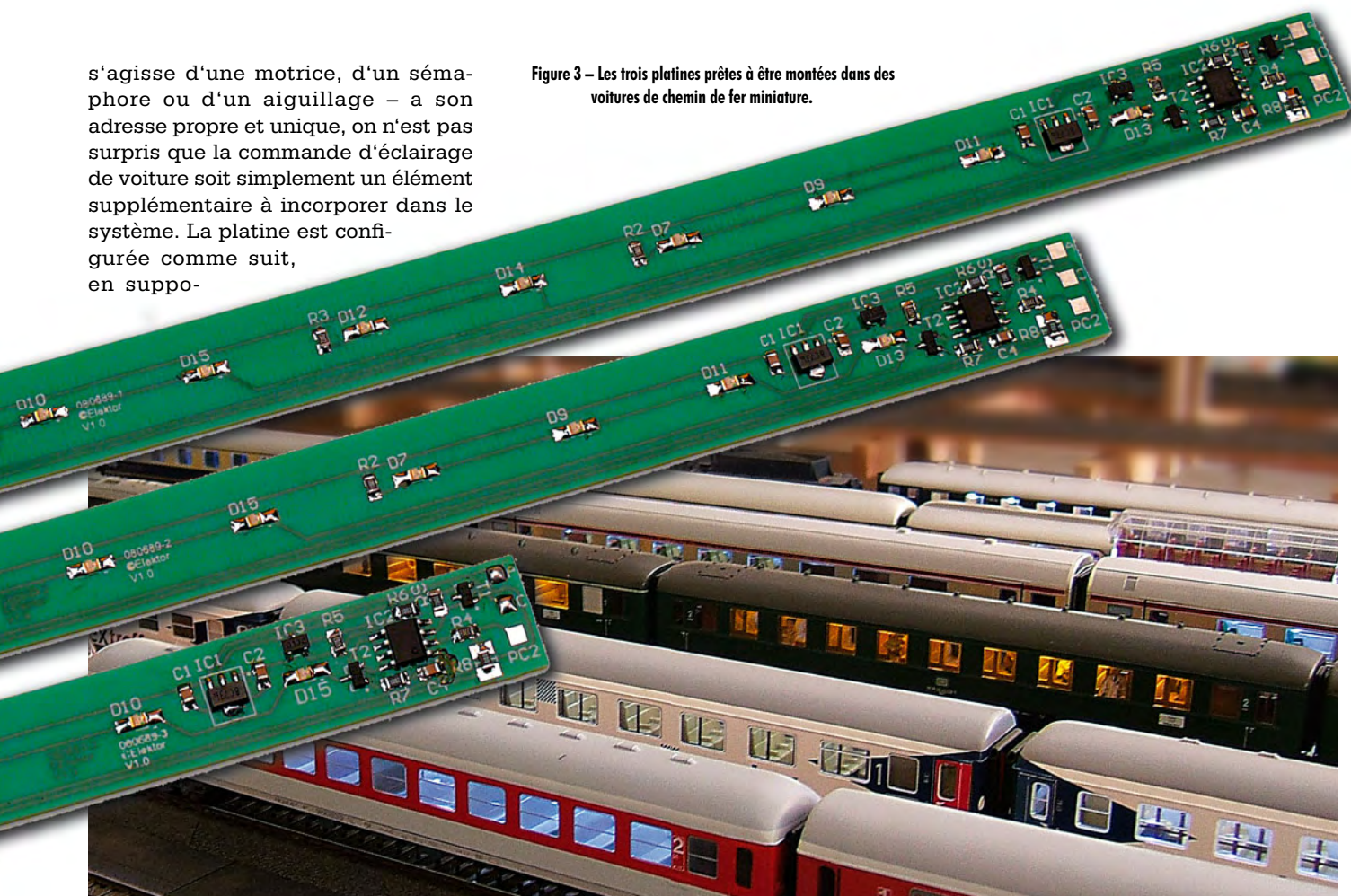
Liste des composants

- Résistances :**  
(toutes en CMS 0805 sauf R8)  
R1,R2 = 1 kΩ  
R3 = 2,2 kΩ  
R4 = 270 kΩ  
R5 = 47 kΩ  
R6,R7,R9 = 10 kΩ  
R8 = 0 Ω (pont en fil ou CMS 1206)
- Condensateurs :**  
C1,C2,C4 = 100 nF (CMS 0805)  
C3 = 22 µF / 35 V tantale (en boîtier D ou X)
- Semi-conducteurs :**  
D1, D3, D4, D5 = BAS16  
D2 = 27 V / 500 mW diode zener en boîtier SOD-80C, p.ex. BZV55C27

- D6 à D14 = LED CMS blanche ou jaune, min. 100 mcd, min. 110° (voir texte)  
T1,T2 = PMBF170 (ou équivalent du BS170 en boîtier SOT-23)  
IC1 = 78L05 (boîtier SOT-89-3, p.ex. L78L05ABUTR)  
IC2 = PIC12F683-E/SN ou -I/SN, programmé, E-choppe Réf. 080689-41  
IC3 = TLE4913 (capteur à effet Hall)
- Divers :**  
Platine, version longue (l = 230 mm), Réf. 080689-1\*  
Platine, version moyenne (l = 190 mm), Réf. 080689-2\*  
Platine, version courte (l = 110 mm), Réf. 080689-3\*
- \* choisir la longueur qui convient à la voiture.

s'agisse d'une motrice, d'un séma-  
phore ou d'un aiguillage – a son  
adresse propre et unique, on n'est pas  
surpris que la commande d'éclairage  
de voiture soit simplement un élément  
supplémentaire à incorporer dans le  
système. La platine est confi-  
gurée comme suit,  
en suppo-

Figure 3 – Les trois platines prêtes à être montées dans des  
voitures de chemin de fer miniature.



sant qu'elle fonctionne correctement,  
qu'elle est connectée et montée dans  
une voiture.

1. Poser la voiture sur les rails.
2. Remettre à zéro le module de  
commande Märklin : presser et  
maintenir les boutons *Stop* et *Go*  
jusqu'à la remise à zéro.
3. Entrer l'adresse de machine, par  
exemple 06.
4. Tenir un aimant au-dessus de  
l'extrémité de la voiture jusqu'à

ce que l'éclairage de la voiture  
clignote.

5. Retirer l'aimant, puis attendre que  
le clignotement cesse.
6. Allumer l'éclairage avec F0.
7. Régler l'intensité lumineuse au  
niveau désiré avec le cadran de  
vitesse.
8. Activer F1 pour mémoriser le  
niveau.
9. L'éclairage de la voiture clignote  
quelques fois pour signaler que le  
niveau est mémorisé.

Le décodeur est programmé main-  
tenant avec l'adresse 06. Les lumiè-  
res peuvent être allumées et éteintes  
avec les boutons F0. Si vous avez rac-  
cordé des feux de queue, vous pouvez  
les actionner avec les boutons F3. F1  
n'a plus de fonction après le réglage  
de l'intensité lumineuse pendant la  
programmation.

Amis ferroviaphes, envoyez-nous des  
photos de vos voitures équipées de cet  
éclairage !

(080689-I, trad. Brodier)

## Configuration du programmeur

Device: PIC12F683 SOIC-8

Oscillator internal RC, no clock

Watchdog: enabled

Power Up Timer: enabled

CPD: disabled

Brown out: enabled

SBORN: disabled

MCLR pin: enabled

Internal External Switch Over: disabled

Fail-Safe Clock Monitor: enabled

## Au sujet de l'auteur

Graham Guthrie a travaillé pendant 31 ans  
comme ingénieur commercial sur gros sys-  
tèmes pour IBM.

Il a pris une retraite anticipée fin 1999 et  
s'est mis au modélisme ferroviaire pour  
s'occuper. Il construit maintenant pendant  
ses loisirs des décodeurs et contrôleurs di-  
vers pour la confrérie locale des modélistes.

Il s'occupe aussi d'aider les autres amateurs  
à automatiser leurs réseaux.

Courriel : [grahamg7@telkomsa.net](mailto:grahamg7@telkomsa.net)





## Pilote de LED de puissance PR4101 à gradateur

## Burkhard Kainka

De nombreux lecteurs d'Elektor ont déjà fait la connaissance de PREMA Semiconductor [1], le fondeur de Mayence, grâce à son pilote de LED 4401 [2], supplément au numéro de septembre 2007. Il permettait d'alimenter une LED blanche sur une seule pile. Avec le PR4101, PREMA a, depuis, développé un intéressant circuit intégré pour LED à haut rendement.

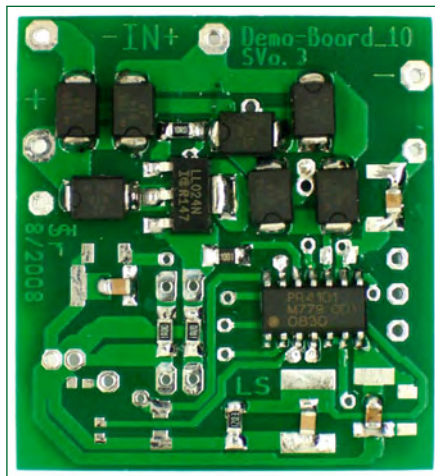
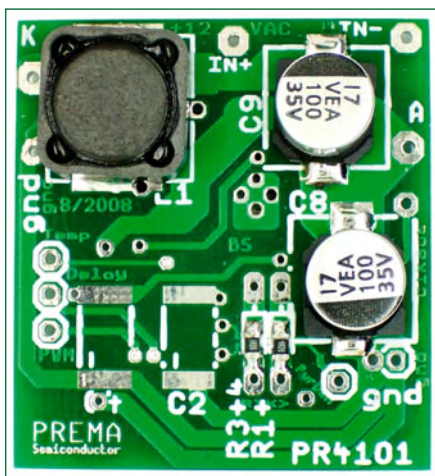
Le PR4101 est un pilote de LED abaisseur (step-down) prévu pour des tensions comprises entre 7 V et 40 V. Il fournit à une ou plusieurs LED câblées sur ses sorties un courant constant, même quand sa tension d'entrée vient à fluctuer. La durée de la LED gagne au courant continu lissé dans une large mesure. Le circuit se complète à l'extérieur par le câblage d'un FETMOS de puissance, d'une bobine torique, d'une diode Schottky ainsi que de quelques résistances et condensateurs.

Le courant de sortie peut dépasser 1 A. Le PR4401 convient très

bien à des applications alimentées par batterie (de véhicules)

lampes halogènes de 12 V alimentées par des transformateurs,

(comme pour les lampes halogènes). Ce n'est pas tout : le cir-



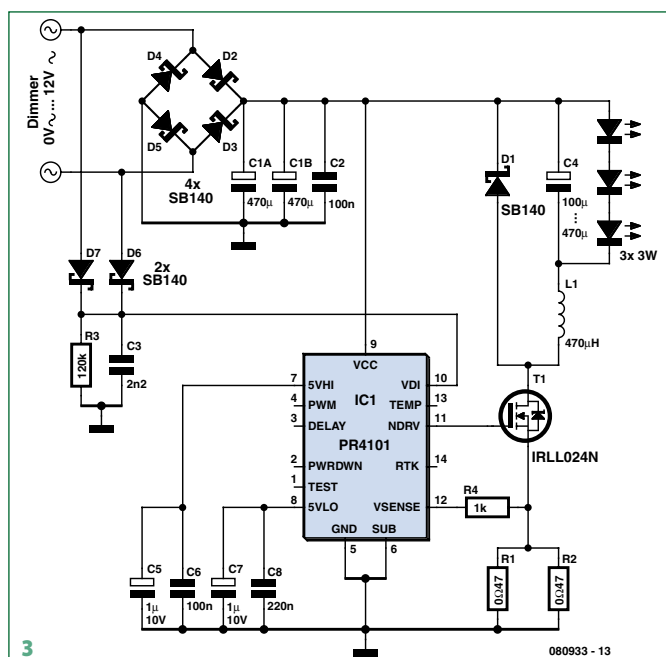
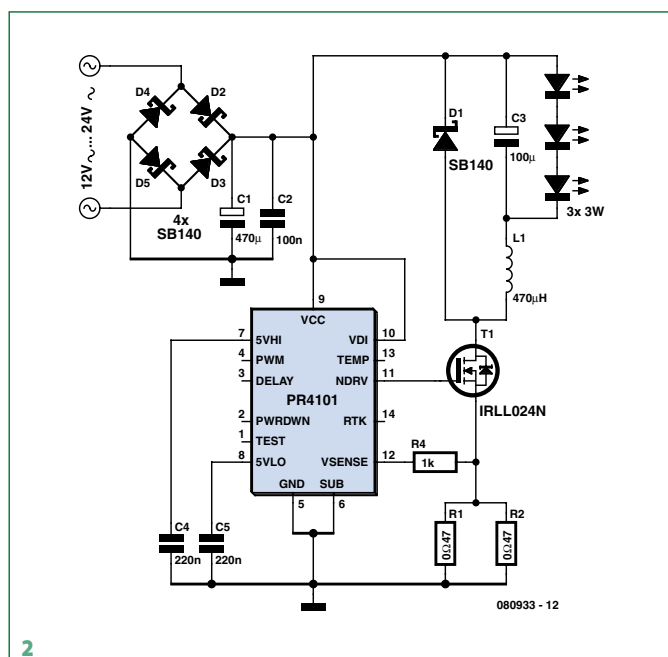
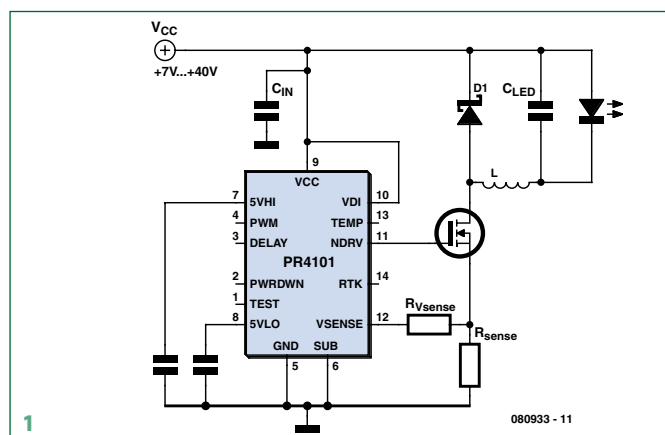
cuit intégré dispose d'une protection contre l'échauffement et protège également les LED grâce à une sonde de température externe qui rabat le courant aux températures élevées. Une entrée de commande permet d'éteindre (à l'aide d'une télécommande, par exemple) et de ne plus laisser passer qu'un courant d'attente inférieur à 35  $\mu$ A. Il est possible de ré-

de 12 V ou de 24 V. Le circuit intégré est, en outre, conçu pour permettre le remplacement de

par des lampes à LED, une commande par angle de phase permettant de faire varier leur flux

gler le courant de LED par l'intermédiaire d'une résistance de détection externe en appliquant un signal d'impulsions modulées en durée sur une entrée PWM. Le rendement du convertisseur, pour une application typique alimentée sous 12 V avec trois LED de 3 W en série, dépasse 90%. Il est encore supérieur à 60% dans le pire des cas (40 V de tension d'entrée et quelques LED de 3 W).

La **figure 1** présente le montage de base du PR4401. Le réglage du courant de LED est commandé par la résistance de détection du courant  $R_{\text{sense}}$ , avec une résistance  $R_{\text{Vsense}}$  fixe, de 1 kΩ. La pre-



mière se calcule comme suit :

$$R_{\text{sense}} = 0,2 \text{ V} / I_{\text{LED}}$$

Ce qui donne, par exemple, comme valeurs moyennes  $0,27 \Omega$  pour des LED de 3 W ( $I_{\text{LED}} = 750 \text{ mA}$ ) ou  $0,56 \Omega$  pour des LED de 1 W ( $I_{\text{LED}} \approx 350 \text{ mA}$ ).

La **figure 2** présente un montage d'application de connexion directe à un transformateur de lampe halogène – qu'il s'agisse d'un transformateur courant à noyau de fer doux ou d'un transformateur électronique (voir Elektor n°366, p. 82). En temps normal, la tension alternative fournie est de 12 V, le montage peut toutefois fonctionner à des tensions plus élevées (maxi. 24 V). Grâce au pont redresseur à diodes Schottky et au condensateur de

charge de 470  $\mu\text{F}$ , la tension est suffisante, même pour une tension alternative de 12 V à l'entrée, pour alimenter jusqu'à trois LED de 3 W en série. La résistance de détection de courant de  $0,47 \Omega / 2 \approx 0,24 \Omega$  est obtenue par le groupement en parallèle de R1 et R2 et le courant de LED est de l'ordre de 850 mA.

Le montage de la **figure 3** se distingue de celui de la figure 2 par un redresseur supplémentaire (D6/D7). Il permet de faire varier le flux lumineux à l'aide d'un gradateur. Vous pouvez l'utiliser aussi bien avec un gradateur classique à thyristor ou triac qu'avec un gradateur électronique d'un autre type ou un transformateur électronique. Le PR4401 fonctionne aussi bien avec un gra-

dateur commandé par angle de phase en avance qu'en retard. Pour ce faire, l'entrée de détection de manque de tension VDI du circuit intégré est reliée à la tension alternative redressée par D6/D7, mais non lissée, ce qui permet de détecter le « rapport cyclique » de la tension d'entrée à « phase découpée » et de mettre le convertisseur sous et hors tension de façon synchrone. Le gradateur commande alors le flux lumineux des LED comme il le ferait pour une lampe à incandescence.

Sur la fiche technique du PR4401 [3], PREMA décrit une carte de démonstration qui permet d'essayer toutes les possibilités du circuit intégré PR4101. Citons, par exemple : une me-

sure de température interne, une compensation en température du courant des LED et une fonction d'ouverture « en fondu » (*soft start*). Le circuit de la *demoboard* correspond, pour l'essentiel à celui de la figure 3 et permet de faire la démonstration d'un gradateur à commande par angle de phase du côté primaire d'un transformateur pour halogènes (230 V).

(080933-1)

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

[2] [www.elektor.fr/070100](http://www.elektor.fr/070100)

[3] [www.prema.com/pdf/pr4101.pdf](http://www.prema.com/pdf/pr4101.pdf)

## Émetteur de guitare tout simple

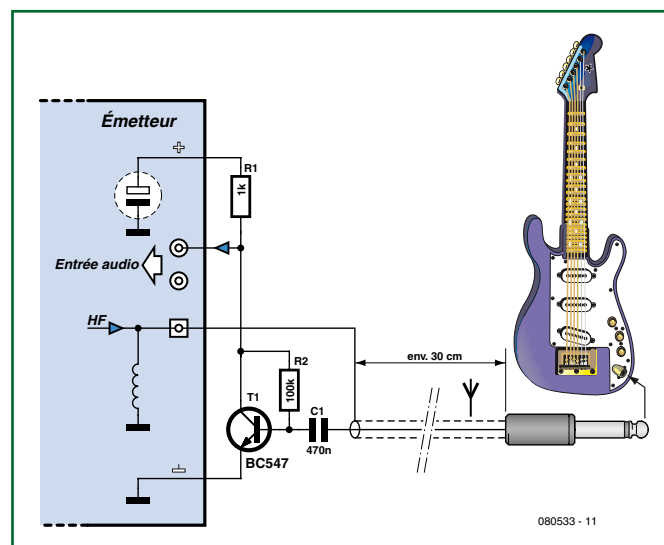
Gert Baars

Une guitare est électrique à cause de l'amplificateur de guitare avec lequel elle fonctionne obligatoirement. Elle y est, en principe, reliée par un câble, ce que l'on peut considérer comme un fil à la patte. L'amplificateur n'est pas non plus indépendant, il lui faut, la plupart du temps, ses 230 V et il est relié au réseau.

Une guitare électrique équipée d'un petit émetteur aurait bien des avantages. Elle pourrait, par exemple, se faire entendre par l'intermédiaire d'un tuner-amplificateur FM. Le câble comme l'amplificateur seraient alors superflus. Le guitariste sort avec sa radio FM portable, voire son radiocassette dans la rue, et peut réjouir les passants sans être obligé de demander un branchement au réseau.

Concevoir un petit émetteur à cet effet ne semble absolument pas nécessaire. On trouve actuellement, dans le commerce, plusieurs petits lecteurs « Walkman/MP3 » émetteurs qui ne coûtent pas 10 euros. Leur portée dépasse rarement 10 mètres, bien suffisants pour cette application.

Nous nous sommes ainsi procurés un de ces petits appareils, un FMtrans20 de marque König. Après lui avoir fourni des piles



et mis sous tension, nous avons pu capter la porteuse de l'émetteur sur un poste de radio. L'appareil dispose de quatre voies, il est donc pratiquement toujours possible de trouver une petite place sur la bande FM.

Un petit câble terminé par un jack stéréo de 3,5 mm sort du boîtier. Il s'agit de l'entrée audio. Des mesures montrent que le niveau du signal audio doit être d'environ 500 mV<sub>cc</sub> pour une modulation suffisante. Lorsque nous branchons directement la guitare, le son n'est suffisant que si nous montons à fond le volume

de la radio. Le bruit du modulateur est alors fâcheusement élevé. Il est donc indispensable d'introduire un pré-amplificateur pour le signal de la guitare.

Pour construire ce pré-amplificateur, nous devons opérer l'émetteur. Sur notre modèle, l'opération consiste à retirer quatre vis du compartiment des piles. Après ouverture, nous constatons, après quelques recherches, que les deux canaux audio sont fusionnés. Il s'agit donc d'un petit émetteur à un seul canal (mono). Comme le pré-amplificateur audio peut être mis hors

ou sous tension avec l'émetteur, nous pouvons utiliser la « tension de bord » de l'émetteur comme alimentation. Elle est, sur notre modèle, d'environ 2,2 volts. La tension est accessible sur la broche plus d'un condensateur électrolytique. Ces 2,2 volts sont trop limités pour un amplificateur à opérateur opérationnel. Un simple transistor suffit toutefois pour le gain dont nous avons besoin et le signal de la guitare est suffisamment modulé.

La modification finale consiste à souder le circuit de l'amplificateur par un bord contre le circuit imprimé et l'ensemble tient encore précisément dans le boîtier. Nous remplaçons le petit câble stéréo par une longueur de câble de microphone de 30 cm avec une prise de guitare (jack mono). Le blindage du câble sert simultanément d'antenne et de connexion de masse du signal de la guitare. La petite bobine couple le signal de basse fréquence à la masse tout en isolant le signal de haute fréquence d'antenne. Lorsqu'il joue, le guitariste laisse pendre le câble avec l'émetteur sous la guitare. Si cela le gêne, un ruban adhésif double-face permet de coller, au besoin, l'émetteur contre le corps de l'instrument.

(080533-1)



# Programmer en C !

## Altium Designer fait la corrélation

Volker Brandstetter (Altium Allemagne)

**On décrivait communément les FPGA dans des langages spéciaux, d'apprentissage difficiles, comme le VHDL. Pourtant, ça marche aussi en C ! Comparé à un programme séquentiel d'un microcontrôleur, une considérable accélération est possible. Nous allons montrer, à l'aide de la suite de développement Altium Designer, comment s'opère la conception d'un FPGA en langage C.**

Un FPGA est un ensemble matériel que l'on peut façonner par un logiciel. À la manière d'un programme qui s'exécute en séquence dans un microcontrôleur, l'utilisation de ce genre de puces permet de gagner beaucoup de temps. Comparé à un ASIC, un FPGA est nettement plus flexible, ce qui, lors de la conception de petits ou moyens systèmes, en termes de nombre de composants, fait réaliser de sérieuses économies.

Traditionnellement, au cours du développement d'un FPGA (mais aussi d'un ASIC) on décrit en langage spécialisé, VHDL ou Verilog, la structure et les fonctions des circuits internes de la puce au niveau du registre-transfert (RTL). Après que le programme a été simulé et vérifié, la description en RTL est converti en données GDSII pour la réalisation du masque ou en fichiers binaires pour configurer le FPGA.

La méthodologie de développement RTL, ciblée vers l'optimisation de la vitesse de traitement et la réduction des pertes d'énergie, permet alors de conformer le matériel instantanément sur la surface de silicium. Mais ceci ne revêt qu'une importance mineure, en comparaison de l'économie réalisée et du temps gagné lors du développement. Voici quelques inconvénients du processus de développement selon le concept RTL :

- Le niveau d'abstraction est plus bas : là où un programme écrit en langage de haut niveau demande 100 lignes de code, celui en RTL en réclame environ 1 000.
- Les possibilités de réutiliser du code sont rares : en RTL, la structure du système doit y être décrite explicitement.

Les facultés d'optimisation sont minces ; en RTL, l'optimisation se déroule à un niveau inférieur du système, alors qu'une optimisation au niveau du système global est plus performante, en puissance comme en coût.

### C offre des avantages

Il existe des solutions que la Synthèse « du C au matériel » (C-to-Hardware) propose. Des morceaux d'un programme en C, qui tournent peut-être déjà sur un microcontrôleur, le concepteur du logiciel peut les « fondre » dans du matériel FPGA pour accélérer le processus.

Une des suites de développement capable d'offrir ce genre de solutions, c'est Altium Designer du forgeron de silicium australien Altium [1]. Le fabricant propose également des cartes de développement qui, grâce à un très vaste assortiment de périphérie et de possibilités d'extension, peut servir de matériel de prototypage. Un exemple nous en est fourni par la Nanoboard (cf. encadré) que nous avons utilisée et testée pour illustrer cet exposé.

La Synthèse du C au matériel permet le développement de prototypes sur base de FPGA sans devoir connaître le VHDL ni le Verilog. Dans ce but, on dispose, à côté du C comme langage descriptif du matériel, d'une innombrable bibliothèque d'IP (des sections de circuits en FPGA prêtes à l'emploi), comme des cœurs de processeur logiciel (unité centrale de microcontrôleurs réalisable en FPGA), des contrôleurs de mémoire et divers composants périphériques. Dans le voisinage de la synthèse du C au matériel, Altium Designer convertit en C / C++ ISO des programmes écrits en code RTL synthétisable. Le compilateur génère le circuit au départ d'une liste d'unités fonctionnelles et de leurs propriétés (différents types d'additionneurs, d'unités arithmétiques et logiques, de MAC, de diviseurs, et ainsi de suite). En plus, il cherche à en améliorer la performance, éventuellement par la mise en parallèle de fonctions qui répondent à la description.

Les avantages de la synthèse du C au matériel se révèlent spécialement utiles quand il y a des cœurs de processeurs intégrés en FPGA. Le code C peut alors être exporté du processeur ou bien converti en RTL. On trouve en Altium Desi-

# es FPGA

## entre C et matériel

gner le compilateur pour le logiciel enfoui et la synthèse du C au matériel étroitement associés. Une sélection des fonctions qu'on attend du processeur et des circuits qui doivent y être intégrés ne demande que quelques clics de souris.

### Double accès

Il y a, en principe, deux manières de mettre en oeuvre le compilateur du C au matériel.

1. Pour le développement de blocs fonctionnels, on utilise le compilateur pour traduire en matériel des fonctions en C individuelles. On y associe le code C à l'aide de symboles. Le résultat est un bloc FPGA que l'on peut relier à d'autres blocs du circuit, lesquels peuvent par exemple être constitués de composants VHDL, Verilog ou des IP préalablement synthétisés, en définissant les entrées et les sorties (**figure 1**).

Le compilateur du C au matériel peut alors réaliser la fonction C en circuit combinatoire pur ou à cycle multiple. Le choix du type de circuit s'opère par la définition des caractéristiques des symboles du code C (**figure 2**).

Avec des blocs combinatoires, seuls apparaissent dans le symbole en code C les paramètres de la fonction C comme entrée ou sortie. Si l'on choisit un circuit à cycle multiple, Altium Designer y ajoute toute une série d'E/S indispensables (comme par exemple l'horloge et la mise à zéro) pour le câblage.

2. Quand il s'agit d'accélérer un système à processeur, la conversion en matériel s'opère à partir des fonctions logicielles choisies en code C, qui pourront être appelées par le programme enfoui, lequel s'exécute alors sur le processeur FPGA intégré. Pour réaliser une multiplication de la vitesse du matériel au profit de fonctions dont le temps d'exécution est critique, l'outil utilise le compilateur du C au matériel en relation avec le logiciel compilé enfoui (**figure 3**).

### En parallèle, ça va plus vite

Les opérations qui sont décrite séquentiellement en C mais que le matériel doit pouvoir exécuter en parallèle, le compilateur les traduit en code RTL correspondant. Pour de nombreux algorithmes utilisés par exemple en traitement d'images ou de signaux, la possibilité de mettre en parallèle les ressources matérielles en FPGA ne résulte pas seulement en un soulagement du processeur, mais aussi en une augmentation considérable de la vitesse.

Le bloc du processeur d'applications spécifiques (ASP Block) fonctionne ici comme interface entre les opérations exécutées dans le matériel et les autres, qui tournent sur le processeur du programme en C. Si une fonction matérielle est appelée par le programme enfoui, le processeur passe les valeurs des paramètres à l'ASP qui lance alors la fonction et, après récolte du résultat, arrête le programme.

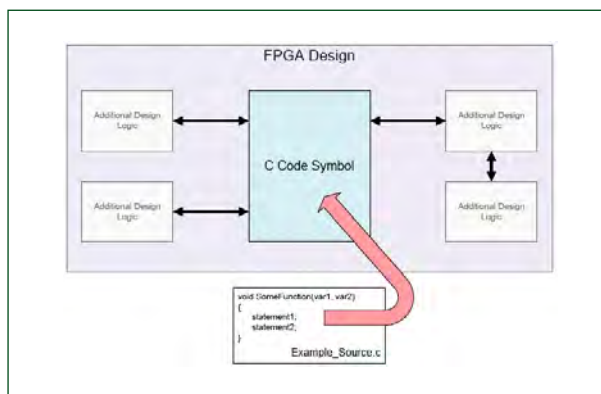


Figure 1.  
Une fonction C est associée au système par un symbole en code C et convertie en bloc FPGA.

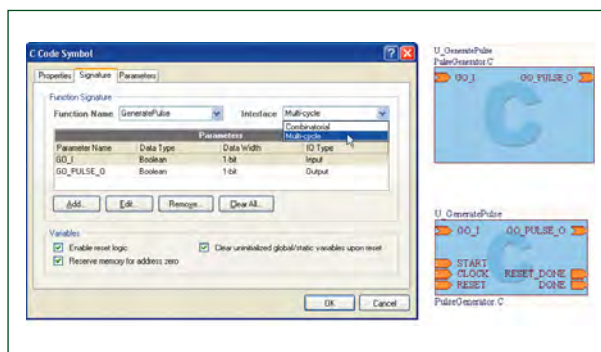


Figure 2.  
Sélection du type de circuit (combinatoire ou à cycle multiple) pour un symbole en code C.

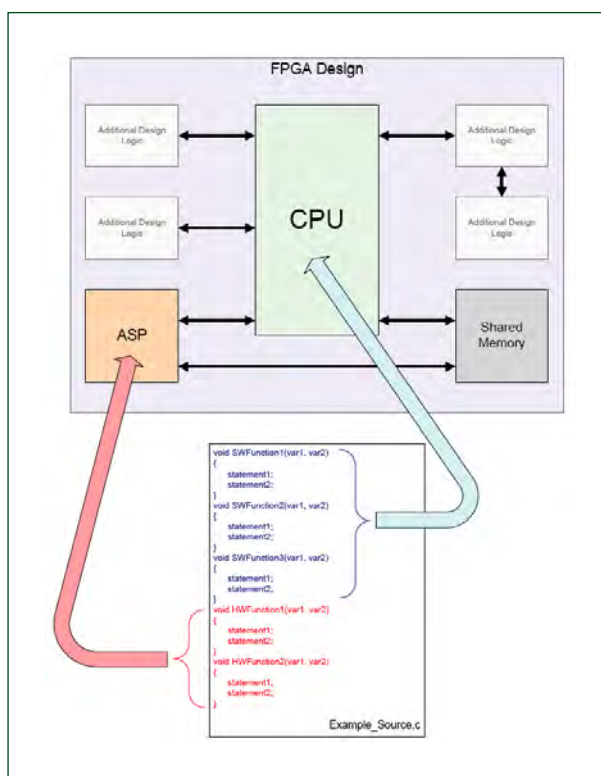
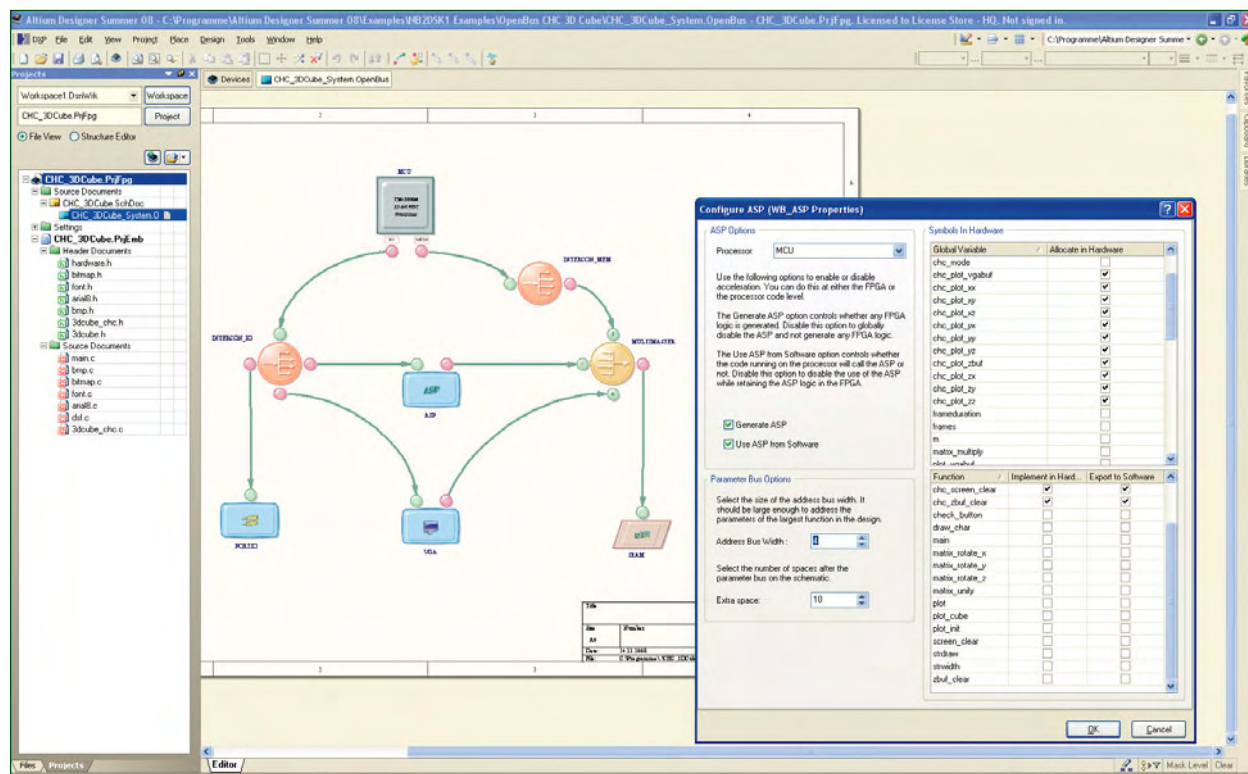


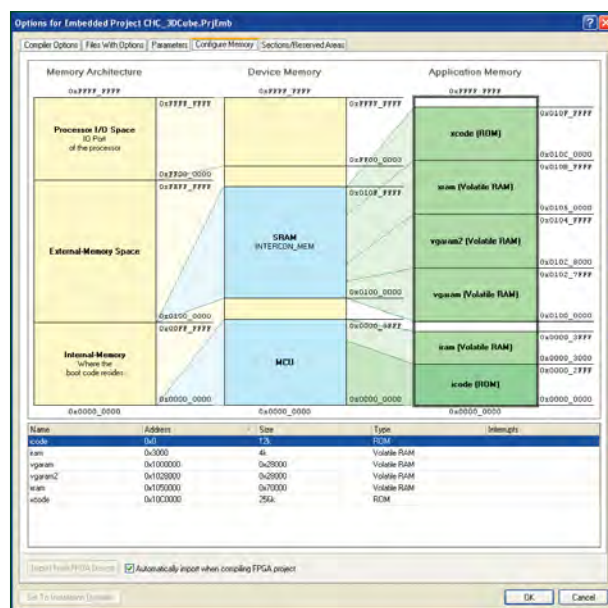
Figure 3.  
Les fonctions C dont la vitesse d'exécution est critique sont implantées dans le matériel, le reste tourne sur le processeur.

L'ASP a aussi accès à la même mémoire que le processeur, ce qui fait que les variables communes aux fonctions logicielles et matérielles restent disponibles. Le logiciel peut alors fournir à l'ASP un pointeur vers l'adresse de mémoire où se trouvent les valeurs des paramètres de la fonction. La configuration du Block ASP est simplifiée par une boîte de dialogue (**figure 4**). Au-dessus, à gauche, une liste

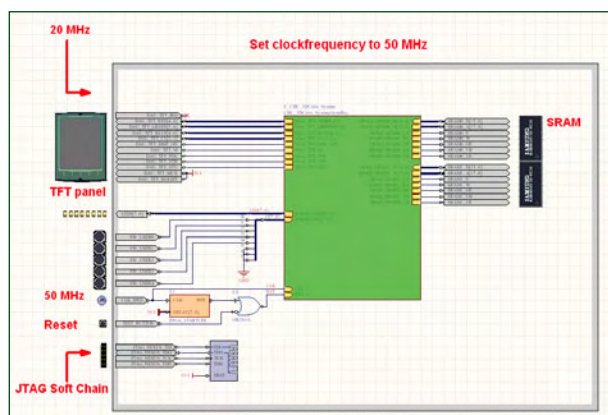




**Figure 4.**  
Configuration intuitive d'un  
bloc ASP pour la synthèse  
du C au matériel.



**Figure 5.**  
**La configuration des**  
**domaines d'adresses**  
**utilisés.**



**Figure 6.**  
**La mise en relation du**  
**FPGA avec les ressources de**  
**la plateforme NanoBoard.**

dans laquelle le concepteur détermine quelles variables globales seront introduites dans la RAM du Block ASP. L'accès à la mémoire interne de l'ASP est beaucoup plus rapide qu'à celle dont dispose le processeur. Dans une autre liste, en bas, à droite, on peut choisir les fonctions en code C à réaliser en matériel ou en logiciel.

## Le dé 3D qui roule pour vous

À présent, une démonstration de tout ceci avec un cube tridimensionnel mis en rotation sur l'écran TFT de la plateforme NanoBoard.

Pour l'inauguration du système, nous avons choisi comme exemple la récente introduction de l'OpenBus d'Altium Designer. Il suffit de sélectionner dans une liste les composants du système nécessaires pour l'application, de les mettre en place et de les relier les uns aux autres sans avoir à se soucier des détails de l'installation.

La saisie des codes source en C s'opère dans l'éditeur C intégré. Altium Designer possède un compilateur, un éditeur de liens et un débogueur pour le TSK3000, un processeur RISC à 32 bits, mais aussi pour toute une série d'autres processeurs à 8 et 32 bits.

Après avoir introduit les composants et leurs liens réciproques, la configuration du système se réalise sur les masques correspondants. À côté de la configuration de l'ASP déjà évoquée, on peut aussi adapter les domaines d'adresses de chacun des composants périphériques, par exemple, ou de la mémoire externe du processeur (**figure 5**). La configuration graphique représentée du domaine d'adresses est attribuée automatiquement au logiciel enfoui du projet à chaque changement, ce qui est un bel exemple des avantages d'un environnement de développement homogène.

Le test et la vérification du programme C et du matériel se déroulent en temps réel avec l'aide de la NanoBoard. Les ressources de la carte Nano et celles de l'interface vers les platines interchangeables sont reliées par des symboles correspondants, choisis dans une bibliothèque garnie

d'origine, aux circuits logiques en FPGA (**figure 6**). Pour l'exemple du dé 3D, à côté de l'écran TFT, l'horloge, la mise à zéro, un afficheur à LED et deux bancs de SRAM sont ainsi mis à disposition. Le document descriptif du circuit à FPGA a été produit automatiquement au moyen de symboles de blocs hiérarchisés dans le schéma et rattachés aux ressources de la NanoBoard.

## À peine constitué, ça tourne déjà !

Après avoir donné les caractéristiques complètes du système, le processus de construction commence (la **figure 7** montre le « Cockpit » correspondant). Il se passe en quatre étapes, dont la première est la synthèse du C au matériel pour les fonctions demandées (étape de compilation) et tout de suite après, la compilation du restant des codes C. Dans l'étape suivante, la synthèse RTL, le placement et le routage fournissent alors un fichier binaire pour la programmation du FPGA. Le programme en C peut, à travers la chaîne dite Soft-JTAG, être débogué pendant qu'il tourne dans le coeur du processeur enfoui dans le FPGA. Dans notre exemple, les boutons situés au bas de l'écran TFT permettent de choisir si l'on veut ou non la présentation du calcul des fonctions C matérielles synthétisées pour chaque face du dé 3D qui sont appelées par le programme (**figure 8**).

L'expérience montre que le processeur ne peut calculer qu'environ 2,7 faces par seconde. En utilisant l'ASO, c'est-à-dire l'exécution des fonctions du matériel pour lesquelles la vitesse est critique, le traitement est accéléré d'un facteur 10, ce qui donne à peu près 28 faces par seconde !

(080625-1)

## Références

- [1] [www.altium.com/products/altiumdesigner](http://www.altium.com/products/altiumdesigner)
- [2] [www.altium.com/products/thenanoboard](http://www.altium.com/products/thenanoboard)

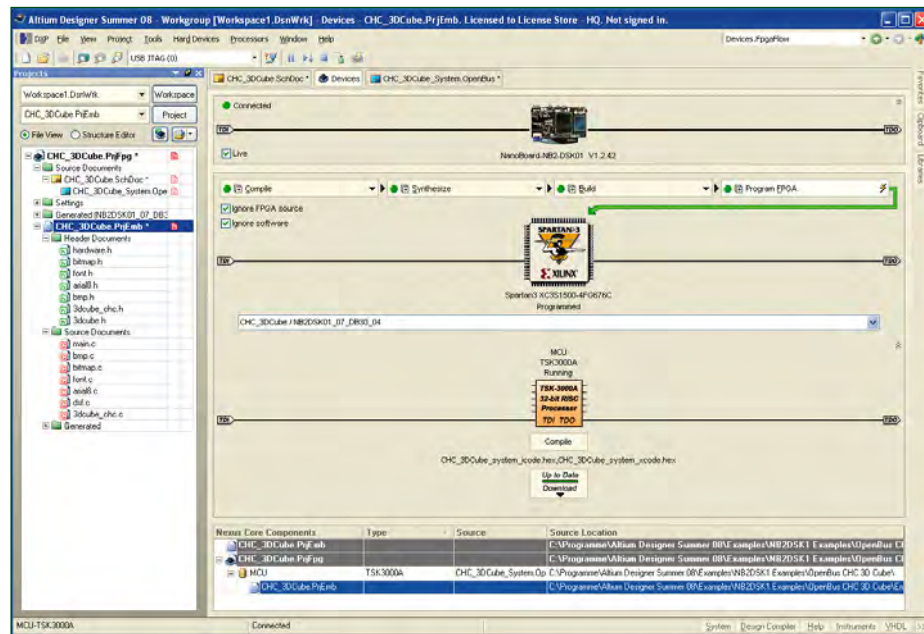


Figure 7.  
Vue des appareils de l'Altium Designer, le Cockpit pour le processus de construction et la vérification en temps réel.

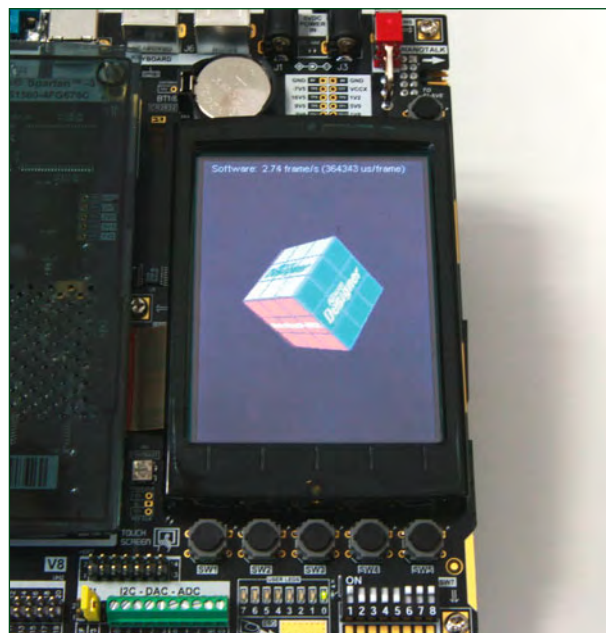


Figure 8.  
Utiliser la synthèse du C au matériel permet ici une accélération d'un facteur 10 !

## La NanoBoard

En plus du logiciel de développement électronique Altium Designer, qui prend en charge la conception du circuit, le développement du logiciel enfoui et la réalisation des solutions FPGA, Altium propose une plateforme matérielle reconfigurable, la NanoBoard [2].

Les platines de périphérie interchangeables de la NanoBoard autorisent une large palette de fonctions d'E/S et de matériel. Avec le secours des cartes filles interchangeables (Swappable Daughter Boards), il est aisé de tester différents FPGA et processeurs pour le développement d'un premier prototype.





# C plus que C

## Programmes pour PC avec .NET et C# première partie

Veikko Krypczyk

**Les électroniciens sont souvent contraints d'écrire de (petits) programmes pour PC, par exemple pour évaluer des mesures. L'infrastructure .NET (.NET Framework) pour les ordinateurs Windows s'y prête à merveille. Elle supporte divers langages, épargne le gros du travail au programmeur et – avec l'environnement de développement compris – ne coûte pas un centime. Notre petite introduction est basée sur le langage C# qui offre de nombreux avantages.**

L'électronicien actuel tend à se métamorphoser de plus en plus en un concepteur de logiciel. Il lui faut par exemple programmer un microcontrôleur. Notre électronicien doit aussi fréquemment écrire de (petits) programmes pour PC. Ceux-ci transmettent des données de commande par une interface de sortie, affichent des mesures, etc. Dans d'autres cas, il faudrait adapter ou étendre un logiciel existant. Le développement de logiciel dépend bien entendu très largement de la plate-forme choisie et du système d'exploitation. Cette série d'articles concerne la réalisation de programmes conçus au départ (et à l'arrivée) pour fonctionner avec Windows XP et Vista de Microsoft. Ces programmes reposent sur une bibliothèque d'exécution nommée « .NET Framework » (prononcer « dotnet ») contenue en standard dans Vista et disponible gratuitement pour XP. Les possesseurs d'une version de Windows plus ancienne ne repartent toutefois pas les mains vides. Les pages de support de Microsoft constituent la case départ d'une solution praticable. N'oublions pas de mentionner aussi une version source ouverte de .NET Framework. Cette infrastructure dénommée « Mono » [1] tourne même sur des ordinateurs Linux !

### Classes comme modules de programmation

L'introduction de .NET Framework a considérablement modifié le développement du logiciel Windows. Simplement dit, il s'agit d'une riche bibliothèque de classes de base. Elle offre des concepts et des solutions de base pour un grand nombre de problèmes, ce qui simplifie d'autant la tâche du programmeur. En outre, .NET Framework a coupé l'herbe sous les pieds des adeptes de la discussion (oiseuse) sur le « meilleur » langage de programmation. Le choix du langage est devenu secondaire, car les classes de base sont les mêmes et il existe même un système de types commun.

Un programme est toujours traduit dans le même langage intermédiaire, quel que soit le langage source. Ce langage intermédiaire est exécuté ensuite par l'environnement d'exécution .NET (il s'agit plus précisément d'une seconde compilation lors du lancement du programme). L'utilisation de ce « Common Intermediate Language » offre une meilleure sécurité. Il permet aussi de programmer indépendamment du système d'exploitation – le remplacement de l'environnement (léger) d'exécution suffit à assurer l'adaptation.

### C à la sauce Microsoft

Parmi les langages disponibles pour .NET Framework, on compte Visual Basic for .NET (de Microsoft) ou Delphi for .NET (d'Embarcadero Technologies). Un aperçu des langages compatibles avec .NET Framework est disponible par exemple sous [2].

Ces deux langages ont été tout au plus adaptés à .NET (VB.NET n'est en effet pas entièrement compatible avec le VB classique. Le code source de VB6 peut toutefois être adapté sans trop d'effort). Le langage C# (autrement dit : C Sharp) représente par contre une réalisation entièrement nouvelle de Microsoft. La complexité des langages C et C++ qui rebute tant de débutants n'a pas sa place dans C#. L'arithmétique des pointeurs, par exemple, haïe par de nombreux programmeurs, a raté le coche. De nombreux concepts efficaces et conviviaux provenant d'autres langages ont été introduits en contrepartie. Un autre facteur parle en faveur de ce dérivé de C. Une infrastructure .NET Framework très compacte est disponible pour la programmation des microcontrôleurs. Elle comporte même un compilateur gratuit – pour C# [3] !

Notre série ultracourte ne comporte que deux parties. Elle ne peut pas prétendre être un cours sur C#, mais peut servir au lecteur à évaluer l'opportunité d'un passage à C#. Dans



## Quelques concepts linguistiques du langage C#

Vous reconnaissez peut-être quelques éléments linguistiques de C# déjà présents dans d'autres langages. Consultez l'assistance en ligne [6] pour toute question ayant trait à la syntaxe.

1. C# distingue entre majuscules et minuscules. La variable « nombre » est différente de la variable « Nombre ».
2. La présence de // indique que la suite est un commentaire d'une ligne ; /\*...\*/ marque la limite d'un commentaire de plusieurs lignes.
3. Les principaux types de données élémentaires sont : *byte*, *int*, *float*, *double*, *decimal* et *string*.

4. Le type des données est un préfixe de la déclaration et de l'initialisation des variables :  
`int Nombre = 5;`  
`string texte = « Une chaîne de caractères »;`
5. Opérations de base : `+`, `-`, `*`, `/`
6. On dispose des boucles *for*, *while*, *do* et *foreach*. Cette dernière boucle permet d'effectuer une itération sur des objets appartenant à une collection.
7. Les structures de choix sont : *if*, *else if*, *else* et *switch* (...) *case*.
8. Les types de données définis par l'utilisateur sont déclarés par le mot clé *struct* (les classes définies par l'utilisateur sont signalées par *class*) :  
`struct Enregistrement`  
{  
    *int* age;  
    *string* nom;  
}

cette partie-ci, nous développerons un premier petit programme au moyen de l'environnement de développement Visual Studio (téléchargement gratuit). La seconde partie sera consacrée à quelques éléments de l'orientation objet qui forment la base de .NET Framework et de C# lui-même. Un exemple de graphique 2D issu de la pratique nous permettra d'aborder une application qui devrait intéresser particulièrement les électroniciens.

### C'est parti !

Il faut configurer l'ordinateur avant de procéder à la réalisation. Il est indispensable de disposer d'un compilateur C# pour traduire le code source. Un simple éditeur suffit déjà pour la saisie du code. Il est toutefois préférable et plus convivial de faire appel à un environnement de développement intégré (Integrated Development Environment = IDE). On est confronté à deux possibilités. Microsoft offre d'une part le téléchargement gratuit de l'édition « Express » de Microsoft Visual Studio. Cette version n'offre qu'une fonctionnalité limitée par rapport aux versions professionnelles. Cela se remarque à peine tant que les projets sont d'envergure réduite à moyenne. Chaque édition Express est prévue pour un langage spécifique (C#, Visual Basic for .NET ou C++). Un outil supplémentaire (Visual Web Developer Express) permet d'effectuer des travaux de développement pour le Web. Visual C# Express dans la version 2008 est le bon choix pour notre minisérie. Se référer à [4] pour des informations sur les produits mentionnés et pour le téléchargement.

Une alternative gratuite source ouverte, Sharp Develop [5], est également disponible. Elle offre aussi la possibilité de programmer en C#. Il faut que .NET Framework SDK (Software Development Kit) soit installé.

### Mon premier programme

Pour créer une nouvelle application, démarrez Microsoft Visual C# 2008 Express Edition. La page de démarrage de l'environnement d'application vous accueille. Elle permet entre autres d'ouvrir le projet le plus récent traité. Pour un nouveau projet, cliquez sur Fichier | Nouveau | Projet... Sélectionnez « Application Windows Forms » parmi les modèles proposés (figure 1). Il faut aussi baptiser le projet (Nom). Le squelette du programme pour cette application

est créé par l'IDE après la confirmation par OK. Le résultat apparaît instantanément sur l'écran. Vous doutez qu'il s'agisse déjà d'une application complète ? Faites-lui faire un galop d'essai (menu Déboguer | Exécuter ou cliquez sur la flèche verte de la barre d'outils, voir la figure 2). À vrai

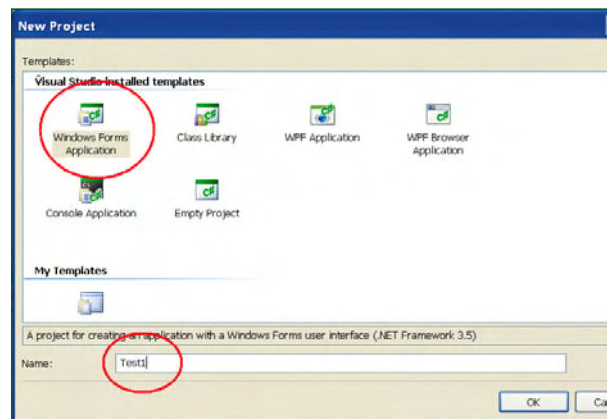


Figure 1. Création simple d'une application Windows (squelette) avec l'assistant.

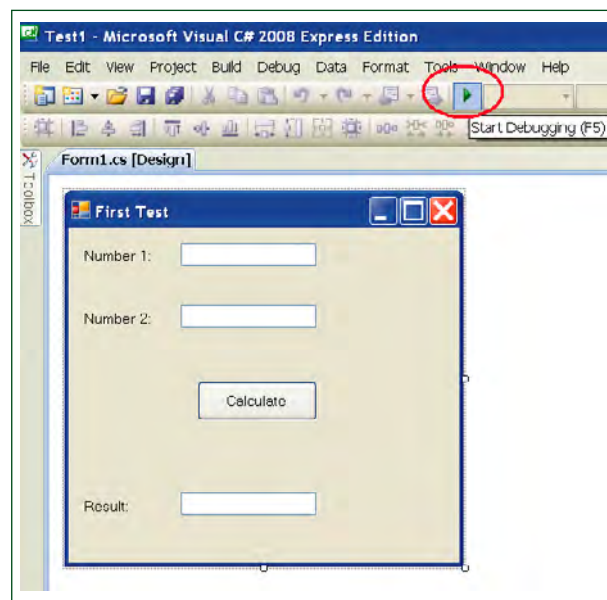


Figure 2. Démarrage d'une application en mode débogage à partir de l'environnement de développement.

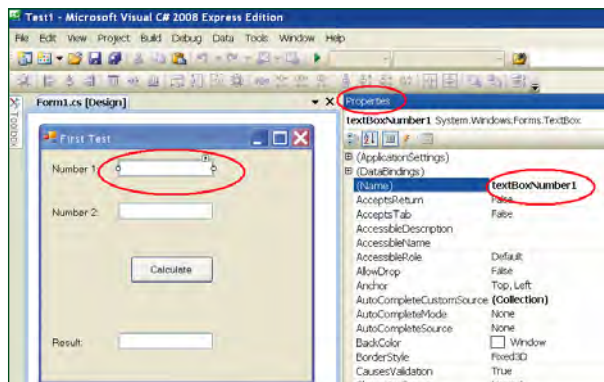


Figure 3.  
Modification aisée des propriétés de l'objet dans la fenêtre des propriétés.

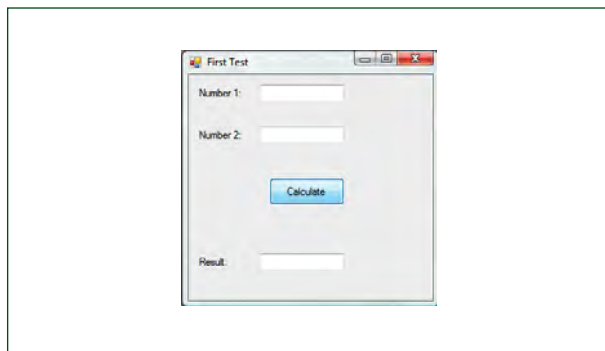


Figure 4.  
La première application (très simple) en action.

dire, l'application ne consiste à ce stade qu'en une fenêtre vide. Mais les fonctions importantes d'une application Windows sont déjà implémentées. La fenêtre peut être par exemple déplacée ou réduite et l'application fermée par le menu Système.

L'arrêt de l'application renvoie à l'environnement de développement qui nous permet d'inclure les premiers éléments

dans le modèle. Programmons le simple exemple suivant : saisie de deux nombres, leur addition et affichage du résultat dans une zone de texte. Commençons par la structure de l'interface avec l'utilisateur. Les ingrédients :

- trois zones de texte pour les nombres et le résultat ;
- trois étiquettes avec légende ;
- un bouton pour effectuer le calcul.

L'IDE permet de parvenir rapidement au but. Appelons la Toolbox (Affichage | Boîte à outils) lorsque le formulaire est actif. On peut alors choisir les éléments de l'interface utilisateur avec la souris et les disposer dans le formulaire. Les lignes d'assistance facilitent le placement. On peut traiter les propriétés d'un élément une fois qu'il est disposé sur l'interface utilisateur. Cette tâche est facilitée par la fenêtre des propriétés (presser sur la touche F4 lorsque l'objet est sélectionné). Indiquons par exemple « textBoxNumber1 » pour la propriété « Nom » de la première zone de texte (figure 3).

### Code source du bouton

L'apparence de l'interface utilisateur est définie, passons à la création du code. Le calcul doit être effectué quand le bouton prévu à cet effet a été pressé. Un clic de souris sur ce bouton quand le programme est actif provoque un « événement clic ». Le programmeur doit allouer à cet événement le code source qui sera alors exécuté, en l'occurrence l'addition de deux nombres. L'IDE assiste le programmeur en créant automatiquement le squelette de l'événement clic d'un bouton. Il suffit de cliquer deux fois sur celui-ci lors de la phase de développement. On introduit alors dans cette structure de base :

```
float number1 = float.Parse(textBoxNumber1.Text);
float number2 = float.Parse(textBoxNumber2.Text);
float result = number1 + number2;
textBoxResult.Text = result.ToString();
```

## Commande d'interfaces

L'électronicien peut se demander comment piloter les interfaces du PC le reliant au hardware. Le concept de sécurité associé à .NET (pour ordinateur de bureau) est prévu pour que les interfaces ne soient pas directement pilotées à partir du programme de l'utilisateur.

Le programme d'application C# .NET ou VB .NET se fiera davantage au pilote Windows prévu à cet effet. Dans la plupart des cas, il faut tout d'abord installer celui-ci. Le fabricant fournit souvent le pilote avec le matériel acheté (c'est le cas avec de nombreux appareils de mesure).

La programmation d'un pilote est proche du hardware. Cela veut dire que C++, par exemple sera plus approprié sous Windows. Faute de place, il nous faut renoncer à nous engager plus avant dans le sujet de la programmation d'un pilote. Nous donnerons toutefois un petit exemple d'appel d'un pilote pour interface parallèle. La simplicité de cette dernière lui a permis de survivre jusqu'à nos jours dans de nombreux circuits.

L'exemple est dû à Levent Saltuklaroglu [8] et repose sur le pilote bien connu inpout32.dll téléchargeable gratuitement en [9] :

```
using System;
using System.Runtime.InteropServices;
```

```
public class PortAccess
{
    [DllImport("inpout32.dll", EntryPoint="Out32")]
    public static extern void Output(int adress, int value);
}
```

La directive Using sert à inclure les bibliothèques des classes de base. L'accès au pilote est alors encapsulé dans une classe définie par l'utilisateur dénommée PortAccess. Une méthode « Output » y est définie. Elle comporte 2 paramètres (adresse du port parallèle, par exemple décimal 888 pour LPT1, ainsi que la valeur de sortie désirée). Les paramètres sont passés dans la classe au sous-programme Out32 faisant partie de inpout32.dll. Le sous-programme place finalement un octet sur le port parallèle.

Une fois la classe implémentée, l'application C# permet d'envoyer commodément des données sortantes, par exemple :

```
PortAccess.Output(888, 255);
```

Encore plus simple ? L'interface série. Les versions de .NET Framework à partir de 2.0 offrent en effet la classe System.IO.Ports.Serialport qui permet d'accéder à l'interface série sans même devoir lier un fichier .dll. Consultez la documentation en ligne [10] pour de plus amples informations.

Les valeurs des zones de texte (de type chaîne de caractère) sont tout d'abord lues et converties en un nombre. C'est le rôle de la méthode `float.Parse`. Le résultat de l'addition effectuée ensuite est attribué à la zone de texte dénommée `textBoxResult`. Plus précisément, la zone de texte affiche la valeur du résultat quand la valeur « `result.ToString` » est attribuée à la propriété « `Text` ». Comme on le voit, les propriétés des éléments de l'interface utilisateur peuvent être modifiées lors de la conception du programme (comme ci-dessus), mais aussi à partir du programme. C'est pourquoi le nom de l'élément de l'interface utilisateur doit être suivi d'un point, puis du nom de la propriété (`textBoxResult.Text`).

Il faut convertir préalablement le résultat (un nombre en format virgule flottante de type « `float` ») en une chaîne de caractères pour qu'il puisse être affiché dans la zone de texte. C'est pourquoi le nom de la variable « `result` » apparaissant dans le code doit être suivi d'un point et du nom de la méthode (`ToString`). Vous venez de faire appel à la programmation orientée objet en spécifiant les propriétés et en utilisant les méthodes de l'objet « `textBoxResult` » de la classe « `Textbox` ». Il pourrait difficilement en être autrement car l'orientation objet forme la base de C# (et VB.NET) : Les éléments de l'interface utilisateur ne sont donc pas les seuls à être des objets. Toute variable en est un !

## Étapes suivantes...

Une fois démarrée, l'application devrait être similaire à la figure 4. Rien n'interdit de continuer à programmer : une calculatrice maîtrisant les 4 opérations ne devrait pas poser un trop grand défi.

Pour qui a pris son pied avec C#, n'importe quelle librairie bien achalandée offre un vaste choix de littérature spécialisée destinée tant aux novices qu'aux utilisateurs avancés. Le nombre de livres sur C# dépasse largement celui des livres sur VB.NET (et sur VB, toujours plus en recul). Les solutions des problèmes .NET dans les revues techniques destinées aux concepteurs ou sur Internet sont presque toujours décrites en C#.

Comme nous l'avons déjà fait remarquer, les programmeurs .NET ne doivent pas tout programmer eux-mêmes. Ils disposent en effet d'une énorme bibliothèque de classes .NET (tous les objets sont répartis dans des « classes » qui déterminent leurs propriétés et leurs méthodes). La documentation (en ligne) de l'entreprise ayant créé le langage [6] peut servir d'ouvrage de référence. Visual Studio comporte une partie de l'assistance en version « offline » (attention : très exhaustif !). Tous les éléments d'une classe (méthodes, propriétés, événements) sont décrits (figure 5). On trouve souvent des exemples d'application classés selon le langage de programmation utilisé. Comme on dispose fréquemment de plusieurs possibilités d'appeler les méthodes d'une classe (override, autrement dit surcharge), il devient souvent indispensable de consulter la documentation. Par exemple, la méthode « `Show` » de la classe `MessageBox` (affichage d'une boîte de dialogue) dispose en tout d'un grand nombre (20 !) de méthodes d'appel. Celles-ci se distinguent par les paramètres transmis lors de l'appel.

La fonctionnalité « `IntelliSense` » de l'environnement de développement (figure 6) s'avère très utile lors de la saisie du code source. Il suffit par exemple de saisir le nom d'un objet d'une classe particulière pour que toutes ses propriétés et méthodes s'affichent de manière interactive.

La seconde partie de cette série d'articles sera consacrée à un bref aperçu de la programmation orientée objet. Elle sera consacrée à un sujet touchant de près au domaine de

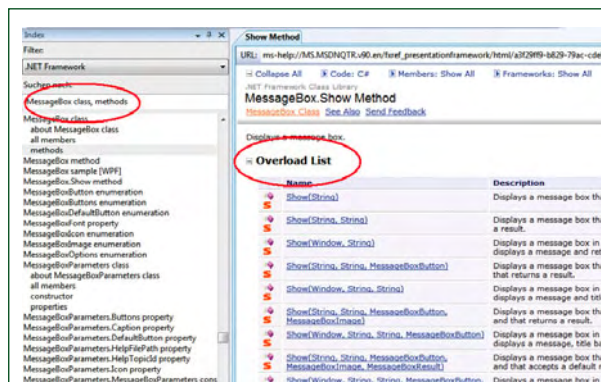


Figure 5.  
Le système d'aide constitue un véritable « ouvrage de référence » des classes de base.

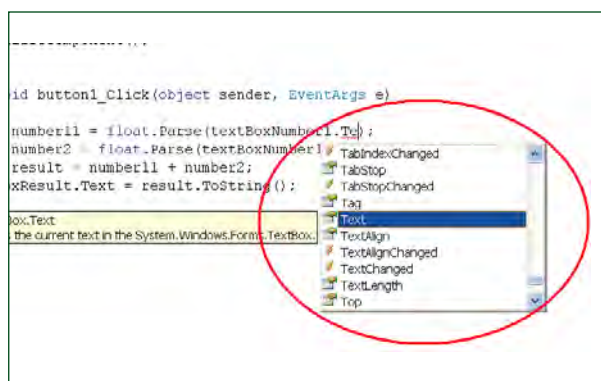


Figure 6.  
Soutien interactif du programmeur par « `IntelliSense` ».

la pratique : la programmation des graphiques vectoriels en 2D.

Le code source des exemples se trouve sur le site Web d'Elektor [7].

(080668-I, trad. Softcraft)

## Liens sur le Web

- [1] [www.mono-project.com/Main\\_Page](http://www.mono-project.com/Main_Page)
- [2] [dotnetframework.de/dotnet/produkte/sprachen.aspx](http://dotnetframework.de/dotnet/produkte/sprachen.aspx)
- [3] [www.elektor.fr/080450](http://www.elektor.fr/080450)
- [4] [msdn.microsoft.com/fr-fr/express/default.aspx](http://msdn.microsoft.com/fr-fr/express/default.aspx)
- [5] [www.icsharpcode.net/OpenSource/SD](http://www.icsharpcode.net/OpenSource/SD)
- [6] [msdn.microsoft.com/fr-fr/vcsharp/default.aspx](http://msdn.microsoft.com/fr-fr/vcsharp/default.aspx)
- [7] [www.elektor.fr/080668](http://www.elektor.fr/080668)
- [8] [www.codeproject.com/KB/cs/cspplers.aspx](http://www.codeproject.com/KB/cs/cspplers.aspx)
- [9] [logix4u.net/](http://logix4u.net/)
- [10] [msdn.microsoft.com/fr-fr/library/system.io.ports.serialport.aspx](http://msdn.microsoft.com/fr-fr/library/system.io.ports.serialport.aspx)

## À propos de l'auteur

Veikko Krypczyk a étudié l'informatique de gestion. Ce mor-  
du de l'informatique consacre depuis plusieurs années ses  
efforts au langage C# et à .NET Framework. Il s'intéresse  
particulièrement au logiciel d'application et à la conception  
d'applications graphiques en 2D (vous pourrez suivre son  
projet actuel sous <http://easy-grafiker.de>). Il se métamorpho-  
se régulièrement en bricoleur électronicien, s'arme d'un fer à  
souder et s'essaie à quelque projet intéressant. Vous pouvez  
le joindre par courriel sous [veikko2000@yahoo.de](mailto:veikko2000@yahoo.de).



# Hygromètre program

## Mesure du taux d'humidité relative et com

### Une application pratique du microcontrôleur PICBASIC PB-3B

La gamme PICBASIC du coréen Comfile Technology compte huit PIC programmables en langage BASIC. Le PICBASIC-3B utilise un PIC16C876 et quelques composants discrets. C'est le composant idéal pour débiter dans la programmation des microcontrôleurs sans assembleur ni langage C.

Faute de place, nous nous conten- tons ici de survoler la gamme dont les huit références se distinguent entre elles par le PIC utilisé, la capacité de mémoire de programme et de mémoire vive, le nombre d'entrées/sorties et la vitesse d'exécution des instructions en fonction du quartz utilisé (fig. 1). Grâce au langage BASIC, la mise en œuvre et la programmation de ces modu- les PICBASIC sont simples ; un PC avec le logiciel de développement gratuit **PIC- BASIC Studio** suffit. C'est ce logiciel qui transformera le programme BASIC en code hexadécimal, que l'on transférera ensuite dans la mémoire du module PIC- BASIC au moyen d'un cordon raccordé au port imprimante ou USB de l'ordi- nateur. Une fois le code chargé dans le module, celui-ci pourra être autonome.

#### Le µC PICBASIC PB-3B

Vendu environ 30 € il traite 56 000

codes hexadécimaux par seconde, ce qui compense la lenteur du langage BASIC. Son boîtier à 28 broches dis- pose de 21 ports d'entrées/sorties (fig. 2). La figure 3 met en évidence les par- ticularités suivantes :

- Active au niveau logique bas (0 V), la broche 1 remet le module à zéro.
- Les ports 0 à 3 (br. 2 à 5) peuvent être utilisés en entrées (I0 à I3), en sorties (O0 à O3), ou bien encore en entrées analogiques (AD0 à AD3). Dans ce dernier cas, la réso- lution est de 10 bits : 5 V sur une entrée seront convertis en la valeur binaire 11 111 1111 (=1023).
- L'entrée CLKIN ( br. 6) est une entrée de comptage (fréquence max. 20 kHz) associée à un compteur de 16 bits (comptage entre 0 et 65535).
- La fréquence de fonctionnement du PIC16C876 interne est donnée par un quartz (20 MHz) et deux condensa- teurs (22 pF).

- Relatives à un octet appelé « Bloc 1 », les huit entrées/sorties I/O8 à I/O15 sont disponibles sur les broches 11 à 18. Les broches 12 (PWM0) et 13 (PWM1) produisent des signaux à largeur d'impulsion modulée (PWM : Pulse Width Modulation) afin de commander la vitesse de moteurs à courant continu (par l'instruction BASIC PWM).
- La tension d'alimentation est appli- quée aux broches 19 et 20.
- Les broches 21 à 25 sont des entrées/ sorties ordinaires.
- La sortie PICBUS (br. 26) communi- que avec des périphériques, par ex. un afficheur LCD.
- Les broches 27 (PCIN) et 28 (PCOUT) servent à charger le code hexadé- cimal du programme BASIC compilé.

Les ports sont TTL : niveau logique HAUT au-dessus de 1,4 V et niveau logique BAS en dessous de 1,3 V, ou à trigger de Schmitt (ST) : niveau HAUT au-dessus de 3,4 V et niveau BAS en dessous de 3,3 V. Pour les ordinateurs sous Windows XP, il faut un câble adapté au port impr- imante du PC ou à la prise USB.

#### Logiciel PICBASIC Studio

Dans le cadre de cet article nous n'en- trerons pas dans le détail du logiciel de développement. PICBASIC Studio est téléchargeable gratuitement sur le site du constructeur ou celui du distribu- teur, de même que le manuel d'utilisa- tion. Ses fonctions principales :

- Un **éditeur** pour saisir les program- mes en langage BASIC.
- Un module de **téléchargement** pour le transfert (port parallèle ou USB du PC) du programme compilé vers la

	PB-1B	PB-1S	PB-2S	PB-2H	PB-3B	PB-3H	PBM-R1	PBM-R5
Gamme	PB	PB	PB	PB	PB	PB	PBM	PBM
Mémoire prog.	2 K	4 K	8 K	16 K	4 K	4 K	32 K	64 K
Mémoire RAM	96 octets	96 octets	96 octets	96 octets	79 octets	79 octets	8 K	32 K
Ports E/S	16	16	27	27	21	29	34	34
CPU	PIC16C73	PIC16C73	PIC16C74	PIC16C74	PIC16F876	PIC16F877	PIC16F877	PIC16F877
Fréq. quartz	4,19 MHz	4,19 MHz	4,19 MHz	20 MHz	20 MHz	20 MHz	20 MHz	20 MHz
Vitesse*	13,1 fois	13,9 fois	13,9 fois	3,1 fois	1 fois	1 fois	1,4 fois	1,4 fois
EEPROM pour données	—	—	—	—	—	—	8 K	32 K
Nb de broches	21	21	34	34	28	40	40	40
Conv. A/N (résolution)	5 (8 bit)	5 (8 bit)	8 (8 bit)	8 (8 bit)	5 (10 bit)	8 (10 bit)	8 (10 bit)	8 (10 bit)
A/N 12 bits	—	—	—	—	—	—	—	2 canaux
Sorties PWM (résolution)	2 (8 bit)	2 (8 bit)	2 (8 bit)	2 (8 bit)	2 (8 bit)	2 (8 bit)	2 (10 bit)	2 (10 bit)
Buffer RS232	—	—	—	—	—	—	Disponible	Disponible
Horloge RTC	—	—	—	—	—	—	—	Disponible

Figure 1. Vue d' ensemble de la famille PICBASIC. Cet article se contente d' une application du PB-3B.

# mable

## mande d'un humidificateur

Gérard Guihéneuf & René De Misy

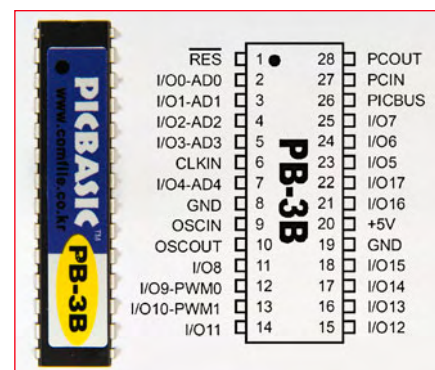


Figure 2. Description des broches du PICBASIC PB-3B.

- mémoire interne du PB-3B.
- Tant que le PB-3B est relié au PC, un module d'**émulation** peut exécuter le programme pas à pas et vérifier ou modifier des variables.
- **Download from Object ...** : Chargement du code hexadécimal d'un programme compilé et sauvegardé, et transfert vers le PB-3B.
- **Toggle Bookmark** : Place un index visuel dans la marge de la ligne.
- **Run** : Compilation du programme saisi en Basic et transfert du code hexadécimal vers la mémoire flash.
- **Break** : Affiche la fenêtre de l'option de mise au point du programme.
- **Syntax Check** : Vérifie la syntaxe du programme saisi en Basic.
- **View PICBASIC Flash Memory ...** : Affiche le code de la mémoire.

- **BYTEOUT** : Affecte une valeur binaire aux 8 sorties O8 à O15.
- **CAPTURE ()** : Mesure la durée d'une impulsion sur une entrée (I0 à I3).
- **COUNT ()** : Affecte à une variable de 16 bits (integer) le nombre d'impulsions reçues sur l'entrée de comptage rapide CLKIN (br. 6).
- **CLS** : Efface le contenu de l'écran de l'afficheur LCD.
- **DELAY** : Temporisation (en ms).
- **FREQOUT** : Produit un signal rectangulaire à une fréquence prédéfinie sur les sorties O9 (PWM0) ou O10 (PWM1).
- **IN ()** : Affecte à une variable byte le niveau logique d'une entrée.
- **KEYDELAY** : Délai lors de l'attente de l'action sur une touche.
- **KEYIN ()** : Donne l'état d'un poussoir sur une entrée.

- **LOCATE** : Coordonnées horizontales et verticales du curseur.
- **OUT** : Place une sortie à un niveau logique 0 ou à un niveau logique 1.
- **PADIN ()** : Gère un clavier à 16 touches (broches I/O8 à I/O15).
- **PEEK ()** : Lit les registres du PIC16F876.
- **POKE** : Ecrit dans les registres du PIC16F876.
- **PRINT** : Affichage de messages sur l'afficheur LCD.
- **PRINT DEC** : Envoie à l'afficheur la valeur décimale d'une variable.
- **PRINT HEX** : Envoie à l'afficheur la valeur hexadécimale d'une variable.

### Les instructions en Basic

Outre les instructions arithmétiques (+, -, \* et /), logiques (AND, OR, XOR et MOD) et de comparaison (=, >, <, >=, <=), les modules programmables PICBASIC reconnaissent 92 instructions Basic. Le P3-3B en connaît 52 réparties en « commandes » (par ex. PRINT, GOTO, RETURN) et en « fonctions » (par ex. ADIN(0), KEYIN(2), PADIN(1)).

Il est important de connaître la liste des instructions qui sont autant d'expressions réservées à ne pas attribuer à des variables du programme sous peine que le vérificateur de syntaxe renvoie un message d'erreur. Voici une **liste partielle de ces instructions** (liste intégrale dans la documentation téléchargeable) :

- **ADIN ()** : Lecture de la tension sur une entrée analogique (AD0 à AD3) et conversion en valeur numérique (10 bits) affectable à une variable du type *Integer* (16 bits)
- **ADKEYIN ()** : Gestion de poussoirs via les entrées A/N AD0 à AD3.
- **BCD ()** : Conversion d'une valeur ou d'une variable (8 ou 16 bits) en format Binaire Codé Décimal.
- **BYTEIN ()** : Lit simultanément l'état des 8 entrées I8 à I15 (bloc 1).

Broche n°	Description	Bloc	Niveau	Fonctions
1	/RES	Reset		
2	I/O0-AD0	Port 0	TTL	Conv. A/N
3	I/O1-AD1	Port 1	TTL	Conv. A/N
4	I/O2-AD2	Port 2	TTL	Conv. A/N
5	I/O3-AD3	Port 3	TTL	Conv. A/N
6	CLKIN	Entrée compteur		
7	I/O4-AD4	Port 4	TTL	
8	GND	Masse		
9	OSCIN	Connexion quartz		
10	OSCOUT	Connexion quartz		
11	I/O8	Port 8	Bloc 1	ST
12	I/O9-PWM0	Port 9	Bloc 1	ST
13	I/O10-PWM1	Port 10	Bloc 1	ST
14	I/O11	Port 11	Bloc 1	ST
15	I/O12	Port 12	Bloc 1	ST
16	I/O13	Port 13	Bloc 1	ST
17	I/O14	Port 14	Bloc 1	ST
18	I/O15	Port 15	Bloc 1	ST
19	GND	Masse		
20	+5V	Alimentation 5V		
21	I/O16	Port 16		ST
22	I/O17	Port 17		ST
23	I/O5	Port 5		ST
24	I/O6	Port 6		ST
25	I/O7	Port 7		ST
26	PICBUS	Port LCD		
27	PCIN	Connexion PC (IN)		
28	PCOUT	Connexion PC (OUT)		

Figure 3. Fonctions des broches du microcontrôleur PICBASIC PB-3B.

- **PULSE** : Produit des impulsions positives (sortie activée) ou des impulsions négatives (sortie désactivée) durant 2 à 3 microsecondes.
- **PWM** : Construit sur les sorties O9 ou O10 un signal rectangulaire (19,53 kHz) à rapport cyclique variable.
- **SERIN** : Réception série.
- **SEROUT** : Emission série.
- **SERVO** : Gestion d'un servomoteur.
- **SHIFTIN ()** : Communique avec des composants sériels (I2C ou SPI).
- **SHIFTOUT** : Communique avec des composants sériels 2 fils.
- **SOUND** : Produit un son.
- **STEPOUT** : Gère un moteur pas à pas.
- **TOGGLE** : Inverse le niveau d'une sortie.

Se reporter impérativement au manuel pour les informations complètes.

Utilisation du logiciel PICBASIC Studio

Sous Windows XP, le logiciel PICBASIC Studio communique avec le PB-3B via le port parallèle (imprimante) ou un port USB de l'ordinateur. Dans le premier cas, il faut ajouter une imprimante HP Laser Jet 4 non affectée par défaut et sans que cette imprimante soit matériellement raccordée à l'ordinateur, faute de quoi le PB-3B risque de ne pas être reconnu par le logiciel ; ce qui interdirait le téléchargement d'un programme du PC vers le PB-3B. De même, l'usage du port USB de l'ordinateur impose d'installer préalablement un pilote adapté au câble de programmation ; les fichiers de ce pilote sont téléchargeables sur le site du distributeur Lextronic (Drp10504.zip).

Figure 5. Exemple d'humidificateur domestique commandé par l'hygromètre programmable.

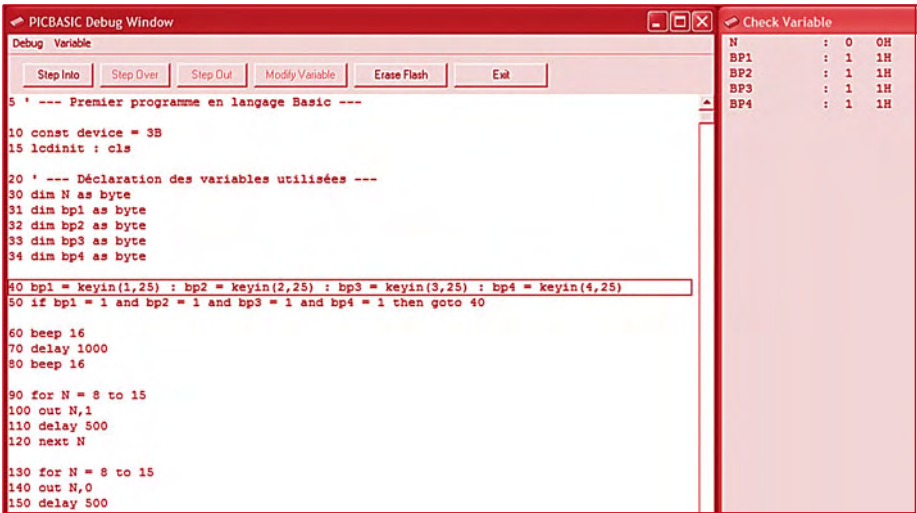


Figure 4. Les fenêtres de l'option de mise au point permettent de suivre pas à pas la bonne exécution du programme.

Ne connectez ou déconnectez le câble de programmation à l'ordinateur que lorsque le PICBASIC PB-3B est hors tension. La page principale de l'éditeur de programme s'apparente à un traitement de texte. Sauvegarder avec la commande *Save As* pour créer un fichier dont le suffixe est .bas. Vérifier la syntaxe avec *Syntax Check*. Compiler ensuite le programme à l'aide de la commande *Run* puis transférer le code hexadécimal vers le PB-3B raccordé au PC. Auparavant paramétrer le type de communication (parallèle ou USB) en exécutant *PC Interface Setup* (menu SETUP). Le code hexadécimal est observable grâce à *View Object Code* (menu RUN). Vérifier le fonctionnement de l'application du PB-3B.

En cas de problème, utiliser l'option de mise au point (émulateur) du logiciel avec la commande *Break* (arrêt), qui affiche deux fenêtres juxtaposées (fig. 4). À gauche le programme, la ligne en cours d'exécution lors de l'arrêt est encadrée. À droite les valeurs de variables utilisées. L'exécution du programme est figée. Continuer en mode pas à pas avec le bouton *Step Into* de la fenêtre de gauche. L'insertion de l'instruction *Break* à un ou plusieurs endroits du programme permet aussi de suivre l'évolution des variables et diagnostiquer un dysfonctionnement. Dès que le programme fonctionne, sauvegarder la version « objet » du programme (liste du code hexa-

décimal du programme compilé) avec la commande *Save Object* (menu FILE) qui crée un fichier avec suffixe .obj que l'on enverra directement au PB-3B avec la commande *Download Object*. **Quand l'application est au point, elle peut fonctionner déconnectée de l'ordinateur.** Le programme est initialisé à chaque mise sous tension du PICBASIC PB-3B.

Hygromètre programmable

Le moment est venu d'illustrer la mise en œuvre du PICBASIC PB-3B dans l'hygromètre programmable, un appareil qui commande un humidificateur en fonction du taux d'humidité ambiante. On distingue deux types d'hygrométrie: la mesure de l'humidité relative (rh) et celle de l'humidité absolue (H). L'humidité relative est le rapport entre pression partielle de vapeur d'eau contenue dans un gaz (Pv) et pression de vapeur d'eau max. possible, donc la pression de saturation (Ps), à une température donnée. L'humidité relative est une grandeur sans dimension. C'est un rapport indiqué en %. Respirer un air trop sec est non seulement désagréable, mais aussi malsain. Bien utilisé, un humidificateur domestique commandé par un hygromètre programmable apporte donc un confort appréciable. Voici les caractéristiques de notre hygromètre :

- Affichage du taux d'hygrométrie ambiante sur un afficheur LCD
- Sortie de puissance statique programmable (max. : 500 W)
- Programmation de deux seuils de fonctionnement
- Commande marche/arrêt manuelle
- Alimentation sous 230 V



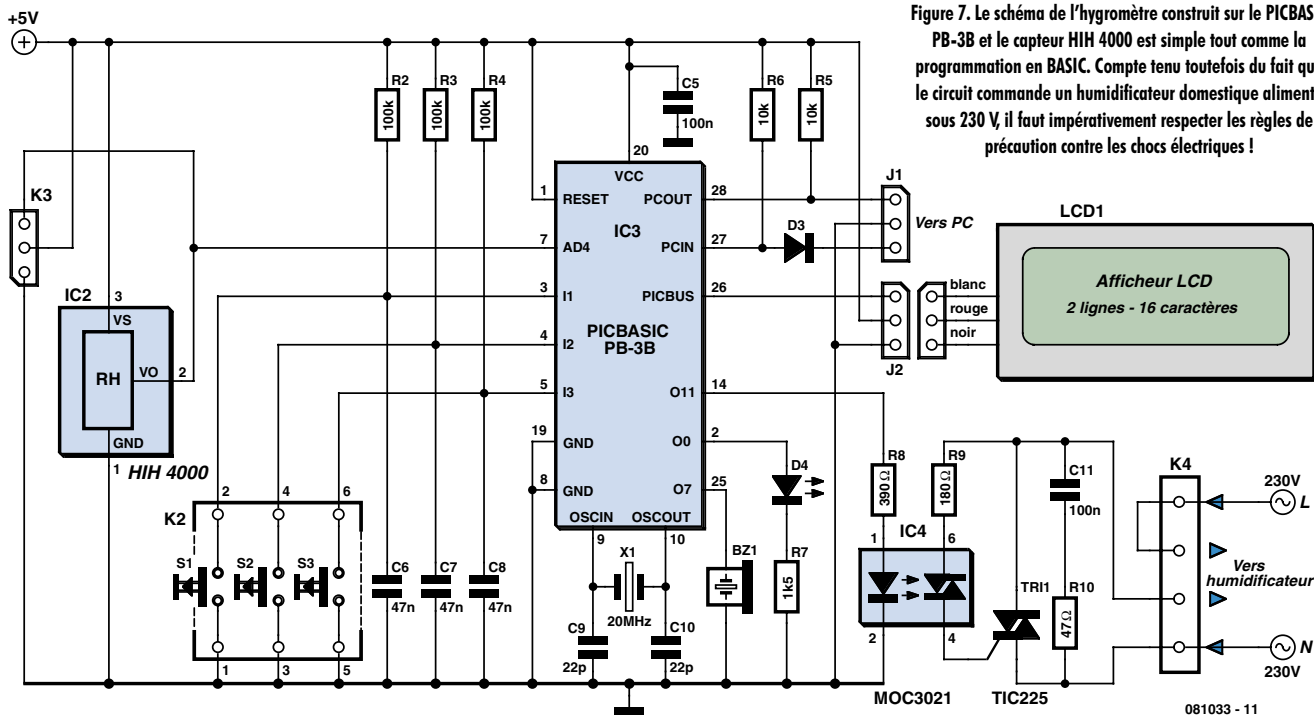


Figure 6. Brochage et fonction de transfert du capteur d'hygrométrie HIH 4000 (d'après Honeywell)

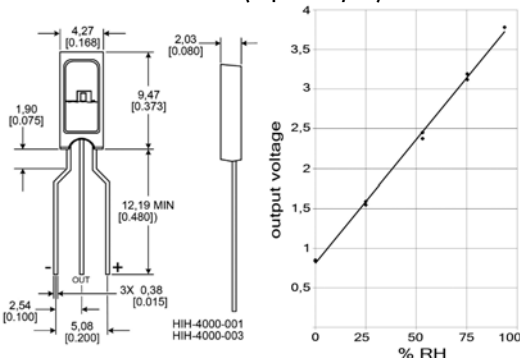


Figure 7. Le schéma de l'hygromètre construit sur le PICBASIC PB-3B et le capteur HIH 4000 est simple tout comme la programmation en BASIC. Compte tenu toutefois du fait que le circuit commande un humidificateur domestique alimenté sous 230 V, il faut impérativement respecter les règles de précaution contre les chocs électriques !

La mesure de l'humidité est assurée par le **capteur HIH 4000** de Honeywell, sous forme d'un boîtier en plastique à trois broches (fig. 6) qui fournit une tension entre 0 V et +5 V proportionnelle à l'humidité relative mesurée. Cette plage de tension est directement compatible avec les entrées analogiques du module PB-3B.

Contrairement à d'autres capteurs d'humidité résistifs ou capacitifs, le HIH 4000 mesure tous les taux entre 0% et 100%, selon l'expression mathématique suivante :

$V_{OUT} = V_S \times ((0,00062 \times RH) + 0,16)$   
avec RH : taux d'humidité relative en % (hygrométrie).

Il faut encore jongler un peu pour que le PICBASIC, qui ne connaît que des nombres entiers positifs, s'en accommode :

$$RH = \frac{100 \cdot V_{OUT} - 1600}{62}$$

avec  $V_{OUT}$  en centièmes de volt.  
Exemple : si l'entrée AD4 du PB-3B voit une tension de 2,5 V issue du capteur HIH 4000 alimenté sous  $V_S = 5$  V, l'exécution de la ligne 1230 du programme (se reporter au listing) attribuera à cette tension la valeur 250 (centièmes de volt) tandis que la ligne 1240 la convertira en un taux d'hygrométrie de :

$$RH = \frac{100 \cdot 250 - 1600}{62}$$

$$\frac{5000 - 1600}{62} = 54\%$$

## Analyse du schéma

Le capteur HIH 4000 (IC2) est raccordé à l'entrée analogique AD4 du PICBASIC (fig. 7). Un triple bornier à vis permet le raccordement d'un capteur déporté à l'extérieur du circuit imprimé. Les entrées I1, I2 et I3 du PB-3B recueillent les informations issues des trois poussoirs S1 (incrémentement), S2 (décrémentement) et S3 (validation/initiaisation). Des résistances de polarisation de 100 kΩ (R2, R3 et R4) et des condensateurs de découplage de 47 nF (C6, C7 et C8) atténuent les parasites. La LED D4 à faible consommation ( $I_F = 2$  mA) est directement raccordée à la broche O0 du PICBASIC et, en clignotant, témoigne du fonctionnement de l'hygromètre. Pour commander un humidificateur sous 230 V, la sortie O11 du PB-3B active un opto-triac (IC4) qui lui-même commande un triac de puissance (TRI1).

Réalisation pratique

L'ensemble des composants tient sur un circuit imprimé simple face de 130 x 100 mm (fig. 8). Le tracé est téléchargeable sur notre site. Si le capteur est monté en-dehors du circuit imprimé, on le raccorde à K3. L'afficheur LCD1 est monté sur le circuit imprimé à l'aide d'entretoises de 35 mm. Le buzzer Bz1 n'apparaît pas sur la figure 9 où il est caché par l'afficheur, comme D3 et C5. Ces composants sont sur le plan d'implantation sans afficheur dans la version téléchargeable sur notre site.

Si la puissance de l'humidificateur commandé demande un courant de l'ordre de 2 A, étamer les pistes du circuit imprimé qui aboutissent au bornier K4, à l'aide de soudure et d'un fer à souder.

Essai – Mise au point

Conditions initiales : le circuit intégré IC3 (PICBASIC PB-3B) n'est pas inséré dans son support et l'afficheur LCD n'est pas raccordé à (J2). L'humidificateur n'est pas raccordé au bornier K4 et celui-ci n'est pas alimenté sous 230 V.

Mettre sous tension (230 V) le circuit imprimé en alimentant les bornes 1 et 3 de K1 : la diode électroluminescente D2 s'allume. Vérifier la tension de 5 V entre les broches 20 (+) et 19 (-) du support de IC5.  
Hors tension : raccorder l'afficheur LCD au connecteur J2 avant de relier le câble de programmation à J1.  
Mettre le montage sous tension : un curseur fixe s'affiche alors sur l'afficheur LCD.  
Transférer le programme dans le microcontrôleur PICBASIC pour constater

Liste des composants

Résistances 1/4 W – 5% :

- R1, R7 : 1,5 kΩ
- R2, R3, R4 : 100 kΩ
- R5, R6 : 10 kΩ
- R8 : 390 Ω
- R9 : 180 Ω
- R10 : 47 Ω

Condensateurs chimiques (25 V) :

- C1 : 1000 µF
- C2 : 220 µF

Condensateurs polyester (63 V) :

- C3, C5 : 100 nF
- C4 : 10 nF
- C6, C7 et C8 : 47 nF

Condensateur polyester (400 V) :

- C11 : 100 nF

Condensateurs céramique :

- C9, C10 : 22 pF

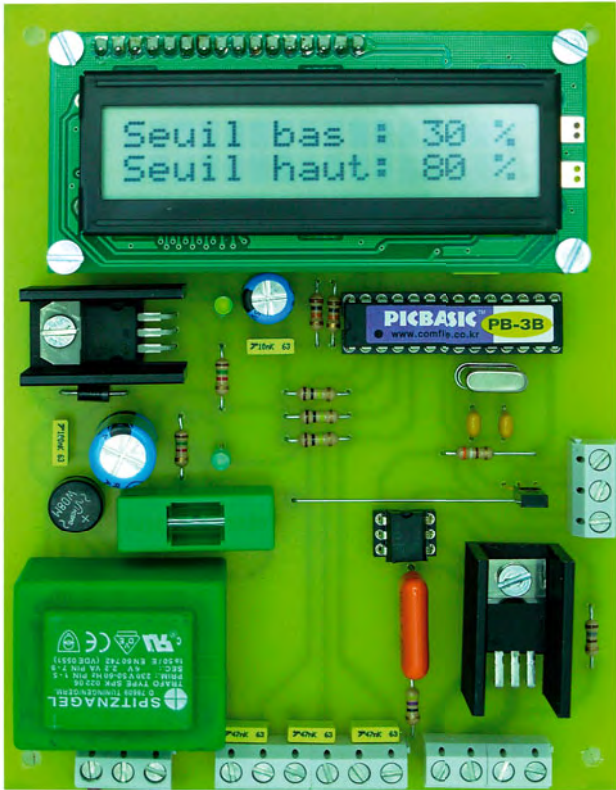
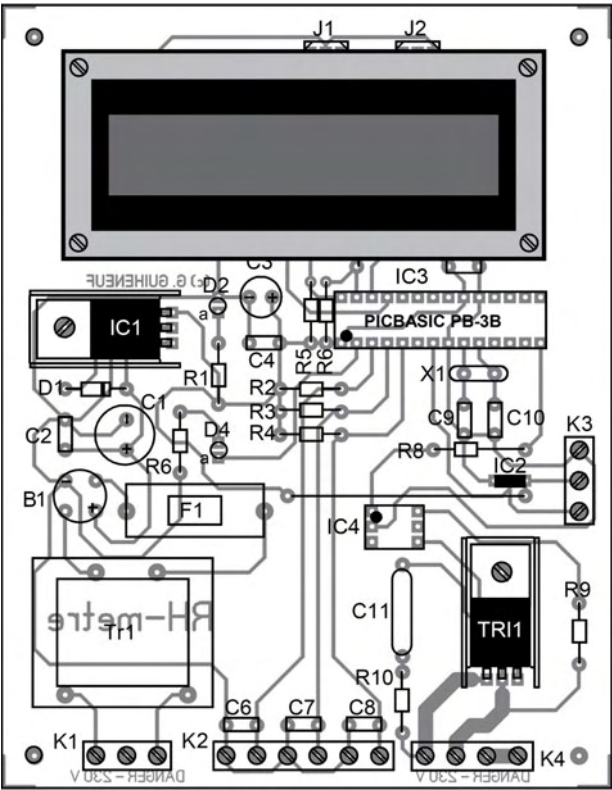
Semi-conducteurs :

- D1 : diode 1N 4004
- D2, D4 : diode électroluminescente 3 mm verte (faible consommation : I<sub>F</sub> = 2 mA)
- D3 : diode 1N 4148
- IC1 : régulateur de tension 7805 (TO220)
- IC2 : capteur d'hygrométrie Honeywell HIH 4000-001 (fournisseur : Farnell)
- IC4 : opto-triac MOC 3021
- B1 : pont redresseur 1 A
- TRI1 : triac 8 A – 600 V (TIC 225 M ou équiv.)

Divers :

- Bz1 : transducteur piézo 5 V (sans oscillateur)
- S1, S2, S3 : bouton poussoir NO (normalement ouvert) à contact fugitif
- Tr1 : transformateur 230 V / 6 V / 2,2 VA

- F1 : porte fusible + fusible 500 mA
- X1 : quartz 20 MHz
- K1, K3 : Borniers à souder 3 plots
- K4 : Bornier à souder : 4 plots (2 x 2 plots)
- K2 : Bornier à souder : 6 plots (2 x 3 plots)
- 1 support à souder DIL 28 « tulipe » (étroit)
- 1 support à souder DIL 6 « tulipe »
- 2 dissipateurs thermiques TO220
- Visserie et entretoises
- Afficheur : LCD 2 lignes de 16 caractères (réf. ELCD162)
- Composants distribués par LEXTRONIC :**
- Connecteurs à souder :
  - J1 : réf. CON-PB3
  - J2 : réf. CON-AFF
- IC3 : microcontrôleur PICBASIC PB-3B (COMFILE Technology)
- Câble de programmation USB ou parallèle



Figures 8 et 9. Implantation des composants et prototype : l'implantation du capteur HIH 4000 sur le circuit imprimé est optionnelle ; il est également possible de raccorder un capteur déporté (bornier K3).

## Lecture commentée de l'extrait du programme

La version intégrale est téléchargeable sur le site. La vocation de cet extrait (environ la moitié du programme) est de montrer la simplicité de la programmation en BASIC. La ligne 37 ordonne la mesure et l'affichage du taux d'hygrométrie ambiante (effectuée par les sous-programmes 1200-1250 et 1100-1120). À partir de la ligne 41 s'opère

la saisie du seuil bas de fonctionnement de la sortie statique. Ainsi, avec une valeur affichée par défaut de 50%, ce seuil bas peut être incrémenté par S1 (ligne 45). À l'inverse, il est possible de décrémenter la valeur du seuil bas avec S2 (ligne 46). Une fois la valeur du seuil bas ajustée, S3 ordonne la validation de la consigne saisie (ligne 47). De la même manière, la saisie du seuil haut, qui conditionne l'arrêt automatique de la sortie de l'hygromètre, s'effectue à partir de la ligne 51. Les lignes 55, 56 et 57 gèrent

l'incrément, la décrément et la validation de la consigne « seuil haut » ; le nouveau seuil est validé s'il est supérieur au seuil bas saisi auparavant (instruction SH > SB). Ensuite, la gestion du fonctionnement de l'hygromètre s'appuie sur un Grafset à deux étapes (omis ici faute de place). De façon alternée, l'hygromètre affiche le taux d'hygrométrie ambiante (sous-programmes 1100-1120 et 1200-1250) ou les seuils hauts et bas programmés (renvoi au sous-programme 1000-1030).

```
[...]  
36 ' --- Affichage du taux d'hygrométrie ambiante ---  
37 cls : locate 0,0 : print « Rh ambient : » : gosub  
1200 : gosub 1100 : delay 5000  
40 ' --- Configuration des seuils ---  
41 ' --- Saisie seuil bas ---  
42 SB = 50 : cls : locate 0,0 : print "Rh niveau bas :"  
" : locate 6,1 : print dec (SB);" % " : locate 4,1 :  
print "Rh :"  
43 S1 = keyin (1,10) : S2 = keyin (2,10) : S3 = keyin  
(3,10)  
44 if S1 = 1 and S2 = 1 and S3 = 1 then goto 43  
45 if S1 = 0 and S2 = 1 and S3 = 1 and SB < 99 then SB  
= SB + 1 : locate 6,1 : print dec (SB);" % " : locate  
4,1 : print «Rh :» : sound 7,159,2  
46 if S1 = 1 and S2 = 0 and S3 = 1 and SB > 0 then SB =  
SB - 1 : locate 6,1 : print dec (SB);" % " : locate  
4,1 : print «Rh :» : sound 7,159,2  
47 if S1 = 1 and S2 = 1 and S3 = 0 then locate 0,0 :  
print « VALIDATION ! » : sound 7,255, 16 : delay  
1000 : goto 50  
48 goto 43  
50 ' --- Saisie seuil haut ---  
51 SH = SB  
52 cls : locate 0,0 : print «Rh niveau haut :» : locate  
6,1 : print dec (SH);" % " : locate 4,1 : print «Rh :»  
53 S1 = keyin (1,10) : S2 = keyin (2,10) : S3 = keyin  
(3,10)  
54 if S1 = 1 and S2 = 1 and S3 = 1 then goto 53  
55 if S1 = 0 and S2 = 1 and S3 = 1 and SH < 100 then SH  
= SH + 1 : locate 6,1 : print dec (SH);" % " : locate  
4,1 : print «Rh :» : sound 7,159,2  
56 if S1 = 1 and S2 = 0 and S3 = 1 and SH > SB then SH  
= SH - 1 : locate 6,1 : print dec (SH);" % " : locate  
4,1 : print «Rh :» : sound 7,159,2  
57 if S1 = 1 and S2 = 1 and S3 = 0 and SH > SB then  
locate 0,0 : print « VALIDATION ! » : sound 7,255,  
16 : delay 1000 : goto 60  
58 goto 53  
[...]  
1000 ' --- SP / Rappel des seuils programmés (affichage)  
---  
1010 locate 9,0 : print dec (SB);" % " : locate 0,0 :  
print "Seuil bas : "  
1020 locate 9,1 : print dec (SH);" % " : locate 0,1 :  
print "Seuil haut : "  
1030 return  
1100 ' --- SP / Affichage hygrométrie «  
1110 locate 6,1 : print dec (RH);" % " : locate 0,1 :  
print « Rh :»  
1120 return  
1200 ' --- Mesure de la tension analogique ---  
1210 U1 = ADIN(4)  
1220 ' --- Conversion Tension / Hygrométrie  
1230 UH = ((50*U1)/102)  
1240 RH = (((100*UH)/5)-1600)/62  
1250 return
```

l'affichage pendant 5 s du message «\*\*\*Hygrometre \*\*\* / © ELEKTOR 2009 » : le programme a commencé son exécution.

Un premier affichage du taux d'hygrométrie ambiante intervient (5 s).

Appuyer sur les poussoirs S1, S2 et S3 pour configurer puis valider le seuil bas de fonctionnement de la sortie statique de l'hygromètre. La même procédure permet de saisir et valider la consigne du seuil haut qui ordonnera l'arrêt de l'humidificateur.

L'hygromètre affiche alternativement les deux consignes (seuils bas et haut) saisies et le taux d'hygrométrie ambiante. Si ce taux se situe entre les deux seuils programmés, la sortie statique de l'hygromètre est au repos. Il est alors possible de mettre en marche la sortie statique en appuyant sur S1, puis d'en ordonner l'arrêt avec S2. A défaut d'un arrêt ordonné par S2, le simple fait de souffler sur le capteur HIH 4000 entraînera également un arrêt de la sortie statique à condition d'avoir saisi un seuil haut d'hygrométrie de l'ordre de 70% !

Alors que la sortie statique de l'hygromètre est à l'arrêt, raccorder l'alimen-

tation 230 V sur les bornes K4-1 et K4-4 avant de connecter un humidificateur entre les bornes K4-2 et K4-3.

Appuyer sur S3 pour initialiser l'hygromètre afin de programmer de nouveaux seuils bas et haut proches de la valeur du taux d'hygrométrie ambiante. En fonction de ce taux et des deux seuils programmés, la sortie statique de l'hygromètre active ou non l'humidificateur associé pour obtenir une pseudo-régulation du taux d'hygrométrie : l'hygromètre programmable réalisé est opérationnel !

## Conclusion

Par sa simplicité, cette réalisation montre que la mise en œuvre des microcontrôleurs est à la portée de tous. Vous retrouverez le PICBASIC PB-3B dans le nouveau livre « 12 applications pratiques pour maîtriser le PICBASIC PB-3B » dont le titre est à lui seul un programme, qui couvre les domaines de la mesure, de la domotique, de la protection des biens, de l'automatisme ou bien encore de l'électronique de puissance : Automate programmable, Thermomètre, Station météo, Alimentation/

numérique, Capacimètre, Luxmètre, CO-mètre, Centrale d'alarme, Gradateur à deux voies, Centrale/domotique, Disjoncteur/programmable, Variateur de vitesse/à commande PWM.

(081033-1)

[www.comfile.co.kr/english2/](http://www.comfile.co.kr/english2/)

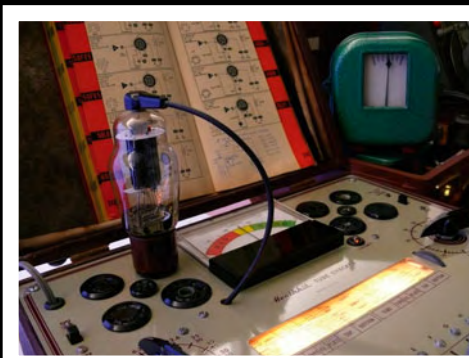
[www.lextronic.fr](http://www.lextronic.fr)







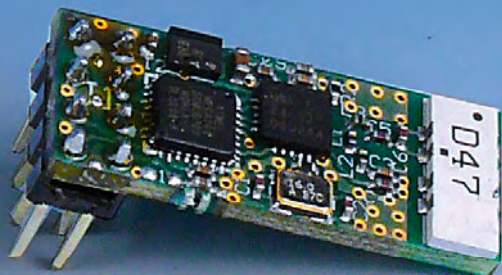
Elektor Live!, le salon de l'électronique organisé par Elektor et qui s'est déroulé à Eindhoven (Pays-Bas) le samedi 22 novembre 2008, a rencontré un grand succès ! Dans une ambiance décontractée avec un jeu, des offres spéciales et des démonstrations spectaculaires, plus de 1 000 visiteurs ont assisté à des ateliers et des conférences. Ils y ont également eu l'occasion de rencontrer des revendeurs et les employés d'Elektor. Elektor remercie tous les participants et attend déjà la prochaine édition avec impatience !





# Kit de développement radio 2,4 GHz Decibit

Jan Buiting



## Contenu du kit Decibit

- CD-ROM: logiciel de programmation, AVRasm2, WinAVR, sources des programmes d'exemple, pilote USB
- 1 programmeur USB
- 1 câble USB
- 2 modules DCBT-24AX
- 1 télécommande DCBT-24R6
- 2 testeurs portables
- 6 piles AA
- 1 fiche technique

Je n'aurai jamais envisagé l'utilisation d'un coupleur 3 piles AA ou d'autres composants volumineux jusqu'à ce que j'ouvre le kit radio v2.1 de Decibit. C'est pour sûr grâce aux minuscules modules transmetteurs (25 x 10 mm) équipés de CI plus petits encore (5 x 5 mm et 4 x 4 mm).

Decibit s'appuie sur des techniques éprouvées : un microcontrôleur ATmega168 d'Atmel en duo avec une puce *transceiver* nRF24L01 de Nordic Semiconductor forment la partie CI d'une série de modules émetteur/récepteur (E/R) intelligents opérants sur 2,4 GHz (bande ISM), programmables *in-situ* et disposant de la faculté remarquable de permettre l'utilisation de vos propres micrologiciels.

## Du Plug & Play à l'assembleur AVR

Actuellement (déc 2008) affiché à \$69, ce kit possède un fort potentiel pédagogique ; il y a beaucoup à faire et à découvrir pour des gens de tous niveaux. Les écoles, par exemple, peuvent suivre l'approche *plug & play* : installer le contenu du CD sur le PC ; brancher les modules sur le programmeur USB et lancer la programmation d'un fichier fourni. Puis, insérer les modules programmés dans les TP (testeurs portables), appuyer sur une touche d'un des TP et observer les LED sur le deuxième situé à l'autre bout de la classe.

Les plus expérimentés devraient jeter un œil aux exemples mettant en œuvre différentes méthodes de transfert de données et peut-être étendre ou modifier les bouts de codes fournis. Ces der-

niers couvrent les liaisons uni- ou bidirectionnelles utilisant les protocoles Shockburst, avec ou sans acquittement. Il y a même une démo type ping-pong permettant d'évaluer la portée des E/R en se baladant (j'ai pu atteindre jusqu'à 50 m en intérieur et extérieur). On peut voir (enfin, deviner) les actions d'envoi/renvoi de données sur les LED des TP.

## Outils de développement

Le programmeur USB du kit est un vrai joyau qui vous permettra de mettre votre propre code dans l'ATmega des modules. Les ingrédients : le logiciel de programmation Decibit ; AVRasm2 d'Atmel, le compilateur C/C++ WinAVR ainsi qu'un pilote USB. Si vous pouvez faire écrire « hello world » à l'ATmega, vous êtes en mesure de dialoguer avec l'E/R des modules Nordic de la même manière qu'avec un périphérique tel un afficheur LCD. Par exemple, un CALL

suffit pour un transfert de données. Les possibilités offertes par ce système vont bien au delà de celles que proposent habituellement les modules E/R « passifs » réagissant à des commandes style AT - les modules Decibit DCBT-24 peuvent être contrôlés depuis (ou devrait-on dire jusqu'à ?) l'assembleur AVR ce qui les rend très intéressants pour les développeurs créant des applications de télécommande radio « à la mode de l'embarqué ».

## Un large choix de modules

En plus du module E/R DCBT-24AX se trouvant dans le kit, Decibit fournit les modules suivants :

- **DCBT-24N** (E/R basse conso de très petite taille) ;
- **DCBT-24B** (E/R basse conso avec prise antenne externe) ;
- **DCBT-24C-PA** (80 mW E/R amplifié avec un connecteur antenne SMA) ;
- **DCBT-24R6** (télécommande porte-clés 6 touches).

Les modules sont disponibles en plusieurs versions matériel/logiciel que les clients peuvent préciser grâce au très complet système de numérotation des produits.

Selon Decibit, les certifications FCC et ETSI sont en cours pour les produits de la gamme DCBT-24. Des modules avec plus de broches ou basés sur les dernières technologies AVR comme XMEGA sont en route même si, pour le moment, l'ATmega168 donne entière satisfaction.

(08068-I, trad. Kévin PETIT)

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



# Traceur de courbes

## pour transistors bipolaires et à effet de champ, sur port USB

Rainer Schuster

Souvent imité, jamais égalé, le traceur de courbes caractéristiques reste bien le seul moyen de tester, mesurer et apparier des diodes et des transistors. L'ordinateur ayant envahi le labo, il est fait appel à une petite électronique à microcontrôleur R8C/13 branché sur un port USB, le tout géré par une application sous Windows.

### Caractéristiques :

- Pilotage par microcontrôleur R8C/13
- Connectable sur PC via USB
- Application Windows compatible Win2000/XP/Vista avec fonction d'aide exhaustive
- Exportation et importation via Excel
- Mode de fonctionnement prévu pour l'appariement des semiconducteurs
- Polarité NPN et PNP (N et P)
- Transistors bipolaires, MOSFET et JFET
- Impression de la caractéristique sur papier
- Exportation graphique de la caractéristique vers Word ou vers d'autres applications.

Ce n'est pas la première fois qu'Elektor décrit un traceur de courbes caractéristiques. Un premier article de ce genre est apparu en 1979, suivi par plusieurs autres en juin 1980, décembre 1988 et décembre 1989, mais ne faisant pas appel au PC pour l'affichage, et confinés à la mesure des transistors bipolaires NPN. Grande nouveauté en juin 1990 avec la possibilité de se connecter sur le port Centronics (imprimante parallèle) d'un ordinateur Atari, décidément bien en avance sur l'IBM-PC car il a fallu attendre janvier 1993 pour bénéficier d'une connexion sur un IBM-PC. Et c'est en avril 1998 qu'est intervenue l'adaptation pour Windows 95. Aujourd'hui s'opère le grand saut avec

Windows XP (ou Vista), la connectivité USB et la possibilité de traiter dans les quatre quadrants tous les semiconducteurs comportant jusqu'à trois bornes : NPN, PNP, NMOS, PMOS, NFET et PFET.

### Principe

Rien de neuf sous le soleil. Dans le cas d'un transistor bipolaire NPN, on injecte un courant de base précis tout en augmentant graduellement la tension collecteur – émetteur, tension que l'on représente en abscisse. C'est la valeur du courant de collecteur (qui dépend du gain notamment) que l'on mesure et que l'on représente en ordonnée. On peut réaliser différentes mesures avec différents courants de base. L'on obtient ainsi le réseau de courbes caractéristiques (**figure 1**).

### USB et $\mu C$

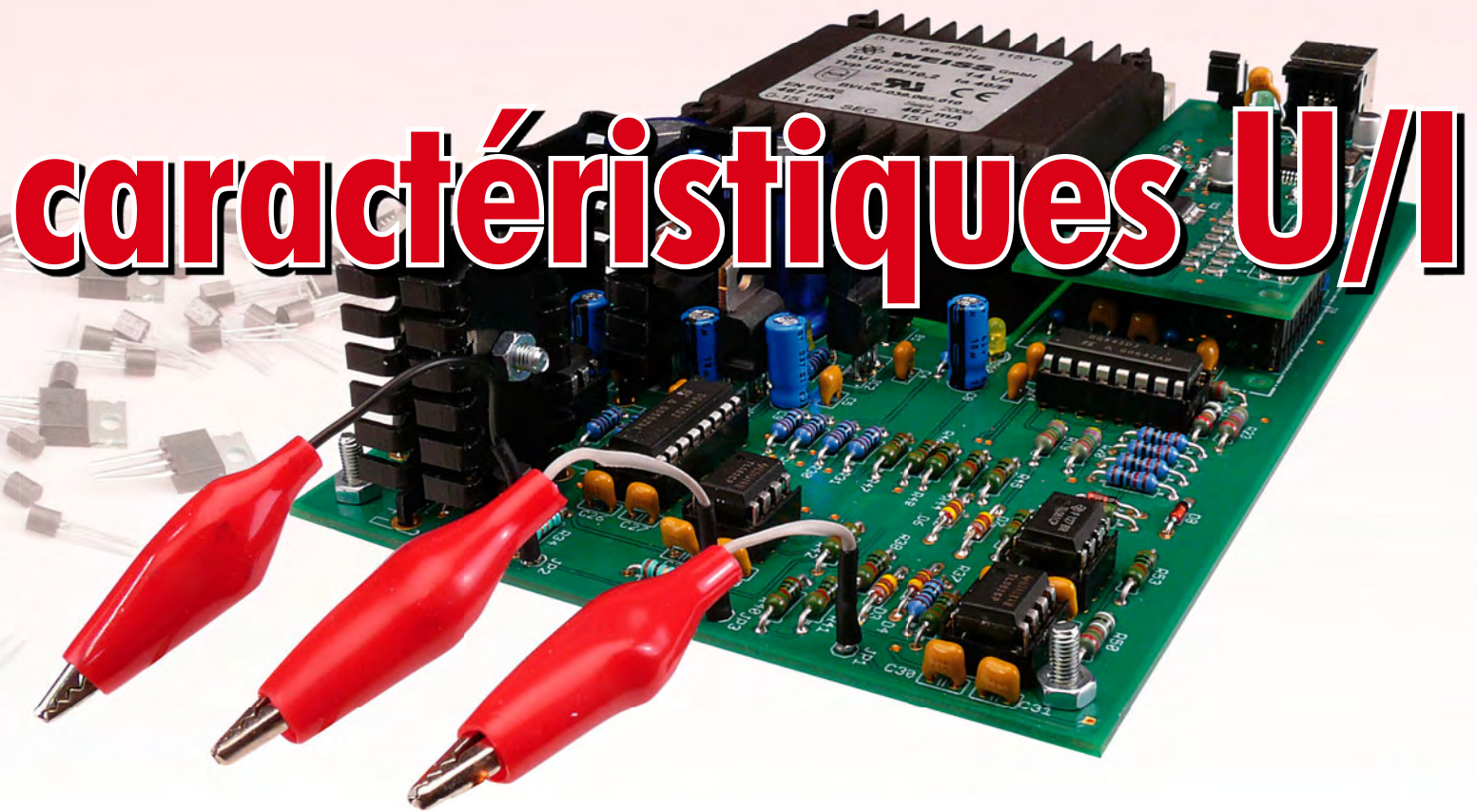
Nous nous référons à la **figure 2a**. Pas de surprises : c'est un petit microcontrôleur qui figure en bonne place sur le schéma. L'auteur a choisi un R8C/13 (Renesas, anciennement Hitachi) connecté sur une passerelle USB PL2303 (Prolific), couple qui a fait l'objet d'une description dans nos numéros de janvier 2006 [1] et mars 2006 [2]. Il en résulte une jolie platine universelle mesurant à-peine 80 mm sur 35 mm. La passerelle USB PL2303 est alimentée via le +5 V présent sur l'embase USB avec la tension VDD1 qui devient localement VDD une fois régulée et décou-

plée par le PL2303. C'est également le +5 V provenant de l'embase USB qui alimente le R8C/13. Cette tension de +5 V est mise à disposition de la platine analogique de mesure (**figure 2b**) via la broche 19 de l'embase K1, pour servir localement en tant que potentiel haut pour les résistances de rappel connectées sur les entrées de contrôle des commutateurs DG441. Revenons à la **figure 1a**. L'embase K1 fournit toutes les E/S dont l'application a besoin, y compris une tension de +5 V. Nous nous trouvons là devant un concept universel. Avis aux amateurs ! Toutes les E/S sont munies de résistances de limitation du courant, sachant qu'en pratique on n'est jamais totalement à l'abri d'une configuration erronée. Le sempiternel bouton Reset figure bien sur le schéma. Nous verrons qu'il joue un rôle déterminant dans la programmation de la mémoire programme du  $\mu C$ .

Une particularité du R8C/13 est qu'il peut être reprogrammé via son port série, ici connecté au port USB via le PL2303. Le cavalier JP1 installé sur la broche Mode du  $\mu C$  est prévu à cet effet, en conjonction avec l'environnement de développement de Renesas qui est disponible en téléchargement libre [3]. Pour pouvoir utiliser cette possibilité, on se doit de maîtriser l'écriture d'un programme, effectuer sa conversion en fichier Hex Motorola, et last but not least, veiller à installer tous les pilotes pour Windows, choses décrites en leur temps sur le site web de Elek-



# caractéristiques U/I



tor via les pages qui sont consacrées à la famille R8C de Renesas.

## Circuit de mesure

Nous nous référons à la **figure 2b**. Toute la partie analogique comporte sa propre alimentation régulée de 12 V et de 15 V symétriques à partir du secteur.

Les tensions de pilotage (courant de base et tension collecteur) sont produites via MLI à 2 kHz filtrée en passe-bas, éventuellement inversées pour les bipolaires PNP et les FETs type P. Les tensions recueillies (tension collecteur effective et courant de collecteur mesuré) rentrent dans deux entrées de conversion A/D du  $\mu C$ , précédés par des redresseurs à double alternance.

## Courant de base et tension de grille

Le signal TYOUT qui a une largeur d'impulsion comprise entre 0 et 100% est énergiquement filtré en passe-bas Butterworth du 3<sup>ème</sup> ordre (C12, C11, C13) via IC5, puis amplifié par IC5.B pour atteindre 10 V au maximum. Voir **figure 3**.

Le circuit IC8.A effectue l'inversion du signal pour obtenir une tension négative égale en valeur absolue.

Le commutateur quadruple IC7 alimenté en +15 V et -15 V joue un double rôle, en choisissant entre la tension positive de IC5.B ou négative de IC8.A, et en interposant un jeu de résistances suivant la gamme du courant de base qui est de 10  $\mu A$ , 100  $\mu A$  ou 1 mA.

Pour le test d'un JFET ou d'un MOSFET, peu importe la résistance en série

car seule importe la tension. Pour ne pas subir les effets de la capacité de grille, on a préféré utiliser la plus petite résistance série.

## Tension d'émetteur ou de drain

Le signal TZOUT qui a une largeur

d'impulsion comprise entre 0 et 100% est énergiquement filtré en passe-bas Butterworth du 3<sup>ème</sup> ordre (C17, C16, C18) via IC5, puis amplifié par IC6.B pour atteindre 10 V au maximum.

Le circuit IC8.B effectue l'inversion du signal pour obtenir une tension négative

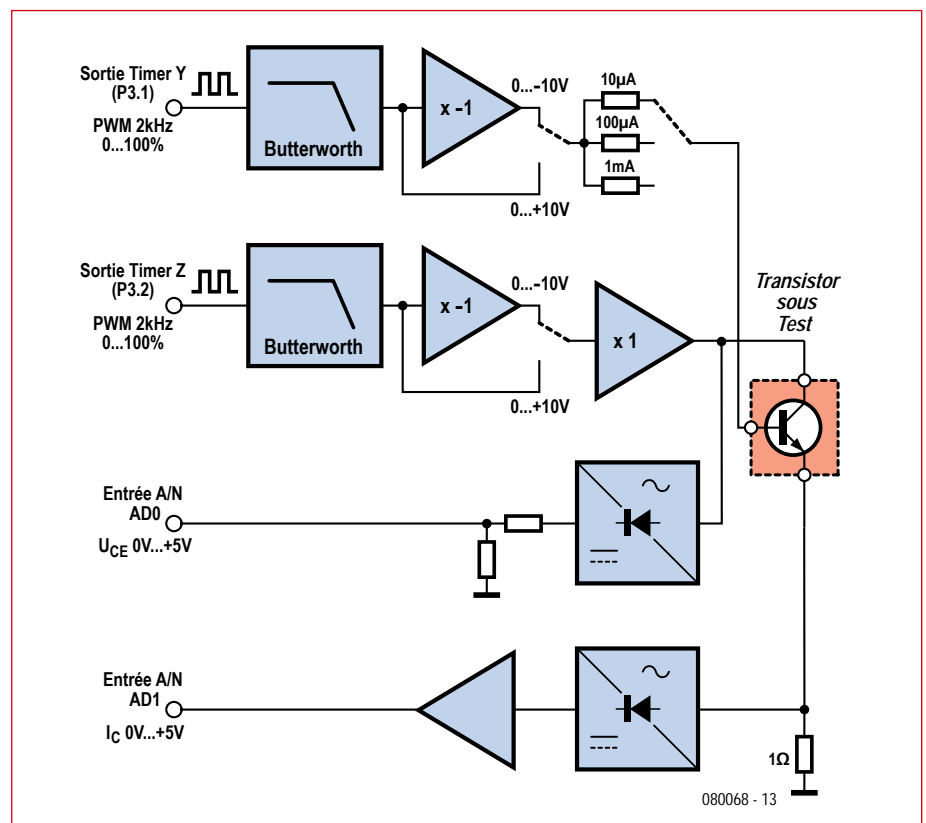


Figure 1. Schéma-bloc.

tive égale en valeur absolue. Le commutateur quadruple IC9 alimenté en +15 V et -15 V choisit entre la tension positive de IC6.B ou négative de IC8.B. On obtient de la sorte toute la gamme des tensions de collecteur ou de drain, positives et négatives, qu'il ne reste plus qu'à tamponner au moyen de IC10 monté en suiveur de tension avec sortie sur les émetteurs de T3 / T4 et limitation du courant de court-circuit à environ 500 mA grâce à R34 / T1 et R35 / T2.

**Mesure des paramètres**  
Commençons par remarquer qu'étant donnée la résistance mise en série avec la tension de base, le courant de base

Tableau 1 : Brochage K1.			
Pin	Signal	Pin	Signal
1	P1.7	11	P3.0
2	GND	12	P3.1
3	P1.3	13	P0.7
4	P1.6	14	P0.6
5	P1.1	15	P0.4
6	P1.2	16	P0.5
7	P4.5	17	P0.2
8	P1.0	18	P0.3
9	P3.2	19	+5V
10	P3.3	20	P0.1

peut être connu de façon mathématique, selon la loi d'Ohm. Il n'y a donc pas de circuit de mesure du courant de base. La grande précision qui est requise en cas d'appariement est ici fournie par l'interface graphique, plus parlante qu'une série de chiffres dans l'absolu (beta, gm, seuil).

En ce qui concerne la tension de collecteur ou de drain, la mesure intervient via IC11 monté en redresseur double alternance pour garantir une sortie positive quelque soit la polarité, puis via le diviseur de tension R42 / R43 et D5 en protection contre les surtensions, qui débouche sur le convertisseur A/D du  $\mu$ C. En ce qui concerne le courant de collecteur ou de drain, la mesure intervient via la résistance de shunt R49 de 1  $\Omega$  (1 mV par mA) suivie de IC12 monté en redresseur double alternance pour garantir une sortie positive quelque soit la polarité. Un amplificateur x10 fait suite, construit autour de IC13 qui aboutit sur le convertisseur A/D du  $\mu$ C avec R53 et D8 qui protègent contre les surtensions. La sensibilité vaut donc 10 mV par mA à ce niveau.

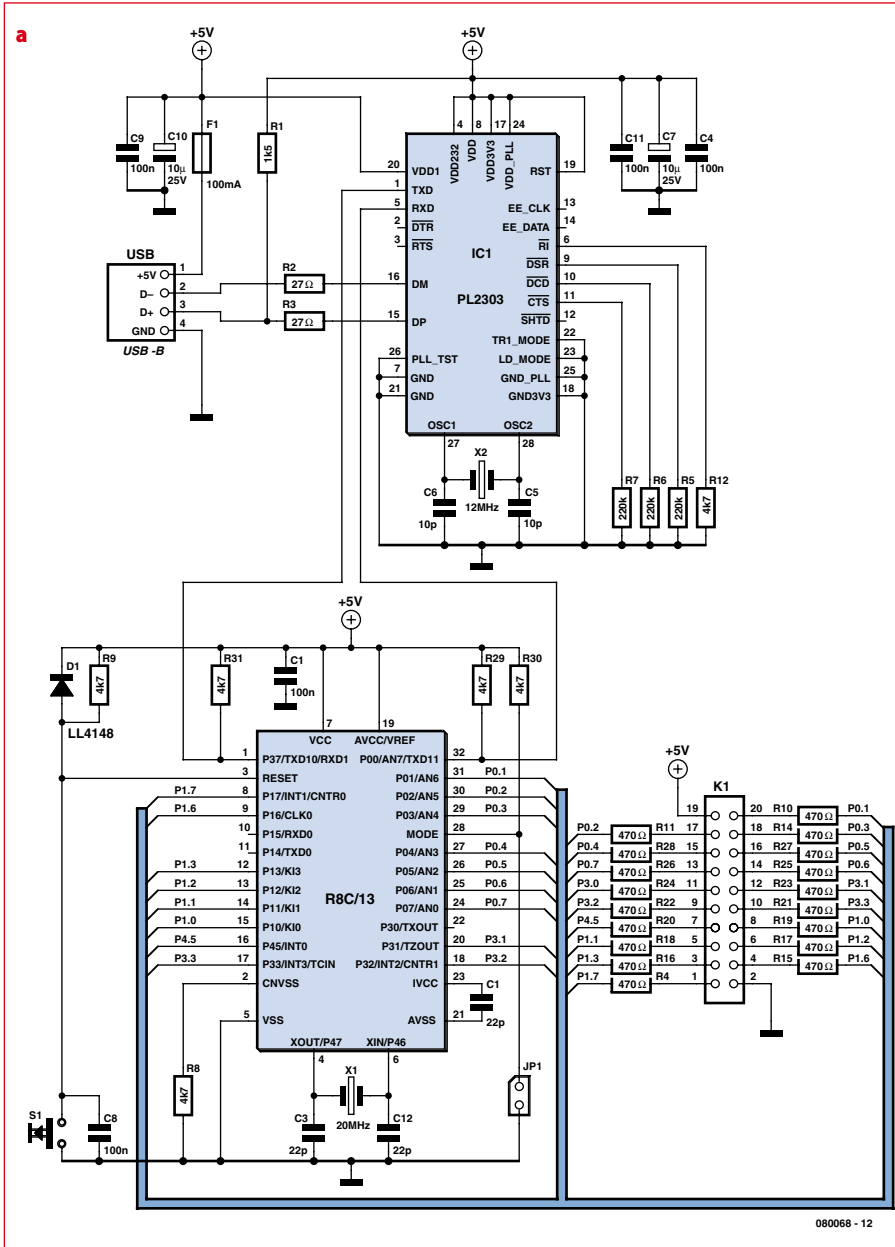


Figure 2. Schéma détaillé de la platine numérique (a) et de la platine analogique (b).

Logiciel R8C

Conformément à la tendance du moment, c'est un petit programme écrit en langage C qui gère le microcontrôleur R8C. Pour plus d'information à-propos de l'environnement de développement en langage C pour R8C, on voudra bien se reporter aux articles déjà parus dans Elektor [1] [2]. Tout est disponible en téléchargement libre [3].

Initialisations

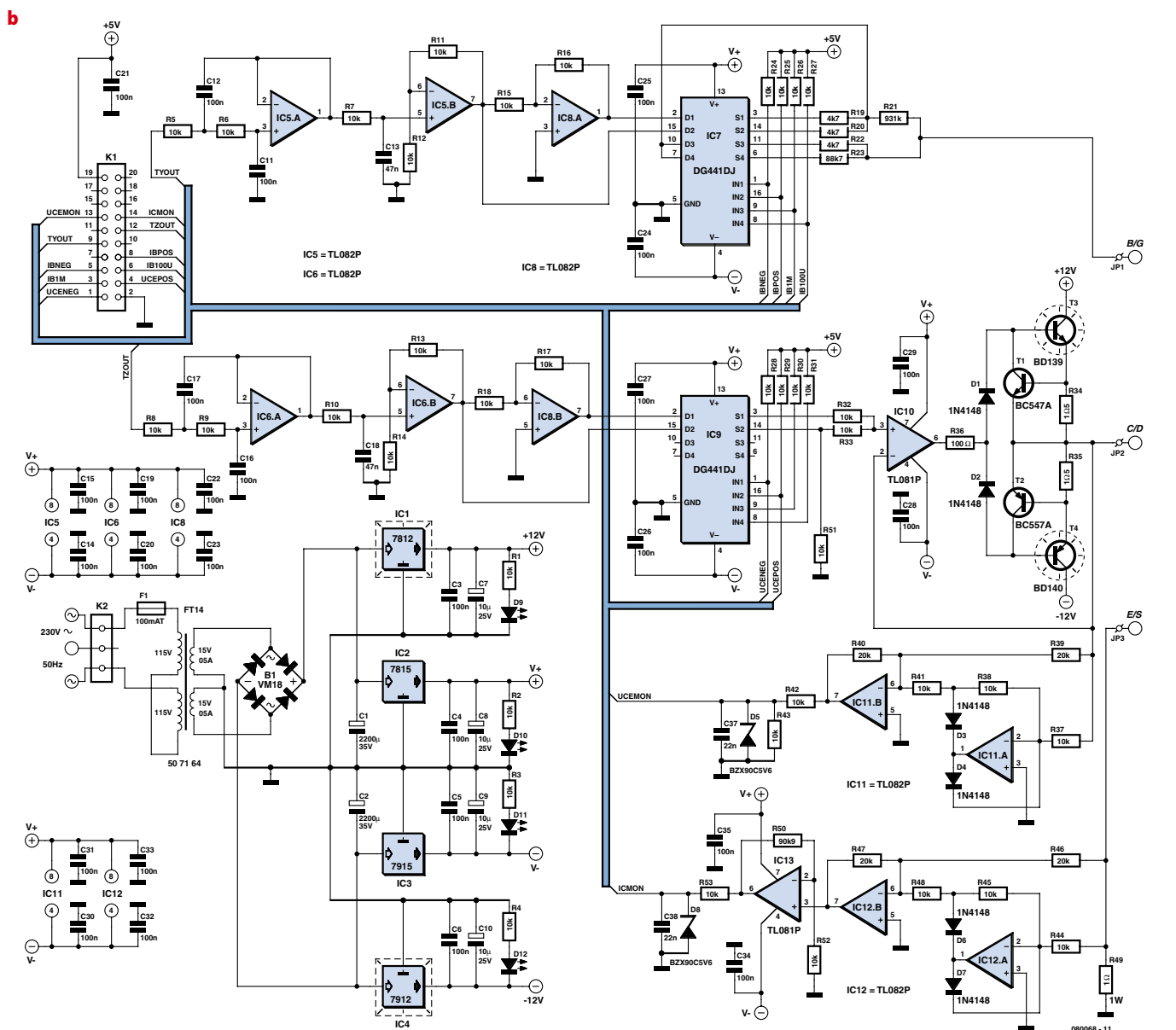
Les Timers Y et Z sont configurés pour la génération du MLI à 2 kHz et le Timer Y est configuré pour définir l'intervalle de temps qui sépare chaque prise de mesure. Vient ensuite le tour de l'UART confi-

guré à 9600 bauds, 8 bits de données, 1 bit d'arrêt, aucune parité. L'on valide l'interruption 6 couplée à l'UART. On termine par la définition du mode de mesure par défaut :

- courant maxi 100 mA
- type NPN
- gamme de courant de base : 10  $\mu$ A
- réseau à 5 courbes

Gamme de courant de base

L'on pilote les sorties P1.2 et P1.3 avec dans l'ordre respectif :  
1 et 1 10  $\mu$ A  
0 et 1 100  $\mu$ A  
1 et 0 1 mA  
Noter qu'il s'agit de sorties à collecteur



ouvert, d'où l'inversion.

### Polarités

L'on pilote P1.0, P1.1, P1.6 et P1.7 (voir le **tableau 2**) pour la polarité du courant de base et pour la polarité de la tension de collecteur.

### Boucle principale

On guette la réception de caractères sur l'UART, qui proviennent de la passerelle USB. Le PC émet des ordres. La platine répond à ces ordres.

Un ordre se compose d'une commande (lettre A...Z) suivie de paramètres optionnels qui prennent la forme

**Tableau 2 : Modes**

P1.0	P1.1	P1.6	P1.7	Base-Grille	Collecteur-Drain	Transistortyp
0	1	0	1	+	+	NPN/N-MOSFET
0	1	1	0	+	-	P-JFET
1	0	0	1	-	+	N-JFET
1	0	1	0	-	-	PNP/P-MOSFET

de caractères ASCII séparés par des virgules.

Un ordre se termine par la succession habituelle *Carriage Return* (0x13) et *Line Feed* (0x10) que nous symbolisons par CRLF dans le texte.

Le **tableau 3** donne un aperçu de partiel des commandes supportées.

### Séquencement d'un relevé de courbe

On commence par initialiser la tension de base à 0 V. Puis on commute la gamme 10  $\mu$ A pour le courant de base.

Avec des MOSFETs, on préfère diminuer la résistance série pour se rapprocher d'une attaque en tension, donc



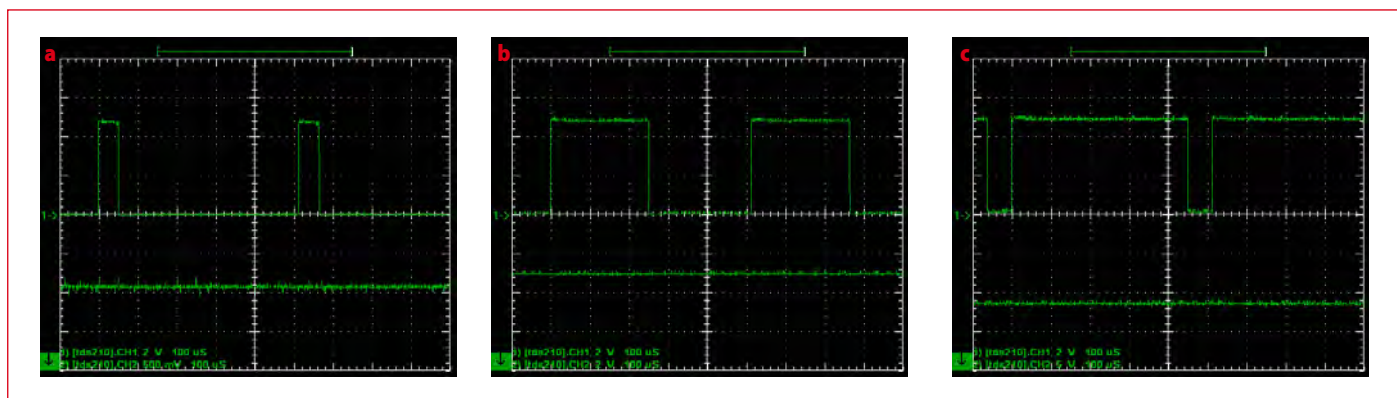


Figure 3. Signal MLI en sortie de ICSB avec une modulation à 10% (a), 50% (b) et 90% (c).

en réalité c'est la gamme 1 mA qui est sélectionnée.

Une distinction est opérée entre les JFETS et les MOSFETS au niveau du balayage en tension de grille. Le balayage en tension de grille commence à -10 V pour un N-JFET (+10 V pour un P-JFET). Alors que pour un MOSFET, il commence à 0 V.

En ce qui concerne la tension collecteur ou la tension de drain, on pourra aller de 0 à 10 V.

Une séquence d'auto-apprentissage commence alors, avec le courant de base ou la tension de grille qui aug-

mente jusqu'à ce que le courant de collecteur ou de drain maximum soit atteint. Dans le cas d'un transistor bipolaire, arrivé au bout de la première gamme de  $10 \mu\text{A}$ , on passe éventuellement à la gamme suivante ( $100 \mu\text{A}$ ) si le courant collecteur maxi n'est toujours pas atteint. Puis éventuellement arrivés sans encombre au bout de la gamme de  $100 \mu\text{A}$ , on passe à la gamme de 1 mA.

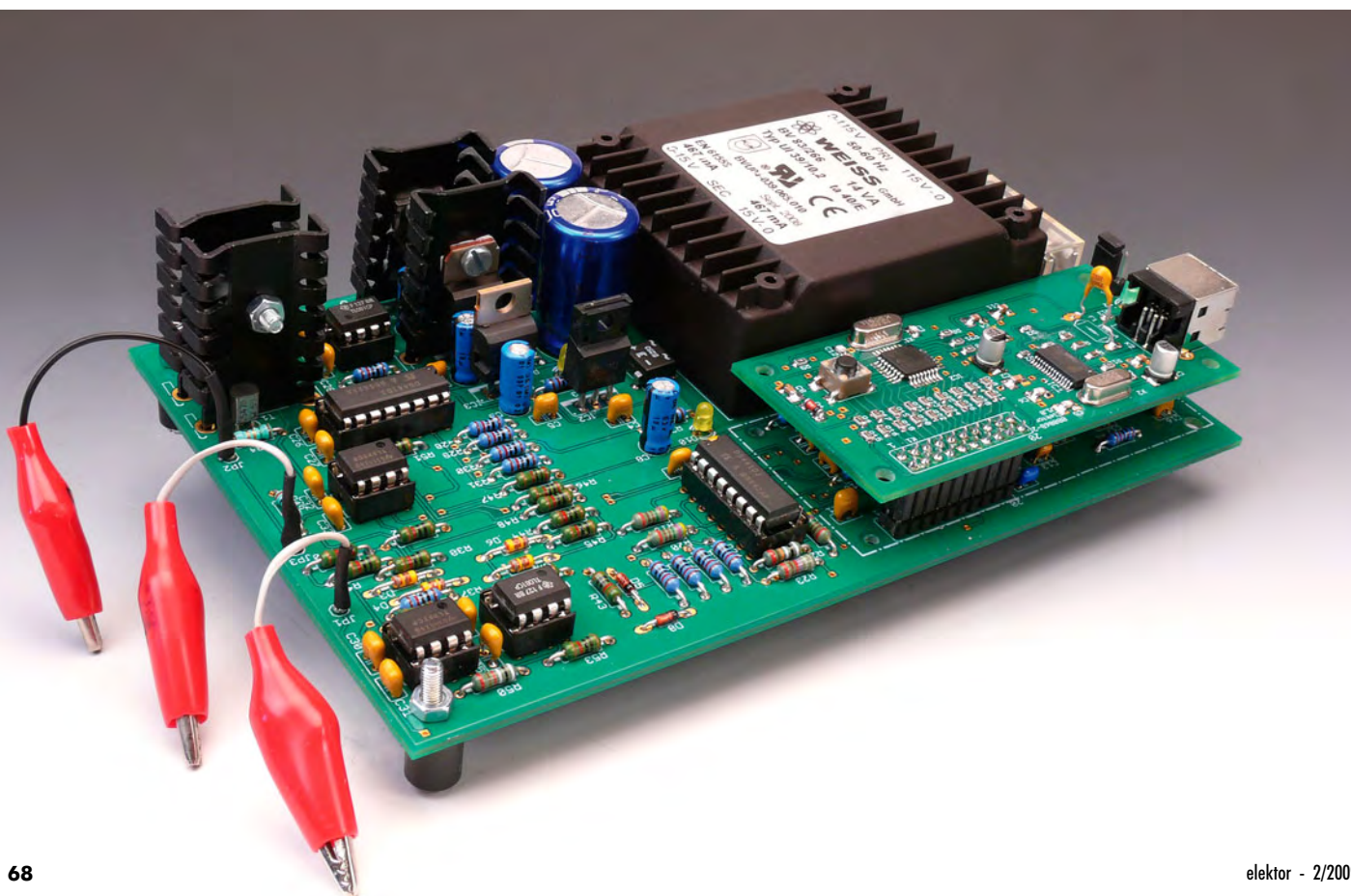
Si le courant de collecteur ou de drain n'a pas pu atteindre la valeur limite, le cycle de test se termine par la réponse « M1 » CR LF. Si en revanche le courant de collecteur ou de drain est anorma-

lement élevé, le cycle se termine par la réponse « M2 » CR LF, qui signale la présence d'un court-circuit.

La commande « K » permet de définir le nombre de courbes figurant sur le relevé de caractéristiques. Par conséquent, les valeurs du courant de base s'échelonnent alors régulièrement, avec pour intervalle  $\text{IBmax}$  divisé par le nombre de courbes. On arrive donc à un testeur intelligent qui s'adapte au composant avant de le tester en détail.

Si aucune erreur n'a été constatée durant cette phase d'apprentissage, le relevé des caractéristiques peut enfin commencer.

Chaque mesure se passe de la façon



**Tableau 3 : Aperçu partiel des commandes**

Commande	Paramètre	Libellé	Réponse attendue
A	–	Demande identification	Curve_Tracer Vx.y
B	0...255	Tension de commande sur circuit de base, max = 10 V	–
C	0...255	Tension de commande sur circuit de collecteur, max = 10 V	–
D	0 = 10 $\mu$ A 1 = 100 $\mu$ A 2 = 1 mA	Gamme courant de base	–
E	0 = pos 1 = neg	Polarité circuit de base	–
F	0 = pos 1 = neg	Polarité circuit de collecteur	–
G	–	Demande tension de collecteur	De 0 à 1024, pour 0 à 10,24 V
H	–	Demande courant de collecteur	De 0 à 1024, pour 0 à 512 mA
I	0...1024	Courant collecteur maxi de 0 à 1024 pour 0 à 512 mA	–
J	1 = NPN 2 = PNP 3 = N-MOSFET 4 = P-MOSFET 5 = N-JFET 6 = P-JFET	Mode	–
K	0...10	Nombre de courbes	–
L	–	Démarrage boucle de test	Voir texte

Note 1

Le paramètre x.y est le numéro de version.

A partir de cet instant les paramètres par défaut sont utilisés : ICmax 100 mA, mode NPN et 5 courbes.

suivante : alors que le courant de base reste fixe et déterminé par IBmax et « K », la tension de collecteur est graduellement augmentée, avec le circuit qui mesure la tension de collecteur effective, et qui mesure le courant collecteur.

Les résultats sont communiqués en continu au PC de la façon suivante :

« P » # Courbe, IB, Gamme CRLF

# Courbe : n° de la courbe

IB : courant de base (tension de grille)

Gamme : 0 = 10  $\mu$ A, 1 = 100  $\mu$ A, 2 = 1 mA

« N » UCE, IC CRLF

UCE : tension de collecteur (valeur 1024 = 10,24 V)

IC : courant de collecteur (valeur 1024 = 512 mA)

Le courant de base (transistors bipolaires) se calcule comme suit, en  $\mu$ A :

Gamme 10  $\mu$ A : IB x 4/100

Gamme 100  $\mu$ A : (IB - 14) / 2,3

Gamme 1 mA : ((IB - 16) / 2,6) x 10

Bien évidemment, pour les JFETs et les MOSFETs, c'est la tension de grille qui importe (valeur 1024 = 10,24 V).

Lorsque toutes les valeurs ont été transférées, le traceur de courbes envoie le

message « O » CRLF à destination du PC qui peut alors commencer la représentation des courbes.

### Flashage du $\mu$ C

Dans l'environnement de développement Renesas, il faut commencer par charger le fichier Hex Motorola intitulé *Curve\_Tracer.mot*. Connecter la platine. Insérer le cavalier sur JP1. Briellement appuyer sur le bouton Reset. Attendre la fin de la programmation. Enlever le cavalier de JP1. Briellement appuyer une nouvelle fois sur le bouton Reset.

## Application Windows

Elle a été écrite au moyen de Visual Basic 6.0. Il faut donc commencer par installer les DLL et autres composants Active-X qui ont trait à VB6. Cela se fait via l'installateur fourni compatible avec Windows 2000, XP et Vista, simplement repéré *Setup.exe*.

Il faut ensuite s'occuper du pilote de la passerelle USB Prolific qui est intitulé *PL-2303 Driver Installer.exe*.

On peut alors installer l'application proprement dite, intitulée *Curve\_Tracer.exe*.

### Utilisation et résultats

L'application sous Windows commence par tester la présence de la platine. En cas de succès, la fenêtre de travail apparaît conformément à la **figure 4**.

Tout en bas nous trouvons la définition du mode : NPN, PNP, NMOS, PMOS, NFET, PFET.

En haut à gauche nous trouvons le paramètre qui définit le nombre de courbes désirées par réseau.

En haut à droite nous trouvons le réglage du courant maximum qui permet de protéger les transistors les plus délicats.

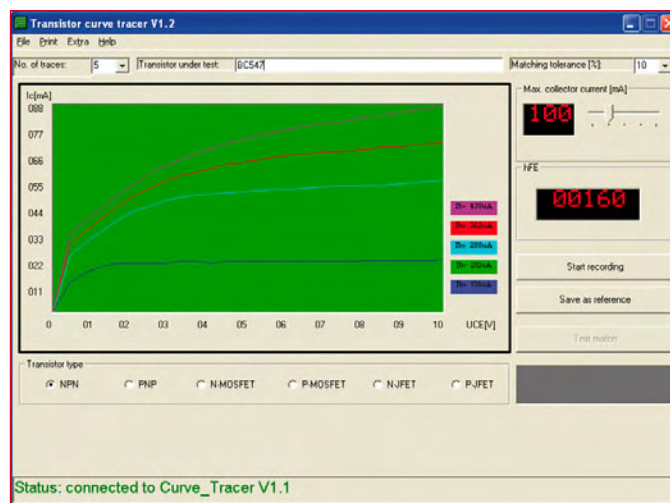


Figure 4. Voici un relevé de caractéristique.

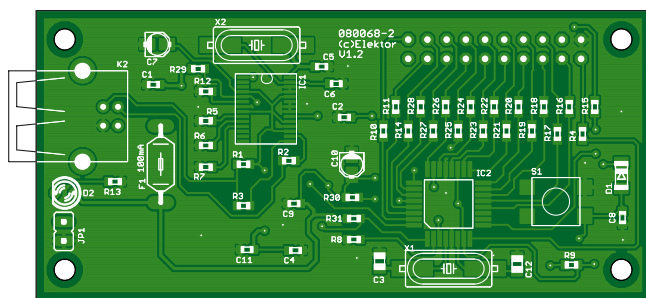


Figure 5. Platine numérique.

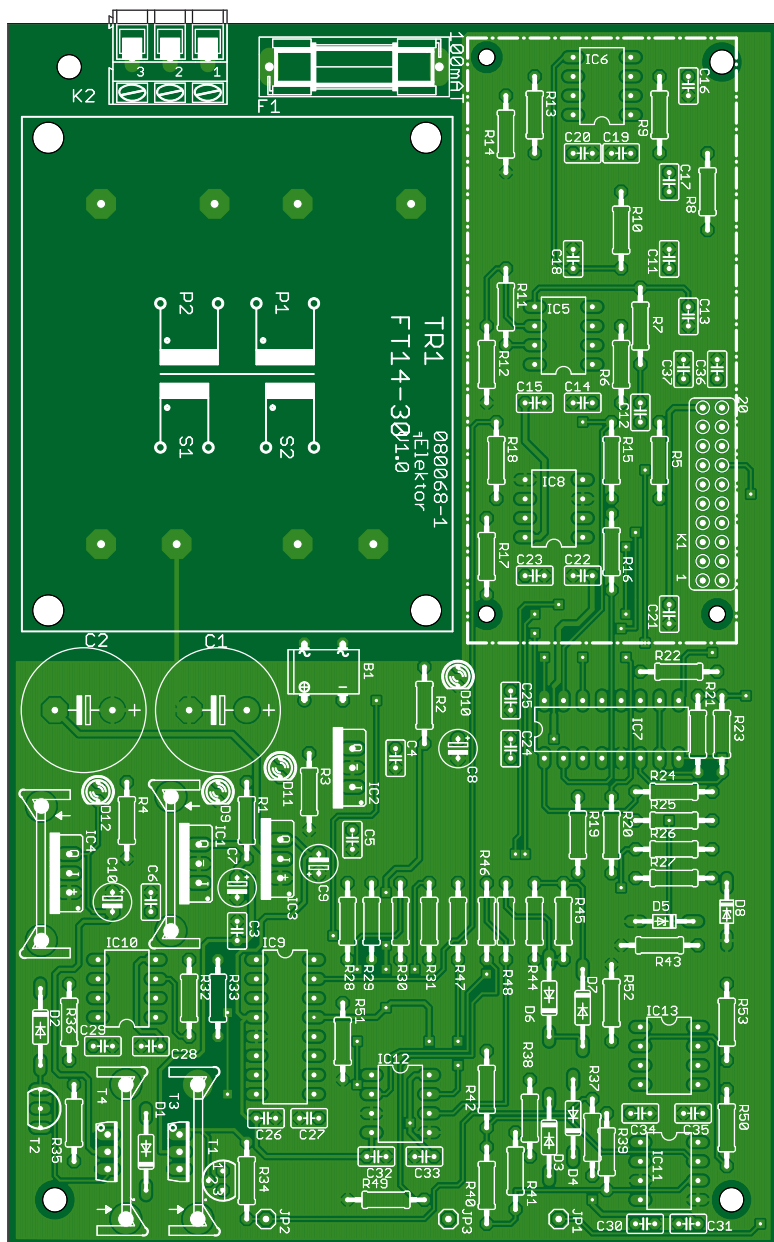


Figure 6. Platine analogique.

## Liste des composants

### Platine numérique (080068-2)

#### Résistances :

R1 = 1kΩ5 CMS 0603  
 R2, R3 = 27 Ω, CMS 0603  
 R4, R10, R11, R14 à R28 = 470 Ω, CMS 0603  
 R5 à R7 = 220 kΩ, CMS 0603  
 R8, R9, R12, R29 à R31 = 4kΩ7, CMS 0603  
 R13 = 1 kΩ, CMS 0603

#### Condensateurs :

C1, C2, C4, C8, C9, C11, C13 = 100 nF, CMS 0603  
 C3, C12 = 22 pF, CMS 0603  
 C5, C6 = 10 pF, CMS 0603  
 C7, C10 = 10 μF/25 V, SMD boîtier A

#### Semiconducteurs :

D1 = LL4148

Il ne reste plus qu'à déclencher la mesure à l'aide du grand bouton à droite, mesure qu'il est possible de mémoriser dans le cadre d'une procédure d'appariement avec la tolérance tout en haut à droite.

Via le menu il est possible d'accéder à d'autres fonctions telles l'impression via imprimante, la sauvegarde sous un format compatible avec Excel, ou la copie des courbes pour insertion graphique dans un document Word ou autre. L'aide en ligne, bien fournie, explicite encore d'autres détails.

## Assemblage

On trouve des CMS et rien que des CMS pour la platine numérique (figure 5) qui comporte le R8C/13 et le PL2303.

Par contre, la platine analogique ne comporte que des composants traversants. On veillera à munir les régulateurs +12 V et -12 V de refroidisseurs, de même pour les transistors T3 et T4 de l'étage de sortie. Il faut éviter que les refroidisseurs ne se touchent : gare aux courts-circuits ! Un isolant à ce niveau n'est peut-être pas superflu.

Avec l'embase K1 qui établit la liaison entre les deux platines, le montage reste relativement compact, capable de rentrer dans un boîtier de marque Bopla ref. E450 FVL.

Pour faire joli, on peut fixer sur le boîtier différents supports pour transistors. Pour faire simple et efficace, on peut se contenter de laisser pendre les trois fils de mesure munis d'une pince crocodile. Pour une dépense somme toute modique, l'instrument permet de se rapprocher de l'idéal lorsqu'il s'agit d'amplificateurs différentiels, d'étages de sortie



D2 = LED 3 mm, faible courant  
IC1 = PL2303X  
IC2 = R5F21134FP (R8C/13)

#### Divers :

K2 = embase USB type B  
K1 = embase 2x10 DIL (male)  
S1 = bouton SMD 6 mm  
X1 = quartz 20 MHz (CMS)  
X2 = quartz 12 MHz (CMS)  
JP1 = barrette 2 points avec cavalier  
F1 = Polyfuse 100 mA

#### Platine analogique (080068-1)

##### Résistances :

R1 à R18, R24 à R33, R37, R38, R41 à  
R45, R48, R51 à R53 = 10 k $\Omega$   
R19, R20, R22 = 4k $\Omega$   
R21 = 931 k $\Omega$   
R23 = 88k $\Omega$   
R34, R35 = 1 $\Omega$

R36 = 100  $\Omega$   
R39, R40, R46, R47 = 20 k $\Omega$   
R49 = 1  $\Omega$ /1 W  
R50 = 90k $\Omega$

##### Condensateurs :

C1, C2 = 2200  $\mu$ F/35 V radial  
C3 à C6, C11, C12, C14 à C17, C19 à  
C35 = 100 nF au pas de 2,5 mm  
C7 à C10 = 10  $\mu$ F/25 V radial  
C13, C18 = 47 nF au pas de 2,5 mm  
C36, C37 = 22 nF au pas de 2,5 mm

##### Semiconducteurs :

D1 à D4, D6, D7 = 1N4148  
D5, D8 = Zéner 5,6 V/500 mW  
D9 à D12 = LED 3 mm, faible courant  
B1 = redresseurs en pont VM18  
(1 A/100 V) format DIP4  
T1, T2 = BC547A  
T3 = BD139  
T4 = BD140  
IC1 = 7812

IC2 = 7815  
IC3 = 7915  
IC4 = 7912  
IC5, IC6, IC8, IC11, IC12 = TL082P  
IC7, IC9 = DG441DJ  
IC10, IC13 = TL081P

#### Divers :

K1 = embase 2x10 DIL (femelle)  
K2 = bornier à vis, 3 points, au pas de  
2,5 mm  
F1 = fusible en verre 100 mA, avec  
protège-fusible  
TR1 = Transfo secteur, 2x 15 V/14 VA  
(Weiss ref. 83/266)  
4x refroidisseurs FK218/SA32 (21 °K / W)

Logiciel PC et firmware  $\mu$ C : EPS  
080068-11, voir [3].

Les platines nues 080068-1 et -2 sont dis-  
ponibles sur l'e-choppe Elektor.  
La platine numérique est disponible déjà  
montée sous la référence 080068-91.

complémentaires, de miroirs de courant  
qui tous exigent des composants aux  
tolérances serrées. Sans parler du fait  
que d'un coup d'œil, ainsi équipé, on  
arrive à déterminer si un semiconduc-  
teur douteux est bien dans les spécifica-  
tions. Des applications détournées sont

envisageables comme l'appariement de  
diodes HF silicium et germanium et l'ap-  
pariement de diodes de puissance. Sans  
parler de la mise en évidence des « mous-  
taches » dans les anciens transistors,  
casse-tête des rétro-électroniciens.

(080068-I, trad. Stéphane CNOCKAERT)

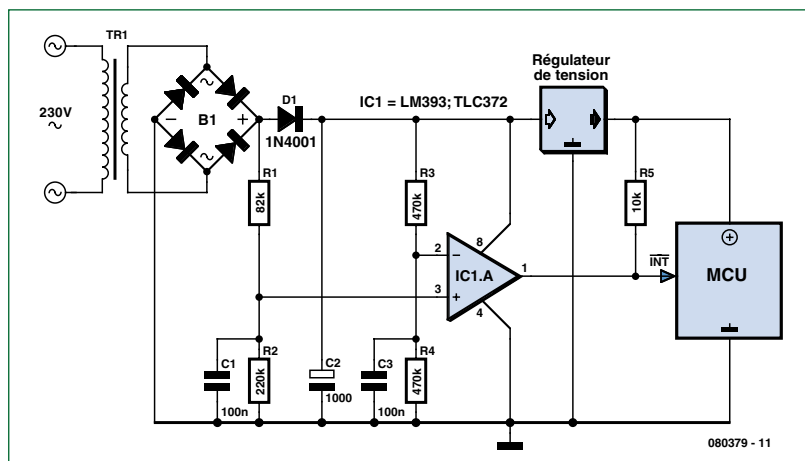
#### Weblinks

- [1] [www.elektor.fr/050179-2](http://www.elektor.fr/050179-2)
- [2] [www.elektor.fr/050179-3](http://www.elektor.fr/050179-3)
- [3] [www.elektor.fr/080068](http://www.elektor.fr/080068)

## Sauve qui peut!

### Rainer Reusch

Pour conserver les données  
de fonctionnement et les pa-  
ramètres d'un appareil, on a  
souvent recours à l'EEPROM  
d'un microcontrôleur. Ces in-  
formations sont ainsi à nou-  
veau disponibles à la pro-  
chaine remise sous tension.  
Comme le nombre de cycles  
d'écriture d'une EEPROM est  
limité, il n'est pas recomman-  
dé d'enregistrer immédiate-  
ment chaque modification.  
On les sauvegarde ensemble  
et rapidement juste avant de  
mettre hors tension. L'automati-  
sation de la sauvegarde deman-  
de la détection de la manoeuvre  
de l'interrupteur secteur ou d'une panne  
de tension. La solution présentée  
ici a été mise au point pour ça.  
Le montage se compose pour l'es-  
sentiel des composants classiques  
d'une alimentation secteur régulée  
linéaire (pont redresseur, conden-  
sateur de lissage et régulateur de  
tension). On a juste implanté une  
diode supplémentaire (D1) sur la



ligne d'alimentation. En fonction-  
nement normal, nous avons une  
tension redressée et lissée aux  
sorties des diviseurs de tension –  
et donc aux entrées du compa-  
rateur IC1. Le premier diviseur de  
tension (R1, R2, C1) est à calculer  
de telle façon que sa tension de  
sortie (0,5 à 1 V) soit un peu  
plus élevée que la tension de sortie  
du second diviseur de tension. La  
sortie du comparateur est alors

au niveau haut.  
Si la tension du secteur est cou-  
pée, la tension de sortie du pre-  
mier diviseur de tension chute très  
rapidement puisque la constante  
de temps choisie est vraiment très  
petite. Le condensateur de lissage  
(C2) fait que la tension en amont  
de la diode D1 persiste encore  
un certain temps. Le compa-  
rateur délivre alors un niveau bas  
qui déclenche une interruption du

microcontrôleur. Cette inter-  
ruption est pour lui le signal  
de lancement de la sauve-  
garde des données. Le dé-  
lai dont il dispose dépend  
de l'énergie fournie par le  
condensateur de lissage  
C2. La capacité de C2 doit  
donc être assez grande ! Le  
cas échéant, le microcontrô-  
leur commencera par cou-  
per quelques récepteurs  
connectés (LED, relais, etc.)  
de façon à gagner un peu  
de temps.

Le montage présenté est,  
en principe, conçu pour  
des appareils fonctionnant sur pi-  
les ou accumulateurs (le transfor-  
mateur et le pont redresseur n'ont  
donc plus lieu d'être là). Dans ce  
cas, vous pouvez laisser tomber le  
condensateur C1. La présence du  
condensateur C2 n'est pas tout à  
fait impérative. Il empêche toute-  
fois une impulsion au niveau bas  
à la mise sous tension.

(080379-I)

# Détecteur d'ultrason

## À la recherche des fréquences élevées

Ton Giesberts

On peut, avec le montage décrit ici vérifier rapidement et facilement s'il y a production. où que ce soit, de sons nocifs à fréquence élevée, en particulier dans les endroits où les boîtiers anti-jeunes à ultrason ont été installés. Dix LED visualisent la pression sonore des signaux entre 16 et 40 kHz.

On utilise souvent, pour chasser les « groupes de jeunes traînant à certains endroits », un système à « ultrason » (tel que le Mosquito). Ce genre de système, installé près de la zone sensible, produit des sons de fréquence comprise entre 17 et 20 kHz. Pour les jeunes (oreilles) ces sons sont si agaçants qu'ils se voient forcés de quitter les lieux rapidement. Les personnes plus âgées, qui n'entendent plus ces fréquences, ne sont donc pas concernées.

Le terme « ultrason » n'est pas correct dans ce contexte, car en général on ne l'emploie que pour les fréquences dépassant les capacités auditives de l'homme, au-delà de 20 kHz donc. Comme les sons émis contiennent aussi des fréquences inférieures à 20 kHz, ils sont captés non seulement par les adolescentes visés (si leur ouïe le permet encore !), mais plus encore par les nourrissons, les (très) jeunes enfants et les animaux (domestiques). La pression sonore n'est pas très élevée, dit le fabricant, cependant si ce son ennuisant gêne les jeunes, il contrarie encore plus les nourrissons et les petits enfants (sans parler du risque d'endommagement de leur ouïe s'ils y sont exposés longtemps).

Les nourrissons et les animaux (hormis les perroquets peut-être) sont dans l'incapacité de faire savoir ce qui ne va pas ; si donc on se promène en ville avec son bébé, on court le risque que l'enfant soit exposé à ce genre de sons sans que les parents ne le remarquent.

Pour détecter rapidement, où que ce soit, la génération de sons nocifs de fréquences élevées, nous avons conçu un indicateur capable de les détecter et d'en mesurer l'intensité. Dix LED montrent la pression sonore des sons émis entre 16 et 40 kHz. Les fréquences plus basses sont atténuées par un filtre passe-haut Butterworth du cinquième ordre, évitant que l'indicateur ne réagisse aux autres sons courants. La graduation de l'affichage est basée sur une pression sonore de 90 dB maximum. Si l'on est exposé plus d'une heure à 90 dB, on risque un endommagement de l'ouïe de façon irréversible. Les dommages auditifs dus à des fréquences inaudibles, l'usure des ossicules par exemple, n'a jusqu'à présent pas encore fait l'objet d'une étude.

### Schéma

À première vue le schéma électrique de la **figure 1** aurait de quoi impressionner, mais en regardant de plus près, il est moins complexe qu'il n'y paraît. Le circuit se compose d'un préamplificateur de microphone, d'un filtre passe-haut du cinquième ordre, d'un redresseur actif, d'un étage tampon et finalement d'un afficheur à dix LED. Le pilotage des LED se fait à l'aide de l'habituel LM3915 utilisé ici en mode Point (choisi pour limiter la consommation de courant) dans son application la plus simple, sa broche REF ADJ donc reliée à la masse. Dans ce cas, le signal d'entrée à pleine échelle est de  $1,28 V_{CC}$  (typique).

À partir de cette valeur et de la sensibilité du microphone à électret nous pouvons calculer le gain nécessaire. Le microphone utilisé est un modèle disponible chez Farnell (le KEEG1542TBL-A de Kingstate). Ce choix a été guidé par le diagramme de la réponse en fréquence de la fiche technique (disponible en téléchargement chez Farnell). Celui-ci montre une légère tendance montante à 20 kHz, ce qui nous laisse supposer que ce microphone est également adapté aux fréquences juste au-delà de 20 kHz. Même si la réponse en fréquence au-delà de 20 kHz pourra avoir une forme quelque peu bizarre, il ne s'agit ici que d'une mesure approximative censée donner grossièrement la pression sonore.

À 90 dB le microphone fournira environ 8 mV. Si nous tenons compte de la présence dans le redresseur d'un détecteur de crêtes (cela nous fait gagner un facteur de  $\sqrt{2}$  en gain) et du fait que la sensibilité du microphone à électret peut varier de  $\pm 3$  dB, le gain total minimum du montage devrait se situer entre 80 et 160.

On trouve, à l'entrée du montage, un étage amplificateur (IC1A) d'un gain de 40. À ce niveau, les fréquences les plus basses sont déjà atténuées (C1 et C4). C3 et R5 compensent la légère pente au-delà de 20 kHz. Nous avons choisi, comme ampli-op, un quadruple ampli-op rail-à-rail TS924IN de ST, composant capable de fournir pas moins de 80 mA. Cet ampli-op a un GBW (Gain-BandWidth product, produit

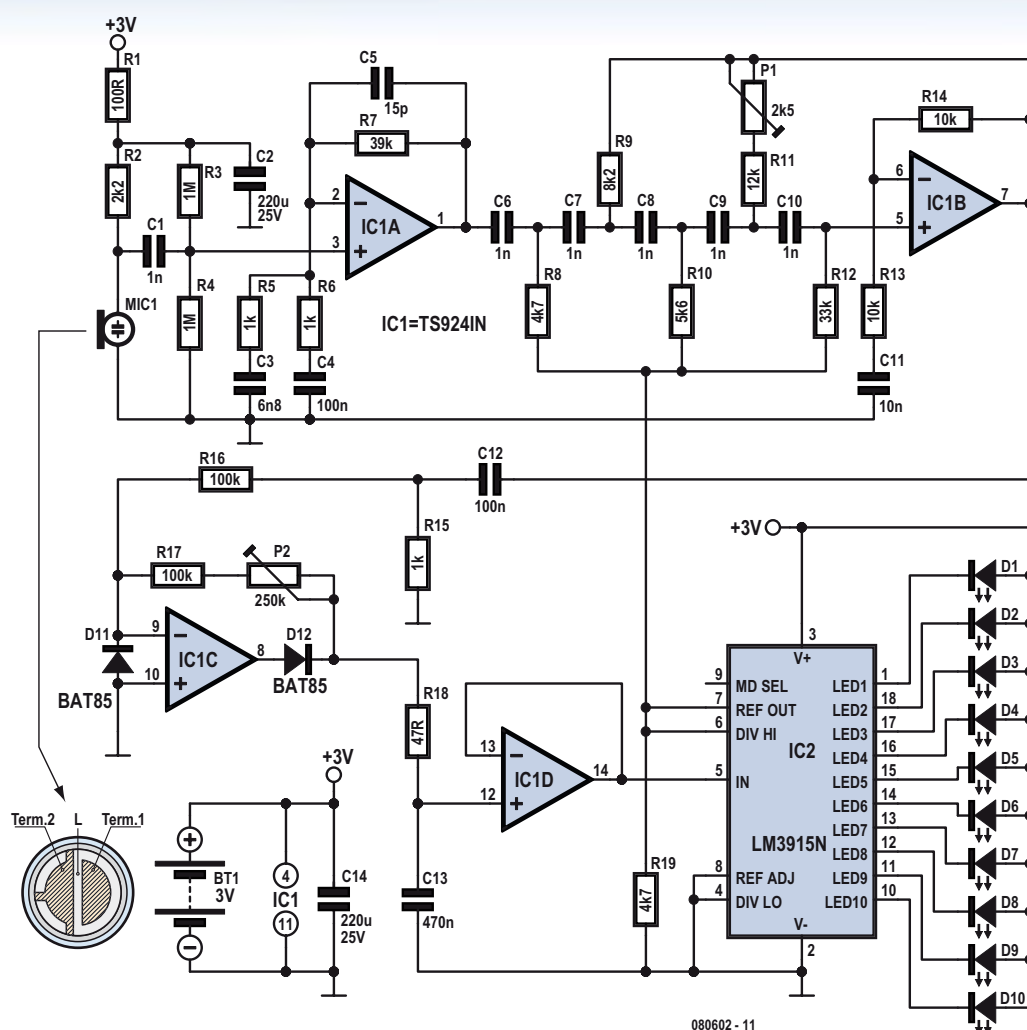
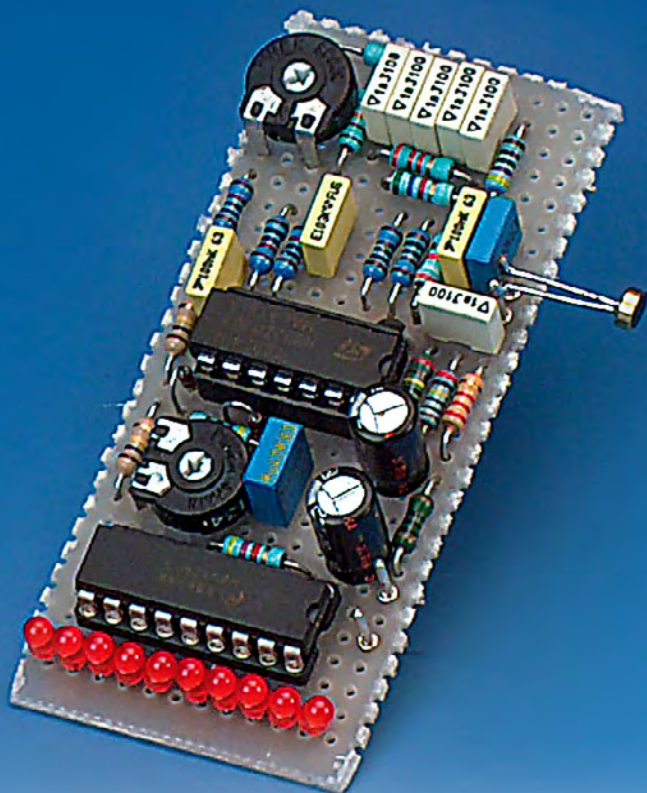


Figure 1. Ce montage permet de détecter des signaux audio de fréquence élevée.



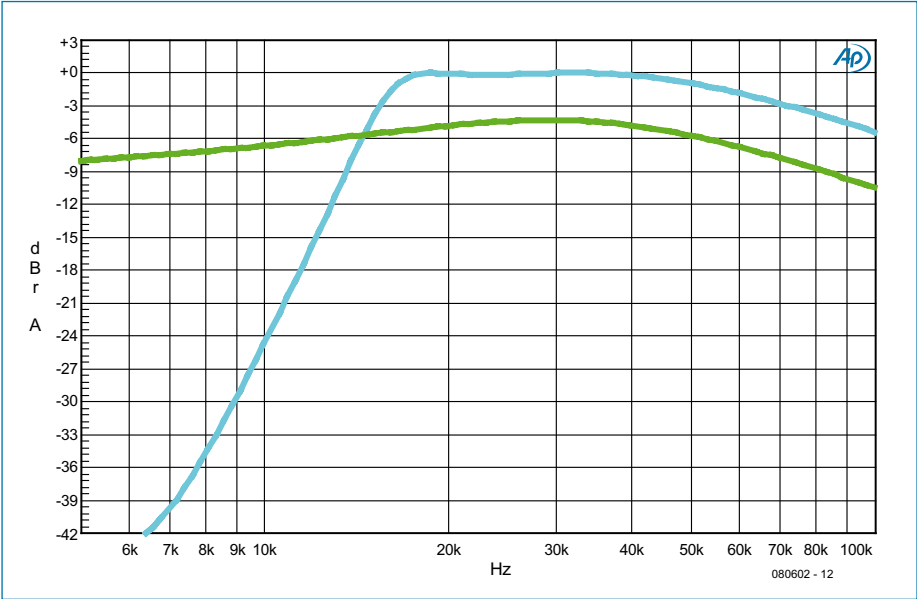


Figure 2. La réponse en fréquence de l'amplificateur d'entrée seul (bleu) et en combinaison avec le filtre passe-haut (vert).

gain x bande passante) relativement important, de 4 MHz, ce qui se traduit par une bande passante de l'amplificateur d'entrée de 100 kHz pour un gain de 40x (C5 sert uniquement au découplage HF), plus qu'il n'en faut pour cette application. En aval de l'amplificateur d'entrée un filtre à pente raide, construit autour d'IC1B, ne laisse passer que les fréquences supérieures à 16 kHz (-3 dB). La **figure 2** montre les réponses en amplitude de l'amplificateur d'entrée seul et en association avec ce filtre. Pour garantir un bon fonctionnement du filtre du cinquième ordre il faut plus qu'un étage-tampon, l'ampli-op doit en outre avoir un gain de 2 (bien visible dans la courbe). Cela tombe bien, car grâce à ce gain supplémentaire les autres étages peuvent avoir un gain un peu moins élevé.

L'un des composants du filtre est ajustable de manière à compenser, si nécessaire, les tolérances des autres composants. Avec P1 on peut ajuster la réponse en fréquence du filtre au plus plat. Si l'on n'a pas les moyens de mesurer cette réponse, on pourra donner à P1 une valeur de 870 Ω. Par contre, il est conseillé de choisir, en mesurant, les condensateurs C6 à C10 de façon qu'ils aient tous la même valeur. Le redresseur basé sur IC1C est un montage standard, où la diode usuelle entre la sortie de l'ampli-op (broche 8) et l'entrée inverseuse (broche 9) n'est pas nécessaire, puisque la sortie ne peut pas devenir négative. Pour un fonctionnement correct, l'entrée du redresseur est couplée en CA (R15/C12) avec la sortie du filtre. D11 protège l'entrée de l'ampli-op contre les surtensions négatives.

La tension à la sortie du redresseur (la cathode de D12) charge, en passant par R18, le condensateur C13 jusqu'à la valeur de crête. R18 limite le courant de charge. Avec les valeurs des composants données, C13 est chargé relativement rapidement, le temps RC n'étant que de 22 μs. Il est possible, en jouant sur P2, d'ajuster le gain total du montage sur une plage assez large, ce qui rend possible l'utilisation d'autres microphones (moins sensibles). Le temps de décharge de C13 varie avec la position de P2, mais cela n'est pas très important. Le quatrième ampli-op (IC1D) est employé comme tampon pour piloter le LM3915. Il est vrai que la résolution du LM3915 est de 30 dB (10 x 3 dB), mais en raison du montage simpliste du redresseur, la tension à travers C13 n'est pas parfaitement proportionnelle au signal d'entrée. La LED en bas s'allume à -21 dB par rapport à la tension nécessaire pour allumer la LED du haut (D10).

La consommation du montage, entre 11 et 15 mA, est assez faible pour permettre 6 jours d'utilisation continue. Le montage fonctionne encore avec une tension d'un peu plus de 2 V, il est donc possible d'utiliser des piles rechargeables au lieu de piles alcalines. C'est quand même mieux pour l'environnement.

Ceux qui aiment expérimenter pourraient essayer de fabriquer une petite parabole pour localiser la source des ultrasons. Le microphone à électret doit être placé dans le foyer de la parabole. Peut-être qu'un petit entonnoir pourrait marcher...

(080602-1)

Publicité

L'offre pertinente pour vos Circuits Imprimés professionnels

**EURO CIRCUITS**

On-line: calculez vos prix  
On-line: passez vos commandes  
On-line: suivez vos commandes  
On-line: 24H/24 & 7J/7

**Nouveau - Pochoirs Pooling**

Pas de minimum de commande !  
Pas de frais d'outillages !

Une équipe novatrice à votre écoute: +33 (0)3 86 87 07 85  
**www.eurocircuits.fr**

**Verified**

- "Standard pooling" à prix très attractifs
- de 1 à 6 couches
- de 1 à 1000 pièces
- délais à partir de 3 jours ouvrés

**A la carte**

- "Technologie pooling" à prix attractifs
- de 1 à 8 couches
- de 1 à 1000 pièces
- délais à partir de 3 jours ouvrés

**On demand**

- "Technologie particulière" au juste prix
- de 1 à 16 couches
- de 1 pièce à la moyenne série
- délais à partir de 3 jours ouvrés

# μWatch : retour de la montre calculatrice scientifique



**David L. Jones (Australie)**

Ces dernières années ont vu le retour de montres rétros en tout genre pour *polars* (les montres de tube de Nixie, montres à LED, montres PONG,...). Ce mois-ci, la page Retronics arbore un visage inhabituel, en présentant μWatch, qui devrait assouvir votre soif nerdinesque de montages faits maison.

La dernière montre calculatrice scientifique fut la Casio CFX-400, sortie en 1985 ! Elle est d'ailleurs tellement recherchée aujourd'hui qu'elle s'arrache, même cassée, à plusieurs centaines de dollars. Et une neuve en état de fonctionnement ? Inutile de demander le prix, vous ne pourriez pas vous le permettre. Ainsi, non seulement il n'était pas possible de s'en procurer pendant ces 20 dernières années (les ingénieurs mis à part), mais vous n'aviez pas non plus le loisir de fabriquer la vôtre jusqu'à aujourd'hui. Le projet μWatch présenté dans ce court article est une occasion unique de raviver les souvenirs de ces montres utilisées à l'université ou au collège dans les années 70 et 80. Levez la main s'il vous plaît !

μWatch est une incarnation moderne de la montre calculatrice, désormais fabriquée avec des composants plus contemporains tels que des CMS et un microcontrôleur PIC. Elle supporte à la fois les modes de calcul algébrique et NPI (Notation Polonaise Inverse). Pourquoi NPI ? Pourquoi pas ! NPI est non seulement simple à coder et à

utiliser, mais il lui reste de plus le soutien d'une large communauté d'utilisateurs de calculatrices HP. Elle a été inventée par Hewlett Packard, qui l'utilisa pour certains de ses tout premiers calculateurs et qui continuent de vendre des calculatrices NPI aujourd'hui.

NPI est basé sur la pile classique HP à 4 niveaux de registres (X/Y/Z/T) et sa fonctionnalité de copie du registre T, comme suit :

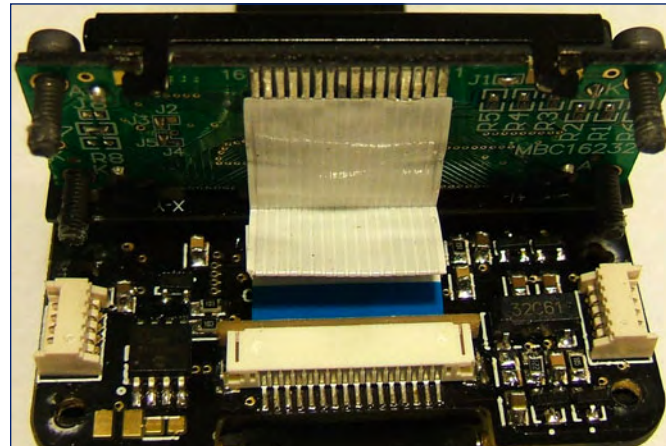
RPN Stack
T Register
Z Register
Y Register
X Register

En mode NPI, les valeurs sont chargées dans la pile en appuyant sur la touche ENTER. La fonction ENTER sera automatiquement exécutée par les opérateurs et fonctions si cela n'a pas déjà été fait.

Tous les opérateurs et fonctions travaillent soit sur X, soit sur X et Y. Par conséquent, pour ajouter 1+2, vous devrez rentrer 1 ENTER 2 +.

Une valeur chargée dans le registre T y reste jusqu'à ce qu'elle soit effacée manuellement, fonction utile pour répéter les calculs.

Les options de menu sont identiques en mode NPI et algébrique, mais elles diffèrent dans leur manière de fonctionner. Comme NPI opère sur les valeurs pré chargées d'une pile, des fonctions telles que  $x^y$



sont effectuées sur des valeurs pré chargées, c'est la « notation postfixe ». En mode algébrique par contre,  $x$  est rentré en premier, suivi par la commande  $x^y$ , puis la routine doit attendre que la valeur  $y$  soit rentrée. C'est la « notation infixe ».

En mode algébrique, certaines opérations telles que le SIN ou LOG utilisent la notation postfixe, comme dans NPI, portant parfois à confusion. Ainsi, vous entrez 10 SIN, et non SIN 10, comme sur les calculatrices algébriques du commerce. C'est pourquoi de nombreuses personnes apprécient NPI, car la notation est toujours postfixe et il n'y a donc jamais de confusion.

Si vous comparez le code source des méthodes NPI et algébrique, vous remarquerez que NPI est beaucoup plus simple et plus facile à comprendre. Le système de pile NPI permet d'effectuer rapidement des calculs complexes en utilisant des commandes très simples, sans parenthèses ni gestion de la priorité des opérateurs. Toutefois, NPI exige que vous pensiez à la façon dont vous faites votre calcul avant de le saisir, ne calculant pas de la même manière que vous voyez ce dernier sur papier.

A contrario, la méthode algébrique nécessite l'emploi de parenthèses emboîtées et une gestion de la priorité des opérateurs, ce qui peut devenir très compliqué dans une perspective de programmation. En revanche, vos calculs sont au final effectués davantage comme vous les voyez

sur le papier.

Le mode algébrique supporte six niveaux de parenthèses, comme la plupart des calculatrices scientifiques de base. La priorité des opérateurs est maintenue complètement avec chaque niveau de parenthèses.

Il y a dans la « pile » algébrique un lot de registres contenant X, Y et Z. Le registre Z est utilisé lorsque l'opérateur de priorité est nécessaire. Par exemple,  $1+2*3$  entraînerait le déplacement du 1 dans le registre Z. Si une parenthèse est ouverte, alors la totalité du contenu de ces registres de travail ainsi que les opérateurs sont décalés d'un niveau dans les registres de travail pour les opérations entre parenthèses. De même, quand une parenthèse est fermée, les registres et les opérateurs redescendent d'un niveau. On utilise donc une pile de sept niveaux de profondeur avec cinq valeurs par niveau. Comparez cela avec la pile simple 4 par 1 de NPI. Pourtant, la pile NPI, utilisée correctement, peut être plus puissante que la pile algébrique.

Plus de détails sur la fabrication et la programmation de μWatch peuvent être trouvés sur le site de l'auteur [1]. Le projet a été développé entièrement avec le compilateur C Microchip C30 sous environnement MPLAB.

(080816-I, trad. Nicolas FRADIN)

## Lien Internet

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

Photograph courtesy Kris Staller

# Le décodeur stéréo

## pour la mini radio FM

Ton Giesberts

Le mois dernier, nous avons publié un mini récepteur FM monophonique. Un décodeur stéréo parachèvera tout naturellement l'ouvrage. Nous y avons titularisé un circuit intégré qui, depuis de nombreuses années, en raison de sa qualité, a largement donné la preuve de son talent.



La petite radio FM décrite le mois passé ne dispose que d'une sortie mono. Nous avons alors précisé qu'il était question d'une version stéréopho-

nique et qu'un mini projet ultérieur y serait consacré. Chose promise, chose due, voici donc le décodeur stéréo destiné à s'associer à la mini radio FM

pour en faire un récepteur à modulation de fréquence complet, une sorte de mise à niveau, dirions-nous.

### D'un certain âge, mais d'une valeur certaine

La puce que nous avons attelée à la tâche de décodeur stéréophonique est un TDA7040T de chez NXP (précédemment Philips). Le montage autour de cette puce n'a pas varié depuis 1986, déjà. L'intégré, tout comme celui du récepteur, n'est disponible qu'en exécution CMS. On ne le produit plus aujourd'hui sous le bon vieux boîtier DIP.

Tout comme pour la radio FM, nous avons conçu une petite platine pour vous faciliter la construction. Dans la même veine, nous avons tâché de respecter des mensurations minimalistes, avec pour résultat l'incorporation de nombreux CMS. Seuls quelques électrolytiques y sont en exécution à fils. C'est ainsi que le circuit imprimé ne mesure que 2,5 sur 4 centimètres.

### Traitement du signal

Le décodeur fonctionne selon le principe de la boucle à phase asservie

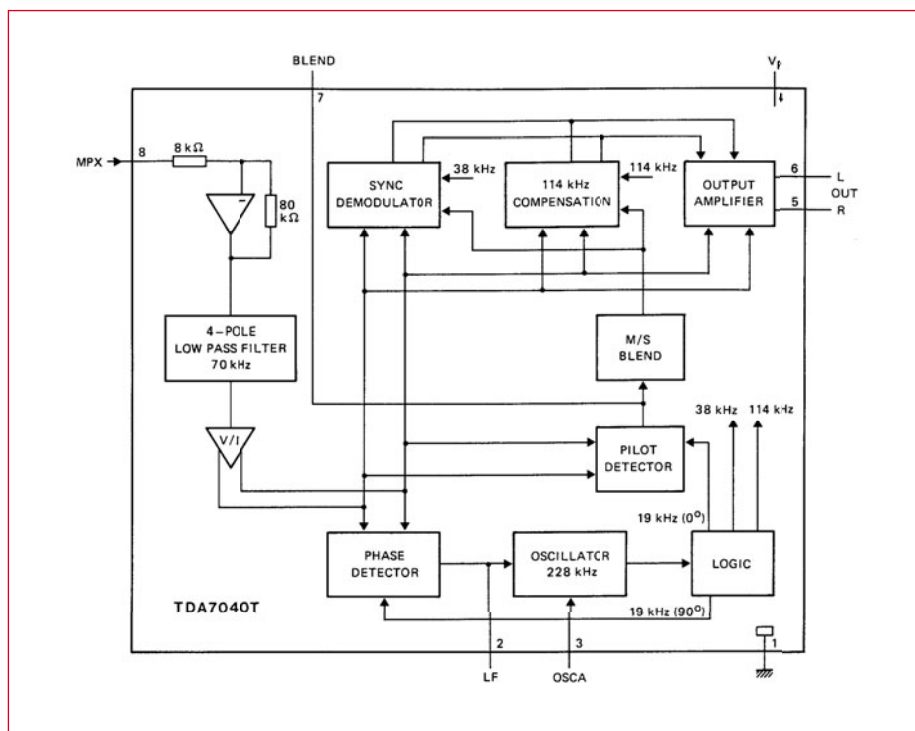


Figure 1. Dans le diagramme synoptique, nous retrouvons l'organisation interne du TDA7040.



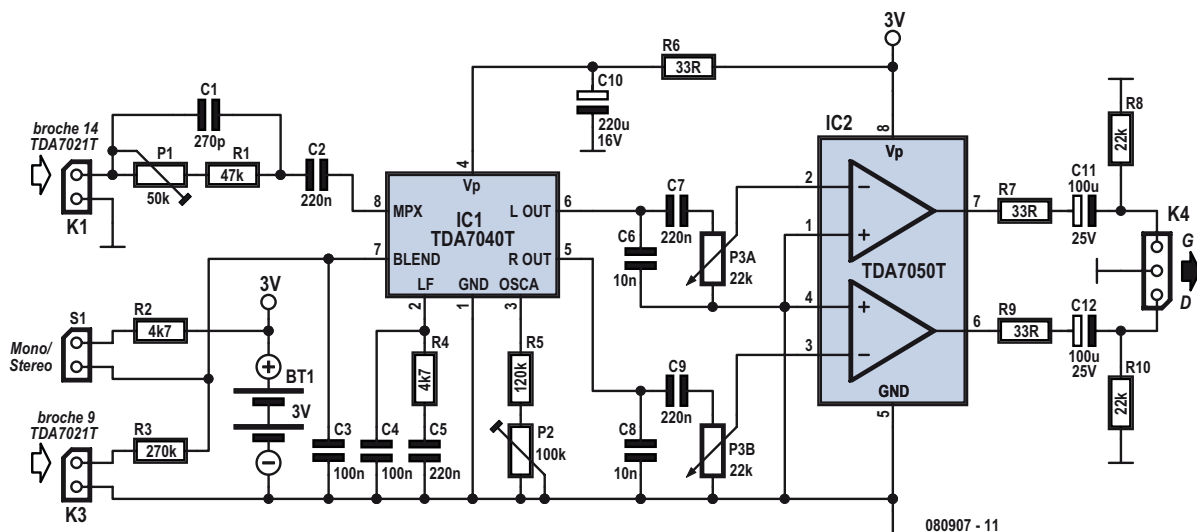


Figure 2. Le schéma du décodeur ne s'écarte guère du montage d'essai décrit dans la feuille de caractéristiques.

(PLL) et ne réclame que quelques composants supplémentaires. La **figure 1** vous en donne le schéma synoptique. La sortie du syntoniseur se branche, à travers un filtre, à l'entrée MPX de la puce. Après amplification et filtrage, le signal sera divisé en trois branches. Commençons par le détecteur de fréquence pilote. Il indique s'il s'agit d'un signal stéréo et commande le commutateur interne mono/stéréo en conséquence. On peut aussi forcer manuellement la monophonie par S1, en toute indépendance. Du filtre, le signal va aussi à la PLL composée d'un détecteur de phase, d'un oscillateur commandé en tension (VCO) et d'un diviseur. Celui-ci fournit les fréquences nécessaires à l'extraction des informations stéréophoniques. Le décodage se passe dans le troisième bloc rejoint par le signal filtré. Finalement, le signal audio stéréophonique quitte la puce en passant par deux tampons. Pour permettre au signal d'attaquer un casque d'écoute ou deux petits haut-parleurs, nous utilisons un TDA7050T. C'est une petite puce d'amplificateur capable de délivrer deux fois 35 mW sur 32  $\Omega$  à partir d'une tension d'alimentation de 3 V.

## Circuit

Vous trouverez le schéma complet du montage à la **figure 2**. La réalisation

s'écarte peu de l'exemple trouvé dans la feuille de caractéristiques, mais à l'instar du circuit de réception, il y a peu de place ici pour l'inconstance. On relie la prise K1 du décodeur à K1 du récepteur ainsi que K3 et K2 (le montage fonctionne aussi sans cette liaison-ci). L'interrupteur mono/stéréo S1 se raccorde à K2 (c'est vraiment K2, mais nous n'avons inscrit que S1 dans le schéma). La sortie K4 sert à brancher

Pour un fonctionnement efficace sur haut-parleurs, la puissance de sortie du TDA7050 est trop faible, mieux vaut lui adjoindre un étage final supplémentaire. Le potentiomètre stéréo de volume (P3) se branche sur une embase à 6 contacts. Si les fils de raccord sont suffisamment courts, nul besoin de câble blindé.

Tous les raccords se font sur embases, mais il est aussi possible de souder directement des fils par les trous correspondants de la platine. Comme le recours à des embases nous forçait à utiliser une carte perforée, nous avons choisi des condensateurs électrolytiques radiaux ordinaires. Ils occupent moins de place et sont généralement de meilleure qualité. Vérifiez bien leur diamètre maximum, il est renseigné dans la liste des composants. À la **figure 3**, vous trouverez d'ailleurs la disposition de ces composants et le tracé des pistes est disponible au téléchargement sur le site d'Elektor.

Sous 6 V d'alimentation, la consommation totale du récepteur et du décodeur est un peu plus élevée qu'avec 3 V : 17,3 mA au lieu de 12,5 mA. Comme vérification, nous avons branché deux mini haut-parleurs de 8  $\Omega$  à la sortie. La consommation à la puissance de pointe atteignait 70 mA, alors qu'en

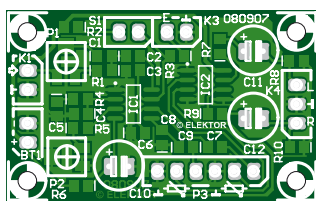


Figure 3. La platine du décodeur n'est pas vraiment plus grande que celle du récepteur.

un ou plusieurs casques d'écoute de 32  $\Omega$ , mais pour un essai, on peut aussi y brancher de petits haut-parleurs. Les résistances de sortie R7 et R9 évitent une surcharge de IC2 tandis que R8 et R10 s'arrangent pour que les condensateurs C11 et C12 soient toujours chargés, ce qui vous épargnera le toc dans les oreilles au moment de l'introduction de la fiche.

régime moyen, elle restait aux environs de 40 mA. Libre à vous de décider si vous utilisez 2 ou 4 piles mignon (R6 ou encore AA). Avec quatre éléments, on utilise toute la capacité des piles, puisqu'il ne faut que 2 V pour fonctionner. La tension minimum du décodeur stéréo est de 1,8 V et celle du récepteur de 1,6 V. Nous avons remarqué à l'usage que l'accord dépendait un peu plus du changement avec une tension basse qu'avec la plus haute.

## Correction

Pour qu'il fonctionne de concert avec le décodeur, le récepteur FM a besoin d'une petite modification : la suppression de C15 et C16 et l'ajout d'un condensateur de 100 pF entre les broches 14 et 15 de la puce. Si l'on en prend un au format 0603, on pourra le souder directement entre les deux broches (cf. **figure 4**).

La raison de cette modification, sans laquelle le décodeur ne peut d'ailleurs pas fonctionner, c'est que le filtre de sortie du récepteur ne réclame plus de désaccentuation (C15) ni d'un gain (C16) aussi grand. À leur place, le filtre devient du type passe-bas du second ordre avec une largeur de bande de 53 kHz environ. Il est

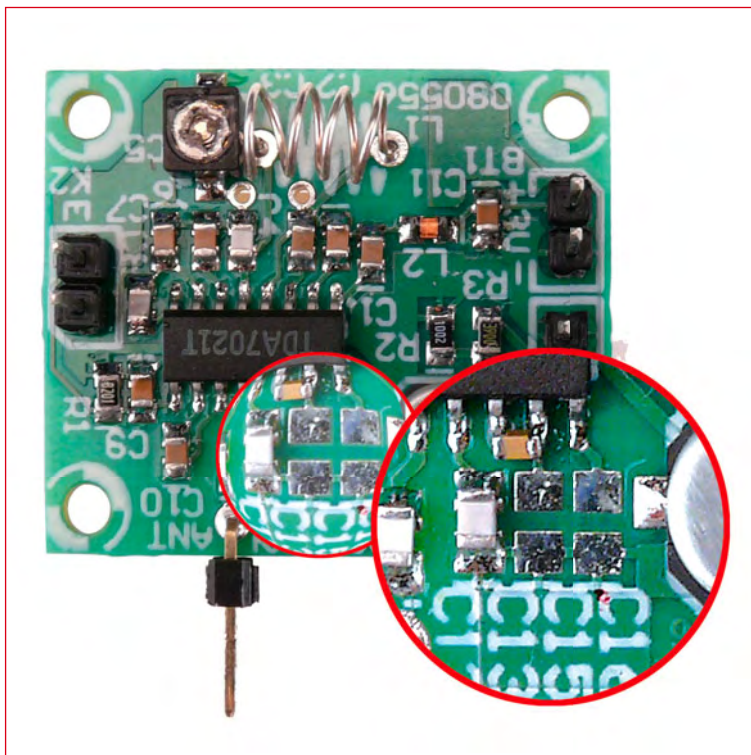
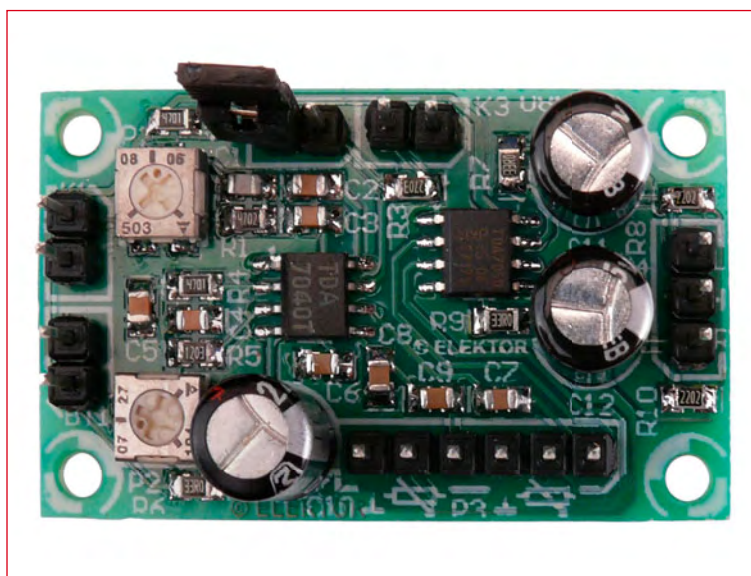


Figure 4. Il y a quelques modifications à apporter à la carte du récepteur, comme l'installation d'un petit condensateur entre les pattes du TDA7021T.



nécessaire de procurer cette bande passante au décodeur qui doit effectuer une démodulation d'amplitude sur 38 kHz avec une onde porteuse supprimée.

## Réglage

Dans le décodeur stéréo, un réseau passif (P1, R1 et C1) accentue le signal multiplex, il faut le régler pour obtenir la meilleure séparation des canaux. C'est le premier des deux ajustements à effectuer.

Une mauvaise position de P1 donnera une reproduction qui ressemble à de la monophonie ou bien l'effet rendu par un réglage de la largeur de base stéréo, l'information mono est atténuée et l'effet stéréo s'en trouve exagéré. Le meilleur réglage, on le trouve (à l'oreille, à défaut d'instruments) quelque part entre ces deux extrêmes. Naturellement, il convient de choisir pour ce faire un excellent émetteur stéréo et une musique d'aussi bonne qualité.

Le second point de réglage est aussi simple à ajuster. Sur une station puissante, on tourne P2 jusqu'à trouver la position idéale pour l'oscillateur (VCO), à égale distance entre les endroits où le décodeur commute en monophonie.

(080907-1)

## Liste des composants

### Résistances (toutes CMS au format 0805) :

R1 = 47 kΩ  
R2, R4 = 4,7 kΩ  
R3 = 270 kΩ  
R5 = 120 kΩ  
R6, R7, R9 = 33 kΩ  
R8, R10 = 22 kΩ  
P1 = 50 kΩ ajustable CMS TS53YJ503MR10 Vishay Sfernice, code de commande Farnell 1557940  
P2 = 100 kΩ ajustable CMS TS53YJ104MR10 Vishay Sfernice, code de

commande Farnell 1557934  
P3 = embase à 6 contacts + potentiomètre stéréo 22 kΩ logarithmique.

### Condensateurs :

C1 = 270 pF CMS 0805  
C2, C5, C7, C9 = 220 nF CMS 0805  
C3, C4 = 100 nF CMS 0805  
C6, C8 = 10 nF CMS 0805  
C10 = 220 μF/16 V radial au pas de 2,5 mm, diamètre 6,5 mm max.  
C11, C12 = 100 μF/25 V radial au pas de 2,5 mm, diamètre 6,5 mm max.

### Semi-conducteurs :

IC1 = TDA7040T CMS SO8  
IC2 = TDA7050T CMS SO8

### Divers :

K1, K3 = embase à 2 contacts  
S1 (K2) = 1 x fermé + embase à 2 contacts  
K4 = embase à 3 contacts  
BT1 = embase à 2 contacts + piles de 3 à 6 V + coupleur de piles  
Platine 080907-1, disponible auprès de [www.ThePCBShop.com](http://www.ThePCBShop.com)

# Hexadoku

## Puzzle pour les électroniciens

La version de février demande encore beaucoup de votre cerveau, patience et endurance.

Mais, vu la quantité des réponses tous les mois, un grand nombre de lecteurs est prêt à faire l'effort en espérant de gagner peut-être un superbe E-blocks Starter Kit Professional ou l'un des 3 bons-cadeau Elektor mis en jeu.

Les instructions pour la résolution de ce puzzle sont enfantines. Le Hexadoku utilise les chiffres du système hexadécimal, à savoir de 0 à F. Du tout cuit pour les électroniciens et programmeurs ! Remplissez le diagramme de 16 x 16 cases de façon à ce que tous les chiffres hexadécimaux de 0 à F (0 à 9 et A à F) ne

soient utilisés **qu'une seule et unique fois** dans chaque rangée, colonne et carré de 4 x 4 cases (identifiés par une ligne plus grasse). Certains chiffres sont déjà placés dans le puzzle et en définissent ainsi sa situation de départ. La solution de ce puzzle vous permettra de gagner de jolis prix. Il vous suffit de nous envoyer la **série de chiffres** en grisé.

### PARTICIPEZ ET GAGNEZ!

Nous tirerons au sort l'une des réponses internationales correctes qui nous seront parvenues; son auteur recevra un

#### E-blocks Starter Kit Professional

d'une valeur de  
**€ 365,75**

nous offrirons en outre  
**3 bons Elektor**  
d'une valeur de **€ 50**

chacun. Faites travailler vos méninges !

Tout recours est exclu de même que le sont, de ce jeu, les personnels d'Elektor International Media B.V. et leur famille.



### OÙ ENVOYER ?

Envoyez votre réponse (les chiffres de la section grisée) **avec vos coordonnées** par E-mail, télécopie ou courrier **avant le 23 février 2009** à :

Elektor c/o Regus Roissy CDG  
Le Dôme - 1, rue de la Haye  
BP 12910 - 95731 Roissy CDG  
Email : hexadoku@elektor.fr

### LES GAGNANTS

La solution de l'Hexadoku du numéro 366 (décembre) est : **E5071**

Le gagnant du **E-blocks Starter Kit Professional** est : **Nicole STAHLSCMIDT** (Kreuzau, Allemagne).

Les **3 bons Elektor** d'une valeur de **€50** chacun vont à : **Pau SASTRE** (La Vall de Uxó, Espagne), **Guido CLAES** (Nijlen, Belgique) et **Helmut BALCKE** (Pabu, France).

Nos félicitations aux lauréat(e)s !

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

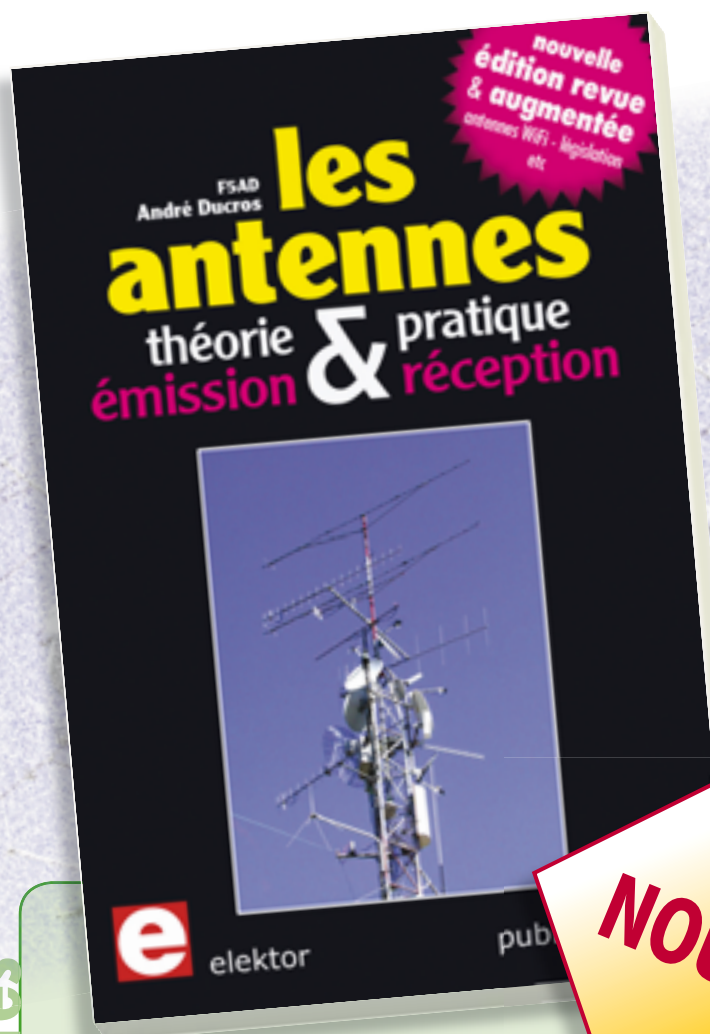
(c) PZZL.com

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



# Morceaux choisis

La bibliothèque des passionnés d'électronique



Nouvelle édition augmentée

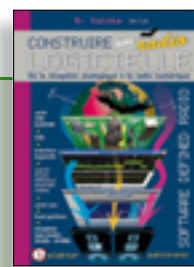
## Les antennes

La première partie traite de la propagation des ondes dans l'espace et sur les lignes ainsi que des caractéristiques fondamentales des antennes (gain, rayonnement, courant, tension...). Cette étude théorique est suivie de réalisations pratiques : antennes filaires, antennes à gain, antennes THF, antennes courtes, antennes à large bande et multibandes, antennes de réception.

La dernière partie est consacrée aux ultimes réglages : adaptation des impédances, appareils de mesure, conseils de sécurité (poussée du vent, résistance des matériaux, pylônes et haubans, foudre...).

472 pages • ISBN 978-2-86661-165-1 • 48,50 €

**NOUVEAU**



Software Defined Radio

## Construire une radio logicielle

Pour dessiner une radio à l'ancienne, prenez un HP et une antenne, et entre les deux des transistors, bobines, transformateurs, diodes, condensateurs etc. Pour une radio logicielle (ou définie par le logiciel), on garde juste l'antenne et les HP du PC. Entre les deux on écrit quelques équations appliquées par un traitement numérique du signal (DSP) sur l'ordinateur. Imaginez les possibilités inouïes de ce procédé !

Le matériel est certes extensible et adaptable, mais les logiciels, par nature, le sont infiniment plus. Ils sont tous mis à disposition gratuitement et enrichis en permanence par des auteurs passionnés.

176 pages • ISBN 978-2-86661-163-7 • 33,50 €



À la recherche de l'optimal par l'étude des circuits élémentaires

## Traité moderne des amplificateurs haute-fidélité à tubes

Les amplificateurs haute-fidélité à tubes électroniques, restés les favoris des audiophiles, reviennent sur le devant de la scène. Pendant cinq ans, les deux auteurs, G. Fiderspil et G. Lallié, ont effectué 6000 simulations informatiques et disséqué pas moins de 1200 maquettes de circuits à tubes pour en livrer les secrets et en découvrir les limites. Dans ce livre, le lecteur acquiert une connaissance approfondie de la conception des amplificateurs à tubes pour mieux apprécier la qualité du matériel existant et optimiser ses propres créations.

344 pages • ISBN 978-2-86661-160-6 • 45,00 €



Écoutez la DRM, c'est magique !

## Construire des récepteurs de radio numérique sur ondes courtes

Ce livre d'Elektor dit tout ce que vous avez toujours voulu savoir sur les récepteurs superhétérodynes, à détection directe, pour les bandes amateur ou bien encore à tubes simples (audion), la transmission de données par radio, la radio numérique, les antennes intérieures, les oscillateurs programmables, les techniques de mesure... Mais il ne s'arrête pas là.

210 pages • ISBN 978-2-86661-157-6 • 35,50 €



Initiation aux microcontrôleurs PIC

## 50 réalisations à PIC, pour le débutant et l'expert

parmi lesquelles *variable speed flashing led, dark activated switch, infrared object detection, pictures of light, electric candle, recording sound, tachometer etc.* Écrit à la demande d'Elektor pour ses lecteurs, tourné vers la pratique, ce livre s'ouvre sur la mise en route du matériel et du logiciel, puis s'articule autour de thèmes variés comme les moteurs, les capteurs, le son ou les mémoires. Idéal pour les débutants et les enseignants, il fait appel à quatre microcontrôleurs : 12F675, 16F628, 16F876A et 16F877, et propose une initiation au langage de programmation open source JAL (*Just Another Language*).

446 pages • ISBN 978-0-905705-70-5 • 37,50 €



Entièrement en couleurs

Traité de paléoelectronique

## Tubes audio anciens & récents

À l'heure des nano-technologies Western Electric fabrique toujours la triode 300B. Qu'y-a-t-il donc de magique dans ces tubes de verre pour qu'ils continuent de nous enchanter ? Quels secrets les rendent irremplaçables à nos oreilles de mélomanes ?

Rien d'autre que le savoir faire transmis – et enrichi – sur plus de quatre générations d'hommes passionnés.

128 pages • ISBN 978-2-86661-155-2 • 39,50 €



Un livre ELEKTOR en anglais

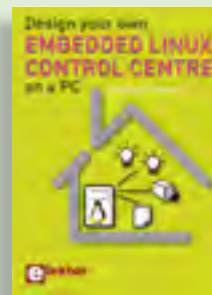
La micro au service de l'électronique

## Visual Basic

for Electronics Engineering Applications

Ce livre invite à un voyage sans passeport dans le monde plein d'imprévu qui s'ouvre derrière les connecteurs d'interface du PC. Avec ce livre dont le titre aurait pu être *Visual Basic as Rapid Application Development Tool* votre ordinateur devient une Machine Universelle. Pilotez vous-même, en Visual Basic, du matériel existant ou des circuits de votre cru !

476 pages • ISBN 978-0-905705-68-2 • 42,50 €



Ce livre est intégralement en anglais

## Design your own Embedded Linux Control Centre on a PC

Grâce à l'évolution récente des techniques de commande et de régulation qui permet une grande efficacité à partir de moyens techniques bien meilleur marché qu'il y a encore quelques années, la domotique est aujourd'hui un sujet encore plus passionnant pour les électroniciens. Ce nouveau livre en anglais ne traite ni de ZigBee, ni de Z-wave ni de X10 ni d'aucun autre protocole commercial, mais propose un système fait à la maison, à partir d'éléments récupérés.

234 pages • ISBN 978-0-905705-72-9 • 32,50 €

Informations complémentaires et gamme complète sur notre site  
**www.elektor.fr**  
flambant neuf !

Elektor / Publitronic SARL  
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  
E-mail : ventes@elektor.fr

**elektor**  
CHOPPE



**Inoculez-leur le virus de l'électronique avant qu'il soit trop tard**

## DVD Elex

Quad, console, manga, rat, portable, MP3... vous cherchez des idées de cadeaux pour vos enfants ou vos petits-enfants ? Inoculez-leur le virus de l'électronique avant qu'il soit trop tard.

Elex revient sur DVD avec la collection complète des 58 numéros du mensuel d'initiation à l'électronique publiés par Elektor d'avril 1988 à sept. 1993, au format PDF imprimable, et un livret de 16 pages en couleur. Bonus : vidéo d'initiation avec les personnages de la bande dessinée Rési & Transi.

ISBN 978-2-86661-156-9 • 44,50 €



**Inusable ! Indispensable !**

## CD Elektor 2007

Ce CD-ROM réunit tous les articles d'Elektor, le mensuel d'électronique et de micro-informatique appliquées, parus au cours de l'année 2007. 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.

ISBN 978-90-5381-218-1 • 27,50 €



**NOUVEAU**

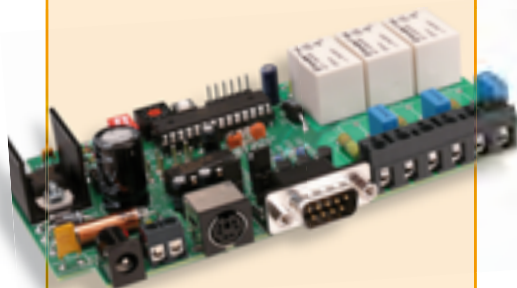
## Toupie électronique

(Elektor décembre 2008)

Alignez et allumez quelques LED sur un plan. Mettez le tout en rotation et vous découvrirez une série de jolis anneaux concentriques. Plus excitant : ajoutez un microcontrôleur et moyennant un peu d'astuce vous pourrez afficher un motif, du texte ou un nombre. Reste la question du séquençement et de la synchronisation. Dès lors les applications s'étendent aux compte-tours et aux boussoles.

*Kit composants, platine avec composants en CMS et contrôleur programmé montés*

Réf. : 080678-71 • 44,95 €



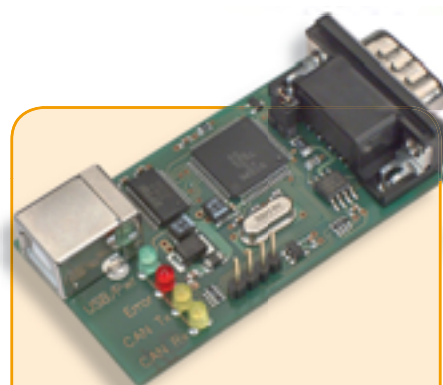
## Télécommande par téléphone portable

(Elektor novembre 2008)

S'il est vrai que les télécommandes GSM sont, dans une certaine mesure, assez répandues, les solutions bon marché et fiables manquent sous bien des aspects. Nous allons vous présenter un projet qui utilise, comme récepteur GSM avec sortie de données, des mobiles très répandus et à des prix extrêmement intéressants (récupération).

*Kit comportant tous les composants avec la platine et le contrôleur programmé*

Réf. : 080324-71 • 69,95 €



## Adaptateur USB-CAN tous usages

(Elektor septembre 2008)

Le protocole CAN est sériel, mais il n'est pas possible de le connecter directement à un ordinateur. L'adaptateur USB-CAN tous usages de cet article offre une solution compacte et pratique. Le logiciel qui l'accompagne permet de suivre le transfert de données et d'effectuer des traitements tels que filtrage ou enregistrement en un tour de main, ou plutôt par un clic de souris.

*Platine partiellement montée*

Réf. : 071120-71 • 69,95 €



## Centrale de commande DCC

(Elektor septembre 2008)

L'essentiel dans un réseau à commande numérique actuel est l'unité de commande qui produit les signaux numériques de conduite des locomotives, mais s'acquitte aussi de tâches annexes, telles que la communication avec un PC. Cette centrale de commande de réseau ferroviaire DCC utilise un puissant processeur ARM7 (celui de la carte de développement ARM-Mee publiée par Elektor en avril 2005, fournie avec le kit).

*Kit comportant tous les composants, platine nue comprise + carte de développement ARM-Mee montée prête à l'emploi*

Réf. : 070989-71 • 119,95 €



**F368 février 2009**+++ Retrouvez sur [www.elektor.fr](http://www.elektor.fr) toutes les références disponibles +++**F367 janvier 2009****Du hertzien pour le contrôleur / Emission-réception réussie !**

071125-71 ..... Platine montée et testée..... 8,50

**De parole et d'argent**

080396-41 ..... Contrôleur programmé ..... 9,95

**Accès au 32 bits**

080632-91 ..... Platine montée et testée..... 44,50

**Détection capacitive et appareils distributeurs**

080875-91 ..... Kit d'évaluation Boutons capacitifs..... 32,50

080875-92 ..... Kit d'évaluation Variateur capacitif ..... 32,50

**F366 décembre 2008****Pilote pour DEL de puissance**

071129-1 ..... Platine ..... 7,50

**Luminaire 3D**

080355-1 ..... Platine ..... 32,50

**Casque sans fil haute fidélité**

080647-1 ..... Platine (émetteur) ..... 12,50

080647-2 ..... Platine (récepteur) ..... 12,50

**Toupie électronique**080678-71 ..... Kit composants, platine avec composants en CMS  
et contrôleur programmé montés..... 44,95**F365 novembre 2008****Commande de volume motorisée**

071135-41 ..... Contrôleur programmé ..... 7,95

**Contrôleur à clé USB**

071152-91 ..... Platine montée et testée, module VDIP1 ..... 29,95

**Télécommande par téléphone portable**

080324-1 ..... Platine ..... 22,50

080324-41 ..... Contrôleur programmé ..... 7,95

080324-71 ..... Kit comportant tous les composants avec la platine et  
le contrôleur programmé ..... 69,95**ATM18 – détecteur IR**

080358-1 ..... Platine ..... 12,50

**Avertisseur de radar**

080615-1 ..... Platine ..... 19,50

080615-41 ..... Contrôleur programmé ..... 14,95

**F364 octobre 2008****Vobulateur HF et analyseur de spectre**

040360-41 ..... Contrôleur programmé ..... 29,95

**ATM18 – carte de relais**

071035-72 ..... Kit Composants: platine, tous les composants et relais..... 49,95

071035-95 ..... Platine Port d'Extension, montée et testée ..... 16,95

**F363 septembre 2008****Récepteur GPS polyvalent**

070309-41 ..... Contrôleur programmé ..... 15,50

**Centrale de commande DCC**070989-71 ..... Kit comportant tous les composants, platine nue comprise +  
carte de développement ARMee montée prête à l'emploi ..... 119,95**Adaptateur USB-CAN tous usages**

071120-71 ..... Platine partiellement montée ..... 69,95

**F361/362 juillet-août 2008****Apesanteur totale !**

071035-71 ..... Kit composants barreau ferrite, HAL815, aimant ..... 12,95

**Horloge pointeuse**

070349-41 ..... Contrôleur programmé PIC12F683 ..... 7,50

**Affichage de consommation pour accu**

070821-41 ..... Contrôleur programmé PIC16F676-20I/P ..... 7,50

# vos favoris

livres	1	↑	<b>12 applications pratiques</b> <small>pour maîtriser le PICBASIC PB-3B</small>	ISBN 978-2-86661-166-8..... 42,50 €
	2	↘	<b>Construire une radio logicielle</b>	ISBN 978-2-86661-163-7..... 33,50 €
	3	→	<b>Les antennes</b>	ISBN 978-2-86661-165-1..... 48,50 €
	4	↗	<small>Traité moderne des</small> <b>amplificateurs</b> <small>haute-fidélité à tubes</small>	ISBN 978-2-86661-160-6..... 45,00 €
	5	↘	<b>Tubes audio anciens &amp; récents</b>	ISBN 978-2-86661-155-2..... 39,50 €
cd & dvd-rom	1	→	<b>Cours FPGA</b>	ISBN 978-90-5381-225-9..... 19,95 €
	2	↗	<b>DVD Elex</b>	ISBN 978-2-86661-156-9..... 44,50 €
	3	↗	<b>CD Elektor 2007</b>	ISBN 978-90-5381-218-1..... 27,50 €
	4	↘	<b>ECD 4</b>	ISBN 978-90-5381-159-7..... 26,50 €
	5	↘	<b>USB Toolbox</b>	ISBN 978-90-5381-212-9..... 29,50 €
kits & modules	1	→	<b>Toupie électronique</b>	Réf. : 080678-71 ..... 44,95 €
	2	↗	<b>DigiButler</b>	Réf. : 071102-71 ..... 39,00 €
	3	↘	<b>Télécommande par téléphone portable</b>	Réf. : 080324-71 ..... 69,95 €
	4	↗	<b>Analyseur OBD-2 compact (Module)</b>	Réf. : 070038-93 ..... 89,00 €
	5	↘	<b>Adaptateur USB-CAN tous usages</b>	Réf. : 071120-71 ..... 69,95 €

Commandez tranquillement sur

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

ou à l'aide du bon de commande encarté à la fin de la revue.

Les commandes en ligne de livres ou de CD &amp; DVD-ROM bénéficient

d'une **remise spéciale de 5%.**


Elektor / Publitrone SARL  
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  
E-mail : ventes@elektor.fr

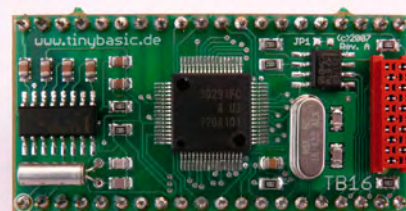


## 16 à partir de 4 - Extension de port facile

Il arrive souvent que le microcontrôleur choisi pour un montage ne possède pas assez de ports pour piloter toutes les fonctions du montage. Notre astucieuse carte d'extension de port est la solution, car elle n'utilise que quatre connexions (SPI). La première carte dispose de 16 sorties binaires et si cela ne suffit pas, il est possible d'en mettre plusieurs en série. Notre deuxième carte, aussi facile à piloter, permet la connexion d'un afficheur LCD compatible HD44780. Les cartes sont accompagnées par des exemples de programmation pour contrôleur AVR.

## TinyBrick - Module M16C au format DIL

Notre carte M16C, programmable en BASIC et en C, a rencontré un franc succès en 2008. Voici une variante plus compacte encore, puisque c'est un module au format DIL. La carte « TinyBrick », qui pourrait servir de cœur pour pleins de montages, est programmable en TinyBasic. Pour faciliter le développement, nous vous proposerons également une simple carte porteuse. Mais ce n'est pas tout : une petite application d'exemple, un système d'alarme capable d'envoyer des SMS, montre comment le module est facile à mettre en œuvre.



## De R8C à R32C

Il y a trois ans, Elektor, ensemble avec la société Glyn, a introduit dans une série d'articles à succès, le R8C/13, un petit contrôleur 16 bits de Renesas. Ce petit contrôleur a récemment eu un frère, le R32C. Ce nouveau contrôleur 32 bits tourne à 50 MHz et possède une unité de calcul en virgule flottante. La programmation du R32C/111 est, comme pour le R8C/13, très facile. Le matériel consiste en une petite carte avec le microcontrôleur et une carte d'application avec les interfaces et une surface d'expérimentation à trous. En plus, il y a une application pour piloter un afficheur OLED...

*Il arrive que la publication de certains articles soit retardée par des impératifs rédactionnels. Attention, le numéro de mars 2009 devrait être en kiosque à partir du 16 février.*



### Prix au numéro

France	6,50 €
DOM Surface	7,50 €
DOM Avion	9,45 €
Belgique	7,10 €
Suisse	12,20 FS
Canada	8.95 \$Can

### Abonnement d'un an standard

France	68,00 €
Belgique	75,00 €
Suisse	126,00 FS
DOM Surface	89,00 €
DOM Surface Priorité	116,00 €

Étudiant -/- 20%

### Abonnement de 2 ans standard

France	120,00 €
Belgique	135,00 €
Suisse	228,00 FS
DOM Surface	160,00 €
DOM Surface Priorité	208,00 €

Étudiant -/- 20%

### Abonnement PLUS d'un an

France	80,50 €
Belgique	87,50 €
Suisse	154,00 FS
DOM Surface	101,50 €
DOM Surface Priorité	128,50 €

Étudiant -/- 20%

### Abonnement PLUS de 2 ans

France	145,00 €
Belgique	160,00 €
Suisse	280,00 FS
DOM Surface	185,00 €
DOM Surface Priorité	233,00 €

Étudiant -/- 20%

*Sous réserve de modification de prix.*

### Abonnements

E-mail : abonnements@elektor.fr

### Commandes/Ventes

E-mail : ventes@elektor.fr

*Il est possible de faire démarrer un abonnement à tout moment. Nous vous rappellerons en temps utile l'approche de la fin de votre abonnement. La méthode la rapide et la moins chère de vous*

*abonner est de le faire par le biais de notre site Internet [www.elektor.fr/abo](http://www.elektor.fr/abo), mais vous pouvez également le faire à l'aide du bon de commande se trouvant en fin de magazine. Il est possible de commander d'anciens numéros dans la limite de leur disponibilité (cf. le bon de commande, leur prix est celui d'un numéro à l'unité).*

*Veuillez SVP nous fournir un changement d'adresse au moins 3 semaines auparavant en mentionnant votre numéro d'abonné (cf. le label accompagnant votre magazine), l'ancienne et la nouvelle adresse.*

*Le département Clients est accessible du lundi au jeudi de 8h30 à 17h00 et le vendredi de 8h30 à 12h30.*

*Si vous avez des questions concernant votre abonnement, vous pouvez appeler ce département au numéro 01.49.19.26.19.*

*Pour le traitement de votre abonnement, Elektor vous demande des données personnelles. Conformément à la loi « Informatique et Liberté », vous bénéficiez d'un droit d'accès à ces données et vous pouvez en demander la rectification. Sauf refus écrit de votre part auprès du service Abonnement, ces informations pourront être utilisées par des tiers.*

atuitement  
P3 (2 Go). \*

atuitement  
P3 (2 Go). \*

atuitement  
P3 (2 Go). \*

atuitement  
P3 (2 Go). \*

- atuitement  
P3 (2 Go). \*

atuitement  
P3 (2 Go). \*

atuitement  
P3 (2 Go). \*

atuitement  
P3 (2 Go). \*

atuitement  
P3 (2 Go). \*

atuitement  
P3 (2 Go). \*

atuitement  
P3 (2 Go). \*

atuitement  
P3 (2 Go). \*

atuitement  
P3 (2 Go). \*

atuitement  
P3 (2 Go). \*

atuitement  
P3 (2 Go). \*

atuitement  
P3 (2 Go). \*

atuitement  
P3 (2 Go). \*

atuitement  
P3 (2 Go). \*

atuitement  
P3 (2 Go). \*

atuitement  
P3 (2 Go). \*

atuitement  
P3 (2 Go). \*

atuitement  
P3 (2 Go). \*

atuitement  
P3 (2 Go). \*



POUR VOS COMMANDES (kits, modules, livres, CD & DVD-ROM etc.)  
VEUILLEZ INDIQUER ICI VOS COORDONNÉES COMPLÈTES :

☐ Je commande les références indiquées au verso  
pour un montant de  € par :

Mode de paiement (cochez la case de votre choix) :

- ☐ Chèque bancaire / CCP à l'ordre de ELEKTOR
- ☐ Virement bancaire pour la France : ABN AMRO Paris  
Compte : 18739-00001-00200797026-03
- ☐ Virement bancaire pour la Belgique :  
CCP 000-1347214-78 à Warneton
- ☐ Virement bancaire International : ABN AMRO Paris  
IBAN : FR76 1873 9000 0100 2007 9702 603 (BIC : ABNAFRPP)

☐ Carte de crédit (complétez et signez ci-dessous s.v.p.) :

Numéro de carte de crédit (MasterCard ou VISA)

Date de validité

Signature

Voici mon adresse :

Nom

Prénom

Date de naissance

FR

Le cas échéant, veuillez indiquer votre numéro d'abonné

Société

Adresse

Code Postal

Localité

E-mail

Pays DOM/TOM

Forfait de port et d'emballage :

- 8.50 € envoi standard France métropolitaine & Europe
- 12.50 € envoi standard DOM/TOM et le reste du monde
- 10.00 € envoi express France métropolitaine & Europe
- 15.00 € envoi express DOM/TOM et le reste du monde

Code de vérification  
(les trois derniers chiffres au dos de la carte)

Envoyez cette carte

accompagnée du paiement

sous enveloppe affranchie à :

Elektor / Publitronic SARL  
c/o Regus Roissy CDG  
1, rue de la Haye  
BP 12910  
95731 Roissy CDG Cedex  
FRANCE

Oui, je souscris un abonnement d'un an à Elektor (11 numéros).  
Dès l'enregistrement de mon paiement, je recevrai automatiquement  
et sans aucun engagement de ma part le baladeur MP3 (2 Go).\*

Cochez la case correspondant à l'abonnement de votre choix :

- ☐ Abonnement standard d'un an pour 68,00 € (Belgique : 75,00 €)
- ☐ Abonnement ELEKTOR PLUS d'un an pour 80,50 € (Belgique : 87,50 €)  
(le DVD-ROM annuel 2008 compris\*\*)

Mode de paiement (cochez la case de votre choix) :

- ☐ Chèque bancaire / CCP à l'ordre de ELEKTOR
- ☐ Virement bancaire pour la France : ABN AMRO Paris  
Compte : 18739-00001-00200797026-03
- ☐ Virement bancaire pour la Belgique :  
CCP 000-1347214-78 à Warneton
- ☐ Virement bancaire International : ABN AMRO Paris  
IBAN : FR76 1873 9000 0100 2007 9702 603 (BIC : ABNAFRPP)

☐ Carte de crédit (complétez et signez ci-dessous s.v.p.) :

Numéro de carte de crédit (MasterCard ou VISA)

Date de validité

Signature

Voici mon adresse :

Nom

Prénom

Adresse

Code Postal

Pays DOM/TOM

Localité

Envoyez cette carte

accompagnée du paiement

sous enveloppe affranchie à :

Elektor / Publitronic SARL  
c/o Regus Roissy CDG  
1, rue de la Haye  
BP 12910  
95731 Roissy CDG Cedex  
FRANCE



# RUBRIQUE PUBLI-RÉGION

Liste d'annonceurs classés par pays, puis par région  
(fabricants, distributeurs, revendeurs, librairies...).

## REGION NORD

### BE. RECHERCHE

Ingénieur ou Electronicien compétent, esprit autonome, conception et programmes en langageC, spécialisé 'Microchip'.

#### Mission :

Evolution produits actuels, actualisés par nouveau concept breveté.

- A créer : Programme de saisie instructions sur ordinateur en Visual-Basic.
  - Programme appareil de recueil des instructions par USB.
  - Programmes des appareils contrôlés.
- Outils et environnement adaptable.

Poste à pourvoir dans le Pas de Calais.  
25 Km de Lille  
CV+ Prétentions à : ttemp@jouvence.fr

## FRANCHE COMTÉ (39)

### IMPRELEC

#### NOUVELLE ADRESSE

32, rue de l'Egalité - 39360 VIRY  
Tél: 03 84 41 14 93 - Fax: 03 84 41 15 24  
E-mail: imprelec@wanadoo.fr

Réalise vos CIRCUITS IMPRIMES de qualité professionnelle SF ou DF, étamés à chaud et percés sur V.E. 8/10° ou 16/10°, (Eillets, trous métallisés, sérigraphie, vernis d'épargne. Face aluminium et polyester multicolours pour façade.

De la pièce unique à la série, vente aux entreprises et particuliers. Tarifs contre une enveloppe timbrée, par Tél. ou mail.

Pour tous renseignements sur cette rubrique,  
veuillez contacter

SL Régie - Sophie Lallonder  
12, allée des Crételles, 37300 Joué-Lès-Tours  
Tél. : 02 47 38 24 60  
E-mail : sophie.lallonder@wanadoo.fr

## Répertoire des annonceurs

BETA LAYOUT	www.pcb-pool.com	21
EUROCIRCUITS	www.eurocircuits.fr	74
IMPRELEC		87
JOUVENCE		87
LEXTRONIC	www.lextronic.fr	88
MIKROELEKTRONIKA	www.mikroe.com	14, 15
SCHAEFFER	www.schaeffer-ag.de	21

## Réservation d'espaces publicitaires

Réservez dès aujourd'hui votre espace publicitaire dans le  
magazine Elektor du mois de avril 2009 !  
Date limite de réservation : **le 17 février 2009**

Pour toute information concernant la publicité  
aussi bien dans notre magazine que sur notre site internet  
[www.elektor.fr](http://www.elektor.fr) contactez :

SL Régie - Sophie Lallonder  
12, allée des Crételles, 37300 Joué-Lès-Tours  
Tél. : 02 47 38 24 60  
E-mail : [sophie.lallonder@wanadoo.fr](mailto:sophie.lallonder@wanadoo.fr)

# Cours FPGA en 9 leçons



Le train des FPGA  
est en marche,  
montez-y avant  
qu'il ne prenne  
de la vitesse !



Il en va un peu aujourd'hui des FPGA comme autrefois des microcontrôleurs : apparus dans les années 70, les  $\mu$ C ne se sont imposés que bien plus tard, une fois les outils appropriés mis au point et disponibles pour le plus grand nombre. Les Field Programmable Gate Arrays, c'est-à-dire « réseaux de portes logiques programmables de l'extérieur », brillent par leurs performances et leur versatilité. Pourtant leur utilisation dans des réalisations est encore trop timorée. L'avènement récent des nouveaux compilateurs C2H va accélérer le mouvement. Désormais les FPGA ont le profil requis pour battre en brèche le monopole des  $\mu$ C.

Elektor a réuni sur CD-ROM son cours en 9 leçons, qui entend mener le lecteur de la logique numérique à la construction d'un serveur web FPGA.

ISBN 978-90-5381-225-9 • 19,95 €

Elektor / Publitrone SARL  
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  
E-mail : [ventes@elektor.fr](mailto:ventes@elektor.fr)

**elektor**  
CHOPPE

Informations complémentaires et catalogue complet sur  
[www.elektor.fr/e-choppe](http://www.elektor.fr/e-choppe)



## Développements sur PIC / dsPIC™



**EasyPIC5** Platine de développement pour **PIC 8 à 40 broches** - Programmeur / Débogueur \* USB et nombreux périphériques intégrés (Leds, BP, 7 segments...) - LCD en option.



**BIGPIC5** Platine de développement pour **PIC 64 et 80 broches** - Programmeur / Débogueur \* USB et périphériques intégrés (Leds, BP, connecteur SD™...) - LCD en option.



**LV18FJ** Platine pour **PIC18FxxJxx** avec **64, 80 et 100 broches** - Programmeur / Débogueur \* USB et périphériques (Leds, BP, port série, Ethernet...) - LCD en option.



**LV24-33A** Platine pour **PIC24F/24H/dsPIC33F** avec **64, 80 et 100 broches** - Programmeur / Débogueur \* USB et périphériques (Leds, BP, ports séries...) - LCD en option.



**EasydsPIC4** Platine pour **dsPIC™** avec **18, 28 et 40 broches** - Programmeur / Débogueur \* USB et périphériques intégrés (Leds, BP, Leds, port série...) - LCD en option.



**dsPICPRO4** Platine pour **dsPIC™** avec **64 et 80 broches** - Programmeur / Débogueur \* USB et périphériques intégrés (Leds, BP, Leds, Ethernet...) - LCD en option.



**PICFlash** Programmeur ICSP / Débogueur \* à connexion USB pour microcontrôleurs **PIC** à technologie Flash. Existe pour **PIC 18FxxJxx**, microcontrôleurs **PIC24**, **dsPIC33** et **dsPIC™**.

\* Le mode débogueur est uniquement compatible avec les compilateurs ci-dessous



**MikroBASIC** Compilateur Basic pour PIC ou dsPIC™

**MikroPASCAL** Compilateur Pascal PIC ou dsPIC™

**MikroC** Compilateur "C" pour PIC ou dsPIC™



**PIC-P14** Platine de développement pour **PIC 14 broches** (PIC non livré) - Interface RS232 intégrée.



**PIC-P18** Platine de développement pour **PIC 18 broches** (PIC non livré) - Interface RS232 intégrée.



**PIC-P28** Platine de développement pour **PIC 28 broches** (PIC non livré) - Interface RS232 intégrée.



**PIC-P28-USB** Platine de développement pour **PIC 28 broches** (PIC non livré) - Connecteur USB intégré.



**PIC-P40** Platine de développement pour **PIC 40 broches** (PIC non livré) - Interface RS232 intégrée.



**PIC-P40-USB** Platine de développement pour **PIC 40 broches** (PIC non livré) - Connecteur USB intégré.



**PIC-USB-STK** Platine d'évaluation pour **PIC 18F4550** (USB + connecteur carte SD™)...



**PIC-18LF8490** Platine d'évaluation pour **PIC18LF8490** + afficheur LCD + interface série.



**PIC-MT-USB** Platine pour **PIC 40 broches** (PIC non livré) - LCD et interface série / USB intégrés.



**PIC-P32MX** Platine d'évaluation pour **PIC 32MX340F512** - Port série et zone de développement.



**PIC-P67J60** Platine d'évaluation pour **PIC 18F67J60** (avec contrôleur Ethernet intégré) + Port série.



**PIC-MINI-WEB** Platine serveur web avec **PIC 18F25J10** + contrôleur Ethernet ENC28J60.



**PIC-WEB** Platine serveur web avec **PIC 18F452** + contrôleur Ethernet ENC28J60.

## Modules radiofréquences

Modem radio **ZigBee™** permettant une liaison série entre 2 microcontrôleurs (2 modules sont nécessaires) - Dim.: 24 x 10,5 mm - Alim.: 3,3 V Prix unitaire **22,72 €**

**F2M03GLA** Module **Bluetooth™** permettant une liaison série avec périphérique Bluetooth™ au protocole SPP - Dim.: 28,5x 15,2 mm - Alim.: 3,3 V Prix unitaire **32,72 €**

**TDL2A** Modem radio **synthétisé 5 canaux bande 433 MHz** permettant une liaison série transparente entre 2 microcontrôleurs (2 modules nécessaires) Prix unitaire **40,66 €**

**SET150** Ensemble de 2 **télécommandes** porte-claf 433,92 MHz type monocanal à code anti-scanner + 1 **récepteur** à sortie relais - Portée: 30 m **49,00 €** Eco-participation 0,01 € inclus

**TM2** Module **GSM/GPRS** Quad Band - Compatible protocole voix, fax, SMS - Pilotage très simple via commandes AT séries - Prévoir antenne en sus **59,00 €**

**ET-312** Module **GPS** 20 canaux - Dimensions: 27,9 x 20,2 mm - SIRF III™ - Alim. 3,3 V - Prévoir antenne externe - Prix unitaire **70,56 €** Prix unitaire (par 5 pcs) **58,60 €**

**EM-406** Module **GPS** 20 canaux avec antenne intégrée - Dimensions: 30 x 30 x 10,5 mm - SIRF III™ - Alim. 5 V - Prix unitaire **75,00 €** Prix unitaire (par 5 pcs) **64,58 €**

**UM005** Module de lecture/décodage TAG **RFID 125 KHz Unique™** - Sortie série **25,00 €**

**RFID-CARD1** Carte RFID Unique **2,00 €** Prix unitaire (par 20 pcs) **1,32 €**

**AJV24E** Module émetteur vidéo 2,4 GHz 4 canaux - Dim.: 31 x 29 x 4 mm ..... **12,95 €**

**AJV24R** Module récepteur vidéo 2,4 GHz 4 canaux - Dim.: 41 x 32 x 6 mm ..... **19,95 €**

**AJV58E** Module émetteur vidéo 5,8 GHz 4 canaux - Dim.: 37 x 31 x 4 mm ..... **17,39 €**

**AJV58R** Module récepteur vidéo 5,8 GHz 4 canaux - Dim.: 51 x 31 x 13 mm ..... **24,95 €**

**CAF257** Moniteur oem TFT couleur 2,36" (56 x 43,5 mm) livré sous forme d'une platine Alim.: 5 à 12 Vcc / 200 mA - Entrée vidéo 1 Vpp ..... **49,94 €**

## Développements sur ARM™



**LPC-H2124** Module sur base d'un **ARM7TDMI™** LPC2124 - Port série et JTAG intégrés.



**LPC-P2138** Patine sur base d'un **ARM7TDMI™** LPC2138 - Nombreux périphériques.



**LPC-H2294** Patine sur base d'un **ARM7TDMI™** LPC2294 - Port USB et JTAG intégrés.



**LPC-L2294** Platine sur base d'un **ARM7TDMI™** LPC2294 - Nombreux périphériques intégrés.



**LPC-2378STK** Platine sur base d'un **ARM7TDMI™** LPC2378 - Nombreux périphériques intégrés.



**LPC-E2468** Platine sur base d'un **LPC2468** avec uCLinux embarqué - Nombreux périphériques.



**STR-H711** Module sur base d'un **ARM7TDMI™** STR711FR2T6 - Port USB et JTAG intégrés.



**STR-P712** Module sur base d'un **ARM7TDMI™** STR712FR2T6 - Nombreux périphériques.



**STR-E912** Module sur base d'un **ARM966E-S** STR912FW44X - Nombreux périphériques.



**TMS470-P256** Module sur base d'un **ARM7TDMI™** TMS470A256 - Nombreux périphériques intégrés.



**STM32-H103** Module à base de **ARM Cortex-M3** STM32F103RBT6 - Port USB et JTAG intégrés.



**STM32-P103** Patine à base de **ARM Cortex-M3** STM32F103RBT6 - Nombreux périphériques.



**SAM7-H256** Module sur base d'un **ARM7TDMI™** AT91SAM7S256 - Port USB et JTAG intégrés.



**SAM7-P256** Platine sur base d'un **ARM7TDMI™** AT91SAM7S256 - Nombreux périphériques.



**SAM7-EX256** Platine sur base d'un **ARM7TDMI™** AT91SAM7S256 - Nombreux périphériques.



**SAM7-LA2** Platine sur base d'un **ARM7TDMI™** AT91SAM7A2 - Nombreux périphériques.



**SAM9-L9260** Platine sur base d'un **ARM9™** AT91SAM9260 - Nombreux périphériques.



**SAM9-L9261** Platine sur base d'un **ARM9™** AT91SAM9261 - Nombreux périphériques.



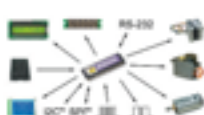
**CS-E9302** Module sur base d'un **ARM920T** EP9302 - Ports USB, Ethernet, RS232....



**ADuC-P7026** Module sur base d'un **ARM7TDMI™** ADuC-P7026 - Nombreux périphériques.

## Développements sur PICBASIC

Vos connaissances en microcontrôleurs sont limitées (ou nulles) ? Vous avez un budget "serré" et vous voulez développer des applications capables de piloter des afficheurs LCD alphanumériques ou graphiques, des communications séries, I2C™, SPI™, des signaux PWM, mesurer des valeurs analogiques, piloter des servomoteurs, des moteurs pas-à-pas, des moteurs "cc"... Alors comme des milliers d'utilisateurs (professionnels, particuliers, centres de recherche, écoles d'ingénieurs, industriels, etc...) découvrez les **PICBASIC** !



Ces microcontrôleurs se programment en langage BASIC grâce à un logiciel (**disponible en libre téléchargement**) qui transférera vos instructions dans leur mémoire par un câble raccordé à un PC. Une fois "téléchargés", ces derniers pourront être déconnectés de l'ordinateur pour être totalement autonomes. Leur prise en main est immédiate grâce à leur documentation **entièrement en Français**.

**PICBASIC-3B** (Mémoire 4 K flash - 79 octets de RAM - 18 entrées/sorties) ..... **28 €**  
**PICBASIC-3H** (Mémoire 4 K flash - 79 octets de RAM - 29 entrées/sorties) ..... **37 €**

### Ouvrage technique

#### 12 applications pratiques pour maîtriser le PICBASIC-3B

Référence: 8666 ..... **42,50 €**

Cet ouvrage propose 12 applications pratiques pour le microcontrôleur PICBASIC-3B dans les domaines de la domotique (gradateur à 2 voies pour convecteurs, thermomètre numérique, gestionnaire d'éclairage), de la protection des biens (centrale d'alarme, disjoncteur programmable), de la mesure (CO-mètre, luxmètre, capacimètre, station météo), de l'automatisation (automate programmable) et de l'électronique de puissance (alimentation numérique, variateur de vitesse à commande PWM). L'auteur décrit chaque application en détail, avec toutes les informations propres à la réalisation (circuit imprimé, liste et implantation des composants, mise au point), puis fait une lecture commentée du programme BASIC.

