

elektor

convertisseur USB ↔ 4 UART

RS-232 et RS-422/RS-485
& concentrateur pour
3 ports USB

■ sans microcontrôleur
■ ni logiciel d'aucune sorte

MADE IN
MUNICH

Visual Basic sur Raspberry Pi | alimentation pour PIC à arrêt contrôlé | débogage d'applications Arduino avec Atmel Studio | détecteur universel | shield polyvalent pour Arduino balise de charge Li-Ion par USB | l'atelier du microcontrôleur (7) | banc d'essai : kit Ido WiSmart hors circuits : PCB : bourdes en HF | rétronique : générateur audio HP (1960) | palpeurs de courant intégrés | condensateurs de traversée CMS | cube à LED RGB avec Flowcode



Commander confortablement !

Il vous suffit de scanner le code QR et tous les articles dont vous avez besoin au démarrage de Red Pitaya sont dans le panier de produits !

Micro SDHC-Card pour le Red Pitaya

- Lecture : 24 MB/s
- Spécifications : Classe 6


 SAMS MB-MS08DA 8 GB **9,10**
Tête de détection modulaire du Red Pitaya

- Répartition : x 1/x 10
- Largeur de bande : 15/150 MHz
- Temps de montée : 24/2,3 ns


 TESTEC LF 312 **16,95**
Adaptateur Micro-USB du Red Pitaya

- Tension d'entrée : 100–240 V, 50–60 Hz
- Sortie : 5 V DC, 1,2 A


 NT MICRO USB 1,2 1,40 m **3,95**
Cat. Câble de réseau 5 du Red Pitaya

- Prise mâle RJ45
- Couleur : noir


 CABLE PATCH 1 SW 1 m **0,80**
Adaptateur HF du Red Pitaya

Prise mâle SMA sur connecteur femelle BNC


 K 512 **1,15**

<http://rch.lt/RPC>

CPU/FPGA :
 Xilinx Zynq 7010 SoC,
 Dual ARM® Cortex™-A9


RED Pitaya

Laboratoire de mesure Open-Source, 50 MHz


Caractéristiques:
446,25

- Canaux : 2
- Largeur de bande : 50 MHz
- Taux d'échantillonnage : 125 MS/s
- Base temps horizontale : 4 ns
- Résolution verticale : 14 bits
- DDR3 RAM 512 MB
- Alimentation électrique : micro-USB
- Dimensions : 107 x 21 x 60 mm

Avec le RedPitaya, transformez votre smartphone, votre tablette ou votre PC en un instrument complet !


Le Red Pitaya :

Red Pitaya est un appareil de mesure Open-Source qui a le format d'une carte de crédit et que vous pouvez configurer individuellement selon vos spécificités. Vous pouvez ainsi remplacer une multitude d'appareils de mesure coûteux tels que les oscilloscopes, les générateurs de fonction, les analyseurs de spectres, etc. par un seul

appareil. L'appareil est intégré dans un réseau, il fonctionne sur un système d'exploitation GNU/Linux et peut être programmé au cas par cas avec une multitude d'interfaces logicielles. L'interface utilisateur pour toutes les procédures de mesure est assurée par un service Internet à l'aide de pages HTML. Ce qui permet de piloter la plateforme de mesure à l'aide de smartphones, de tablettes ou d'ordinateurs.

 Commander maintenant! **www.reichelt.fr**

Assistance téléphonique en anglais : +49 (0)4422 955-333

Les langues de notre boutique:



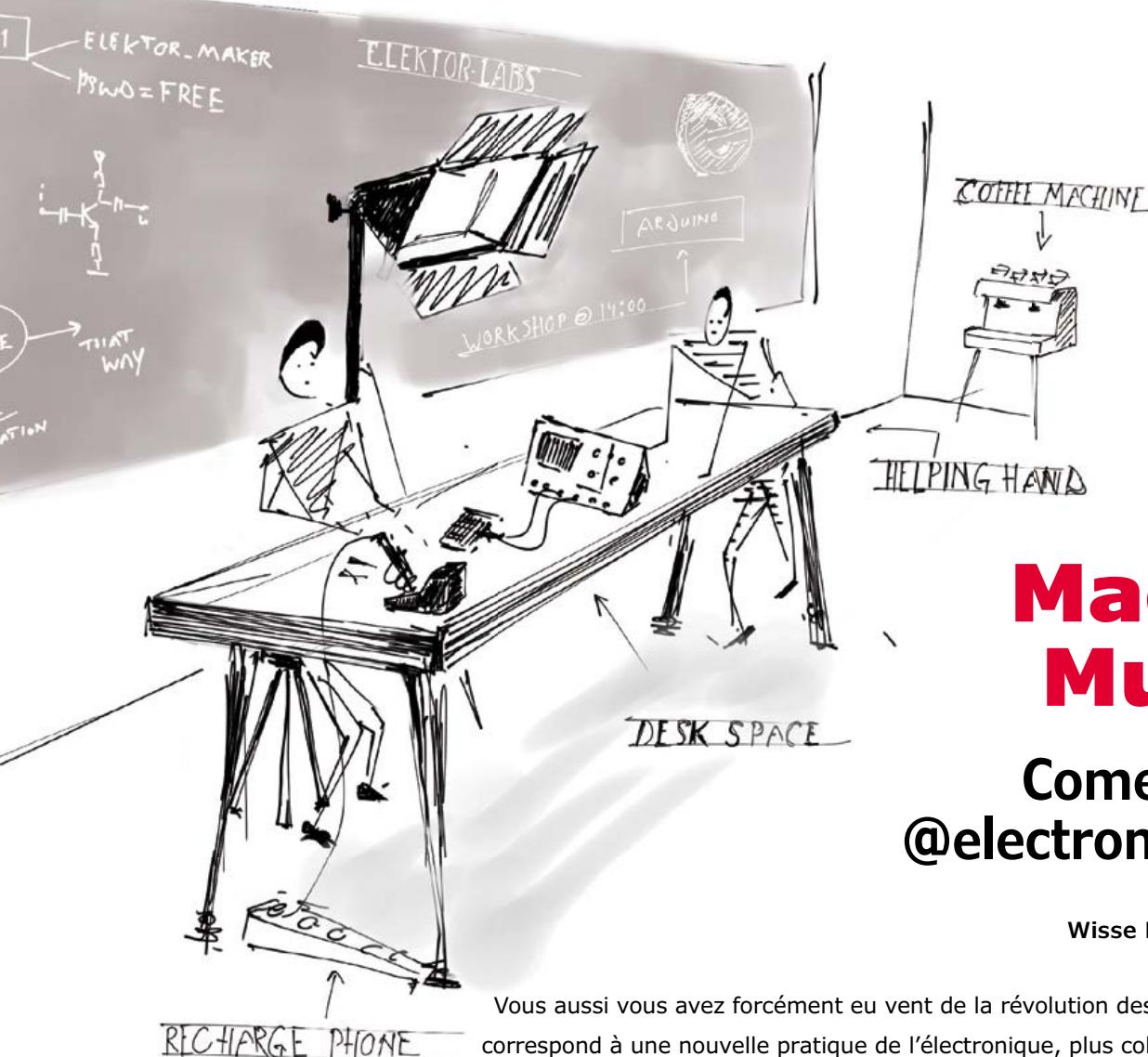
Modes de paiement internationaux:



Prix du jour ! Prix à la date du : 05. 09. 2014

Pour les consommateurs : Les règles relatives à l'annulation s'appliquent. Tous les prix sont indiqués en € y compris la TVA, de l'entrepôt de Sande/Allemagne, et en plus des frais d'envoi pour l'ensemble du panier de produits. Nos conditions générales de vente s'appliquent de manière exclusive (sur www.reichelt.com/agb). Vente intermédiaire réservée. Tous les noms de produits et les logos sont la propriété des fabricants respectifs. Il en est de même pour les illustrations. Fautes d'impression, erreurs et modifications de prix réservées. reichelt elektronik GmbH & Co. KG, Elektronikring 1, 26452 Sande/Allemagne (HRA 200654 Oldenburg)

ELEKTOR-MAKER-SPACE



Made in Munich

Come MAKE it
@electronica 2014

Wisse Hettinga (Elektor Labs)

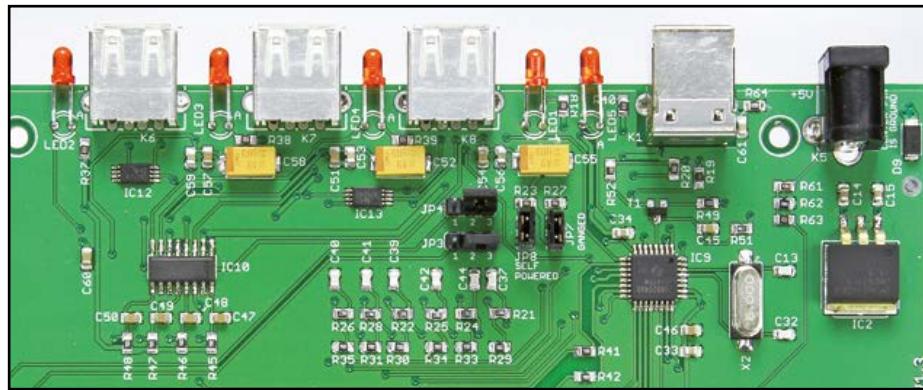
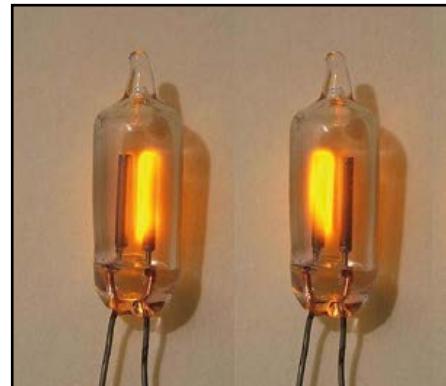
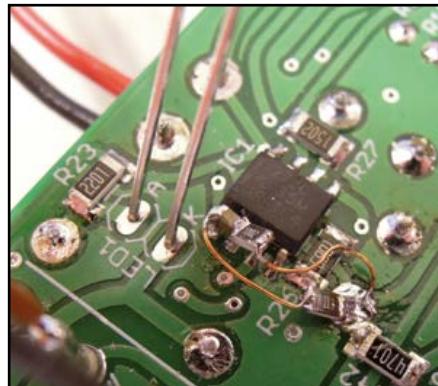
Vous aussi vous avez forcément eu vent de la révolution des *makers*. Ce mot correspond à une nouvelle pratique de l'électronique, plus collective, plus ludique, mais en quoi cette évolution concerne-t-elle la grand-messe annuelle de l'électronique ? Précisément là où habituellement efficient plutôt des cols blancs pour qui le fer à souder est un outil moyenâgeux, Elektor vous propose de participer à la création d'un espace nouveau... enfin, pas si nouveau que ça : l'idée, c'est tout simplement de **faire de l'électronique** comme on l'a toujours faite, mais dans un endroit où cela ne se pratiquait plus depuis belle lurette.

Une grande partie de l'équipe d'Elektor vous accueillera pour donner forme à ce projet **AVEC VOUS**. Peu importe votre degré de maîtrise des langues, allemand, anglais, néerlandais, espagnol ou C++, on se débrouillera. Profitez des ressources mises à votre disposition : tables, outils, appareils de mesure, imprimante 3D et WiFi à l'oeil. Pour le café, nous acceptons les dons. Et si vous avez oublié le chargeur de votre téléphone, raison de plus de passer au stand d'Elektor, vous en trouverez un chez nous.

Elektor Maker Space | hall A6 | stand 380

Suivez les activités d'Elektor !

Demandez notre lettre d'information gratuite, tout en bas à droite de notre page d'accueil www.elektor.fr



● communauté

- 3 Elektor International Makers *Made in Munich*
- 6 de nous à vous
- 8 les gagnants du concours *WIZnet Connect the Magic :*
15 000 \$ de prix pour concevoir un système novateur avec le module Ethernet WIZ550io
- 75 commande par gestes 3D pour vos µC et vos PC
avec kit d'évaluation Microchip/
Elektor et plaque 3D touchpad

● e-labs

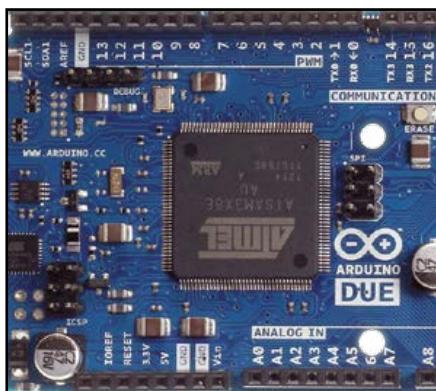
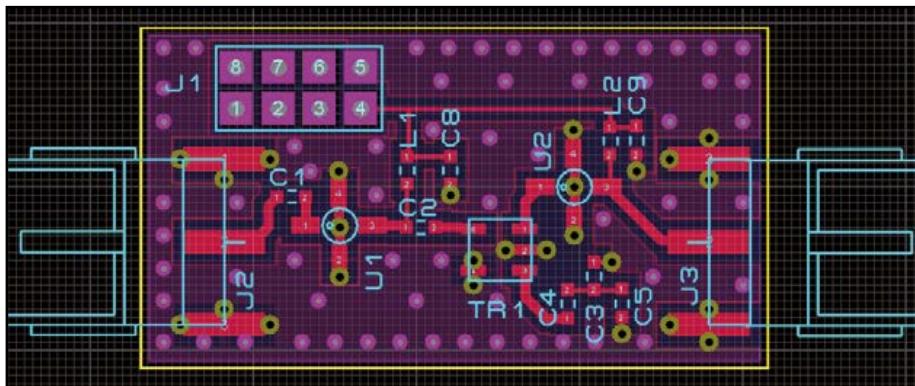
- 31 MMS : menhirs montés en surface
La manipulation des CMS
ressemble à celle des mégalithes :

les uns et les autres s'effondrent
tout le temps

- 35 trouvez un nouveau boulot
en postant sur Elektor.Labs
Les spaghetti peuvent mener loin
dans la vie.
- 61 à quoi sert l'impression 3D ?
Clemens Valens revient sur ses
doutes au sujet de l'impression 3D.
- 72 shield polyvalent pour Arduino
Une plaque d'essais à contacts à
ressorts, c'est bien... pour essayer,
mais que dirais-tu d'un shield
Arduino conçu justement pour
l'expérimentation ?
- 82 touche pas à mon câble !
Les mésaventures du banc d'essai
d'un programmeur avec la fiche
pour le câble micro-USB

● projets

- 15 balise de charge par USB
pour accu Li-Ion
Un régulateur shunt TL431 permet
de limiter la charge d'un accu au
lithium à 4,2 V, sa tension limite.
- 16 Visual Basic sur Raspberry Pi
Si vous êtes familier de Visual
Basic, vous éprouverez sans doute
quelques difficultés à transposer
dans Python vos jolies interfaces
graphiques.
- 18 alimentation pour PIC à arrêt
contrôlé
Un µC – peu importe lequel – doit
se mettre dans une configuration
donnée avant de s'éteindre.
- 21 lampes au néon
drôle de composant n° 10
Cet article a été inspiré par une
vieille alimentation Tektronix



22 DesignSpark PCB

[15^e jour : renumérotation de composants](#)

Apprenons à renuméroter des composants.

24 Arduino avec Atmel Studio

[débogage d'applications Arduino Due](#)

Voici comment installer les pattes Arduino dans le moelleux du Studio Atmel.

32 mise à niveau de

[la carte SD pour le RPi B+](#)

Vous venez de déballer votre Raspberry Pi B + et souhaitez utiliser la carte SD avec le logiciel de la version B dont vous êtes familier. Il va falloir ruser.

36 convertisseur USB ⇄ UART RS232/422/485

Sans microcontrôleur ni aucun logiciel

Ce convertisseur USB universel offre deux ports RS-232 et 2 ports RS-422/RS-485 utilisables simultanément sur les PC modernes sans port série DB9.

48 détecteur universel

Comment transformer un détecteur de fumée en détecteur polyvalent.

50 cube à LED RGB

animé avec Flowcode

[projet Instructables enlluminé et amélioré](#)

62 l'atelier du microcontrôleur (7)

[Le bus I²C](#)

Développé à l'origine par Philips pour connecter de façon simple à un microprocesseur les circuits intégrés contenus dans ses produits grand public, le bus I²C sert à transmettre des données sur deux fils au moyen d'une ligne de données et d'une ligne d'horloge.

● hors circuit

56 circuit imprimés : éviter les

[bourdes en HF](#)

[et avec les fréquences élevées en général](#)

● industrie

10 palpeurs de courant intégrés

[mesure d'intensité avec shunt de quelques mΩ](#)

La résistance d'un shunt doit être faible et la chute de tension mesurable de préférence par rapport à la masse quand c'est un µC qui mesure l'intensité. Il existe pour cela d'excellents petits circuits intégrés palpeurs de courant.

12 condensateurs de traversée CMS

[-60 dB d'interférences avec un CMS](#)

Même l'électronicien blasé peut avoir un coup de cœur en retrouvant des condensateurs de traversée qui ont connu l'ère de Bob Pease et des pionniers de la TSF. Ils débarquent sous forme de CMS, avec des spécifications brillantes en termes de suppression du bruit.

● banc d'essai

70 kit de développement IdO WiSmart

Beaucoup des produits amenés par la vague de l'internet des objets sont des boîtes noires. Il nous faut autre chose, quelque chose d'ouvert.

● magazine

69 Futurologie

avant-première des prochains numéros d'Elektor

76 générateur audio

Hewlett-Packard 200AB

[digne descendant du 200A](#)

L'histoire de Hewlett-Packard a commencé en 1939, quand William Hewlett a utilisé une ampoule à filament de tungstène dans un circuit oscillateur à pont de Wien à tubes.

81 hexadoku

C'est l'heure de la pause. Un peu de gymnastique arithmétique ?

37^e année, n° 437
novembre 2014

ISSN 0181-7450
Dépôt légal : octobre 2014
CPPAP 1113 U 83713
Directeur de la publication : Donatus Akkermans

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

Tél. : (+33) 01.49.19.26.19
du lundi au jeudi de 8h30 à 17h
le vendredi de 8h30 à 12h30
Fax : (+33) 01.49.19.22.37
www.elektor.fr | www.elektor-magazine.fr

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

DROITS D'AUTEUR :

© 2014 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.

Imprimé aux Pays-Bas
par Seneffeler Misset – Doetinchem

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

Elektor, c'est (aussi) vous !

L'essence des contributions des auteurs francophones au contenu du magazine Elektor et du site www.elektor-labs.com se distingue-t-elle de celle des autres, exprimées dans d'autres langues ? Fait-on de l'électronique différemment selon que l'on est papou ou eskimo, écossais, autrichien, norvégien, breton ou alsacien ? La langue dans laquelle on appréhende la réalité influence-t-elle la manière dont on conçoit l'électronique ?

Les avis sont partagés.

Les tenants de l'analyse simplificatrice arguent de la prépondérance de l'*anglais-que-tout-le monde-comprend* ainsi que de l'universalité des lois de la physique qui ne connaissent ni patois ni frontières. Selon eux, le courant circule partout dans le même sens ; les propriétés des composants ne changent pas avec les mots qui les désignent et le microprocesseur se fiche bien de la langue dans laquelle pense celui qui le programme. Il n'y aurait donc pas lieu de chercher dans l'électronique des nuances liées à la langue et à la culture de ceux qui la pratiquent, la conçoivent et la décrivent. Point final.

Si j'en parle ici, c'est parce que je ne partage pas cet avis. Le contraire me paraît évident à la lecture des articles publiés au fil des mois par Elektor, mais je me garde de la discussion, faute d'argument massue pour étayer ma conviction. En revanche, je souhaite intensifier la diversification des sources d'Elektor en augmentant la part des auteurs francophones. Je compte administrer ainsi la preuve *de facto* de l'existence d'une spécificité liée à la langue et à la culture.

Pour réussir, j'ai besoin de vous et de votre participation active.

Vous jouissez d'une solide expérience de l'électronique ou d'un domaine connexe : audio, domotique, robotique, micro-informatique, énergies renouvelables, modélisme... peu importe. Vous avez envie de partager votre savoir et le fruit de votre créativité : le moment est venu de publier dans et avec Elektor ! Nos productions s'adressent aussi bien aux amateurs qui souhaitent se perfectionner qu'aux professionnels qui aiment ce qu'ils font. Ce que tous nos lecteurs partagent, c'est la passion ; ce que toutes nos publications ont en commun, c'est encore la passion. Un jour vous avez été un électronicien débutant ; grâce à l'étincelle que des maîtres ont su faire jaillir, vous pouvez, aujourd'hui, devenir un auteur débutant. Grâce à Elektor. Notre équipe se met à votre service pour développer votre passion de communiquer. Vous en tirerez non seulement d'immenses satisfactions, mais aussi des revenus substantiels. Chaque année, Elektor paie à ses auteurs des centaines de milliers d'euros de droits. Pourquoi pas à vous ?

Denis Meyer



Notre équipe

Rédacteur en chef :	Denis Meyer (redaction@elektor.fr)
Rédaction internationale :	Harry Baggen, Jan Buiting, Jaime González-Arintero, Jens Nickel
Laboratoire :	Thijs Beckers, Ton Giesberts, Wisse Hettinga, Luc Lemmens, Mart Schroijsen, Clemens Valens, Jan Visser, Patrick Wielders
Coordination :	Hedwig Hennekens
Ont coopéré à ce numéro :	Rémi Descistes, Robert Grignard, Hervé Moreau, Kévin Petit, Guy Raedersdorf, NN
Service de la clientèle :	Cindy Tyssen & Vanessa Noville
Graphiste :	Giel Dols
Elektor en ligne :	Daniëlle Mertens

	France Denis Meyer +31 46 4389435 d.meyer@elektor.fr
	United Kingdom Carlo van Nistelrooy +44 20 7692 8344 c.vannistelrooy@elektor.com
	USA Carlo van Nistelrooy +1 860-289-0800 c.vannistelrooy@elektor.com
	Germany Ferdinand te Walvaart +49 241 88 909-17 f.tewalvaart@elektor.de
	Netherlands Ferdinand te Walvaart +31 46 43 89 444 f.tewalvaart@elektor.nl
	Spain Jaime González-Arintero +34 6 16 99 74 86 j.glez.arintero@elektor.es
	Italy Maurizio del Corso +39 2.66504755 m.delcorso@inware.it
	Sweden Carlo van Nistelrooy +31 46 43 89 418 c.vannistelrooy@elektor.com
	Brazil João Martins +31 46 4389444 j.martins@elektor.com
	Portugal João Martins +31 46 4389444 j.martins@elektor.com
	India Sunil D. Malekar +91 9833168815 ts@elektor.in
	Russia Natalya Melnikova +7 (965) 395 33 36 Elektor.Russia@gmail.com
	Turkey Zeynep Köksal +90 532 277 48 26 zkoksal@beti.com.tr
	South Africa Johan Dijk +31 6 1589 4245 j.dijk@elektor.com
	China Cees Baay +86 21 6445 2811 CeesBaay@gmail.com

Notre réseau



VOICE COIL



audioxpress



vous connecte à



Nos annonceurs



CadSoft

www.cadsoftusa.com 29



pico

www.picotech.com/PS366 83



Eurocircuits

www.elektorpbservice.com 47



Reichelt

www.reichelt.fr 2



Microchip

www.microchip.com/gestic 84

Pour placer votre annonce dans le prochain numéro d'Elektor
veuillez contacter Mme Ilham Mohammedi par téléphone au (+31) 6.41.42.25.25
ou par courrier électronique : i.mohammedi@elektor.fr

Vos correspondants

Nous sommes à votre service pour toute question relative à votre commande ou votre abonnement
par téléphone au (+33) 01.49.19.26.19 du lundi au jeudi de 8h30 à 17h
et le vendredi de 8h30 à 12h30 ou par courriel : service@elektor.fr



concours WIZnet Connect the Magic: les gagnants

Avec à la clé 15 000 \$ de prix, le concours WIZnet « connect the magic » vous mettait au défi de concevoir un système novateur à accès Internet exploitant le module Ethernet WIZ550io de WIZnet. Le jury a commencé à noter les entrées début août, peu après la fin du concours. Les projets ont été jugés d'après leur mérite technique, leur originalité, leur utilité, l'optimisation de leur conception et leur rapport coût-bénéfice. Voici le podium et les mentions honorables !

Les projets complets (documentation, schémas, photo, code, etc.) sont en ligne sur <http://circuitcellar.com/wiznet2014/>. Merci aux participants et félicitations aux vainqueurs !

Premier prix

Chimaera : le théréminal poly-magnéto-phonique

Hans Peter Portner (Suisse)

Chimère analogique et numérique du théréminal, le *Chimaera* est un instrument polyphonique prêt pour le réseau et à matériel libre (*open source*). Un réseau de capteurs à effet Hall linéaires et analogiques engendre un espace continu d'interactions à deux dimensions. Les capteurs sont excités par les aimants au néodyme que l'instrumentiste porte aux doigts. Le dispositif suit et interpole en permanence la position des aimants le long du réseau de capteurs



et produit les signaux à faible latence des évènements associés. Ces signaux sont codés au format *Open Sound Control* et transmis via UDP/TCP à un synthétiseur logiciel. Basé sur un ARM Cortex M4 et une puce W5500 de WIZnet prenant en charge les protocoles réseau de bas niveau, l'unité de traitement numérique du signal est chargée de lire les capteurs, de traiter les évènements détectés et de communiquer avec l'hôte.

Deuxième prix

Serveur LCDTV : diffusion en flux avec adaptateur Ethernet/USB

Lindsay Meek (Australie)

Le serveur LCDTV exploite le WIZ550io pour permettre à une TV LCD équipée d'un port USB de diffuser des contenus en *streaming* à travers un LAN. Un adaptateur convertit les requêtes par USB du périphérique de stockage en requêtes pour les connecteurs LAN à l'aide d'un système de fichiers virtuel. Combiné à un adaptateur CPL/Ethernet, il permet de regarder des vidéos numériques sur une TV distante plus ancienne.



Troisième prix

Réseau de surveillance WIZ

Claudiu Chiculita (Roumanie)

WIZ est un système de surveillance composé de nœuds capables de collecter et traiter indépendamment des informations, de déclencher des alarmes et de communiquer entre eux de façon à considérer les menaces sous plusieurs angles. Chaque nœud est doté d'une puce réseau W5500 de WIZnet, d'un détecteur IR, d'un servomoteur, d'une interface PoE, d'une caméra avec traitement d'images, d'un dispositif de stockage et d'un logiciel PC de surveillance et de configuration.



Mention honorable**Sentry****David Penrose (USA)**

Sentry est un réseau de détecteurs à infrarouge passif à placer en différents endroits d'une maison de retraite pour en détecter les mouvements.



Le système utilise aussi des détecteurs de mouvement fixés à des chaises et des lits pour déterminer si une personne y est présente. Grâce à la puce WIZ5500io, les systèmes sont reliés par radiofréquence à un processeur qui surveille en toute discrétion l'activité des résidents et décide si la situation est normale ou non. Si un comportement est considéré comme anormal, une alerte est envoyée à la famille ou au personnel soignant. La personne âgée est rassurée car elle sait qu'une aide sera présente en cas de besoin, avec l'assurance que son intimité est préservée.

Mention honorable**Mangeoire automatique****Dean Boman (USA)**

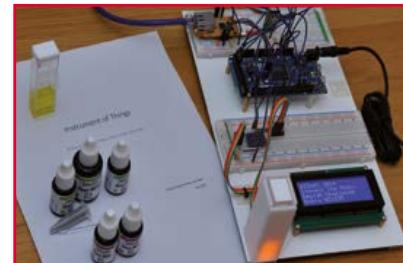
La mangeoire est conçue pour délivrer de façon automatique du foin à des petits animaux de ferme comme des chèvres ou des moutons. La connexion avec l'interface réseau du système de commande est assurée par le module réseau WIZ5500io de WIZnet. Ce projet permet de contrôler et surveiller à distance des mangeoires, ce qui repousse les frontières de l'Internet des Objets jusqu'aux basses-cours !

**Mention honorable****WIZpix****Contrôleur de pixels connecté****Robert Gasiorowski (USA)**

Le contrôleur de pixels WIZpix utilise une puce W5500 de WIZnet pour se connecter au réseau, ainsi qu'un microcontrôleur pour l'interface avec le W5500 et le pilotage intelligent des pixels. Son objectif premier est de créer des lumières de Noël élégantes, mais il peut servir en toute occasion où une animation lumineuse est de la partie. Il élimine le besoin de recourir à un système DMX complexe et coûteux. Un seul câble suffit grâce à son interface PoE intégrée.

**Mention honorable****The Instrument of Things****Radko Bankras (Pays-Bas)**

Le projet *Instrument of Things* (IoT) fournit aux instruments électriques des capacités de télécommande aux normes industrielles via une interface TCP/IP. Le module WIZ5500io de WIZnet a été utilisé pour créer un serveur web basique, un service de redirection de port, ainsi qu'un serveur pour la télécommande de l'instrument via le protocole de communication VXI-11. L'objectif final de l'IoT est de permettre d'ajouter à n'importe quel instrument électrique le protocole VXI-11 et la technologie LXI (LAN eXtensions for Instruments).

**Mention honorable****Contrôleur de radiotélescope****Clayton Gumbrell (Australie)**

Ce circuit commande un radiotélescope pour observer l'univers dans la raie d'émission de 21 cm de l'hydrogène (fréquence de 1420 MHz). Le télescope possède une antenne parabolique de 1,7 m fixée sur une monture azimutale motorisée et orientable dans n'importe quelle direction au-dessus de l'horizon. Un module Ethernet WIZ5500io fournit la connexion avec le contrôleur de l'antenne. Le télescope peut ainsi être placé loin de toute pollution lumineuse et commandé à l'aide d'ordinateurs ou d'opérateurs distants reliés par l'internet.



(140309 – version française : Hervé Moreau)

Devenez auteur chez Elektor !

Vous avez une grande expérience de l'électronique ou d'un domaine connexe (audio, robotique, micro-informatique, domotique...) et vous ne demandez qu'à partager votre savoir :

Elektor vous aidera à en faire un livre et le publiera avec vous !

Un jour vous avez été un électronicien débutant, bientôt vous serez un auteur débutant.

palpeurs de courant intégrés

mesure d'intensité avec shunt de quelques mΩ

Viacheslav Gromov
(Allemagne)

La résistance des shunts doit être faible et la chute de tension mesurable de préférence par rapport à la masse quand c'est un µC qui mesure l'intensité. Il existe pour cela de petits circuits intégrés palpeurs de courant qui font ça très bien.

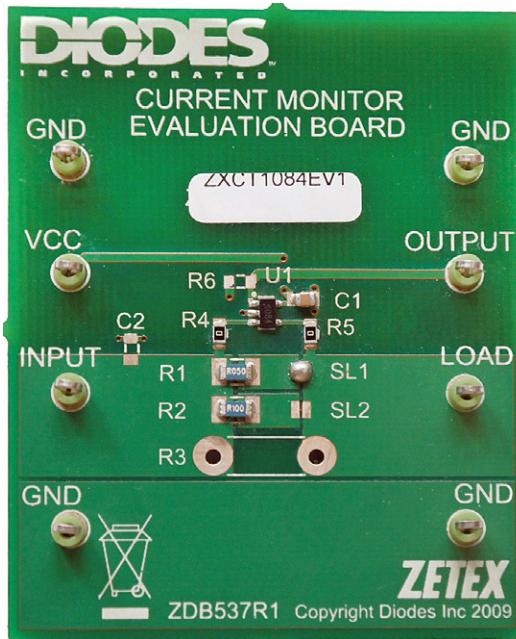


Figure 1.
Carte d'évaluation typique.
Sur le modèle ZXCT1084EV1 se trouve le ZXCT1084.
La notice fournie est très complète et donne des exemples d'application.

Le mesure de courant avec un shunt est classique et simple. Qui dit faible résistance dit faible chute de tension et donc faible dissipation. Très bien, mais il faut du coup amplifier la chute de tension pour l'adapter à la plage d'entrée du convertisseur A/N d'un microcontrôleur (µC). Souvent, l'insertion du shunt dans le circuit de masse de l'alimentation n'est pas sans inconvénient, car le µC attend un potentiel mesurable soit directement par rapport à la masse, soit, avec des entrées différentielles, au moins dans les limites du mode commun de l'étage d'entrée analogique. On pourrait régler ça avec quelques amplificateurs opérationnels astucieusement configurés,

mais je propose des circuits intégrés spécialement conçus pour ça. Ils amplifient la chute de tension aux bornes du shunt, et délivrent un potentiel mesurable par rapport à la masse de sorte que le procédé ne pose aucun problème même en présence d'une tension d'alimentation élevée.

La consommation de ces composants analogiques de la série Zetex, fabriqués par Diodes au Texas, est remarquablement faible. Il en existe différentes variantes. Dans leur boîtier SOT23 à 5 broches, ils sont peu encombrants et se contentent de quelques composants périphériques. Il existe des cartes d'évaluation pour certaines références (**fig. 1**). Leur champ d'application est vaste grâce notamment à l'étendue de leur plage de tensions. Ils s'utilisent aussi bien dans de simples circuits de surveillance que dans de lourdes installations de gestion de batteries ou dans l'électronique automobile.

Vie de famille

Les membres de la famille Zetex se distinguent d'abord par leur gain en tension : le ZXCT1084 et le ZXCT1085 ont un gain fixe de 25 ; pour les variantes ZXCT1086 et ZXCT1087, il est de 50. La tension délivrée par la broche *OUT* est applicable telle quelle à l'entrée du convertisseur A/N d'un µC. La résistance assez élevée (125 kΩ) de cette sortie ne devrait pas poser de problème à la plupart des convertisseurs dont la résistance d'entrée est en général très forte aussi. Le gain fixe permet de se passer du diviseur de tension avec lequel on fixe autrement le facteur d'amplification. Si vous avez besoin d'un gain variable, utilisez le ZXCT1082 ou le ZXCT1083 ; il leur

faut juste trois résistances de plus pour fixer leur gain à votre guise.

La faible intensité ($0,6 \mu\text{A}$) du courant consommé est idéal pour des circuits alimentés par piles. Lors de la mesure, le courant à travers le shunt ne dépasse pas $1,7 \mu\text{A}$ (l'intensité exacte dépend du courant à mesurer). Le temps de réaction typique de ces palpeurs est de l'ordre de 5 à 7 μs , ce qui leur permet de suivre les variations les plus rapides de l'intensité du courant à mesurer (fig. 2). La plage de tension d'alimentation couvre 2,7 à 60 V pour les ZXCT1082/84/86, et 2,7 à 40 V pour les ZXCT1083/85/87.

Au boulot

Les figures 3 et 4 montrent comment brancher les modèles à gain fixe ainsi que les autres, à peine moins simples, avec leurs trois résistances R_{GT} et R_G . Le gain des ZXCT1082/83 est dans l'équation suivante :

$$U_{OUT} = U_{SENSE} * R_G / R_{GT}$$

Ces deux modèles sont d'ailleurs dotés d'une sortie de courant que les autres n'ont pas. Grâce à leur gain, la résistance des shunts pourra être faible, ce qui réduit la dissipation : un avantage quand on doit mesurer du courant de forte intensité, d'autant plus appréciable quand les shunts sont des CMS. Prenons le convertisseur A/N d'un μC dont la plage d'entrée serait de 1 V : le gain de 50 pour une plage de mesure de 1 A nous donne un shunt de $20 \text{ m}\Omega$ au lieu de 1Ω . Avec l'intensité maximale de 1 A, la dissipation de notre shunt actif ne sera que de 20 mW, contre 1 W avec un shunt passif. La réduction de la dissipation est directement proportionnelle au gain.

Je vous recommande de découvrir les autres possibilités de ces palpeurs intégrés [1]. Ces feuilles de caractéristiques sont riches. Les chronogrammes des signaux et les courbes de réponse en fréquence des composants méritent votre attention.

(140179 – version française : Rémi Descistes)

Lien

[1] http://diodes.com/catalog/voltage_output_80

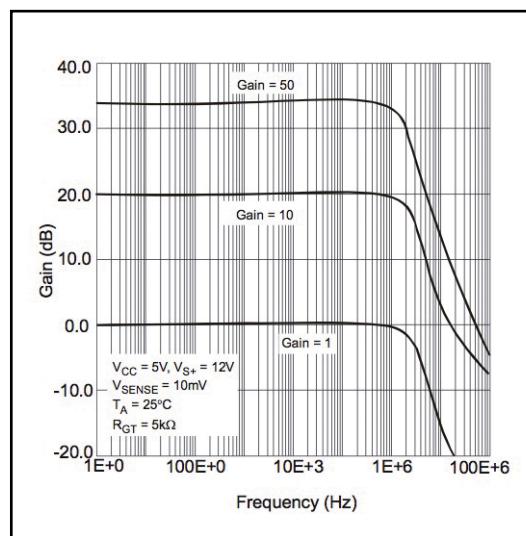


Figure 2.
Réponse en fréquence selon le gain.

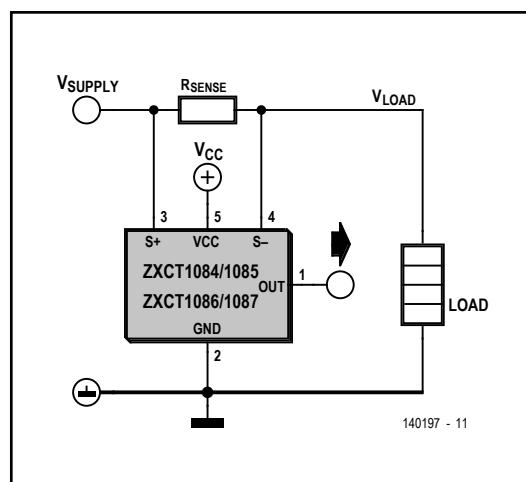


Figure 3.
Le ZXCT1084/85/86/87 fait le travail tout seul.

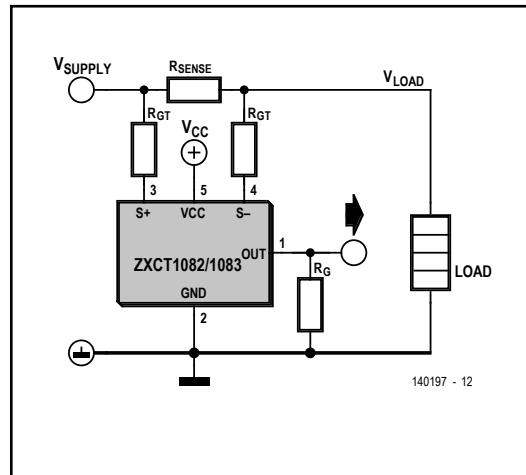


Figure 4.
Pour fixer le gain, le ZXCT1082/83 n'a besoin que de trois résistances.

condensateurs de traversée CMS -60 dB d'interférences avec un CMS



Même l'électronicien blasé peut avoir un coup de cœur en retrouvant des condensateurs de traversée qui ont connu l'ère de Bob Pease et des pionniers de la TSF. Ils débarquent sous forme de CMS, avec des spécifications brillantes en termes de suppression du bruit.

Kirk Sceviour
(Canada)

Cette découverte, je l'ai faite récemment tandis que je mettais à niveau de vieux projets de communication. Le nouveau modèle impliquait d'autres composants et une alimentation symétrique, le tout à ajuster aux dimensions de la carte

d'origine. On arrive vite à court de place sur une petite carte, même quand on n'y installe que des CMS. Mon attention s'est tout de suite portée sur un gros encombrant : le volumineux filtre LC du convertisseur CC/CC. Peut-on le réduire, voire le remplacer ? La combinaison traditionnelle de résistances, condensateurs et inductances est-elle l'unique solution pour filtrer des signaux avec des composants discrets ? Mes recherches m'ont confronté à un composant remarquable, tout neuf pour moi : le condensateur de traversée (*feed-through*), qu'on appellera *cdt* pour faire court. Sans surprise, l'objet se décline en une foule de boîtiers et Google m'a révélé une variété d'options, depuis le gros ustensile pour montage sur bâti jusqu'au CMS 0402.

J'ai donc appris qu'un cdt laisse passer le courant continu aussi bien qu'alternatif tout en court-circuitant à la masse les interférences électromagnétiques (*EMI*), du moins à la fréquence de résonance. On peut le voir comme un condensateur de découplage. On l'utilise couramment en conjonction de fort courant et haute fréquence, quand un signal traverse une enceinte blindée. Certains filtres, dits tubulaires, peuvent en utiliser un ou plusieurs en combinaison avec des inductances pour atteindre la réponse souhaitée. Les points de mise à la masse des composants internes sont ramenés à une borne, au boîtier ou à l'appareil lui-même pour forcer toutes les nuisances EMI qui transitent sur les données, le signal audio ou les lignes d'alimentation vers la

Figure 1.
Perte d'insertion pour différentes valeurs de cdt
(source : Murata).

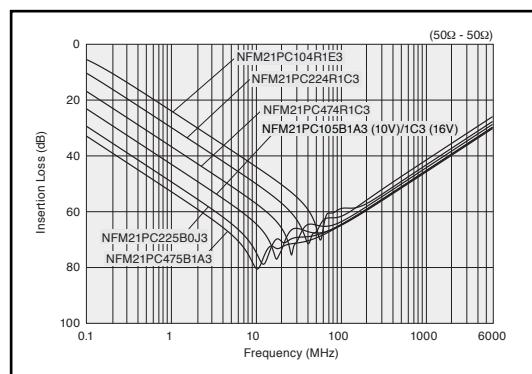
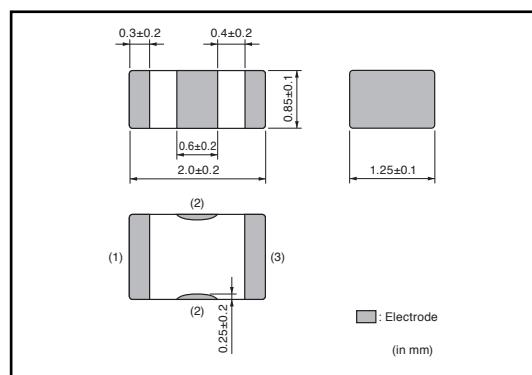


Figure 2.
Dimensions de cdt CMS typiques (source : Murata).



masse ou le châssis de l'appareil par un chemin à basse réactance. On rencontre moins souvent les cdt hors du champ des gros bâtis massifs, mais j'ai flashé sur la variété en technologie CMS, disponible en céramique multicouche X7R ou NP0. Expérimenté, on peut les souder selon les techniques normales. Indépendamment de leur habillage, les cdt sont des filtres de bruit condensés, avec des caractéristiques de choc.

Petite taille, grands effets

Un exemple : les caractéristiques données par Murata pour les cdt de 0,1 μF 0805 NFM21PC indiquent une perte d'insertion de plus de 60 dB, centrée sur 55 MHz (**fig. 1**). Quel paquet de parasites filtrés par un si petit boîtier (**fig. 2**) ! Ces composants se trouvent facilement, dans une gamme de capacités raisonnable et ne sont pas des poids plume question performances. Parmi ces CMS, il y en a qui tiennent des centaines de volts et 5 A ou plus. Ils excelltent vraiment aux hautes fréquences. La **figure 3** montre le graphe présenté sur le site de Murata [1] qui compare les caractéristiques de perte d'insertion de condensateurs céramiques et de filtres de traversée à trois contacts, entre les exécutions traditionnelles à fils et les puces.

Et pour vos applications ?

La vraie question, c'est de savoir quelle différence cela fera si vous utilisez un cdt plutôt que n'importe quel condensateur ou réseau de filtre conventionnel. La distinction tient à la manière de les mettre en œuvre dans le circuit, à leur structure interne et à leurs caractéristiques électriques. En plus des habituelles électrodes et couches de céramique, les cdt étalement d'autres couches imprimées développées à la perpendiculaire. Ce sont ces couches supplémentaires qui en font des composants à 3 contacts.

À la **figure 4**, on voit une application typique de condensateur de traversée (a) comparée au même circuit équipé d'un condensateur conventionnel (b). Les deux circuits sont des filtres. Le diagramme montre bien la différence de câblage. Un condensateur CMS normal n'a que deux contacts, aucun n'est destiné à la masse. La traversée capacitive dispose de 3 contacts : 2 pour le signal et 1 pour la masse.

Toute la différence, comme d'habitude, du moins pour les briscards de la radio, réside dans ce que « voit » le signal quand il arrive, ou quand il sort du circuit. Le cdt a une inductance moins grande

vers la masse, comparé au condensateur conventionnel. L'écart peut atteindre un ordre de grandeur. Les circuits électriques équivalents (*EEC*) sont reproduits aux figures 4a et 4b pour les deux types de filtre. Ils démontrent des performances électriques très différentes. Le cdt a une inductance parallèle minimale et une inductance série supérieure en comparaison avec un condensateur CMS normal. La plupart des fabricants indiquent une plus large courbe de réponse en fréquence.

L'interface parfaite entre données et HF

Alors, pourquoi ne trouvons-nous pas plus souvent ces composants dans les circuits modernes,

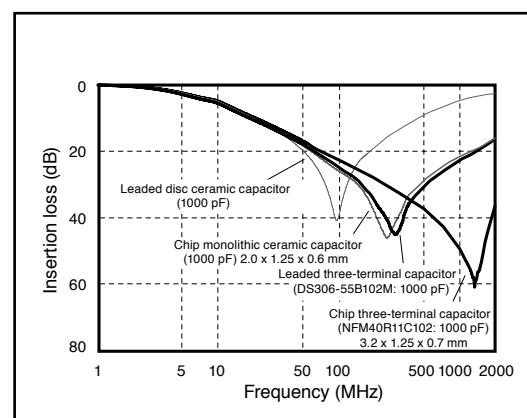


Figure 3.
Pertes d'insertion pour des puces de traversée comparées à des modèles de condensateurs ordinaires.
(source : Murata)

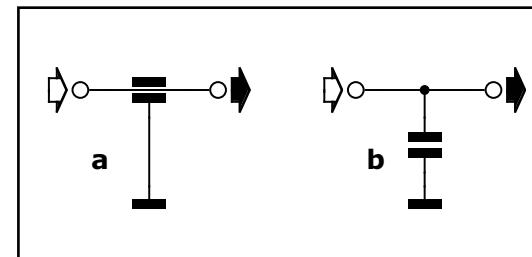


Figure 4.
Filtre de traversée (a) et filtre conventionnel (b).

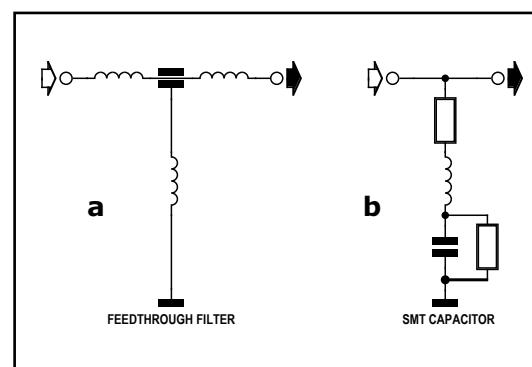
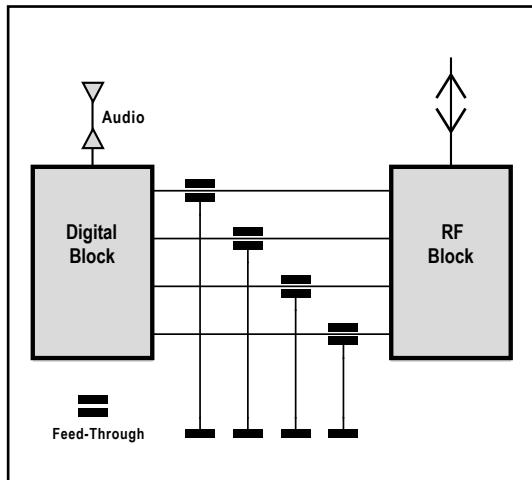


Figure 5.
Circuits électriques équivalents pour filtre de traversée (a) et filtre conventionnel (b).

●composants

Figure 6.
Jeunes concepteurs de matériel IoT et WiFi, adoptez les cdt sur toutes lignes de données et d'alimentation entre le numérique et la radiofréquence associée (source : AVX).



spécialement sur les horloges à radiofréquence et les circuits numériques à haute vitesse, de ceux qu'Elektor publie régulièrement ? Une petite exploration m'a révélé un large éventail d'applications potentielles, depuis le conditionnement de la tension d'alimentation jusqu'aux horloges, sur les lignes de données et comme filtre sur les amplis de puissance.

Le condensateur de traversée, appelé aussi traversée capacitive, semble particulièrement approprié au développement rapide de l'internet des objets (*IoT*) et à la prolifération des appareils sans fil (**fig. 6**). L'exemple de choix réside dans l'interface entre données et radiofréquence, essentielle à tout système autonome. Quand on permet au bruit d'envahir la liaison entre la base numérique et l'émetteur récepteur radio, il corrompt les données transmises jusqu'à les rendre inutilisables. Des cdt convenablement choisis sur les lignes de données offrent une protection antiparasite efficace, spécialement en environnement bruyant. Les avantages obtenus par l'adjonction de ce filtrage EMI valent bien la dépense et la petite place qu'il occupe sur le circuit imprimé.

Toute ligne de sortie numérique qui exerce une commande de précision sur un autre appareil est susceptible de produire des interférences, c'est là l'occasion d'un filtrage de traversée. Une situation évidente concerne les systèmes de radiocommande. Un tel appareil peut être victime de son propre émetteur s'il travaille très près du circuit imprimé. Un filtre de traversée centré sur ou au voisinage de la fréquence d'émission, installé sur la broche d'entrée du microcontrôleur serait très efficace pour en éliminer les interférences.

Il serait intéressant de voir son influence sur le circuit de Jan Lichtenbelt : Commande de vitesse de moteur CC pour modèle RC, publié par Elektor [2], qui rencontrait des difficultés avec une composante à 40 MHz.

Je n'ai pas encore pu me procurer assez d'échantillons de cdt pour les essayer. Mais je crains que la fréquence de commutation à 2,7 MHz de mon convertisseur soit trop basse pour profiter à fond de ce type de filtre. Ce qui ne doit pas me dispenser de les utiliser sur des alimentations à basse fréquence. S'il est possible d'avoir un filtrage acceptable pour elles, on pourrait réduire le nombre de composants et libérer de l'espace sur les cartes.

Pour beaucoup d'entre vous, l'existence des cdt peut sembler du réchauffé, mais ce fut pour moi une découverte digne d'intérêt. Je compte bien ne pas en rester là et les essayer dans mon prochain projet... sans savoir où cela peut m'entraîner.

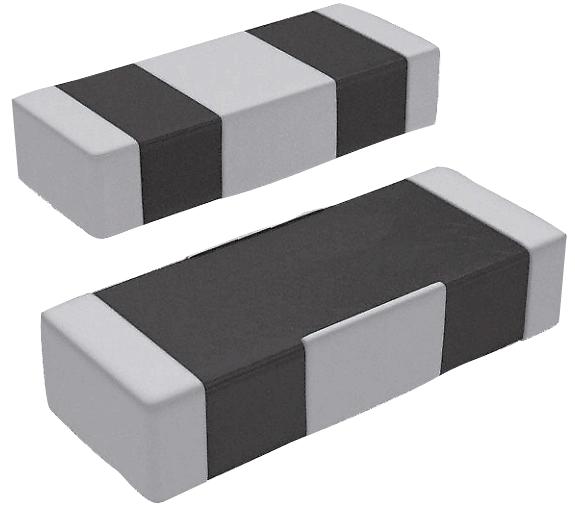
(140147 – version française : Robert Grignard)

Liens

[1] Murata:

www.murata.com/~media/webrenewal/products/emc/emifil/knowhow/20to22.ashx

[2] www.elektor-magazine.fr/130259



balise de charge par USB pour accu Li-Ion

Un régulateur *shunt* TL431 permet de limiter la charge d'un accu au lithium à 4,2 V, sa tension limite. Celle-ci une fois atteinte, le TL431 va dériver tout le courant issu de la résistance en série. Une valeur d'environ 10 Ω évite toute surcharge du TL431.

Mais l'idée consiste à adjoindre ici un transistor de sortie PNP au TL431, comme dans une paire Sziklai. Ainsi, le courant de dérivation maximal correspond aux 100 mA que le TL431 peut supporter, multipliés par le gain du transistor PNP. Plus de souci, dès lors avec le courant nominal maximal du TL431, la dernière contrainte réside dans le courant le plus élevé (500 mA) que le port USB est capable de fournir. Lorsque la tension de l'accumulateur est au plus bas, la résistance de limitation de courant doit valoir 4,5 Ω. La valeur existante la plus proche est 4,7 Ω.

Pour ce projet, le transistor choisi est un 2SB857 (T2), simplement parce que j'en avais un sous la main. Il n'y a rien de particulier à signaler à son propos, il est spécifié à 50 V, 4 A avec un $h_{fe(min)}$ de 60, des caractéristiques que l'on retrouvera avec n'importe quel PNP au format TO220 du même acabit. Sur le prototype, je n'ai mis aucun radiateur. Hors charge (courant de dérivation maximal), il chauffe modérément sans devenir gênant.

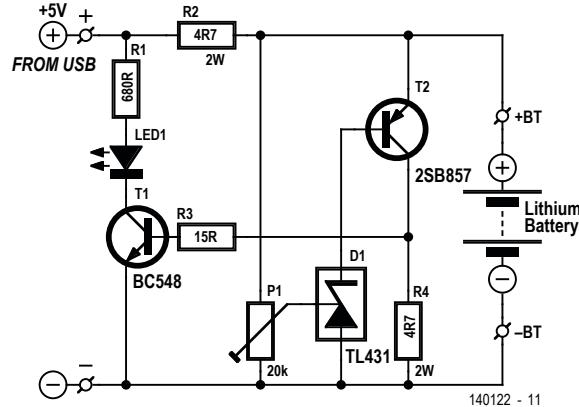
Comme c'est le transistor PNP qui supporte maintenant le plus gros du lissage et que son collecteur est raccordé au pôle négatif, on peut insérer une résistance de mesure du courant, R4, à la broche du collecteur afin de piloter un deuxième transistor, T1, qui va servir de témoin lumineux « batterie pleine » avec la LED1. Il aurait été plus compliqué, rien qu'avec le TL431, d'insérer une résistance de détection de courant en série avec l'anode, cela aurait entraîné une variation de la

Ian Field
(Royaume-Uni)



tension de décalage et donc contrarié son référentiel. La résistance de détection de courant dans la branche du collecteur s'intègre à la boucle de rétroaction et affecte très peu la régulation. Enfin, le potentiomètre P1 permet d'ajuster avec précision la tension de fin de charge.

(140122 – version française : Robert Grignard)



Visual Basic sur Raspberry Pi



Bert van Dam
(Pays-Bas)

Habituellement, le *Raspberry Pi* se programme en *Python*, un langage puissant et facile à utiliser. Si vous êtes familier de *Visual Basic*, vous éprouverez sans doute quelques difficultés à transposer dans *Python* vos jolies interfaces graphiques.

Il existe des solutions intermédiaires, comme l'utilisation d'un gabarit graphique décrite dans l'ouvrage de M. Jansee [1], mais il serait quand même préférable de pouvoir programmer en *Visual Basic* sur un PC, pour transposer le résultat ensuite sur *Raspberry Pi*. Cette solution combinerait la puissance de calcul du PC aux puissants outils de programmation de *Visual Basic* !

Sur le PC, *Visual Basic* fait appel à .NET, une structure logicielle qui réunit une vaste collection de bibliothèques que les programmeurs (et leurs applications) peuvent utiliser à leur guise. La compilation des programmes produit un code intermédiaire qui s'appuie sur .NET. On parle de *Common Language Interface*, une espèce d'espéranto qui permet d'utiliser sur un ordinateur d'un certain type des programmes écrits pour des ordinateurs d'un autre type, à condition qu'ils connaissent .NET. Proposée par *Microsoft*, la structure .NET concerne certes les PC *Windows*

au premier chef, mais il en existe des variantes pour Linux. La plus répandue s'appelle Mono.

Je propose dans cet article d'installer la variante Linux de .NET sur un *Raspberry Pi* pour lui confier ensuite l'exécution d'un programme écrit en *Visual Basic* sous *Windows 7* (64 bits) à titre de démonstration de la portabilité du code obtenu.

Que dois-je faire ?

Si vous n'avez pas *VB* sur votre ordinateur, téléchargez-en la version *Visual Basic Express 2010* [2]. Je vous déconseille la version plus récente (2013), car la version correspondante de .NET pour Linux n'est pas encore sortie.

Composez un programme de test avec *Visual Basic*. Si vous êtes à court d'idées, je vous propose [3] le code source d'un programme de test simple, avec un bouton que j'ai appelé `monotest`. Lancez ce programme, une fenêtre apparaîtra

avec ce bouton. Cliquez dessus et vous verrez le texte «Hello World» (**fig. 1**). Ceux qui n'ont pas (encore) *Visual Basic* sur leur bécane trouveront dans le téléchargement une version compilée de ce programme (*monotest.exe*).

Connectez votre *Raspberry Pi* à l'internet et installez le programme 'mono' à l'aide des commandes suivantes :

```
sudo apt-get update
sudo apt-get install mono-vbnc
```

Patiencez un peu puis répondez Y(es) à toutes les questions posées pendant l'installation.

Copiez le code exécutable (le fichier *monotest.exe*) du PC vers le *Raspberry Pi*. C'est très commode avec le programme WinSCP, comme le décrit M. Jansee dans son livre [1]. Ce programme figure aussi dans le téléchargement lié à cet article [3]. Il ne nécessite pas d'installation. Lancez le programme WinSCP.exe et saisissez comme Host name dans Session l'adresse IP de votre *Raspberry Pi*. Vérifiez que c'est le port 22 qui est choisi. Établissez la connexion. Si c'est la première fois que vous utilisez cette adresse IP avec WinSCP, vous verrez un avertissement qu'il suffit d'accepter.

Lancez ensuite Commander : vous verrez à gauche une liste de fichiers du PC à gauche et en face la liste de ceux du *Raspberry Pi*. Cherchez les liens exécutables et tirez-les vers le RPi.

Lancez le programme sur le *Raspberry Pi* à l'aide de la commande `mono monotest.exe`

Vous voici armé pour toutes sortes d'expériences intéressantes. Comme vous écrirez vos programmes sur le PC, il ne sera pas évident d'utiliser des ressources spécifiques au *Raspberry Pi*, comme p. ex. les broches GPIO. Il faut donc accepter les limites de cette astuce, qui n'a d'autre ambition que de faciliter la création de graphiques attrayants, p. ex. des jeux.

Cette méthode a été testée avec la carte SD associée au livre *Raspberry Pi* déjà cité, mais devrait fonctionner aussi avec des cartes standard *Raspbian Wheezy*.

(140263 – version française : Rémi Descistes)

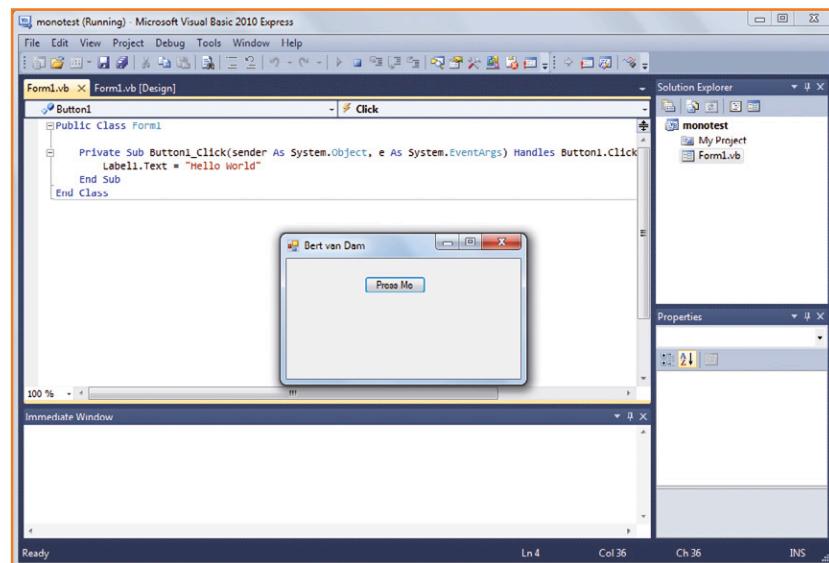


Figure 1. Le programme *Monotest*, dans *Visual Basic* sous *Windows* sur un PC



Figure 2. À gauche le programme sur un PC sous Windows, à droite sur un Raspberry Pi.

Liens

- [1] www.elektor.fr/rpi
- [2] www.visualstudio.com/fr-fr/downloads#d-2010-express
- [3] www.elektor-magazine.fr/140263



Je vous recommande l'excellent ouvrage de Maartje Jansee, publié par Publitronic-Eletor, intitulé **Raspberry Pi cool | 45 APPLICATIONS UTILES POUR L'ELECTRONICIEN**, et judicieusement sous-titré : **L'ALLIANCE DE LA PROGRAMMATION ET DE L'ELECTRONIQUE**

commande d'alimentation à arrêt contrôlé pour tout circuit à microcontrôleur

Jack Jouas (France)

La fonction de ce module de commande est d'obtenir qu'à la coupure d'un circuit à microcontrôleur – peu importe lequel – celui-ci se mette dans une configuration donnée avant de s'éteindre.

Figure 1.
Trois transistors et un régulateur de tension constituent l'essentiel de cette commande d'extinction d'alimentation d'un circuit à microcontrôleur.

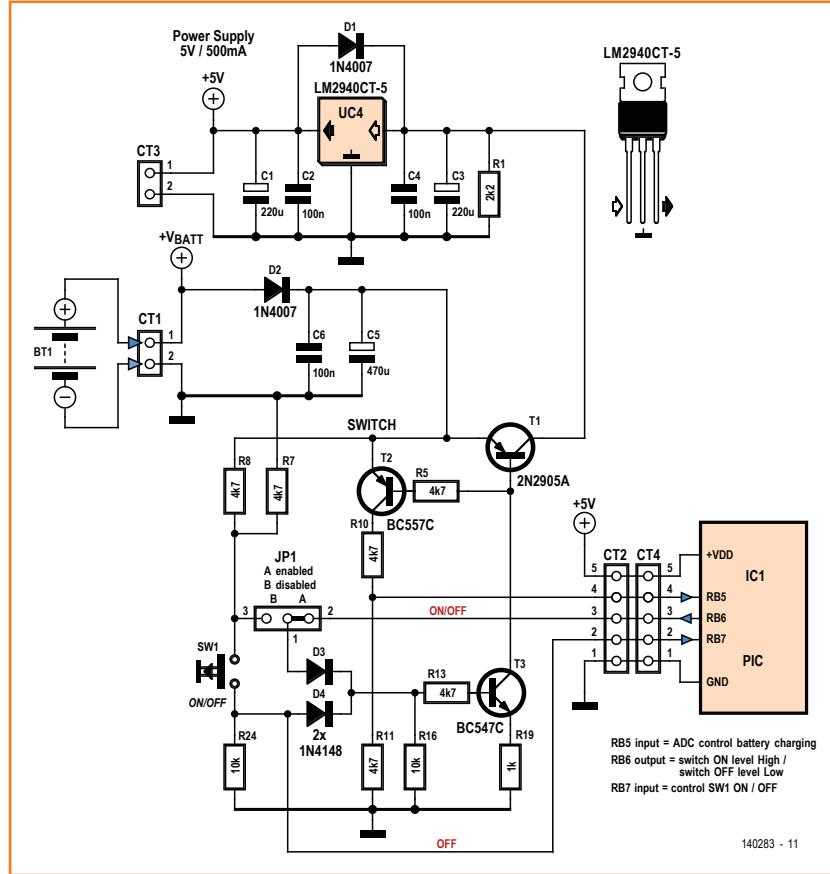
Ce circuit est un accessoire en lui-même 100 % analogique, mais conçu pour fonctionner en étroite collaboration avec un microcontrôleur (μC) dont il permet de programmer l'arrêt de manière à remplir, avant la coupure définitive de la tension d'alimentation, des conditions prédéfinies. Toute l'astuce consiste à faire interagir la partie analogique avec le programme du microcontrôleur. Quand nous mettons hors tension un appareil commandé par un μC , que ce soit délibérément (à

l'aide d'un bouton marche/arrêt) ou accidentel (par coupure de courant), l'état de notre circuit juste avant la coupure est perdu et nous ignorons ce que deviennent les périphériques abandonnés à eux-mêmes par l'extinction du μC . Dans bien des cas, il serait préférable que le μC procède, avant de s'éteindre, à une sauvegarde des données, p. ex. dans une EEPROM. La fonction de mon circuit d'arrêt contrôlé est précisément de ménager ce laps de temps au μC pour effectuer une telle opération.

On pourrait obtenir un effet similaire en mettant à profit la forte capacité d'un condensateur de filtrage de la tension d'alimentation ; la réserve d'énergie stockée dans un tel composant offre un court répit au μC pour faire une sauvegarde de données, mais la solution reste hasardeuse. Comment savoir par exemple si la sauvegarde a bien été faite ? Et si on souhaite aussi que le μC ne s'éteigne qu'après avoir mis certains organes périphériques dans une configuration de repos donnée, cette opération d'initialisation plus complexe risquerait fort de n'être pas achevée alors que le condensateur, lui, sera déjà déchargé depuis longtemps.

Autodétermination

Le circuit d'alimentation à arrêt contrôlé que je propose permet même au μC de décider lui-même de l'arrêt du système après avoir effectué toutes les tâches programmées (p. ex. sauvegarde et/ou mise au neutre des servos et/ou mise en position sécurisée d'autres appareils, etc.). Je n'utilise pour cela que des composants discrets, faciles à intégrer dans un autre montage. Je n'utilise pas de relais, dont la consommation serait plus forte que celle du microcontrôleur à alimenter. Sur le schéma (fig. 1), commençons par le transistor NPN T3. Il se voit appliquer sur sa base



140283 - 11

une tension de commande qui le rend conducteur (niveau haut) ou bloqué (niveau bas). En effet, quand le contact est établi (même fugitivement) par l'interrupteur SW1, qui est notre bouton marche/arrêt, le pont diviseur R7-R8 applique la moitié de la tension de la batterie à la base de T3 via la diode D4, tandis que R13 en limite le courant de base. Le collecteur de T3 commande la base de T1, un PNP monté en commutateur de tension de la batterie BT1. Celle-ci alimente via UC4 le circuit à µC représenté par le PIC en bas à droite du schéma. Ce circuit est maintenant sous tension. Le 2N2905A peut, sans chauffer, lui fournir un courant de 500 mA. Si votre circuit à µC a besoin de plus de courant que 0,5 A, vous remplacerez T1 par un transistor plus puissant, voire par un MOSFET de type P.

Interaction

La suite du fonctionnement de notre circuit analogique d'alimentation à arrêt contrôlé est tributaire du programme exécuté par le µC qu'il commande. Voyons comment cela se passe, d'abord lors de la mise sous tension normale. Le µC, une fois alimenté, doit placer sa sortie RB6 au niveau haut, ce qui, via D3, maintiendra la base de T3 au niveau haut, même quand le poussoir SW1 a été relâché. Le circuit alimenté par UC4 reste alors sous tension et fonctionne normalement. Quand on appuie sur SW1 pour *démarrer* le circuit comme nous venons de le voir, la tension haute appliquée à l'entrée de RB7 du PIC n'est pas prise en compte par lui ; il est alors occupé

à l'initialisation et au démarrage des appareils qu'il commande. Quand SW1 est relâché (contact ouvert), le niveau de cette entrée est bas du fait de la présence de R24. Il ne se passe rien. Nous en arrivons à la fonction principale de mon circuit : la coupure contrôlée de l'alimentation. Considérons que le circuit est en service normalement, et appuyons sur le bouton poussoir SW1 : l'entrée RB7 du PIC passe au niveau haut, ce qui est pour le µC l'ordre de sauvegarder ses données et d'exécuter les autres opérations jugées utiles avant l'arrêt (dans mon application, la mise en position neutre de servos). Une fois ces opérations effectuées, le microcontrôleur place sa sortie RB6 au niveau bas, ce qui pour effet de bloquer T3 et T1. Le circuit en aval n'est plus alimenté par UC4 et le microcontrôleur s'éteint. Rideau.

Ce n'est pas tout

Le cavalier JP1 peut être utilisé durant la programmation du µC, phase au cours de laquelle il faut éviter que la présence d'un niveau sur RB6 n'entraîne la coupure de l'alimentation. C'est pourquoi, si le µC utilisé place, par défaut, ses ports en sortie au niveau bas, il faut planter le cavalier en position B, de sorte que l'alimentation ne soit pas interrompue. Autrement, le cavalier doit être en position A.

Une fonction accessoire de ce circuit est la surveillance du niveau d'énergie de la batterie pour permettre son affichage par le PIC. Le transistor PNP BC557 T2 voit sur son émetteur la tension de la batterie. Comme sa base est reliée par R5

Mesure et affichage de la tension de la batterie

Voici un exemple de calcul du diviseur de tension pour une batterie li-ion de 7,4 V (nominal). Rechargée à 100 %, sa tension est de 8,30 V sous une charge de 50 mA. Retenons 6,5 V comme plancher pour considérer la batterie comme déchargée (0 %). La conversion A/N sur 8 bits, avec une échelle numérique de 0 à 255 pour une plage analogique de 0 à 5 V, donne $5 / 256 = 0,01953$ V par pas numérique. En général, le µC ne traite pas les nombres en virgule flottante, mais utilise des entiers au format Byte ou Word. Pour se débarrasser de la virgule dans le calcul, il suffit d'appliquer aux valeurs un facteur de 10, 100 ou plus au besoin, à condition toutefois de rester inférieur à 65535 (seuil de dépassement de capacité avec 16 bits) ou à 256 (avec 8 bits). Certaines valeur fixes (ratio) peuvent être calculées manuellement et introduites en constante dans le programme.

$$[1] \text{ avec } V_{\text{Bat}}/2 : 830 / 2 = 415 \text{ (100 \%)} \quad 650 / 2 = 325 \text{ (0 \%)} \quad [8.3 \text{ E}^3 \text{ et } 6.5 \text{ E}^3] \\ \text{valeur du pas numérique} = 1953 \quad [0.01953 \text{ E}^5]$$

$$[2] \text{ Ratio Min/Max } 212 - 166 = 46 \quad 100 / 46 = 2.174 \quad 2174 \quad [2.174 \text{ E}^3] \\ (415 * 1000 / 1953) = 212 \\ (325 * 1000 / 1953) = 166$$

La plage de la valeur de conversion 'ADC' du port variera entre 212 pour 100 % et 166 pour 0 %.

[3] Calcul en % de la tension de batterie à afficher :

$$100 - [(212 - \text{ADC}) / 1000]$$

Exemple de valeur de contrôle à afficher :

$$(\text{ADC}) = 180 \quad >>> | \quad 100 - [(212 - 180) / 1000] = 30 \% \quad |>>> \quad 7,04 \text{ V}$$

Figure 2.
Dessin et plan
d'implantation du circuit
imprimé de l'auteur.

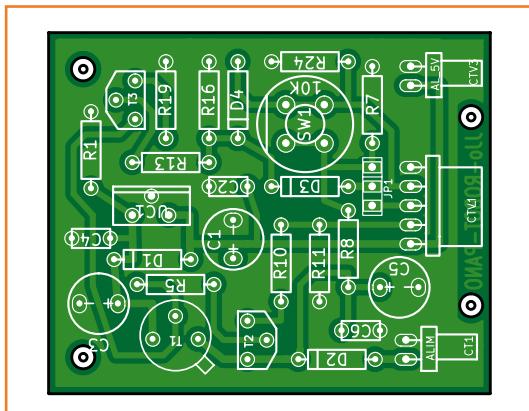
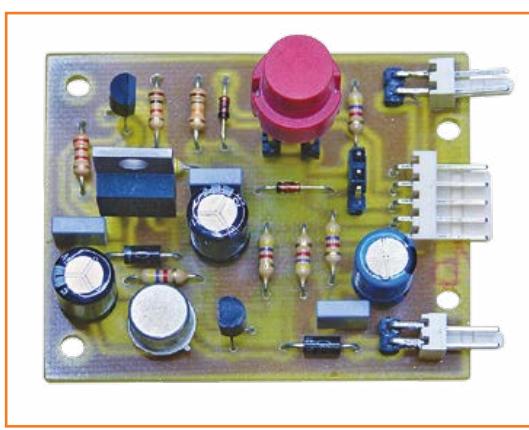


Figure 3.
Prototype de l'auteur.



à la base de T1, il devient conducteur en même temps que celui-ci ; le PIC pourra donc mesurer sur son entrée RB5 (configurée en convertisseur analogique-numérique) la tension prélevée par le

pont diviseur R10/R11 sur le collecteur de T2. Ce transistor, bloqué pendant la coupure de l'alimentation, empêche toute consommation sur la batterie au repos. En effet, il ne doit régner aucune tension sur les ports d'un microcontrôleur lorsqu'il n'est pas lui-même alimenté. D'autre part, la division de tension sur l'entrée RB5 est nécessaire, car la tension sur un port d'un microcontrôleur ne doit jamais dépasser sa tension d'alimentation nominale. Avec un facteur de division de 2, même quand la batterie sera pleinement chargée, la tension sur RB5 ne dépassera jamais 5 V. Le circuit est par ailleurs protégé des erreurs de polarité par D2, tandis que D1 et R1 déchargent les condensateurs de lissage C1 et C3 à l'arrêt.

Programmation du µC

J'ai utilisé ce montage pour commander des circuits complexes, avec des servos pour commander les mouvements d'un robot. Le code du PIC est volumineux et ne présenterait pas grand intérêt ici. Je donne néanmoins les grandes lignes du déroulement des actions (**listage 1**) dont vous pourrez vous inspirer pour intégrer ce circuit dans vos propres applications à base d'un microcontrôleur quelconque (PIC, AVR, Arduino). Le principe sera le même, quel que soit le langage que vous utiliserez (C, BASIC...).

Si le µC est fortement sollicité par des routines en attente d'événements, il est recommandé d'utiliser une interruption pour lire le port B7.

(140283)

Listage 1

```

//-----
Initialisation :
Port B 5 et B7 en entrée (IN) et B6 en sortie (OUT)
Initialisation port B5 ADC 8 bits
Port B6 mise au niveau haut (high).
Suite de l'initialisation . .
. .
. .
//-----
main : (boucle principale du microcontrôleur)
do
. . . . . (diverses actions)
. . . .
//-----
lecture du port B7
    si niveau haut (high)
exécuter la routine de sauvegarde (données, servos au neutre . . . )
mettre le port B6 au niveau bas (low) (coupure de l'alimentation)
//-----
lecture du port B5 (ADC)
    exécuter la routine de mise à jour du coefficient charge de la batterie
    affichage
loop
//-----
```



lampes au néon

drôle de composant n° 10

Cet article m'a été inspiré par une vieille alimentation Tektronix qui utilise une lampe au néon. J'ai d'abord cru qu'elle servait d'indicateur de tension, mais un rapide examen m'a confirmé qu'elle faisait en fait partie du circuit de protection, ce qui, vous l'imaginez, m'a surpris. Examinons le mode de fonctionnement des lampes au néon et leur utilisation dans ce type d'applications.

Les lampes au néon — connues aussi comme *petites loupiotes* — sont des composants très simples. Elles sont fabriquées en remplissant un petit tube de verre muni de deux électrodes d'un mélange de gaz néon à basse pression (**fig. 1**). Lorsqu'elles sont alimentées par une tension continue, seule leur électrode négative brille ; une tension alternative fera briller les deux. Leur symbole (**fig. 2**) rappelle leur structure.

Une lampe au néon se comporte comme un circuit ouvert lorsqu'elle est éteinte et ne s'illuminera que lorsque la tension à ses bornes aura atteint la tension de claquage du gaz. Une fois illuminée sa tension de fonctionnement baissera significativement et restera constante sur une plage de courants étendue. C'est ce comportement qui fait que l'on trouve généralement une résistance de forte valeur ($100+ k\Omega$) en série avec la lampe. La tension de claquage se situe habituellement aux environs de 90 V pour une lampe NE-2 classique qui, une fois illuminée, voit la tension à ses bornes décroître à environ 60 V. Étant donné que le seuil de tension du gaz peut être affecté par la luminosité ambiante, on ajoute habituellement un gaz radioactif afin de s'assurer que la lampe s'illuminera comme prévu.

Comment les lampes au néon sont-elles utilisées dans une alimentation ? Leur haute tension de déclenchement et l'hystéresis qu'elles présentent les rendent d'autant plus utiles comme protection contre les surtensions qu'elles donnent une indication visuelle du défaut. Aujourd'hui, il est préférable d'utiliser des varistances et des diodes *transil* pour la protection, mais la loupiote au néon reste un superbe indicateur de tension pour les alimentations.

Leur résistance négative rend les lampes au

néon utiles pour bien d'autres applications, par exemple, en conjonction avec une résistance et un condensateur, comme élément actif d'un oscillateur à relaxation (**fig. 3**). Deux lampes peuvent également être utilisées comme multivibrateur astable. En mettant plusieurs en cascade, on peut réaliser des circuits complexes tels que des compteurs en anneau, des diviseurs (**fig. 4**) ou d'autres circuits logiques simples.

Les loupiotes au néon existent depuis longtemps et possèdent un grand nombre d'applications, comme en témoigne leur présence à des endroits inattendus ; je pense par exemple à un orgue électrique des années 60. Ce qui étonnant avec ces lampes, c'est qu'elles sont toujours disponibles dans une gamme étendue de modèles, faciles à trouver. En plus, elles se prêtent bien à l'expérimentation, à condition de savoir que les modèles récents présentent un écart typique moindre entre tension de claquage et tension de maintien : selon cet écart, il vous faudra peut-être ajuster certains circuits pour qu'ils fonctionnent correctement.

(140291 – version française : Kévin Petit)

Neil Gruending
(Canada)



Figure 1.
Lampes au néon.

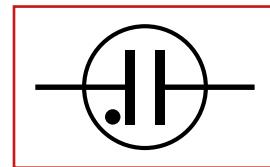


Figure 2.
Symbole d'une lampe au néon.

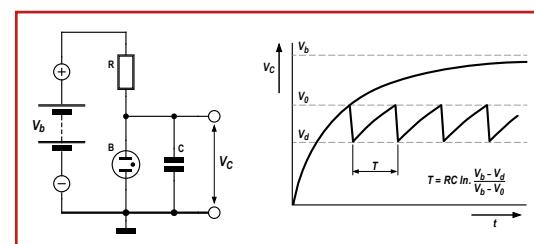


Figure 3.
Une lampe au néon comme oscillateur à relaxation et la forme d'onde typique qui en résulte.

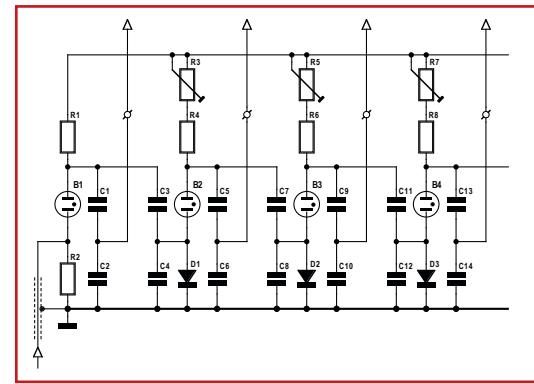


Figure 4.
Ces lampes au néon mises en cascade servent de diviseur de fréquence dans un orgue électrique.



15^e jour : renumérotation de composants

Neil Gruending
(Canada)

Aujourd’hui nous allons apprendre à renuméroter des composants dans DesignSpark PCB.

Vous êtes-vous déjà demandé comment se fait la numérotation des cartes afin qu'il soit facile d'y retrouver les composants ? Aujourd’hui nous allons apprendre à renuméroter les composants d'un C.I. puis à mettre à jour le schéma en utilisant l'outil de rétro-annotation.

Renumérotation de composants

L'outil de renumérotation de DesignSpark fonctionne en divisant chaque face d'une carte en bandes puis en y changeant les références si nécessaire. L'outil se configure en utilisant les directions physiques : de gauche à droite, de haut en bas, de droite à gauche ou de bas en haut. Habituellement on renomme les composants de gauche à droite et de bas en haut (**fig. 1**, gauche), mais on peut utiliser n'importe quelle combinaison. La partie droite de la figure donne un aperçu de la combinaison droite à gauche et bas en haut. Dans DS on spécifie d'abord le sens des bandes puis le sens à utiliser au sein des bandes. Dans notre exemple de gauche à droite et de haut en bas, le sens des bandes va de gauche à droite

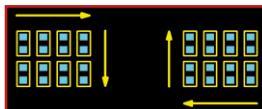


Figure 1.
Sens de la renumérotation.

et le sens à l'intérieur des bandes de haut en bas. La seule contrainte imposée par DS est que vous devez utiliser une direction verticale et une horizontale ; vous ne pourrez donc pas utiliser de haut en bas et de bas en haut par exemple. L'outil de renumérotation parcourt chaque bande dans le sens spécifié à la recherche de composants. Le premier composant sera renuméroté avec la première référence disponible, en commençant au numéro un. Si plusieurs composants sont alignés dans le sens de parcours, l'outil les renumérotera en utilisant le sens intra-bande. Dans notre exemple, l'outil parcourrait chaque bande de gauche à droite puis de haut en bas à la recherche de composants alignés verticalement (fig. 1, gauche).

Jetons maintenant un œil à la fenêtre de renumérotation de composants (**fig. 2**) à laquelle on accède via l'entrée *Auto Rename Components* du menu *Tools*. La partie *Rename Which Components* vous permet de sélectionner les composants à renommer. Normalement vous choisissez *All Components*, mais vous pouvez également choisir les composants à renommer sur la base de leur référence ou en fonction du côté de la carte sur lequel ils se situent. La partie *Multiple Board Outlines* est utile si votre projet comporte plusieurs cartes ; le réglage par défaut inclura toutes les cartes dans la même numérotation. Nous avons déjà évoqué le réglage des sens de parcours de la partie *Direction*, mais n'avions alors pas mentionné le paramètre *Strip Width* qui permet de choisir la largeur des bandes de parcours. Je vous recommande également d'activer l'option *Reverse Left-Right* afin que DS inverse le sens de parcours lorsqu'il s'occupera du dessous de la carte. C'est très pratique et produit le même effet que de retourner virtuellement la carte avant d'effectuer l'opération de renumérotation. Enfin, la partie *Other Settings* permet de spécifier le point de départ de la numérotation.

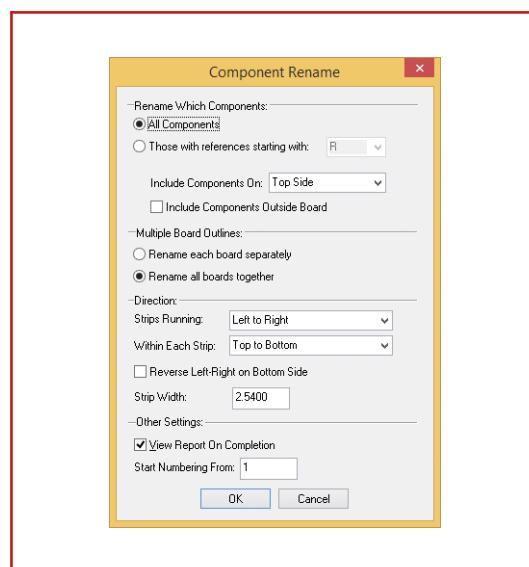


Figure 2.
Fenêtre de renumérotation de composants.

Vous trouverez en **figure 3** un exemple de renumérotation automatique : DS a créé la carte de droite à partir de celle de gauche. J'ai réglé le sens de parcours sur « de gauche à droite » et

« de haut en bas ». J'ai délibérément choisi une bande étroite (5 mm) afin de vous montrer un exemple avec plusieurs bandes. Vous remarquerez que la référence du transistor n'a pas changé : DS utilise un compteur par type de composant.

Mise à jour du schéma

Maintenant que nous avons renommé le C.I., il est temps de mettre à jour le schéma avec les nouvelles références. Le processus est nommé rétro-annotation et, c'est heureux, DS comporte dans ce but un outil automatique auquel on accède via l'entrée *Tools* → *Backwards Annotation* du menu. L'outil importera dans le schéma les références depuis le fichier du circuit imprimé afin que les deux correspondent. Il ne transférera pas les autres changements tels que ceux affectant les signaux ou les propriétés des composants. L'outil d'annotation suppose toujours qu'il y a un ou plusieurs schémas associés au fichier C.I. Si vous n'utilisez pas de fichier de projet, l'outil supposera alors que vous n'utilisez qu'un seul schéma dont le fichier porte le même nom que celui du C.I. Si vous avez un fichier de projet, il est possible d'utiliser plusieurs fichiers de schéma avec des noms différents. Malheureusement il n'est pas possible de changer ce comportement ou de sélectionner manuellement le fichier de schéma à mettre à jour.

La **figure 4** vous donnera un aperçu de l'aspect de l'outil de rétro-annotation. La mise à jour du schéma est effectuée quand vous cliquez sur OK, mais il est possible d'obtenir un aperçu des changements qui seraient effectués en cliquant sur *View Renames*. C'est pratique pour vérifier vos changements sans que le schéma soit modifié. Le bouton *Delete Renames* est un peu différent : il ne déclenchera pas de mise à jour du schéma, mais enregistre les modifications faites dans le circuit imprimé. Vous pourriez l'utiliser après une édition manuelle du schéma par exemple. Soyez toutefois prudent : une fois les renomérotations effacées, vous ne pourrez pas les réappliquer plus tard. La dernière option, *View Report on Completion*, permet de demander la production d'un rapport détaillé sous la forme d'un fichier texte. Je vous recommande de toujours sauvegarder vos schémas avant de cliquer sur OK : les modifications ne peuvent pas être annulées avec la fonction d'annulation.

La **figure 5** vous montre un exemple de rétro-annotation : avant à gauche, après à droite. Comme vous pouvez le voir, l'outil d'annotation a rem-

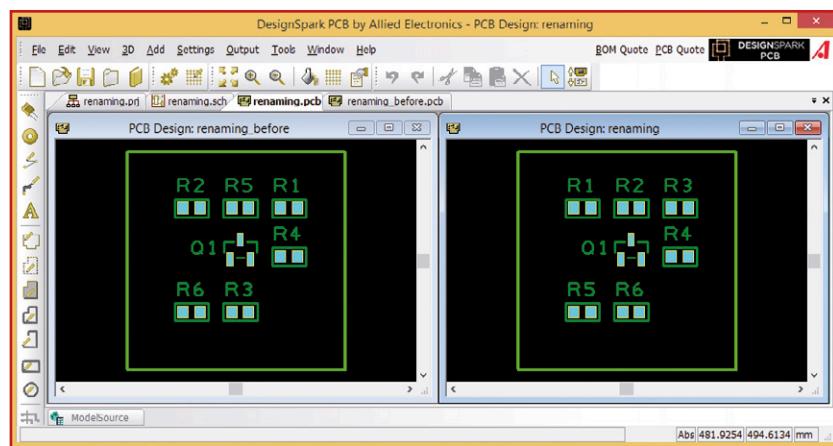


Figure 3.
Un exemple de
renumérotation, avant et
après.

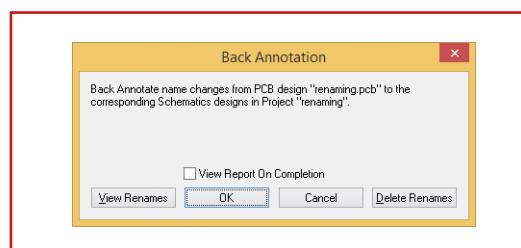


Figure 4.
Outil de rétro-annotation.

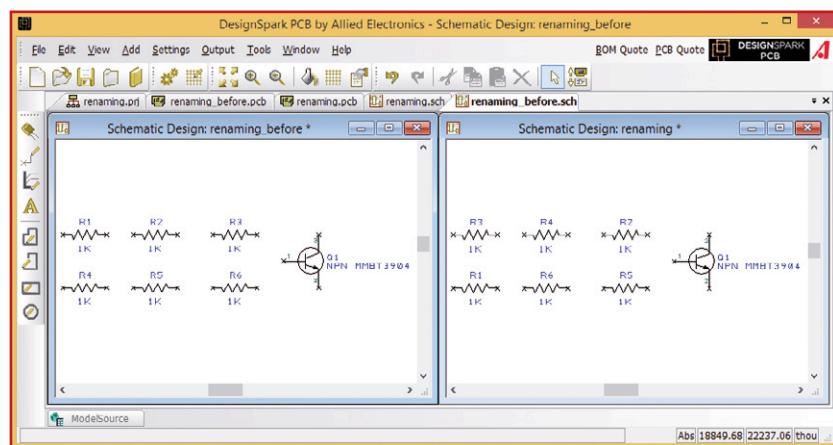
placé les références par celles du C.I. sans faire aucun autre changement.

Conclusion

Nous venons de jeter un œil à l'outil de renomérotation de composants de DesignSpark. Il constitue le seul moyen de renommer automatiquement des composants et vous permettra de facilement retrouver les composants sur votre prochaine carte. La prochaine fois nous parlerons encore de composants et de leur utilisation dans les schémas et les circuits imprimés.

(140292 – version française : Kévin Petit)

Figure 5.
Les composants du schéma
ont été renommer.



Arduino avec Atmel Studio débogage d'applications Arduino Due

Wolfram Pioch
(Allemagne)

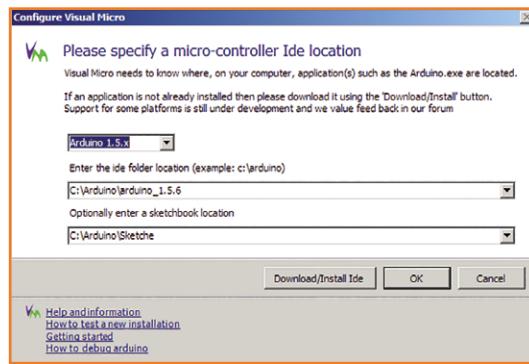


Lorsqu'on est habitué au confort que procurent Atmel Studio et un débogueur pour programmer des AT(x)mega, le passage à l'EDI Arduino pour programmer l'ARM Cortex M3 de la carte Arduino Due peut être assez abrupt. Voici donc comment installer les pattes Arduino dans le moelleux du Studio Atmel.

Figure 1.
Sélection de la version d'Arduino, de la carte et de l'interface de programmation.



Figure 2.
Entrée du répertoire Arduino dans le gestionnaire de configuration.



Si l'environnement de développement Arduino dispose de nombreux croquis d'exemple indiscutablement utiles pour qui cherche à obtenir rapidement des résultats, la programmation de contrôleur ARM sous cet EDI n'en souffre pas moins de quelques défauts :

- la compilation des programmes dure trop longtemps ;
 - le chargement d'un programme est trop long ;
 - il manque un véritable débogueur matériel.
- Ces trois points gênant le bon déroulement de mon travail, j'ai cherché une solution de remplacement.

Une des raisons de vouloir passer à un microcontrôleur à 32 bits est qu'avec ce type de puce la communication avec le matériel est plus rapide, en particulier avec les broches d'E/S. Un bon moyen de mettre à l'épreuve la puissance d'un contrôleur est donc de mesurer des impulsions de tension très

courtes. Plutôt que de montrer comment coder ces impulsions depuis l'EDI Arduino, je vais montrer comment y parvenir avec Atmel Studio 6.2 [1].

Compatibilité Arduino

Pour rendre compatibles les environnements Arduino et Atmel, il vous faudra non seulement Atmel Studio 6.2 (ou une version plus récente) mais également le greffon gratuit appelé Visual Micro [2]. Ce programme transforme Microsoft Visual Studio et Atmel Studio en EDI Arduino. Il suffit de l'installer dans Atmel Studio pour aussitôt pouvoir l'utiliser.

Le débogueur USB optionnel et payant que propose l'éditeur de Visual Micro n'est pas indispensable, je trouve le programmateur/débogueur Atmel-ICE beaucoup plus pratique. Atmel-ICE coûte de 100 à 120 € chez différents revendeurs en ligne, un bon investissement pour qui se sert souvent de microcontrôleurs de chez Atmel (cf. le numéro précédent du magazine [3]).

Atmel-ICE permet de placer huit points d'arrêt sans avoir à recompiler le code, de surveiller des variables dans la fenêtre *Watch*, d'inspecter et modifier le contenu de la mémoire et, surtout, de visualiser les larges registres d'E/S et de les configurer à la souris.

Une nouvelle barre d'outils apparaît dans Atmel Studio après l'installation de Visual Micro. Nous y sélectionnons (**fig. 1**) la version d'Arduino (1.5.x), la carte (Arduino Due) et l'interface de programmation (*Programming Port*). Nous devons également paramétrer l'interface série virtuelle. Ce numéro de COM est indiqué dans le gestionnaire de périphériques lorsque la carte Arduino est reliée au PC par câble USB. Le bouton à droite de ces menus déroulants lance le *Moniteur série*. Si ces éléments ne sont pas présents, c'est probablement que vous avez installé l'EDI Arduino dans un répertoire non standard. Dans ce cas, ouvrez le gestionnaire de configuration (*Configuration Manager*) et entrez-y le chemin d'accès vers Arduino (**fig. 2**). Un clic sur le point d'interrogation du menu de la figure 1 ouvre la fenêtre *Micro Explorer*, le pendant de l'explorateur *Solution Explorer* d'Atmel Studio (**fig. 3**). L'onglet *References* contient une table des matières de la documentation en ligne du site Arduino. Cliquer sur un lien ouvre la page associée dans votre navigateur. L'onglet *Examples* contient la liste thématique des codes sources d'exemple fournis par votre plateforme Arduino. Vous pouvez les charger comme projet ou ouvrir le code source d'un seul fichier.

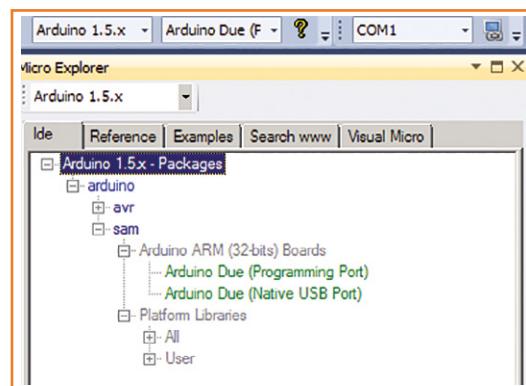


Figure 3.
L'explorateur *Micro* de
Visual Micr

Écrire des croquis dans Atmel Studio

Un des avantages d'Atmel Studio est le complètement du code, une fonction désormais également utilisable dans les croquis (*sketches*) Arduino. Pour disposer du complètement, activez *Visual Assist X* via *VAssistX* → *Enable/Disable*. Pour chaque lettre tapée s'affiche alors la liste des mots pouvant compléter la ou les lettre(s). Si p. ex. vous tapez « S », les choix seront *Serial*, *Server*, *SPI*, etc. Le complètement évite bien sûr de faire des fautes de frappe, mais possède également une vertu documentaire : si vous tapez un point derrière le nom d'une classe, p. ex. derrière *Serial*, l'éditeur vous propose la liste des méthodes et propriétés associées à cette classe (fig. 4).

Nous pourrions maintenant ouvrir ou créer un croquis dans l'EDI Arduino puis l'importer dans Atmel Studio, mais il serait dommage de se priver de la possibilité de pouvoir créer un croquis directement depuis Atmel Studio. Après avoir créé et enregistré le croquis *SpeedTest* (**listage 1**) selon l'une ou l'autre de ces deux méthodes, l'explorateur de solution montre l'arborescence de la **figure 5**. Le croquis peut alors être ouvert et modifié comme n'importe quel autre fichier. On peut se passer de l'EDI Arduino à ce stade, mais pas le supprimer puisqu'il doit être présent sur le PC.

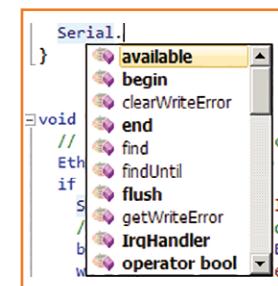


Figure 4.
La fonction de complètement
de l'éditeur d'Atmel Studio
évite les erreurs de saisie.

Adaptateur pour le SAM-ICE

Le circuit du schéma permet de relier le débogueur SAM-ICE à une carte Arduino Due. Il relie le connecteur JTAG à 10 contacts et au pas de 0,05 pouce (1,27 mm) de la Due au connecteur JTAG à 20 contacts du SAM-ICE.

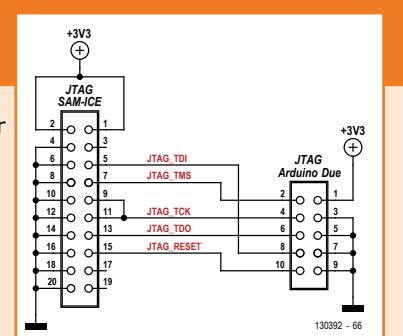


Figure 5.
L'explorateur de solution
d'Atmel Studio.

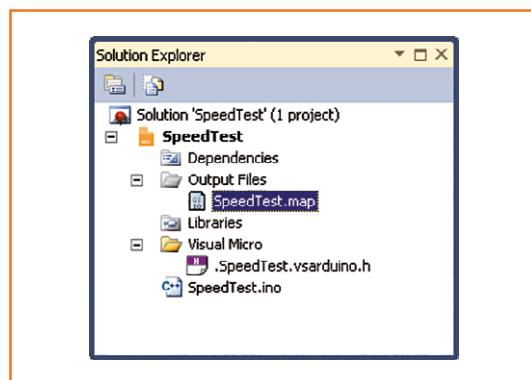


Figure 6.
La plateforme indiquée n'est
pas la bonne, mais tout
fonctionne.

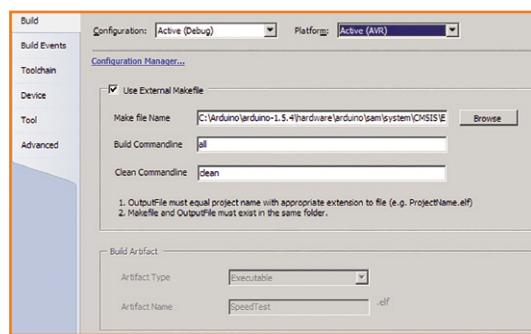
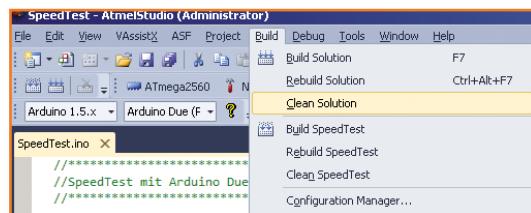
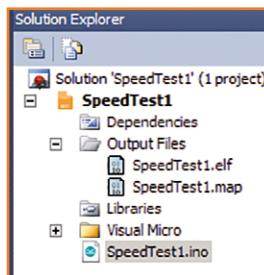


Figure 7.
L'option de compilation
Clean Solution.



Un point déconcertant, au moins dans la version 6.2 actuelle, est le « Active AVR » qui s'affiche comme plateforme du projet (clic droit sur *SpeddTest*, puis *Properties* → *Build*), une erreur hélas impossible à corriger. Cette bizarrerie s'incruste également sous *Toolchain* et *Device*, mais néanmoins tout fonctionne. Sélectionner *Arduino Due* comme carte déclenche la compilation du projet comme avec la plateforme SAM (pour les contrôleurs ARM d'Atmel). La **figure 6** montre une soi-disant plateforme AVR, mais le nom du fichier « make » (*Make file name*) est correct pour la chaîne d'outils SAM de l'EDI Arduino. Comme d'habitude sous Atmel Studio, le projet se compile avec F7 et le code avec F5 (ou en cliquant sur la flèche verte) via le chargeur d'amorçage de la carte. Utiliser le menu *Build* → *Clean Solution* (**fig. 7**) est capital pour que tous les fichiers du projet soient compilés comme ils le seraient avec l'EDI Arduino.

Figure 8.
Les fichiers produits par la
compilation.



Comme je l'ai dit, un des grands avantages d'Atmel Studio par rapport à l'EDI Arduino est sa vitesse de compilation. Rien de plus désespérant en effet que d'avoir à attendre entre chaque compilation-réécriture du code. Là où l'EDI Arduino compile systématiquement l'ensemble des fichiers d'un projet, Atmel Studio ne recompile que ceux qui ont été modifiés, et en général uniquement le croquis .ino contenant le code. Mon PC a mis 3,7 s pour compiler une première fois le croquis *SpeedTest*, et seulement 0,23 s pour le recompiler après modification.

Débogage avec Atmel-ICE

Se servir d'un « vrai » débogueur matériel comme Atmel-ICE rend l'écriture de programmes beaucoup plus agréable et pratique. Relier Atmel-ICE à un PC nécessite uniquement le câble fourni avec le kit *Basic*. On peut alimenter la carte Arduino Due via l'un de ses connecteurs USB ou via une alimentation externe. Le débogueur est alimenté par la liaison USB.

Atmel-ICE est équipé de deux connecteurs, un SAM et un AVR. Pour la carte Due, c'est dans le connecteur SAM qu'il faut brancher le câble de débogage. L'autre extrémité se relie au connecteur JTAG de la carte.

Dans la version actuelle 6.2 d'Atmel Studio, il est à vrai dire impossible de déboguer un croquis Arduino avec Atmel-ICE. Il existe tout de même une solution : ouvrir une seconde instance d'Atmel Studio et sélectionner *Open* → *Open Object File For Debugging*. Sous Windows 7, les fichiers créés par la compilation se trouvent dans le dossier :

```
C:\Users\xxxx\AppData\Local\  
VMicro\Arduino\Builds\SpeedTest\  
arduino_due_x_dbg\
```

S'y trouve le fichier objet *SpeedTest.elf*, que nous allons ouvrir pour la session de débogage. En présupposant que le croquis a été compilé et sauvegardé depuis l'EDI Arduino, le dossier de débogage en contient également une copie. On retrouve ce fichier dans la section *Output Files* de l'explorateur de solution (**fig. 8**). L'emplacement du fichier *SpeedTest.elf* se sélectionne depuis la fenêtre de dialogue ouverte. Vous devriez voir une fenêtre semblable à celle de la **figure 9**. J'ai ajouté au nom du projet le suffixe *_Debug* et laissé l'emplacement du projet proposé par défaut. Les paramètres sauve-

débogage d'applications Arduino Due

gardés restent associés au projet de débogage à chaque ouverture.

La fenêtre suivante nous demande de choisir la plateforme et le microcontrôleur cible (**fig. 10**). La famille de la carte Arduino Due est *SAM3, 32 bits*, et le microcontrôleur est le *ATSAM3X8E*.

Dans l'explorateur de solution apparaît alors le fichier *SpeedTest.cpp*, par ailleurs identique au croquis d'origine *SpeedTest.ino*. Un clic droit sur le nom du projet (*SpeedTest_Debug*) ouvre la fenêtre des propriétés.

Lorsque le débogueur Atmel-ICE est relié au PC par USB, on peut le sélectionner depuis *Tool → Selected debugger/programmer*. Veillez à bien choisir SWD (*Serial Wire Debug*) comme interface et non pas JTAG, sinon rien ne fonctionnera. Atmel-ICE est maintenant prêt et peut aussi être utilisé comme programmeur en ouvrant la fenêtre correspondante avec Ctrl-Maj-P ou encore en cliquant sur l'icône représentant une puce zébrée d'un éclair. Les fonctions de programmation sont disponibles après avoir cliqué sur le bouton *Apply*. Dans le menu *Interface Settings* nous pouvons régler la fréquence d'horloge SWD sur sa valeur maximale de 2 MHz. Le fichier *SpeedTest.elf* peut ensuite être chargé directement dans la mémoire flash du contrôleur depuis l'onglet *Memory*. Le programme se lance une fois terminé le processus de programmation.

Une façon plus efficace de charger le programme pour le déboguer ou l'exécuter est de passer par le menu *Debug → Start Debugging and Break* ou par *Start Without Debugging* (ou de cliquer sur la flèche bleue flanquée de deux barres verticales, ou sur la flèche verte).

Le chargement du programme dans le microcontrôleur ne prend qu'une seconde avec Atmel-ICE, ce qui est vraiment rapide.

Après le chargement, la structure complète du programme Arduino apparaît dans l'explorateur de solution. Le fichier *main.cpp* est ouvert dans l'éditeur et le curseur est positionné sur la première ligne de la fonction *main* (**fig. 11**). On voit bien dans l'explorateur que le fichier *SpeedTest.cpp* ne représente qu'une fraction de l'ensemble du programme, ce qui explique que la compilation soit si longue.

Tous les avantages d'Atmel Studio peuvent maintenant être mis à profit avec le débogueur : débogage instruction par instruction avec F11,

points d'arrêt avec F3, conditions pour les points d'arrêt, etc.

Si nous modifions le croquis *SpeedTest.ino* dans la première instance d'Atmel Studio, puis le com-

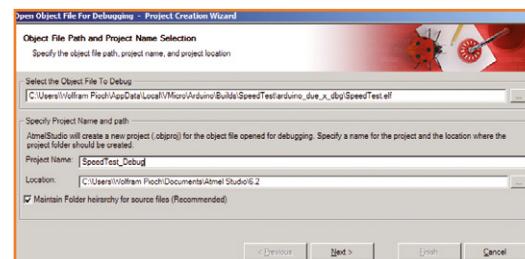


Figure 9.
Sélection du fichier objet et nom du projet.

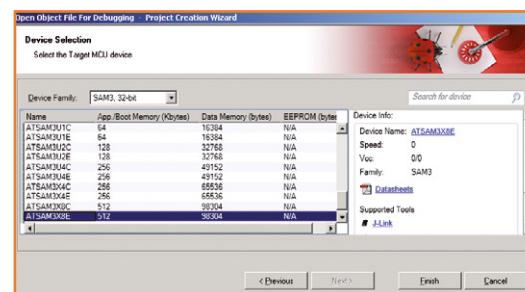


Figure 10.
Sélection de la puce cible et de sa famille.

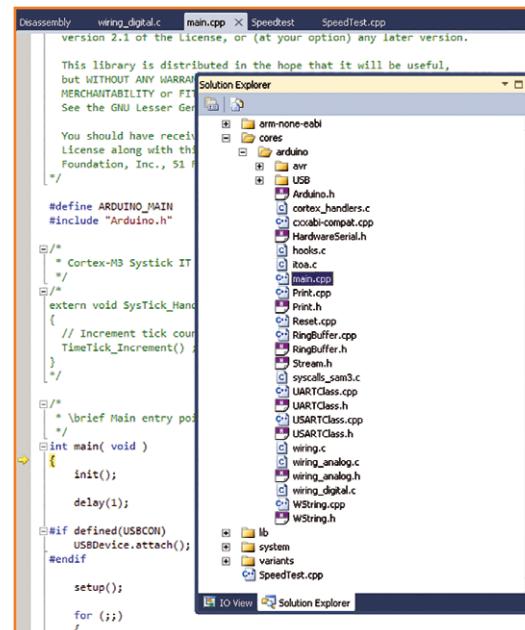


Figure 11.
La complexité d'un projet Arduino révélée par l'explorateur de solution.

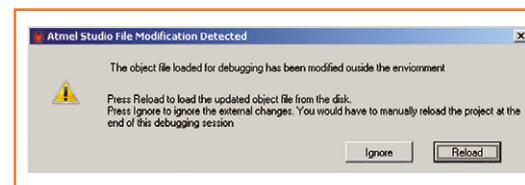


Figure 12.
Détection d'une modification : l'avertissement peut être ignoré.

Figure 13.
Oscilloscopogramme du signal rectangulaire de 150 kHz.

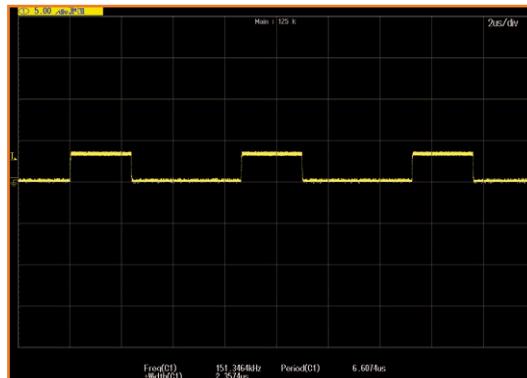


Figure 14.
Les registres PIO du contrôleur.

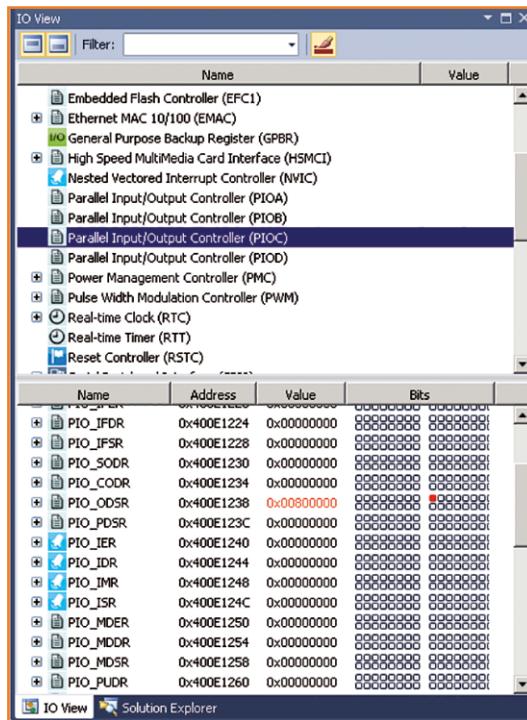
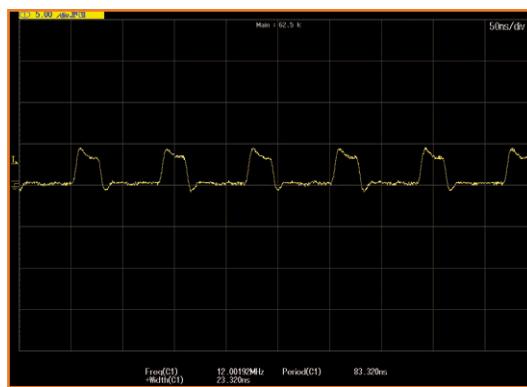


Figure 15.
Oscilloscopogramme du signal de 12 MHz obtenu avec le croquis Speed-Test 2.0.



pilons à nouveau, la seconde instance (celle de débogage) émet l'avertissement reproduit sur la **figure 12**. Cliquons sur *Reload* sans nous préoccuper de ce message : Atmel Studio abandonne la session de débogage et charge le croquis modifié dans le contrôleur. Le curseur est de nouveau positionné sur la première ligne de la fonction *main* et le programme peut être redémarré.

Si vous êtes habitué à travailler avec des AVR à 8 bits sous Atmel Studio, le passage aux ARM à 16 bits devrait se faire en douceur car le programme s'utilise quasiment de la même façon avec ces deux familles. Notez en passant que l'effacement et la réécriture de la mémoire flash avec Atmel-ICE n'a aucun effet sur le chargeur d'amorçage, car celui-ci se trouve dans la ROM du contrôleur et ne peut donc être ni supprimé ni réécrit. Le chargement d'un programme par interface USB fonctionne donc toujours.

Le test de vitesse

La boucle principale du croquis d'exemple (**listage SpeedTest**) n'exécute que deux instructions :

```
digitalWrite(TP1,1); //On
digitalWrite(TP1,0); //Off
```

L'observation de la broche de test à l'oscilloscope (**fig. 13**) montre un signal rectangulaire dont la largeur d'impulsion vaut 2,35 μs et la période 6,6 μs, soit une fréquence d'environ 150 kHz. Devant cette valeur relativement faible, la première pensée qui vient à l'esprit est de se demander si l'utilisation d'un ARM cadencé à 84 MHz est vraiment judicieuse quand on sait qu'un simple AVR à 8 bits peut faire mieux.

Cette valeur n'a à vrai dire rien de suprenant : sous la simple instruction `digitalWrite` se cachent en effet plusieurs routines C, comme on peut d'ailleurs le vérifier avec le débogueur en mode « instruction par instruction » (F11). Pour obtenir une vitesse de traitement élevée, il faut donc éviter le « surcoût » (*overhead*) dû à ces routines. Par chance, le langage Arduino n'est pas un « véritable » langage mais un C ordinaire (avec un compilateur GNU configuré spécialement pour Arduino).

Nous pouvons ainsi accéder à une broche configurée en sortie en manipulant directement un bit du registre PIO (*Parallel Input/Output*). L'instruction

```
const int TP1 = 7; //Testpin
```

sélectionne la broche Arduino 7 comme broche de test ou, selon le brochage Arduino, la broche PC23. Le bit correspondant du registre PIO est le 23. Si, dans la fenêtre *IO View* du débogueur (**fig. 14**), nous cliquons sur le bit 23 du registre *PIO_ODSR* (= *Set*) après avoir placé un point d'arrêt dans le programme, la broche est mise au niveau haut. Pour la remettre au niveau bas, il faut modifier le registre *PIO_CODR* (= *Clear*). Dans la boucle principale, c'est donc par l'intermédiaire du registre *PIO_ODSR* que nous mettons directement la broche 7 au niveau haut :

```
PIOC->PIO_SODR = 1<<23;
```

J'ai également écrit une version du croquis qui teste la compilation conditionnelle (**listage Speed-Test 2.0**). Lorsque le compilateur rencontre la directive `#ifdef Direct`, il regarde si le nom `Direct` est ou non défini à ce moment-là. S'il l'est (`#define Direct`) le code qui suit l'ins-

Listage 1. Speed-Test.

```
////////////////////////////////////////////////////////////////////////
//Speed-Test with Arduino Due
////////////////////////////////////////////////////////////////////////

const int TP1 = 7; //Testpin

void setup()
{
    /* add setup code here */
    // set the digital pin as output:
    pinMode(TP1, OUTPUT);
}

void loop() {
    // put your main code here, to run repeatedly:

    digitalWrite(TP1,1); //On
    digitalWrite(TP1,0); //Aus
}
```

Publicité

The new EAGLE has landed!

Version 7
now available

For more information,
visit www.cadsoftusa.com

Listage 2. Speed-Test 2.0.

```

//*****
//Speed-Test 2.0 with Arduino Due
//*****

#include „arduino.h“
//
//
const int LED1 = 13;
    int LED2 = 12;
    int LED3 = 11;
    int TP1 = 7; //Testpin

#define Direct

void setup()
{
    /* add setup code here */
    // set the digital pin as output:
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
    pinMode(LED3, OUTPUT);
    pinMode(TP1, OUTPUT);
    // set output low
    digitalWrite(LED1,0);
    digitalWrite(LED2,0);
    digitalWrite(LED3,0);
    digitalWrite(TP1,0);
}

void loop() {
    // put your main code here, to run repeatedly:
#endif Direct
//= state of PIO_SODR with bit 23 = 1
int x = PIOC->PIO_SODR | 1<<23;

while (1){ //Only for testpurposes, don't exit this loop
    PIOC->PIO_SODR = x;
    PIOC->PIO_CODR = x;
}

#else
    digitalWrite(TP1,1);
    digitalWrite(TP1,0);
#endif
}

```

truction `#ifdef` est exécuté, sinon (`#else`) c'est l'instruction (lente) `digitalWrite` qui est utilisée. Ce code fonctionne bien sûr aussi avec l'EDI Arduino.

Le résultat de cette optimisation est notable : la largeur d'impulsion du signal rectangulaire n'est maintenant plus que de 23,2 ns, et la période est descendue à 83,3 ns (**fig. 15**), ce qui correspond à une fréquence de 12 MHz. Autrement dit nous avons multiplié par 80 la fréquence précédente ! Utiliser un ARM est donc loin d'être sans effet.

Pour résumer

Le greffon Visual Micro pour Atmel Studio ne recompile que le code modifié et réduit donc considérablement l'inévitable cycle modification-compilation d'un code Arduino. Recourir au programmeur/débogueur Atmel-ICE lorsqu'on développe des projets reposant sur des microcontrôleurs Atmel n'est cependant pas un luxe car, outre les fonctions de débogage professionnelles qu'il apporte, cet outil accélère sensiblement la vitesse d'écriture des programmes dans la mémoire du contrôleur.

Pour le débogage des projets ARM, nous pouvons également utiliser le SAM-ICE d'Atmel. Ce programmeur/débogueur matériel réduit encore le temps de programmation du microcontrôleur d'un facteur 4 puisqu'il peut travailler à 8 MHz, contre 2 MHz pour Atmel-ICE. L'adaptateur décrit dans **l'encadré** permet de l'utiliser avec une carte Arduino Due. Un développeur professionnel amortira vite son investissement, même si cet outil ne peut être utilisé qu'avec les contrôleurs Atmel de la série AT91.

(130392 – version français : Hervé Moreau)

Liens :

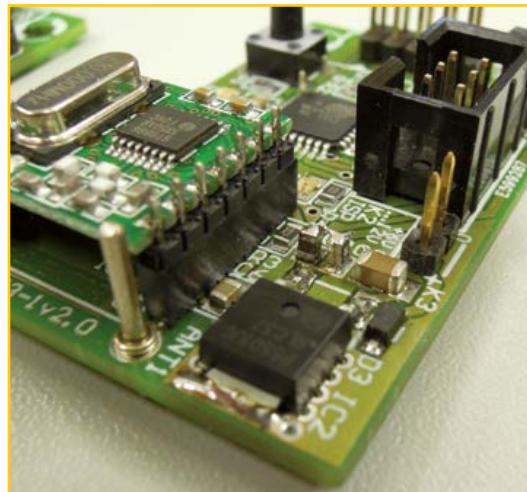
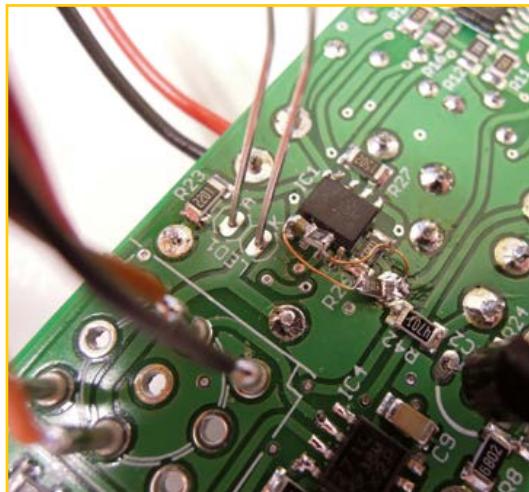
- [1] Atmel Studio 6.2 :
www.atmel.com/microsite/atmel_studio6
- [2] Arduino-IDE pour Visual Studio :
www.visualmicro.com
- [3] Atmel-ICE :
www.atmel.com/tools/atatmel-ice.aspx
Article Elektor :
www.elektor-magazine.fr/140275
- [4] SAM-ICE :
www.atmel.com/tools/atmelsam-ice.aspx

MMS : menhirs montés en surface

Ah les phases du prototypage ! Les heures passées le regard vide devant le programme de CAO... La petite hésitation au moment de commander enfin les composants et le circuit imprimé... Quelques jours après, le croisement de doigts au moment (suprême) de basculer le bouton de l'alimentation. Après la première mise sous tension du nouveau projet, l'étape suivante consiste souvent en un retour à la conception ou, si on a

broches 1 et 2 d'IC1 ainsi que la résistance de contre-réaction entre la broche 2 et la résistance série de $100\ \Omega$ à la sortie de la broche 1. Avec des composants traversants, ces modifications auraient été faciles. Persévérant, Ton a réussi avec des résistances 0805 et un condensateur 0603, ce dernier entre les broches du boîtier SOIC-8, et du fil de cuivre de seulement **0,2 mm** !

Thijs Beckers
(Elektor.Labs)



de la chance, à la soudure pour déloger soudures froides et courts-circuits sur ces fichus CMS.

Au labo d'Elektor, cette boucle c'est la routine : conception, test, correction (sauf coup de chance), nouveau test. Parfois il faut revoir complètement la conception de la carte, mais on commence tous par essayer de rafistoler le prototype à la main. Cela paraît plus facile, mais c'est trompeur. Je pense aux tribulations récentes de mon collègue Ton Giesberts sur un de ses protos (à gauche). La sortie d'IC1, double ampli op TLC272, oscillait. Pour y remédier, il a fallu mettre une résistance d'amortissement de $100\ \Omega$ en série avec la sortie (broche 1). Les broches 1 et 2 ont dû être dessoudées et relevées de leur pastille, mais ce n'était pas tout : regardez bien, vous apercevez le condensateur ajouté entre les

La photo de droite montre un superbe cas de *tombstoning** intentionnel. Au lieu d'utiliser un simple condensateur, une version avec haute résistance série (ESR, pour *Equivalent Series Resistance*) a été créée à l'aide d'un condensateur standard relié à deux résistances de $13\ \Omega$ en parallèle, pour obtenir une ESR de $6,5\ \Omega$. Un pont de fil, véritable petit dolmen, relie les composants placés à la verticale.

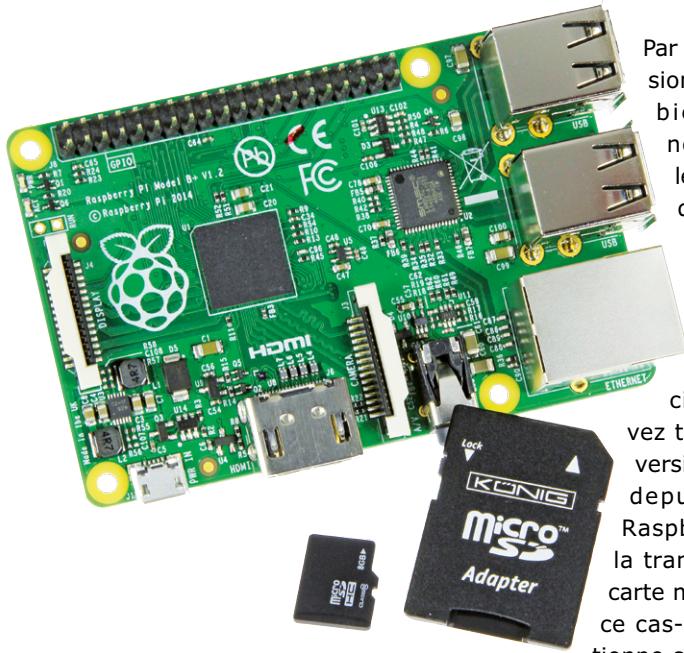
(140290 – version française : Kévin Petit)

* Les termes *tombstoning* (NdT : l'art de dresser des pierres tombales), *Stonehenging*, et effet *Manhattan* décrivent la tendance qu'ont les petits composants sans pattes à se dresser lors du soudage, tels des menhirs, sous l'effet de la tension superficielle de la soudure en fusion.

mise à niveau de la carte SD pour le RPi B+

Bert van Dam
(Pays-Bas)

Vous venez de déballer votre Raspberry Pi B+ et souhaitez utiliser la carte SD avec le logiciel de la version B dont vous êtes familier ; ce choix pose deux problèmes. Primo, pour la version B+ il faut une carte *micro-SD*. Deuzio, le logiciel d'origine n'est pas (entièrement) compatible avec le B+ ; tel quel, celui-ci ne démarra pas. Je vous propose une solution relativement facile.



Par rapport à la version B, le B+ s'est bien amélioré, notamment par le remplacement de certains des composants matériels, ce qui implique une adaptation du logiciel. Vous pouvez télécharger une version Linux à jour depuis le site du Raspberry-Pi [1] et la transférer sur une carte micro-SD ; dans ce cas-là, le B+ fonctionne sans le moindre problème. Les utilisateurs du RPi B qui possèdent une carte SD avec le logiciel système + certains logiciels additionnels qu'ils maîtrisent (p. ex. la carte SD accompagnant le livre d'Elektor « Raspberry Pi - 45 applications pour l'électronicien » [2]) . Il est facile de mettre à niveau la carte SD, sous la forme d'une version micro-SD, pour pouvoir l'utiliser, dans le B+.

La première étape, identique pour toutes les cartes SD utilisant *Debian Wheezy*, est la mise à niveau du système d'exploitation lui-même. La seconde consiste à modifier les fichiers et les programmes qui ne fonctionnent plus. Cette étape dépend des logiciels installés et diffère d'une carte SD à l'autre. Dans cet article, nous utilisons la carte SD accompagnant le livre Elektor [3]. Il nous faudra votre ancien Raspberry Pi

modèle B, sa carte SD, une connexion à l'internet et une bonne alimentation 5 V/1 A. Pour le nouveau B+ il nous faut une carte micro-SD de 4 Go (en classe 4 ou plus) avec adaptateur pour carte SD.

Préparatifs

Sous Windows, faites une image de votre carte SD à l'aide d'un lecteur de carte SD et du programme *DiskImager*. Ce dernier est intégré dans le téléchargement gratuit associé au présent article [4]. Insérez ensuite l'adaptateur de carte SD pourvu de la carte micro SD dans le lecteur et copiez l'image enregistrée sur la carte micro-SD en utilisant *DiskImager*.

Insérez maintenant l'adaptateur + carte micro SD dans le RPi B et connectez-le à l'internet. Utilisez un câble et pas une connexion WiFi, car il faut éviter à tout prix une coupure de la liaison au cours de la mise à niveau. Vous pouvez utiliser un clavier et une souris, mais il est plus facile de commander le Raspberry Pi depuis un PC sous Windows à l'aide du programme *Putty* [4]. De cette façon, vous n'avez rien à saisir à la main et vous pouvez simplement recopier les commandes proposées dans le téléchargement gratuit.

Lancez *Putty* et saisissez l'adresse IP de votre Raspberry Pi. Si vous ne la connaissez pas, connectez-vous à votre routeur qui vous donnera la liste de tous les périphériques connectés, Raspberry Pi compris. Assurez-vous que le port 22 a été sélectionné et que l'option SSH est cochée. Cliquez ensuite sur *Open*. Si *Putty* vous informe que cette adresse IP n'est pas encore utilisée, cliquez sur *OK*. Très vite, vous serez connecté au Raspberry Pi qui vous demande votre nom et

votre mot de passe. Les données de connexion par défaut sont :

```
nom pi
mot de passe raspberry
```

Attention : Lors de la saisie du mot de passe, vous ne verrez rien, pas même des astérisques ! On croirait qu'il ne se passe rien, mais en fait ça marche !

Conversion - étape 1

Cette étape est commune à toutes les cartes SD utilisant la distribution *Debian Wheezy*. Connectez-vous au Raspberry Pi avec *Putty* et saisissez la commande suivante :

```
sudo apt-get update
```

Vous pouvez copier cette commande (de même que toutes les autres utilisées dans cet article) depuis le document « commands.txt » du fichier à télécharger [4] et la coller avec le bouton droit de la souris dans la fenêtre de *Putty*. Appuyez ensuite sur *Enter*.

Cette commande réactualise dans le Raspberry Pi les tableaux qui lui indiquent où trouver certains fichiers et programmes sur la Toile. Cette opération peut durer quelques minutes. Puis saisissez la commande suivante pour effectuer la mise à niveau proprement dite :

```
sudo apt-get upgrade
```

Répondez à chacune des questions par « *y* » (*Yes*). **Attention** : Cette étape peut prendre 45 minutes ou plus et requiert une alimentation assez puissante (1 A). À défaut, le processus risque de s'interrompre intempestivement.

Relancez ensuite le Raspberry Pi par la commande :

```
sudo reboot
```

Si vous travaillez sous *Putty*, la liaison s'interrompt à ce moment-là. Attendez que le Raspberry Pi ait démarré et reconnectez-vous.

La conversion est terminée et la carte micro SD est maintenant utilisable avec le Raspberry Pi B+. De nombreux logiciels existants fonctionneront, mais pas tous (cf. les modifications décrites ci-après). Pour savoir lesquels modifier, testez bien tous vos anciens programmes installés. Le

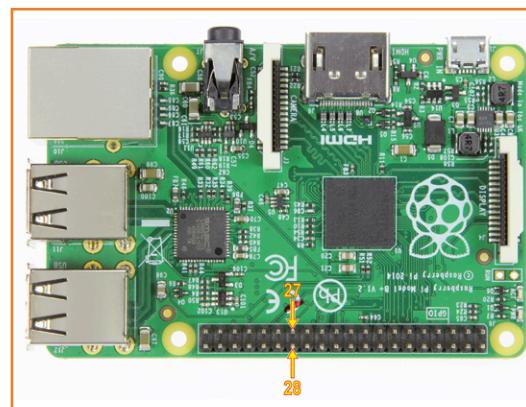


Figure 1.
Le connecteur GPIO du B+ s'est rallongé de deux broches qui pourraient gêner l'insertion de certains connecteurs à 26 broches.

détail de ce qu'il vous faudra faire dépend du logiciel à modifier.

Conversion - étape 2

À titre d'exemple, parcourons ensemble les étapes de l'adaptation de la carte micro-SD du livre *Raspberry Pi - 45 applications pour l'électronicien*. Ce que je ne mentionne pas fonctionne sans modification et le code source du livre se trouve toujours sur la carte micro-SD.

1. Broches additionnelles

Si vous utilisez des connecteurs (normaux ou à pression) pour raccorder des montages électroniques au Raspberry Pi B+, vous ne pourrez peut-être plus le faire, en raison de l'extension des broches GPIO. Les 26 premières n'ont pas changé, mais les deux broches supplémentaires (**fig. 1**) peuvent gêner (en fonction du type de connecteur utilisé, certains modèles à broches soudables s'y connectent sans problème). Il s'agit de ID_SD (broche 27) et de ID_SC (broche 28), destinées exclusivement à l'EEPROM *Pi-Plate* I²C ID ; elles rendent possible la détection automatique d'une *Pi-Plate* (module d'extension) reliée au système. Si vous n'avez pas l'intention d'utiliser une *Pi-Plate*, vous pouvez écarter ces deux broches, voire les supprimer en les fraisant. Une solution moins radicale consiste à utiliser des connecteurs adaptés.

2. Clavier

La mise à niveau a pour conséquence un éacement des paramètres du clavier. La commande :

```
sudo raspi-config
```

vous permet d'appeler les paramètres d'écran et, sous l'onglet *internationalisation options*, de

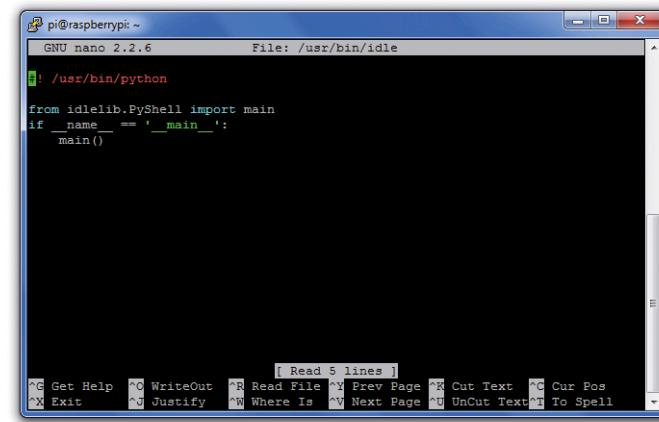


Figure 2. Le fichier (erroné) Idle ouvert dans Putty. On en efface le texte et on le remplace par celui du listing 1.

Listing 1.

```

#!/usr/bin/env python

# Launch IdleX
import sys

def show_error():
    if sys.version < '3':
        import Tkinter as tk
        import tkMessageBox as messagebox
    else:
        import tkinter as tk
        import tkinter.messagebox as messagebox

    root = tk.Tk()
    root.withdraw()
    messagebox.showerror(title='IdleX Error',
                         message=('Impossible de trouver «idlelib».\n' +
                                  'Placez-le dans le répertoire' +
                                  'de «idlelib» ou exécutez setup.py pour' +
                                  'installer IdleX.\n' +
                                  '    python setup.py install --user'))
try:
    import idlelib
except ImportError:
    show_error()
    sys.exit(-1)

from idlelib.idlexMain import main
main()

```

sélectionner le clavier que vous utilisez. Le nouveau clavier ne sera actif qu'après redémarrage du Raspberry Pi (par sudo reboot).

3. Sons

Après la mise à niveau, il se peut que la fonction de mélangeur *amixer* se trouve sur *mute* (silencieux), vous n'entendrez donc rien. Désactivez le silencieux par la commande :

```
amixer set PCM unmute
```

4. IdleX

La mise à niveau écrase le fichier Idle existant de sorte que ce n'est pas IdleX mais Idle qui est lancé. Ouvrez l'éditeur Nano avec la commande suivante :

```
sudo nano /usr/bin/idle
```

Le curseur se trouve en début de fichier (**fig. 2**). Maintenez enfoncée la touche Del jusqu'à ce que tout le contenu soit effacé. Copiez le contenu du **listing 1** (disponible aussi dans le téléchargement gratuit) et collez-le dans l'éditeur à l'aide du bouton droit de la souris. Si vous ne travaillez pas avec un PC et Putty, mais directement par écran et clavier, il vous faut saisir ce texte *sans faire de faute*. Pour le défilement, préférez les touches fléchées à la souris.

Enregistrez le fichier par Ctrl-O, Enter, Ctrl-X.

Terminé !

Votre nouvelle carte micro-SD pour le Raspberry Pi B+ est maintenant prête à servir. Éteignez le Raspberry Pi B par la commande :

```
sudo shutdown -h now
```

Sortez la carte micro-SD de l'adaptateur, placez-la dans le Raspberry Pi B+, mettez l'alimentation en fonction et, c'est parti mon Kiki !

(140342 – version française : Guy Raedersdorf)

Liens

- [1] www.raspberrypi.org/downloads/
- [2] www.elektor.fr/rpi
- [3] www.elektor.fr/129025-81
- [4] www.elektor-magazine.fr/140342

trouvez un nouveau boulot en postant sur Elektor.Labs

Clemens Valens
(Elektor.Labs)



Elektor.Labs Preferred Parts (ELPP)

Dans le numéro de sept. 2014 d'Elektor, nous avons lancé la liste de composants **Passe-Partout** d'**Elektor LABS** ou **Elektor.Labs Preferred Parts** (ELPP), une bibliothèque de composants de base, utilisés par la plupart de circuits. Pour vous faciliter la tâche, ils sont associés à une empreinte validée pour leur implantation sur circuit imprimé et à un numéro de référence pour vos commandes. ELPP comporte aussi des composants électromécaniques (connecteurs,

Get Elektorized!

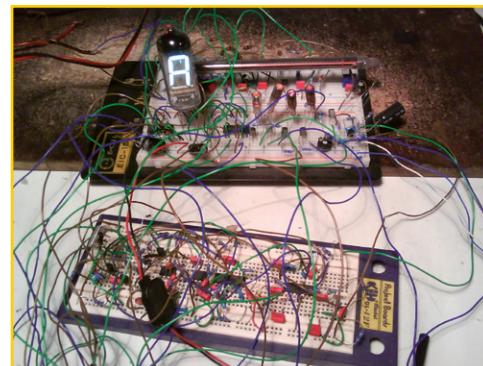
Vous souvenez-vous d'avoir vu ça quelque part ? Oui, au resto italien, bien sûr. Mais ces spaghetti-là, vous vous souvenez peut-être de les avoir vus *aussi* sur notre site communautaire Elektor.Labs. Une personne qui s'en souviendra, c'est **lux36**, l'auteur de cette magnifique pièce démontée lui-même. Et pour cause, sa publication sur Elektor.Labs lui a valu d'être recruté pour un nouveau travail. Comme il a eu la bonne idée de m'en informer, j'en profite pour partager la nouvelle avec toute la communauté. **Félicitations, lux36 ! J'espère que votre nouveau patron ne verra pas d'inconvénient à ce que vous publiez sur Elektor.Labs.** Les bonnes nouvelles de ce genre sont comme les trains qui arrivent à l'heure, on n'en parle moins volontiers que des expériences frustrantes.

La vie de **lux36** a changé grâce à ce projet publié sur Elektor.Labs, maintenant il gagne (encore) mieux sa vie dans son nouvel emploi, mais ce n'est pas fini. Comme il a posté d'autres projets, ceux-là pourraient aussi lui rapporter de l'argent s'ils sont publiés dans le magazine Elektor. Mais pour cela, il faudra qu'il nous donne d'abord plus de détails ; pour l'instant, sur le site, il manque les schémas... et le dessin de circuit imprimé paraît inachevé. **Lux36** (et tous les autres membres actifs), si vous voulez que vos nous fassions quelque chose de vos propositions, n'oubliez pas de nous fournir un dossier cohérent. www.elektor-labs.com/node/3169

ELPP est hébergée désormais par GitHub

relais...). L'accueil favorable que vous lui avez fait confirme que notre tentative de rationalisation est bonne pour nous tous.

Pour en garantir l'accès confortable à tout le monde, nous avons déménagé sur **GitHub** la bibliothèque ELPP publiée initialement sur notre serveur FTP. Cette plateforme à la réputation bien établie présente, outre son accessibilité, l'avantage d'un système de gestion des versions, qui enregistre aussi l'historique des modifications. La dernière en date est l'adjonction de la version d'ELPP pour la bibliothèque de *DesignSpark PCB*. <https://github.com/ElektronLabs/PreferredParts>



Les spaghetti, ça peut mener loin.

(140305 – version française : Remi Descistes)

convertisseur USB ⇄ 4 UART RS-232 et RS-422/RS-485 & concentrateur pour 3 ports USB sans microcontrôleur ni logiciel d'aucune sorte

Sébastien
Guerreiro de Brito
(Montrieu-en-Sologne)

Depuis la disparition du port série DB9 sur les PC modernes, il devient compliqué de raccorder d'anciennes applications munies d'une liaison RS-232. Certes, il existe des solutions de fortune appréciables, mais bridées par des inconvénients gênants notamment quand se multiplient les appareils à desservir. C'est ce qui m'a incité à étudier ce convertisseur universel puis à vous le proposer, avec ses deux ports RS-232 et 2 ports RS-422.

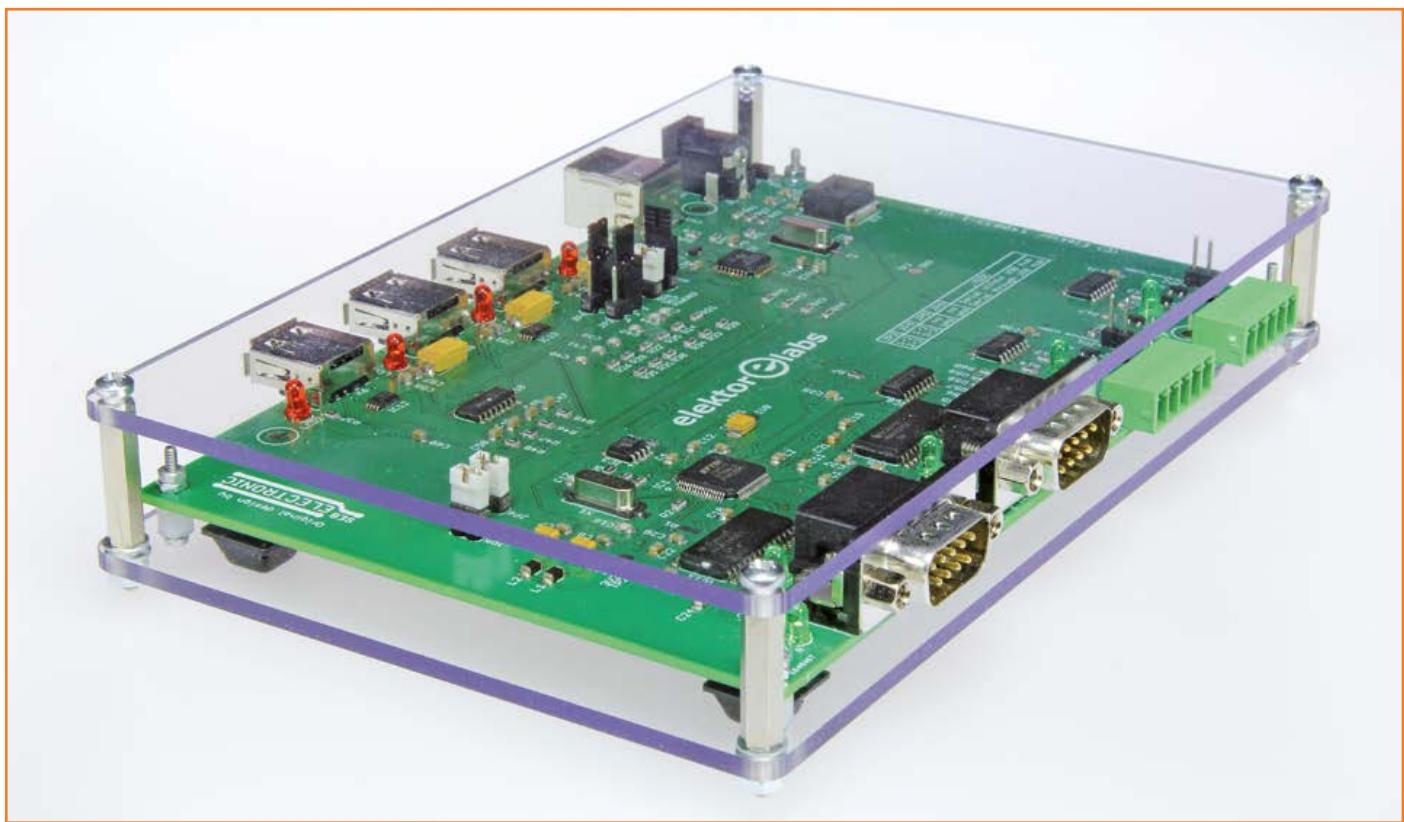
Oui, les convertisseurs USB<->RS-232 sont bien pratiques, mais si on doit utiliser plusieurs cartes équipées de connecteurs RS-232, il faut autant de convertisseurs. Et si en plus l'interface est une vraie liaison différentielle RS-422/RS-485, il n'est plus aussi simple de trouver un convertisseur adéquat. L'idée m'est donc venue de développer un convertisseur qui gère simultanément plusieurs liaisons RS-232 et RS-422/RS-485. Comme souvent, la recherche d'un boîtier pour recevoir le montage en cours d'étude est une étape importante, surtout quand il y a beaucoup de connecteurs à caser. C'est même le choix du boîtier qui dans une large mesure fixe les dimensions du circuit. Après en avoir trouvé un qui soit bien adapté et me permette de placer confortablement les différents connecteurs, je me suis aperçu que mon circuit d'origine n'occupait que 50 % de la place disponible. Que faire de l'espace libre ? Compléter le montage par un concentrateur USB (*hub*) !

Deux étapes

Comme j'ai conçu ce circuit en deux étapes -- d'abord le convertisseur puis l'ajout de la partie HUB -- il est logique de le retrouver sous la forme

de deux blocs distincts sur le synoptique (fig. 1). Le schéma détaillé apparaît sur la figure 2 ; le va-et-vient de l'une à l'autre vous facilitera la lecture. Commençons par l'**alimentation du circuit**. Elle provient de deux sources distinctes, soit l'interface USB (K1), en provenance du PC hôte, soit un bloc secteur externe (K5) (5 V). En fonction de la source choisie, un cavalier doit être implanté. C'est JP8, nous y reviendrons ci-dessous. Quand on utilise une alimentation externe, le circuit est protégé contre les inversions de polarité par la diode schottky D9. Le régulateur LM2937ES-3.3 (IC2) assure la conversion des 5 V en 3,3 V pour conformer la tension d'alimentation à la tension de service des circuits.

La partie concentrateur (**HUB USB** sur la fig. 1) de mon interface repose essentiellement sur IC9, un TUSB2046BVF de *Texas Instruments*. Avec le commutateur de puissance IC10, il gère automatiquement la distribution des alimentations. À partir d'un unique port d'entrée (K1), il gère jusqu'à quatre ports USB en sortie. Basé sur une machine d'état, ce composant n'embarque aucun logiciel et ne nécessite aucun paramétrage pour assurer la commande des alimentations et la gestion des



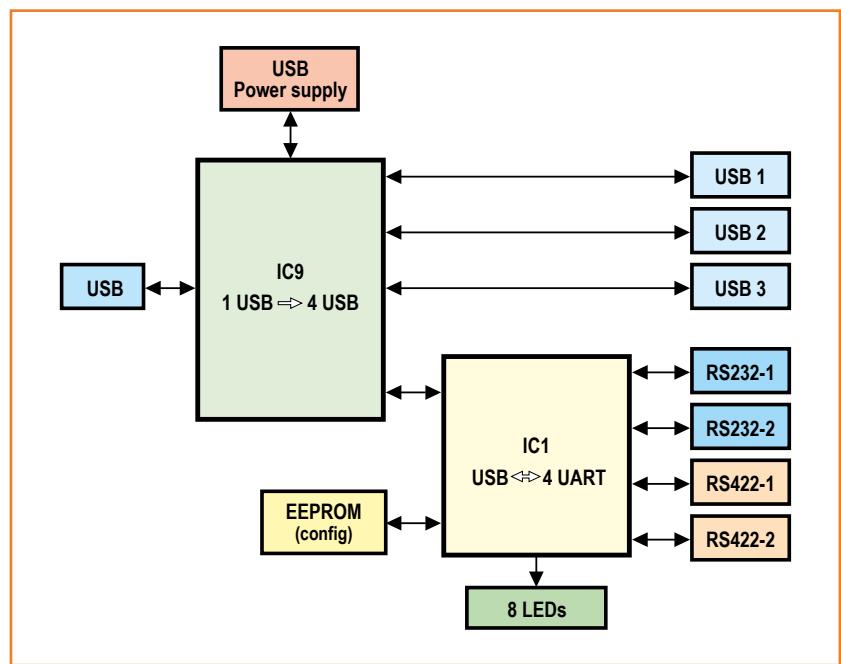
ports USB. Pas besoin de microcontrôleur ! C'est IC10 qui pilote la **distribution du courant d'alimentation** aux appareils USB (ainsi qu'à la partie UART de ce circuit). Le TPS2044 délivre jusqu'à 500 mA par voie. Non seulement il limite le courant pour que son intensité ne dépasse jamais ce seuil, mais il vérifie aussi sa propre température au niveau du transistor interne. Ses sorties de défaut ($\overline{OC_n}$) sont à drain ouvert, c'est pourquoi les résistances de polarisation haute R45 à R48 en fixent le potentiel. Les réseaux RC sur les voies $\overline{OC_n}$ permettent de filtrer les erreurs de détection de surconsommation liées aux vifs appels de courant susceptibles de se produire lors de la commutation de l'alimentation de la voie.

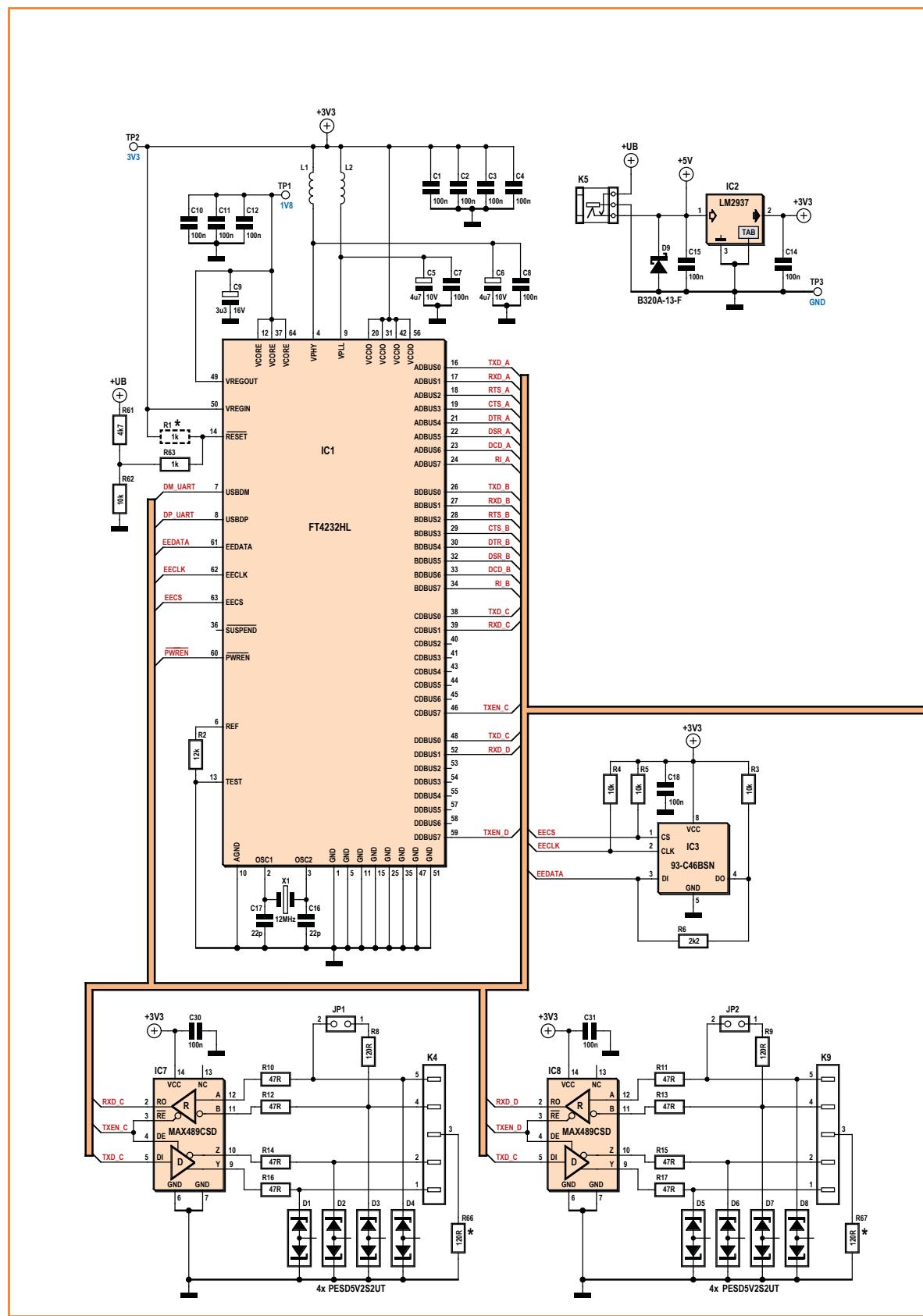
Les LED1 à 5 rendent compte de l'activation des périphériques USB, la LED1 pour le concentrateur USB et les LED2 à LED4 pour les quatre éléments sur le concentrateur.

Toutes les entrées ou sorties de ce circuit peuvent être soumises à des perturbations transitoires. Si elles sont d'amplitude et de durée suffisantes, celles-ci peuvent endommager le concentrateur

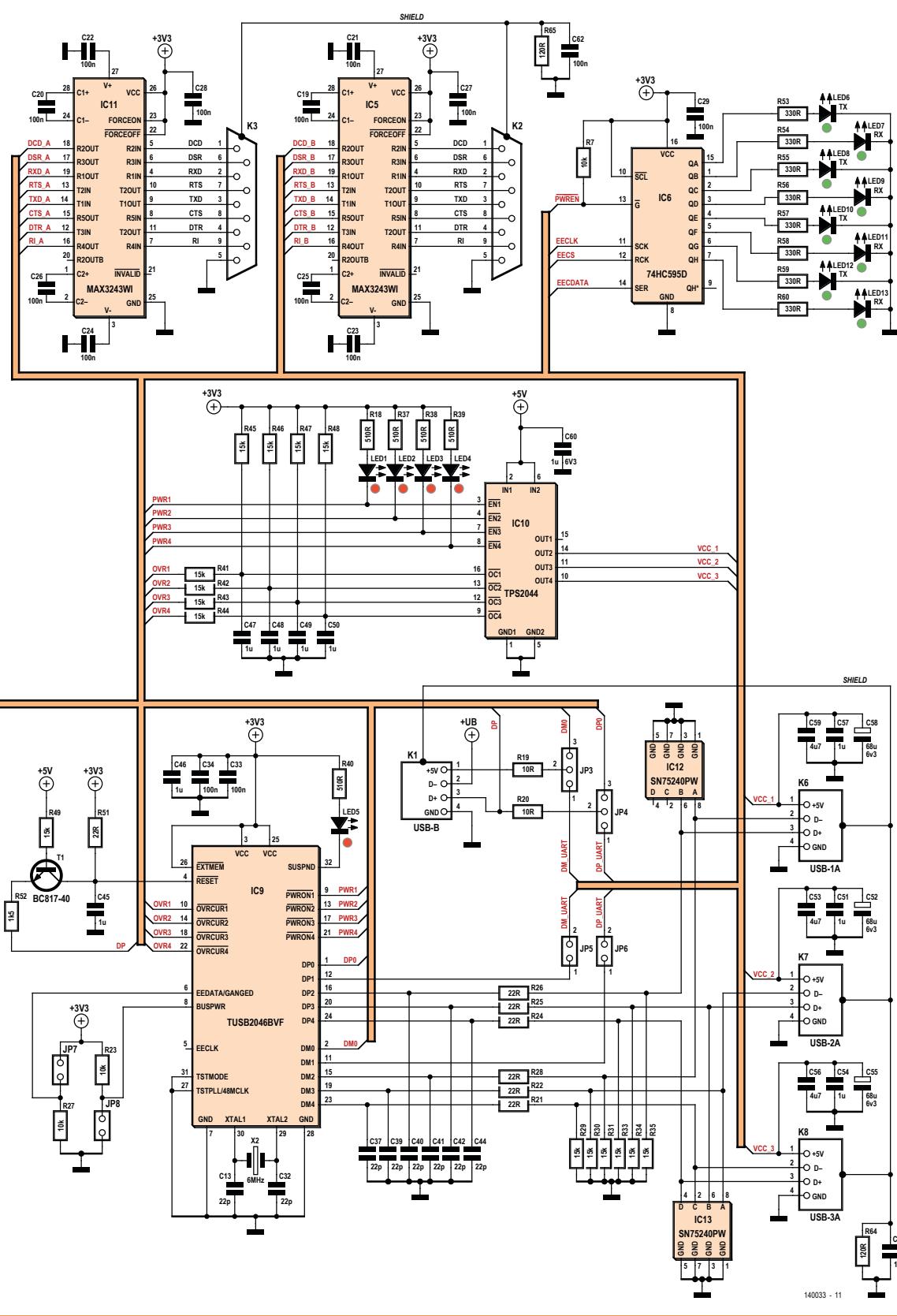
USB ou le convertisseur USB-UART. Pour contrer ces perturbations, le circuit utilise des protections spécialisées SN75240PW de TI (IC12, IC13).

Figure 1.
Synoptique du convertisseur/concentrateur





convertisseur USB ⇄ 4 UART



Attention : ces suppresseurs de parasites introduisent une capacité défavorable à l'utilisation de ce convertisseur pour les applications USB 2.0 à vitesse élevée.

UART

Le bloc UART de la figure 1, c'est IC1 sur la **figure 2**, un FT4232HL de FTDI, qui assure la conversion USB 2.0 (480 Mb/s) vers quatre UART (*Universal Asynchronous Receiver Transmitter*). Dans notre cas, deux UART seront utilisés pour les liaisons RS-232 et les deux autres configurés en liaisons RS-422/485 (nous y reviendrons ci-dessous au sujet du paramétrage).

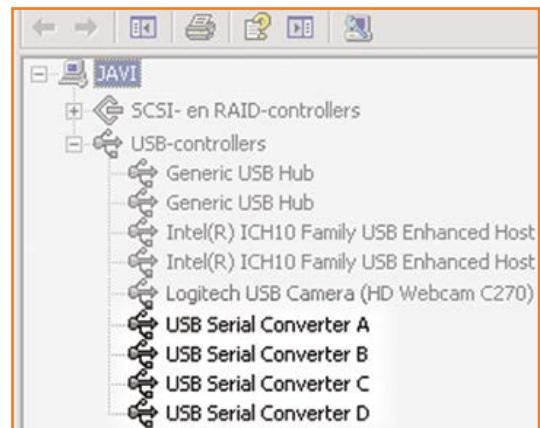
La configuration choisie est sauvegardée dans une EEPROM (IC3).

Afin de visualiser les signaux entrant et sortant sur les liaisons série, les LED6 à LED13 sont connectées sur IC6, un registre à décalage de 8 bits, à chargement série et à sortie parallèle, interfacé directement avec le convertisseur IC1. Pour éviter que les LED ne s'allument intempestivement, le signal PWREN émis par IC1 est utilisé pour désactiver les sorties parallèles d'IC6 tant que le convertisseur USB n'a pas été reconnu et configuré par le pilote du PC.

Pour convertir en signaux RS-232 les signaux de type TTL issus d'IC1, j'utilise deux convertisseurs MAX3243EIDW (IC11 et IC5). Ils assurent la conversion de l'ensemble des signaux utiles sur une liaison RS-232 (*full-duplex*) (K2 et K3 en haut à droite de la fig. 2). Ce composant est conçu pour assurer aussi une protection contre les décharges électrostatiques de $+/-15\text{ kV}$ dans l'air et de $+/-8\text{ kV}$ pour les décharges électrostatiques au contact.

Le réseau formé de R65 et C62 relie le blindage (SHIELD) du connecteur à la masse de notre circuit tout en apportant un filtrage d'un point de vue CEM (cette fonction est également utilisée pour la protection sur le blindage du connecteur USB).

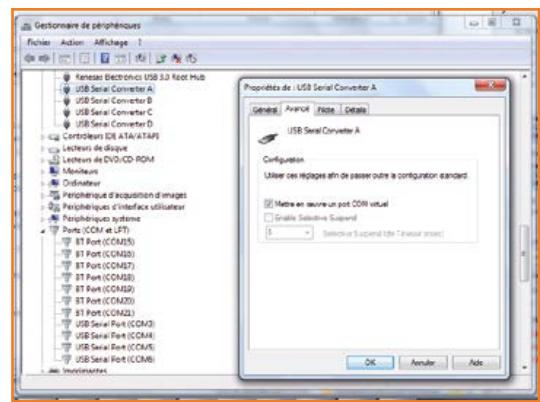
Pour la liaison RS-422/RS-485 (K4 et K9, en bas à gauche), j'utilise des convertisseurs MAX-489CSD+ (IC7 et IC8). Leurs sorties sont protégées contre les décharges ESD par D1 à D8. Il reste à fixer la résistance de charge ; ce sont les cavaliers JP1 et JP2 qui chargent le récepteur avec la résistance caractéristique du câble : 120 Ω (voir ci-dessous l'encadré sur la résistance de terminaison).



Lors du branchement du circuit, le système d'exploitation du système hôte va vous demander d'installer les pilotes (*drivers*) [1]. Une fois ces pilotes installés, vous disposerez sur votre machine de 4 ports COM (port COM virtuels). Les deux premiers sont les liaisons RS-232 et les deux autres les liaisons RS-422/RS-485.

Le circuit a été testé sous Windows et Linux. Et il a été adopté aussitôt par notre labo où il restera en service pour la plus grande satisfaction de tous, car il manque toujours une interface RS-232 ou RS-485 quelque part.

Note : Pour le PC, un *USB Serial Converter* n'est pas un port série. Pour le mettre en port série, le plus simple est d'ouvrir ses propriétés dans *Device Manager* et, dans les paramètres avancés, de cocher la case VCP ou *Virtual Com Port*. Répéter l'opération pour chaque *USB Serial Converter*. Ensuite il faut débrancher/rebrancher la carte pour inciter *Windows* à énumérer les ports série. Cela peut prendre 5 min ou plus... après, on se retrouve avec quatre ports série (cf copie d'écran).



La carte assemblée et testée, prête à l'emploi, est disponible dans l'e-choppe www.elektor.fr/USB-serial-hub

Assemblage et mécanique

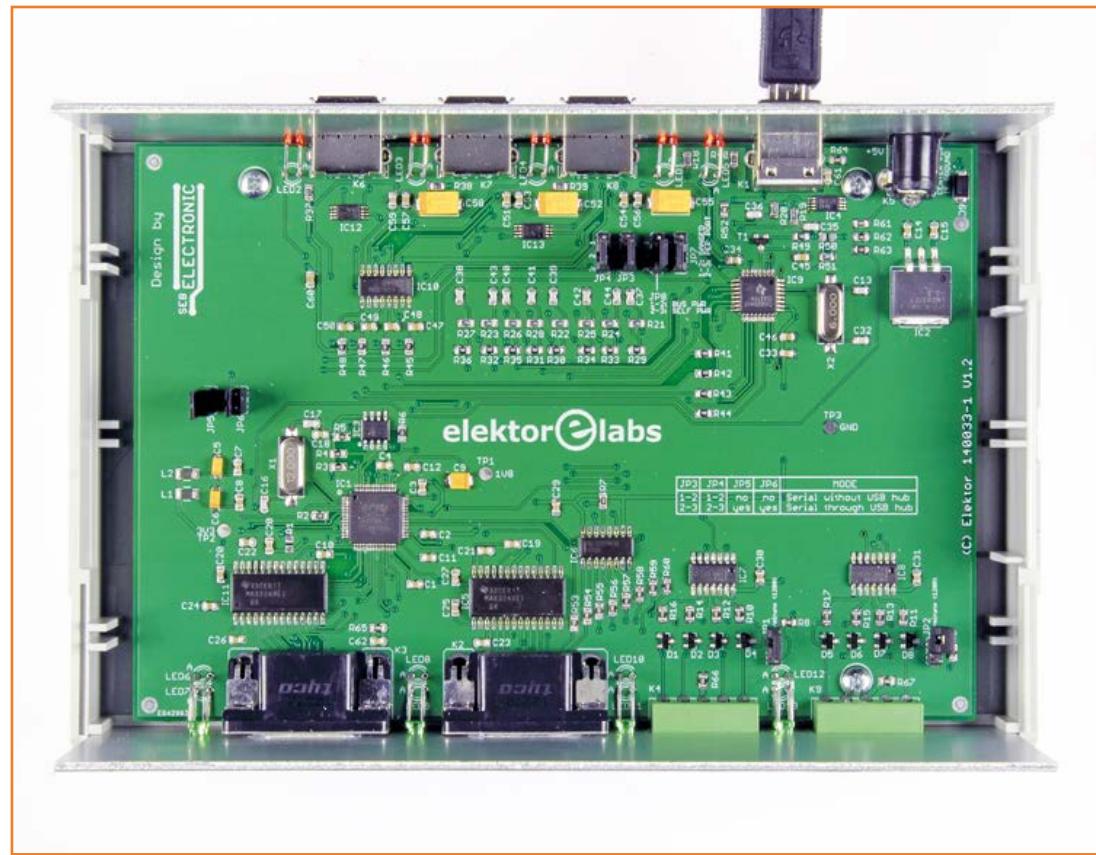
Avec le dessin de circuit imprimé (**fig. 3**), la photographie du prototype et la liste des composants, la construction de cet appareil ne devrait pas poser de problème particulier, à condition de disposer de l'outillage approprié pour l'assemblage de composants montés en surface et d'une solide expérience en la matière, surtout pour IC1, mais aussi IC9, IC12 et IC13 dont les broches sont serrées. Si vous vous lancez, ne négligez pas les menus détails comme p. ex. la polarité des condensateurs au tantalum (C5, C6, C9, C52, C55 et C58). Donnez-vous la peine d'en consulter la fiche technique !

Ne montez pas R1. C'est un reliquat d'une version antérieure du schéma.

Pour l'alignement des LED vertes superposées en trois paires, il faudra s'appliquer afin que le coulage de leurs pattes soit aussi rigoureusement que possible le même.

Compte tenu de l'intérêt présenté par cette réalisation pour tout électronicien qui a encore à sa disposition d'excellents appareils, en parfait état de marche, mais qui n'ont comme seul « défaut » que celui de ne communiquer que par RS-232 ou RS-422/RS-485, Elektor estime opportun d'en proposer une version assemblée et prête à l'emploi (140033-91). Profitez vous aussi du confort que vous garantit *elektorPCBservice* [2].

La carte assemblée par vos soins ou achetée prête à l'emploi se monte dans un coffret, par un exemple un modèle de VERO, pour lequel les



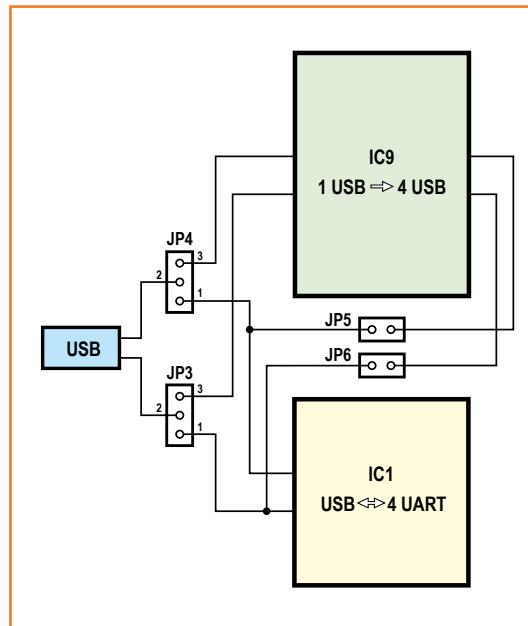


Figure 4.
Configuration des cavaliers
JP3, 4, 5 et 6.

plans de découpe des façades sont disponibles [2] au format DXF ou au format FPD (format natif Schaeffer [3] dont je suis un client satisfait). Le connecteur d'alimentation K5 (dont on remarquera au passage que c'est un modèle avec interrupteur incorporé) n'existe pas dans la version initiale de mes plans de découpe, je l'ai rajouté à la dernière minute pour cette version publiée, qui n'a été vérifiée que par simulation : le trou correspond parfaitement, mais je ne garantis rien.

Si vous montez le circuit imprimé dans le coffret VERO mentionné dans la liste des composants, vous remarquerez l'absence de trou de fixation près de K3 ! Pour obtenir une fixation robuste, utilisez les colonnettes des connecteurs DB9 pour les rendre solidaires de la face avant.

Attention, la broche centrale du connecteur d'alimentation est à la masse !

Liste des composants

Résistances (SMT 0805) :

R19, R20 = 10 Ω
R21, R22, R24, R25, R26, R28, R51 = 22 Ω
R10, R11, R12, R13, R14, R15, R16, R17 = 47 Ω
R8, R9, R64, R65, R66, R67 = 120 Ω
R53, R54, R55, R56, R57, R58, R59, R60 = 330 Ω
R18, R37, R38, R39, R40 = 510 Ω
R23, R27 = 10 kΩ
R1*, R63 = 1 kΩ
R52 = 1,5 kΩ
R6 = 2,2 kΩ
R61 = 4,7 kΩ
R3, R4, R5, R7, R62 = 10 kΩ
R2 = 12 kΩ 1%
R29, R30, R31, R33, R34, R35, R41, R42, R43, R44,
R45, R46, R47, R48, R49 = 15 kΩ
* ne pas implanter

Condensateurs :

(SMT 0805 sauf mention contraire)
C13, C16, C17, C32, C37, C39, C40, C41, C42, C44 = 22 pF
C1 à C4, C7, C8, C10 à C12, C14, C15, C18 à C31,
C33, C34, C61, C62 = 100 nF
C45 à C51, C54, C57, C60 = 1 µF
C9 = 3,3 µF 16V, tant. (E)
C53, C56, C59 = 4,7 µF
C5, C6 = 4,7 µF 10V, tant. (A)
C52, C55, C58 = 68 µF 6,3 V, tant. (D)

Selfs :

L1, L2 = ferrite, 70 Ω @ 100 MHz, SMT 1206

Semi-conducteurs :

D1 à D8 = PESD5V2S2UT
D9 = B320A-13-F
IC1 = FT4232HL
IC2 = LM2937ES-3.3

IC3 = 93LC46B-I/SN

IC12, IC13 = SN75240PWR

IC5, IC11 = MAX3243EIDW

IC6 = 74HC595D

IC7, IC8 = MAX489CSD

IC9 = TUSB2046BVF

IC10 = TPS2044BD

LED6, LED7, LED8, LED9, LED10, LED11, LED12,

LED13 = LED verte 3 mm

LED1, LED2, LED3, LED4, LED5 = LED rouge 3 mm

T1 = BC817-40

Divers :

JP1, JP2, JP5, JP6, JP7, JP8 = embase à 2 picots
Pour JP1 et JP2 : résistance de terminaison de 120 Ω pour RS-485 (optionnel - voir encadré)
JP3, JP4 = embase à 3 picots
cavalier au pas de 2,54 mm for JP1-8
K1 = embase USB type B coudée, encartable
K2, K3 = connecteur sub-D, 9 broches, mâle
K4, K9 = bornier mâle encartable coudé à 5 broches, pas de 3,81 mm
K5 = fiche d'alimentation cylindrique 12 V/3 A avec interrupteur, broche de 1,95 mm
K6, K7, K8 = embase USB type A coudée, encartable
X1 = quartz 12 MHz CMS
X2 = quartz 6 MHz CMS
coffret 65x180x120 mm p. ex. VERO 75-265742
circuit imprimé 140033-1-v1.2
module assemblé prêt à l'emploi 140033-91 [2]

Note : les trous dans la liste des composants (comme IC4, R50...) ne sont pas des oubli, mais des composants devenus inutiles sans que leur suppression ait donné lieu à une renumérotation.

Figure 3.
Circuit imprimé du convertisseur-concentrateur. Selon le type de coffret utilisé, les LED rouges et les LED vertes doivent être coudées pour qu'on les voie bien une fois la carte mise en boîte.

Configuration, paramétrage et pilotes

Pour la mise au point de la partie UART (il faut en effet configurer le convertisseur USB pour fonctionner avec une liaison RS-485, nous y reviendrons) **sans** la partie concentrateur USB, un ensemble de cavaliers (**fig. 4**) permettent d'isoler les deux blocs. Pour isoler la partie UART de la partie concentrateur USB, il suffit :

- de placer JP4 entre 1 et 2
- de placer JP3 entre 1 et 2
- de retirer JP5 et JP6.

En mode de fonctionnement normal, les cavaliers doivent être positionnés comme suit :

- JP4 entre 2 et 3
- JP3 entre 2 et 3
- JP5 en position
- JP6 en position

Cavaliers			
JP	fonction		
1	$R_t = 120 \Omega^*$		
2	$R_t = 120 \Omega^*$		
3	1-2 :		2-3 :
4	1-2 :	serial without hub	2-3 : 1 : serial through hub
5	0 :		1 :
6	0 :		
7	1 : <i>all outputs switched</i>	0 : <i>single output switched</i>	
8	1 : <i>external power</i>	0 : <i>bus powered</i>	

1 = présent | 0 = absent | * (option, cf texte)

Deux autres cavaliers sont utilisés pour configurer le comportement d'IC9 :

- JP7 détermine la réaction du composant à une information de surconsommation sur un des ports : si l'entrée GANGED (broche 6)

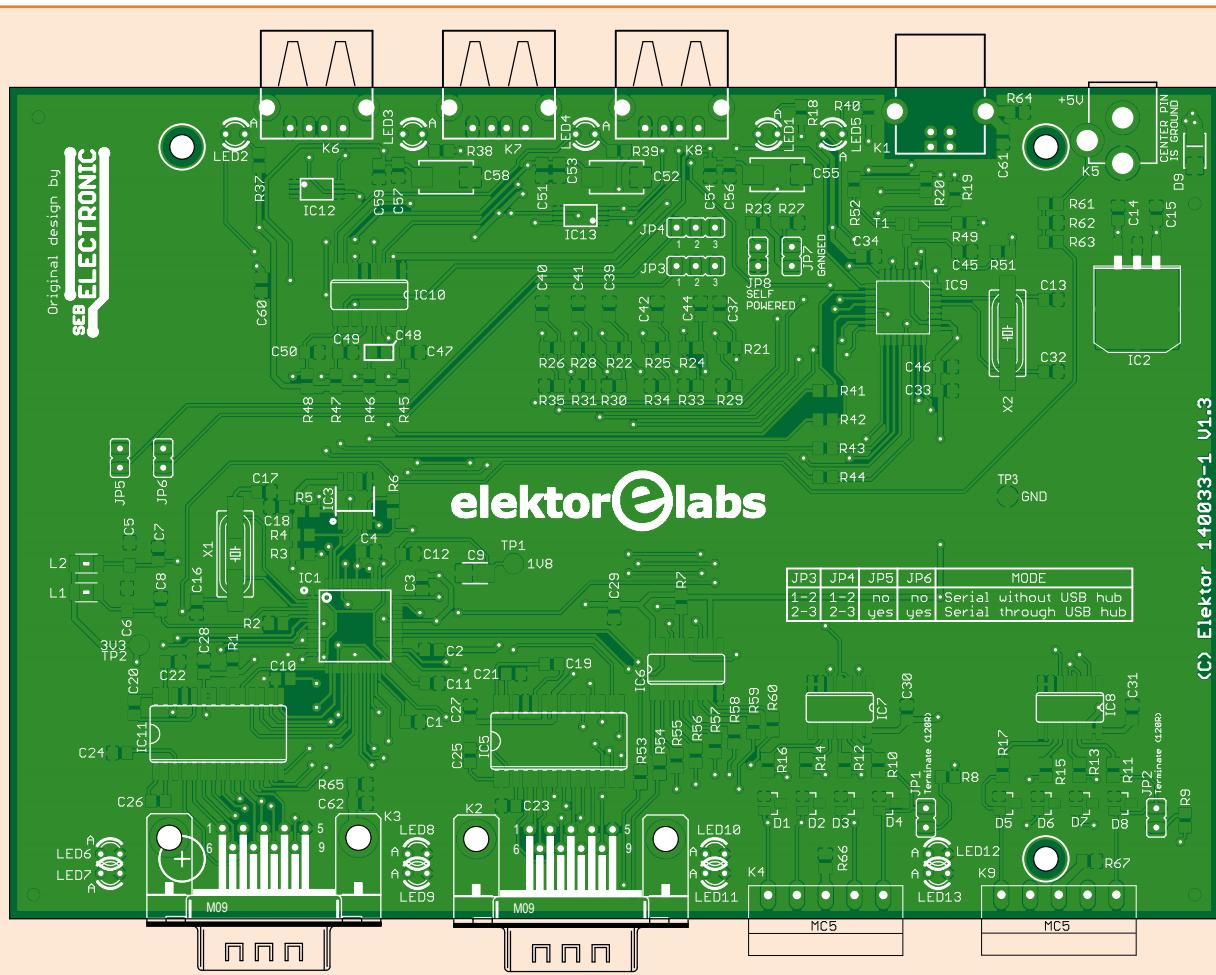




Figure 5.
Le convertisseur dans
son boîtier avec la façade
dessinée par l'auteur.

est à 3,3 V (le cavalier est placé), toutes les sorties sont coupées en cas de détection de surconsommation sur une des voies. Si l'entrée GANGED est à 0 (le cavalier n'est pas placé), seule la sortie en surconsommation sera coupée.

- JP8 signale au composant la source d'alimentation. Si JP8 est omis, BUSPWR est à 3,3 V, on est sur une alimentation par le bus USB. Si le cavalier JP8 est en place, on est sur une alimentation extérieure.

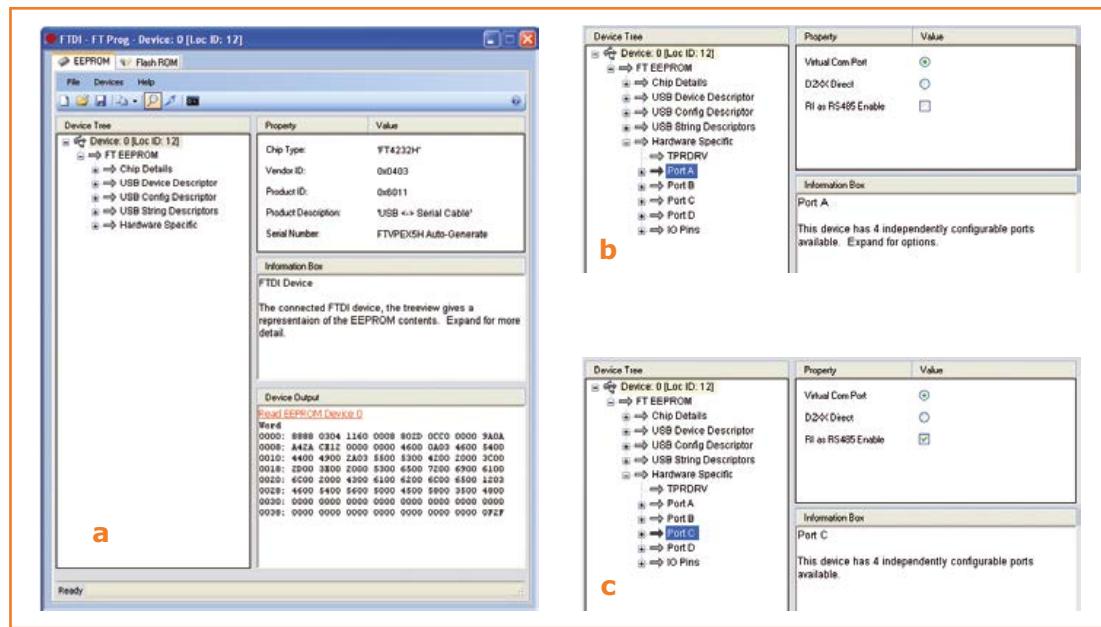


Figure 6.
Trois copies d'écran de
la mise en route et du
paramétrage à l'aide du
logiciel FTPROG de FTDI.

Les liaisons différentielles RS-422/RS-485 (rappel)

Le bus RS-422 est défini dans la norme EIA-422-B-1994 comme un bus simplex, c'est-à-dire qu'il n'y a qu'un seul émetteur à la fois, qui peut piloter jusqu'à 10 UL (Unit Load ou charges unitaires). L'acheminement des signaux est différentiel afin d'obtenir une grande robustesse de transmission des données sur de longues distances, même dans un environnement bruité.

Débit (Baudrate) contre portée

L'émetteur est capable d'émettre des données sur une distance d'env. 1 200 m (4000 pieds), mais la portée croît au détriment du débit. La norme spécifie que le débit peut atteindre 10 Mbps mais sur de faibles distances (**fig. 7**).

La résistance de terminaison

En raison de la différence d'impédance entre le câble et le récepteur, un phénomène de réflexion de l'onde émise peut se produire sur les lignes lors de transmissions sur de longues distances ou à débits élevés. Pour pallier ce phénomène, une résistance de terminaison est nécessaire (sur JP1 et JP2) pour adapter la charge à la résistance caractéristique du câble (typiquement $120\ \Omega$).

D'autres modes de terminaison existent, mais comme ils sont moins courants, je ne les présenterai pas ici.

La liaison RS-485

Le bus RS-485 (**fig. 8**) est basé sur le bus RS-422 avec certaines fonctions supplémentaires comme l'augmentation du nombre d'UL à 32 maximum. Le bus RS-485 peut supporter la configuration multi-point. Cette configuration permet d'avoir un émetteur communiquant à plusieurs récepteurs. C'est la configuration de base de ce bus, avec une communication semi-duplex (deux fils pour les données). En semi-duplex, les récepteurs doivent être en mesure de reconnaître que la trame en cours de transmission leur est destinée. Cela implique qu'un système d'adressage permette d'identifier le récepteur. On peut aussi envoyer des commandes en broadcast en envoyant une trame à destination d'un récepteur virtuel ayant une adresse réservée (par exemple 255). Ainsi tous les récepteurs prennent en considération la trame reçue.

Sinon on peut évidemment envoyer une commande à un récepteur particulier en mentionnant son adresse dans la trame, mais il y a là un inconvénient majeur : les récepteurs ne peuvent pas prendre la ligne pour envoyer des informations sans en avoir expressément reçu l'ordre (notion de maître-esclave).

La pratique veut que dans ce type de communication, le maître émette et que les esclaves (récepteurs) écoutent. Quand le maître a demandé des informations précises à un esclave, celui-ci et lui seul répond. Si le maître a envoyé des informations en broadcast (à tous les récepteurs), les récepteurs ne doivent pas répondre.

Contrairement à une croyance répandue, pour communiquer, le bus RS-485 ne nécessite pas deux mais bel et bien trois fils : Data+, Data – et la masse !

Unit Load

La norme précise que pour un émetteur donné, on peut connecter au maximum 32 émetteurs/récepteurs. C'est la notion d'UL (Unit Load ou charge unitaire) liée à la résistance d'entrée des récepteurs (fixée à $12\ k\Omega$ par la norme). Comme on trouve maintenant des convertisseurs qui ne présentent qu' $1/8$ UL, il est donc (théoriquement) possible de mettre 8×32 , soit 256 récepteurs sur le même bus.

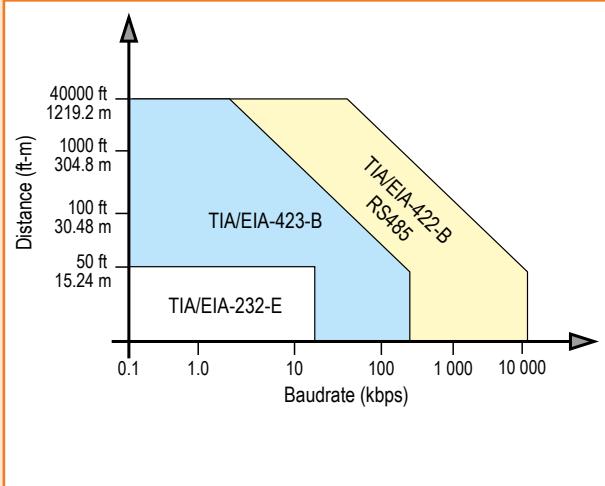
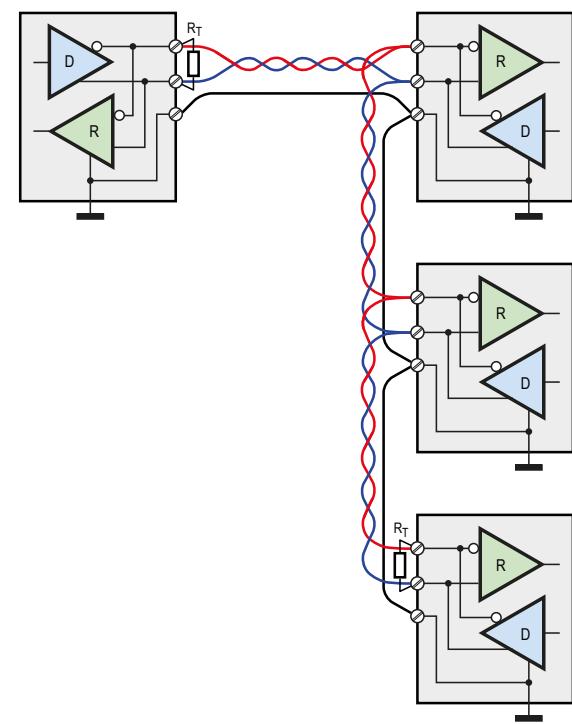


Figure 7.
Relation portée / débit des liaisons différentielles RS-422/RS-485.

Figure 8.
Topologie RS-485 typique.





Une passion

C'était il y a 25 ans pendant les vacances d'été ! Il pleut. Pour passer le temps, je demande à ma mère de m'acheter un magazine que je ne connaissais pas, le numéro double d'Elektor... Révélation ! Depuis l'électronique accompagne ma vie. Vie professionnelle en tant que manager du bureau d'étude de *EmkaElectronique*, mais aussi vie privée où j'y consacre pratiquement tout mon temps libre. Il ne passe pas une journée (et parfois quelques nuits) où je ne pense, dessine ou réalise une carte. Cette passion, je ne peux la vivre pleinement que grâce à la bienveillance de ma femme et de mes filles.

Merci également à Pascal, le tourangeau passionné pour ses conseils avisés.

Puissance du concentrateur USB dépassée
Le concentrateur n'a pas une alimentation suffisante pour faire fonctionner le périphérique USB composite.
Cliquez sur ce message pour afficher de l'aide.

Cet appareil peut fonctionner plus rapidement
Ce périphérique USB peut être plus rapide si vous le connectez à un port USB 2.0 à haut débit.
Pour obtenir la liste des ports disponibles, cliquez ici.

IMPORTANT : Il est nécessaire d'implanter le cavalier JP8 lorsque la carte est connectée à un ordinateur pour la première fois (avec ou sans alimentation externe). Cette procédure permet de contourner un avertissement de surconsommation signalé par le concentrateur USB à l'ordinateur, lequel interdirait alors l'installation des pilotes de la carte. Une fois les pilotes installés, JP8 peut être retiré (ou laissé en place avec une alimentation externe).

Pour paramétriser le montage avec ses deux liaisons sérielles différentielles, il convient d'utiliser le logiciel FT PROG disponible sur le site de FTDI [1] ; tous les pilotes USB pour le système d'exploitation y sont également disponibles. Une fois

le(s) pilote(s) installé(s), lancez tout d'abord le logiciel FT PROG, puis cliquez sur la loupe pour que le convertisseur soit détecté (**fig. 6a**).

Le convertisseur est alors correctement détecté par le logiciel. Cliquez ensuite sur « Hardware Specific ». (fig. 6b)

Pour les ports **Port A** et **Port B**, il n'y a rien à retoucher. Pour les ports **Port C** et **Port D**, il faut valider la broche *TX enable* : toujours avec l'outil FT PROG, cochez la case **RI as RS-485 Enable**. (fig. 6c)

Il reste enfin à sauvegarder la configuration dans l'EPPROM en cliquant sur l'icône avec l'éclair.

140033

FTDI & Android

Juste avant de boucler cet article, nous avons fait quelques essais avec des tablettes. Celles dont l'interface USB fonctionne en mode *host* reconnaissent le concentrateur USB, mais nous ne sommes pas arrivés beaucoup plus loin que la connexion d'une (seule) clé USB. En fait, il serait intéressant de pouvoir communiquer avec les ports RS-232/RS-485, mais pour cela il faudrait que le FT4232 converse avec Android, comme on le voit faire ici [6] ; ce lien est intéressant pour la documentation à laquelle il renvoie. Ce sera peut-être le sujet d'un prochain article, avec votre contribution ?

Liens

- [1] Site de FTDI pour les pilotes et le logiciel FT PROG :
www.ftdichip.com
- [2] www.elektor.fr/USB-serial-hub
www.elektor-magazine.fr/140033
- [3] Usinage des façades : www.schaeffer-ag.de

- [4] AN-1031 "TIA/EIA-422-B Overview
www.ti.com/lit/an/snla044b/snla044b.pdf
- [5] RS-422 and RS-485 Applications eBook
www.bb-elec.com/Learning-Center/All-White-Papers/Serial/RS-422-and-RS-485-Applications-eBook.aspx
- [6] FTDI et Android :
www.youtube.com/watch?v=QSR7IAAWL1c



powered by Eurocircuits

PCBs
Multilayer
Prototypes
Small series

Professional Quality
Trusted Service
Secure Ordering



Elektor PCB Service at a glance:

- ▶ 4 Targeted pooling services and 1 non-pooling service
- ▶ Free online PCB data verification service
- ▶ Online price calculator available
- ▶ No minimum order value
- ▶ No film charges or start-up charges

Delivery
from 2
working
days

Further information and ordering at www.elektorpcbservice.com



transformer un détecteur de fumée en détecteur universel

Jürgen Friker
(Allemagne)

Modifions un détecteur de fumée pour en faire un détecteur universel. Il suffit pour cela de quelques composants externes et vous rajoutez les capteurs de votre choix. Sans perdre la fonction de détecteur de fumée !

J'ai dans ma cave une espèce de puits dont le niveau d'eau est régulé automatiquement par une pompe électrique. Pour surveiller ce petit monde, j'ai eu l'idée de détourner un détecteur dont j'avais remarqué la puissance sonore inégalable.

Barouf

Après avoir essayé plusieurs avertisseurs sonores, dont notamment une vraie sirène alimentée par le secteur (restée désespérément muette lors d'une coupure de courant), j'ai été déçu par leur manque d'efficacité. Quand mon frère fait ses gammes au tuba, je ne risque pas d'entendre ce qui se passe à la cave, sauf si

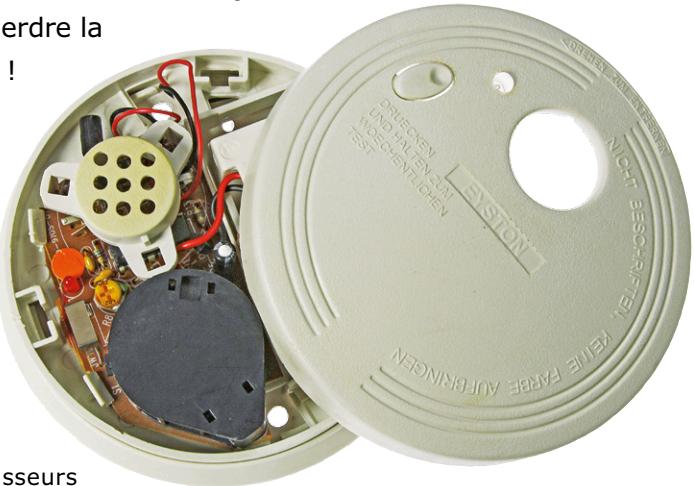


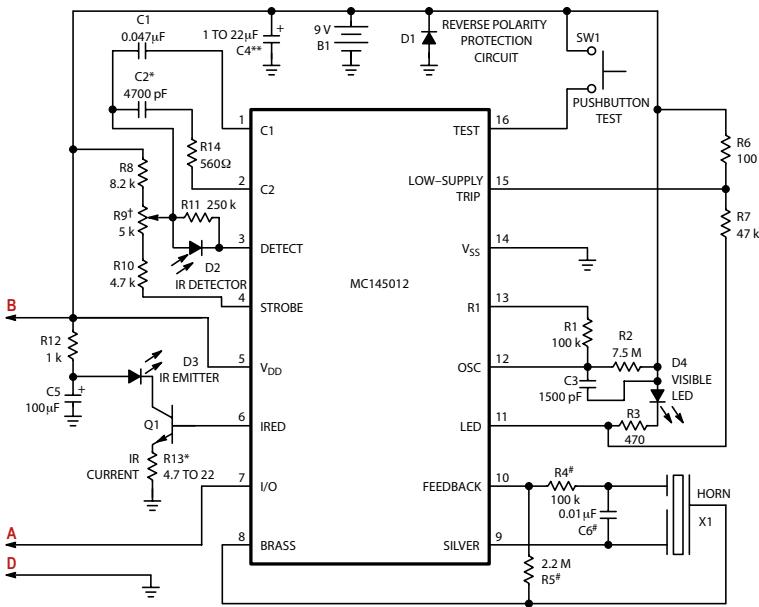
Figure 1.
Configuration standard du circuit intégré pour détecteur de fumée MC145012 selon Freescale.

l'engin fait vraiment du raffut. C'est ainsi que j'ai eu l'idée de squatter un banal détecteur de fumée. Ça marche sur pile, mais il faut être sourd comme un pot pour ne pas entendre ces engins en action. Et en plus, le bon détecteur de fumée surveille sa propre pile et prévient quand il faut en changer. Les frais que ça représente sont sans commune mesure avec le bénéfice obtenu. Le détecteur coûtera peut-être même moins cher que les composants que vous allez rajouter pour le transformer en détecteur universel.

Sur tous ceux que j'ai dépiautés, j'ai trouvé une puce Freescale (pas forcément la même), et une application de Motorola. La broche 7 du MC145012 (et consorts) permet de la mettre en réseau (**fig. 1**). Avec un bus à deux fils, vous formez ainsi des troupeaux de détecteurs : si l'un d'entre eux se met à beugler, c'est la bronca : les autres font aussitôt chorus et propagent la bonne nouvelle.

Cette entrée/sortie inutilisée sur un détecteur isolé va nous servir sans qu'il soit nécessaire de modifier le circuit. La fonction originale de détection de fumée restera intacte.

La broche 7 est forcée au niveau bas par une résistance (interne). Quand l'alarme est active,



cette ligne passe au niveau haut. Il peut donc y régner des tensions entre $-0,25\text{ V}$ et $V_{DD}+10\text{ V}$, soit jusqu'à $+19\text{ V}$ avec une pile de 9 V. Le seuil de commutation est toujours $+3,2\text{ V}$. Par sécurité, le MC145012 et les puces équivalentes sont dotés d'un drain de courant interne. On notera aussi que D4 indique si l'alarme a été déclenchée par ce détecteur (elle clignote au rythme des secondes) ou par un autre détecteur et transmise par la ligne d'E/S (D4 est éteinte).

N'utilisez jamais la broche 16 pour déclencher une alarme, cette *entrée de test* n'est pas faite pour recevoir des fils de grande longueur. Elle force le test en poussant au maximum le gain du circuit de mesure afin d'exploiter la lumière résiduelle dans le réceptacle à fumée. Un déclenchement intempestif se produirait si des fils de grande longueur, tels une antenne reliée à cette entrée, acheminaient des perturbations électromagnétiques vers le détecteur. D'ailleurs, un test ne se solde pas nécessairement par une alarme s'il n'y pas de fumée.

Détecteur de belle-mère

Le détecteur est sensible, mais pas raciste : qu'il détecte de la flotte, l'ouverture de la porte du frigo ou l'arrivée de la belle-mère, il se mettra invariablement à brailler. Vous pouvez donc connecter à peu près n'importe quel capteur à condition que pour l'activation de l'alarme le point A soit mis au niveau de la tension d'alimentation positive (dans les limites évoquées ci-dessous). Une combinaison logique OU câblée permettrait même de connecter plusieurs capteurs en parallèle. Si A reste en l'air ou s'il y règne moins de $1,5\text{ V}$, il n'y aura pas d'alarme. Pour mon puits, j'ai connecté un interrupteur flotteur, mais cela pourrait aussi être un transistor commandé par un circuit plus perfectionné.

Le chargeur en bas à gauche de la **figure 2** maintient en charge l'accu NiMH du détecteur. Pas de risque de surcharge : la loupiote LA1a avec sa résistance talon intégrée limite le courant de charge entre 1,5 et 1,9 mA sous 230 V, juste assez pour compenser l'auto-décharge de l'accumulateur. Accessoirement, cette loupiote signale la présence de la tension du secteur : c'est important, car même si le détecteur n'en a pas besoin, ma pompe si ! La zener D1 délivre une faible tension continue. Et comme elle conduit aussi dans l'autre sens, la lumière de la loupiote est assez

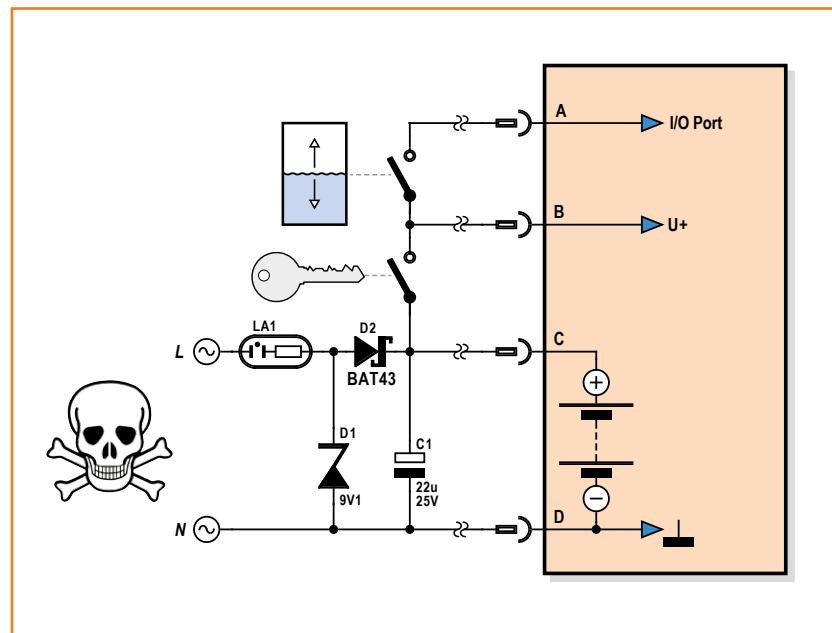


Figure 2.
Attention à cette petite alimentation sans transfo pour laquelle Elektor et moi dégageons toute responsabilité. Ne réalisez ce genre de circuit que si vous disposez des connaissances nécessaires pour appliquer les règles de sécurité.
Danger de mort !

vive. D2 protège le circuit contre la décharge, C1 est le classique condensateur de charge. Il est en parallèle avec C4 du circuit du détecteur. Pas question cependant d'augmenter la capacité de C1, cela perturberait la surveillance de l'état de la pile. En effet, toutes les 30 s environ, une impulsion de 8 ms est envoyée à la LED de contrôle D4 via R3 par la puce qui mesure en même temps le niveau de la batterie (sous charge) à l'aide du diviseur R6/R7. Si la capacité du condensateur de charge était plus forte que prévu, l'impulsion de courant de test en charge serait fournie par C1 bien plus que par la batterie, de sorte que le détecteur serait mal renseigné sur l'état réel de la charge de la batterie.

Tout ça est intéressant et plutôt original, ne trouvez-vous pas ? Mais c'est aussi très dangereux. D'abord parce que nous bidouillons des détecteurs dont peuvent dépendre des vies... et parce qu'avec des capteurs du genre *fil dénudé dans l'eau*, il ne saurait être question de rajouter une fonction de chargeur : **il règnerait sur tout ce circuit des tensions potentiellement mortelles et il faut donc impérativement isoler les capteurs.**

J'ai rajouté l'interrupteur à clé pour être sûr que la seule personne de la maison qui puisse la couper, c'est moi (et pas mon tubiste de frère).

(140184 – version française : Rémi Descistes)

cube à LED RGB animé avec Flowcode

un projet *Instructables* enlluminé et amélioré

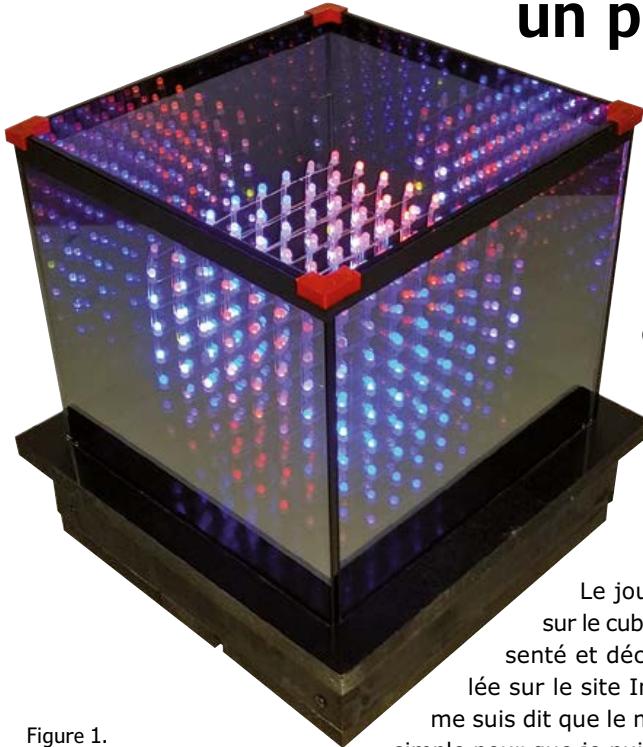


Figure 1.
Placez le cube de LED entre deux miroirs à l'équerre pour augmenter l'effet visuel.

Le jour où je suis tombé sur le cube de 8x8x8 LED présenté et décrit de façon détaillée sur le site Instructables [1], je me suis dit que le montage était assez simple pour que je puisse l'enrichir.

J'avais d'abord songé à une version 16x16x16 et unicolore du cube, mais j'ai réalisé que doubler la surface d'affichage revenait à quadrupler le nombre de LED et d'interconnexions. La perspective de piloter 2048 LED ne m'emballait guère, donc j'ai revu à la baisse mes ambitions et décidé de simplement remplacer les LED unicolores du cube d'origine par des LED RGB (rouge, verte, bleue). Autant prévenir tout de suite, même un cube de 512 LED RGB, cela fait encore près de 2200 soudures !

En préparant mon montage, j'ai découvert des projets dans lesquels les LED sont prises en sand-

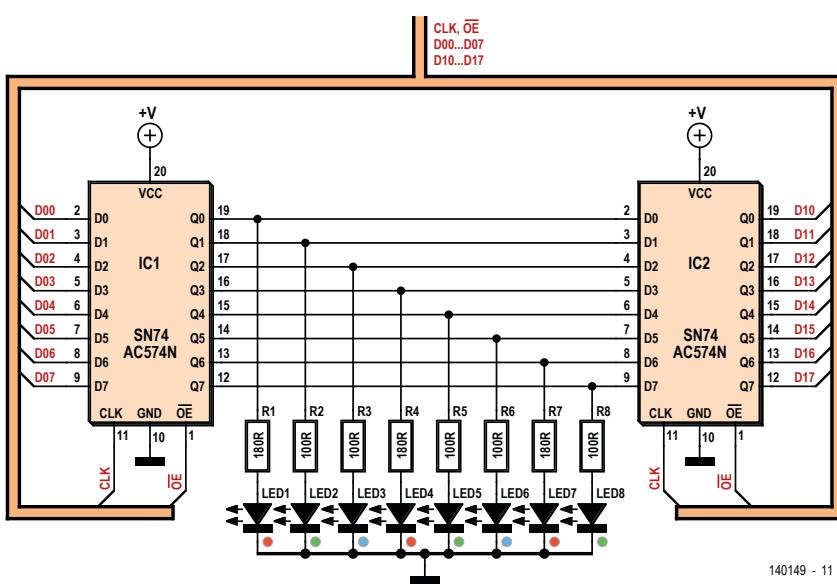
wich entre un miroir et une vitre recouverte d'un film miroir sans tain. Les multiples réflexions qui en résultent donnent l'impression que les LED fuient vers l'infini (**fig. 1**). J'ai appliqué certaines de ces techniques simples pour augmenter la surface d'affichage délimitée physiquement par les LED. Allumer et éteindre 512 LED RGB, chacune dotée de trois LED internes, nécessite un grand nombre de signaux de commande. J'ai donc recouru au multiplexage pour que ces signaux soient partagés entre les LED. Leur cathode est connectée à la masse et leur anode est reliée à une résistance-talon.

Outil de multiplexage n° 1 : CI tampon

J'ai utilisé deux méthodes de multiplexage pour piloter de façon indépendante les 512 LED RGB. La première utilise des tampons constitués de 8 bascules de type D (SN74AC574N). Ces CI permettent de créer suffisamment de signaux de commande pour piloter un segment de 8x8 LED, soit un huitième du cube. Pour piloter les 3 canaux de couleur de ces 64 LED, il faut donc 192 signaux de commande. Et pour piloter ces signaux, il faut chaîner 24 des CI de façon à ce que la sortie du premier pilote l'entrée du deuxième, qui lui-même pilote le troisième, etc. La **figure 2** montre comment deux tampons sont chaînés et connectés pour piloter les LED.

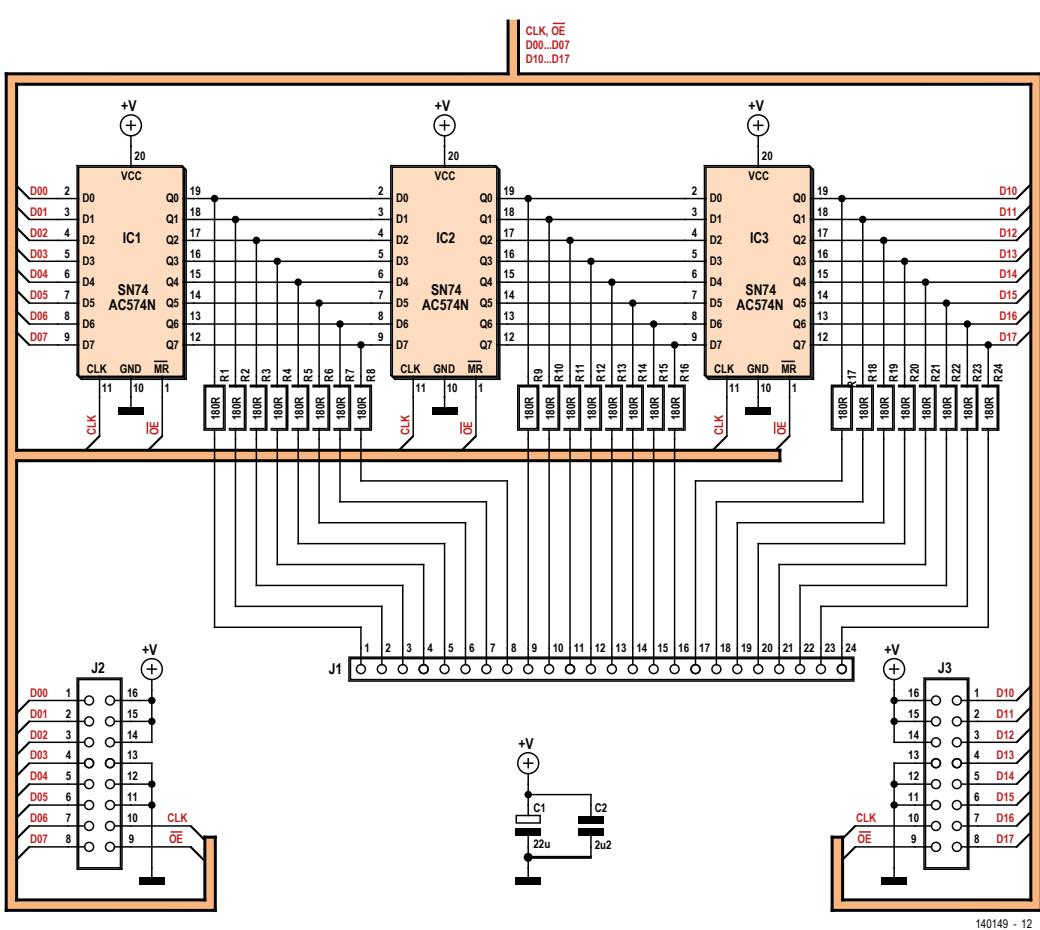
Caractéristiques

- 512 LED RGB pour un cube 8x8x8
- LED adressables individuellement
- peut être utilisé pour un effet « réflexions infinies »
- modélisation possible des animations
- développé et programmé avec Flowcode 6 et ECIO40P16
- code adaptable aux cartes Arduino



140149 - 11

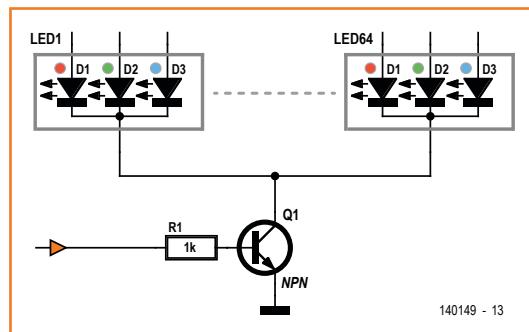
Figure 2.
Une façon de multiplexer les LED est de les placer entre deux tampons auxquels sont appliqués des signaux de commande.



140149 - 12

Figure 3.
Schéma de la carte tampon modulaire. Huit de ces cartes sont nécessaires pour piloter le cube de 8x8x8 LED RGB.

Figure 4.
Un des huit transistors NPN utilisés pour activer et désactiver de façon logicielle les segments 8x8, et ainsi multiplexer l'ensemble des 512 LED RGB.



Le signal d'horloge est commandé par le microcontrôleur et utilisé pour décaler et transmettre les signaux de sortie d'un tampon au suivant. J'ai créé un pilote de tampons modulable qui permet d'augmenter ou réduire la taille du cube en fonction du nombre de LED souhaité. Pour cela j'ai fabriqué une carte dont les trois CI tampon commandent une rangée de 8 LED RGB. En chaînant huit de ces cartes « pilote », j'ai ainsi pu commander un segment complet de 8x8 LED. Un câble en nappe à 16 fils délivre d'une carte à l'autre les 10 signaux de commande ainsi que l'alimentation et la masse. Un autre câble en nappe, à 24 fils celui-ci, transmet les signaux à appliquer sur les LED via les résistances embarquées. Chaque carte est équipée de condensateurs servant de sources d'appoint locales pour le besoin en courant des LED.

Outil de multiplexage n° 2 : transistors NPN

Sur la **figure 3**, chaque segment de 8x8 LED est piloté par une chaîne de tampons. Pour commander les sept autres segments du cube, une autre méthode de multiplexage était nécessaire. Pour la mettre en œuvre, j'ai relié ensemble toutes les cathodes d'un segment de 8x8 LED, puis je les ai connectées à la masse via un transistor NPN (**fig. 4**). Un unique signal de commande appliqué sur la base du transistor permet d'activer ou désactiver toutes les LED du segment. La **figure 5** montre le montage final : huit cartes « pilote » empilées.

Relier ensemble les cathodes de chaque segment horizontal (de 8x8 LED) et utiliser 8 transistors NPN me permettait donc d'activer ou désactiver chacun des 8 segments. J'étais bien en mesure de commander chaque LED de chaque segment, mais manifestement je ne pouvais éclairer qu'un huitième du cube à la fois à un instant donné. L'œil est heureusement facile à berner : en acti-

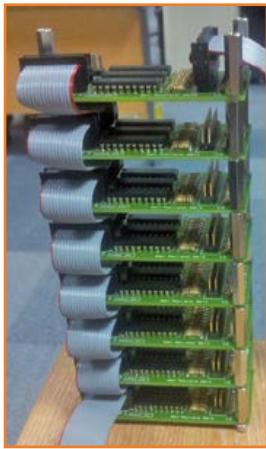


Figure 5.
La pile des cartes pilote.

vant et désactivant constamment et suffisamment rapidement les segments successifs, le cube semble éclairé en permanence.

Montage

Voilà pour la théorie, la pratique se résume quant à elle à prendre une grande inspiration ou un breuvage corsé et à se lancer dans une joyeuse séance de pliage et de soudage. Dois-je vraiment préciser que plier les pattes des LED et les souder est... comment dire... assez fastidieux ? Un gabarit de pliage et un gabarit de soudage ont rendu l'opération plus pratique, et surtout m'ont aidé à maintenir une qualité aussi constante que possible au fur et à mesure que je perdais l'envie de vivre. Par bonheur, mes mains et mon cerveau ont fini par divorcer et j'ai donc aussi eu des pensées heureuses et libres durant cette période.

Pliage des LED

La première étape de l'assemblage consiste donc à plier correctement les pattes des LED à l'aide d'un gabarit, que j'ai fabriqué en perçant un trou de 5 mm de diamètre dans un morceau de panneau de fibres (MDF). Le trou doit être parfaitement adapté à la dimension des LED et doit permettre de les insérer et de les retirer facilement. J'ai ensuite tracé des repères de pliage autour du trou (**fig. 6**) : une ligne verticale (A) et une ligne horizontale (B) passant toutes deux par le centre du trou, et une ligne horizontale (C) tangente au bord supérieur du trou. Plier les pattes à l'aide d'une règle métallique s'est révélé très pratique. Les photos de la **figure 7** illustrent les étapes à suivre.

1. Insérer la LED dans le gabarit en vérifiant son orientation. Toutes les pattes doivent être alignées avec la ligne horizontale (B), et la cathode doit être la seconde patte en partant de la droite.
2. Premier pliage : plier la cathode vers le bas et les anodes vers le haut selon la ligne (A) de façon à ce qu'elles soient parallèles au bois et au « fond » de la LED.
3. Deuxième pliage : utiliser une règle métallique pour plier la cathode vers la gauche parallèlement à la ligne (B).
4. Troisième pliage : placer la règle selon la ligne horizontale (C) et plier à nouveau les anodes vers le haut. Voilà, plus que 511 LED à plier !

Préparation des segments 8x8

Une fois vos 64 LED pliées, vous pouvez vous détendre en passant à l'assemblage d'un segment

de 8x8 LED. Là aussi j'ai fabriqué un gabarit, mais cette fois-ci pour maintenir en place les LED. Si vous avez une imprimante 3D, vous trouverez le fichier de son modèle 3D sur le site Thingiverse [2]. Si vous n'en avez pas, vous trouverez sur le site Instructables une méthode différente pour maintenir les LED. J'ai attaqué les LED une par une. Pour chaque LED placée sur le gabarit, j'ai soudé ensemble les anodes de la colonne en cours, et fait de même avec les cathodes de la rangée en cours (**fig. 8**). Il faut prendre son temps, et s'assurer que chaque soudure est propre et robuste. Référez toute envie d'aller vite, le cube pourrait se transformer en dé pendant au moment de l'allumer.

J'ai testé chaque segment terminé sans le retirer de son gabarit. Je voulais m'assurer que chaque LED s'allumait et que le segment ne comportait ni coupure ni court-circuit. Pour cela j'ai relié des grandes longueurs de fils à mon alimentation de 5 V, puis j'ai soudé à l'extrémité d'un de ces fils la plus grande des deux résistances du port 1 de la carte pilote. J'ai ainsi pu commander sans risque toutes les LED : en appuyant ma règle en fer contre les fils communs non connectés, je mettais brièvement en court-circuit les broches communes avec la masse, ce qui m'a évité d'avoir à tester chaque LED individuellement, puisqu'ainsi je pouvais tester une colonne d'anodes à la fois. Pour « démouler » le segment du gabarit, il m'a suffi de soulever doucement chaque tête de LED jusqu'à ce que l'ensemble se dégage naturellement. Ensuite, retour au pliage des LED du segment suivant !

Assemblage des segments 8x8

Une fois terminé le soudage des rangées et colonnes de LED, je suis passé à l'assemblage des huit panneaux obtenus. J'ai posé le premier

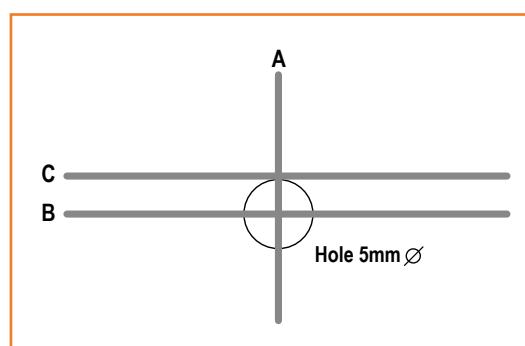


Figure 6.
Les lignes qui vous aideront à plier les LED sans devenir fou.

à plat sur une table, puis soudé deux longueurs de fil monobrin bien raide avec chaque fil des cathodes. Souder de chaque côté du panneau renforce la solidité de l'ensemble et fournit une petite sécurité au cas où la soudure d'un des fils lâcherait. Pour obtenir des fils rigides et droits, j'ai simplement agrippé chaque extrémité avec des pinces et tiré fort. J'ai dû essayer plusieurs sortes de fils avant de trouver le bon, à savoir celui adapté à la fois au courant de 64 x 3 LED et à mes muscles partisans du moindre effort. J'ai utilisé des boîtes d'allumettes pour écarter le panneau suivant du premier. Je les ai tous montés de cette façon, en prenant chaque fois le temps de bien aligner le tout. Pour finir, j'ai effectué un dernier test complet des LED afin d'être certain de n'avoir rien oublié. La vue 3D du cube en pleine gestation virtuelle (**fig. 9**) vous montre la position des segments 8x8.

Support

Ainsi assemblé le cube de LED sera assez solide, mais allez savoir quelle sorte de moineau pourrait venir y poser son surpoids. Il serait prudent de le renforcer, ne serait-ce que pour éviter de ruiner



Figure 7.
Comment plier les pattes des LED pour assembler le cube.

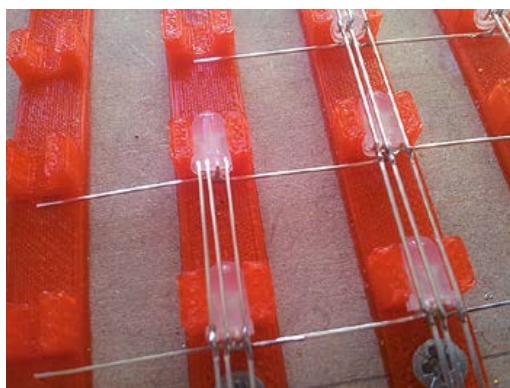


Figure 8.
Le gabarit utilisé pour maintenir les segments de 8x8 LED afin de souder les anodes et les cathodes et créer les rangées et les colonnes.

Figure 9.
Vue en 3D de l'emplacement d'un segment 8x8 à l'intérieur du cube.

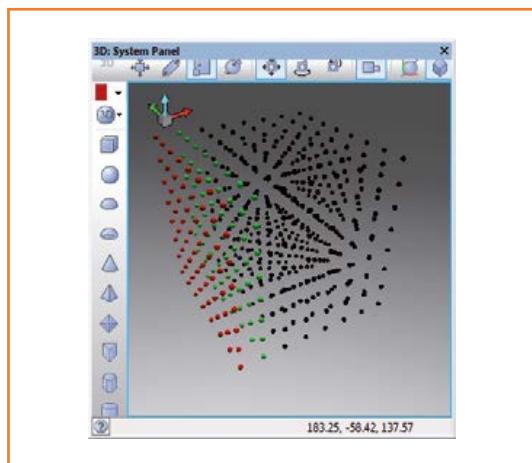
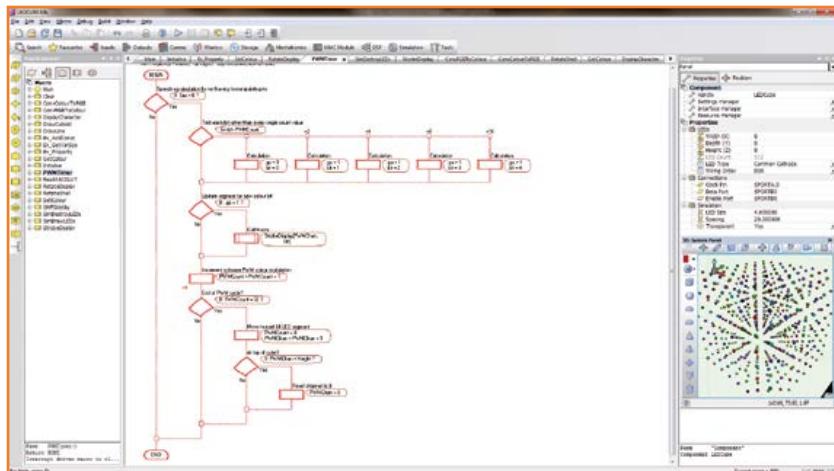


Figure 10.
Flowcode 6, sous lequel a été programmé le cube à LED RGB. Le panneau de droite contient les éléments de la modélisation. Le programme complet et quelques bonus sont dans l'archive 140149-W.zip du lien [3].

bêtement tant d'efforts. Dans le même panneau de fibres (MDF) qui m'avait servi à fabriquer les gabarits, j'ai percé des trous de 5 mm espacés de façon à accueillir les pattes de la rangée inférieure de LED. Pour déterminer l'espacement des trous, j'ai mesuré sur deux côtés parallèles du cube les distances entre chaque LED, de celle de gauche à celle de droite, puis établi une moyenne. J'ai refait le même calcul pour les deux autres côtés. Ensuite j'ai percé une autre série de 8 trous de 3 mm le long d'une arête, de façon à ce que les fils des cathodes passent à travers le cube. Dernière étape, j'ai tracé une rainure de 6 mm pour le vitrage, puis peint le panneau de fibres.

L'insertion du cube dans son support, l'ajout des rails pour les cathodes communes ainsi que le schéma de câblage, sont détaillés dans un document que vous pouvez télécharger gratuitement depuis le site Elektor [3].



Cerveau

J'ai utilisé des morceaux de plaque d'essai pour relier directement une « brique » ECIO40P16 à mon alimentation, et aussi pour la circuiterie du pilotage par NPN. Une alimentation de 5 V est indispensable pour que le courant relativement fort qui circule ne produise pas de chaleur excessive. Le composant Flowcode « LED Cube » est adapté à pratiquement n'importe quelle famille de microcontrôleurs, à l'exception des PIC à 8 bits car ils ne peuvent pas manipuler des tableaux de plus de 256 octets. Ne vous fiez pas à sa petite taille, le contrôleur ECIO40P16 est vraiment puissant. Adapter le code permet néanmoins de piloter le cube avec une carte Arduino équipée d'un AVR standard.

Quelle que soit la méthode choisie pour loger ou fixer l'électronique de commande, pensez à ajouter un trou pour l'alimentation. Un moyen de réduire le fouillis est d'utiliser une alimentation à découpage logée dans un boîtier métallique et de l'insérer dans le boîtier principal. Vous pouvez aussi laisser le câble USB connecté à l'ECIO pour, d'une part ne pas avoir à ouvrir le boîtier à chaque reprogrammation, d'autre part disposer d'une voie USB permanente si vous souhaitez utiliser les composants USB de Flowcode.

Flowcode 6

Je me suis servi de Flowcode 6 pour modéliser le cube et écrire le code. Ce programme possède tout ce qu'il faut pour aboutir à ce prodige psychédélique qu'est le cube à LED. J'ai d'abord créé une sphère sur le panneau, je l'ai rendue invisible, puis je l'ai clonée de façon à pouvoir remplir les dimensions X, Y et Z. J'ai ensuite créé des routines pour placer ces sphères et paramétrier leur couleur. Partant de là, j'ai pu produire les routines de tracé des « cubes » (cuboids) et des lignes, ainsi que les macros de translation et de rotation. J'ai extrait d'un ancien composant LCD graphique une macro d'affichage de caractères. L'ajout d'un système de double tampon a conclu la modélisation du composant.

Pour les fonctions embarquées, j'ai créé un tableau de variables de 16 bits chargées de stocker les valeurs de couleur des LED. J'ai ensuite écrit une routine pouvant être appelée régulièrement pour transférer les données vers les tampons ou encore commuter entre les huit canaux partagés. À ce stade, tout ce qu'il me restait à faire était d'ajouter le code et de paramétrier les

Liens

- [1] Instructables: www.instructables.com/id/Led-Cube-8x8x8/
- [2] Thingiverse: www.thingiverse.com/thing:231031
- [3] BOM, CAD, 3D Print, Flowcode 6 Program, Demos: www.elektor.com/140149
- [4] Vidéo: <https://www.youtube.com/watch?v=odFljHeCNaYt>

Ressources matérielles

Flowcode 6: www.elektor.com/flowcode-6-for-pic-fc6cp01nepic

ECIO40P16: www.elektor.com/pic-ecio-40-pin-ecio40p

Bibliographie

Create 30 PIC projects with Flowcode 6: www.elektor.com/flowcode-6-book



macros de couleur pour que le tableau puisse être lu et écrit en condition réelle.

Le composant final est disponible en [3]. Il suffit de le glisser dans le panneau Système de Flowcode. La **figure 10** montre un ordinogramme du programme.

Viens chez moi, j'ai un LEDoscope cubique

Flowcode m'a aussi servi à écrire des programmes produisant un effet de pluie d'orage et d'éclairs,

une boule de feu, la fusion de deux plasmas, un affichage de texte, et une animation que je vous laisse découvrir sur Youtube [4]. L'étape suivante consistera à ajouter un microphone à la circuiterie pour commander les ballets de lumière. L'idée serait de hacher le signal audio en gammes de fréquences par transformée de Fourier rapide. Beaucoup de défis ici, dites-nous si vous aurez pu les relever !

(140149 – version française : Hervé Moreau)

Liste des composants

Description	Quantité	Vendeur	Référence
LED RGB, 5 mm, lentille de diffusion, cathode commune	512	eBay	
Alimentation à découpage CC, 5 V, 5 A	1	eBay	
Microcontrôleur (ECIO40P16 ou Arduino)	1	e-choppe Elektor	
Bascules SN74AC574N	24	Farnell	165-2006
Câble en nappe 16 voies avec IDC	2 m	Farnell	120-7442
Câble en nappe 24 voies avec IDC	4 m	Farnell	120-7439
Condensateur électrolytique aluminium 10 µF	8	Farnell	211-2751
Condensateur céramique 100 nF	8	Farnell	187-1027
Résistance 100 Ω 0,25 W (LED verte et bleue)	128	Farnell	933-9043
Résistance 180 Ω 0,25 W (LED rouge)	64	Farnell	933-9230
Connecteur femelle IDC à 16 contacts	16	Farnell	221-5235
Embase à 2 rangées et 16 broches	16	Farnell	102-2240
Connecteur femelle IDC à 24 contacts	8	Farnell	221-5253
Embase à 2 rangées et 24 broches	8	Farnell	159-3448
Transistor TIP122 NPN 5 A	8	Farnell	980-4021
Morceaux de plaque d'essai	1	Farnell	120-1473
Vitre en verre trempée d'épaisseur 6 mm	5		
Panneau de fibre (MDF) d'épaisseur 20 mm	1		
Peinture en bombe pour peindre le MDF	1		
Film teinté réfléchissant pour vitre auto	1		

circuits imprimés : éviter les bourdes en HF et avec les fréquences élevées en général

Robert Lacoste (Chaville)

Bienvenue dans ce nouvel article de notre rubrique « Hors Circuits ». Après vous avoir présenté des notions relatives aux adaptations d'impédance, aux lignes microruban et aux problèmes de réflexion de signaux [1] qui peuvent se produire si l'on ne respecte pas ces règles, je conclurai cette mini-série ce mois-ci par une **mise en œuvre pratique** de ces notions : quelles sont les « bonnes manières » pour la conception d'un circuit imprimé ? En particulier, quelles précautions prendre en manipulant des signaux de fréquence élevée. Non, ne partez pas quand je lâche ce mot... qu'il s'agisse d'un projet radio ou simplement de circuits numériques rapides, des fréquences dites *hautes*, il y en a partout. Un simple microcontrôleur peut de nos jours avoir des fréquences d'horloge indécentes, et il est bien rare qu'il n'y ait une interface USB ou Ethernet qui traîne...

Masse, masse, masse...

Commençons par la principale source d'ennuis : la masse, ou plutôt l'absence de masse. Pour avoir des problèmes dans un circuit, rien de tel qu'un plan de masse insuffisant ou inexistant. Prenez n'importe quel schéma et imaginez une différence de potentiel entre deux points théoriquement connectés à la masse et donc (supposés) équipotentiels : cette différence peut entraver voire empêcher le fonctionnement du montage. Ceci arrive fréquemment : Une connexion à la masse établie par un fil un peu long se comporte comme une self. Or, dès que les fréquences montent, une self devient une impédance de forte valeur. Et les soucis arrivent.

La seule solution fiable pour éviter de tels problèmes, c'est un plan de masse complet. Enfonçons le clou : Complet, ça veut dire qu'une face du circuit imprimé est entièrement recouverte de cuivre à laquelle sont raccordés le plus directement possible tous les points de masse du circuit. Malheureusement cette solution ne peut en général être réalisée qu'avec un circuit imprimé à plusieurs couches. Par exemple avec un circuit à 4 couches, l'une des couches internes peut être utilisée comme plan de masse, laissant 3 couches pour les alimentations et les connexions (**fig. 1**). Cette solution garantit une impédance minimale entre deux points quelconques du circuit reliés à masse, et c'est pour cela que la plupart des circuits professionnels, à moins d'être soumis à des contraintes de coût drastiques, sont multicouches. Même avec une carte multicouche, respectez la règle sacro-sainte du plan de masse : ne jamais faire passer sur cette couche quelque autre piste que ce soit, au risque d'introduire des impédances parasites. Selon la loi de Murphy, si ça arrive, ce sera toujours au plus mauvais endroit...

Même quand on n'a pas le budget pour un circuit multicouche, il faut toujours se tenir au plus près possible du concept de plan de masse *complet*, c'est-à-dire sur toute la surface de la carte. Demandez à votre programme de CAO préféré de remplir par un plan de masse toutes les zones inutilisées, sur les deux faces, et raccordez-les entre elles par le plus de vias possibles pour minimiser les impédances parasites (**fig. 2**). Ce sera moins bon que le plan de masse intégral, mais restera acceptable si vous laissez assez de plan de masse surtout sous les zones les plus sensibles.

Certains préconisent d'utiliser *plusieurs* plans de masse distincts, p. ex. l'un pour la partie numérique d'un circuit, l'autre pour la partie analogique, raccordés en un seul point. C'est le cas p. ex. des notes d'application de convertisseurs A/N qui souvent recommandent de placer le point de raccordement des deux plans sous le circuit intégré lui-même. L'intention est de réduire l'impact du bruit circulant dans le plan de masse. En 25 ans de carrière dans l'électronique professionnelle, je n'ai jamais rencontré une seule situation où ce conseil était réellement applicable ! La raison est simple (**fig. 3**). S'il n'y a qu'un seul point de raccordement entre les deux plans de masse, tous les autres fils de raccordement entre les sections

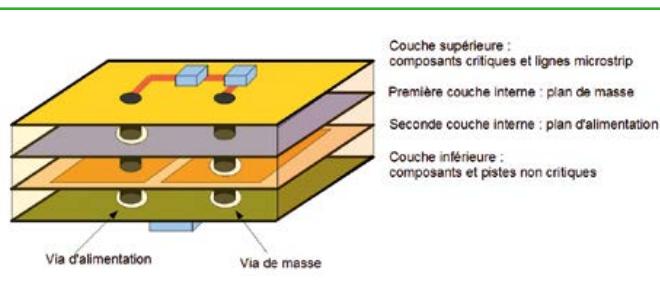


Figure 1 : Un circuit multicouche permet de dévier un plan interne comme plan de masse complet, ce qui est l'idéal et de loin le plus simple.

analogique et numérique devraient passer exactement sur ce même point. Sinon des boucles se forment dans le chemin aller-retour emprunté par le courant, créant de redoutables antennes. Celles-ci peuvent être soit des sources de parasites soit en capter. Tant que le circuit est sur une carte d'évaluation, il reste possible de s'en sortir, mais sur un projet réel, c'est souvent insoluble ; la segmentation du plan de masse fait beaucoup plus de mal que de bien. Je recommande donc un beau plan de masse unique, sauf si vous savez vraiment ce que vous faites (ce qui peut arriver pour certaines applications audio, à des fréquences moins élevées).

Impédances contrôlées

Je vous renvoie aux informations sur les adaptations d'impédances de mes trois précédents articles que je ne vais pas répéter. Pour mémoire, une piste reliant deux circuits doit être d'impédance caractéristique égale à celle des deux circuits, si l'on veut éviter des pertes de puissance et des réflexions parasites. En général, les circuits ont une impédance de $50\ \Omega$, et la piste devra donc aussi présenter une impédance de $50\ \Omega$. Deux techniques classiques sont utilisées : la ligne *microstrip* ou microruban, qui est une simple piste de largeur donnée circulant sur un plan de masse complet, et la ligne *stripline* qui est en sandwich entre deux plans de masse. La **figure 4** donne les largeurs classiques de pistes à utiliser.

L'usage de pistes d'impédances caractéristiques bien adaptées ne fait jamais de mal, mais n'est bien entendu pas toujours indispensable. En fait, la bonne démarche consiste à calculer d'abord la longueur d'onde correspondant à la fréquence la plus élevée du circuit. Imaginons que ce soit $F = 2,4\ \text{GHz}$. La longueur d'onde correspondante dans le vide est $L = c/f$ avec c la vitesse de la lumière, soit $L = 3,10^8 / 2,4 \times 10^9 = 125\ \text{mm}$. Comme le signal se propagera non pas dans l'air mais sur le circuit imprimé, il faut donc, croyez-moi, diviser cette distance par la racine carrée de la constante diélectrique du substrat utilisé (4,3 pour le FR4 classique), d'où une longueur d'onde effective de l'ordre de 60 mm. La règle prudente est alors la suivante : toute piste critique de longueur supérieure au dixième de cette longueur d'onde, soit ici 6 mm, doit être réalisée en impédance contrôlée. Bien sûr, cette longueur limite sera de 6 cm pour 240 MHz ou 60 cm pour 24 MHz, ceci vous explique pourquoi on n'a, en général, à se préoccuper de l'adaptation d'impédance des pistes que dans les circuits radio ou dans les circuits numériques rapides.

Placement des composants

La conception du circuit imprimé commence dès le choix des composants. Bien sûr, des composants montés en surface (CMS) sont à privilégier dès que les fréquences montent, car leur petite taille permet de limiter les capacités et inductances parasites. Quelle taille de boîtier choisir, en particulier pour les composants passifs ? Une bonne approche consiste à utiliser si possible un boîtier de même largeur que la piste "50 Ω ",

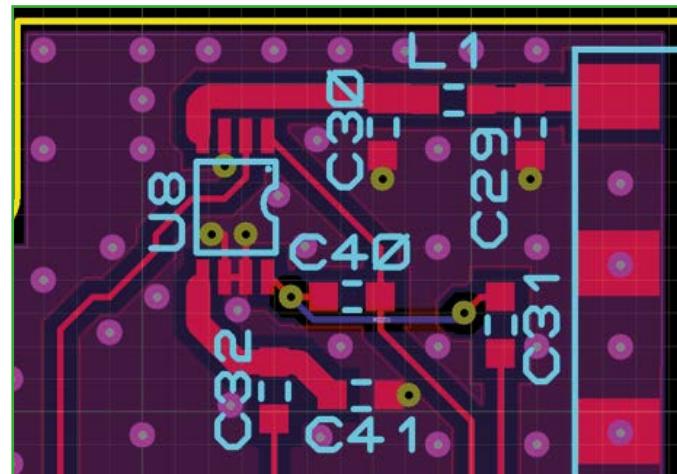


Figure 2 : Avec un circuit double-face la seule solution consiste à remplir toutes les zones non utilisées des deux faces par un plan de masse, et surtout de relier entre eux ces plans par de multiples via.

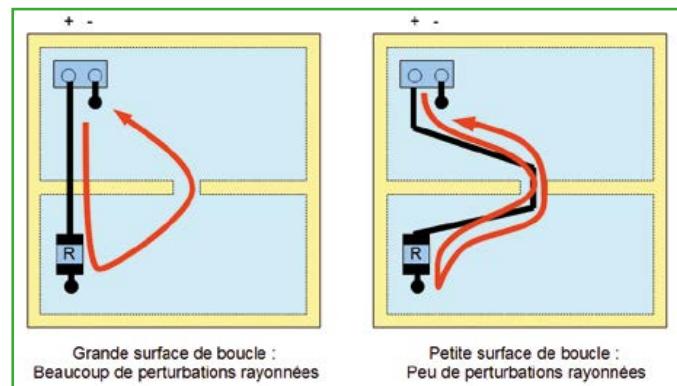


Figure 3 : Réaliser deux plans de masse est le plus souvent une très mauvaise idée. En effet dans ce cas toutes les autres liaisons doivent passer exactement sur le point de raccordement entre les deux plans afin d'éviter de créer des boucles de courant qui sont autant d'antennes...

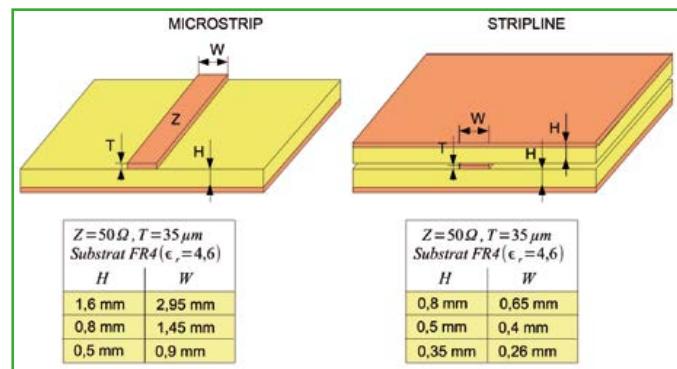


Figure 4 : Une ligne microstrip est une simple piste de largeur constante circulant sur un plan de masse. Ces tableaux donnent les largeurs usuelles pour une impédance de $50\ \Omega$.

•hors-circuits

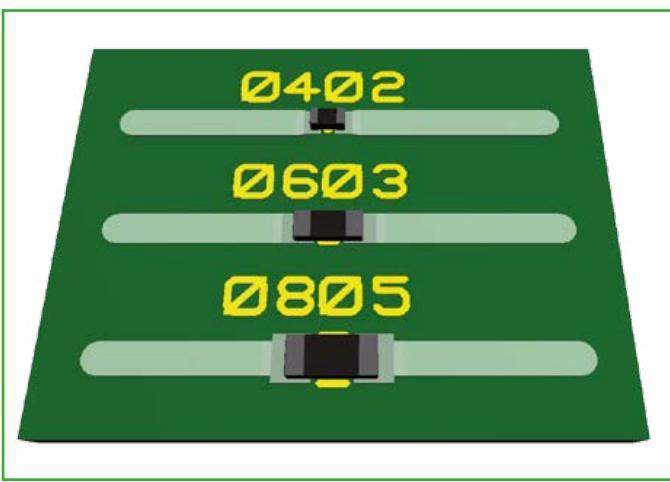


Figure 5 : Utiliser des composants de même taille que la largeur de la piste minimise les réflexions parasites. Ici le choix d'une taille 0603 serait idéal.

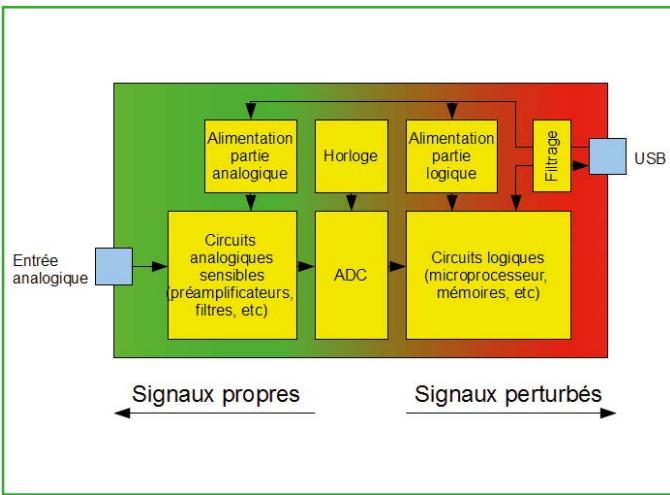


Figure 6 : La première étape est de bien penser l'agencement du circuit imprimé, en éloignant les sections sensibles autant que possible des sections perturbatrices.

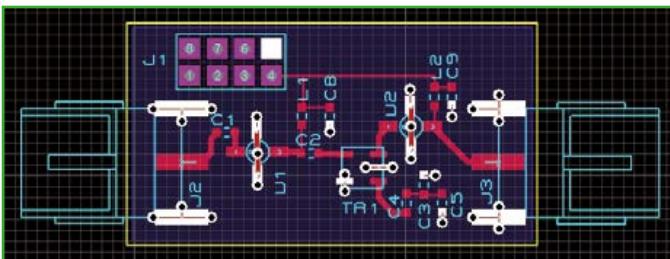


Figure 7 : Si votre CAO le permet (ici Proteus de Labcenter), mettre en surbrillance les points reliés à la masse permet facilement de repérer les emplacements où mettre des vias vers le plan de masse.

cela minimise les changements brusques de largeur et donc les désadaptations (**fig. 5**).

Vient ensuite la phase de placement des composants sur le circuit imprimé, cruciale pour obtenir de bonnes performances. Par expérience, mieux vaut prendre son temps sur le placement, le routage n'en sera que plus facile. Comment s'y prendre ? Le plus efficace est d'abord de segmenter le projet en ses différentes sections et de réfléchir à leur meilleur agencement sur le PCB (**fig. 6**) : éloigner les parties les plus sensibles du champ de rayonnement d'un microcontrôleur, agencer les connecteurs pour réduire le plus possible la longueur des pistes et les risques d'interférences... Mieux vaut y réfléchir sérieusement !

Après ces préliminaires, il est temps de positionner chaque composant. Pour cela, je vous conseille de garder devant vous une copie imprimée du schéma. Placez d'abord les composants critiques sur le plan de la qualité des signaux, ainsi que leurs lignes d'alimentation et leurs capacités de découplage. Évitez autant que possible de croiser des pistes "sensibles" comme celles transportant des signaux RF ou rapides : tout changement de couche implique une petite désadaptation. Puis placez les composants moins critiques, en éloignant le plus possible les parties les plus bruyantes des zones sensibles. Lorsque vous utilisez des CMS, n'oubliez pas qu'ils peuvent être implantés sur l'une ou l'autre face ; c'est très utile pour les condensateurs de découplage qui, implantés sur la face inférieure, peuvent être placés *au plus près* des composants dont ils découplent l'alimentation.

Masse, encore...

Une fois les composants placés, il est temps de router les pistes. Je vous déconseille de lancer à ce stade un logiciel d'auto-routage, même si vous avez de très bons outils de CAO : seul le concepteur sait ce qui est critique ou ne l'est pas... à moins de passer beaucoup de temps à l'expliquer au logiciel ! Commencez toujours par relier au plan de masse toutes les connexions qui doivent l'être, et par une piste la plus courte possible. En pratique, cela revient à mettre un via vers le plan de masse au quasi-contact de toutes les broches des composants concernés. Si votre programme de CAO le propose, une option bien pratique pour cela est de mettre en surbrillance toutes les broches connectées à la masse (**fig. 7**). Ces liaisons vers la masse doivent être très courtes. Rappelez-vous que l'inductance d'une piste de 10 mm de long est de l'ordre de 1 nH. Cela peut sembler peu, mais à 2 GHz cela représente une impédance de 12 Ω, ce qui n'est pas forcément négligeable ! Toujours pour minimiser les impédances parasites, évitez d'utiliser un seul via pour raccorder plusieurs points à la masse. Mieux vaut mettre plus de vias et ne prendre aucun risque...

Une fois les masses reliées, passez aux pistes critiques, en utilisant si besoin des lignes microruban, c'est-à-dire avec la

Une connexion à la masse établie par un fil un peu long se comporte comme une self

bonne largeur pour obtenir l'impédance caractéristique voulue. Passez ensuite aux alimentations, en utilisant si possible des plans d'alimentation en couche interne, puis terminez par les signaux les moins critiques.

Au niveau des pistes critiques, une erreur commune est de mettre des "T" sur celles-ci. Imaginez un circuit avec un connecteur susceptible d'avoir deux implantations différentes, en haut ou en bas (**fig. 8**). Si vous n'y prêtez pas attention, vous aurez une piste qui se sépare en deux pour aller vers l'un et vers l'autre connecteur. Or, l'une de ces deux pistes sera forcément terminée en circuit ouvert du côté où il n'y aura pas de connecteur, mais elle ne sera pas du tout transparente sur le plan de l'impédance de la piste. En fait, vous aurez là un beau résonateur qui, à une certaine fréquence, sera équivalent à un court-circuit vers la masse !

Ce sujet mériterait à lui seul un article, mais sachez que, par exemple, une piste ouverte de 40 mm de long résonne à approximativement 2,7 GHz, ce qui peut arriver. La solution pour éviter cela est de prévoir deux résistances $0\ \Omega$ au point de raccordement, dont seulement l'une des deux sera montée, comme l'illustre la figure 8, du côté où le connecteur sera implanté.

Plans de masse additionnels...

Nous n'avons pas fini de parler des plans de masse. Imaginez un circuit à 4 couches, avec un beau plan de masse en couche interne. Les zones inutilisées des autres couches doivent-elles être remplies par un autre plan de masse ou faut-il les laisser vides ? En général, la première solution limite les rayonnements parasites des pistes et améliore le découplage entre elles, c'est donc la meilleure, mais deux précautions doivent être prises (**fig. 9**). Tout d'abord le plan de masse doit rester suffisamment loin des pistes microruban. Plus exactement, si le plan de masse se rapproche trop de la piste, son impédance sera modifiée. Rien ne vous empêche de le faire, mais il faut alors recalculer la largeur de celle-ci, c'est une structure appelée *Coplanar Waveguide* (CPW).

L'autre précaution est de veiller à ce que les différents plans de masse soient équipotentiels. Pour cela, il ne suffit pas de les raccorder électriquement, il faut que ce raccordement présente une impédance suffisamment basse. La seule solution pour cela est de multiplier les vias d'interconnexion entre les plans de masse. Une règle empirique ? La distance entre deux vias de masse ne doit jamais dépasser un dixième de la longueur d'onde sur le circuit imprimé de la plus haute fréquence utilisée. Par exemple, pour 2,4 GHz, cela donne

au moins un via tous les 6 mm. La présence de tels vias de masse sur le pourtour de la carte et autour des pistes les plus sensibles réduira les pertes et les couplages par la tranche du circuit imprimé. Maintenant vous comprenez pourquoi les circuits imprimés hyperfréquence ressemblent souvent à des gruyères plein de vias de masse : si les fréquences dépassent 10 à 15 GHz, les vias sont espacés de 1 mm, ce qui fait 100 vias par centimètre carré !

Que faire quand deux pistes critiques doivent se croiser ? Cherchez s'il n'y a vraiment pas d'autre solution pour éviter le croisement : déplacer des composants voire en ajouter un ? Si le circuit imprimé est double face, vous n'aurez pas beaucoup de choix et le croisement sera nécessairement une catastrophe en termes de rupture d'impédance et de couplage parasite.

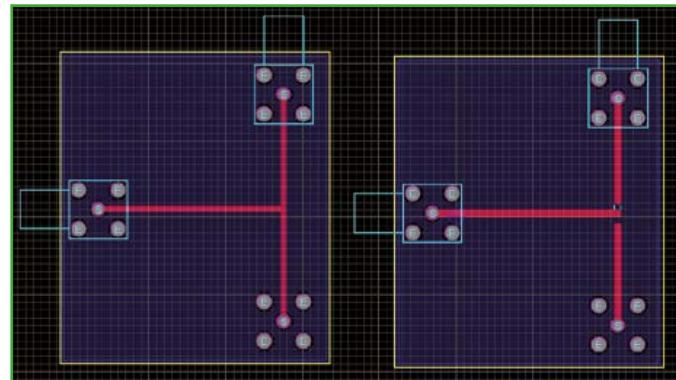


Figure 8 : Dans cet exemple un connecteur peut être implanté soit en haut soit en bas. La configuration de gauche est à éviter car la ligne non utilisée causera une résonance parasite. Celle de droite, en implantant l'une des deux résistances, est bien meilleure.

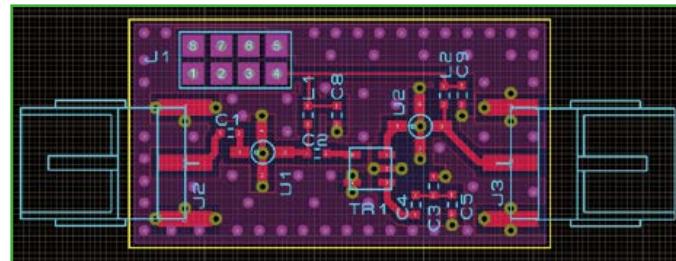


Figure 9 : Des plans de masse supplémentaires peuvent utilement être ajoutés sur toutes les zones libres des couches supérieures et inférieures, mais avec suffisamment de vias de masse pour garantir l'équipotentialité du tout.

Quand deux pistes critiques doivent se croiser, cherchez d'abord s'il n'y a vraiment pas de solution pour éviter le croisement

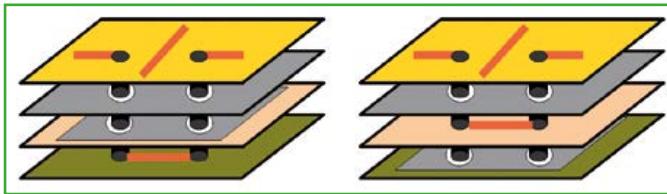


Figure 10 : Deux solutions possibles pour réaliser un croisement de pistes critiques dans le cas d'un circuit à 4 couches : soit en face inférieure soit, mieux, en couche interne en sandwich entre deux plans de masse.

En multicouche, la meilleure solution est de passer l'une des deux pistes sur une couche enfouie, en sandwich entre deux plans de masse (**fig. 10**). Avec des largeurs de pistes bien calculées et plein de vias de masse autour du croisement, cela se passera sans trop de dommages. La même technique peut d'ailleurs s'utiliser pour sortir un signal d'un capot de blindage sans risque de courts-circuits entre la piste et celui-ci.

Pour conclure

Même si vous n'aurez sans doute pas à concevoir de circuit imprimé hyperfréquence tous les jours, toutes ces bonnes pratiques ne sauraient nuire. De plus, vous pourriez en avoir besoin plus souvent que vous ne le pensez, et pas uniquement pour des projets radio. Tout d'abord à cause des vitesses impressionnantes des composants numériques modernes : un FPGA ou un ARM9 à quelques centaines de MHz nécessitent indubitablement quelques précautions de routage. Il en va de même pour la mise en œuvre d'un bus de communication rapide type USB ou Ethernet. Gare aux effets inattendus des composants ultra-rapides : si pour amplifier une horloge lente vous utilisez une porte logique rapide, alors les fronts du signal obtenus seront très raides ; or, qui dit fronts raides, dit harmoniques à très hautes fréquences, d'où de potentiels problèmes de réflexions parasites si les impédances ne sont pas bien gérées. Une anecdote à ce sujet : L'un de mes collègues a développé un projet d'un de nos clients en utilisant des FPGA haut de gamme. Les signaux rapides avaient été routés correctement, mais le port de programmation du FPGA avait été considéré non critique. En effet, la programmation du composant se fait à quelques MHz seulement. Pourtant, sur le premier prototype, impossible de programmer le FPGA. Nous avons perdu quelques jours avant de comprendre que, bien que les signaux de programmation soient lents, les temps de montée des signaux JTAG issus du circuit interne ultra-rapide du FPGA étaient de l'ordre de la nanoseconde, ce qui cause des

réflexions parasites sur les lignes de programmation et *in-fine* un échec de la programmation. L'ajout d'un petit réseau RC pour ralentir les fronts des signaux a résolu le problème, mais il aurait été plus malin de prendre en compte cette contrainte dès la conception...

Ne vous laissez pas effrayer par ces règles, elles relèvent du bon sens, et il mieux vaut les connaître pour éviter des désagréments. Les règles concernant les plans de masse énoncées dans cet article sont aussi celles qui permettent d'optimiser les performances d'un circuit en termes de compatibilité électromagnétique : un plan de masse solide est souvent la seule solution pour espérer passer sans encombre sous les fourches caudines des contraintes du marquage CE. Tiens, le marquage CE... ce sera le sujet de mon prochain article !

(140355)

FORUM : Que diriez-vous de prolonger votre lecture par un échange direct avec moi et d'autres lecteurs sur le forum d'Elektor, dans la nouvelle rubrique **hors-circuits** créée pour cela ? Vos remarques, vos suggestions ou vos questions y sont les bienvenues.
Manifestez-vous : <http://goo.gl/WOImHk>
Cela m'inspirera pour les prochains articles.

Liens

- [1] Les articles précédents de cette mini-série :
adaptations d'impédance | Elektor juin 2014, p. 50
www.elektor-magazine.fr/140079
- lignes microruban** | Elektor juillet 2014, p. 100
www.elektor-magazine.fr/140171
- réflexion de signaux** | Elektor septembre 2014, p. 100
www.elektor-magazine.fr/140269
- [2] A Practical Guide to High-Speed Printed-Circuit-Board Layout
John Ardizzone, Analog Devices
www.analog.com/library/analogdialogue/archives/39-09/layout.html
- [3] High speed board layout guidelines
Altera
www.altera.com/literature/an/an224.pdf
- [4] RF/Microwave PC Board design and layout
Rick Hartley, L-3 Avionics systems
www.jlab.org/accel/eecad/pdf/050rfdesign.pdf

à quoi sert l'impression 3D ?

À lutter contre le cancer avec des matériaux cancérigènes

Dans le numéro 432 (juin 2014 p. 49) j'exprimais mes doutes au sujet de l'impression 3D et sur l'utilité des objets produits. Je vous mettais au défi d'imprimer quelque chose de vraiment utile et de m'en envoyer des photos. J'ai donc reçu des photos et des liens qui d'une part confirment qu'il existe bien des interprétations possibles du mot « utile » et d'autre part montrent que les applications de l'impression 3D sont infiniment variées. Vous verrez sur Elektor.Labs [1] comment les uns impriment en 3D des pièces pour restaurer un avion de la Première Guerre mondiale tandis que les autres produisent des accessoires hyperréalistes pour le décor de leurs modèles réduits ferroviaires. Tout cela est utile, certes... mais le décrochement de mâchoire, c'est ailleurs qu'il s'est produit pour moi. Je m'étais rendu au *Fablab du MakerSpace56* de Vannes, où presque tout le monde semblait mordu d'impression 3D. Et j'y suis tombé sur quelqu'un qui imprime des molécules, à une échelle très fortement agrandie. Le but : montrer une protéine destinée à piéger une autre protéine produite par des cellules cancéreuses. Là j'ai enfin eu l'impression d'être tombé sur une application vraiment utile de l'impression 3D, de nature à changer la vie de très nombreuses personnes.

J'ai également assisté il y a quelques mois à un rassemblement d'impression 3D organisé par RS Composants, et assorti d'un concours. Le résultat ne m'a pas laissé un souvenir impérissable, mais je suis sûr que le modèle 3D de protéine anticancéreuse aurait remporté haut la main ce concours s'il avait été en lice.

Tout cela confirme que l'impression 3D a encore du chemin devant elle avant de devenir l'appareil domestique d'usage universel que nous ont prédit experts et journalistes. Pour l'instant, l'impression 3D coûte cher, elle est très lente et, plus souvent qu'à son tour, elle échoue à produire un objet satisfaisant.

En lançant mon appel, j'avais promis une récom-

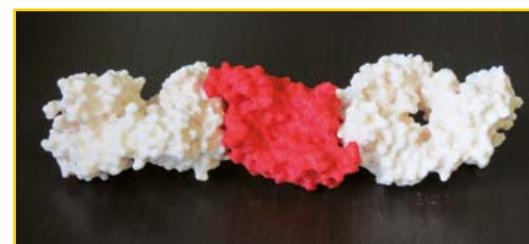
pense. Je ne peux pas l'attribuer à la protéine bretonne puisqu'elle n'a pas participé à mon petit concours. Ce sera donc pour François-Xavier Dufour qui utilise l'impression 3D pour mouler des touches de clavier en silicone [2]. Vous me direz que l'utilité de cette application-là est discutable aussi. Certes, mais c'est l'idée que j'ai trouvée la plus stimulante, car elle ne vise pas à produire directement l'objet final, mais elle est une contribution significative au processus de fabrication. Et j'avoue que j'ai un faible pour les coussinets en silicone. Bravo François-Xavier!

(140048 – version française – Rémi Descistes)

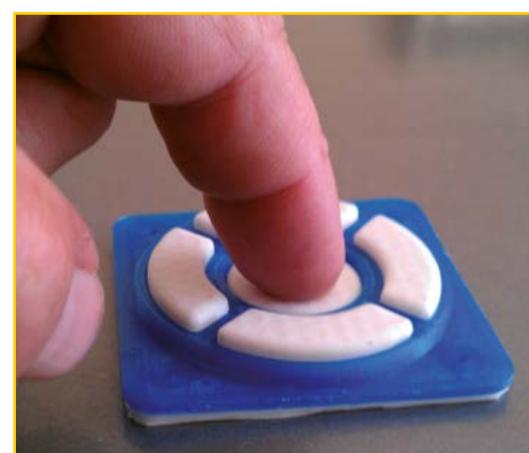
[1] www.elektor-labs.com/node/4056

[2] www.elektor-labs.com/node/4104

Clemens Valens
(Elektor.Labs)



Modèles de protéines imprimées en 3D. La rouge, produite par une cellule cancéreuse, est la molécule ennemie, que les blanches essayent de piéger.



Coussinets en silicone du clavier en rosace de François-Xavier Dufour.

Les photos de vos propres applications de l'impression 3D restent les bienvenues sur Elektor.Labs [1]. Soit vous me les envoyez (labs@elektor.com) soit vous les postezi vous-même.

L'atelier du microcontrôleur (7)

le bus I²C

Burkhard Kainka
(Allemagne)

Vous êtes bloqué sur un projet Arduino parce qu'il vous manque quelques lignes. Avez-vous pensé au bus I²C ? Il n'utilise lui-même que deux fils mais permet d'accéder à pas moins de 127 circuits intégrés externes de tout plumage : duplicateur de ports, capteurs, EEPROM... le catalogue est épais. Ce dernier atelier de notre série se devait donc de vous familiariser avec le bus I²C au moyen de quelques applications intéressantes.

Développé à l'origine par Philips pour connecter de façon simple à un microprocesseur les circuits intégrés (CI) contenus dans ses produits grand public, le bus I²C (ou I2C, pour *Inter Integrated Circuit*) sert à transmettre des données sur deux fils au moyen d'une ligne de données et d'une ligne d'horloge. Prenons l'exemple d'un téléviseur dont le microprocesseur central doit commander plusieurs sous-ensembles. Comment ce processeur peut-il les superviser avec un seul bus de données à seulement deux fils ? Simplement en choisissant comme maître le processeur (comme dans le cas d'un bus SPI), et comme esclaves les composants périphériques. Chaque esclave possède une adresse représentée sur 7 bits, seul moyen de relier plusieurs CI à un même bus sans s'exposer à un risque de collisions. Outre les classiques RAM, EEPROM, duplicateurs de ports, horloges temps réel, convertisseurs AN et NA, il existe un grand nombre de puces I²C spécialisées, p. ex. des pilotes d'afficheur ou encore des PLL.

Transfert de données et adressage

Le bus I²C utilise une ligne de données série SDA et une ligne d'horloge SCL. Comme dans le cas des registres à décalage (cf. l'article précédent), un bit est transféré sur la ligne de données à chaque coup d'horloge, d'abord les bits d'adresse, puis les bits de données. Les deux lignes sont dotées d'une résistance de rappel vers le haut et peuvent être mises au niveau bas par chaque participant du bus (**fig. 1**). Seul le maître peut fournir le signal d'horloge. Les données peuvent

quant à elles provenir aussi bien du maître que des esclaves.

Le bus I²C fonctionne indifféremment avec les microcontrôleurs ou puces alimentés en 5 V ou 3,3 V. On peut même relier les deux à un seul et même bus ! Les deux résistances de rappel, p. ex. de 2,2 kΩ, forcent les lignes au niveau haut (3,3 V ou 5 V) tant qu'un participant ne force pas les lignes au niveau bas avec ses transistors de rappel. Le système « 5 V » voit aussi 3,3 V comme un niveau haut car cette valeur se situe bien au-dessus de la moitié de la tension d'alimentation (2,5 V). Et le niveau bas est partout 0 V. Nous pouvons ainsi relier une carte Uno à un esclave alimenté en 3,3 V. Une aubaine, puisque de nombreux CI ne fonctionnent que sous 3,3 V. L'ATmega328 de la Uno fournit une interface I²C matérielle sur les ports PC4 et PC5. Bascom dispose quant à lui de commandes permettant de réaliser une interface I²C logicielle sur certains ports (et nous pouvons bien sûr aussi écrire des fonctions pour placer les lignes au niveau haut ou bas). Nous exploiterons uniquement l'interface I²C logicielle de Bascom, mais avec les mêmes broches que celles de l'interface I²C matérielle de l'ATmega.

Ces ports sont également utilisés sur le shield Elektor conçu pour la carte Uno [1]. Ils mènent au connecteur ECC/Gnublin (K2) auquel peuvent être connectés via un câble plat des modules d'extension Gnublin, p. ex. le module équipé de

huit relais [2]. Les deux lignes du bus I²C (broches 5 et 6 de K2) possèdent deux résistances de 330 kΩ. Elles auraient pu être omises, mais elles protègent contre les faux signaux que peuvent provoquer les réflexions dans les lignes longues, et elles diminuent les risques d'erreurs susceptibles d'apparaître en présence de deux tensions d'alimentation différentes. Si p. ex. vous mettez par mégarde la ligne PC4 au niveau haut en permanence, un courant trop fort pourrait circuler dans un périphérique de 3,3 V ; avec la résistance, l'intensité est limitée.

Le protocole I²C définit un certain nombre de situations qui permettent à chaque participant du bus de reconnaître le début et la fin d'une transmission :

- état de repos : SDA et SCL sont au niveau haut et donc inactives. La commande Bascom I2cinit place ces deux lignes à l'état de repos, mais sans résistances de rappel car elles se trouvent à l'extérieur ;
- condition de départ : SDA est forcée au niveau bas par le maître, tandis que SCL reste au niveau haut (Bascom : I2cstart) ;
- condition d'arrêt : SDA passe du niveau bas à haut, tandis que SCL reste au niveau haut (Bascom : I2cstop) ;
- transfert de données : l'émetteur place 8 bits de données sur la ligne de données SDA ; cet octet est transféré sur impulsion d'horloge fournie par le maître sur SCL. Le bit de poids fort est le premier transféré (Bascom : I2cbyte val, avec val = variable ou constante) ;
- acquittement (ACK, pour *acknowledge*) : le destinataire acquitte la réception d'un octet par mise au niveau bas de la ligne SDA en attendant que le maître fournisse la nouvelle impulsion d'horloge. L'acquittement signifie en même temps qu'un nouvel octet attend d'être reçu. Si l'émetteur souhaite mettre fin à la transmission, il doit envoyer un bit de non-acquittement (NACK), qui correspond cette fois-ci à une ligne SDA laissée au niveau haut. La fin de la transmission est effective lorsque la condition d'arrêt est réalisée (Bascom : i2cbyte var, Ack ou i2cbyte var, Nack).

Les adresses sont transmises et acquittées comme le sont les octets de données. Prenons l'exemple d'un transfert de données du maître vers un esclave, disons un port de sortie : le maître crée la condition de départ, puis transfère les bits 7 à 1 constituant l'adresse du port ;

le bit 0 doit quant à lui coder la « direction » du transfert, ici un 0 pour « écriture ». L'esclave destinataire acquitte l'octet d'adresse, puis le maître envoie l'octet de données, qui sera de même acquitté ; à cet instant le maître peut soit interrompre la transmission en créant une condition d'arrêt, soit envoyer d'autres octets à l'esclave.

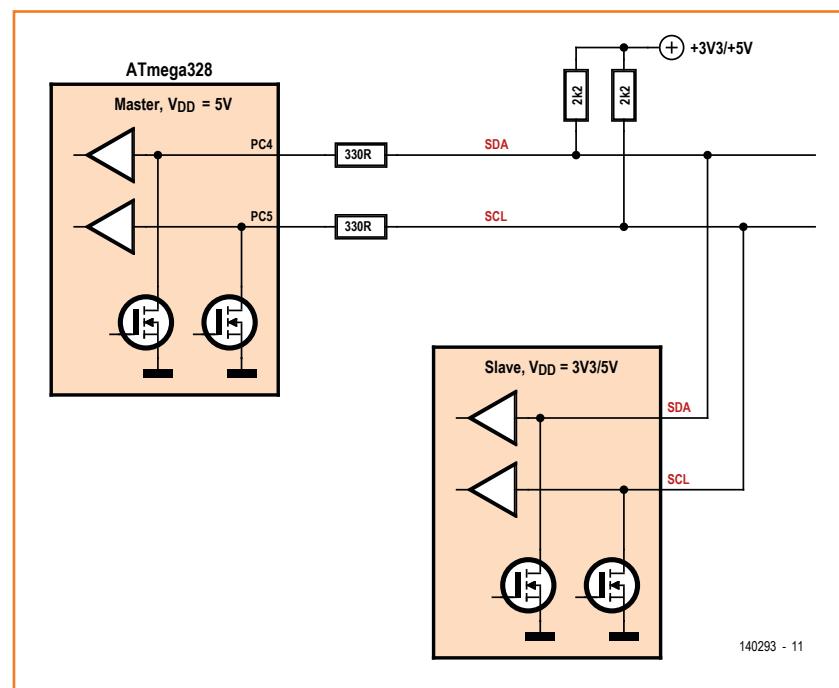
Adresse I²C avec bit d'écriture/lecture

A6	A5	A4	A3	A2	A1	A0	R/W
----	----	----	----	----	----	----	-----

Pour lire des données en provenance d'un esclave, le maître doit transmettre l'adresse correspondante avec un bit positionné sur « lecture ». Il produit alors huit impulsions d'horloge pour recevoir les 8 bits de l'octet de données. Il pourra ensuite recevoir d'autres octets en acquittant la réception avec un bit ACK à la neuvième impulsion d'horloge. Il met fin à la transmission en positionnant le bit d'acquittement à NACK et en créant la condition d'arrêt.

Chaque composant I²C possède une adresse fixe. Une partie de cette adresse est invariable et dépend du type du composant, mais l'autre partie peut être définie à l'aide des lignes d'adresse A0, A1, etc. Placer ces lignes au niveau haut ou bas revient en effet à mettre à 1 ou à 0 les bits correspondants de l'adresse. Les 3 broches d'adresse du duplicateur de port PCF8574 per-

Figure 1.
Liaisons I²C entre le maître et les esclaves.



Listage 1.
Test des adresses I²C actives.

```
'-----  
'UNO_I2C1.BAS Test for valid Adresses  
'-----  
$regfile = "m328pdef.dat"  
$crystal = 16000000  
$baud = 9600  
  
Dim Addr As Byte  
...  
Config Scl = Portc.5  
Config Sda = Portc.4  
I2cinit  
  
For Addr = 2 To 254 Step 2  
    I2cstart  
    I2cwbyte Addr  
    If Err = 0 Then  
        Print Addr  
        Locate 2 , 1  
        Lcd Addr  
        Waitms 1000  
    End If  
    I2cstop  
    Next Addr  
End
```

- Convertisseur AN PCF8591 : $90_{\text{hex}} = 144$
- RAM et EEPROM : $A0_{\text{hex}} = 160$
- Circuit d'horloge PCF8583 : $A0_{\text{hex}} = 160$

Nombreux sont les CI dont les trois derniers bits d'adresse peuvent être librement configurés. L'horloge temps réel PCF8583 possède la même adresse interne fixe qu'une RAM car elle en contient une elle-même. Il faut donc, si l'on souhaite utiliser une horloge et une RAM sur le même bus, attribuer à chacune une adresse différente.

Parfois source de confusion voire de ratés, l'adresse peut se présenter sous deux formes « concurrentes ». Certaines fiches techniques ne donnent qu'une adresse sur 7 bits en omettant le bit de lecture/écriture. Sous cette forme, l'adresse du PCF8574 est 32 (20_{hex}). Pour Bascom en revanche, la puce a 64 comme adresse d'écriture ($40_{\text{hex}} = \&H40$) et 65 pour adresse de lecture ($\&H41$).

Pour déterminer les adresses des CI connectés, le programme du **listage 1** (à télécharger [3]) frappe à toutes les portes et demande s'il y a quelqu'un. Après l'adressage d'un composant, la variable système *ERR* de Bascom (non dimensionnée avec *Dim*, car Bascom l'a déjà déclarée) contient un 0 si le composant a répondu par un bit d'acquittement, un 1 dans le cas contraire. Un 0 signifie donc que l'adresse était valide. Le programme frappe uniquement aux portes d'adresses paires comprises entre 2 et 254, car les adresses impaires sont simplement les « adresses de lecture » correspondant aux adresses paires. Pour le circuit de la **figure 2**, les adresses 64, 144, 160 et 162 ont répondu comme attendu.

Le protocole I²C recèle une autre chausse-trappe :

mettent ainsi de construire 8 adresses différentes, donc de câbler jusqu'à 8 de ces composants sur un même bus I²C. Le PCF8574 fournit 8 sorties numériques ; les niveaux appliqués sur ces sorties sont transmis à la puce qui les interprète comme les 8 bits d'un octet de données. Le PCF8574 a pour adresse interne fixe $40_{\text{hex}} = 64$. Quelques adresses fixes de composants Philips de la première heure :

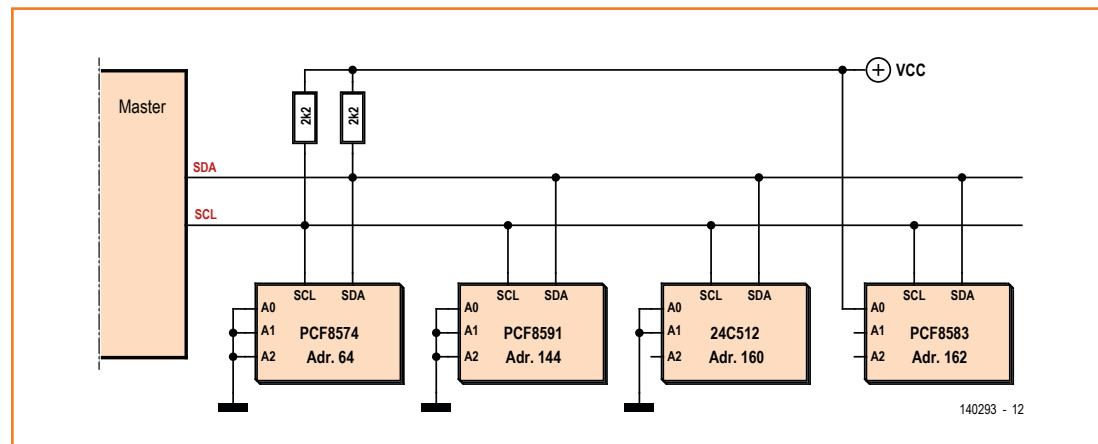


Figure 2.
Un maître et quatre esclaves.

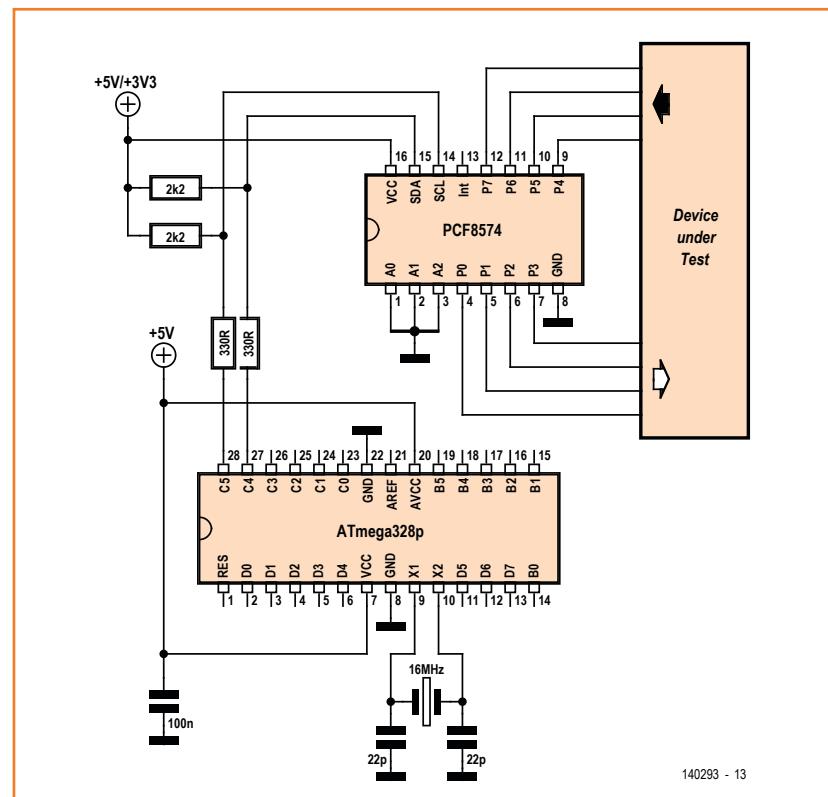
chaque participant du bus peut bloquer le maître lorsqu'il a besoin d'un peu plus de temps. Pour cela il force la ligne d'horloge SCL au niveau bas de façon à obliger le maître à attendre que la ligne soit de nouveau libre. Bascom respecte cette convention, mais au risque de laisser un programme planter si aucune résistance de rappel n'est présente. Est-ce parce que certaines applications nous ont habitués à pouvoir tester un programme sans qu'aucun matériel ne soit connecté, toujours est-il qu'il peut nous arriver de lancer un programme et d'attendre sur l'écran de notre oscilloscope les signaux des lignes SCL et SDA... alors que nous n'avons pas encore connecté le câble plat, p. ex. à une carte Gnublin. Et nous sommes là à attendre Godot sur la ligne I²C, car les résistances I²C de rappel vers le haut se trouvent sur la carte Gnublin, et en leur absence aucun signal ne se manifeste.

Le duplicateur de ports PCF8574

Le duplicateur PCF8574 dispose de 8 sorties bidirectionnelles. Pas de commutation de la direction des données ici, les broches sont équipées de résistances internes de rappel vers le haut qui définissent un niveau haut à l'état de repos. Le port est donc à l'état 255 (11111111) à la mise sous tension, et il n'y aura de 0 que si la ligne correspondante a été mise à la masse. Les lignes peuvent être utilisées comme entrées ou sorties ; toutes les lignes de lecture doivent d'abord être mises à l'état haut comme s'il s'agissait de sorties.

Le schéma de la **figure 3** montre le câblage du bus et du PCF8574, utilisé ici comme testeur numérique. Le duplicateur peut être alimenté en 3,3 V ou en 5 V, selon la tension d'alimentation du dispositif à tester. Les sorties P0 à P3 peuvent commander un circuit numérique quelconque doté de 4 entrées. On pourra alors, p. ex., incrémenter toutes les broches de façon à couvrir l'ensemble des combinaisons possibles pour les entrées du circuit. On testerait de la même façon un câble ou un connecteur, et n'importe quel court-circuit ou coupure serait détecté.

Le **listage 2** illustre cette utilisation mixte des entrées et des sorties. Les broches P0 à P3 sont les quatre sorties et sont incrémentées dans la boucle. Les lignes P4 à P7 servent d'entrées et doivent donc être mises au niveau haut durant le processus d'écriture (0r &B11110000). La puce est adressée deux fois, une fois en écri-



140293 - 13

ture (adresse 64), une fois en lecture (adresse 65). Cette incrémentation des quatre sorties est la variante la plus simple d'un testeur. Selon sa nature, le dispositif à tester peut également nécessiter un tout autre procédé, et par conséquent un autre programme.

Duplicateur d'E/S à 16 bits PCA9555

Disposer de huit lignes supplémentaires n'est pas toujours suffisant. Le module *Gnublin IOExpander* [2] offre 16 ports d'E/S grâce au duplicateur PCA9555 de NXP. Cette carte peut être directement reliée au shield Elektor via le connecteur EEC. Le module Gnublin « 8 relais » est lui aussi équipé du PCA9555.

Tout comme NXP est le successeur de Philips, le PCA9555 est le successeur du PCF8574, ce qui explique qu'il possède la même adresse interne : 64 (&H40). L'espace d'adressage est très limité, ce recyclage d'adresse n'a donc rien de surprenant. PCF8574 et PCA9555 sont un peu le Minitel et le PC des duplicateurs. Le plus récent possède en effet deux fois plus de lignes de port et offre en outre d'autres fonctions, comme une commutation de la direction des données et une fonction d'inversion de l'état des ports lus.

Figure 3.
Schéma du testeur avec PCF8574.

Listage 2. Lecture/écriture des ports du PCF8574.

```
'-----  
'UNO_I2C2.BAS input/output PCF8574  
'-----  
  
$regfile = "m328pdef.dat"  
$crystal = 16000000  
$baud = 9600  
  
Dim N As Byte  
Dim D As Byte  
...  
Do  
    For N = 0 To 15  
        I2cstart  
        I2cwbyte 64      'write  
        D = N Or &B11110000  
        I2cwbyte D  
        I2cstop  
        Print N;  
  
        Print " ";  
        Locate 2 , 1  
        Lcd N  
        Lcd " "  
  
        I2cstart  
        I2cwbyte 65      'read  
        I2crbyte D , Nack  
        I2cstop  
        Shift D , Right , 4  
        Print D  
        Locate 2 , 5  
        Lcd D  
        Lcd " "  
        Waitms 100  
        Next N  
    Loop
```

La **figure 4** montre comment relier le duplicateur au shield Elektor via le connecteur EEC. Les lignes du bus I²C se relient à l'alimentation de 3,3 V avec un câble plat. Les deux résistances de rappel de la carte Gnublin s'activent au moyen de deux cavaliers. Ces cavaliers sont à placer lorsqu'une seule carte est connectée. Si plusieurs

cartes sont reliées à la carte Gnublin, p. ex. au moyen de la carte de répartition *Bridge Module* [2], les résistances de rappel ne doivent être activées que sur une seule des cartes. Exploiter plusieurs duplicateurs de ports est possible car les broches d'adresse A0 à A2 sont elles aussi équipées de cavaliers sur la carte. Les trois lignes

Listage 3. Utilisation du duplicateur de ports PCA9555.

```
'-----  
'UNO_LCD1.BAS input/output PCA9555  
'-----  
  
$regfile = "m328pdef.dat"  
$crystal = 16000000  
$baud = 9600  
  
Dim N As Byte  
Dim D As Byte  
...  
I2cstart  
I2cwbyte 64      'Gnublin port expander  
I2cwbyte 6  
I2cwbyte 0      'Port 0 output  
I2cwbyte 255     'Port 1 input  
I2cstop  
  
Do  
    For N = 0 To 255  
        I2cstart  
        I2cwbyte 64      'write  
        I2cwbyte 2  
        Print N;  
  
        Print " ";  
        Locate 2 , 1  
        Lcd N  
        Lcd " "  
  
        I2cstart  
        I2cwbyte 65      'read  
        I2crbyte D , Nack  
        I2cstop  
        Shift D , Right , 4  
        Print D  
        Locate 2 , 5  
        Lcd D  
        Lcd " "  
        Waitms 100  
        Next N  
    Loop
```

sont reliées par défaut à la masse, ce qui équivaut à une adresse de bus de 64. Comme il est possible de relier 8 cartes d'adresses 64 à 72, avec 16 lignes de ports par carte nous disposons de 128 broches supplémentaires.

Le programme du **listage 3** permet de tester d'autres circuits. Comme dans l'exemple précédent, les broches sont réparties en deux groupes : une moitié pour la sortie, l'autre pour l'entrée. Pour traiter l'ensemble, plus complexe ici, nous devons systématiquement envoyer un « octet de commande » après l'adresse [4]. Cet octet sert de pointeur, autrement dit détermine le registre à lire ou à écrire. Nous pouvons ensuite envoyer ou recevoir 1 ou 2 octets. Pour configurer p. ex. les ports 0 (8 broches) et 1 (les 8 autres broches), on envoie l'octet « 6 » après l'adresse, puis on écrit les 2 octets qui définiront le sens des broches dans les registres de configuration 6 et 7. Des octets à 0 configurent les broches correspondantes en tant que sorties, des 1 en tant qu'entrées. Notre programme définit toutes les lignes du port 0 comme des sorties, celles du port 1 comme des entrées. En passant, « entrées » signifie la présence dans la puce d'une résistance d'environ $100\text{ k}\Omega$, donc de 1 lus lorsqu'il ne circule pas de courant. Les octets de commande sont :

- 0 Entrée Port 0
- 1 Entrée Port 1
- 2 Sortie Port 0
- 3 Sortie Port 1

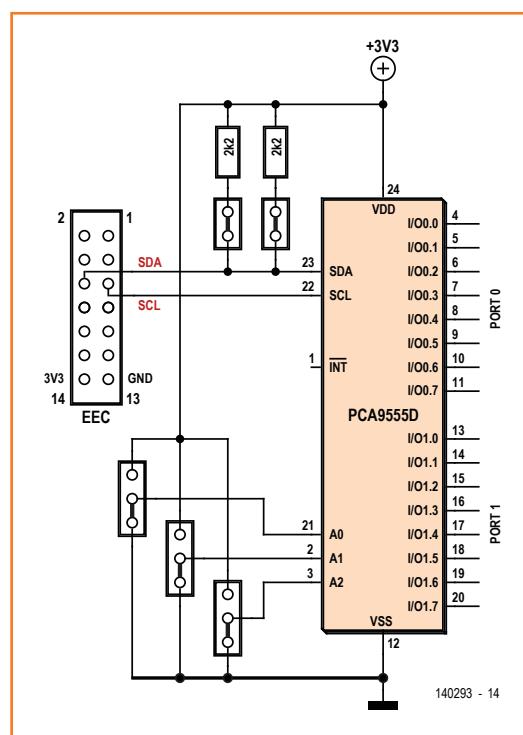


Figure 4.
Le PCA9555 sur la carte Gnublin.

- 4 Inversion polarité Port 0
- 5 Inversion polarité Port 1
- 6 Configuration Port 0
- 7 Configuration Port 1

Le code répète les commandes « 2 » (Sortie Port 0) et « 1 » (Entrée Port 1) dans la boucle d'écriture/lecture des données. Le composant doit être adressé une nouvelle fois à chaque

Quelques composants I²C d'intérêt

Presque aussi inévitables dans les pages d'Elektor que les miettes autour d'un mangeur de biscotte, les composants dotés d'une interface I²C permettent de mener à bien toutes sortes de projets. En voici quelques-uns :

- EEPROM I²C jusqu'à 64 Ko (ex. : la 24C512) pour construire p. ex. un enregistreur de données.
- CY27EE16, un générateur d'horloge programmable, utilisé dans la série *radio logicielle* avec AVR. La commande par microcontrôleur élargit un peu plus son champ d'application.
- SI4735, un récepteur AM/FM complet, lui aussi exploité à maintes reprises dans Elektor avec Bascom. Un petit nombre de commandes I²C suffit pour régler chaque fréquence.
- Les convertisseurs AN et NA à haute résolution ont souvent une connexion I²C. Ex : le convertisseur AN à 16 bits ADS1115, présenté récemment dans le magazine.

En potassant le sujet dans toutes ses largeurs, on peut même construire son propre composant I²C ! Une bibliothèque Bascom permet d'ailleurs de programmer un esclave I²C. L'affaire se révèle toutefois plus délicate que pour un maître, car l'esclave doit pouvoir composer avec la vitesse élevée imposée par le maître.

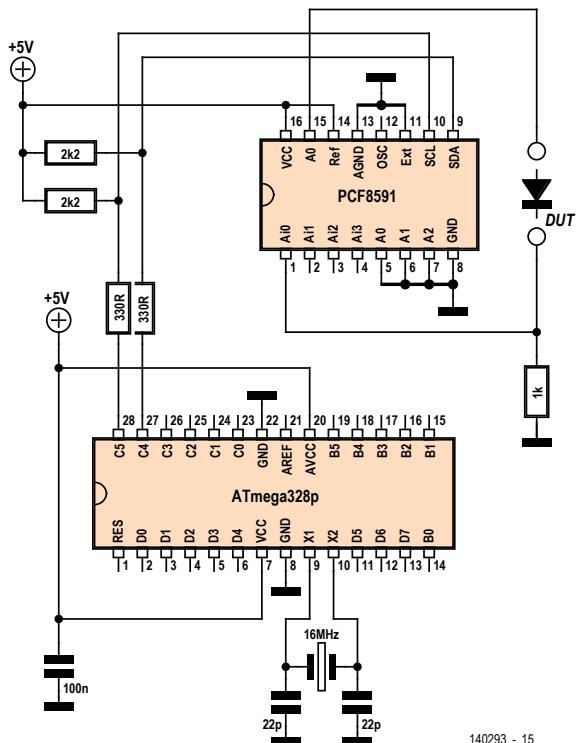


Figure 5.
Un testeur de diodes avec le
PCF8591.

Listage 4. Testeur de diodes avec le PCF8591.

```

'-----  

'UNO_I2C4.BAS AD/DA PCF8591  

'  

$regfile = "m328pdef.dat"  

$crystal = 16000000  

$baud = 9600  

Dim N As Byte  

Dim D As Byte  

Dim U As Word  

...  

N = 0  

Do  

    I2cstart  

    I2cwbyte 144           'write  

    I2cwbyte 64            'DA enable  

    I2cwbyte N  

    Print N;  

    Print " ";  

    Locate 2 , 1  

    Lcd N  

    Lcd " "

```

commande, et également pour chaque lecture d'un port, mais cette fois-ci en lecture, donc avec un bit d'écriture/lecture à 1 (adresse 65). À regarder le code de près, n'y manque-t-il pas un I2cstop ? Non, il s'agit ici d'un « départ répété » sans condition d'arrêt préalable, car les deux accès vont de pair : écriture du registre à lire, puis lecture.

Entrée/sortie analogique avec le PCF8591

Le PCF8591 est un convertisseur AN à 8 bits et 4 entrées, et dans le même temps un convertisseur NA à 8 bits et 1 sortie. Le convertisseur AN n'a que peu d'intérêt car la carte Uno dispose déjà de suffisamment d'entrées analogiques, qui plus est dotées d'une meilleure résolution. Le convertisseur NA est plus utile : outre une sortie MLI, il peut délivrer une vraie tension continue. C'est à partir de cette brique NA que j'ai monté un simple testeur de diodes capable de déterminer la tension directe en fonction du courant (**fig. 5**). Il est alimenté en 5 V de façon à pouvoir aussi être utilisé avec des LED tolérant une tension directe élevée.

Le composant possède un registre de commande dont le contenu doit être envoyé immédiatement

après l'adresse de bus. Un octet de commande « 64 » met le convertisseur NA sous tension et sélectionne le canal d'entrée 0. L'octet suivant parvient au registre NA et détermine la tension de sortie : 0 équivaut à 0 V, 255 à 5 V. Avec une résolution de 8 bits, le pas de mesure vaut environ 20 mV. Le programme du **listage 4** produit une rampe de tension croissante tout en mesurant la tension aux bornes de la résistance de shunt de 1 kΩ. Pour cela le PCF8591 doit à nouveau être adressé en écriture (adresse 145). L'octet lu représente la tension en A/I0. Une valeur de 50 correspond à une tension de 1 V, et en conséquence à un courant de 1 mA. La rampe est stoppée une fois cette valeur atteinte. La tension directe est obtenue en calculant la différence entre les tensions de sortie et d'entrée, puis affichée en mV. Le LCD a affiché les valeurs suivantes pour une LED couleur bleu-vert :

1 mA
2920 mV

On ferme !

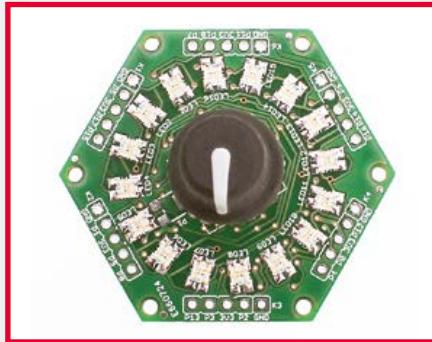
L'atelier du microcontrôleur ferme ses portes, mais nous ne vous laisserons pas seuls dehors puisque d'autres applications Bascom pour la carte Uno et le shield Elektor sont prévues.

En attendant, peut-être les projets de cette série vous auront-ils inspiré d'autres programmes ? Si c'est le cas, n'hésitez pas à les partager sur www.elektor-labs.com ! Y sont volontiers accueillis les petits projets ou applications encore au stade de développement.

(140293 – version française Hervé Moreau)

Liens

- [1] www.elektor-magazine.fr/140009
- [2] www.elektor.fr/developpement/gnublin
- [3] www.elektor-magazine.fr/140293
- [4] www.nxp.com/documents/data_sheet/PCA9555.pdf



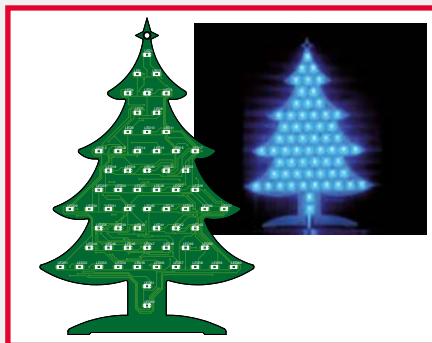
Cool Controller Concept : CoCo-ri-Co

Sur beaucoup d'appareils, un ou deux boutons de commande, astucieusement associés à un afficheur, suffisent ; nous proposons donc cette carte hexagonale universelle, facile à monter en un ou plusieurs exemplaires derrière la façade de n'importe quel projet personnel. La position du codeur rotatif (sans fin) avec bouton poussoir est indiquée sur une couronne de 16 LED bicolores. Cocorico !



Afficheur VariLab 402

L'alimentation de labo 40 V / 2 A annoncée le mois dernier est prête, mais l'analyse minutieuse du schéma est si longue que même scindée en deux, elle ne tenait pas dans ce numéro-ci. Le 1er volet sera pour le mois prochain. Le 2^e volet sera entièrement consacré à l'afficheur LCD à 4 lignes universel. Il est facile d'y faire défiler de nombreux paramètres grâce à deux codeurs rotatifs avec poussoir.



arbre de Noël programmable

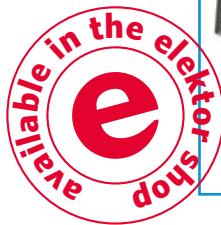
Si vous cherchez quelque chose d'inédit et d'original pour renouveler le genre de la guirlande lumineuse, vous êtes à la bonne adresse chez Elektor. Avec notre partenaire Eurocircuits, fabricant de circuits imprimés, nous vous proposerons un arbre de Noël fait d'une matrice de 62 LED, aux motifs lumineux libres, programmables très facilement, directement depuis le clavier de votre PC.

Sous réserve de modification

Publication du numéro de décembre : 18 novembre

kit de développement IdO WiSmart

Jan Buiting
(Elektor)



Beaucoup des produits amenés par la vague de l'internet des objets (IdO) sont des boîtes noires, incompatibles avec la recherche ou le développement de produits. Si vous voulez faire la différence et travailler au niveau de la puce sans abandonner le logiciel et les outils libres, il vous faut autre chose, quelque chose d'ouvert.

La puce EC19D 'WiSmart' d'Econais, entreprise gréco-californienne, devrait attirer les ingénieurs confirmés et les amateurs souhaitant développer des produits astucieux pour l'IdO – des produits primo intelligents, secundo respectueux de l'environnement et tertio utilisant le Wi-Fi. Le **EC19D01DK**, nouveau kit d'Econais, propose une approche prometteuse pour développer vos propres appareils IdO en petits ou gros volumes.

Les outils standard se débrouillent très bien

En ne vous imposant pas d'emblée de télécharger une suite d'outils propriétaires, Econais se distingue de la concurrence. À l'exception d'un petit programme Python nommé *SimpleCom.py*, leur processus de développement, débogage et implantation d'un appareil IdO avec Wi-Fi, repose entièrement sur des noms et des fabricants dont la familiarité et/ou la gratuité/liberté vous réjouiront : Linux, Windows, SPI, UART, JTAG, jeu de commandes AT, FTI USB-TTL, carte SD, Open Hardware, Gerber, Python.

Après vous être identifié sur le site d'Econais [1], vous pourrez télécharger l'abondante documentation sur la puce EC19D et le kit de développement associé EC19D01DK (lancé fin août 2014). Elektor était parmi les premiers destinataires du nouveau kit. La version DK remplace son ainé et comprend, pour quasiment le même prix, une carte de débogage.

Du TTL-TxD/RxD au MAC personnalisé

Vous trouverez dans la boîte, en plus des classiques câbles et d'une alimentation universelle, la carte Wi-Fi EC19D00SD (1), la carte d'extension EC19D01EX (2), et la carte de débogage JTAG EC19D01DBG (3). Vous trouverez aussi des

feuillets d'aide au démarrage dont la concision paraît en contradiction avec le nombre de questions qui vous assaillent à ce stade-là.

La doc est un peu spartiate, mais le potentiel éducatif du kit est réel. Le premier obstacle est toutefois significatif : la carte -EX ne m'a pas semblé très bavarde au début. Il y a deux solutions, suivant ce que vous souhaitez voir sortir de la puce EC19D (à travers la carte -EX) : dialogue « machine » ou « humain ». Le premier n'est que commandes et nombres tandis que le deuxième n'est pas très différent des commandes AT. Les deux méthodes utilisent un port série virtuel FTDI USB-TTL (RxD-TxD-CTS-RTS) et un émulateur de terminal tel que *RealTerm* (Windows) ou *Cuteterm* (Linux) pour l'approche à l'ancienne, ou la méthode à la Python avec *SimpleCom*. Ce dernier nécessitera l'installation de Python.

Une fois la communication établie, votre ARM, RPi, AVR ou Arduino pourra prendre le contrôle du port SPI/UART sur la carte -EX et en route pour l'Ido Wi-Fi ! Tout le trafic est déchiffrable par un humain : il consiste en commandes, mots de données et paramètres.

Dans une installation sophistiquée, les cartes -SD, -EX et -DBG seront reliées comme sur la photo et liées à un PC avec un câble USB. Cet assemblage n'est pris en charge que sous Linux.

La carte -DBG servira également de programmeur flash avancé pour la puce EC19D. C'est un produit relativement nouveau et les informations techniques et exemples d'applications manquent. Si vous avez de grandes ambitions en programmation d'objets connectés, ce kit est pour vous. Voici une courte liste des réjouissances qui vous attendent : mise à jour sans fil (OTA, *over the air*), mesures intelligentes, capture et enregistrement de données avec *Wireshark*, vous pourrez aussi réveiller les appareils Wi-Fi cachés ou endormis en utilisant le mode *Wi-Fi Direct* et l'application *ProbMe* sur votre téléphone tactile Android.

Conclusion

AU prix de quelques efforts, le EC19D01DK permet d'aborder le développement d'une application IdO sous tous ses angles : programmation et soudure, Linux et Windows, USB et E/S sur microcontrôleur. Tout dans ce kit vous laissera l'impression d'être l'explorateur d'un monde nouveau ; la tonne d'exemples de programmation Linux vous le confirmera.

L'équipe d'Econais s'est mise en cheville avec



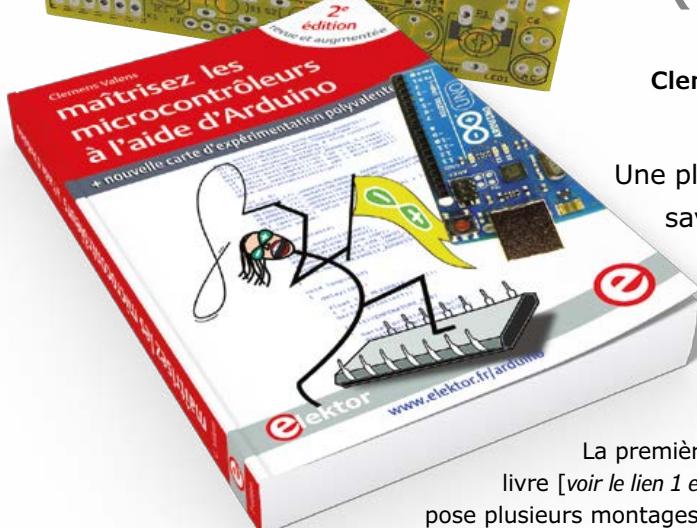
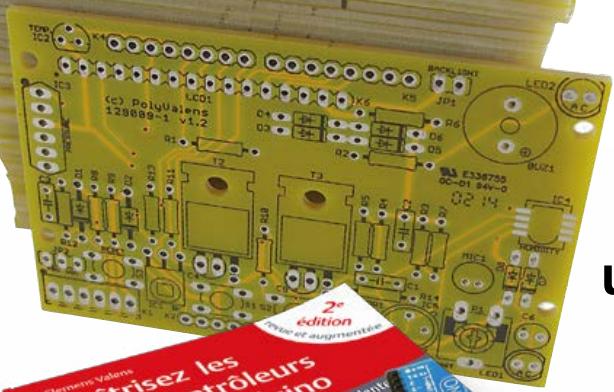
celle Elektor afin de vous proposer leur dernier produit à un prix réduit, réservé à nos lecteurs. Oui, le EC19D01DK est disponible exclusivement depuis l'e-choppe d'Elektor [2]. Je ne serais pas surpris que ce soit un lecteur d'Elektor qui le premier nous éblouira avec une application IdO époustouflante basée sur le EC19D.

(140338 – version française : Kévin Petit)

Liens

- [1] www.econais.com
(suivre Support à Dev Kit Resources)
- [2] www.elektor.com





shield polyvalent

un(e) caméléon(e) pour Arduino

Clemens Valens (Elektor Labs)

Une plaque d'essais à contacts à ressort, c'est bien... pour essayer, mais, si tu veux garder le fruit de tes essais, ce n'est pas dépourvu d'inconvénients. Que dirais-tu d'un *shield* Arduino conçu justement pour l'expérimentation et qui s'adapte aux besoins les plus divers, même les tiens ?

La première édition de mon livre [voir le lien 1 en fin d'article] propose plusieurs montages pour illustrer les techniques de programmation enseignées. Certains sont simples et faciles à essayer même sans fer à souder. D'autres sont plus complexes ; les câbler sur une plaque de prototypage est un travail assez fastidieux. Pour cette raison, j'ai décidé de dessiner, pour la seconde édition, un **vrai circuit imprimé** qui permet d'essayer, sur du matériel réel et de manière confortable, les techniques de programmation apprises dans le livre. Cette platine est décrite en détail dans le chapitre 11 de la deuxième édition du livre. Elle est également disponible dans l'e-choppe d'Elektor [lien 2]. Je me suis débrouillé pour que cette platine soit aussi polyvalente que possible. Elle permet bien sûr d'essayer un grand nombre des circuits décrits dans le livre et réunis sur le schéma ci-contre (**fig. 1**), mais pas uniquement. Sa polyvalence rend possible son utilisation dans bien d'autres applications. Télécharge le PDF de cette carte et examine-le, tu remarqueras que sa polyvalence a une conséquence inhabituelle : les composants

dont l'empreinte est sérigraphiée sur le circuit imprimé ne peuvent pas y être implantés tous à la fois. Ce serait physiquement impossible, car certaines pastilles sont communes à des composants distincts. Là où il n'y a qu'un trou, on ne peut évidemment mettre qu'un seul composant. Autrement dit, selon que tu réaliseras tel ou tel montage, certains composants seront implantés, d'autres pas, comme le montre la **série de plans d'implantation** de composants ci-contre à droite. Tout cela est expliqué dans le livre.

Tu captes ?

Tu pourras équiper ta carte polyvalente des capteurs suivants :

- température (DS18B20)
- pression atmosphérique (MPX4115A)
- humidité (SHT11)
- lumière (une photorésistance)
- lumière infrarouge (MLX90614AAA)
- télécommande (TSOP34836)
- son (un microphone avec amplificateur)
- pression de doigt (*naaaaan*, pas un écran tactile, mais de simples boutons-poussoirs)

Elektor te veut du bien

Pour faciliter tes premiers pas et t'encourager, Elektor te propose le livre et la carte, mais aussi un kit avec les composants pour réaliser trois des montages décrits dans le livre :

129009-71 – kit gradateur/émeutier/nuisance sonore. Ce sera un investissement modeste, car Elektor te veut du bien mais ne veut pas ton bien. Bon investissement, mais gare aux bénéfices... n'en parle surtout pas à ton percepteur, il te ferait probablement payer illico l'impôt sur ta bonne fortune. Un rendement pareil, c'est scandaleux ! Commence peut-être par jeter un coup d'œil aux vidé(m)os [4] que j'ai préparées pour toi.

- un potentiomètre (je le mentionne même si ce n'est pas vraiment un capteur... en fait, n'est-ce pas un détecteur de mouvement ?)

Côté actionneurs, elle t'offre :

- 2 transistors de puissance (MOSFET ou bipolaires) avec diodes de roue libre pour commander des relais par exemple
- 2 LED (couleur ou longueur d'onde au choix)
- 1 ronfleur piézo
- 1 afficheur à cristaux liquides (deux lignes de seize caractères)

Pour compléter la carte, il y a un (petit) bornier à vis à six contacts, partagés entre :

- une entrée analogique avec filtre de bruit, une protection contre les surtensions et une limitation de courant
- deux entrées tout ou rien avec protection contre les surtensions
- trois sorties tout ou rien avec limitation de courant

- deux sorties de puissance
 - un port série
 - les alimentations VCC, +3,3 V et GND.
- VCC peut être +5 V ou VIN de la carte Arduino (à choisir avec un cavalier sur JP2)

Regarde bien le schéma ci-dessous, tu remarqueras quelques composants bizarrement connectés, comme R14 ou C4 et C5. Ces deux derniers servent surtout pour l'exemple de l'émetteur DCF77 décrit dans le livre (**photo** page suivante) et n'ont (probablement) pas d'utilité dans d'autres applications. R14 peut être insérée entre l'émetteur du transistor T1 et la masse, ce qui rend cet amplificateur plus universel. En plus, elle permet l'utilisation d'un transistor PNP pour T1, ce qui donnera encore plus de flexibilité.

R10 et R11 connectées aux grilles des MOSFET T2 et T3 sont en fait des résistances de base au cas où tu auras choisi des transistors bipolaires NPN à la place des MOSFET. Sur la platine, tu

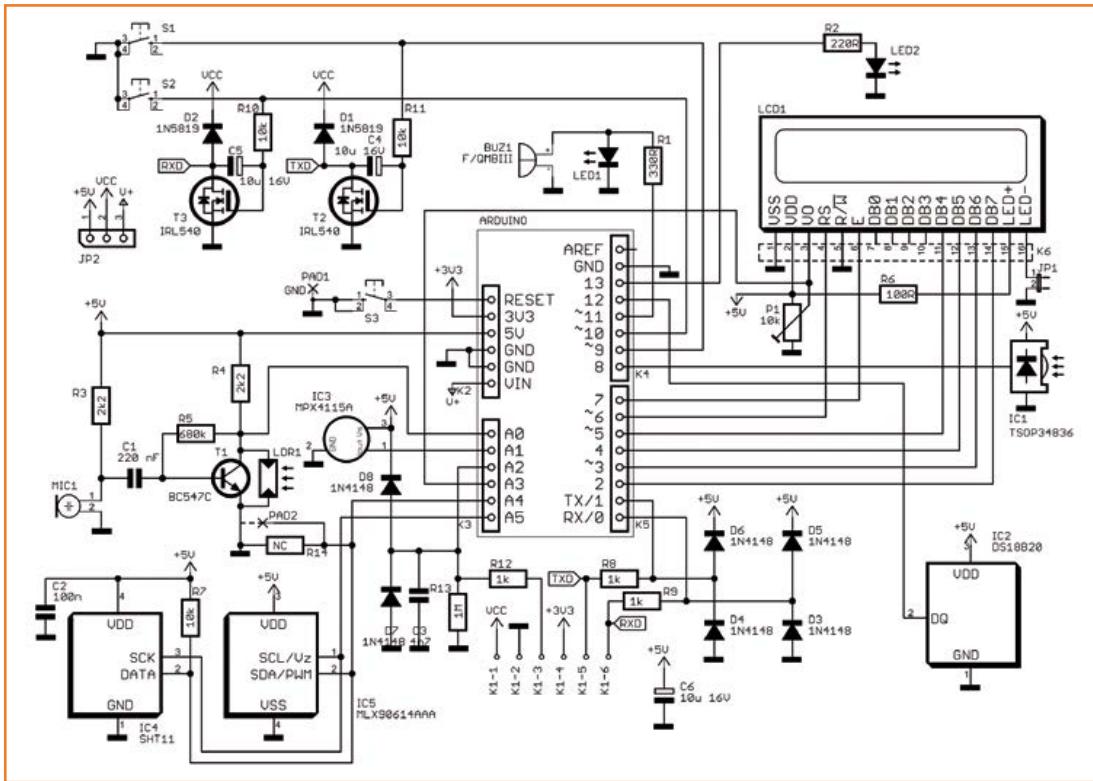
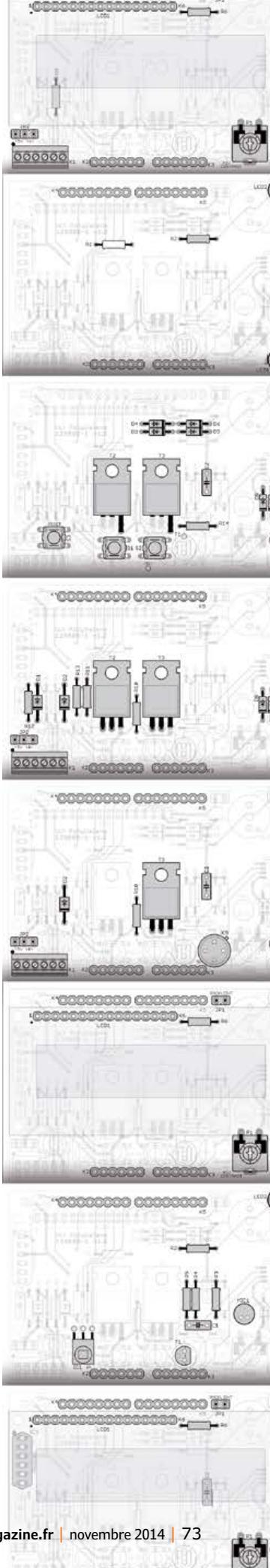


Figure 1. Le schéma de la carte polyvalente est assez dense en raison de la multitude de montages possibles. Certains composants partagent des broches Arduino, aussi ne faut-il pas les monter tous en même temps. La valeur de la plupart voire de toutes les résistances dépend de l'application dans laquelle elles sont utilisées. Pour pouvoir emboîtrer la platine polyvalente sur une carte Arduino, il faut monter les barrettes à picots K2, K3, K4 et K5 sur le côté cuivre de la platine. Tous les autres composants doivent être soudés sur le côté composants.





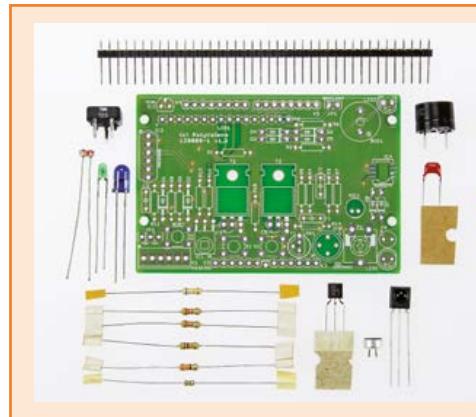
Imagine ce que tu désires, souhaite ce que tu imagines, tu finiras par créer ce que tu veux

trouveras, voisines de la source de ces transistors, des pastilles supplémentaires qui permettent l'utilisation de transistors PNP.

Tu l'auras compris, si la carte d'extension polyvalente pour Arduino présentée dans cet article n'est pas une platine classique, c'est parce qu'elle a été bien étudiée pour être très utile en pratique. Utilisée comme afficheur de données intégré dans un joli coffret ou embarquée dans une usine à gaz comme carte de commande de quelque chose, elle saura s'adapter à tous les rôles qu'elle jouera à merveille. Son large choix de capteurs la rend intéressante pour de nombreuses expériences ; son port série (par USB ou en traditionnel TTL) autorise le transport des données collectées, par exemple vers un ordinateur qui les stockera et les traitera. J'ai préparé plusieurs vidéos de démonstration [4] que je te recommande de visionner. Étudie de près le schéma de la **figure 1** en le comparant au dessin de circuit imprimé que tu auras téléchargé, puis laisse libre cours à ton imagination : combine des montages ; remplace des composants. L'imagination est le commencement de la création.

Après tout, *c'est toi le maître du micro.*

(140359)



Liste des composants du kit

Résistances (1/4 W, 5 %) :

R1 = 330 Ω
 R2 = 150 Ω
 R3, R4 = 2,2 k Ω
 R7 = 10 k Ω
 R5 = 680 k Ω
 LDR1/R14 = LDR 200 k Ω
 P1 = pot. 10 k Ω

Condensateur :

C1 = 220 nF, X7R, 50 V

Semi-conducteurs :

LED1 = LED, verte, 3 mm
 LED2 = LED infra-rouge, 940 nm, 5 mm
 T1 = BC547C
 IC1 = TSOP34836 (récepteur infrarouge)

Divers :

MIC1 = microphone à électret, 6 mm
 BUZ1 = buzzer 12 mm
 barrette sécable à 40 points

Liens

- [1] **2^e édition du livre** : ISBN 978-2-86661-195-8 (chapitre 11 inclus)
 Les lecteurs de la première édition du livre (ISBN 978-2-86661-190-3) **sans le chapitre 11** pourront en télécharger une version PDF :
www.elektor.fr/maitrisez-les-microcontrôleurs-a-l'aide-d-arduino-pdf
- [2] **shield d'expérimentation dans l'e-choppe d'Elektor**
www.elektor.fr/arduino-multipurpose-shield-129009-1
- [3] **kit** : 129009-71-kit-1-arduino-book-cpv
- [4] **trois vidé(m)os en ligne** : www.youtube.com/watch?v=A2j2p-ZcYZc
 Elles sont aussi ici : www.elektor.fr/maitrisez-les-microcontrôleurs-a-l'aide-d-arduino
 (pour les voir, il faut naviguer vers le vers de bas de la page)

commande par gestes 3D pour µC et PC

avec kit d'évaluation Microchip/Elektor et plaque 3D touchpad

Microchip Technology Inc. et *Elektor International Media* lancent conjointement un produit combiné qui passionnera les concepteurs et les programmeurs actifs dans le domaine de la commande par gestes (ou *3D sensing*) sur des systèmes embarqués et sur des ordinateurs (sans oublier la commande tactile).



La clé de voûte du projet est le **MGC3130**, première puce GestIC® de *Microchip* pour la commande gestuelle ; le principe consiste à détecter les variations du champ électromagnétique sous forme de changements de capacité, de l'ordre de grandeur du femtofarad ($1 \text{ fF} = 10^{-15} \text{ F}$).

Notre offre combinée réunit un **kit d'évaluation MGC3130 Hillstar** et une plaque **3D TouchPad**. Le kit comporte un module MGC3130 Module, une passerelle I²C-USB, une électrode de référence à 4 couches (capteur de 95 x 60 mm), un ensemble d'accessoires à assembler soi-même (4 blocs de mousse, 1 film de cuivre) ainsi qu'un câble USB pour la connexion avec le PC. Le kit facilite le paramétrage du MGC3130 et permet d'explorer en profondeur la technologie Gestic®. L'outil téléchargeable **Aurea software tool** facilite la visualisation des signaux, assure la journalisation des données et le paramétrage du MGC3130. Le kit propose aussi des conseils sur la conception des électrodes adaptées à vos besoins.

La plaque 3D Touchpad proposée est un périphérique de suivi 3D et de commande par gestes avec fonction de souris incorporée. Elle est prête à l'utilisation telle quelle, avec n'importe quels

ordinateur, tablette ou système embarqué dotés d'une interface USB. Le circuit imprimé porte à la fois les électrodes pour la capture 3D et la matrice du capteur tactile.

La plaque 3D Touchpad et le kit Hillstar MGC3130 offrent une base solide universelle pour se lancer dans la commande par gestes avec des systèmes tels Arduino, RPi, BBB, T-Boards et bien d'autres. Elektor publiera des applications de la technologie Gestic® (TuxRacer sera parfait pour commencer...), avec l'appui technique de concepteurs de *Microchip*.

Cette offre exclusive sera disponible à un prix fortement réduit dans l'e-choppe d'Elektor.

La disponibilité du produit et d'autres informations seront communiquées dans la lettre d'information électronique Elektor.Post, dans le mensuel Elektor, sur le site Elektor-Labs et dans l'*Elektor Maker Space* au salon *electronica 2014* à Munich.

(140408)



MGC3130 Hillstar Single Zone Development Kit (Part # DM160218)

3DTouchPad DM160225



générateur audio Hewlett-Packard 200AB digne descendant du 200A

Chuck Hansen (USA)

L'histoire de l'entreprise Hewlett-Packard a commencé en 1939, quand William Hewlett a utilisé une ampoule à filament de tungstène dans un circuit oscillateur à pont de Wien à tubes. L'ampoule introduisait le coefficient de température variable nécessaire pour stabiliser la tension de

Figure 1.
Le générateur audio HP 200A. Si vous en avez un, appelez votre assureur.



Figure 2.
Le générateur sinusoïdal HP 200AB (vers 1952).



sortie de l'oscillateur, sans pour autant influer négativement sur la distorsion.

Hewlett s'est associé à David Packard pour construire leur premier appareil commercial, le *200A Audio Oscillator* (**fig. 1**). Les HP 100A à 100E étaient des étalons de fréquence à quartz, destinés au calibrage et aux laboratoires scientifiques, c'est pourquoi je suppose qu'ils n'étaient pas considérés comme des produits « commerciaux ». La dénomination HP20x a été affectée à des générateurs à fréquence variable.

Le modèle 200A délivrait 1 W (22,4 V_{rms}) dans 500 Ω sur une plage de fréquence de 35 à 35 000 cps (cycles par seconde suivant la terminologie de l'époque ; hertz). Le châssis en tôle mesurait 267 mm (L) × 178 mm (H) × 254 mm (P). Vous trouverez de la lecture au sujet de l'histoire de Hewlett-Packard sur le site *HP Virtual Museum* [1].

Les studios Walt Disney demandèrent à HP quelques modifications de conception du modèle 200 pour tester les systèmes audio dans des cinémas équipés spécialement pour présenter *Fantasia* en 1940. Ces modifications donnèrent le HP 200B. Il délivrait 1 W dans 500 Ω sur une plage de fréquence de 20 à 20 000 Hz.

Le HP 200C était similaire au modèle 200A mais son étage de sortie comportait deux étages à tubes avec couplage R-C au lieu du transformateur de sortie des 200A et 200B. Le modèle 200C a une plage de fréquence étendue vers le haut jusqu'à 200 kHz mais avec une puissance de sortie ramenée à 100 mW dans 1 000 Ω.

Le modèle 200I était un modèle à bande étalée de 6 Hz à 6 kHz, avec une sortie à couplage R-C de 10 V_{rms} dans 1 000 Ω. Le I dans la référence signifiait « interpolé » parce que le vernier de fréquence avait 750 points calibrés répartis sur 300 degrés. Les modèles 200AB et 200CD ont été présentés ensemble en 1952. Leurs coffrets ont été redessinés, pour passer des proportions larges et basses aux proportions hautes et étroites (191 mm (L) × 292 mm (H) × 318 mm (P)) pour réduire la surface occupée. Le diamètre du cadran de fréquence a été agrandi de 5 à 6 pouces (**fig. 2**).

Le modèle 200AB avait un meilleur transfo de sortie avec une plage de fréquence de 20 à 40 kHz et une puissance de 1 W ($24,5 \text{ V}_{\text{rms}}$) dans 600Ω . Il remplaçait le 200A et le 200B pour le même prix. Le taux de distorsion harmonique était donné pour inférieur à 1 % jusqu'à 20 kHz et inférieur à 2 % de 20 kHz à 40 kHz.

Le modèle 200CD couvrait une plage de fréquence beaucoup plus étendue, de 5 Hz à 600 kHz, avec une puissance limitée à 160 mW ($9,8 \text{ V}_{\text{rms}}$) dans 600Ω . Le taux de distorsion harmonique était donné pour inférieur à 0,5 % à 1 W et à 1 % au maximum à 3 W. Il avait aussi un atténuateur 0 à 40 dB, par échelons de 10 dB avec réglages grossier et fin. Les modèles 200AB et 200CD avaient une douille de masse du châssis (noire) avec un cavalier de terre amovible (en bas). La douille et le cavalier ont été supprimés du modèle 201C.

Les premiers générateurs HP à semi-conducteurs

La génération suivante des oscillateurs sinusoïdaux HP fut celle des modèles 204C et 204D, « tout transistor ». La taille du châssis a été ramenée à 130 mm (L) × 155 mm (H) × 203 mm (P) (fig. 3). Le 204C couvrait une plage de 5 Hz à 1,2 MHz, avec une puissance de 10 mW ($2,5 \text{ V}_{\text{rms}}$ dans 600Ω). Un JFET était utilisé comme résistance dynamique de stabilisation à la place de l'ampoule à filament des modèles 200. Le taux de distorsion harmonique était de 0,5 % de 100 Hz à 300 kHz, et 1 % de 300 kHz à 1,2 MHz. Deux options d'alimentation étaient possibles : une batterie de piles au mercure ou une batterie d'accus rechargeables CadNi. Il y avait deux gammes de distorsion à basse fréquence de 5 Hz à 100 Hz, avec un commutateur à l'arrière. Le mode faible distorsion était de 1 %, le mode normal de 5 % mais la sortie se stabilisait beaucoup plus vite. Chaque fois que je règle la fréquence trop vite sur mon 204C en mode faible distorsion, la sortie devient instable.

Le 204D était identique au 204C, hormis le potentiomètre d'amplitude de sortie, remplacé par un atténuateur à échelons de 0 à 80 dB accompagné d'un potentiomètre sur 10 dB. Un atténuateur passif séparé de 0 à 110 dB/ 600Ω a été proposé sous la référence HP 350D. Il passait une puissance de 5 W ($55 \text{ V}_{\text{rms}}$).

Le HP 209A délivrait en même temps du sinus et du carré, sur une plage de fréquence étendue de 4 Hz à 2 MHz. Le taux de distorsion était identique à celui des 204C/D jusqu'à 1 MHz, il montait jusqu'à 5 % au-delà.

HP 200AB – vu de plus près

Retirez seulement deux vis cruciformes au-dessus du cordon secteur, sur la face arrière, et vous pouvez faire coulisser le capot. Mon 200AB est très propre à l'intérieur et semble avoir tous ses tubes d'origine. La vue de dessus (fig. 4) montre le condensateur variable à quatre cages, avec un réducteur mécanique incorporé pour faciliter l'accord. La figure 5 montre le côté gauche. Les tubes V1 (6AU6A) et V2 (6AQ5) portent sur le verre le petit logo HP et celui du fabricant General Electric. Les exemplaires précédents utilisaient des tubes octal, 6SJ7 pour V1 et un autre 6K6 pour V2. Contrairement aux composants passifs, le tableau *Replaceable Parts* de la section V du manuel du 200AB ne spécifie pas de fabricant de

ESTP 2004

Rétronique est une rubrique mensuelle sur les pages glorieuses et jaunies de l'électronique, avec occasionnellement des montages de légende décrits dans Elektor. Si vous avez des suggestions de sujets à traiter, merci de les télégraphier à redaction@elektor.fr

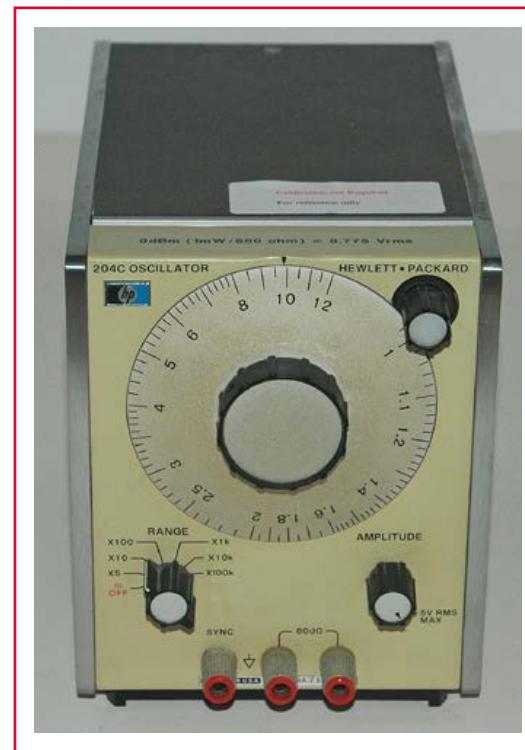


Figure 3.
Le générateur sinusoïdal HP 200C.

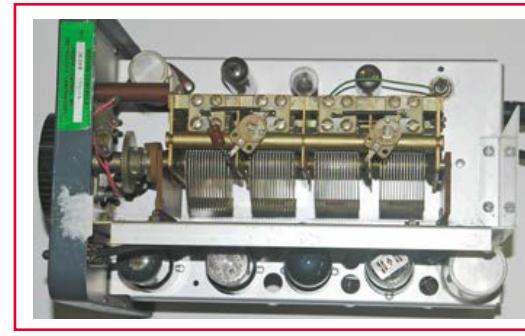
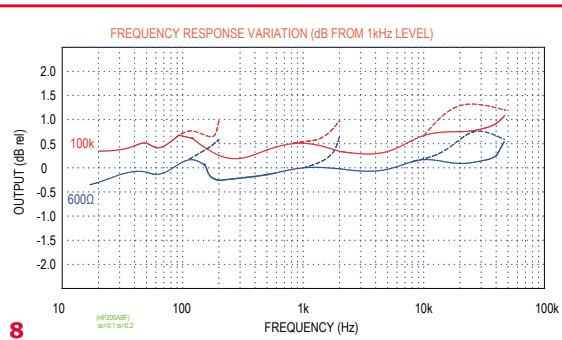


Figure 4.
L'intérieur du HP 200C vu du haut.



8

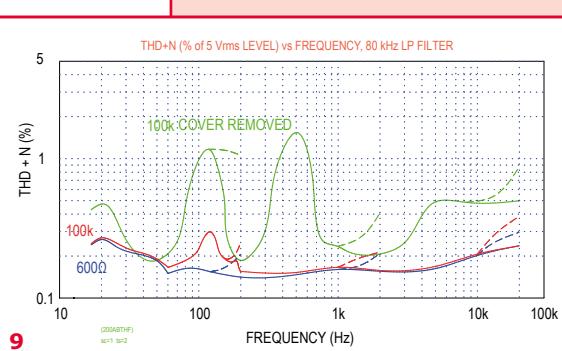
HP 200AB, tests électriques

J'utilise mon oscillateur audio HP 200AB quand j'ai besoin d'une sinusoïde de plus forte amplitude que celles que délivrent mes générateurs à amplis opérationnels ou le distorsiomètre HP 339A. Il fonctionne toujours parfaitement. J'ai effectué quelques mesures audio sur mon HP 200AB.

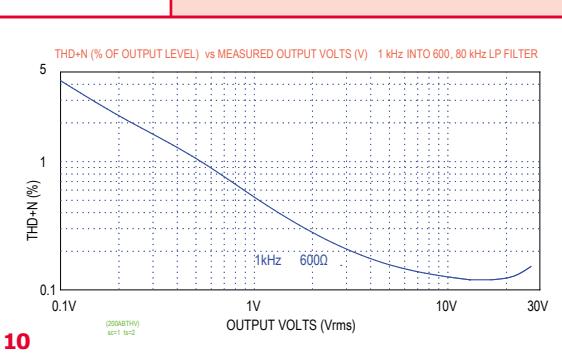
La **figure 8** montre la réponse en fréquence sur des charges de 100 Ω et 600 Ω avec 5 V_{rms} en sortie (environ 20 sur l'échelle de 0 à 100). Les valeurs aux fréquences 20 et 200 se superposent dans le sens où vous pouvez obtenir la même fréquence aux deux extrémités du cadran en changeant de gamme, par exemple on peut régler à 2 kHz avec 20 x 100 ou 200 x 10. Les segments courbes montants en pointillés à 200 Hz, 2 kHz et 20 kHz représentent l'extrême 200 du cadran. On voit que l'amplitude monte du côté 200 mais reste raisonnablement plate du côté 20 du cadran pour la même fréquence. Le 200AB se conforme haut la main à la spécification de +/- 1 dB de réponse en fréquence, référencée à 1 kHz à puissance de sortie nominale.

La charge en sortie affecte l'amplitude comme prévu. L'impédance de charge minimale est de 75 Ω . Avec les valeurs de tension sur 600 Ω et 100 k Ω , j'ai déterminé la résistance interne de mon HP 200AB : 38,5 Ω à 1 kHz, avec une variation de +/- 1 Ω sur toute la plage de fréquence. Lors de toutes les mesures de charge, fréquence et puissance de sortie, je n'ai jamais vu l'ampoule R11 s'éclairer. La sortie sinusoïdale n'a jamais manifesté de signe d'écrêtage, quelle que soit l'amplitude de sortie.

La **figure 9** montre le taux de distorsion harmonique totale en fonction de la fréquence pour les charges de 100 k Ω et 600 Ω , toujours sous 5 V_{rms}. Le graphe à 600 Ω est remarquablement plat, avec un léger creux à 60 Hz où le filtre éliminateur du distorsiomètre HP 339A coupe la composante du secteur et la fondamentale du signal en même temps. Je me suis servi de l'oscilloscope pour vérifier le signal après le filtre coupe-bande du



9



10

test de distorsion. Il montre surtout le troisième harmonique sur toute la plage de fréquence. Quand la sortie est ouverte, à part la charge de 100 k Ω du distorsiomètre, la distorsion montre deux pics à 120 Hz et 180 Hz.

Pour effectuer mes premiers tests sur le HP 200AB, j'avais retiré le capot en aluminium pour accéder au potentiomètre de calibrage d'amplitude R 100 Ω . Il est évident que le capot fait partie du système de blindage : sans lui, le taux de distorsion harmonique + bruit monte au-delà de la spécification de +/- 1 dB. Le HP 200AB utilise des mégohms et des picofarads dans le pont de Wien, c'est pourquoi il ne faut pas beaucoup d'énergie extérieure pour perturber l'oscillateur. Les segments montants en pointillés des courbes montrent les parties pour lesquelles j'ai utilisé l'extrême 200 du cadran. L'augmentation d'amplitude s'accompagne d'une augmentation de la distorsion.

Dans la **figure 10**, la plage de test va de 100 mV à 27,7 V_{rms}, la tension de sortie maximale du HP 200AB à 1 kHz. La composante bruit de la distorsion totale décroît suivant une loi linéaire de la plus petite tension de sortie jusqu'à 13 à 16,9 V_{rms}, où la distorsion totale est de 0,12 %. Au-dessus de 17 V_{rms}, elle recommence à augmenter avec les produits de distorsions supérieurs au niveau constant du bruit.

Le niveau de bruit en sortie de l'amplificateur avec la commande d'amplitude à zéro est de 4,62 mV_{rms}, soit 74,5 dB en dessous de la puissance de sortie nominale de 1 W. C'est impressionnant, vu l'âge de ce HP 200AB et la limite spécifiée de 66 dB en dessous du niveau nominal.

tube. La désignation de fabricant est « ZZ », ce qui signifie toute marque conforme aux normes RETMA [2].

Une ampoule à filament G.E. de 10 W/240 V (désignée par R11 sur le schéma) est située entre les deux tubes. J'ai vu d'autres oscillateurs de la série 200 avec deux ampoules de 3 W/120 V montées en série. Le potentiomètre de réglage du taux de distorsion est à l'intérieur à l'arrière du châssis, un condensateur de couplage Sprague de 20 μ F/450 VDC (C8) se trouve à l'avant.

Le commutateur rotatif de gamme est placé à l'avant en bas, avec de grosses résistances Elektra à 1 % à couche de carbone sur la galette avant et des résistances de calibrage, plus petites, en carbone aggloméré sur la galette arrière. Le cordon secteur et le porte-fusible sont sur la face arrière. La **figure 6** montre les sections amplificateur et alimentation du côté droit. Le deuxième tube en partant de l'avant est V3, un 6SN7-GT, à nouveau avec les logo HP et GE sur le verre. Autour de V3 se trouvent V4 et V5, les tubes de sortie 6K6-GT/G, marqués *United Electron*. Le redresseur V6, 5Y3-G, est un tube RCA, bien qu'un 5AR4 soit admis aussi selon les inscriptions au-dessus du tube. Le tableau *Replaceable Parts* du premier manuel ne cite que le 5Y3.

Le potentiomètre d'amplitude est situé sur la face avant sous le châssis des tubes. Le transformateur de sortie 600 Ω est au milieu en bas, le transfo d'alimentation à l'arrière. Les deux noyaux sont à angle droit pour limiter le couplage magnétique mutuel.

Description du circuit

Le schéma (figure 4-3 du manuel qui accompagne mon HP 200AB) est reproduit par la **figure 7**. Les manuels de ces appareils HP d'époque et d'autres sont disponibles sur l'internet sous la forme de PDF gratuits. Comme je l'ai dit plus haut, les tubes de la section oscillateur de mon appareil sont 6AU6A pour V1 et 6AQ5 pour V2. Mon schéma montre V1 comme 6SJ7 et V2 comme un autre 6K6. L'un et l'autre sont des tubes octal, alors que mon HP 200AB a des tubes miniatures pour V1 et V2. Mon appareil porte le numéro de série 17527, supérieur au dernier numéro de la série visée par ce manuel.

L'oscillateur est un pont de Wien stabilisé par une ampoule. Un condensateur variable à air à quatre cages est commandé par le cadran de fréquence via un réducteur mécanique et le commutateur de gamme utilise des résistances fixes. La réaction

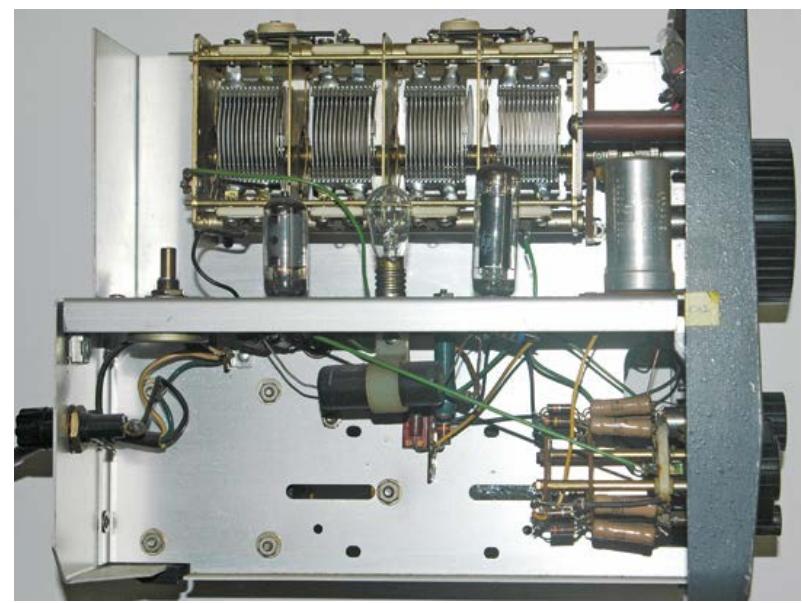


Figure 5.
La section oscillateur du HP 200AB avec l'ampoule qui n'est pas destinée à éclairer quoi que ce soit.

positive nécessaire pour maintenir l'oscillation est ramenée de la sortie de V2 à la section oscillateur du pont de Wien R-C par le condensateur de couplage C8. La réaction négative, nécessaire pour limiter la distorsion et les variations d'amplitude sur toute la plage de fréquence, est constituée de R9, du potentiomètre de réglage d'amplitude R10 et du filament de l'ampoule R11. La résistance de l'ampoule augmente proportionnellement à la tension de l'oscillateur, ce qui renforce la contre-réaction et ramène l'oscillateur à sa tension de fonctionnement normale.

Figure 6.
Les sections amplificateur et alimentation du HP 200AB.



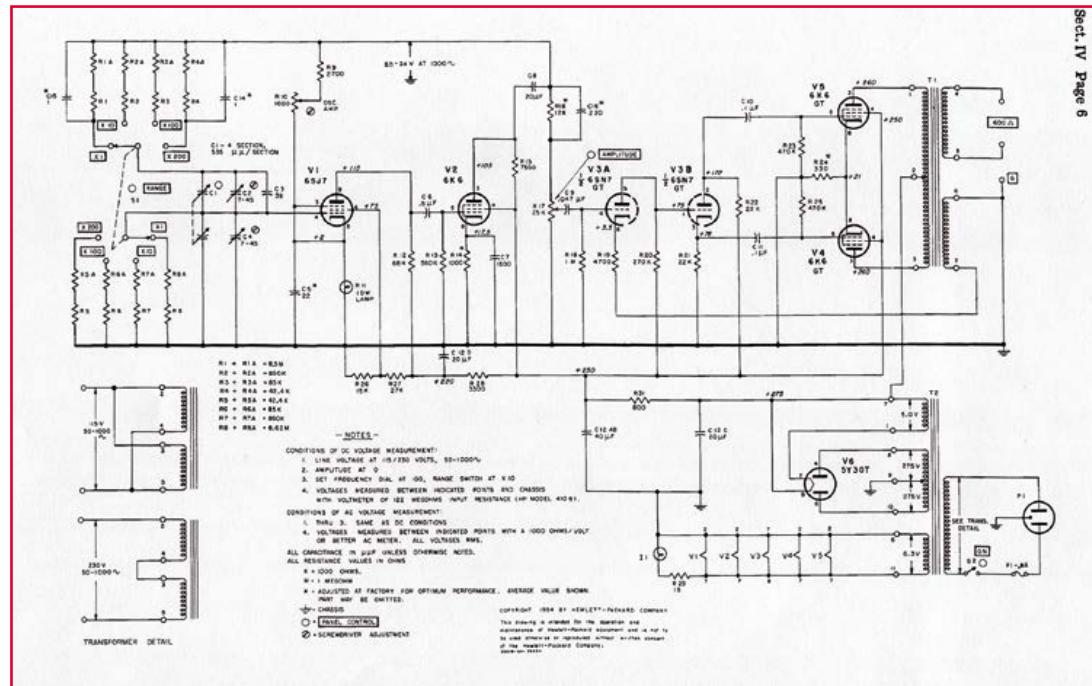


Figure 7.
Le schéma de l'oscillateur audio HP 200AB.

La plage d'oscillation stable est représentée de 20 à 24 V_{rms} sur le schéma. L'amplitude de mon HP 200AB était de 28,7 V_{rms}. Elle est ramenée à 15,3 V_{rms} par R16 en série avec le potentiomètre d'amplitude R17, pour éviter l'écratage en sortie dans toutes les gammes d'amplitude et de fréquence. La tension de l'oscillateur varie légèrement de +/- 0,05 V_{rms} quand le réglage d'amplitude passe de 0 à 100.

La section amplificateur est un étage *push-pull* classique (V4, V5), avec V3A comme préamplificateur et V3B comme déphaseur. Comme la tension de l'oscillateur est très élevée, le gain en tension entre la commande d'amplitude et l'enroulement 600 Ω de sortie n'est que de 2 (1,26 dB). Un secondaire supplémentaire sur

le transfo de sortie apporte environ 30 dB de contre-réaction globale. Le tube V3A introduit une certaine contre-réaction locale puisque la résistance de cathode R19 n'est pas découplée. L'alimentation est caractéristique des amplificateurs à tube. Le tube V6 effectue un redressement à deux alternances, suivi par des filtres en pi pour l'alimentation des plaques et la polarisation des grilles écrans.

(130423 – version française : JPB)

Liens ou èbe, références

- [1] www.hp.com/hpinfo/abouthp/histnfacts/museum/index.html
- [2] La RETMA, *Radio Electronics Television Manufacturer's Association* (créée en 1953), a été remplacée par le *Joint Electron Device Engineering Council* (JEDEC) in 1958.

Les codes de fabricants utilisés initialement par HP et d'autres ont été remplacés par les numéros uniques à cinq chiffres CAGE (*Commercial and Government Entity*) qui identifient une usine à un endroit précis. Agilent (anciennement *Hewlett-Packard test equipment group*) possède actuellement vingt codes CAGE enregistrés dans le monde. Avec l'augmentation du nombre de centres de production CAGE, le premier chiffre a été remplacé par une lettre.

Nous recherchons des images originales –
svp contactez editor@elektor.com

HP 200x, les séries

HP a produit d'autres matériels de test tout-transistor avec des numéros de la série 200 :

- 202H Générateur de signaux FM/AM
- 203A Générateur à phase variable
- 207H Convertisseur universel de fréquence RF (*Univerter*)
- 214A/B Générateur d'impulsions
- 226A Générateur marqueur (pour le calibrage des bases de temps d'oscilloscopes)
- 230B Amplificateur RF accordé
- 236A Oscillateur de test téléphonique
- 250B Impédancemètre RF
- Plusieurs autres adaptateurs-guides d'onde de la série 200

hexadoku c'est l'heure de la pause

L'électronique, c'est très bien, mais il faut savoir s'arrêter. Levez-vous, dégourdissez-vous les jambes, faites quelques étirements, aérez-vous, soufflez... vous ne doutez pas du bénéfice d'une interruption ? Ce qui vaut pour les tendons, les articulations et les muscles vaut aussi pour la matière grise. **Allons, c'est l'heure de votre gymnastique arithmétique.** Remplissez la grille selon les règles, envoyez-nous votre solution. Vous serez peut-être l'un des **cinq gagnants** d'un cadeau d'une valeur de **50 €**. Après ça, vous serez d'attaque pour reprendre vos cogitations électroniques.

Une grille hexadoku est composée de chiffres du système hexadécimal, de 0 à F. Remplissez le diagramme de 16 x 16 cases de telle façon que **tous** les chiffres hexadécimaux de 0 à F (0 à 9 et A à F) n'apparaissent **qu'une seule et unique fois** dans chaque rangée, colonne et carré

de 4 x 4 cases (délimités par un filet gras). Certains chiffres, déjà placés dans la grille, en définissent la situation de départ. Pour participer, inutile de nous envoyer toute la grille, il suffit de nous donner **la série de chiffres** sur fond grisé.

Participez et gagnez !

Nous tirons au sort cinq des réponses internationales correctes reçues dans les délais ; leurs auteurs recevront un bon d'achat de livres Elektor d'une valeur de **50 €**. À vos crayons !

Où envoyer ?

Envoyez votre réponse (les chiffres sur fond grisé) avec vos coordonnées par courriel, avant le **1^{er} décembre 2014** à l'adresse **hexadoku@elektor.fr**

Les gagnants

La solution de la grille du numéro de septembre (435) est : **E80F4**

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

Ulf Claesson (Suède), **Larry Burns** (Canada), **A. van Maris** (Pays-Bas), **Jairo Rotava** (Brésil) et **Zvi Herman** (Israël).

Bravo à tous les participants et félicitations aux gagnants !

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

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

Tout recours est exclu de même que le sont, de ce jeu, les personnels d'Elektor International Media et leur famille. Un seul gagnant par foyer.

touche pas à mon câble !

Wolfram Pioch
(Allemagne)

Le banc d'essai du programmeur Atmel ICE publié dans le dernier numéro (oct. 2014), n° 436 p. 10, a valu à son auteur quelques déboires avec l'exemplaire qui lui avait été prêté. Il raconte :

La deuxième fois que j'ai voulu enficher le câble micro-USB dans le programmeur, je n'ai senti aucune résistance mécanique, le connecteur s'enfonçait dans le vide : plus de fiche sur la carte ! Ça commençait bien... un appareil prêté, j'étais censé le rendre dans l'état où je l'avais reçu. Au lieu de tout renvoyer en l'état et de vous priver de banc d'essai, j'ai pris le parti de tenter une réparation. Avant même que ma décision soit prise consciemment, mon bras droit était déjà parti tout seul vers le tournevis. La photo montre l'étenue du désastre : fiche arrachée, plans de masse arrachés, et même une piste (la deuxième en partant de la droite) arrachée ; aucune trace de soudure sur les plans de masse sous la fiche, juste quelques traces de la pâte adhésive pour maintenir les composants jusqu'à la refusion.

Voilà ce qui arrive inévitablement quand, sur un circuit imprimé, la fixation des embases montées en surface est absente ou insuffisante. On peut d'ailleurs considérer que si les pistes intactes avaient été bien soudées, elles auraient été arrachées elles aussi avec la fiche...

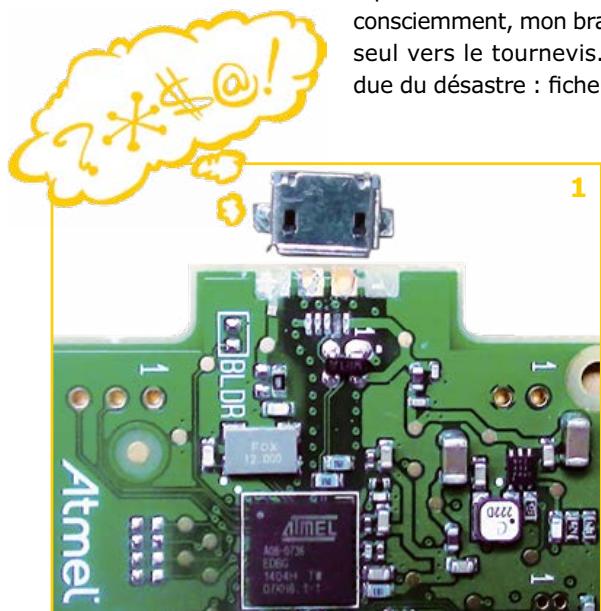
Comment réparer ? Les chances de succès d'un resoudage de la fiche étaient maigres. Électriquement, c'est faisable, mais mécaniquement, ce n'est pas

la solution. J'ai donc opté pour ce que je sais bien faire : le câblage volant en fil de cuivre émaillé pour relier les broches de l'embase micro-USB au circuit imprimé. En dépit de mon expérience, même avec ma panne la plus fine, il m'a fallu du temps, tant les connexions sont serrées. Tandis que je soudais un fil, la chaleur faisait fondre la soudure du fil voisin. Ensuite, il a fallu tester une à une la rigidité mécanique des soudures des fils sur l'embase : deux fils se sont détachés à des positions intermédiaires, il a donc fallu recommencer du début. Quand enfin les cinq fils (pas trop courts – environ 3 ou 4 cm – pour que l'ensemble ne devienne pas trop rigide) ont tenu, il a fallu les souder sur le circuit imprimé, puis procéder à une soigneuse vérification à la loupe. Vint le moment délicat de la remise en place de l'embase micro-USB dans sa position d'origine, en coudant précautionneusement les fils. Comme vous vous en doutez, de nouveau deux fils ont sauté. Et quand j'ai fini par arriver à remettre le circuit dans son boîtier avec l'embase rafistolée, vous devinez ce qui s'est passé... Je vous assure que j'ai beaucoup plus d'expérience en soudures délicates maintenant qu'avant ce banc d'essai ! Pour immobiliser l'embase avant de refermer le programmeur, j'ai utilisé de la colle thermoplastique ; c'est moche, j'en conviens, mais solide et réversible. J'ai renoncé à aligner proprement les fils de cuivre émaillé, l'un ou l'autre aurait encore sauté.

Par précaution, lors du premier essai, je n'ai pas connecté le programmeur directement à l'ordinateur mais à un concentrateur USB. J'ai d'abord vu rouge, mais c'était heureusement la LED témoin qui signalait que la connexion était établie. Il ne restait qu'à connecter le câble ICE pour constater, soulagé, que tout fonctionnait. Comme le pistolet à colle était encore chaud, j'ai fini par coller le câble USB au boîtier. Bien en vue, j'ai ajouté une étiquette avec cet avertissement « touche pas à mon câble ! ».

Ça tiendra ce que ça tiendra... certainement plus longtemps que l'original. Au fait, j'y pense : je comprends maintenant pourquoi dans l'aide d'Atmel Studio 6.2 on explique comment démonter le boîtier. C'est peut-être pour ça aussi que le circuit est disponible séparément sous forme de PCBA Kit.

(140274)



Venez vérifier par
vous-même à Electronica
11-14 Novembre 2014 Munich Expo
Hall A1 Stand 115

PicoScope® OSCILLOSCOPES PC

Pour toute application il y a un PicoScope

Bandé Passante de 5MHz à 1 Ghz • Echantillonnage 10Mé/s à 5Gé/s • Mémoire de 8Ké à 2Gé

PICOSCOPE SERIE 2200A Le PicoScope dans la poche!



- Canaux:** 2 + génér. arb.
- Bandé Passante:** 10 à 200 MHz
- Echantillonnage:** 100Mé/s à 1 Gé/s
- Résolution:** 8 bits
- Mémoire d'acquisition:** 8 à 48 Ké
- Alimentation:** par USB
- Prix à partir de:** £159 \$262 €192

PICOSCOPE MSOs Analyse de Signaux Mixtes



- Canaux:** 2 ou 4 analogiques, 16 logiques et génér. arb.
- Bandé Passante:** 25 à 200 MHz
- Sampling:** 200 Mé/s à 1Gé/s
- Résolution:** 8 bits
- Mémoire d'acquisition:** 48 ké à 512 Mé
- Alimentation:** par USB ou bloc secteur
- Prix à partir de:** £349 \$658 €483

PICOSCOPE SERIE 3200

Oscilloscopes à Grande Mémoire



- Canaux:** 2 + décl. externe et génér. arb.
- Bandé Passante:** 60 à 200 MHz
- Echantillonnage:** 500 Mé/s
- Résolution:** 8 bits
- Mémoire d'acquisition:** 8 to 128 Mé
- Alimentation:** par USB
- Prix à partir de:** £399 \$658 €483

PICOSCOPE SERIE 3400

Oscilloscopes Hautes Performances



- Canaux:** 4 + décl. externe et génér. arb.
- Bandé Passante:** 60 à 200 MHz
- Echantillonnage:** 1Gé/s
- Résolution:** 8 bits
- Mémoire d'acquisition:** 8 to 128 MS
- Alimentation:** USB
- Prix à partir de:** £599 \$988 €725

PICOSCOPE SERIE 4000 Oscilloscopes Haute Précision



- Canaux:** 2 ou 4
- Bandé Passante:** 20 MHz
- Echantillonnage:** 80Mé/s
- Résolution:** 12 bits
- Mémoire d'acquisition:** 32 Mé
- Alimentation:** par USB
- Prix à partir de:** £499 \$823 €604

PICOSCOPE 4824 8 voies, Haute Résolution



- Canaux:** 8
- Bandé Passante:** 20 MHz
- Echantillonnage:** 80Mé/s
- Résolution:** 12 bits
- Mémoire d'acquisition:** 256 Mé
- Alimentation:** par USB
- Prix à partir de:** £1395 \$2302 €1688

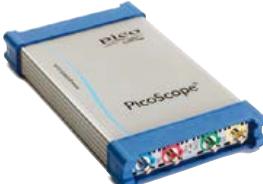
PICOSCOPE SERIE 5000 Oscilloscopes à résolution flexible



- Canaux:** 2 ou 4 + décl. externe + génér. arb. à faible distorsion
- Bandé Passante:** 60 à 200 MHz
- Echantillonnage:** 1 Gé/s
- Résolution:** 8 to 16 bits
- Mémoire d'acquisition:** 8 to 512 MS
- Alimentation:** 2 Ch USB, 4 Ch AC adaptor
- Prix à partir de:** £799 \$1318 €967

PICOSCOPE SERIE 6000

Performance Ultime et USB 3.0



- Canaux:** 4 + décl. externe et génér. arb.
- Bandé Passante:** 250 à 500 MHz
- Echantillonnage:** 5 Gé/s
- Résolution:** 8 bits
- Mémoire d'acquisition:** 256Mé à 2 Gé
- Alimentation:** par secteur
- Prix à partir de:** £1995 \$3292 €2414

LOGICIEL PICOSCOPE® 6 Fonctionnalités haut de gamme en standard



Les pilotes et kit de développement logiciel fournis vous permettent à la fois de développer vos propres programmes et de s'interfacer à des logiciels tiers tels que LabVIEW et MATLAB. Vous pouvez utiliser maintenant la plupart des PicoScopes sous Linux et Mac OSX.

Tous les modèles incluent le logiciel et une garantie de 5 ans. Le logiciel inclut les mesures, l'analyseur de spectre, le SDK, les déclenchements avancés, le mode persistance couleur, les décodages série (CAN, LIN, RS232, I²C, I²S, FLEXRAY, SPI), Masques, Voies calculées, tout est en standard, mises à jour gratuites.

Intégration de la reconnaissance de gestes 3D et du suivi des mouvements

grâce à la technologie GestIC® de Microchip



La technologie brevetée GestIC® 3D de Microchip offre une solution embarquée sur puce, de reconnaissance des gestes en temps réel. Le MGC3230 permet des interfaces utilisateur ultra innovantes sur tout type de produit pour un niveau de prix très économique.

MGC3130

- Le MGC3130 fonctionnant avec la bibliothèque de gestes Colibri, il permet la reconnaissance avancée de gestes 3D complexes sans pour autant nécessiter un processeur hôte
- Le MGC3130 envoie les données détectées (gestes, coordonnées XYZ) à l'hôte via un bus I²C™ ou des E/S à usage général (GPIO)



microchip
DIRECT
www.microchipdirect.com

 **MICROCHIP**
microchip.com/gestic