

elektor

à demander
- FPGA : C quoi ?
- à quoi ça me sert ?
- applications ?
- prix ? outils ?

FPGA

...et l'Homme créa la puce

- Arduino ausculte la tension secteur | retour aux sources (10) **embarquez Linux !** | interface USB 24 E/S | sapins de Noël siamois
- l'internet des objets ● **Bruits de labo** | régulation de ventilateur
- **HP-35 : révolution de poche (1972)** ● le monde d'Elektor





DESIGNSPARK

NOUVELLES SOLUTIONS POUR ACCÉLÉRER LA CONCEPTION

NOUVELLE VERSION DESIGNSPARK PCB 4

NOUVELLES BIBLIOTHÈQUES ÉTENDUES
SÉLECTION ET COTATION DES COMPOSANTS FACILITÉES
PROTOTYPAGE DE CIRCUIT IMPRIMÉ

NOUVEAU ET GRATUIT MODELSOURCE

PLUS DE 80 000 COMPOSANTS
DISPONIBLES DANS PLUS DE 20 FORMATS DONT
PADS, ORCAD, ALTIUM ET CADSTAR

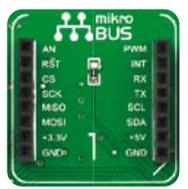
À découvrir sur
www.designspark.com

DÉVELOPPÉ PAR 



\$4900

un SUPPORT



\$1800

pour tous cartes click

Il vous suffit de placer la carte Click™ vous avez besoin dans le nouveau support mikroBUS™ innovant et elle est immédiatement fonctionnelle.



\$1790



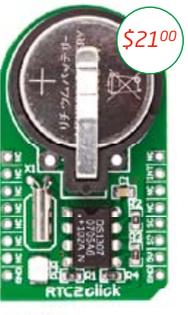
\$4500



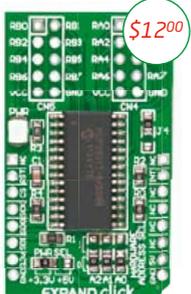
\$4190



\$2900



\$2100



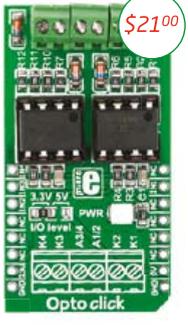
\$1200



\$2600



\$1900



\$2100



\$1390



\$4400



\$1000



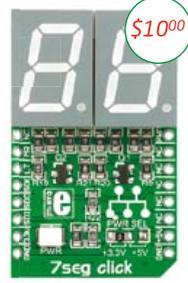
\$2590



\$4900



\$2900



\$1000



\$1900



\$1690



\$2100



\$1800



\$2190



\$1800



\$2400



\$1600



\$900



\$1500



\$2600



\$800



\$1390



\$2100



\$2100



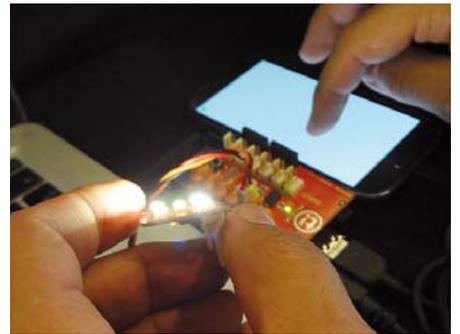
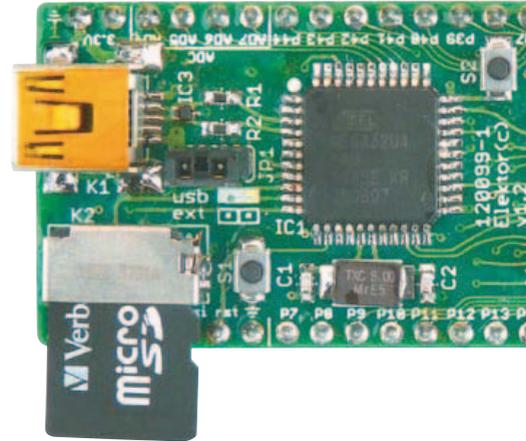
\$2400



www.libstock.com

ALLEZ LE CHERCHER
www.mikroe.com

NOTE: Deux connecteurs mâles 2x8 accompagnent chaque carte Click.



● communauté

12 le monde d'Elektor

- Le mesureur de rayonnement amélioré d'Elektor bientôt sur ConSat-1
- Le mesureur de rayonnement amélioré d'Elektor (bis) à l'université d'Oxford
- Un titanesque Formant d'Elektor sur YouTube
- Perdu dans le labyrinthe électronique
- Circuit Cellar fête ses 25 ans (19_{hex})



● réalisations

18 et l'homme créa sa puce

Elektor a conçu une nouvelle carte d'expérimentation pour FPGA qui facilitera la vie des électroniciens désireux de se familiariser avec cette logique programmable.

26 mettez le cap sur Arduino (3b)

Exploitez sur votre téléphone les données issues de capteurs reliés à Arduino. Une fois les outils opérationnels, l'exécution de code sur votre téléphone et le dialogue avec votre Arduino seront faciles.

32 interface USB 24 E/S

Mesurer, commander et réguler facilement avec un PC et... ce câble qui permet de lire et écrire 24 signaux via l'interface USB d'un PC et bien davantage.

44 Arduino ausculte la tension secteur

Un analyseur FFT en couleurs de la tension du réseau sur une modeste carte Arduino ? C'est possible, la preuve !

52 sapins de Noël siamois

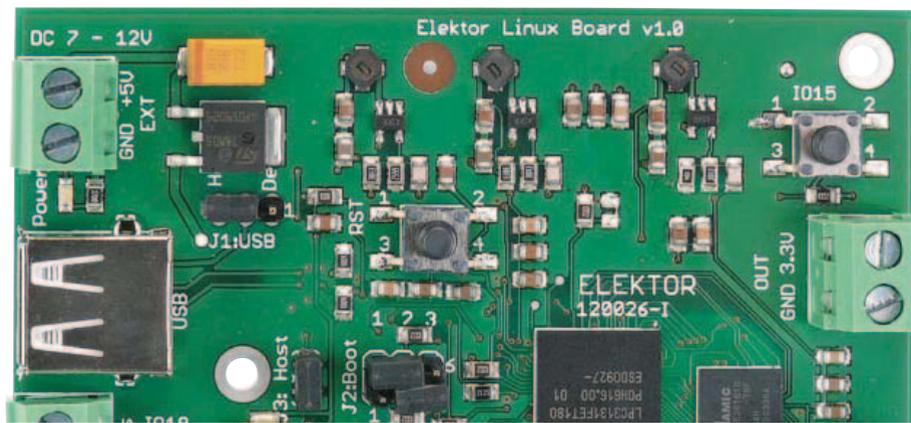
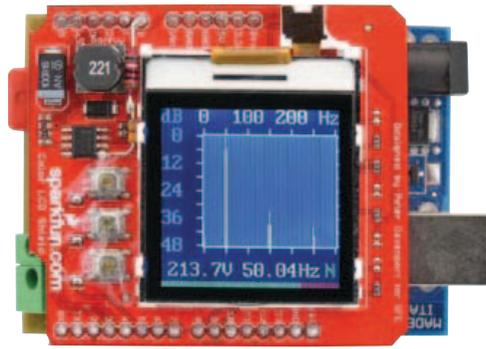
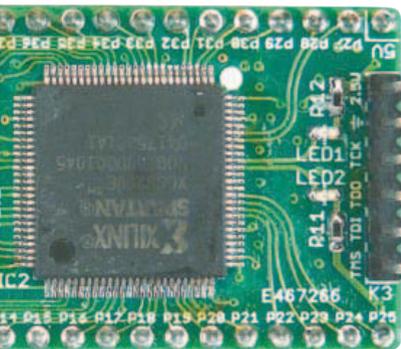
Adieux bougies et ringardes loupottes, même clignotantes, la mode est aux LED, avec effets lumineux embarqués. Et quels effets ! Dignes d'un éclairage de discothèque. Avec ses 56 LED, ce double arbre de Noël offre un spectacle visuel varié, étalé sur plus de trois quarts d'heure.

56 embarquez Linux ! (6)

La modularité de Linux en fait une plateforme parfaite pour qui souhaite une solution complète de mesure, de régulation et de commande. Les langages ne manquent pas pour écrire une application de commande et envoyer son interface à un PC, une tablette ou un téléphone tactile.

62 retour aux sources (10)

Construire une radio reste aujourd'hui un bon moyen de s'initier à l'électronique, à condition que le montage soit simple et sans composants spécialisés.



● labs

42 petit, mais coriace !

... difficultés inattendues pour flasher par ISP sur un programmeur AVRISP MKII

43 « sur un PCB vérifié avec Visualizer le plan de masse ne manquerait pas ! »

Une mésaventure au labo a attiré notre attention sur le nouveau service PCB Visualizer d'Eurocircuits, un utilitaire en ligne sans doute meilleur que le visualiseur Gerber utilisé jusqu'ici au labo.

60 Elektor hot Labs

Clemens Valens nous invite à renifler ce qui mijote au labo ou plus précisément sur www.elektor-projects.com. Ne ratez pas le coche.

82 régulation de ventilateur

Si vous avez un faible pour les ventilateurs, vous apprécierez ces deux circuits intégrés qui assurent une régulation autonome de ventilation.

● industrie

8 infos & actualités

Nouveaux produits & nouveaux services Reichelt | Samsung | Keithley | Piezo-Motor | National Instrument | Linear Technology | Microchip | FLIR



● Tech the Future le futur en tête

68 l'internet des objets

Avec la fusion des mondes physique et numérique, les objets de la vie courante auront bientôt une adresse IPv6 et une existence numérique dans l'internet.

● magazine

6 de nous à vous

Le réseau international d'Elektor

72 le doigt dans l'engrenage des moteurs électriques

Électrotechnique de puissance : démarrage, variation de vitesse, freinage

76 HP-35 : révolution de poche (1972)

Il y a quarante ans, Hewlett-Packard sortait de sa poche la première calculatrice scientifique qui affichait racines carrées, fonctions trigonométriques et logarithmes d'une pression sur une touche — avec une précision incroyable de dix chiffres.

79 Hexadoku

Amusez-vous sans gêne, il n'y a aucun risque d'hexagérer.

83 bientôt dans Elektor

En avant-première, les articles en préparation pour les prochains numéros.

ELEKTOR / PUBLITRONIC SARL

c/o Regus Roissy CDG
1, rue de la Haye
BP 12910
FR - 95731 Roissy CDG Cedex
France

Tél. : (+33) 01.49.19.26.19

Fax : (+33) 01.49.19.22.37

www.elektor.fr

Banque ABN AMRO : Paris

IBAN : FR76 1873 9000 0100 2007 9702 603

BIC : ABNAFRPP

DROITS D'AUTEUR :

© 2012 Elektor International Media B.V.

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

Certains circuits, dispositifs, composants, etc. décrits dans cette revue peuvent bénéficier de droits propres aux brevets; la Société éditrice n'accepte aucune responsabilité du fait de l'absence de mention à ce sujet. Conformément à l'art. 30 de la Loi sur les Brevets, les circuits et schémas publiés dans Elektor ne peuvent être réalisés que dans des buts privés ou scientifiques et non commerciaux. L'utilisation des schémas n'implique aucune responsabilité de la part de la Société éditrice. La Société éditrice n'est pas tenue de renvoyer des articles qui lui parviennent sans demande de sa part et qu'elle n'accepte pas pour publication. Si la Société éditrice accepte pour publication un article qui lui est envoyé, elle est en droit de l'amender et/ou de le faire amender à ses frais; la Société éditrice est de même en droit de traduire et/ou de faire traduire un article et de l'utiliser pour ses autres éditions et activités, contre la rémunération en usage chez elle.

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

Imprimé aux Pays-Bas
par Senefelder Misset - Doetinchem

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

Daltoniens s'abstenir

Voici donc cette nouvelle mouture d'Elektor.Magazine (prononcer : *elektor dot magazine*) dont les changements ne sont pas seulement cosmétiques ; les nouvelles catégories que vous y découvrirez avec leur code de couleur, .communauté en bleu, .réalisations en gris (ou *dot projects* en anglais), .labs en vert, .industrie en noir et .magazine en rouge, incarnent la vision que nous avons de l'électronique et de son avenir.



Au cœur de ce nouveau dispositif : la communauté et ses membres, c'est-à-dire vous, en orbite autour du noyau Elektor. Votre magazine se modernise tout en restant une revue imprimée sur papier. Il se dématérialise et se démultiplie sur les écrans des ordinateurs ou des téléphones et tablettes tactiles, sur l'internet, dans les PDF et surtout dans le labo virtuel, sur le site www.elektor-labs.com. C'est là que va désormais se porter l'essentiel de notre attention et notre enthousiasme.

Le papier n'est plus l'alpha et l'oméga de nos projets électroniques. Avant la publication, ils auront grandi et mûri en ligne, après ils continueront de vivre en ligne au gré de l'intérêt que leur témoignera la communauté. Tout cela reste à inventer. C'est une vision d'avenir qui va maintenant prendre forme. Pour cela, nous comptons sur votre contribution : les allers-retours rapides entre participants du site communautaire .labs, entraîneront une recirculation accélérée de l'information, des adaptations, des discussions, des améliorations, des questions et des réponses, des erreurs et des errances. Tout cela n'ira pas sans *couac*, forcément, avec des participants d'une cinquantaine de pays où Elektor est lu dans 6 langues.

Denis Meyer

Notre équipe

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

**France**

Denis Meyer
+31 46 4389435
d.meyer@elektor.fr

**United Kingdom**

Wisse Hettinga
+31 (0)46 4389428
w.hettinga@elektor.com

**USA**

Hugo Vanhaecke
+1 860-875-2199
h.vanhaecke@elektor.com

**Germany**

Ferdinand te Walvaart
+31 46 4389417
f.tewalvaart@elektor.de

**Netherlands**

Harry Baggen
+31 46 4389429
h.baggen@elektor.nl

**Spain**

Eduardo Corral
+34 91 101 93 95
e.corral@elektor.es

**Italy**

Maurizio del Corso
+39 2.66504755
m.delcorso@inware.it

**Sweden**

Wisse Hettinga
+31 46 4389428
w.hettinga@elektor.com

**Brazil**

João Martins
+55 11 4195 0363
joao.martins@editorialbolina.com

**Portugal**

João Martins
+351 21413-1600
joao.martins@editorialbolina.com

**India**

Sunil D. Malekar
+91 9833168815
ts@elektor.in

**Russia**

Nataliya 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
+27 78 2330 694
j.dijk@elektor.com

**China**

Cees Baay
+86 21 6445 2811
CeesBaay@gmail.com

Notre réseau



VOICE COIL

CIRCUIT CELLAR
THE WORLD'S SOURCE FOR EMBEDDED ELECTRONICS ENGINEERING SOLUTIONS

audio X PRESS



vous connecte à



Nos annonceurs



Beta Layout

www.pcb-pool.com14

DesignSpark

www.designspark.com 2

Eurocircuits

www.elektorpcbservice.com55

Jackaltac

www.jackaltac.com71

Microchip

www.microchip.com41

MikroElektronika

www.mikroe.com 3

National Instruments

www.ni.com/academic/f84

Pico

www.usbms.com/PS17471

Reichelt

www.reichelt.fr15

Schaeffer AG

www.schaeffer-ag.de14

Pour placer votre annonce dans le prochain numéro d'Elektor

veuillez contacter Mme Ilham Mohammadi par téléphone au (+33) 01.49.19.26.19
ou par courrier électronique : i.mohammadi@elektor.fr

Vos correspondants

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

Prix et performance parfaitement accordés

Le fournisseur de composants électroniques *reichelt* montre avec la station de soudage numérique LF-3000 du fabricant taïwanais *Xytronic* qu'il est possible d'acquérir un outil professionnel à un prix abordable. Sur son site www.reichelt.com les professionnels trouvent de nombreuses offres intéressantes. Dans le domaine des techniques de soudage notamment, *reichelt* offre un programme complet : appareils performants et accessoires adaptés, soudure sans plomb, aspiration des fumées, stations à air chaud, stations de dessoudage, aiguilles à souder, étain, flux décapants, tout y est et bien plus encore. La station de soudage numérique LF-3000



offre un excellent rapport qualité-prix et de nombreuses caractéristiques, particulièrement intéressantes dans le cadre d'une utilisation professionnelle. Son élément chauffant offre un temps de montée en température très court et une précision élevée.

Avec une plage de puissance allant jusqu'à 90 W et une plage de températures entre 100 et 520 °C, la station de soudage *Xytronic* est idéale pour les soudures sans plomb conformes à la directive RoHS. Grâce à un régulateur de tension-tension résiduelle doté d'une isolation galvanique, les composants sensibles aux décharges électrostatiques peuvent être soudés sans risque.

De série ce fer à souder fonctionne sous une basse tension de 36 V. Il est doté d'une aiguille à souder chauffée de l'intérieur avec un capteur de température intégré, pour une précision remarquable de $\pm 3^{\circ}\text{C}$.

www.reichelt.com

Nouveau multimètre numérique à 5 chiffres 1/2

Keithley Instruments propose à un prix intéressant son nouveau modèle de multimètre à double affichage, à 5 chiffres 1/2, utilisable à la fois pour tout type d'applications et sur bancs de test. Comparé aux meilleurs DMM 5-1/2 chiffres de sa catégorie, le modèle 2110 offre une vitesse de mesure jusqu'à 10 fois supérieure, une précision 2 fois supérieure en mode DCV, un prix inférieur et une plus grande souplesse d'intégration. Il est également équipé de fonctions de déclenchement externe, d'un tampon de stockage interne et d'une interface GPIB (en option).



Les modes de mesure du modèle 2110 sont les suivants : mesures de tension DC (1000V max.) et de courant (10A max.), mesures de tension AC (750V max.) et de courant AC (10A max.), mesures de résistance, en deux ou quatre fils (100M Ω max.), mesures de température, de fréquence (10Hz-300kHz), mesures de la période et de capacité (gammas de 1nF à 100 μF). Un très grand nombre de fonctions mathématiques et les modes de mesure de diode et de continuité sont également disponibles. Le convertisseur A-D programmable et les paramètres de filtre permettent d'obtenir le meilleur rapport signal/bruit pour améliorer la précision de mesure. Programmé sur 4 chiffres 1/2, il est capable d'effectuer 50 000 lectures/sec.

Le double affichage de l'instrument peut présenter simultanément les résultats de deux mesures, comme par exemple la mesure d'une tension DC et d'une température. Les utilisateurs sont donc capables de surveiller les variations de température sans interrompre d'autres mesures. Sa face avant intuitive le rend assez simple à utiliser pour les étudiants sans sacrifier ses performances de mesure ou sa flexibilité.

La température est habituellement le paramètre le plus mesuré et le modèle 2110 est capable de mesurer des températures avec des RTD, thermistances NTC ainsi qu'avec des thermocouples de type B, C, E, J, K, N, R, S et T. Une compensation de soudure froide intégrée assure une meilleure précision de mesure des thermocouples sans la nécessité d'autres branchements.

Pour simplifier les conversions des mesures brutes en données utilisables, les fonctions mathématiques les plus utilisées sont disponibles dans le modèle 2110 : pourcentage, moyenne, min/max., valeurs nulles, limites, la fonction mX+B, dB et dBm. Les logiciels KI-Tool et KI-Link permettent de contrôler à distance l'instrument, d'enregistrer les mesures et d'afficher les graphes des résultats en fonction du temps puis de les importer sous Word ou Excel pour éditer des rapports. Les pilotes LabVIEW et IVI ajoutent encore plus de flexibilité de programmation.

Avec les options d'interface USB seules ou USB/GPIB, l'intégration du modèle 2110 dans des systèmes de test avec d'autres instruments est simple, comme par exemple avec la ligne évolutive des alimentations de la série 2200DC. L'interface USB compatible TMC et l'interface GPIB en option permettent de réutiliser simplement des programmes existants utilisant les commandes SCPI.

<http://www.keithley.com/>

Actionneurs piézoélectriques innovants à haute résolution nanométrique

Ce nouveau moteur linéaire *PiezoLEGS Caliper 20N* de *PiezoMotor*, dont la résolution est nanométrique, permet d'obtenir une compacité et une facilité d'intégration dans les platines de déplacement. L'objectif est la miniaturisation pour dégager plus d'espace, afin de faciliter les mouvements. Il est tout à fait adapté aux applications dans les domaines du laser, de la photonique et de l'optique en particulier où la précision extrême dans le positionnement est requise et la répétabilité du mouvement est un paramètre important. L'entraînement direct, sans engrenage ni vis micrométrique, permet d'éliminer tout jeu mécanique et également de garder la position avec un minimum de consommation de courant et une élévation thermique quasi nulle.

Les *PiezoLegs Caliper 20N* fournissent une force de maintien maximale de 20N pour une résolution de l'ordre de quelques nanomètres et la course de déplacement maximale est de 40 mm. Ils peuvent remplacer des micromoteurs conventionnels ou pas à pas associés à des vis micrométriques tout en étant plus précis, plus compacts et offrir un gain du coût de l'équipement total non négligeable.



De la théorie à la pratique grâce aux maquettes NI miniSystems

National Instruments annonce les maquettes *miniSystems*. Ces versions miniaturisées de systèmes réels, comme p. ex. des dynamomètres de banc de puissance à rouleaux ou des réseaux électriques intelligents, sont développées en collaboration avec les principaux fournisseurs pédagogiques pour rendre pertinentes et pratiques des matières théoriques comme les mathématiques et les sciences en début de cycle des études d'ingénieur.

Les plates-formes de laboratoire ont toujours été inaccessibles pour un usage courant dans les cours classiques et sont souvent déconnectées des outils professionnels utilisés sur le terrain dans l'industrie et la recherche. En adoptant l'approche logiciel-matériel intégrée de la conception graphique de systèmes, les maquettes *miniSystems* éliminent les restrictions financières, temporelles et spatiales qui empêchent les enseignants de dépasser l'enseignement théorique, en leur permettant d'impliquer les étudiants grâce à une expérimentation pratique. Les maquettes *miniSystems myQuake*, *myVTOL* (décollage et atterrissage verticaux), *myTemp*, *myGrid* offrent aux

étudiants une liaison concrète entre leurs études théoriques et le monde du travail.

«Nous avons constaté à quel point l'approche de la conception graphique de systèmes aide les scientifiques et les ingénieurs à être plus innovants en leur donnant un ensemble d'outils et un langage unifiés» souligne Dave Wilson, *director of academic programs* chez *National Instruments*. «Nous sommes en train d'étendre cette approche à notre plate-forme pédagogique pour permettre à tous les étudiants, quels que soient leur profil ou leurs parcours, d'accéder aux innovations des grands scientifiques, ingénieurs et enseignants du monde entier.»



ni.com/minisystems

129 Go de capacité d'une mémoire NAND intégrée : Samsung lance la production en masse

Un mois seulement après avoir commencé à livrer les produits eMMC (*embedded multimedia card*) Pro avec des capacités de 16, 32 et 64 Go, *Samsung Electronics* lance la production en masse d'une mémoire intégrée de 128 Go pour les téléphones intelligents, les tablettes et autres appareils mobiles de la prochaine

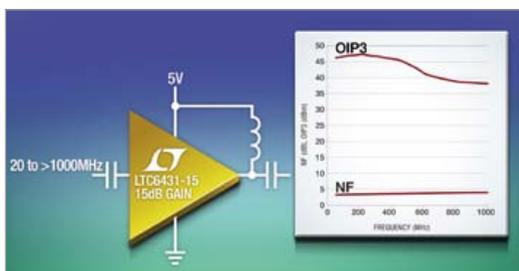


génération. Avec 128 Go de mémoire intégrée, les nouveaux combinés et tablettes mobiles pourront stocker jusqu'à 15 fichiers vidéo full HD équivalents à 8 Go.

Les cartes eMMC Pro de *Samsung* lisent séquentiellement les données jusqu'à 140 Mo/s et écrivent jusqu'à 50 Mo/s, soit cinq fois plus vite que les cartes de mémoire de classe 10. Pour la lecture et l'écriture aléatoires, elles traitent jusqu'à 3500/1500 IOPS (entrées et sorties par seconde).

Ce périphérique à très haut-débit utilise la mémoire NAND de 64 Gb de *Samsung* avec une interface *DDR toggle 2.0* (gravure 20 nm), gérée par son propre contrôleur et un progiciel de gestion de mémoire flash.

Bloc amplificateur de gain de 15,5 dB



Excellente linéarité au-delà de 1 GHz

Linear Technology annonce le LTC6431-15, qui, chargé par 50 Ω , atteint une large gamme dynamique de 20MHz à 1GHz et au-delà. Il est fabriqué à partir d'un processus de pointe, à base de SiGe, et est disponible en deux classes de

fonctionnement. À 240 MHz, l'OIP3 de la classe A est typiquement de 47 dBm et est entièrement testé et garanti pour atteindre au minimum 44 dBm. Le niveau de bruit est de 3,33dB. La consommation est inférieure à 450 mW. Cette combinaison d'un faible niveau de bruit et d'une faible distorsion offre la large gamme dynamique nécessaire aux applications de télécommunications de fréquence intermédiaire, aux performances élevées, et à la télévision par câble (CATV).

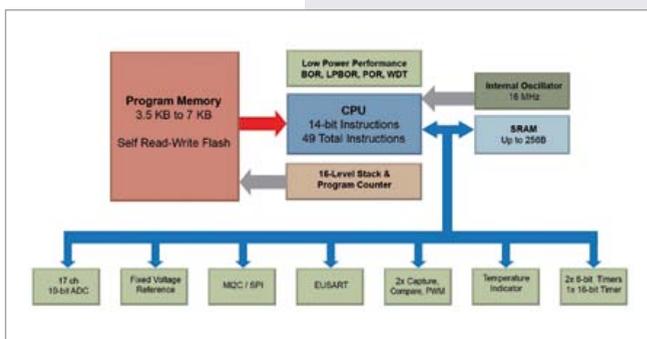
Le LTC6431-15 est facile d'emploi. Il est inconditionnellement stable et l'entrée et la sortie sont adaptées en 50 Ω , en interne, ainsi les composants externes requis

sont les condensateurs de blocage CC et une bobine d'arrêt pour le passage de la polarisation. Il procure une excursion linéaire de la tension de sortie supérieure à 2 V, alimenté sur une seule alimentation de 5 V. Le gain à 1 dB de point de compression (P1dB) est typiquement de 20,6 dBm et la linéarité est maintenue pour une puissance de sortie jusqu'à 17dBm.

Résumé des caractéristiques : LTC6431-15

- Bande passante : 20 MHz à 1700 MHz
- Gain en puissance : 15,5 dB
- OIP3 à 240 MHz :
47 dBm typ., 44 dBm min. en classe A
45 dBm typ. en classe B
- Figure de bruit 3,33 dB à 240 MHz
- Excursion linéaire de sortie > 2 V
- 20,6 dB de gain à P1dB
- Fonctionnement avec entrée et sortie unipolaires adaptées 50ohms
- Alimentation unique 5 V, 90 mA
- Stable en toute circonstance
- Boîtier QFN de 24 broches, 4 mm x 4 mm

Faible consommation et gestion tactile automatisée pour les nouveaux microcontrôleurs à 8 bits de Microchip



Ces nouveaux microcontrôleurs PIC16F1512/13 de la gamme PIC 8 bits *enhanced middle-range* à très faible consommation associent de nombreux périphériques numériques et analogiques à la technologie XLP (très faible consommation), requise pour améliorer la durée de vie des batteries de nombreuses applications. Grâce à ces caractéristiques, ces microcontrôleurs à 28 broches d'usage général sont parfaits pour un vaste éventail d'applications : appareils électroménagers et médicaux, produits grand public, marché automobile, etc. Outre un courant actif atteignant 30 μ A/MHz et une consommation en mode veille de 20 nA, ces microcontrôleurs intègrent un CAN 17x10 bits, un EUSART, une interface I2C™/SPI, deux modules de capture/ comparaison/ PWM, 7 ko de mémoire Flash et 256 octets de RAM.

Par rapport aux autres membres de la famille des PIC16F151X, ces deux nouveaux composants offrent des

Vous souhaitez remplacer votre pyromètre ? Souhaitez-vous mieux ?

Les caméras thermiques trouvent progressivement leur place dans la boîte à outils de chaque électricien. Un nombre croissant d'utilisateurs passent du pyromètre à une caméra thermique pour mesurer les températures sans contact.

Si vous décidez de remplacer votre pyromètre et acheter une FLIR i7 ou une FLIR E60 avant la fin de l'année, FLIR vous offre les packs suivants.

Si vous commandez une FLIR i7, vous recevrez GRATUITEMENT les accessoires suivants :

une sacoche, une ceinture porte-outils, un chargeur à brancher sur votre véhicule lorsque vous allez sur votre prochain site d'intervention

Si vous commandez une FLIR E60, vous recevrez GRATUITEMENT : une sacoche, une batterie supplémentaire, le logiciel FLIR Reporter, qui vous aidera à créer des rapports d'inspection complets

À propos de l'imagerie thermique

L'imagerie thermique est obtenue à l'aide de caméras dotées de capteurs spéciaux qui «voient» l'énergie thermique émise par les objets. L'énergie thermique, ou infrarouge est une lumière invisible pour l'œil humain car sa longueur d'onde est plus élevée que celle que l'œil perçoit. C'est la partie du spectre électromagnétique que nous ressentons sous forme de chaleur. L'imagerie infrarouge nous permet de voir ce que nos yeux ne voient pas. Les caméras thermiques produisent des images de ce rayonnement. L'imagerie thermique produit une image claire des différences de tempé-

rature entre les objets. C'est un excellent outil pour les applications industrielles, l'inspection des bâtiments, la recherche & développement et les applications d'automatisation.

Elle permet de voir dans l'obscurité totale, à travers la brume, à grande distance, à travers la fumée. Elle est aussi utilisée pour la sécurité et la surveillance, dans le secteur maritime, dans l'automobile, pour la lutte contre les incendies et dans de nombreuses autres applications.

FLIR Systems est le premier concepteur et fabricant au monde pour l'imagerie thermique. Le nombre de ses caméras en usage dans le monde se chiffre par milliers. *FLIR Systems* possède sept usines de fabrication, et compte plus de 3.200 spécialistes de l'infrarouge.



possibilités analogiques accrues, comprenant un convertisseur analogique-numérique (CAN) avec diviseur de tension capacitif (CVD) intégré pour l'implémentation du tactile capacitif. Grâce à une logique de contrôle supplémentaire, la détection tactile à échantillonnage automatique, qui permet de réduire la taille du code et de moins solliciter le processeur, est rendue possible. Ce composant offre également la commande automatique de la boucle de garde ainsi qu'une capacité d'échantillonneur-bloqueur programmable, qui conviennent mieux aux capteurs tactiles ou de proximité de grand format.

Les microcontrôleurs PIC16F1512/13 sont compatibles avec la suite complète d'outils de développement standard *Microchip*, dont l'environnement de développement gratuit (IDE) MPLAB® X, ainsi qu'avec les outils d'évaluation F1, dont la plateforme d'évaluation F1 (DM164130-1).



le monde d'Elektor

Jan Buiting (Elektor)

🇨🇦 Le mesureur de rayonnement amélioré d'Elektor bientôt sur ConSat-1

Bonjour, je suis Michelle Boyce, de l'équipe *Space Concordia*. Nous avons récemment construit un satellite (ConSat-1) qui a gagné le *Canadian Satellite Design Challenge*.

L'une des charges embarquées est un détecteur de rayonnement sans pièces mobiles, basé sur les articles suivants d'Elektor :

- *Mesure de rayonnement gamma avec une photodiode BPW34*, Burkhard Kainka, juin 2011
- *Compteur alpha/bêta/gamma*, Burkhard Kainka, novembre 2011

Le projet faisait partie d'un *Educational Outreach Program* mené avec *Space Concordia*

Bien cordialement, Michelle Boyce (Canada)

Dès sa publication, notre mesureur de rayonnement avait fait un tabac un peu partout. Y compris au Canada, d'où vient ce message qui a fait beaucoup d'effet aux responsables de la com' d'Elektor. Nous avons demandé à Michelle d'écrire un article pour Elektor.

Amis lecteurs enseignants, vous aussi, envoyez-nous des infos sur ce que **vous** et **vos** étudiants faites avec **nos** circuits. Nous en parlerons ici.

🇬🇧 Le mesureur de rayonnement amélioré d'Elektor à l'université d'Oxford

« Bonjour Thijs, je suis du département de physique de l'université d'Oxford. Nous travaillons à la construction d'un circuit semblable au vôtre destiné à être placé dans un ballon à hélium pour mesurer les rayons cosmiques. Nous avons acheté le circuit de détection de rayonnements (nous avons essayé plusieurs kits) en espérant pouvoir le tester dans notre labo. En dépit des nombreuses différentes sources auxquelles nous avons accès, nous ne sommes pas parvenus à faire fonctionner ce kit. Je pense l'avoir assemblé et blindé correctement, mais ne parviens pas à trouver ce qui ne va pas. Actuellement il se déclenche lorsque le comparateur est réglé sur L(evel)68 (pas au dessus). Le signal sur K4, vu

à l'oscilloscope, a une amplitude d'environ 500 mV(cc).

Le temps presse et j'aimerais beaucoup que le ballon parte avec un kit qui fonctionne. Pourriez-vous nous prêter un circuit fonctionnel ou bien voulez-vous que nous renvoyions le nôtre (il est monté dans un coffret métallique) ? »

Jeff Lidgard (Royaume-Uni)

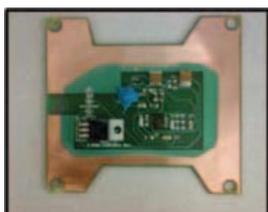
Quoi, vous êtes de l'université d'Oxford et vous n'arrivez pas à faire marcher un des kits les plus populaires d'Elektor ? On vous en envoie un qui marche, Jeff. En retour, on voudrait des nouvelles des mesures faites par votre ballon-sonde.

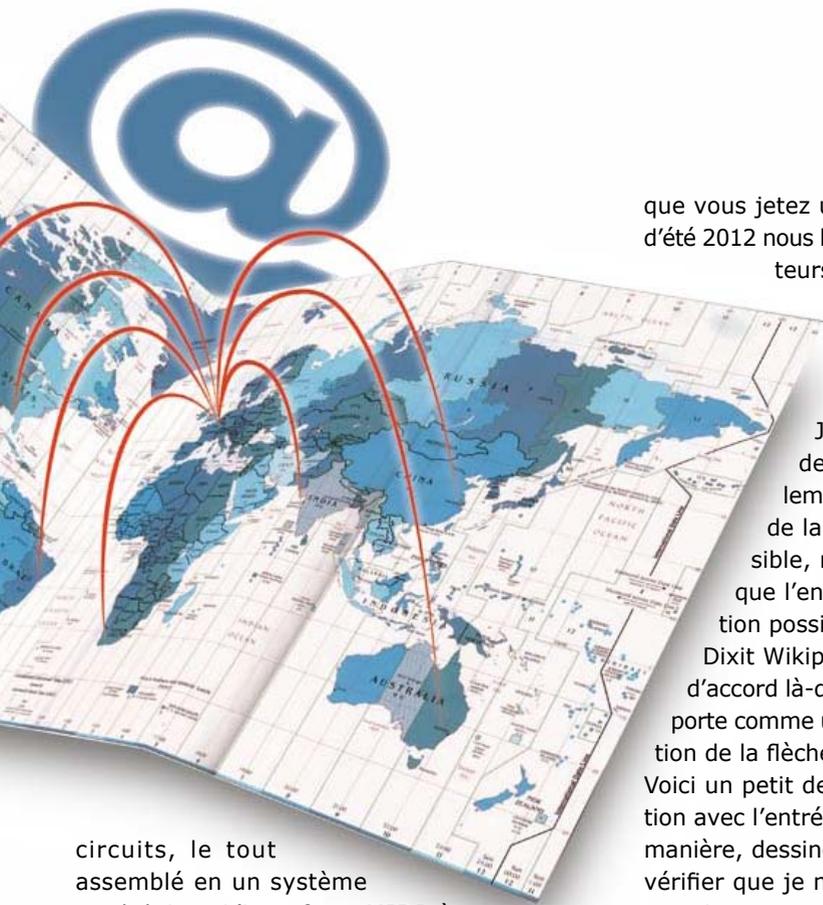
🇩🇪 Un titanesque Formant d'Elektor sur Youtube



Siegfried Brückner possède très certainement la plus grande collection de modules du synthé *Formant* sur Terre et dans la galaxie. À vue de nez, son instrument fait 3 m de large et un bon mètre de haut. Pas moins de 110 modules Formant, 500 boutons et des dizaines de câbles reliés à un séquenceur, une pédale de basse et un clavier produisent le son inimitable qu'est celui du synthétiseur Formant [1].

Siegfried nous écrit : « Cette phase de mon projet comprend 160 modules décrits dans les livres **Elektor Formant 1 & 2**, ainsi que d'autres





que vous jetez un œil à la page 78 du numéro d'été 2012 nous laissons la parole à un de nos lecteurs, Klavs Rommedal, et à Sadettin Commert, le créateur du puzzle.

Salut amis Elektorien,

Je ne suis ni triste ni en colère de ne pas avoir gagné, mais totalement perdu suite à la publication de la solution (Entrée 2). C'est possible, mais mon petit cerveau me dit que l'entrée 1 est également une solution possible ?

Dixit Wikipedia, j'espère que nous serons d'accord là-dessus, une diode zener se comporte comme une diode normale dans la direction de la flèche :-)

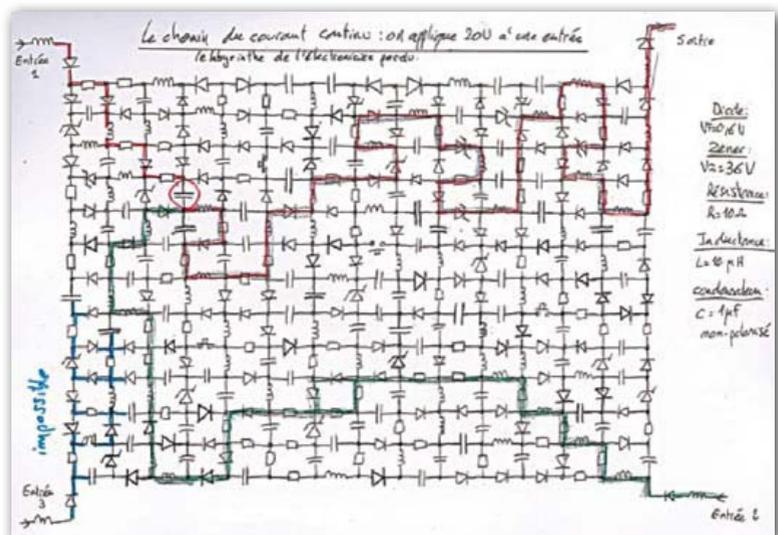
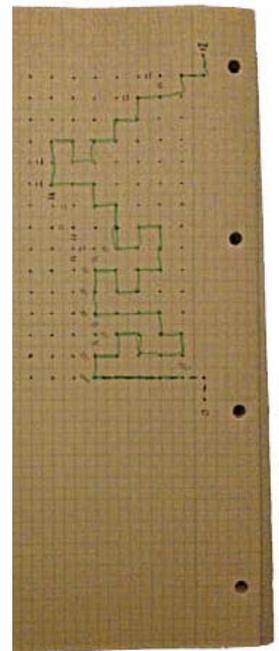
Voici un petit dessin à main levée de ma solution avec l'entrée 1. Pourriez-vous, de la même manière, dessiner la solution pour l'entrée 2 et vérifier que je ne me suis pas trompé ; si oui, pourriez-vous me dire où ?

Sincèrement, Klavs Rommedahl (Danemark)

Chers Jan et Klavs

Vous avez raison, bien sûr, la version du labyrinthe imprimée dans le magazine était une version préliminaire, qui contient de toute évidence une erreur. Je vous envoie la version finale corrigée. Une des résistances aurait dû être un condensateur ! À cause de cette erreur, il y a deux solutions possibles.

Toutes mes excuses, Sadettin Commert (France)



circuits, le tout assemblé en un système modulaire. L'interface MIDI à 8 voies permet de commander le Formant à la fois par MIDI et par CV/Gate.

Mes nouveaux claviers ont 4 et 5 octaves plus un pédalier à 13 boutons. Chaque clavier ou pédale possède une interface C/V.

Des interfaces MIDI supplémentaires ont été ajoutées pour traiter d'autres sons avec le Formant. Elles permettent de jouer en monophonie ou en polyphonie sur des *expanders* (ce sont des modules de synthés sans clavier) dont les sons peuvent être triturés à loisir.

En avril 2012, lors de la rencontre *Happy Knobbing* à Fischbach, en Allemagne, dont le thème était la production de sons à l'aide d'un système modulaire [2], j'ai fait des démos de mon Formant Titan. Malheureusement, l'enregistrement n'est pas terrible. Prochain spectacle à Munich le 17 novembre. »

Nous avons sorti notre Formant de la benne et jouons avec de temps en temps, dans la cave [3]. A vos casques !

[1] http://youtu.be/_5cRr7og4Vc

[2] <http://youtu.be/ynQyGT6t9tU>

[3] <http://youtu.be/JOhxLE3-ecI>

Perdu dans le labyrinthe électronique

Créer un labyrinthe qui ne possède qu'une solution unique est aussi dur que le résoudre. Pendant

 **Circuit Cellar**
fête ses 25 ans (19_{hex})

En septembre 2012 nous avons fêté aux États-Unis le 25e anniversaire du petit, mais non moins costaud, frère d'Elektor : le magazine *Circuit Cellar*. En ma qualité de responsable des éditions en

anglais d'Elektor, j'ai été invité au dîner (discours et bavardages obligatoires), à la table des anciens et des nouveaux collègues de CC, avec une brochette de leurs meilleurs contributeurs pour la technique et la rédaction : Ken Davidson, Dave Tweed, Robert Lacoste (*tiens, un Français !*), Ed

Nisley (*avez-vous lu ses articles sur les moteurs pas-à-pas dans Elektor il y a quelques mois [1] ?*) et Jeff Bachiochi. Une fois les patrons hors de portée d'oreille, les conversations n'ont pas tardé à s'orienter vers l'électronique embarquée. Belle fête ! Merci à nos hôtes Steve Ciarcia et Hugo Van Haecke, l'ancien et le nouveau patron de CC.

[1] pratique de la commande des moteurs pas-à-pas (1)
www.elektor.fr/120358



Publicité


L'ORIGINAL DEPUIS 1994
PCB-POOL
Beta LAYOUT

Pochoir CMS gratuit
avec chaque commande
"Prototype"

EAGLE : Bouton de commande
pcb-pool.com/download-button
20% de remise sur votre première commande de PCB

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



Beta
LAYOUT
create electronics

www.pcb-pool.com

PCB-POOL est la marque déposée de Beta LAYOUT GmbH. Chaque marque déposée appartient à son propriétaire respectif.

Schaeffer AG

FACES AVANT ET BOÎTIERS
Pièces unitaires et petites séries à prix avantageux.

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

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

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

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



Schaeffer AG · Nahmitzer Damm 32 · D-12277 Berlin · Tel +49 (0)30 8 05 86 95-30
Fax +49 (0)30 8 05 86 95-33 · Web info.fr@schaeffer-ag.de · www.schaeffer-ag.de

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

+49 (0)4422 955-333

+49 (0)4422 955-360

Qualité professionnelle à des prix réduits !

Technologie de boîtier

Technique des mesures



Boîtier pour système

- Pour l'installation de circuits imprimés au format Europe
- 3 unités de hauteur: 112,8 mm, profondeur: 236 mm

GEH SG 1-19 **66,50** Ouverture 10" : 203,8 mm

pro Ma
GEH SG 1-10
46,65
Ouverture 19" : 427,8 mm

Hammond Serie 1550W, IP66

- Boîtier en aluminium coulé avec un design solide
- Joint en mousse autocollant, 4 vis en acier inoxydable



| | I x L x H (mm) |
|--------|----------------------|
| 1550WQ | 4,10 60 x 55 x 30 |
| 1550WA | 4,10 89 x 35 x 30 |
| 1550WB | 5,40 115 x 64 x 30 |
| 1550WD | 6,10 115 x 64 x 55 |
| 1550WC | 6,95 115 x 90 x 55 |
| 1550WE | 10,65 171 x 121 x 51 |

Boîtier-étui de haute qualité

- Étanche à la poussière
- Protégé contre les jets d'eau



| | I x L x H (mm) |
|--------------|-------------------|
| BOPLA ET-205 | 4,80 52 x 50 x 37 |
| BOPLA ET-206 | 7,15 65 x 50 x 37 |
| BOPLA ET-208 | 8,95 98 x 64 x 38 |



Plus de choix : <http://rch.it/6C>

Petit boîtier en plastique

- Boîtier standard
- 6 bagues de fixation pour platines



| | I x L x H (mm) |
|-----------|-------------------|
| GEH KS 21 | 0,98 72 x 50 x 21 |
| GEH KS 28 | 0,98 72 x 50 x 28 |
| GEH KS 35 | 1,10 72 x 50 x 35 |
| GEH KS 42 | 1,10 72 x 50 x 42 |



Besoin de l'atelier

Connecteur

Alimentation en courant électrique



NEUTRIK NE-8FDP

9,65



Douille à bride

- Ethercon
- RJ45 sur RJ45

Connecteur Cinch standard

- Avec protection antitorion

CSP RT rouge
CSP GE jaune
CSP WS blanc

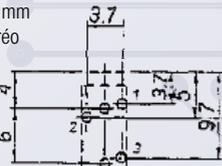
la pièce **0,09**



Tous les connecteurs Cinch <http://rch.it/6D>

Prise jack encastrable de haute qualité

- 2,5 mm
- Stéréo



LUM 1501-03 **0,69**

La gamme Lumberg <http://rch.it/6E>

Prises encastrées professionnelles, verrouillables

- 6,35 mm
- Mono/stéréo



NEUTRIK NJ-3FP **5,50** Boîtier métallique
NEUTRIK NJ-3FPB **5,95** Boîtier chromé noir

Technologie réseau et PC

Outils de développement

Technologie satellite et TV



ARDUINO UNO
27,95

Tableau Microcontrôleur

Plateforme prototype open-source basée sur la technologie Atmel ATmega328

- 14 interfaces I/O, 6 entrées analogiques
- Oscillateur à quartz 16 MHz, collecteur ICSP, Reset



Tableaux microcontrôleur

- Basé sur ATmega 2560
- 54 ergots I/O digitaux
- 16 entrées analogiques
- Oscillateur à cristal 16 MHz
- 4 UART, collecteur ICSP, Reset



ARDUINO MEGA
49,50

WizNet W5100 Ethernet

- Basé sur ATmega 328
- 13 ergots I/O digitaux (4 pour sortie PWM)



ARDUINO ETHERNET

49,50

Boîtier Raspberry-Pi professionnel

- Configuration pratique – montage parfait
- Raccordement de câbles intuitif
- Indicateur de statut
- 4 pieds du boîtier
- Taille de platine : jusqu'à 85 x 56 mm

Livraison sans platine

TEK-BERRY

4,95



Clemens Valens
(Elektor)

Ne ratez pas le coche

Quand j'étais étudiant, mes profs affirmaient qu'il est difficile de suivre les techniques tant elles changent vite. « Pourquoi disent-ils cela ? Moi j'ingurgite sans peine toutes les infos que je peux trouver » pensais-je. Aujourd'hui il ne me reste plus un cheveu sur le caillou, mes pectoraux se font la malle et je dois bien avouer que je ne digère plus aussi bien le flot ininterrompu de nouveautés. Je ne suis pas le seul, ça me console d'autant plus que d'autres s'en sortent encore moins bien que moi.

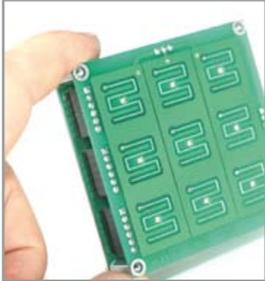
Ainsi, l'autre jour, j'étais en voiture avec un collègue pas chauve mais grisonnant, lui au volant et moi lisant les panneaux ; arrivés à quelques kilomètres de notre destination, je sors mon téléphone tactile pour lancer *Google Maps*. Pour activer l'appareil endormi, je passe mon doigt sur l'écran de veille en faisant un mouvement particulier, comme une signature. Vous connaissez sans doute ça sous le nom de *pattern lock*. Pas mon collègue, qui lui n'avait jamais rien vu de tel. Tout excité par ce qu'il prend pour un gadget dernier cri, il a failli emboutir le camion devant nous. En l'informant que ce procédé de déverrouillage est utilisé depuis des années, je l'ai froissé. Alors, pour compenser, j'ai ajouté que le sujet restait d'actualité puisque même sur notre site .LABS, quelqu'un parlait de construire soi-même un dispositif de ce genre.

Comme par hasard il n'est pas passé trois jours avant que je reçoive un message de l'auteur de ce projet qui m'annonce justement qu'il est achevé. « J'ai mis en ligne une vidéo, vous l'avez vue ? » Non, je n'avais pas visité cette page depuis un moment, mais cette fois c'est moi qui ai fait des grands yeux : description claire et précise, photos de qualité, excellente vidéo, la présentation qui tue. Une telle mise en bouche, moi ça me donne faim. J'ai donc voulu m'y mettre et en construire un moi-même. Vous en aurez des nouvelles bientôt.

Quel plaisir si sur notre site .LABS les projets étaient tous aussi bien présentés ! Si vous-même n'y arrivez pas encore trop bien, ne renoncez pas. Faites de votre mieux, avec les moyens du bord. Même sans outil professionnel pour vos schémas, vous pouvez les dessiner à la main, ce n'est pas grave. Souciez-vous d'être compréhensible. Donnez de votre projet des informations engageantes et aussi complètes que possible. Ce qui se conçoit bien s'énonce clairement. Surprenez-nous, vous ne le regretterez pas.

Le projet *pattern lock* mérite le détour. Vous vous y mettez aussi.

DOT LABS : www.elektor-projects.com/project/android-style-capacitive-sensing-pattern-lock.12407.html



L'origine d'une espèce

J'adore Wikipedia, pas vous ? Voyez ce que j'y ai découvert récemment : *"Les psocoptères (Psocoptera) sont un ordre d'insectes, sous-classe des ptérygotes, section des néoptères, super-ordre des hémiptéroïdés. Ils ont un développement de type hémimétabole. On en connaît des traces fossiles remontant au Permien (il y a 248 à 295 millions d'années) et plus de 5500 espèces ont été décrites, réparties en 41 familles et trois sous-ordres. La plupart n'ont été décrites que récemment."*

Comme je n'ai pas fait de grec et que les insectes c'est pas mon truc, je n'y pige rien. Mais alors pourquoi t'en parles, Charles ? Parce que

dans l'ordre des *psocomorphes* est apparue récemment une nouvelle espèce, très bruyante, le *PSoCaMorph*, un synthétiseur de musique, découvert précisément dans les locaux virtuels de .LABS. Ce n'est plus de l'entomologie. C'est de l'électronique et c'est de la musique : le but est

de construire un synthé analogique avec les techniques les plus modernes de programmation sur puce : *Programmable System-on-Chip*

(PSoC). Il n'y a pas plus modulaire que les synthés de musique analogiques, ils adorent la vie de famille. Il y a ainsi des familles d'oscillateurs commandés en tension (VCO), des lignées de filtres commandés en tension (VCF) et des clans d'amplificateurs commandés en tension (VCA). Et bien d'autres variantes encore. Les psocoptères sont apparus il y a 250 millions d'années, les synthés analogiques ne datent pas d'hier non plus. Si leur son inimitable vous fascine, rejoignez-vous au groupe de doux dingues qui dans .LABS donne forme au projet *PSoCaMorph*.

Et en plus vous pouvez gagner un kit de développement PSOC 5. Pour cela, il suffit de participer activement au projet.

Penchez votre oreille sur le nid de *PSoCaMorph*, ici, dans ce trou : <http://www.elektor-projects.com/project/the-psocamorph.12269.html>





Les préférés de la rédaction

EDITOR'S CHOICE

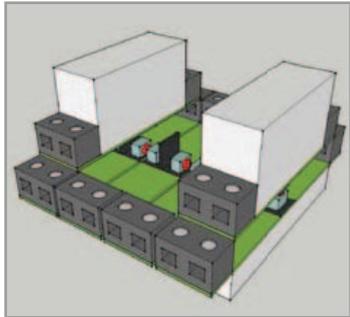
Ce n'est pas parce que .LABS est ouvert à tous que la rédaction n'a pas son mot à dire. Ces messieurs ont fait une sélection de projets qu'ils veulent publier. Malheureusement l'auteur de certains d'entre eux (OP) ne répond plus aux messages. Si vous mettez en ligne l'une ou l'autre de vos idées, repassez de temps à autre sur le site pour voir s'il y a du nouveau pour vous. Si on ne peut même plus vous joindre, on ne risque pas non plus de publier votre proposition.

Voici la liste des projets que la rédaction trouve publiables (les titres sont en anglais, comme sur le site, pour que vous puissiez les y retrouver facilement) :

Geiger Counter Data Logger with WLAN Interface

L'auteur Marcus Vohburger élabore un système complet de mesure et de surveillance de la radioactivité. Il dit que construire un compteur Geiger, c'est assez simple. Du coup, il propose d'autres fonctions.

www.elektor-projects.com/project/geiger-counter-data-logger-with-wlan-interface.12264.html



Mains Gate: Programmable Relay & Energy Monitor

Le père de ce projet, John Hind, écrit : Mon idée est de regrouper cinq circuits de l'édition d'été 2012 d'Elektor pour en faire un circuit de surveillance du réseau électrique domestique, quelque chose de complet avec une flopée de fonctions de mesure, de protection et d'enregistrement de données.

www.elektor-projects.com/project/mains-gate-programmable-relay-energy-monitor.12461.html

Instrument à vent MIDI

Ce projet s'appuie sur une expérimentation antérieure de l'auteur dont était sortie une usine à gaz. Si nous le publions, ce sera simplifié et modernisé. Ça ne vous tenterait pas de participer à l'élaboration d'un instrument à vent avec un capteur de pression et de flux d'air ?

www.elektor-projects.com/project/simple-midi-wind-instrument.12277.html



Anémomètre et girouette

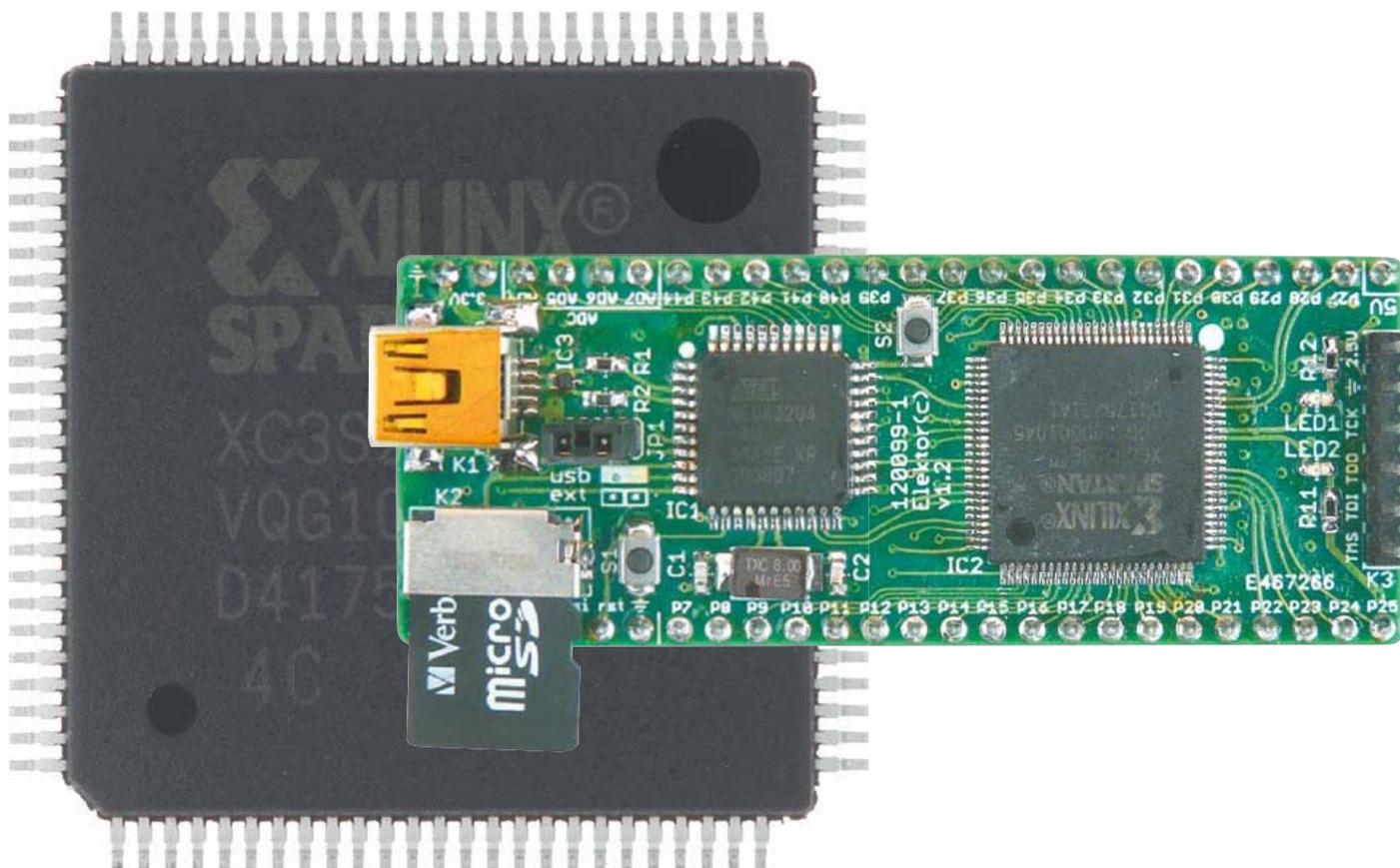
Pris en sandwich entre deux capots en plastique, un petit circuit imprimé avec seulement 4 composants au milieu : un transistor chauffé délibérément et trois thermistances. Le vent décentre la colonne d'air chaud produite par le transistor ; les NTC rendent compte du mouvement d'air. Malin, non ?

www.elektor-projects.com/project/wind-direction-and-speed-instrument.12480.html

Vous êtes encore là à me lire ?

Allez donc plutôt mettre votre grain de sel sur www.elektor-projects.com

et l'homme créa sa puce facile, la vie, avec un puissant FPGA



Clemens Valens & Raymond Vermeulen
(Elektor)

Sans aucun doute, l'un des composants les plus universels, mais aussi des plus complexes dans l'électronique actuelle, c'est le FPGA, dédale de portes et d'autres morceaux de circuit pour composer, sur une seule puce, le circuit dont vous rêvez. Le laboratoire Elektor a conçu une nouvelle carte d'expérimentation pour FPGA, pour faciliter la vie des électroniciens désireux de se familiariser avec cette famille de logique programmable. Cette carte servira aussi de support à une série d'articles qui expliqueront comment s'y prendre. Qui a peur de Virginia, euh ! des FPGA ? Plus personne, depuis qu'il existe une carte FPGA d'Elektor ! Viens ici, ma puce...

FPGA est l'abréviation de *Field Programmable Gate Array*, que l'on peut traduire littéralement par « matrice de portes logiques programmable sur le terrain ». Autrement et mieux dit : de la logique à configurer par l'utilisateur. Un FPGA

contient une grande quantité de portes logiques, de bascules, d'unités de mémoire et bien d'autres choses encore, que le concepteur arrangera à sa guise. On en déduit que la différence fondamentale entre FPGA et microprocesseurs (ou micro-

contrôleurs), c'est que ces derniers exécutent un programme, mais pas le FPGA qui est un circuit.

Le berceau : la simple matrice

Au commencement était... la logique programmable à partir d'une matrice de quelques dizaines de portes logiques dont on pouvait arranger les entrées et sorties plus ou moins à son goût. Au passage on se débarrassait aussi de ces grands circuits imprimés remplis de portes logiques individuelles, maintenant réunies dans une seule puce à fonction complexe. Avantage supplémentaire, la possibilité d'adapter rapidement les fonctions en reconfigurant la puce sans avoir ni à couper de pistes entre les portes ni à ajouter de nouvelles liaisons. Cette sorte de logique programmable ressemble à la mémoire morte programmée (**Read-Only Memory**) ou reprogrammable (**Erasable Programmable Read-Only Memory**) qui offrent de semblables possibilités. Prenons un exemple simple : une porte ET (**AND**) à deux entrées, A et B, et une sortie Q. Sa table de vérité s'écrit comme suit :

| A | B | Q |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Changez les noms des entrées en A1 et A0 et appelez la sortie D0, vous avez une ligne de donnée et deux lignes d'adresse pour sélectionner quatre emplacements de mémoire. Placez-y les quatre valeurs 0, 0, 0 et 1, vous avez une porte ET en mémoire. Vous pouvez aussi y mettre autre chose et utiliser plus ou moins d'adresses. Vous arriverez ainsi facilement à composer des fonctions logiques complexes.

Mais ce qu'une mémoire ne peut pas faire, c'est inscrire le résultat d'une fonction autre part dans la mémoire. Même si elle contient de nombreux registres, une mémoire ne peut pas écrire elle-même. Seul l'utilisateur peut y lire et y écrire. C'est pourquoi la logique programmable a été dotée de registres dans lesquels on peut mémoriser des résultats utilisables ailleurs dans la puce. Par registres, il faut comprendre des bascules ordinaires ou des unités de mémoire d'un seul bit. La **figure 1** montre une version simplifiée d'une cellule logique telle qu'on en trouve dans un FPGA.

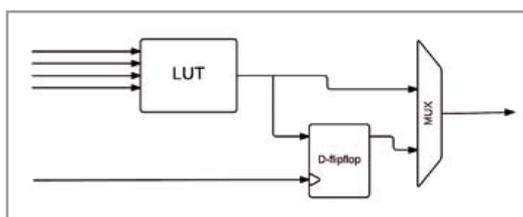


Figure 1.
Diagramme fonctionnel d'une version simple de cellule logique dans un FPGA.

On peut faire plein de choses avec des bascules, surtout quand on en a beaucoup. Vous pouvez en faire des compteurs, des registres à 8 bits ou de toute autre largeur. Mais aussi des registres à décalage pour opérer des conversions de parallèle à série et vice versa ou ralentir un flux de données. En bref, une matrice de portes logiques, c'est tout bon, mais c'est encore mieux quand il y a aussi de la mémoire !

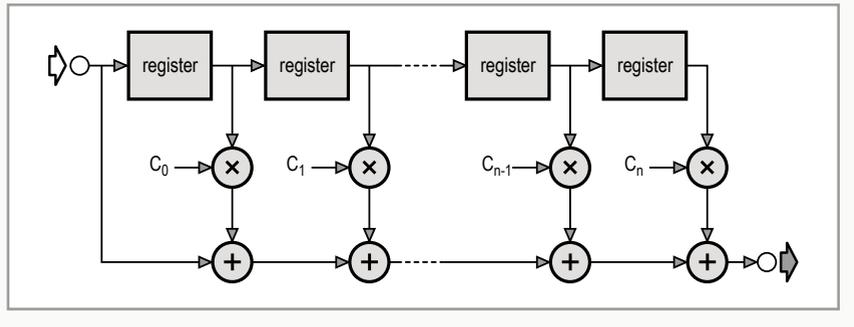
Plus complexe = plus de possibilités

Au fil des ans, la logique programmable s'est complexifiée, comme le dit l'appellation CPLD, pour *Complex Programmable Logic Device*. Les progrès de la technique ont permis, voire rendu nécessaire, de développer des composants de manière à les faire ressembler davantage à de la mémoire vive (**Random Access Memory**,) qu'à de la ROM. Privée de tension d'alimentation, une RAM ne conserve aucune donnée et il en va de même de cette sorte de logique programmable ; aussitôt que la tension d'alimentation disparaît, la configuration de la puce s'efface. Il était donc nécessaire de pouvoir configurer la puce lors du démarrage sans l'aide d'un appareil spécial de programmation, sur le terrain, pourrait-on dire. Ainsi naquit le FPGA. Pour faciliter la reconfiguration, son fichier est souvent gravé dans une mémoire externe que le FPGA va lire au démarrage. Cela fait vraiment penser à un système à microprocesseur qui va aussi chercher son programme dans une mémoire extérieure. Notez qu'il existe même des FPGA qui gardent en permanence leur configuration.

Au cours du temps, les fabricants de FPGA ont réussi à rassembler sur une même puce une quantité gigantesque de portes et de bascules. Cette intégration en constante progression autorise de nouvelles applications et des débits de données extrêmement élevés, mais engendre de nouveaux défis, l'organisation de la puce, par exemple. Pour arriver à relier entre elles, librement, plusieurs millions de portes, il faut une multitude de « petits fils ». Certaines de

Figure 2.

Un filtre numérique se compose d'une ligne à retard, de multiplieurs et de compteurs. Le signal d'horloge qui assure la synchronisation n'est pas représenté ici. La ligne à retard est formée de registres, les constantes sont stockées dans des registres et les multiplieurs et compteurs sont faits de fonctions logiques et de registres. Tout cela peut être facilement implémenté dans un FPGA, ce qui explique pourquoi les FPGA conviennent si bien aux applications de traitement numérique du signal.



ces connexions peuvent devenir longues, par exemple quand il faut brancher une sortie d'un port en haut à gauche à une entrée en bas à droite. Or ces distances constituent un handicap si le débit des signaux est élevé, les impulsions brèves. Des retards apparaissent, car les flancs des signaux s'arrondissent au point que le circuit perd la cadence et que le synchronisme entre toutes les portes est menacé.

Ces contraintes ont imposé la division de l'architecture globale des FPGA en blocs fonctionnels. L'arrangement de ces blocs est choisi de manière à optimiser leur utilisation dans la plupart des applications standard. Pour des applications spéciales, des fonctions particulières sont disponibles qui font qu'un FPGA n'est pas l'autre. Les blocs que l'on rencontre actuellement sur tous les FPGA, mais sous des noms différents, sont :

- les blocs logiques
- les blocs d'E/S

- les blocs de calcul
- les blocs d'horloge

À côté de ces blocs aux fonctions générales, vous pouvez en trouver d'autres tels que des processeurs, de la mémoire flash et des périphériques comme des contrôleurs Ethernet, des contrôleurs de mémoire, des convertisseurs A/N ou des interfaces de communication série SPI, I²C, etc. Avec de telles fonctions annexes, le FPGA tend vers un système complet, aussi parle-t-on parfois de *System-on-chip* (SoC) avec FPGA. On peut relier les blocs entre eux au moyen d'un système souple de liaisons configurables.

Les blocs

Envisageons successivement ces blocs fonctionnels.

Les blocs logiques

Les blocs logiques occupent la plus grande partie du FPGA et en forment le « tissu » (*fabric* en anglais) sur lequel on aurait peint l'application. L'exemple donné au début de cet article devrait se sentir chez lui dans ce type de bloc. Un bloc logique est un circuit configurable formé de portes et de registres, il est basé sur la LUT (*Look-Up Table*, table à consulter). La LUT se présente comme dans l'exemple de porte ET, mais généralement en plus grand, avec plus d'entrées. C'est un registre d'un nombre x de bits, seize par exemple, avec lequel on peut exécuter une fonction logique. La LUT peut également servir de mémoire et l'on parle alors de RAM distribuée, par opposition aux blocs de mémoire. En y ajoutant des multiplexeurs et un peu d'autre logique, on gagne en souplesse et un bloc logique peut aussi avoir une certaine capacité de calcul.

Comme un FPGA dispose de tellement de blocs de ce genre, vous pouvez en relier tout un groupe

Tableau 1. Blocs fonctionnels de la puce Xilinx Spartan 3E XC3S250E-4VQG100C.

| | |
|---------------------------|---|
| Portes | 250 000 |
| Blocs logiques | 612 |
| Blocs d'E/S | 66, dont 7 en entrée seule |
| Blocs de calcul | 12 multiplieurs |
| Blocs de mémoire | 12 pour un total de 221 184 bits (216 Kbit) |
| Mémoire distribuée | 39 168 bits (38,25 Kbit) |
| Blocs d'horloge | 4 |

Le nombre de blocs d'E/S dépend du boîtier de la puce, nous avons utilisé la variante VQ100 à 100 pattes. Un bloc logique, un *Configurable Logic Block (CLB)* en langage de Xilinx, se compose de quatre tranches (*slices*) qui contiennent chacune deux LUT à 16 bits dont la moitié peut servir de RAM distribuée, l'autre moitié est constituée de fonctions logiques. Chaque tranche contient aussi deux bascules, de sorte que chaque CLB dispose au total de 8 bascules (**fig. 3**).

en parallèle pour en faire un multiplieur ou un compteur. Un filtre numérique est constitué d'un certain nombre d'unités de mémoire et, ce qui vaut pour quelques multiplieurs et un compteur, s'applique aussi à une transformation de Fourier numérique (**figure 2**). Un bloc logique a donc en stock tous les ingrédients de base pour faire des filtres numériques ou des transformations de Fourier. Avec un seul de ces blocs, ça n'ira pas, mais si on en prend quelques-uns, ça marchera. C'est ce qui rend les FPGA particulièrement aptes au traitement numérique du signal. Inséré entre un convertisseur A/N et un CNA, un FPGA peut faire merveille en audio, vidéo, radio et dans bien d'autres domaines. Sans oublier la possibilité de traiter de nombreux canaux simultanément.

Aujourd'hui, on peut se procurer des FPGA qui recèlent plus d'un million de blocs logiques de cette sorte : Xilinx Virtex-7 XC7V2000T a 1,2 million de LUT et 2,4 millions de bascules. Cette puce totalise près de sept milliards de transistors. En comparaison, le processeur Intel à 10 cœurs Xeon Westmere-EX ne compte « que » 2,6 milliards de transistors.

Les blocs d'E/S

Les blocs logiques, tout autant que les autres fonctions dans la puce, doivent pouvoir communiquer avec le monde extérieur par les connexions de la puce. Celle dont on vient de parler, XC7V2000T en a 1 200. La communication passe par des blocs dits d'entrée/sortie.

Un bloc d'E/S est bien plus qu'une paire de transistors en montage *push-pull*. Il doit accepter différents standards de communication (DDR, PCI, LVTTTL, LVCMOS, LVDS, etc.), aussi bien en mode asymétrique que symétrique (différentiel) et être capable de le faire à un débit binaire extrêmement haut. Pour y arriver, ces blocs sont équipés de lignes à retard, de bascules de synchronisation et de convertisseurs parallèle/série, des SerDes, abréviation de *Serializer-Deserializer*. Les résistances de forçage haut ou bas n'y manquent pas et les résistances de bouclage non plus.

On regroupe les blocs d'E/S en banques qui disposent chacune de leur propre tension d'alimentation, de manière à ce qu'elles puissent gérer différents standards de communication, ce qui

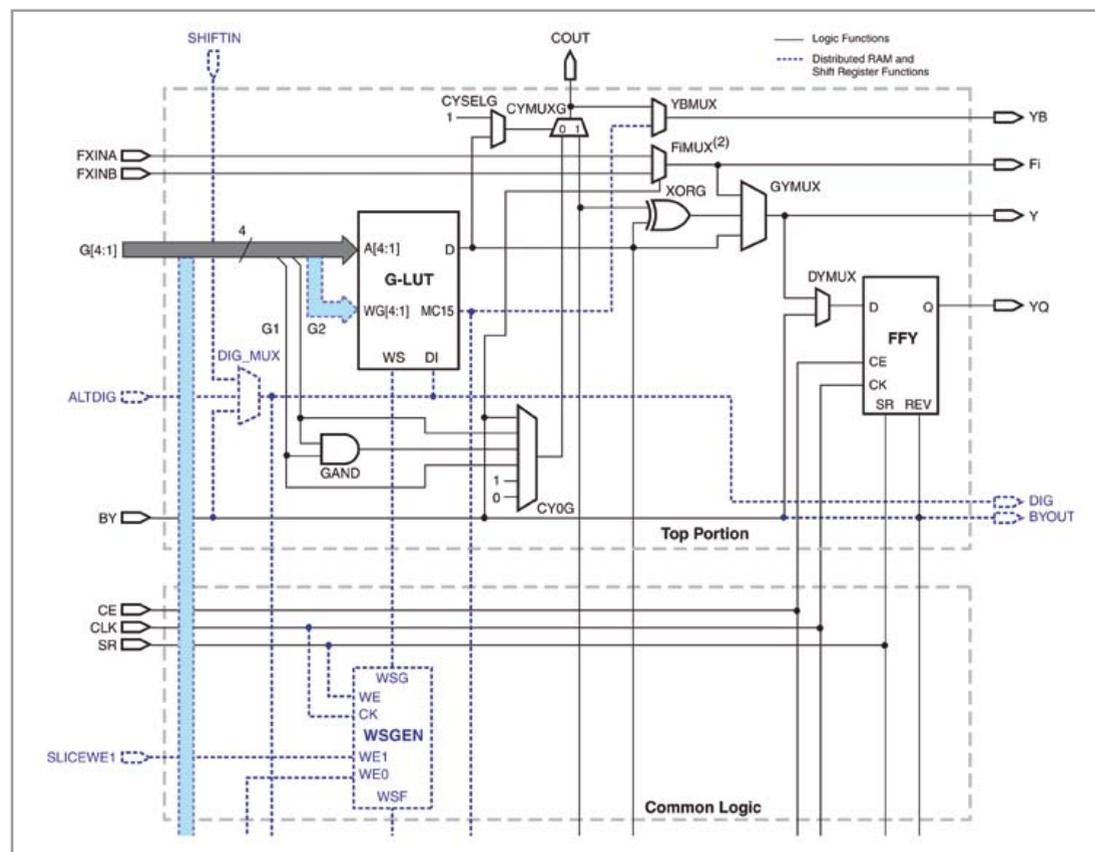


Figure 3. Le diagramme fonctionnel d'une partie de tranche. La partie manquante est celle du bas, appelée *Bottom Portion*, largement similaire à celle dans le rectangle du haut en pointillés et appelée *Top Portion*.

Le circuit

Comme on peut le voir dans le diagramme fonctionnel de la **figure 4**, le circuit se compose principalement d'un ATmega32U4, un FPGA Spartan3e, une section d'alimentation et un support pour carte microSD. Le FPGA est le centre d'intérêt du dispositif, voyez-en également le vrai schéma à la **figure 5**. Nous avons choisi un XC3S250E-4VQG100C, un modèle de la série Spartan3e de Xilinx. Il contient 250 000 portes système, plus de 5 500 cellules logiques et 612 unités logiques, sa fréquence maximale est de 572 MHz... la suite dans le **tableau 1**. Pour assurer au circuit imprimé des dimensions modestes, nous avons cherché un FPGA en petit boîtier. Résultat, c'est un CMS. La puce compte 100 broches dont 66 pour les E/S, ce qui doit convenir à la plupart des applications. Avec la quantité de logique à bord, on peut même se lancer dans des projets assez complexes. Le microcontrôleur du type ATmega32U4 est le petit touche-

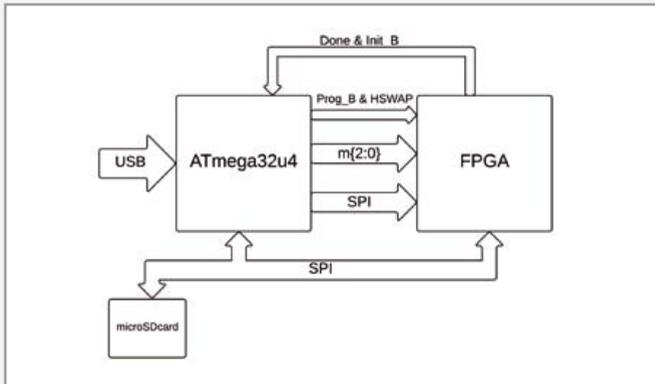


Figure 4. Le diagramme fonctionnel de notre carte d'expérimentation de FPGA révèle que le circuit se compose principalement d'un FPGA, d'un microcontrôleur et d'un connecteur pour carte microSD.

à-tout de la carte. Il assure la communication avec le PC par USB, il offre un port COM virtuel pour permettre au PC de traiter la carte microSD comme un périphérique de stockage de masse. C'est encore lui qui lit la carte mémoire et qui programme le FPGA. Nous avons pris un ATmega bien connu parce qu'il dispose de l'interface pour l'USB. C'est un modèle lent à 8 bits, mais il convient parfaitement dans cet emploi. Le micrologiciel qui lui est dédié est bien sûr disponible sur le site d'Elektor [1], il est préprogrammé sur le circuit imprimé complet que nous fournissons.

Le support pour carte microSD a été choisi particulièrement en raison du peu de place qu'il occupe sur la carte. Il est branché pour travailler en mode SPI. Il est compatible avec

la plupart des cartes microSD. Il possède un interrupteur de détection qui indique si une carte y est insérée. Les deux E/S inutilisées sont dotées de résistances de rappel haut vers le 3,3 V pour éviter qu'elles se trouvent dans un état indéfini. La section d'alimentation est assez étendue pour un circuit de ce format. C'est que le FPGA requiert trois tensions d'alimentation : 3,3, 2,5 et 1,2 V. La carte mémoire et le microcontrôleur utilisent aussi le 3,3 V. Des condensateurs supplémentaires assurent une bonne stabilité ainsi que le découplage des différentes tensions.

La carte est alimentée par la connexion USB si le cavalier JP1 est en place. Cette tension de 5 V est aussi disponible sur P26 de K5 sur le bord de la carte, pour un usage externe. Mais attention ! Les lignes d'E/S sur cette carte ne résistent pas à 5 V. L'inverse est aussi possible. En l'absence du cavalier JP1, on peut alimenter le circuit localement par P26 de K5 sous une tension de +5 V, si bien qu'une liaison USB n'est pas indispensable. En raison du grand nombre de signaux et de niveaux de tension, il a fallu concevoir pour ce projet une carte à quatre couches (**fig. 6**). Pas de panique, vous pouvez, pour un prix très raisonnable, commander chez Elektor un module assemblé [1].

Une méthode de configuration particulière

D'ordinaire, on ne peut pas programmer les FPGA à demeure : à chaque démarrage, il faut les reconfigurer. Sur la plupart des cartes de développement de FPGA, on le fait au moyen d'un PC et d'un programmeur JTAG, ce qui est également possible ici par K3. Sur les installations permanentes avec FPGA, on utilise souvent une mémoire flash raccordée au bus SPI. Lors du lancement, le FPGA lit alors sa configuration dans la mémoire flash.

Nous nous y prenons tout autrement (fig. 3) : un microcontrôleur charge la configuration dans le FPGA par le bus SPI. C'est ce qu'on appelle le mode *SerialSlave*, méthode habituellement réservée aux gros systèmes à plusieurs FPGA, mais rien n'empêche de s'en servir pour un seul.

Le microcontrôleur peut communiquer par SPI avec la carte microSD. Pour pouvoir l'utiliser avec un système d'exploitation (OS), elle doit être formatée en FAT16 ou FAT32. Le microcontrôleur doit donc disposer de ce système de fichier pour arriver à lire un fichier sur la carte microSD. Et pour rendre cette carte accessible à l'OS, il faut que le microcontrôleur soit capable d'interpréter les commandes dites SCSI pour les exécuter. En s'y prenant de la sorte, la marche à suivre est très simple. Créer un fichier de configuration dans le logiciel Xilinx ISE et

le copier sur la carte de mémoire avec l'explorateur Windows. Le circuit imprimé y apparaît en fait comme une clé USB avec une lettre de lecteur attribuée. Il n'y a plus qu'à renommer le fichier en « config.bin ». On peut alors démonter la clé dans l'OS, puis mettre le circuit imprimé à zéro pour qu'il charge la nouvelle configuration au démarrage.

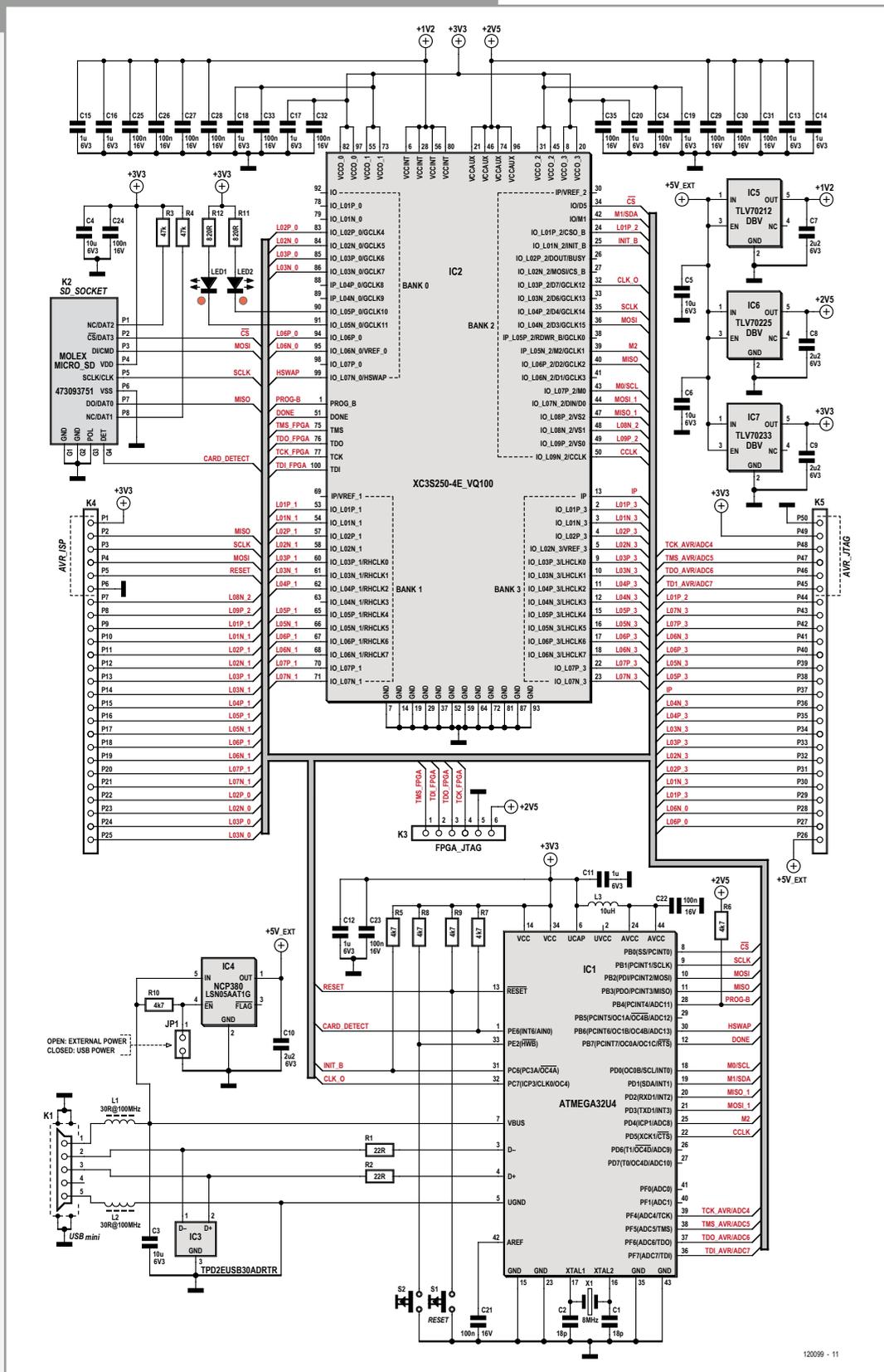
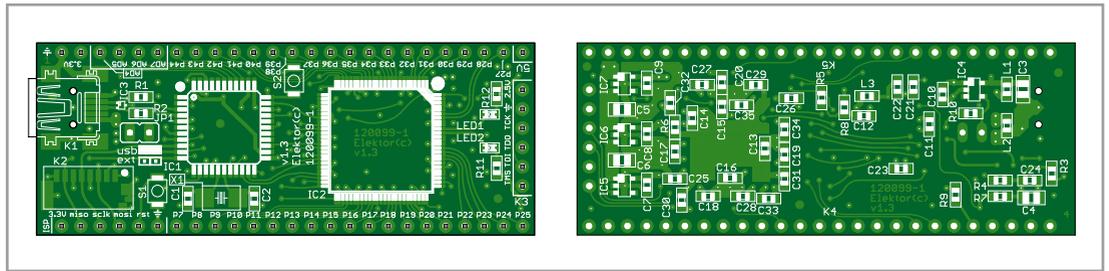


Figure 5. Voici le vrai schéma de la carte d'expérimentation de FPGA. Le schéma complet occupe une page mais le circuit réel tient sur un circuit imprimé de seulement 65 x 25 mm.

Figure 6.

Avec le concours de quatre couches de cuivre, nous avons pu réaliser toutes les interconnexions sur une aussi petite surface de circuit imprimé. Ce module est disponible assemblé, prêt à l'emploi.



favorise la souplesse du FPGA. Pour maintenir raisonnable le niveau de consommation et de rayonnement électromagnétique, l'utilisateur a la possibilité de régler la raideur des flancs et le maximum de courant par sortie.

Les blocs de calcul

Comme les utilisateurs de FPGA ont souvent recours à des blocs logiques pour calculer, les

fabricants ont aussi intégré des blocs de calcul tout préparés dans leurs produits. Il s'agit souvent de multiplieurs, mais certains fabricants vont plus loin et ajoutent des fonctions de DSP qui facilitent des calculs fréquents comme $a + (b \cdot c) \rightarrow a$ (*multiply-accumulate*).

Les blocs de mémoire

Outre la RAM distribuée sous forme de LUT, les FPGA contiennent aussi des blocs de RAM dont le volume total peut atteindre les mégaoctets. Intégrer de la mémoire dans la puce peut y accélérer énormément le traitement des données. Mais cela permet aussi de réaliser tout un système à microcontrôleur dans un seul FPGA. En dotant des blocs de RAM de deux ports indépendants, on peut les faire servir de tampons de données (FIFO) qui permettront, par exemple, de synchroniser des systèmes qui travaillent à des vitesses différentes. La largeur de mot de la mémoire est variable, ce qui fait que des blocs à 1 bit peuvent s'utiliser de manière interchangeable avec des blocs à 32 bits.

Les blocs d'horloge

Même si théoriquement il est possible d'implémenter des fonctions en asynchrone, c.à.d. sans signal d'horloge, c'est rarement une bonne idée sur des systèmes complexes. En raison des différences de distance à parcourir par le signal, il se produit des écarts et des flancs partis au même moment n'arrivent pas toujours à destination simultanément dans le circuit. Cela peut entraîner des états intermédiaires indésirables qui conduisent à des erreurs. La solution est de pratiquer une sorte d'échantillonnage du signal à intervalles réguliers, comme si on en prenait des instantanés photographiques. Cela retarde bien le fonctionnement, mais remet au pas tous les flancs. Cette synchronisation s'opère avec un signal d'horloge et des bascules. Toutefois, les lois physiques s'appliquent au signal d'horloge comme à n'importe quel autre signal, il subira

Figure 7.

Le brochage du FPGA utilisé. Les broches orange et vertes sont raccordées aux connecteurs de la carte, les bleues forment l'interface avec le microcontrôleur. Les LED se branchent sur les deux broches jaunes, tandis que les noires forment l'interface JTAG accessible sur un connecteur séparé.

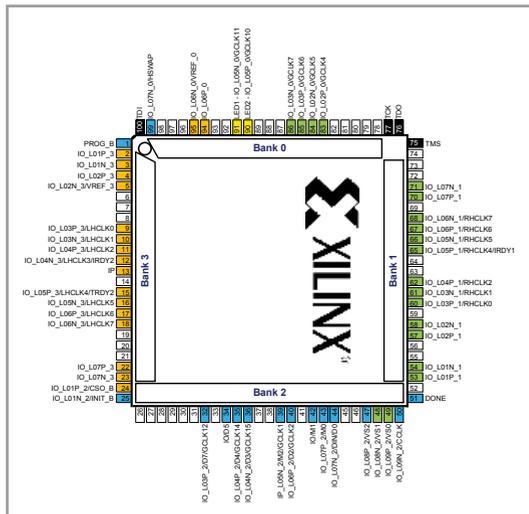


Figure 8.

Les pattes des connecteurs d'extension. Les broches orange et vertes correspondent à celles de la figure 6. Ces connecteurs permettent la programmation du microcontrôleur. On peut aussi utiliser comme entrées analogiques les broches du port JTAG AVR.

| | | | | | |
|----|---------|------------------------|------------------------|----|----------|
| 1 | 3.3V | | | 50 | GND |
| 2 | AVR ISP | MISO | TCK_AVR/ADC4 | 49 | AVR JTAG |
| 3 | | SCLK | TMS_AVR/ADC5 | 48 | |
| 4 | | MOSI | TDO_AVR/ADC6 | 47 | |
| 5 | | RESET | TDI_AVR/ADC7 | 46 | |
| 6 | GND | | | 45 | |
| 7 | 48 | IO_L08N_2/VS1 | IO_L01P_2/CSO_B | 24 | 44 |
| 8 | 49 | IO_L09P_2/VS0 | IO_L07N_3 | 23 | 43 |
| 9 | 53 | IO_L01P_1 | IO_L07P_3 | 22 | 42 |
| 10 | 54 | IO_L01N_1 | IO_L06N_3/LHCLK7 | 18 | 41 |
| 11 | 57 | IO_L02P_1 | IO_L06P_3/LHCLK6 | 17 | 40 |
| 12 | 58 | IO_L02N_1 | IO_L05N_3/LHCLK5 | 16 | 39 |
| 13 | 60 | IO_L03P_1/RHCLK0 | IO_L05P_3/LHCLK4/TRDY2 | 15 | 38 |
| 14 | 61 | IO_L03N_1/RHCLK1 | IP | 13 | 37 |
| 15 | 62 | IO_L04P_1/RHCLK2 | IO_L04N_3/LHCLK3/IRDY2 | 12 | 36 |
| 16 | 65 | IO_L05P_1/RHCLK4/IRDY1 | IO_L04P_3/LHCLK2 | 11 | 35 |
| 17 | 66 | IO_L05N_1/RHCLK5 | IO_L03N_3/LHCLK1 | 10 | 34 |
| 18 | 67 | IO_L06P_1/RHCLK6 | IO_L03P_3/LHCLK0 | 9 | 33 |
| 19 | 68 | IO_L08N_1/RHCLK7 | IO_L02N_3/REF_3 | 5 | 32 |
| 20 | 70 | IO_L07P_1 | IO_L02P_3 | 4 | 31 |
| 21 | 71 | IO_L07N_1 | IO_L01N_3 | 3 | 30 |
| 22 | 83 | IO_L02P_0/GCLK4 | IO_L01P_3 | 2 | 29 |
| 23 | 84 | IO_L02N_0/GCLK5 | IO_L06N_0/REF_0 | 95 | 28 |
| 24 | 85 | IO_L03P_0/GCLK6 | IO_L06P_0 | 94 | 27 |
| 25 | 86 | IO_L03N_0/GCLK7 | | 5V | 26 |

donc un retard, lui aussi, en traversant la puce. Certains fabricants de FPGA adoptent différentes solutions de retardement et de décalage inclus dans le bloc. On peut aussi distribuer les signaux d'horloge à travers des réseaux particuliers.

Une autre raison de passer par des blocs d'horloges se rencontre dans les systèmes où certaines parties travaillent à des fréquences différentes, on les appelle des domaines d'horloge. On peut alors configurer les blocs de telle sorte qu'ils fabriquent les différentes fréquences à partir de la même entrée d'horloge. Ces fréquences peuvent atteindre des valeurs très hautes, jusque dans les gigahertz, question d'augmenter la vitesse du circuit, bien entendu.

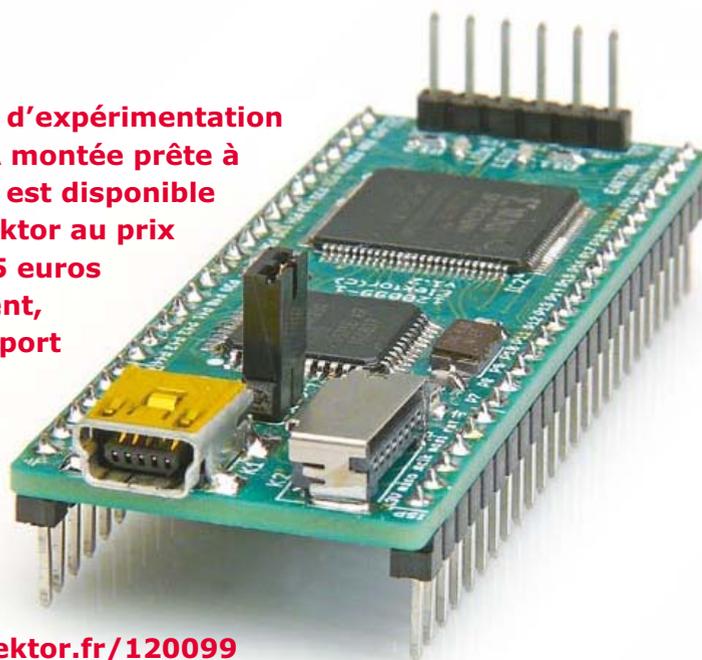
Dans la réalité

Après ces généralités d'introduction, intéressons-nous à la pratique et jouons un peu avec les FPGA. À cette fin, notre collègue concepteur Raymond Vermeulen a développé une petite carte d'expérimentation pour FPGA qui remédie à un grave défaut de la plupart des autres cartes du même style, à savoir la lourdeur de la méthode de programmation de la configuration dans le FPGA. Les outils logiciels proposés sont relativement complexes et il faut souvent y ajouter une console de programmation JTAG externe pour charger le fichier de configuration dans la puce. Raymond a imaginé une solution qui, pour la carte à FPGA, fait passer un PC pour une clé USB évoluée. En réalité, c'est une carte mémoire microSD résidente sur le circuit imprimé. Du coup, on peut copier le fichier de configuration exactement comme n'importe quel autre fichier. Lors du redémarrage du circuit, ce fichier, lu sur cette « clé », est chargé dans le FPGA. Ce qui résout avec élégance l'épineuse question des pilotes spéciaux pour le matériel de programmation.

Au mois prochain

La configuration des FPGA est une matière complexe et nous n'aurons pas trop d'une série d'articles pour exposer clairement comment arriver à d'intéressantes applications. Le mois prochain, si tout va bien, nous allons d'abord expliquer comment installer la suite internet Xilinx ISE et comment s'en servir pour faire clignoter des LED sur la carte d'expérimentation. Pour passer le temps avant la prochaine parution, nous vous recommandons de télécharger le logiciel [2]. Cette version internet de la suite est gratuite et représente un volume d'à peu près 6 Go. À ceux

La carte d'expérimentation de FPGA montée prête à l'emploi est disponible chez Elektor au prix de 59,95 euros seulement, frais de port en sus.



www.elektor.fr/120099

qui ne disposent pas d'une connexion rapide sur internet, il est conseillé de commander le DVD gratuit de ce logiciel auprès de Xilinx (**fig. 9**). En général, on le reçoit à la maison sous quinzaine.

(120099 – version française : Robert Grignard)

Liens :

[1] www.elektor.fr/120099

[2] <http://www.xilinx.com/support/download/index.htm>

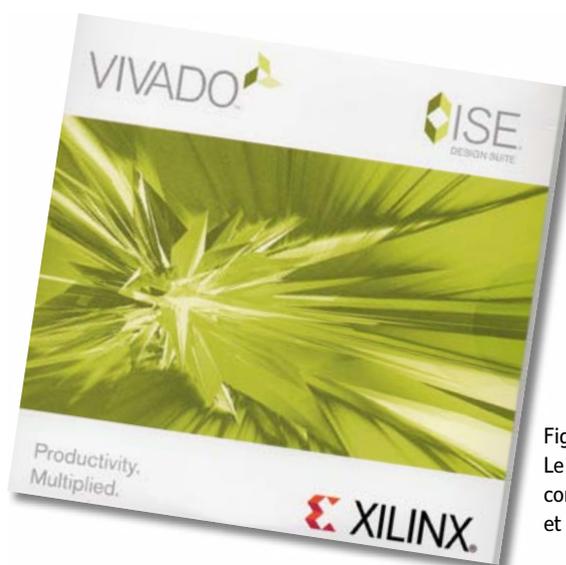


Figure 9.
Le DVD gratuit de Xilinx contient la suite internet ISE et d'autres outils pour FPGA.

mettez le cap sur Arduino (3b)

capteurs et téléphones Android discutent

David Cuartielles
(Espagne) et

Andreas
Goransson (Suède)

Le mois dernier, nous vous avons présenté un ensemble d'outils pour développer avec Android et l'Arduino Mega ADK. Ce mois-ci nous approfondirons et montrerons comment exploiter sur un téléphone les données issues de capteurs reliés à votre Arduino. Maintenant que tous les outils sont opérationnels, l'exécution de code sur votre téléphone et le dialogue avec votre Arduino seront très faciles. Commencez par les exemples livrés avec les bibliothèques ADK !



Les appareils Android sont partout, les façons de développer des applications pour ces appareils sont variées. Nous allons ici continuer à explorer l'utilisation de *Processing* [1] pour écrire, compiler et charger des applications sur les téléphones Android.

Le matériel

Dans le prolongement des expériences du mois dernier, voici la liste de ce qu'il vous faudra cette fois-ci :

- une carte Arduino Mega ADK ou une carte Arduino Uno avec un *shield* USB Host
- un téléphone Android doté d'Android OS 2.3.4 ou plus récent (plus il sera récent et mieux ça sera)
- un câble USB et un câble microUSB
- un *shield* TinkerKit (optionnel, vous pourriez utiliser une plaque à essais et des fils de connexion)
- des modules TinkerKit : potentiomètre et poussoir (également optionnel)

Remarque : ce mois-ci, il vous faudra impéra-

tivement posséder un téléphone Android pour faire des tests. Nous avons tout fait pour nous assurer que le code soit compatible avec la dernière version d'Android disponible au moment de l'écriture : la 4.1, nom de code *Jelly Bean*. *Conseil* : téléchargez sur le site d'Elektor le code complet correspondant à cet article, nous n'en montrerons ici que des extraits.

Bilan des acquis

Grâce à l'article du mois dernier, vous savez comment fonctionne le mode pour accessoires de Google et comment ils demandent aux appareils Android le démarrage d'une application lorsqu'on les y relie. Vous vous êtes familiarisés avec *Processing*, l'outil d'apprentissage Java, et avez créé puis chargé sur votre téléphone ou tablette des petites applis.

La prochaine étape sera de créer de nouvelles applis capables de tirer parti du lien avec un Arduino — utilisé comme accessoire — et votre appareil Android.

La bibliothèque USB Host

L'*Android Open Accessory Protocol* (AOAP) est le langage qu'utiliseront pour se parler votre appareil Android et votre carte Arduino. Techniquement, la carte Arduino sera un hôte USB et le téléphone un client USB. Les appareils Android, comme la plupart des appareils high-tech, comportent un périphérique compatible avec la norme *USB On The Go*. Cela permet au même connec-

teur USB (actuellement micro USB) d'être utilisé sur un client USB (comme une souris ou un clavier) ou un hôte USB (votre PC par exemple). L'AOAP spécifie que les accessoires devront jouer le rôle d'hôtes alors que les appareils Android devront se comporter comme des clients. Cela permet entre autres à ces derniers de recevoir de l'électricité de l'hôte (votre carte Arduino peut recharger votre téléphone).

Nous avons vu le mois dernier qu'afin d'établir la communication entre les deux, il faut intégrer la bibliothèque USB *Host* à votre code Arduino. Pour vérifier que cette dernière est bien installée, tentez de jeter un coup d'œil à l'un des exemples associés.

Remarque : rendez-vous dans le menu *File* → *Exemples* → *USB Host*, si vous n'y trouvez aucune entrée, cela veut dire que la bibliothèque USB *Host* n'est pas correctement installée dans l'EDI Arduino. Jetez un œil aux téléchargements pour cet article [3] et téléchargez-la ; vous allez en avoir besoin. Référez-vous à la **figure 1** pour un aperçu du matériel que nous allons utiliser.

Préparatifs : côté Processing

Voici les logiciels dont vous aurez besoin pour programmer vos applications Android :

- le SDK Android SDK mis à jour jusqu'à la dernière version des API
- *Processing* 2.0b3 ou plus récent (ceux qui ont lu le premier article remarqueront que nous utilisons maintenant la dernière version, vous devriez faire de même)
- l'outil Arduino ADK pour *Processing*
- la bibliothèque USB Arduino ADK pour *Processing*

Si vous avez mis en pratique les exemples du mois dernier, il ne devrait vous manquer que la dernière bibliothèque que nous avons volontairement omise.

Afin de vous assurer que votre carte puisse discuter avec votre téléphone, il vous faudra installer la bibliothèque complémentaire à *USB Host* côté *Processing*. Nous l'appelons Arduino ADK USB. Le paragraphe suivant vous indiquera comment procéder. Vous trouverez en **figure 2** un aperçu du matériel concerné.

Installation de la bibliothèque Arduino ADK USB pour Android

Dans l'archive qui accompagne cet article, vous trouverez un dossier *Processing* contenant un



Figure 1.
Un Arduino Mega ADK avec un *shield* TinkerKit et un téléphone.



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

dossier *libraries*. Copiez le contenu de ce dernier dans le dossier *libraries* de votre installation *Processing*. Celui-ci se trouve dans le dossier *sketchbook* de votre installation *Processing*, habituellement dans *My Documents/Processing* sur les ordinateurs Windows ou *Documents/Processing* avec MacOS et Linux.

Cette bibliothèque est accompagnée de quatre exemples. Nous vous recommandons de baser vos projets sur l'un de ces exemples, cela vous facilitera grandement la vie. Commençons par essayer l'exemple nommé *digitalWrite*. Votre carte Arduino, reliée à votre téléphone, déclenchera l'illumination d'une LED lorsque vous toucherez l'écran du téléphone. Dès que vous retirerez votre doigt de l'écran, la LED s'éteindra.

DigitalWrite

Chaque exemple est composé de deux parties. La première tourne sur la carte Arduino et se charge de la communication avec le téléphone, tout en effectuant les actions nécessaires sur la carte. La deuxième partie est le code compilé depuis *Processing* qui tournera sur l'appareil Android. Cette appli se chargera de lire et écrire des données sur le câble du téléphone ainsi que d'effectuer des actions sur l'écran ou encore interroger les

Listing 1. L'exemple DigitalWrite de la bibliothèque Arduino USB Host.

```
#include <AndroidAccessory.h>

// description de l'accessoire. Voici comment l'Arduino se présente à Android
char accessoryName[] = «DigitalWrite»; // votre carte Arduino
char companyName[] = «Arduino SA»;

// variables pour les LED
int ledPin = 10;

// compteurs
long timer = millis();

// initialisation de l'accessoire
AndroidAccessory usb(companyName, accessoryName);

void setup() {
    // démarrage du lien avec le périphérique par l'USB host
    usb.begin();
    // configuration de la patte avec la LED en sortie
```

capteurs du téléphone comme l'appareil photo, les accéléromètres, etc.

Commençons tout d'abord par analyser le code Arduino. Ouvrez l'EDI Arduino et tapez-y le code du **listing 1**. N'oubliez pas de le charger sur la carte une fois que vous aurez terminé !

Ce code d'exemple écoutera sur le port USB. Une fois que le téléphone sera connecté à la carte Arduino, la fonction `usb.isConnected()` retournera une réponse affirmative. Lorsque le téléphone envoie des données, `usb.available()` retourne une valeur non nulle que l'Arduino récupérera et stockera dans la variable

Listing 2. Un sketch Processing pour détecter les évènements sur l'écran tactile.

```
import cc.arduino.*;

// creation de l'objet ADK
ArduinoAdkUsb arduino;

void setup() {
    // orientation PORTRAIT
    orientation( PORTRAIT );
    // initialisation de l'objet ADK
    arduino = new ArduinoAdkUsb( this );
    if ( arduino.list() != null )
        arduino.connect( arduino.list()[0] );
}

void draw() {
    // dessin en fonction de l'état du lien avec l'Arduino d'un rectangle rempli
    connected( arduino.isConnected() );
```

```
pinMode(ledPin, OUTPUT);
}

void loop() {
  // sortie sur l'USB 10 fois par seconde
  if(millis()-timer>100) {
    if (usb.isConnected()) { // la connexion USB est-elle ouverte ?
      if (usb.available() > 0) { // y-a-t-il des données ?
        char val = usb.read();
        // 'a' illumine la LED, 'b' l'éteint
        if( val == 'a' )
          digitalWrite( ledPin, HIGH );
        else if( val == 'b' )
          digitalWrite( ledPin, LOW );
      }
      timer = millis();
    }
  }
}
```

val. Vous devriez observer un résultat similaire à la **figure 3**.

Côté *Processing* vous devrez écrire un programme qui guettera les événements sur l'écran tactile. Si la connexion filaire est active, le téléphone enverra un 'a' pour indiquer que quelqu'un a commencé à toucher l'écran. Il enverra un 'b'

pour indiquer que le doigt a été retiré de l'écran. Nous ne gérons que les événements *start* et *stop touching* afin de ne pas saturer le port de communication.

Ouvrez l'EDI *Processing* et recherchez l'exemple `adk_digitalWrite`. Vous devriez le trouver dans le menu *Examples* → *Contributed Libraries* →

```
}

public boolean surfaceTouchEvent(MotionEvent event) {
  if ( arduino.isConnected() ) {
    if ( event.getAction() == MotionEvent.ACTION_DOWN )
      arduino.write('a');
    else if ( event.getAction() == MotionEvent.ACTION_UP )
      arduino.write('b');
  }

  // si vous voulez que les variables pour motionX/motionY, mouseX/mouseY etc.
  // fonctionnent correctement, il vous faudra appeler super.surfaceTouchEvent().
  return super.surfaceTouchEvent(event);
}

[...]
```

ArduinoADKUsb si la bibliothèque a été installée correctement (cf **listing 2**).

Le code du **listing 2** se passerait presque d'explications. La seule chose qui pourrait vous contrarier est la déclaration de la méthode `surfaceTouchEvent(MotionEvent event)`. Java est un langage de programmation très puissant et permet de surcharger certaines des méthodes de *callback* existantes. Chaque fois que vous touchez l'écran de l'appareil Android, le système appellera la fonction `surfaceTouchEvent(MotionEvent event)`. Si vous la surchargez dans votre programme (en gros, si vous la réécrivez) vous pourrez lui faire faire ce que vous voulez.

Ici, nous détectons deux des nombreux événements du système : `MotionEvent.ACTION_DOWN` (lorsque l'on pose le doigt sur l'écran) pour envoyer 'a' à l'Arduino et `MotionEvent.ACTION_UP` (lorsque l'on retire le doigt) pour envoyer 'b'.

Allons un peu plus loin

Maintenant que vous avez réussi à faire fonctionner le premier exemple, vous avez sans doute envie de faire quelque chose d'un peu plus amusant, non ? Sur internet, on trouve des chats partout, c'est ce qui nous a décidés à créer l'appli *Elektor_Miau* du mois dernier. Nous voulions faire quelque chose de marrant tout en vous montrant qu'il est possible d'aborder la programmation avec des choses plus attractives que le classique "Hello World".

Ce mois-ci nous voulons continuer sur notre lancée et vous apprendre comment faire de petites animations avec *Processing*. Vous pourrez utiliser n'importe quel GIF animé. La seule condition est que vous devrez en exporter toutes les images et les nommer de la manière suivante : `s00.gif`, `s01.gif`, `s02.gif` ...

Vous pouvez bien entendu utiliser la même image que nous. Nous avons demandé à un illustrateur de nous dessiner un petit lapin qui sort d'un chapeau. Le but de l'exemple est de réaliser un jeu interactif sans prétention dans lequel vous utiliserez des capteurs (un bouton et un potentiomètre) pour interagir avec le lapin visible sur l'écran du téléphone (**fig. 4**).

Mieux vaut, cette fois-ci, commencer par regarder le code tournant sur le téléphone. Tout d'abord, l'animation est construite dans *Processing*. Si vous êtes porté sur Android, vous pouvez essayer de porter la bibliothèque *GIFdecoder* que nous avons utilisée le mois dernier pour jouer des GIF animés dans votre appli Android. Comme

nous voulons que la programmation soit simple, nous vous proposons d'animer votre lapin de la manière suivante : construisez un tableau d'images qui contiendra toutes les images de votre animation. L'idée vient de l'exemple *Processing Animated Sprite* de James Paterson [2] (voir le **listing 3** que vous trouverez sur [3]).

Remarque : le Listing 3 ne contient pas la classe *Animation* qui se chargera d'ouvrir les images `s00.gif`, `s01.gif`, etc. Nous vous recommandons de jeter un œil au listing complet qui accompagne l'exemple *Elektor_Simple_Animation*. Vous trouverez également un dossier nommé *data* dans le dossier du sketch *Processing* qui contient toutes les images faisant partie de l'animation (54 au total).

La totale

L'étape suivante est de modifier l'exemple pour y inclure les appels à la bibliothèque gérant la communication USB. Nous vous recommandons de fusionner le code précédent avec celui de l'exemple `adk_analogRead`. Le résultat devrait ressembler au **listing 4** [3].

Plusieurs choses méritent d'être mises en avant dans cet exemple. Commençons par le tableau `delays[i]`. Il sera utilisé pour contrôler la durée d'affichage de chacune des images de l'animation. Chaque fois que vous chargerez une nouvelle image, vous pourrez changer cette durée dans la condition `millis() - timer > delays[index]`. Nous vous encourageons à jouer avec les valeurs contenues dans ce tableau et voir comment cela affecte l'animation. N'oubliez cependant pas qu'il doit y avoir autant de nombres que d'images dans l'animation.

Le tableau `keyFrames[i]` contiendra la liste des images sur lesquelles le défilement s'arrêtera en attendant une action de l'utilisateur. Les actions dans notre cas se feront sur un poussoir. L'animation s'arrêtera sur ces images jusqu'à ce que l'utilisateur actionne le bouton relié au *shield* TinkerKit (**fig. 5**).

Examinons maintenant la ligne chargée de déterminer le nombre d'octets en provenance de la carte Arduino. Nous les traiterons trois par trois (`arduino.available() >= 3`) ; les informations seront codées comme suit :

- Octet 0 : doit valoir 255, c'est le marqueur utilisé pour repérer le début d'un bloc
- Octet 1 : la position du potentiomètre ;

mettez le cap sur Arduino

codée entre 0 et 254

- Octet 2 : l'état du bouton ; soit 0, soit 1

La position du potentiomètre sera utilisée pour changer la couleur d'arrière-plan de l'écran. Le bouton permettra de déclencher la prochaine séquence de l'animation.

Remarque : utilisez l'outil Arduino ADK Usb (menu *Tools* → *Arduino ADK* dans l'EDI Processing) pour charger ce programme sur votre téléphone. Une boîte de dialogue apparaîtra. Dans le champ *model*, saisissez le nom de l'appli que vous appelez depuis l'EDI Arduino. Le dernier paramètre permet de sélectionner la version du SDK pour le type de votre accessoire. Si votre téléphone ou tablette est animé par Android 3.0 ou plus récent, choisissez SDK v12 ; SDK v10 ne devra être utilisé que pour les appareils tournant sous Android 2.3.4. Si malheureusement votre appareil Android utilise la version 2.3.3 ou antérieure, vous ne pourrez pas utiliser le mode accessoire. Enfin, le code Arduino est pratiquement un copier/coller de l'exemple `adk_analogRead`. Les seules différences sont l'ajout de l'octet de marquage et du bouton et la limitation de la valeur lue sur le potentiomètre à 254. Vous trouverez la fonction *loop* de ce programme dans le **listing 5** [3]. Nous vous conseillons d'examiner l'intégralité du code source qui se trouve dans le dossier *Arduino/code* de l'archive présente sur le site web d'Elektor [1].

Le mot de la fin

Cet article conclut notre exploration d'Android et de ses liens avec Arduino. Vous avez maintenant toutes les cartes en main pour créer de nouveaux projets, construire des robots dont votre téléphone serait le cerveau, ou encore intégrer des capteurs dont votre appareil Android ne serait pas doté. Si vous créez un projet sympa sur la base de notre code, n'hésitez pas à nous en faire part (blog@arduino.cc), nous nous ferons une joie d'en informer la communauté.

(120573 – version française : Kévin PETIT)

Liens

[1] Le projet Processing : <http://processing.org>

[2] Animations à partir d'images dans Processing : <http://processing.org/learning/topics/sequential.html>

[3] www.elektor.fr/120573



Figure 3.
La LED s'illumine lorsque l'on touche l'écran.

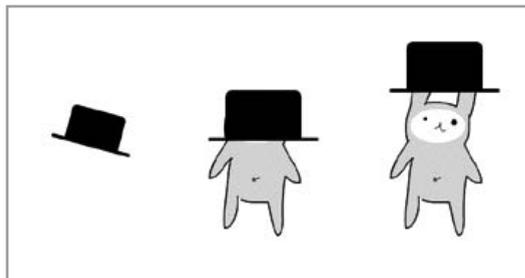


Figure 4.
Quelques images de notre lapin animé

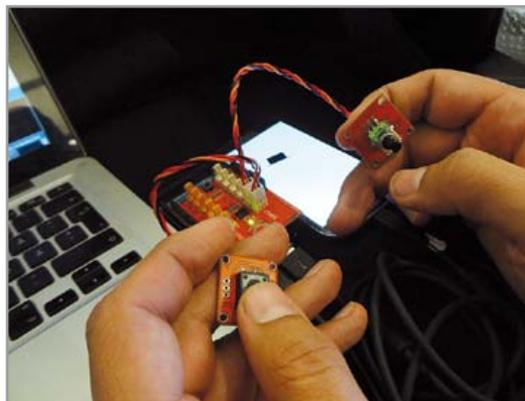


Figure 5.
Le bouton et le potentiomètre par dessus le *shield* TinkerKit.

Remerciements

L'équipe de conception Arduino tient à remercier l'équipe de Circuits@Home pour leur premier travail sur la bibliothèque USB *Host* pour Arduino et Philip Lindsay pour le débogage des toutes premières bibliothèques Arduino officielles pour Android. Merci également à Rodrigo Calvo qui nous a aidés à améliorer la bibliothèque USB *Host* afin qu'elle fonctionne avec les appareils Android 4.1, ainsi qu'à Laura Balboa pour ses illustrations.

interface USB 24 E/S

mesurer, commander et réguler facilement avec un PC

Uwe Altenburg (Allemagne)

Le câble présenté ici permet de lire et écrire 24 signaux tout ou rien via l'interface USB d'un PC. On peut choisir aussi d'acquérir jusqu'à huit entrées analogiques, d'émettre des signaux en modulation de largeur d'impulsion (MLI), de commander des servos et bien davantage. Nous proposons dans l'article un protocole simple de pilotage de l'interface et nous abordons la programmation du PC en C#.

L'ordinateur d'aujourd'hui prend toutes les formes imaginables, de bureau, portable ou tablette, ce qui a aussi un effet bénéfique sur les prix.

Si vous aviez, il y a quelques années, utilisé une mini-tour réformée pour piloter l'éclairage d'un aquarium, vous aurez probablement recours aujourd'hui à un mini-PC élégant avec une interface tactile. Cependant, le raccordement de l'électronique maison par le port USB n'est pas des plus simples et la production de signaux temporels critiques n'est pas le point fort d'un PC. De plus, il faut fixer son choix sur un protocole de communication et trouver un langage de programmation approprié. Nous apportons ici une réponse à toutes ces questions !

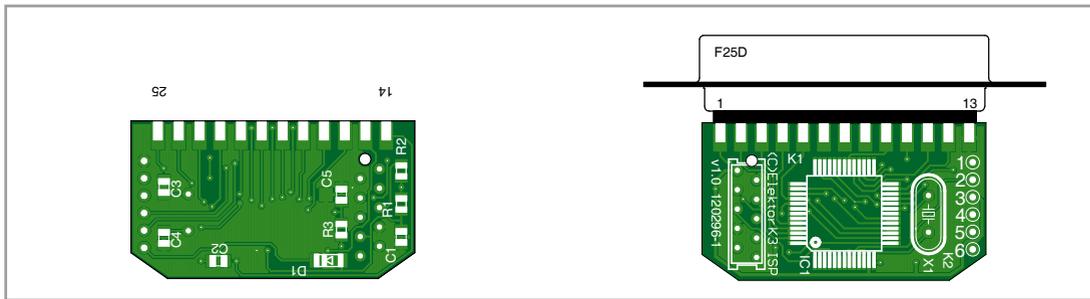


Figure 1.
La platine se loge dans un capot SubD-25.

Liste des composants

Résistances (CMS 0805, 5%) :

R1 = 100 k Ω
R2 = 4,7 k Ω
R3 = 100 Ω

Condensateurs :

C1, C2, C5 = 100 nF, 50 V, 10 %, X7R, SMD 0805
C3, C4 = 22 pF, 50 V, 5 %, NP0, SMD 0805

Semi-conducteurs :

D1 = LL4148 Minimelf
IC1 = R8C/25 CMS TQFP52

Divers :

Câble adaptateur USB/TTL-232R-5V-WE, disponible chez Elektor 080213-71 [6]
K1 = connecteur femelle SubD-25, à souder (77SD B25S Amphenol)
K3 = barrette 10 points MicroMatch à souder, avec connexions dessous (8-215079-0 TE Connectivity/Amp)
X1 = quartz 18,432 MHz, 18 pF, 50 ppm capot SubD-25, Flip Top (A-FT 25 Assmann)

Option :

fiche pour barrette 10 points MicroMatch (8-215083-0 TE Connectivity/Amp)

Le schéma

Depuis quelques années, le fabricant d'outre-Manche FTDI propose un câble adaptateur USB-TTL (TTL-232R-5V-WE [1]), disponible également sur l'e-choppe Elektor [2]. Au moyen de ce câble qui présente, en plus des signaux RxD, TxD, RTS et CTS, une alimentation +5 V, il est très simple de connecter un microcontrôleur à un PC. Les câbles contiennent le convertisseur USB-série FT232R, moulé dans la fiche USB. Rien de plus logique que de munir l'autre extrémité du câble d'une embase SubD à 25 points, avec un microcontrôleur installé dans le capot. Le microcontrôleur à 8 bits proposé ici est un membre de la famille R8C, souvent utilisée chez Elektor. Le R8C25 [3] offre une mémoire suffisante avec ses 64 Ko de flash et 3,5 Ko de RAM. Avec son convertisseur A/N à 10 bits et une puissante série de temporisateurs, il est idéal pour les applications d'automatisme. Les 25 broches de la prise SubD suffisent pour amener à l'extérieur les lignes de trois ports d'E/S et la masse indispensable. Le capot est assez grand pour recevoir une petite platine de 37 x 20 mm (**fig. 1**). Cette platine ne porte, en plus du microcontrôleur, qu'un quartz de 18,432 MHz, un connecteur pour la programma-

tion et quelques composants discrets. La livraison de platines terminées est au programme, voir les détails sur la page ouèbe de cet article [6]. Il suffit de souder un câble FTDI à la platine terminée et de monter le tout dans un capot *ad hoc*.

Le schéma (**fig. 2**) ne présente guère de particularités. Le microcontrôleur est alimenté par le câble USB. La diode D1 relie la ligne RTS du câble USB-série à la broche *Reset* du microcontrôleur. C'est ce qui permet de commander une remise à zéro matérielle à travers le port USB. Les lignes TxD et RxD du câble USB-série permettent la communication entre le PC et le microcontrôleur.

Caractéristiques

- 24 entrée-sorties tout ou rien ou jusqu'à
- 8 entrées analogiques (résolution 10 bits)
- 8 sorties MLI (résolution 10 bits)
- 8 signaux de servo RC (résolution 10 bits)
- 4 entrées de compteurs (16bits croissant-décroissant)
- 2 CI MAX7219 pour piloter jusqu'à 128 LED

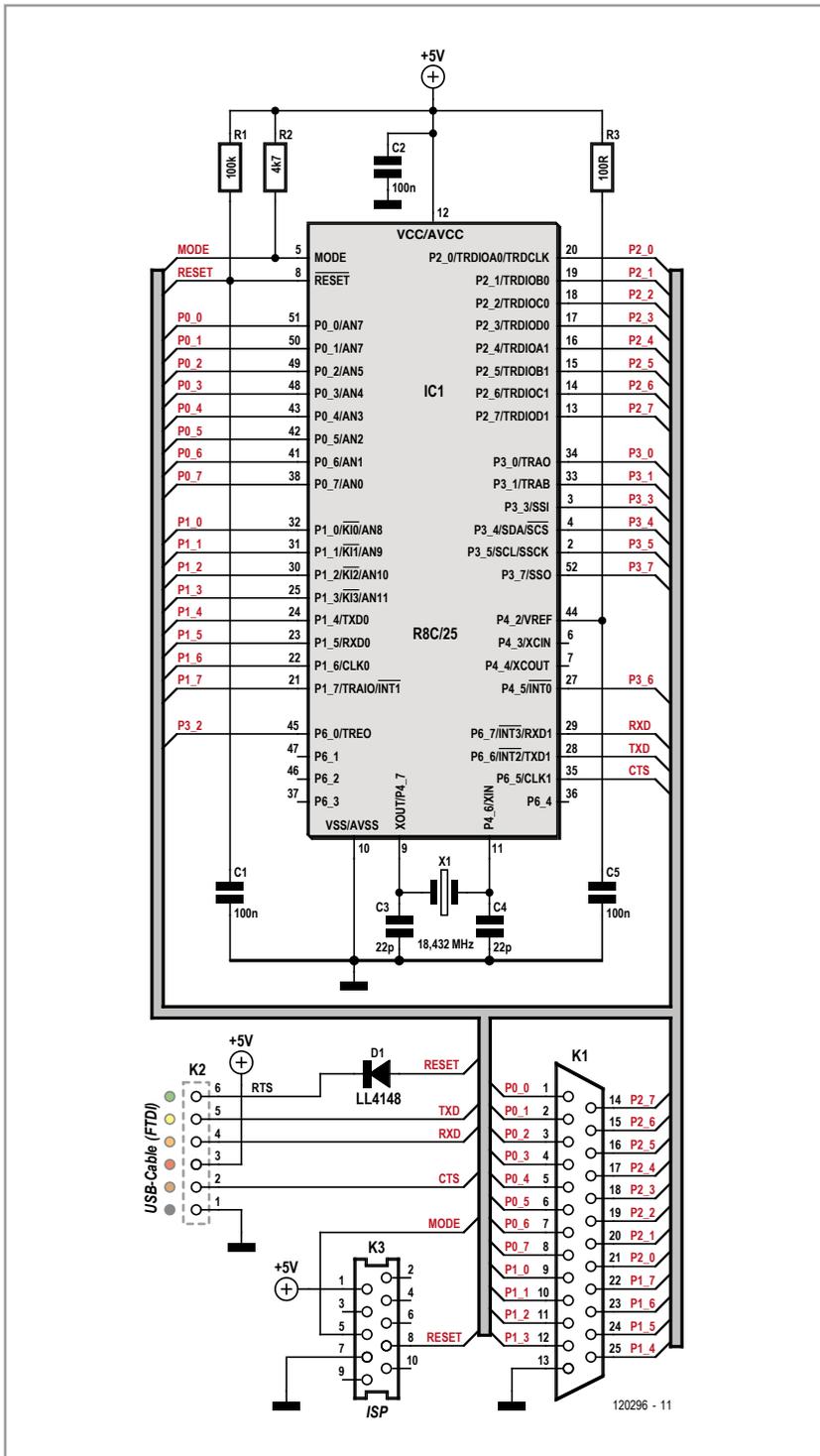


Figure 2. Il ne faut rien de plus que le microcontrôleur, un quartz et quelques composants discrets.

Les ports P0 à P2 du microcontrôleur sont amenés à l'embase SubD (figure 3). Ainsi, tous les signaux essentiels du microcontrôleur sont disponibles. Les broches du port 0 peuvent être configurées en entrées analogiques, ce qui permet l'acquisition

de huit signaux analogiques. L'entrée de la tension de référence du convertisseur A/N est reliée, suivant le procédé standard, à l'alimentation 5 V par un réseau RC. La plage de mesure s'étend de 0 à 5 V, avec une exactitude suffisante. Pour les mesures qui exigeraient une plus grande exactitude, on peut installer une référence de tension de type LM4040.

Le port 1 offre quatre entrées d'interruption qui permettent par exemple de réaliser des compteurs rapides. Accessoirement, le port 1 comporte une interface sérielle exploitable en mode UART ou SPI.

Le port 2 comporte les sorties des deux compteurs à 16 bits du R8C25, ce qui permet de produire jusqu'à huit signaux MLI simultanés. Un signal MLI peut servir à commander la vitesse de rotation d'un moteur à courant continu ou à synthétiser des signaux analogiques. Ainsi, le câble USB-IO24 offre une foule de possibilités. Qu'en est-il du côté logiciel ?

Micrologiciel

L'auteur a créé pour le contrôleur un micrologiciel qui permet un pilotage très simple à partir du PC (le microcontrôleur de la platine câblée est programmé avec ce logiciel). Le transfert des données entre le PC et le microcontrôleur s'effectue avec un débit de 115200 bauds, 8 bits de données, 1 stop, sans parité (115200, 8, N, 1). Au lieu d'un protocole de données binaires dans une trame fixe (avec par exemple forme et longueur des données plus somme de contrôle), la communication met en œuvre de simples commandes en texte. Ce procédé est souple et facile à comprendre ; de plus, la communication est très transparente. L'émission des commandes ne demande rien de plus qu'un programme de terminal ordinaire, ce qui rend l'ensemble indépendant du système d'exploitation utilisé.

Après la remise à zéro, le microcontrôleur lance l'initialisation. Il n'y a pas grand-chose à faire, parce que le mode de fonctionnement des ports et compteurs n'est fixé que par les commandes. Seules l'horloge et naturellement l'interface sérielle de communication avec le PC doivent être initialisées.

En dehors de cela, le programme principal ne comporte qu'une boucle sans fin, l'interpréteur de commandes (listing 1). Une commande consiste

en une séquence de caractères ASCII terminée par un « retour chariot » (*Carriage Return, CR*). Au sens strict, plusieurs commandes peuvent aussi se succéder avec seulement la dernière terminée par un <CR>. Cela apporte de la souplesse à l'utilisation. La suite de caractères X10 P0.0=1 P0.0=0 <CR> est une commande valide qui signifie : mettre dix fois de suite la broche P0.0 à 1 et retour.

La fonction `ReadCmd()` lit une séquence de commandes jusqu'au <CR>. La fonction `SkipBlanks()` supprime tous les espaces, après quoi la fonction `Execute()` est appelée. Si la fonction a pu exécuter toutes les commandes, elle renvoie une valeur positive, dans le cas contraire un zéro. Suivant le cas, le message OK ou ERROR est renvoyé au PC. L'exécution d'une séquence de commandes peut prendre un certain temps. Pendant ce temps, d'autres caractères doivent pouvoir être envoyés par le PC. Avec le débit de 115200 bauds, le transfert d'un caractère ne dure que 90 µs ! C'est pourquoi la réception est assu-

Listing 1 – Routine principale avec l'interpréteur de commandes.

```
// --- Main ---
VOID main()
{
    InitCrystal();           // init crystal
    InitUart();             // init uart
    EI;                     // ints

    SendString("\xCPIOCable V1.5\r\nOK\r\n");// version

    for (;;)                // endless..
    {
        ReadCmd();          // read command
        SkipBlanks();       // remove spaces

        if (Execute())      // execute commands
            SendString("\r\nOK\r\n");
        else
            SendString("\r\nERROR\r\n");
    }
}
```

rée par une routine d'interruption qui stocke les caractères dans un tampon circulaire. Ensuite, c'est dans le tampon que la fonction `ReadCmd()` va lire les caractères.

Listing 2 – Interprétation des commandes de port.

```
// --- Macros ---
#define Digit(p) (*p++ - '0') // get a digit
#define GetIf(p,c) (*p == c ? p++, 1 : 0) // get if char

// --- Read port ---
BYTE GetPort(BYTE nPort)
{
    switch (nPort)
    {
        case 0: return p0; // p0 input
        case 1: return p1; // p1 input
        case 2: return p2; // p2 input
    }
}

// --- Write port ---
VOID SetPort(BYTE nPort, BYTE nBits, BYTE nMask)
{
    BYTE nSet = nBits & nMask;

    switch (nPort)
    {
        case 0: p0 = p0 & ~nMask | nSet; break; // p0 output
        case 1: p1 = p1 & ~nMask | nSet; break; // p1 output
        case 2: p2 = p2 & ~nMask | nSet; break; // p2 output
    }
}

// --- Port command [Pn|Pn=v|Pn.b|Pn.b=v|Pn~|Pn.b~] ---
BOOL PortCmd()
{
    BYTE nPort = Digit(pCmd); // port
    BYTE nMask = 0xFF;
    BYTE nPin = 0;

    if (GetIf(pCmd, '.')) // '.'
    {
        nPin = Digit(pCmd); // pin
        nMask = 1 << nPin; // mask
    }

    if (nPort < 3 && nPin < 8) // check
    {
        BYTE nValue;

        if (GetIf(pCmd, '=')) // '='
        {
            nValue = GetValue() << nPin; // value
            SetPort(nPort, nValue, nMask); // set port
        }
        else if (GetIf(pCmd, '~')) // '~'
        {
            nValue = GetPort(nPort); // get port
            SetPort(nPort, ~nValue, nMask); // set invert
        }
        else
        {
            nValue = GetPort(nPort) & nMask; // get port
            nValue = nValue >> nPin; // get value
            SendValue(nValue); // send value
        }
    }

    return TRUE;
}

return FALSE;
}
```

| Tableau 1. Résumé des commandes reconnues | | | |
|---|--|--|---|
| A à | An | $n = 0 \text{ à } 7$ | La broche An est configurée en entrée analogique et la valeur numérisée (plage de 0 à 1023) est renvoyée. |
| C à | $Cn, Cn = x$ $Cn+, Cn-$ | $n = 0 \text{ à } 3$ $x = 0 \text{ à } 65535$ | La valeur actuelle du compteur CNTn est renvoyée ; peut être fixée à une valeur comprise entre 0 et 65535. Le sens du comptage peut être fixé avec + ou -. |
| P à | $Pn, Pn = x$ $Pn.b, Pn.b = y$ $Pn\sim, Pn.b\sim$ | $n = 0 \text{ à } 2$ $b = 0 \text{ à } 7$ $x = 0 \text{ à } 255$ $y = 0 \text{ à } 1$ | Lecture ou écriture d'un port au format de 8 bits, ou d'une broche. Inversion du port ou d'une broche avec \sim . |
| S à | $Sn, Sn = x$ | $n = 0 \text{ à } 7$ $x = 0 \text{ à } 1023$ | Configure une des broches MLI en sortie servo-moteur. Position neutre du servo : $x = 512$. |
| W à | $Wn, Wn = x$ | $n = 0 \text{ à } 7$ $x = 0 \text{ à } 1023$ | Active une des sorties MLI. La durée d'impulsion est comprise entre 0 et 1023. |
| D à | $Dn =$ x,x,x,x,x,x,x,x | $n = 0 \text{ à } 1$ $x = 0 \text{ à } 255$ | Attaque d'un MAX7219, pour le pilotage de 64 LED. Envoi en une fois de 8 octets de données. |
| R à | $Rn, Rn = x$ | $n = 0 \text{ à } 6$ $x = 0 \text{ à } 255$ | Lecture ou écriture des registres de commande : R0: Sens des données de P0, 0 = entrée, 1 = sortie R1: Sens des données de P1, 0 = entrée, 1 = sortie R2: Sens des données de P2, 0 = entrée, 1 = sortie R3: réservé R4: fréquence MLI, 0 = 1 kHz, 1 = 2 kHz, 2 = 4 kHz R5: retard MLI, 0 = aucun à 99 = lent R6: luminosité 1. MAX7219, 0 = sombre à 15 = clair R7: luminosité 2. MAX7219, 0 = sombre à 15 = clair |
| T à | Tn | $n = 0 \text{ à } 6500$ | Temporisation de $n * 10 \mu s$. |
| X à | Xn | $n = 0 \text{ à } 65535$ | Toutes les commandes de la ligne qui suivent le X sont répétées X fois. |
| ? | ? | | Affiche un résumé des commandes reconnues. |

Les commandes possibles se distinguent par leur premier caractère. Ainsi, la commande Port commence par un P, la lecture d'une entrée analogique par un A, le chargement d'un compteur par C etc. Majuscules et minuscules sont autorisées. Le code contient pour chaque commande une fonction qui reconnaît la syntaxe possible. Le listing 2 reprend les fonctions disponibles pour la commande Port.

Dans le cas le plus simple, la fonction lit l'état actuel d'un port, P0, P1 ou P2. Elle renvoie à l'interpréteur un nombre de 0 à 255. Pour modifier l'état d'un port, on peut lui assigner une valeur

(P0 = 0 à 255). La fonction PortCmd() distingue donc l'écriture de la lecture par la présence du signe égale. Il est possible en outre de lire et écrire individuellement les bits d'un port, en indiquant le numéro du bit après celui du port, P0.0, P0.1 à P0.7. Ici, c'est la présence ou l'absence du point qui indique si l'opération vise un bit ou le port entier. Enfin, on peut inverser l'état d'une broche de port par la commande tilde (P0.0~).

Jeu d'instructions

Les commandes de manipulation de port décrites ci-dessus permettent déjà de piloter parfaitement une carte d'entrées-sorties tout ou rien

avec des relais et des optocoupleurs. Il reste toutefois à configurer le sens des données pour chaque broche du câble USB-IO24. Après la remise à zéro, toutes les broches sont programmées par défaut en entrées. Ce fait doit être pris en considération lors de la conception d'un matériel particulier : les étages de puissance doivent se trouver dans un état sûr. La commande R permet d'écrire dans différents registres de commande. Les registres R0, R1 et R2 régissent le sens des données des trois ports. Le bit 0 de R0 fixe le sens des données de la broche P0.0, le bit 1 celui de la broche P0.1 etc. La valeur 1 signifie sortie et 0 signifie entrée. La commande $R0=15<CR>$ fait passer à « sortie » le sens des données des quatre lignes de poids faible de P0. On peut écrire aussi $R0=\$0F<CR>$ ou $R0=\%00001111<CR>$. Le caractère dollar annonce un nombre hexadécimal, le caractère pourcent un nombre binaire.

Le **tableau 1** présente toutes les commandes reconnues à ce jour par le câble USB-IO24. Il existe d'autres commandes, en plus des commandes de ports et registres déjà vues. Dans les applications avec capteurs, la commande A est particulièrement importante pour la lecture des entrées analogiques. Jusqu'à huit valeurs analogiques peuvent être numérisées. Les broches AI0 à AI7 n'ont pas besoin d'autre configuration, elles sont automatiquement programmées en entrées analogiques par la première commande A.

Tout comme il est possible en principe d'envoyer plusieurs commandes d'un coup, il est possible de lire plusieurs entrées analogiques en une commande. La séquence $A0 A1 A2<CR>$ lit les trois premières entrées analogiques et renvoie les valeurs en une réponse. Les valeurs multiples d'une réponse sont séparées par des espaces.

Le pendant d'une entrée analogique serait une sortie analogique. Le microcontrôleur ne reconnaît aucune sortie analogique directe, mais il est capable de produire jusqu'à huit valeurs analogiques par le biais de signaux MLI. Les signaux MLI sont lancés par la commande Wn (*Pulse Width Modulation*). Comme pour les entrées analogiques, aucune autre configuration n'est nécessaire ; le registre R4 permet de régler la fréquence MLI. Pour tirer une tension analogique d'un signal MLI, il faut passer par un filtre passe-bas. Dans le cas le plus simple, un réseau RC suf-

fit ; le résultat est meilleur avec un amplificateur opérationnel pour construire un filtre d'ordre plus élevé, par exemple un Butterworth.

À ce point, nous pouvons déjà réaliser de nombreuses applications standard avec des entrées et sorties tout ou rien et analogiques. Si on considère qu'il est possible en théorie de connecter jusqu'à 127 câbles USB-IO24 à un même ordinateur, il aurait $127*24 = 3048$ entrées-sorties ! Pourtant, des solutions plus spécifiques sont parfois nécessaires pour satisfaire des exigences telles qu'en rencontre souvent un électronicien amateur. C'est précisément pour cela que d'autres commandes puissantes ont été mises en œuvre et nous allons les expliquer maintenant à l'aide d'exemples d'application.

Pilotage de servos de radio-commande

Les servo-moteurs connus pour la radio-commande, disponibles dans tous les niveaux de puissance, sont devenus courants dans les projets de robotique et pour d'autres entraînements mécaniques. Ils ajustent leur position conformément à une consigne reçue. La consigne des servos de RC est codée par la durée d'une impulsion, comprise entre 1 et 2 ms, répétée toutes les 20 ms (**fig. 4**). Produire directement cette suite d'impulsions n'est pas une tâche qu'on peut assigner au PC. Les caractéristiques temporelles sont critiques, en particulier la durée d'impulsion. Même une gigue minime peut provoquer un ronflement audible du servo (avec une consommation accrue). C'est pourquoi la commande spéciale Sn permet d'utiliser les sorties MLI pour produire jusqu'à huit signaux de servos. Il est possible aussi de produire simultanément des signaux MLI et servo. Toutefois, du fait de la structure des compteurs du R8C25, chaque quartet de broches du port 2 doit avoir la même cadence — donc les broches P2.0 à P2.3 produisent des signaux MLI, alors que les broches P2.4 à P2.7 sont disponibles pour les impulsions de servos.

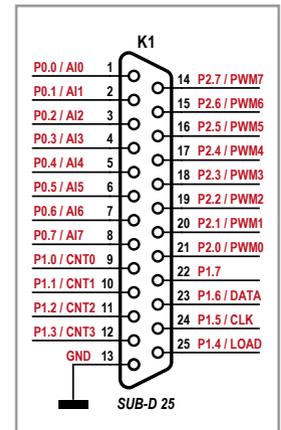
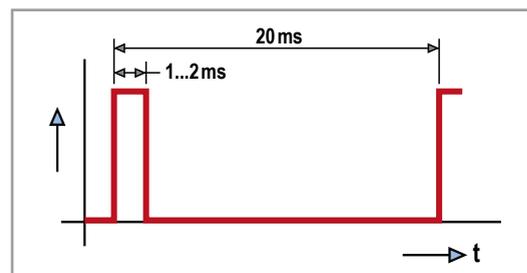


Figure 3.
Le brochage de la prise SubD-25.

Figure 4 .
Séquence de commande de servo RC.

Les servos RC récents sont caractérisés par une très grande vitesse de réponse. En principe c'est un avantage, qui permet de commander un mouvement volontairement lent par de très nombreuses positions intermédiaires. Toutefois, cela autorise une sorte de recul du servo, ce qui peut amener des vibrations considérables dans un bras robotisé. C'est pourquoi le registre R5 permet de réduire à dessein la vitesse de positionnement. Le retard est commandé directement par le générateur d'impulsions et réduit ainsi considérablement le recul du servo. La valeur de R5 s'applique simultanément au générateur

ser le sens de marche, il faut un relais ou un pont complet. La variation de régime est alors assurée par une sortie MLI. Les difficultés commencent quand on veut réguler le régime ou atteindre une position précise. Il est nécessaire pour cela d'installer sur l'arbre un générateur d'impulsions, par exemple un disque cranté qui coupe une barrière lumineuse. Il faut alors compter les impulsions (fronts) produites. L'accumulation donne la position, le rapport à un intervalle de temps donne la vitesse. L'une et l'autre sont obtenues par la commande Cn. Le câble USB-IO24 possède quatre entrées, CNT0 à CNT3, qui peuvent être reliées à l'intérieur du contrôleur chacune à un compteur à 16 bits. La commande C0+<CR> active le comptage croissant ; la commande C0-<CR> active le comptage décroissant. L'état du compteur est lu avec C0<CR>.

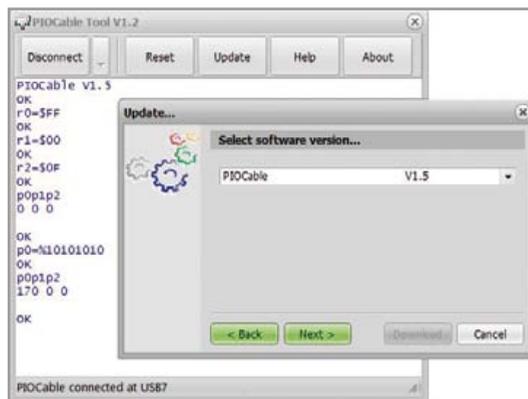


Figure 5.
Le logiciel pour PC
« IOcable-Tool ».

Le sens du comptage peut être inversé à tout moment, sans que ça modifie l'état du compteur. Il est possible ainsi de suivre avec un compteur la position actuelle d'un arbre. Il suffit d'inverser le sens du comptage avant le sens de rotation du moteur. Au contraire, pour lire la vitesse, on commence par mettre le compteur à zéro avec C0=0<CR> puis on le lit au bout d'un temps donné. La valeur du compteur est une mesure du régime du moteur.

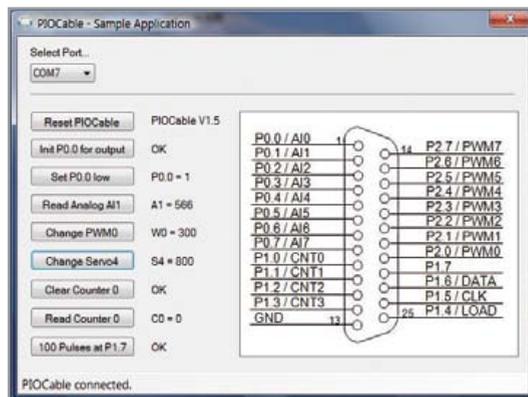


Figure 6 .
Application de
démonstration en C#.

Affichage à LED

Maxim propose, avec le MAX7219 [4], un circuit universel capable de piloter jusqu'à 64 LED. Hormis une résistance, aucun câblage externe n'est nécessaire. Elektor a déjà présenté un affichage universel à 7 segments avec cette puce [5], elle sert souvent aussi à réaliser des matrices.

Deux de ces puces (en cascade) peuvent être raccordées à notre câble USB-IO24. Il est possible ainsi de piloter jusqu'à 128 LED individuelles (ou deux afficheurs à huit chiffres, ou une combinaison des deux). Les données sont transmises en série à l'afficheur. Il suffit pour cela de relier les trois lignes DATA, CLK et LOAD aux broches de même nom du MAX7219. La commande Dn transfère alors en une fois huit octets de données — par exemple D0=48,109,121,51,91,95,38,105<CR>. Les octets sont copiés par le MAX7219 l'un après l'autre dans les registres Digit0 à Digit7 ; chaque bit correspond à une LED connectée. L'agencement exact se trouve dans la feuille de caractéristiques de la puce.

de signaux MLI.

Position angulaire ou vitesse de rotation

Un domaine étendu est celui des petits moteurs à courant continu, comme on en trouve dans beaucoup de projets. Dans le cas le plus simple, il suffit d'un transistor bipolaire, ou à effet de champ pour les courants supérieurs ; pour inver-

PIOCable-Tool

La mise en œuvre est-elle compliquée ? Où trouve-t-on les pilotes ? Faut-il d'abord écrire soi-même un programme pour vérifier le fonctionnement ? Toutes ces questions se posent chaque fois qu'on envisage de raccorder un nouveau matériel au PC. Heureusement, la mise en œuvre de ce câble USB-IO24 est très simple. Après la connexion au PC, Windows cherche un pilote adéquat. Les pilotes FTDI (pour les lecteurs qui disposent d'une connexion Internet) s'installent en général automatiquement. En cas de difficultés, on trouvera sur le site FTDI des directives pour l'installation des pilotes VCP (*Virtual Com Port*).

Câble connecté, pilotes installés, on lance le programme PIOCable-Tool, destiné spécialement à la mise en service et au test de toutes les fonctions (**fig. 5**). Il est téléchargeable de la page ouëbe de cet article [6], où on trouvera aussi le code source. Le programme a été écrit avec Delphi XE2, mais le code source devrait en principe se compiler aussi avec des versions antérieures de Delphi.

Au démarrage, après action sur le bouton « Connect », la recherche du câble démarre. En cas d'échec, il faut aller chercher dans « Gestion des périphériques » le numéro de port COM qui a été affecté au câble, puis le reporter manuellement dans PIOCable-Tool. Il s'agit essentiellement d'un programme de terminal. Toutes les saisies sont transmises directement au microcontrôleur du câble USB-IO24, lequel répond directement. Ainsi peut-on envoyer toutes les commandes manuellement et observer les effets sur les broches, par exemple avec un oscilloscope. Le développement du micrologiciel du câble continue, c'est pourquoi le programme PIOCable-Tool permet de transférer la nouvelle version dans le microcontrôleur. Ce dernier est livré sur la platine câblée avec un programme chargeur (*Bootloader*) installé. Les versions à venir se trouveront également sur le site Elektor [6] et sur le site Tiny-Basic de l'auteur [7].

Le chargeur permet naturellement d'exécuter un micrologiciel maison. On trouvera sur le site de Renesas [8] l'environnement de développement « *High-performance Embedded Workshop* » avec le compilateur C adéquat. Le microcontrôleur peut aussi se programmer sans chargeur, avec un programmeur adapté. Il faut alors fabriquer

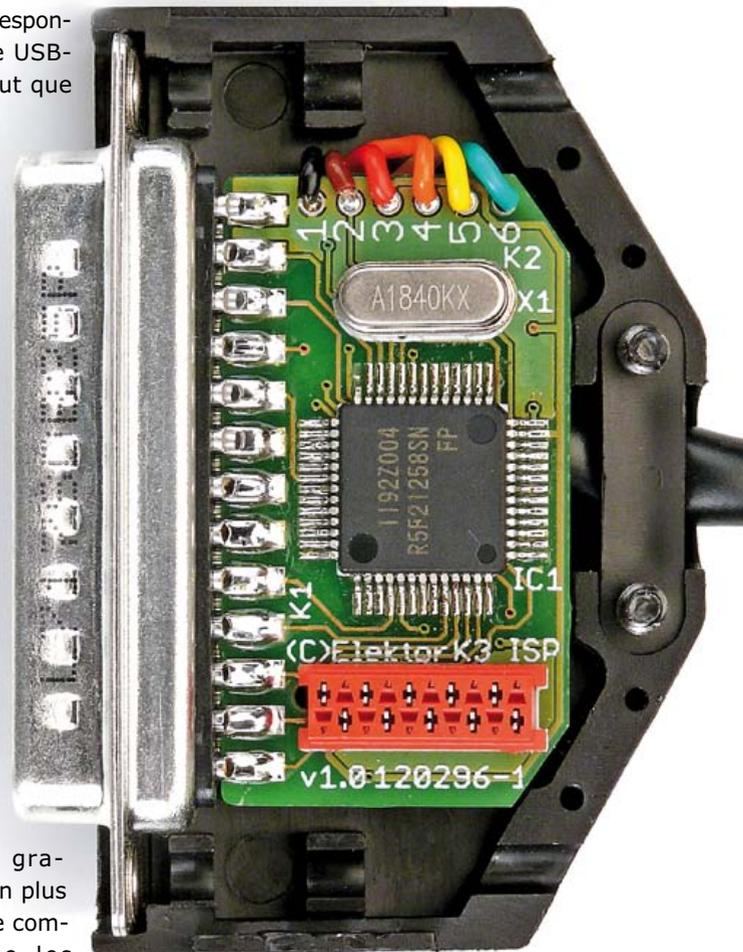
un petit câble adaptateur avec un connecteur MicroMatch (voir liste des composants, Option). Naturellement, le chargeur (comme le micrologiciel original) est écrasé par la programmation directe.

Application de démonstration

Pour les applications simples (comme le test de servos), il suffit de saisir les commandes à l'aide de PIOCable-Tool. Pour toutes les autres applications, il est logique de développer un logiciel sur PC qui enverra les commandes correspondantes au câble USB-IO24. On ne peut que recommander aux programmeurs en Delphi d'exploiter le code source de PIOCable-Tool.

Il est particulièrement simple de créer des applications maison avec le langage C# dans l'environnement de développement Microsoft .NET (il existe une version *Express* gratuite et une version Linux entièrement compatible et gratuite : *Mono*). En plus des éléments de commande comme les menus, boutons ou étiquettes, .NET propose des composants d'interface série qui permettent de communiquer très simplement avec notre câble.

On traîne d'abord un composant de type *Serial-Port* sur un formulaire. Chaque composant doit avoir un nom, par exemple « *PIOPort* ». Tous les autres réglages se font dans le texte source. Ensuite on traîne un bouton sur le formulaire et



on écrit dans sa routine « OnClick » les lignes suivantes :

```
PIOPort.ReadTimeout = 100;  
PIOPort.BaudRate = 115000;  
PIOPort.NewLine = "\r\n";  
PIOPort.PortName = "COM5";  
PIOPort.Open();
```

La valeur de PortName sera le numéro de port effectivement attribué au port du câble. Si tout s'est bien passé, la connexion avec le matériel est établie.

L'auteur a écrit en C# une application de démonstration [6] dont une capture d'écran est visible sur la **figure 6**. Les lignes ci-dessus sont incluses dans le code quand l'utilisateur sélectionne dans la boîte de dialogue une des interfaces disponibles.

La déclaration `PIOPort.NewLine = "\r\n"` signale au composant `PIOPort` que les prochains appels à `PIOPort.WriteLine()` et `PIOPort.ReadLine()` doivent se terminer par les caractères <CR><LF>. Il est alors inutile de les ajouter lors de l'envoi d'une commande ; la lecture des réponses s'effectue jusqu'à la fin de ligne. Cela nous permet par exemple d'écrire une fonction simple qui mettra à 1 la broche P0.0. On tire un nouveau bouton sur le formulaire et on écrit la ligne suivante dans la routine « OnClick » :

```
PIOPort.WriteLine("r0=1p0.0=1");
```

Cette commande configure d'abord la broche en sortie et la passe aussitôt au niveau haut. Il est plus compliqué de lire une valeur analogique.

Il faut d'abord attendre la réponse et ensuite la lire avec `PIOPort.ReadLine()`. On trouvera le détail dans le code de l'application de démonstration.

Plus rien ne s'oppose à l'écriture de logiciels maison de mesure et d'automatisme.

(120296 – version française : Jean-Paul Brodier)

Liens

- [1] www.ftdichip.com/Products/Cables/USBTTLSerial.htm
- [2] www.elektor.fr/080213
- [3] www.renesas.com/products/mpumcu/r8c/r8c2x/r8c25/index.jsp
- [4] www.maximintegrated.com/datasheet/index.mvp/id/1339
- [5] www.elektor.fr/081154
- [6] www.elektor.fr/120296
- [7] www.tinybasic.de
- [8] www.renesas.eu/products/tools/ide/ide_hew/index.jsp

Publicité



Les microcontrôleurs PIC® USB 8 bits sans quartz diminuent les coûts système et la consommation

Une horloge interne dotée d'une précision de 0,25 % permet la connectivité USB tout en se passant de quartz externe



Les microcontrôleurs USB les plus petits et les plus économiques de Microchip sont dotés de 14 broches et existent jusqu'à 100 broches et sont les premiers microcontrôleurs 8 bits à intégrer sur une seule et unique puce le pilotage d'écrans LCD, un module horloge calendrier temps-réel avec entrée batterie de secours et l'USB.

Les tout nouveaux microcontrôleurs PIC® USB de Microchip intègrent des sources d'horloge internes d'une précision de 0,25 % permettant la connectivité USB sans aucun quartz externe. Ce sont également les premiers microcontrôleurs USB à associer dans un brochage compris entre 14 et 100 broches, un degré d'intégration élevé des périphériques, et jusqu'à 128 ko de mémoire Flash. La technologie XLP très faible consommation permet également de maintenir la consommation à 35 µA/MHz en mode actif et à 20 nA en mode veille.

Les plus économiques et les plus petits

Les microcontrôleurs PIC16F145X permettent d'ajouter la connectivité USB et le tactile capacitif en plus d'un large éventail de périphériques intégrés, et ce pour un encombrement minimum de 4 x 4 mm.

Un tactile capacitif performant doté de l'USB

Dotés d'une unité périphérique CTMU de mesure de temps de charge (Charge Time Measurement Unit) et d'une tension de fonctionnement comprise entre 1,8 V et 5 V, les microcontrôleurs PIC18F2X/4XK50 sont compatibles en termes de brochage avec les PIC18 plus anciens, ce qui facilite la migration vers de meilleures performances.

USB, contrôleur d'écrans LCD et Horloge calendrier temps-téel avec Vbat

La famille des PIC18F97J94 permet la connectivité USB, le pilotage d'écrans LCD et intègre un module RTCC (horloge calendrier temps réel) avec batterie de secours, le tout dans un seul microcontrôleur PIC® 8 bits.

Pour de plus amples informations, visitez la page : www.microchip.com/get/eu8bitUSB

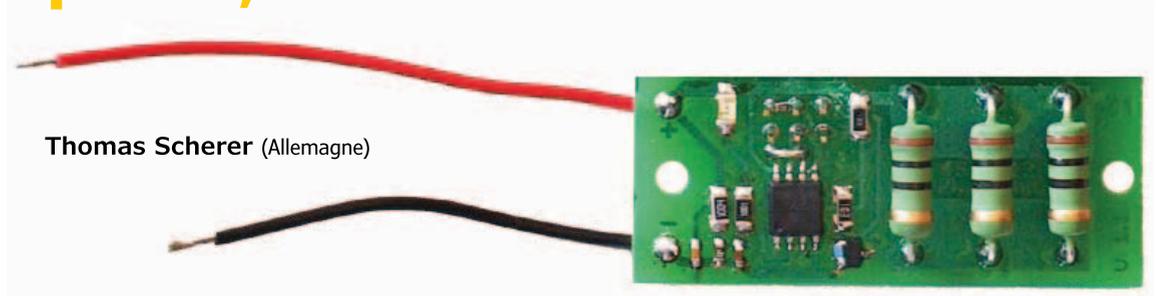
DÉMARREZ VOTRE DÉVELOPPEMENT EN 3 PETITES ÉTAPES :

1. Choisissez le jeu de périphériques et le brochage qui conviennent à votre application en utilisant le logiciel MAPS
2. Utilisez les piles de protocole USB et les pilotes logiciels gratuits pour un développement plus rapide
3. Commencez votre développement grâce à nos kits de développement économiques



Microcontrollers • Digital Signal Controllers • Analog • Memory • Wireless

petit, mais coriace !



...nom d'un chien, qu'ai-je donc fait à ce diantre de circuit pour qu'il ne se laisse-il pas flasher par ISP sur mon programmeur AVRISP MKII ?

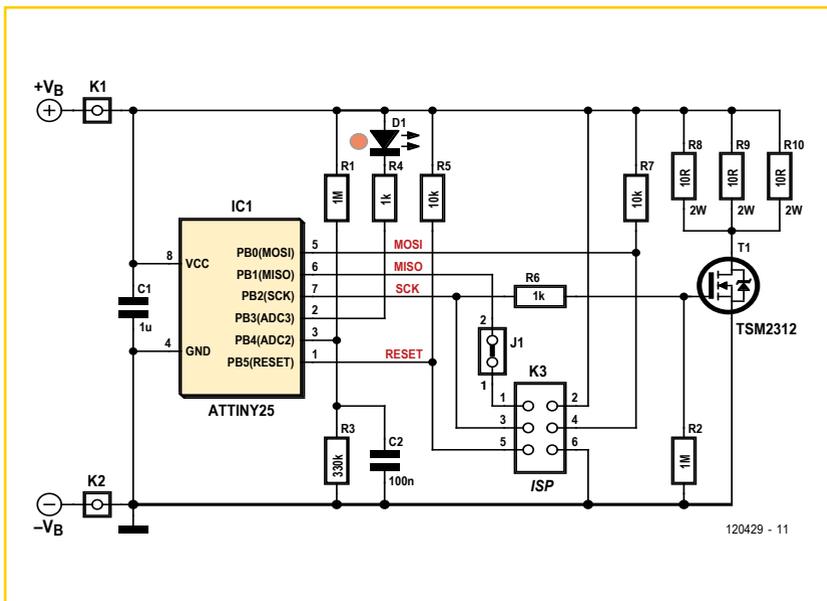
Je me suis creusé la cervelle, j'ai tout essayé, abandonné, réessayé... avant que la lumière jaillisse. Pour résumer : je voulais construire un compensateur simple, précis et souple à souhait pour mes nouvelles batteries LiFePO₄ (lithium fer phosphate). Pour le faire, il existe bien des circuits tout faits ou de dispendieuses puces, mais je tiens mordicus à résoudre ces questions avec de l'électronique simple, un petit contrôleur AVR et un chouia de code. Ce minuscule ATTiny25 à huit pattes, que j'ai déjà mis à toutes les sauces possibles et inimaginables, jamais je n'aurais cru qu'il puisse me contrarier à ce point. *Pourtant, je savais, comme vous, que chaque solution apporte de nouveaux problèmes.*

Je m'escrime à remesurer les pistes et à vérifier l'implantation, puis je me dis que le fabricant

n'a peut-être pas placé les fusibles SPIen (validation SPI), ce qui empêche le flashage par ISP/SPI. Comme il s'agit ici de la version CMS, je ne peux pas simplement programmer le contrôleur sous « haute tension » avec ma carte STK500. Je soude alors de fins câbles à même le circuit imprimé pour le relier aux broches HV du STK500, mais à mon grand étonnement, ça ne marche pas. Pourtant, la programmation sous HT doit toujours fonctionner. Aussi, je soude de petits bouts de fils sur un Tiny25 en CMS, de manière telle qu'il s'adapte à l'un des supports DIL du STK500. Et là, ça fonctionne ; en fait, les SPIen étaient correctement positionnés. Conclusion : la panne, c'est dans *mon* circuit qu'elle se trouve. Mais où ?

Après mûre réflexion, mes soupçons se portent sur la résistance R4 de 1 kΩ, en série avec la LED D1, qui surchargerait l'un ou l'autre des signaux lors de la programmation. Tout faux, encore une fois : la broche 2 (portB.3) n'est absolument pas reliée à K3. C'est alors qu'une idée me vient : même si, physiquement, T1 est très petit, comme MOSFET, il dispose d'une grande faculté de commutation et devrait présenter une fameuse capacité d'entrée. Hypothèse confirmée par sa feuille de caractéristiques : C_{iss} = 900 pF. Ah, serait-ce lui le coupable ? Eh bien, je monte à 10 kΩ la résistance de grille R6, je relie à K3 l'AVRISP et sous AVR Studio, je clique sur *Program*. Tout rentre dans l'ordre pour le petit microcontrôleur.

Si vous aviez compris plus vite que moi, rien qu'à l'examen du schéma, je vous tire mon chapeau !
(120429 – version française : Robert Grignard)



[1] www.elektor.fr/magazines/2011/janvier/i555.1648145.lynxk

[2] www.elektor.fr/magazines/2008/janvier/interrupteur-anti-veille.321959.lynxk

tout passe tout lasse, pas la masse

Thijs Beckers (Elektor)

Ma rubrique de bruits de labo m'impose parfois le détestable rôle de *sycophante*. Dans la Grèce antique, le sycophante est un dénonciateur professionnel qui assignait en justice des citoyens riches afin d'obtenir une part de leurs biens s'ils étaient condamnés. Pourtant je n'ai rien d'un vilain mouchard. Nous sommes ici pour apprendre et nos erreurs sont riches d'enseignements. Et comme nous ne vivons pas assez longtemps pour les faire toutes nous-mêmes, profitons de celles des autres pour apprendre davantage.

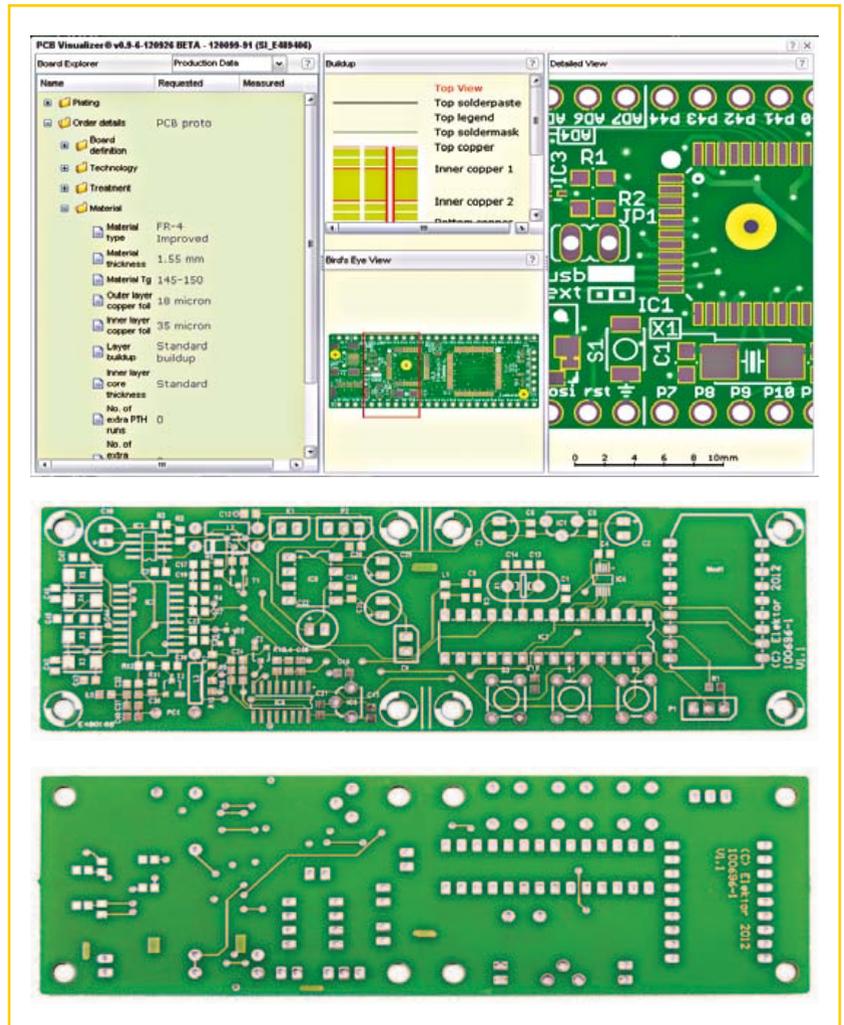
Quiconque doit mener à bien la conception d'un scanner pour la bande aviation devra naviguer parmi les écueils, certains récurrents, comme la

vérifie ton circuit avec PCB Visualizer!

disponibilité des composants, d'autres inédits, comme celui dont je parlerai ici. Car ce sont des histoires vraies que je raconte ici.

Pendant qu'il travaillait sur son PCB avec notre logiciel de CAO, un de mes collègues qui m'a donné son accord pour que je le nomme ici – c'est Luc, avait masqué temporairement les plans de masse afin d'y voir plus clair. Une fois la conception terminée, il lance la production des fichiers Gerber à l'aide du fichier .outjob puis les envoie à Eurocircuits [1] notre fournisseur de prototypes. Aucune erreur n'a été signalée dans ce processus, mais quand les PCB sont arrivés par la poste, il manquait les plans de masse... « *masqués temporairement* » disais-je. Comme la plupart des connexions de masse passaient par ces fameux plans de masse, le PCB était totalement inutilisable.

C'est ainsi que notre concepteur apprend à ses dépens que tout ce qui est masqué est exclus des fichiers Gerber. Ce couac aurait pu être évité avec un visualiseur Gerber (et moins de stress !). Cette mésaventure a ceci de bon qu'elle a attiré notre attention sur le nouveau service *PCB Visualizer* d'Eurocircuits (cf capture d'écran) sans doute meilleur que le visualiseur Gerber utilisé au labo. Leur utilitaire en ligne, bien pratique analyse les fichiers que vous lui donnez (*Extended Gerber* ou



CadSoft Eagle V6) et donne une prévisualisation de votre futur PCB. Avec ça, toute erreur lors de la conception ou de la commande pourra être tuée dans l'œuf, notamment grâce au double affichage : aperçu détaillé des paramètres à gauche et future carte en détail à droite.

Une fois que vous confirmez que tout est conforme à vos attentes, vous validez la commande, sûr désormais que la carte que vous recevrez correspondra exactement à vos spécifications, sans qu'il y manque ni piste, ni plans de masse, ni quoi que ce soit d'autre.

(120568 – version française : Kévin PETIT)

Liens

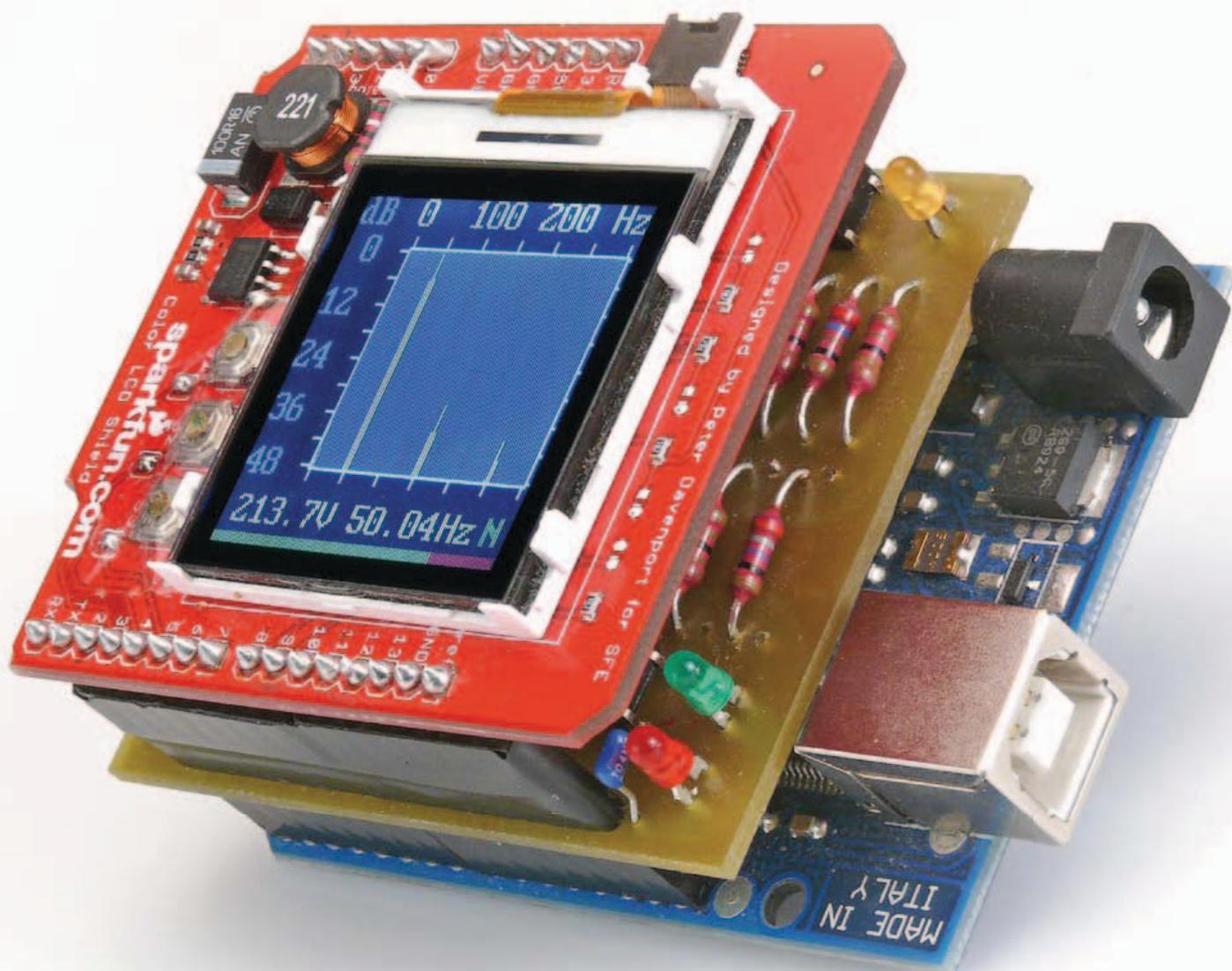
[1] www.eurocircuits.com

Arduino ausculte la tension secteur

une analyse de fréquence en couleurs

Fidelis Theinert
(Delft, Pays-Bas)

Qui penserait à réaliser un analyseur de spectre sur une modeste carte Arduino ? Le tout est d'adapter le logiciel à la puissance de calcul disponible. On ajoute à une Arduino standard un *shield* de filtre et un autre comme écran pour obtenir un analyseur du spectre de la tension du réseau de distribution électrique. Une analyse par transformée de Fourier rapide (TFR) exhibe alors les impuretés qui émanent de la prise de courant, à la maison, au bureau ou à l'atelier.



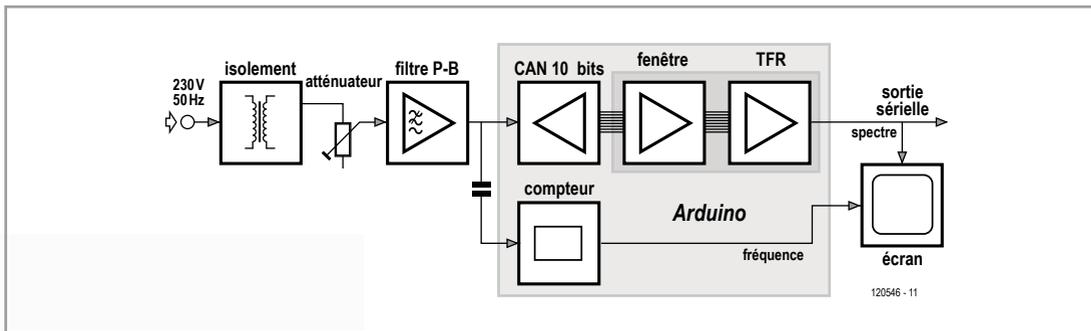


Figure 1.
Diagramme fonctionnel de
l'analyseur de spectre.

Il y a plus qu'on ne croit sur une simple carte Arduino. Elle est puissante et multifonctionnelle, avec son petit contrôleur à 8 bits, mais il faut aussi en connaître les limites. Nous vous montrerons comment tirer le meilleur parti des composantes de ce microcontrôleur, tout en indiquant les outils de développement gratuits nécessaires à la réalisation de ce projet. Le logiciel comme le matériel en sont modulaires, ce qui laisse toute liberté de les réutiliser séparément sur d'autres projets. L'application a également réussi les épreuves sur Arduino Duemilennova, UNO R1, R2 et R3.

Le circuit analyse la tension du secteur à l'aide d'une transformée de Fourier rapide (TFR, *FFT* en anglais) et affiche sur écran les harmoniques présentes, de quoi juger par vous-même de la qualité du signal énergétique livré. Les harmoniques supérieures occasionnent dans les machines et les moteurs des pertes qui se transforment en chaleur, mais que l'utilisateur doit quand même payer. En outre, elles influencent négativement le rendement des alimentations à découpage, dans les PC comme dans les autres appareils, à cause des déformations de la sinusoïde escomptée.

L'analyse de Fourier

La transformée de Fourier rapide est un algorithme mathématique qui convertit un signal discret du domaine temporel vers le domaine de fréquence. Le gain de temps de la TFR par rapport à la transformée de Fourier discrète (TFD, *DFT* en anglais) peut devenir considérable à mesure que le nombre N d'échantillons augmente. Dans la TFR, une transformée de longueur N est divisée en deux transformées distinctes de longueur $N/2$. On réduit ainsi le nombre d'opérations de calcul nécessaires, mais il faut pour cela que le nombre d'échantillons N soit une puissance de deux (64, 128, 256... 4 096...). En pratique, cela

n'est guère contraignant, puisqu'en général, on peut choisir librement la longueur N de la rangée utilisée.

Selon le théorème d'échantillonnage de Nyquist-Shannon, la largeur de bande du signal à explorer ne peut pas dépasser la moitié de la fréquence d'échantillonnage. Il faut donc que cette fréquence soit au moins le double de la fréquence la plus haute à observer. Choisir une fréquence d'échantillonnage de 800 Hz permet donc d'analyser un signal d'une bande passante de 400 Hz, ce qui permet de voir le spectre de la tension secteur jusqu'à la septième harmonique ($50 \text{ Hz} \times 7 = 350 \text{ Hz}$).

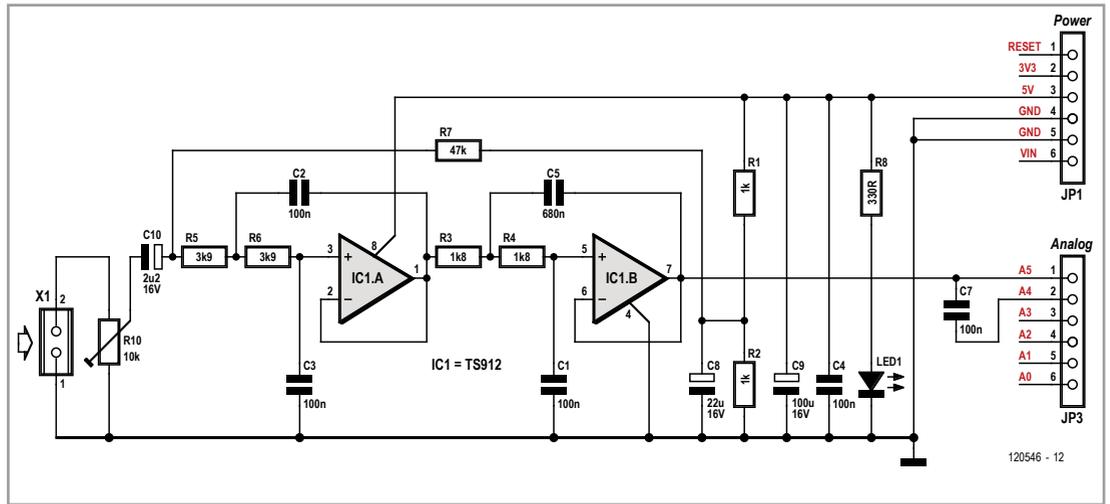
Le matériel

Le matériel se compose de trois circuits imprimés : la carte Arduino [1] et deux *shields* (cartes d'extension) que j'ai développés spécialement, un pour le filtre et un pour l'afficheur LCD [2]. Les *shields* s'empilent sur la carte Arduino, l'écran au-dessus du lot comme on s'en doute, pour former un module compact et solide, sans fils ballants. La **figure 1** vous montre le schéma de l'ensemble du projet.

Il est également possible d'envoyer par port sériel le spectre calculé par l'Arduino vers un PC. Un programme de terminal comme *Putty* en trace alors le graphique. Cette présentation des données coûte bien davantage de temps, parce que le transfert vers l'UART du PC ne s'effectue qu'à 57 600 bauds, mais la solution satisfera à tout le moins ceux qui n'ont pas la *shield* d'écran.

On s'assure qu'il n'y a pas de composante spectrale supérieure à 400 Hz au moyen d'un filtre passe-bas Sallen-Key du quatrième ordre (ou un filtre antirepliement) devant l'entrée. Il a une bande passante de 350 Hz (**figure 2**). Les deux sections de Butterworth sont construites

Figure 2.
Le filtre antirepliement, un passe-bas de Butterworth accordé à 350 Hz.



autour d'un amplificateur opérationnel double TS912. Cet amplificateur dispose d'une sortie à excursion totale (*rail to rail*) et d'une large gamme de tensions d'alimentation entre 2,7 et 16 V. On peut donc tout simplement l'alimenter

sur le 5 V de la carte Arduino.

Le signal de sortie du filtre est acheminé directement de la broche 7 de IC1 au canal 5 du convertisseur A/N (CAN) du microcontrôleur, un ATmega328P. En revanche, c'est par condensateur

Des trucs pour la TFR

L'algorithme *Butterfly* de Cooley-Turkey implémenté ici témoigne de ce qu'on appelle une transformation sur place, c'est-à-dire dans laquelle le résultat après calcul se retrouve dans le tampon même où l'on avait écrit les échantillons du signal d'entrée. Ils ont donc disparu. L'avantage, c'est qu'il faut moitié moins de mémoire et ce n'est pas négligeable quand on sait que le microcontrôleur choisi, l'ATmega328, ne possède que 2 048 octets de mémoire de travail.

Nous l'avons expliqué, le nombre d'échantillons doit être une puissance de deux et précisément, le nombre $256 = 2^8$ satisfait à ce critère. Le CAN ayant une résolution de 1 024 pas, il nous faut 2 octets de mémoire par échantillon, soit 512 octets pour toute la série. En outre, nous avons besoin pour le calcul d'un tampon pour la partie imaginaire des nombres, puisque la transformée de Fourier est une conversion en nombres complexes. Ce tampon est exactement aussi grand que pour la partie réelle et occupe donc aussi 512 octets. La TFR nécessite ainsi 1 024 octets au total rien que pour stocker les valeurs. Or, c'est la moitié de la mémoire disponible.

Nous devons aussi faire appel à la bibliothèque de mathématique du compilateur GCC, parce qu'avec la TFR, les opérations simples de comptage et de multiplication ne suffisent pas, il nous faut effectuer des opérations sur les sinus, cosinus, racines et puissances. Il serait alors pratique pour écrire les échantillons d'utiliser, non pas des

variables entières, mais des variables à virgule flottante en simple précision. Comme une variable *single precision floating point* couvre quatre octets, il en aurait fallu 2 048 pour elles seules, soit la grandeur totale de la mémoire du microcontrôleur, trop pour la plateforme choisie, donc. Une solution consisterait bien à ramener à 128 échantillons la grandeur de la TFR, mais on y perdrait en résolution spectrale, ce qui n'est pas souhaitable. Pour conserver nos 256 échantillons, nous mémorisons la partie réelle et la partie imaginaire des nombres complexes comme valeurs entières à 2 octets par échantillon, mais avant chaque opération mathématique, elles sont converties en variables à virgule flottante, puis après l'opération, reconverties en nombres entières. Le surplus de temps de calcul pour ces deux conversions crée donc un retard, mais il est acceptable. L'erreur d'arrondi introduite par la conversion est de $1/65\,536$, ce qui est un facteur 64 fois plus petit que l'erreur de quantification du CAN ($1/1\,024$) et donc négligeable. Voyez à ce sujet les fonctions `FFT_Int2Float()` et `FFT_Float2Int()` dans le module `fft_lib.c`.

Pour utiliser la mémoire de la manière la plus efficace, des échantillons issus du CAN, c'est la partie imaginaire qui est d'abord placée dans le tampon. À ce moment, ce tampon n'est pas encore pris par la TFR. Les échantillons sont ensuite normalisés et mis à l'échelle, puis stockés dans le tampon pour la partie réelle.

que la même sortie est couplée au contact A4 du connecteur d'entrée analogique de l'Arduino pour la mesure de fréquence.

La **figure 3** présente le circuit imprimé du *shield* dessiné pour le filtre. On y trouve des composants supplémentaires qui ne seront pas utilisés dans ce projet-ci. Le tracé des pistes est disponible sur la page d'Elektor dédiée à cet article [3].

C'est un transformateur à l'ancienne mode qui assure la séparation galvanique et l'adaptation de la tension du secteur, son secondaire est branché sur le domino X1. Comme le filtre entier a un gain unitaire, le signal d'une amplitude de $2 V_{eff}$ ($5 V_{pp}$) suffit à couvrir la plage dynamique du CAN de l'ATmega. Le transformateur dans l'adaptateur ne déforme pratiquement pas le signal parce qu'il n'est que faiblement chargé, comme on le constate à la **figure 5**, alors que l'alimentation pour l'analyseur provient d'un autre adaptateur ou de la prise USB. On peut donc observer avec

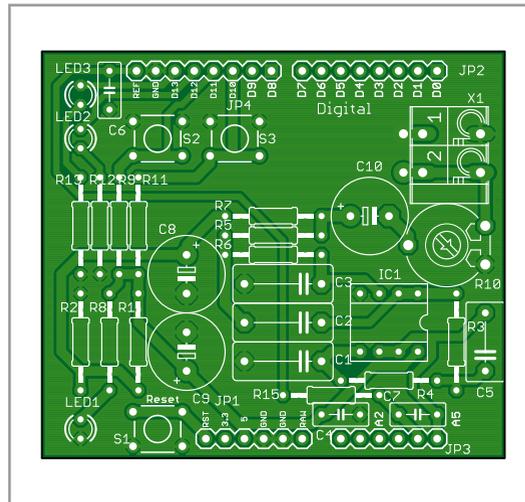


Figure 3. Dessin du circuit imprimé pour le *shield* de filtre.

grande précision toutes les harmoniques de la tension du secteur. Le potentiomètre R10 permet d'atténuer le signal de sortie à environ $1,7 V_{eff}$ sur C10 pour une tension d'entrée de 230 V. Un multimètre qui surveille la tension secteur vous

On peut alors effacer le tampon d'imaginaires et la fonction de fenêtre est utilisée sur la partie réelle des échantillons. Cette fonction de fenêtre est nécessaire pour satisfaire aux conditions de cadrage de la TFR.

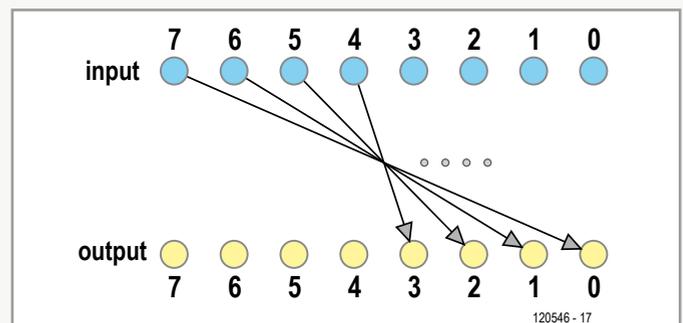
Apostille : L'algorithme de bitreversal

Une partie cruciale de l'implémentation de la TFR réside dans l'algorithme de retournement des bits. On l'emploie pour l'adressage des échantillons dans les tampons. En soi, c'est une opération très simple, il n'y a qu'à renverser l'ordre des bits du début à la fin, mais dans ce contexte, ce n'est si aisé à réaliser. Une méthode rapide consiste à utiliser des registres matériels supplémentaires. On peut aussi faire appel à une table, mais cela réclame beaucoup de mémoire. Or, sur un petit microcontrôleur, elle est très limitée. La variante la plus lente fait décaler tous les bits dans un registre pour les reprendre dans un autre registre dans l'ordre inverse. Pour cela, il faut un compteur dans une boucle, qu'il faudra surveiller, et effectuer 16 opérations de décalage. Un bon compromis entre l'utilisation de la mémoire et le temps de calcul, c'est une combinaison des deux. L'octet est scindé en deux fois quatre bits. Chaque valeur de 4 bits est convertie à part dans une table et les deux résultats sont concaténés dans l'ordre voulu. Voyez pour cela la fonction `fft_BitRev ()`

dans le module `fft_lib.c`. On gagne de la sorte $3 \mu s$ ($2 \mu s$ au lieu de $5 \mu s$) à chaque appel de la fonction, mais on le paye de 75 octets de plus dans le code.

Pour utiliser la mémoire de la manière la plus efficace, des échantillons issus du CAN, c'est la partie imaginaire qui est d'abord placée dans le tampon. À ce moment, ce tampon n'est pas encore pris par la TFR. Les échantillons sont ensuite normalisés et mis à l'échelle, puis stockés dans le tampon pour la partie réelle.

On peut alors effacer le tampon d'imaginaires et la fonction de fenêtre est utilisée sur la partie réelle des échantillons. Cette fonction de fenêtre est nécessaire pour satisfaire aux conditions de cadrage de la TFR.



L'algorithme de bitreversal

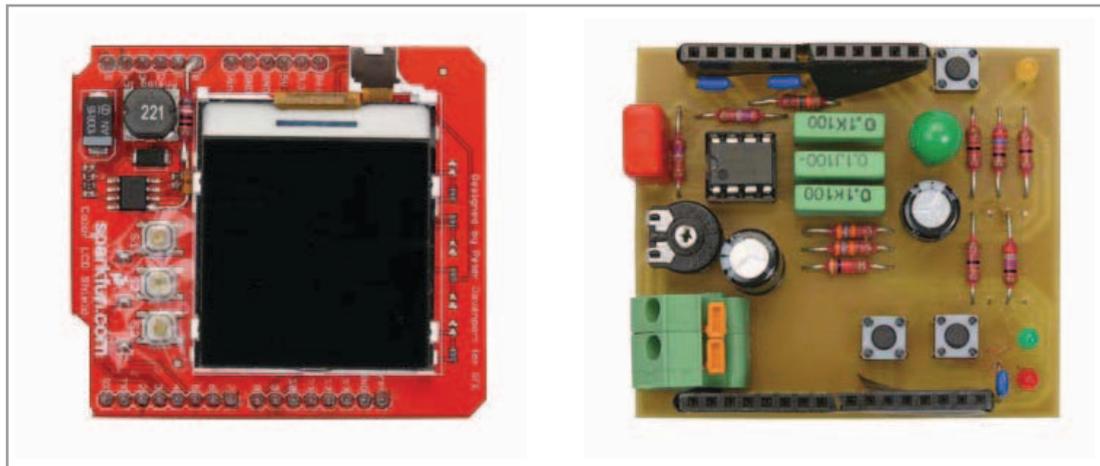


Figure 4.
La collection des trois circuits imprimés.

aidera à ajuster R10 en cas de différence actuelle par rapport à la valeur nominale. Il est recommandé d'utiliser un adaptateur secteur (à transformateur) qui délivre une tension alternative de sortie inférieure à 20 V.

L'environnement de développement

Pour développer le logiciel, je n'ai pas utilisé l'environnement standard pour Arduino, mais le compilateur croisé GCC ciblé sur la série de microcontrôleurs AVR. Le compilateur *cross-avr* peut s'installer sous Linux à partir des dépôts de la distribution avec les utilitaires binaires correspondants ; les adeptes de Windows pourront installer la suite WinAVR [4]. Les deux systèmes

d'exploitation utilisent la version 4.3.3 du compilateur GCC [5] et *avrdude* pour le transfert du fichier hexadécimal. Ce programme fait partie de la suite WinAVR pour Windows, mais il en existe une version pour Linux. Les fichiers script destinés à charger l'exécutable dans l'Arduino accompagnent les deux versions. Il faut, dans ces fichiers, indiquer le bon port sériel pour communiquer avec l'Arduino et le chemin vers l'exécutable sur le PC.

Vous aurez aussi besoin du programme *make* pour compiler et éditer les liens de tous les modules à l'aide de *makefile* ; ce programme fait aussi partie de WinAVR et est également disponible pour Linux.

Tableau 1. Modules logiciels

| Nom de fichier du code source | Nom du fichier d'en-tête | Description des fonctions |
|-------------------------------|--------------------------|---|
| avrfft.c | (aucun) | fichier principal code source |
| drw_fft.c | drw_fft.h | fonctions dessin haut niveau pour graphes sur LCD |
| fft_lib.c | fft_lib.h | définitions et fonctions pour le calcul de la TFR |
| freq_cnt.c | freq_cnt.h | définitions et fonctions pour acquisition de données et mesure de fréquence |
| lcd_driver.c | lcd_driver.h | fonctions bas niveau et définitions pour le pilotage du LCD |
| menu.c | menu.h | fonctions pour lecture des boutons du <i>shield</i> LCD |
| plt_fft.c | plt_fft.h | fonctions haut niveau pour graphes vers port sériel |
| ser_avr.c | ser_avr.h | fonctions bas niveau pour échange de données |
| std_avr.c | std_avr.h | définitions en fonctions standard courantes |
| timer_0.c | timer.h | fonctions pour temporisateurs et réglages de chronométrie système |

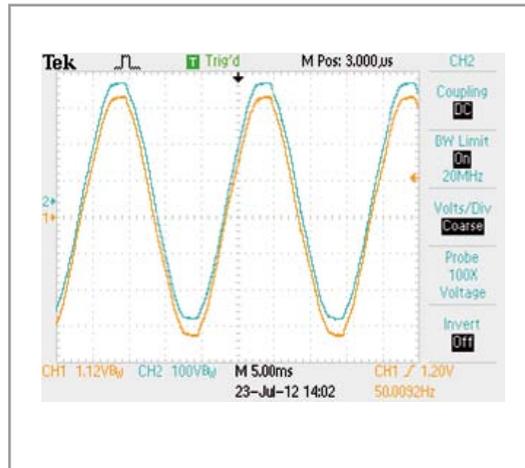
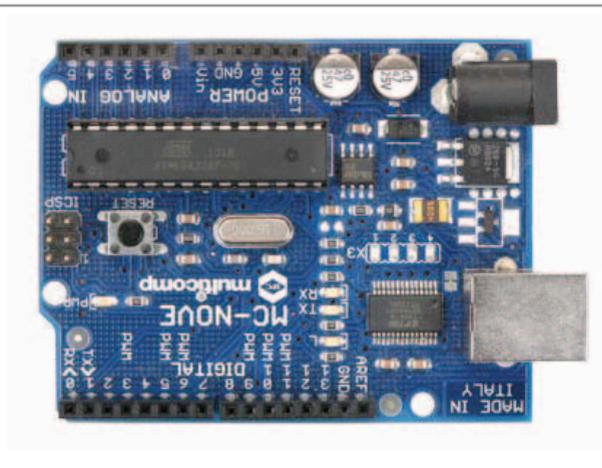


Figure 5. La tension du secteur côté primaire du transformateur en bleu et au secondaire, en jaune.

Pour modifier le code source, utilisez l'éditeur de votre choix. Eclipse avec l'extension CDT est pratique, mais Notepad++, Scite ou tout autre éditeur conviendra.

Le logiciel

La structure du logiciel est modulaire, répartie sur une série de fichiers de code source, comme l'inventorie le **tableau 1**.

Tout programme en C commence par la fonction « `main()` ». On la trouve dans le fichier `avrfft.c`. Au lancement du programme s'opère l'initialisation d'une quantité de variables globales du système et du matériel à l'aide des fonctions `init` correspondantes. Puis on efface l'écran et on prépare le temporisateur système. Vient ensuite l'exécution en boucle sans fin dans laquelle on règle ce temporisateur sur 2,5 s pour relancer un nouveau cycle à ce rythme.

Dans chaque cycle, les différentes étapes s'exécutent toujours dans le même ordre. La **figure 6** vous en propose un ordiogramme simplifié. Après l'annonce par `timID` d'un nouveau cycle, le CAN commence la lecture des 256 échantillons pour la TFR avec une fréquence d'échantillonnage de 800 Hz. On calcule ensuite la valeur efficace (TRMS) de la tension. Puis on mesure la fréquence du signal sur la broche A4 du connecteur *Analog In* à l'aide du comparateur intégré et du `Timer1` du microcontrôleur.

Les étapes suivantes, mise à l'échelle des valeurs mesurées et application de la fonction de fenêtre choisie, servent de préparation au calcul proprement dit de la TFR. Finalement, les résultats sont envoyés à l'écran du *shield* LCD pour exposer le spectre.

Affichage et utilisation

L'affichage sur le *shield* LCD utilisé présente une résolution de 131 x 131 pixels, dont l'image du spectre occupe une fenêtre de 80 pixels en vertical sur 96 horizontalement. Il en résulte un domaine entre 0 Hz et 300 Hz (96 lignes espacées de 3,125 Hz couvrent 300 Hz) et 10 pixels par 6 dB en amplitude. La TFR calcule bien le spectre entre 0 et 400 Hz, mais les fréquences supérieures à 300 Hz n'y sont pas montrées ; elles sont en revanche transmises sur la sortie sérielle. À l'écran s'affichent, sous la fenêtre, les mesures de tension et de fréquence du secteur ; en outre, la tension y est représentée par une barre analogique.

Sur le *shield* LCD, à gauche de l'écran, il y a trois boutons pour le réglage de l'analyseur de spectre. Leurs fonctions sont programmées comme ceci. Au démarrage, l'analyseur commence tout de suite à prendre des mesures et affiche à l'écran le tracé logarithmique du spectre mesuré. Appuyer sur le bouton du dessus commute vers la trace de fréquence avec un axe vertical linéaire. Un deuxième appui fait passer dans le domaine temporel, la fonction d'oscilloscope, et l'écran montre les 96 premiers échantillons dans le tampon d'entrée. Voyez les **figures 7 et 8** pour les réglages possibles. Un troisième appui sur le même bouton montre le contenu complet du tampon d'entrée dans une fenêtre adaptée par la fonction adéquate. Au quatrième coup, on revient au domaine de fréquence avec échelle logarithmique, comme au début.

Le bouton du bas permet de choisir la fonction de fenêtre TFR. Au départ, on a droit à la fenêtre

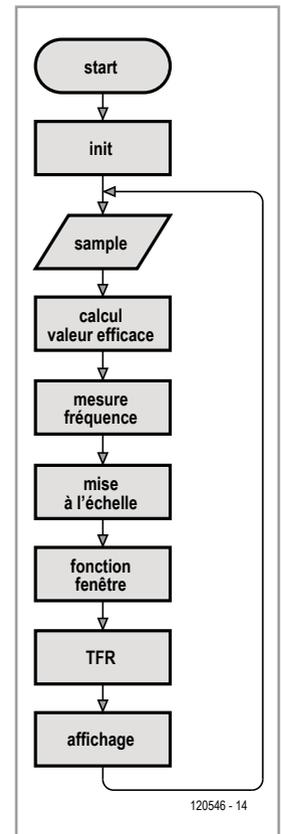


Figure 6. Ordinoigramme du programme.

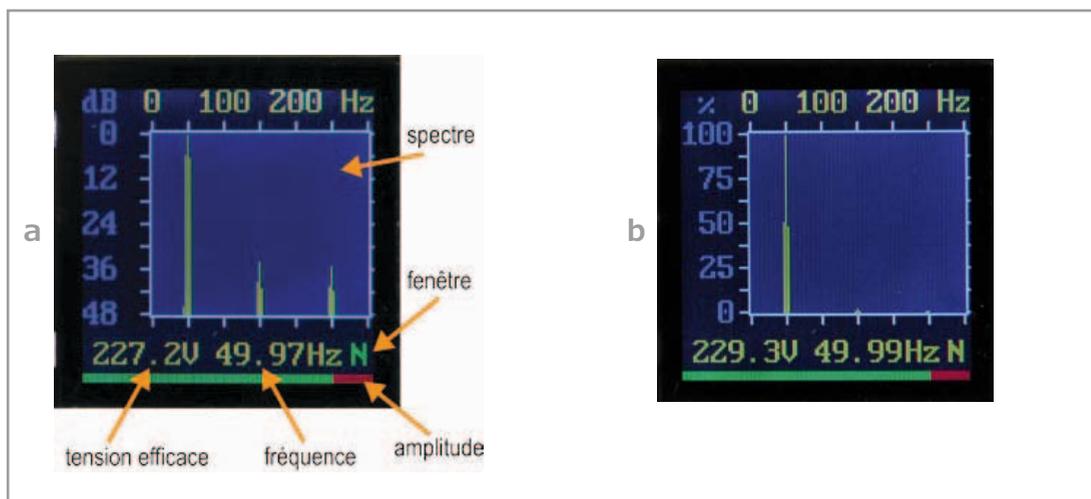


Figure 7. Copie d'écran dans le domaine de fréquence, échelle logarithmique en a, linéaire en b.

Hanning standard. En poussant une fois, on passe à la fenêtre Blackman et une fois supplémentaire, à la fenêtre Hamming. La fonction choisie est indiquée à droite, en bas de l'écran : N pour Hanning, B pour Blackman et M pour Hamming (attention aux jambages !).

Le bouton du milieu sert à activer et désactiver la communication série avec un terminal. Si aucune donnée n'est envoyée à l'ordinateur, l'indicateur de la fonction de fenêtre passe du jaune au vert et inversement après chaque calcul de spectre. Lorsque la communication série est activée, le témoin passe du jaune au rouge.

Performance

Selon les réglages du compilateur, le calcul d'une TFR dure, dans le cas le plus défavorable et sans optimisation (-O0), une seconde et le volume du programme compilé est d'à peu près 26 Ko. Avec une optimisation (-O2), la longueur du binaire est ramenée à quelque 18 Ko et la durée du calcul de la TFR à 700 ms. L'exécution de tout le programme, mesure de fréquence, acquisition, calcul de V_{eff} , mise à l'échelle, fonction de fenêtre, calcul de la TFR, mise à l'échelle et tracé de l'affichage, dure environ 2 100 ms moyennant l'optimisation -O2.

Pour garantir un temps de réaction prévisible et constant du système, j'ai choisi une récurrence du cycle de 2 500 ms. C'est, bien sûr, sans compter l'envoi des valeurs sur le port sériel. Pour une description détaillée des drapeaux -O, reportez-vous à la documentation du compilateur. Des 2 048 octets de RAM disponibles, seuls 1 372

sont mis à contribution, dont 1 024 pour les tampons de la TFR. Cela représente les 2/3 de la mémoire totale disponible. Côté mémoire de programme, la ROM disponible totalise 32 Ko, mais seulement 18 Ko sont en usage. En incluant l'amorce de l'Arduino, d'environ 2 Ko, il reste donc plus de 10 Ko de ROM libres, à disposition pour des extensions.

Conclusion

Le système décrit ici est un analyseur par TFR en temps réel, totalement fonctionnel pour la tension du secteur, capable de calculer toutes les 2,5 ms un nouveau spectre avec une TFR de 256 points en virgule flottante et de l'afficher graphiquement sur écran. Tout ceci a été réalisé sur une plateforme Arduino dont seulement un tiers de la mémoire de programme est utilisée. Cette application donne à voir la puissance et l'efficacité du microcontrôleur simple à 8 bits mis en œuvre pour exécuter des opérations mathématiques complexes et en présenter les résultats sous forme graphique, tout en conversant avec une interface utilisateur et un PC. Mais le microcontrôleur a aussi ses limites, comme la mémoire de travail disponible, ce qui limite la vitesse de calcul, mais aussi la résolution de la TFR à 256 points. Cependant, le champ est ouvert à d'autres optimisations.

Pour le développement, j'ai fait appel exclusivement à des outils à source ouverte ; quant à la plateforme Arduino elle-même, elle est aussi totalement ouverte et disponible.

Le logiciel est conçu en modules, lesquels peuvent aisément servir dans d'autres projets. En outre,

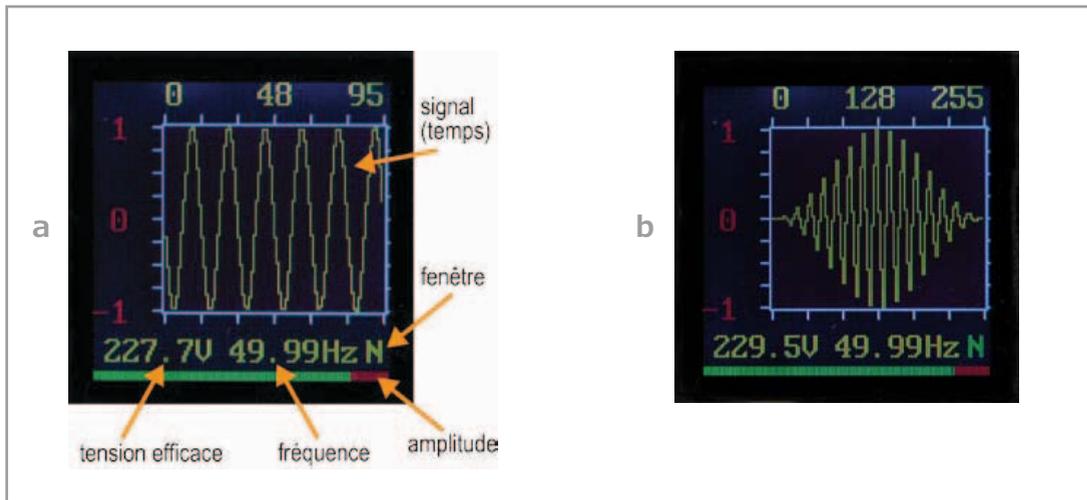


Figure 8.
Copie d'écran dans le domaine tempore :
96 échantillons sans fonction de fenêtre en a,
256 échantillons dans la fenêtre de Hanning en b.

le logiciel est publié sous licence GPL V3, ce qui permet de l'intégrer dans vos projets à vous. Les conditions de la GPL sont publiées sur le site de GNU Operating System [6]. Vous pouvez télécharger, de la page internet de l'article [3], le logiciel complet et le tracé du circuit imprimé.

Extensions possibles

Même s'il y a déjà de nombreuses fonctions implémentées, des extensions sont envisageables, comme le calcul de la DHT, la distorsion harmonique totale.

Une autre application serait la reconnaissance vocale à partir du spectre du signal de la parole, en utilisant la TFR pour l'afficher sur écran. Il faudrait alors élargir la bande passante du filtre passe-bas. Le CAN de l'ATmega 328 peut exploiter à pleine résolution de 10 bits une fréquence d'échantillonnage de 15 kHz, nettement plus que pour le signal de la parole qui se limite à environ 5 kHz.

Pour améliorer la vitesse du système, on peut effectuer l'acquisition de nouveaux échantillons en arrière-plan, pendant le transfert à l'écran. Cela permettrait de gagner 0,3 s, ce qui correspond à fournir 256 échantillons à une fréquence de 800 Hz. De plus, utiliser l'interface SPI de l'ATmega 328 permettrait une transmission vers l'écran encore plus rapide.

Bien sûr, en s'appuyant davantage sur le programme du PC pour l'affichage des graphiques, on pourrait encore améliorer la reproduction, mais ceci sort du cadre de la mise en œuvre d'une TFR sur une carte Arduino.

(120546 – version française : Robert Grignard)

Liens

- [1] <http://arduino.cc/fr/>
- [2] www.sparkfun.com/products/9363
- [3] www.elektor.fr/120546
- [4] <http://sourceforge.net/projects/winavr/>
- [5] <http://gcc.gnu.org/onlinedocs/gccint/index.html>
- [6] <http://www.gnu.org/copyleft/gpl.html>

L'auteur

Après ses études en électrotechnique à l'université technique de Berlin, Fidelis Theinert travaille pendant une vingtaine d'années comme concepteur de matériel et de logiciel dans différentes entreprises en Allemagne et aux Pays-Bas et comme professeur dans l'enseignement supérieur. Il enseigne aujourd'hui l'électrotechnique à la *Haagse Hogeschool* à Delft. Il est spécialisé dans les systèmes embarqués pour le traitement des signaux image et son.



sapins de Noël siamois

motifs lumineux variés, de longue durée et en stéréo

Willem Tak
(Pays-Bas)



Les traditionnelles décorations de Noël, après avoir longtemps mis le feu aux chaumières sous la forme d'abord de petites chandelles puis de guirlandes électriques, sont aujourd'hui remplacées partout par des bidules électroniques. Adieux ringardes loupiotes, même clignotantes, la mode est aux LED, avec effets lumineux embarqués. Et quels effets ! Dignes d'un éclairage de discothèque. Avec ses 56 LED, ce double arbre de Noël offre un spectacle visuel varié, étalé sur plus de trois quarts d'heure. Il est né le divin effet, jouez haut-bois.

Je n'ai plus aucune envie de ressortir du grenier mes vieilles décors de Noël, et encore moins d'installer dans mon salon un sapin, qu'il soit véritable ou véritablement bien imité. Les magasins débordent de guirlandes à diodes électroluminescentes de toutes les formes et couleurs possibles, commandées par des circuits électroniques barbants.

Pour mon petit sapin, moi je voulais quelque chose d'original, loin des guirlandes de pacotille

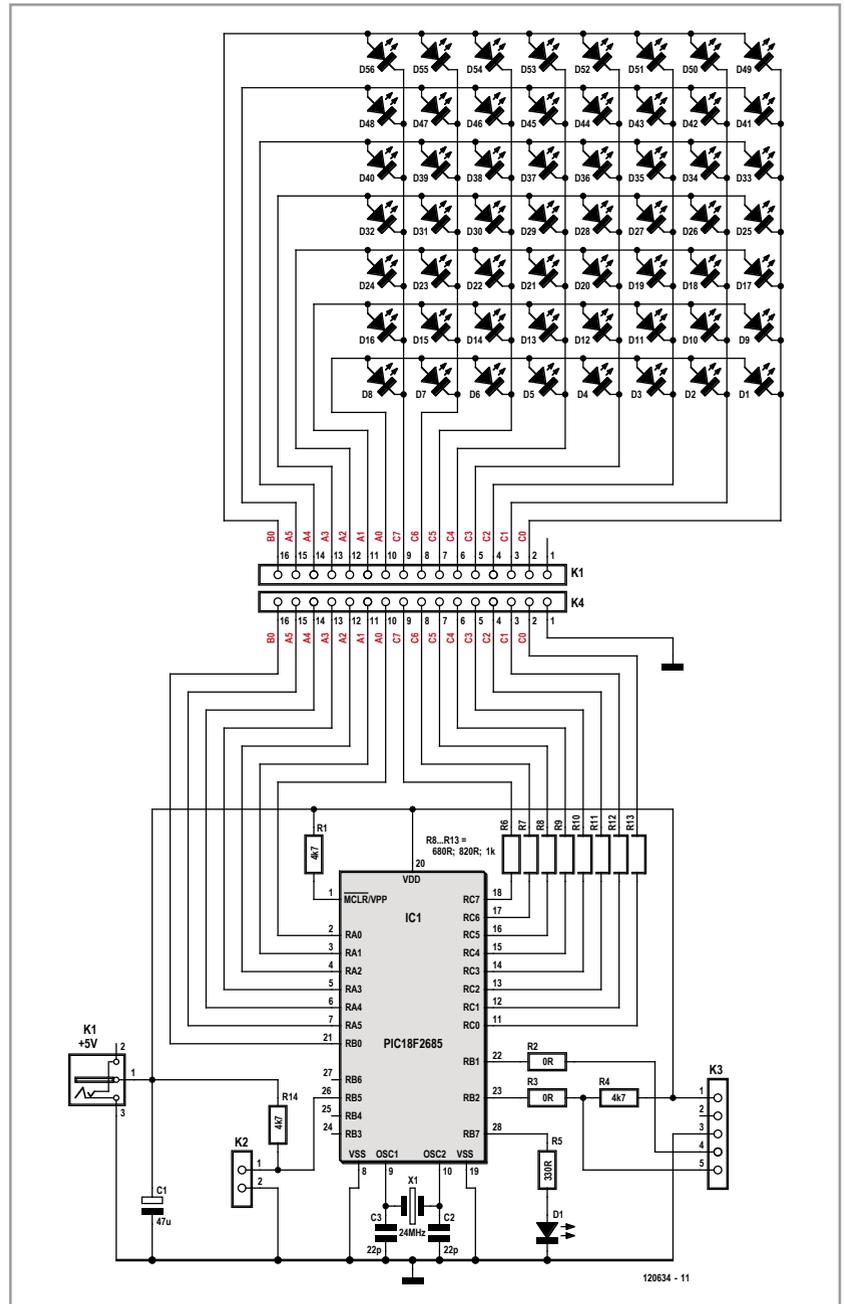
des magasins de bricolage et des jardinerias. Je propose donc un circuit qui ne compte à peu près rien de plus qu'un microcontrôleur et 56 diodes lumineuses. Les LED sont disposées selon le profil classique de l'arbre de Noël, comme sur la photo. Comme souvent, c'est dans le logiciel que réside toute l'astuce ! C'est lui qui fait s'allumer l'arbre selon les motifs les plus intéressants (voyez la vidéo sur [1]). Il est même possible de coupler deux de ces arbres pour

que les séquences passent alternativement de l'un à l'autre. Et ça donne une attraction stéréo-magique étonnante. Si vous habitez dans une rue passante, installez vos deux sapins sur un appui de fenêtre, et vous verrez leur effet sur les passants. Le spectacle dure plus de 40 minutes sans se répéter !

Un petit peu d'électronique, mais une grande mémoire

Pour obtenir les effets optiques à la hauteur de mes attentes et pour qu'on ne puisse pas les mémoriser ni les reconnaître quand ils se répètent, j'ai choisi pour les commander un microcontrôleur doté d'une imposante mémoire. Il s'agit du 18F2685-E/SP de *Microchip*, équipé de 96 Ko de mémoire flash comme espace de programme. Une fois programmée, il n'y reste pratiquement pas de place, en raison de la durée de la variété de la représentation. Le microcontrôleur travaille au rythme d'un quartz classique à 24 MHz, parce que l'horloge interne à 8 MHz s'est révélée trop lente pour certaines séquences. Le circuit proprement dit (**fig. 1**) est simple : les LED sont disposées en matrice de 8 x 7 et sont commandées directement par le contrôleur. Les 7 colonnes, de 8 LED chacune, sont reliées à la tension positive par les broches A0 à A5 ainsi que B0. Les lignes, de 7 LED chacune, sont ramenées à la masse par les sorties du port C à travers des résistances. Les niveaux sur les sorties C déterminent donc quelle LED d'une colonne s'allumera, alors que les six sorties A et B0 s'occupent du multiplexage des colonnes.

En raison même de la structure de la matrice, chaque LED ne peut s'allumer que pendant 1/7^e du temps. Comme en plus le courant délivré par le contrôleur est limité, tout cela a pour conséquence que les LED ne brillent pas très fort. Le circuit étant conçu pour une utilisation dans un environnement pas trop lumineux, la lumière tamisée d'un salon, par exemple. J'ai utilisé partout des LED de la firme SBL [2] qui ont donné des résultats plus que convenables. Plutôt que des modèles ordinaires, mieux vaut utiliser des LED à haut rendement et forte luminosité. Les sorties du contrôleur ne peuvent fournir que 25 mA au maximum. Comme elles sont multiplexées, c'est-à-dire allumées chacune à tour de rôle, elles ne voient circuler qu'à peu près 3 mA. Ne vous inquiétez pas, dans le noir, 3 mA c'est bien assez pour en mettre plein la vue. Vous pouvez adapter la luminosité en choisissant pour les résistances

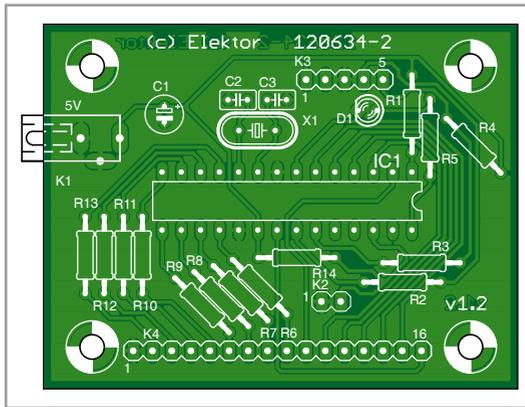


une valeur de 680 Ω, 820 Ω ou 1 kΩ.

Un connecteur de synchronisation K3 (sync), relié aux broches 22 et 23, permet de coupler l'un à l'autre deux sapins. Certaines figures pourront alors être échangées entre eux. Pour obtenir cet effet d'alternance, il faut les interconnecter par K3 au moyen d'une liaison croisée. Le logiciel détermine lequel est le sapin de gauche et celui de droite : l correspond au gauche, r au droit. Les effets sont conçus pour que l'arbre de gauche soit vu, comme son nom l'indique, de face à gauche.

Figure 1. Difficile de faire plus simple : un microcontrôleur commande directement 56 LED disposées en matrice. Le gros travail, c'est le vaste programme qui gère le spectacle.

Figure 2.
Le circuit imprimé pour le microcontrôleur accompagné de quelques composants passifs. Un électronicien averti pourra assembler le circuit sur un bout de platine d'expérimentation.



Le témoin d'état à LED sur la broche RB7 n'est pas indispensable. Lors de la mise sous tension ou d'une remise à zéro, elle clignote quelques fois. Elle sert à vérifier le bon fonctionnement du PIC et de son contenu.

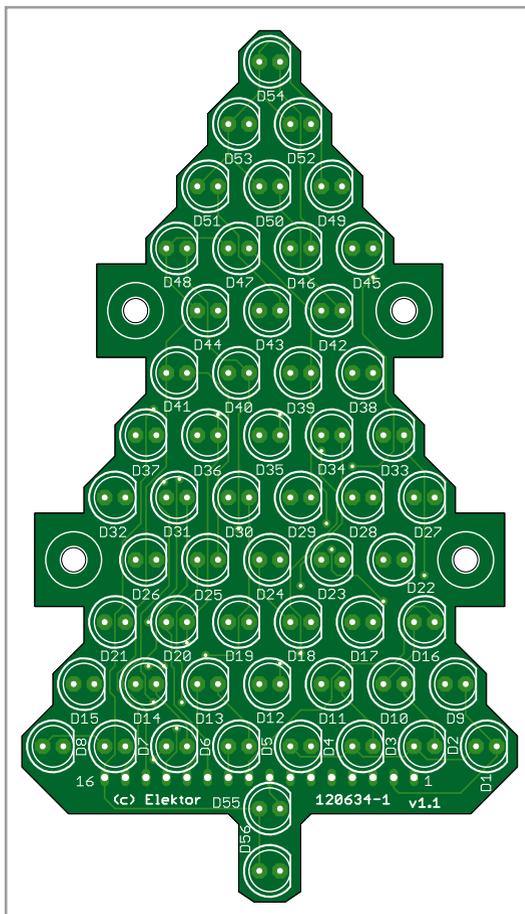
Comme alimentation, on se sert d'un adaptateur secteur de 5 V. N'importe quel modèle suffira certainement pour deux sapins et dans ce cas, la liaison de synchronisation peut aller de pair avec l'alimentation, tout est présent sur le connecteur K3.

Les petits circuits imprimés pour sapin de Noël que vous voyez aux **figures 2 et 3** sont disponibles sur le site d'Elektor [1]. Vous y implanterez facilement les composants, tous traversants. Pour le câblage entre le microcontrôleur et l'affichage, le mieux est d'utiliser un morceau de câble plat à 16 conducteurs.

Le logiciel pour ce circuit est à télécharger gratuitement du site d'Elektor [1]. Il y a des versions différentes pour l'arbre de gauche et celui de droite. Tenez-en compte au moment de la programmation. Bien sûr, si vous n'en construisez qu'un, cela ne fera aucune différence pour le PIC. Je vous recommande de visionner sur le site d'Elektor la vidéo que j'ai réalisée. Vous y trouverez aussi les fichiers pour les circuits imprimés, si vous préférez les graver vous-même. Et si vous désirez créer d'autres effets que les miens, libre à vous de modifier le code source, il est aussi disponible. Contentez-vous pour commencer de la cinquantaine de minutes de motifs lumineux préprogrammés, vous ne vous en lassez pas de si tôt...

(120634 - version française de Robert Grignard)

Figure 3.
Voici à quoi ressemble le circuit imprimé à double face en forme de sapin pour y implanter les LED.



Liens :

[1] www.elektor.fr/120634

[2] www.superbrightleds.com

Liste des composants

Circuit imprimé du contrôleur

Résistances :

- R1, R4, R14 = 4,7 kΩ
- R2, R3 = pont de câblage
- R5 = 330 Ω (optionnel)
- R6 à R13 = 680 Ω (ou 820 Ω of 1 kΩ, cf. texte)

Condensateurs :

- C1 = 47 µF/16 V

C2, C3 = 22 pF céram.

Semi-conducteurs :

- D1 = LED rouge, 3 mm (optionnelle)
- IC1 = PIC18F2685-E/SP (Microchip)

Divers :

- X1 = quartz à 24 MHz
- K1 = connecteur d'alimentation (p.ex. CUI PJ-031 DH)
- K2 = embase SIL à 2 broches (optionnelle)

K3 = embase SIL à 5 broches

K4 = embase SIL à 16 broches (ou câble plat à 16 conducteurs)

évent. poussoir pour mode et mise à zéro circuit imprimé 120634-1 (cf. [1])

Circuit imprimé des LED

- D1 à D56 = LED 5 mm à haut rendement
- K1 = embase SIL à 16 broches (cf. K4 contrôleur) circuit imprimé 120634-2 (cf. [1])

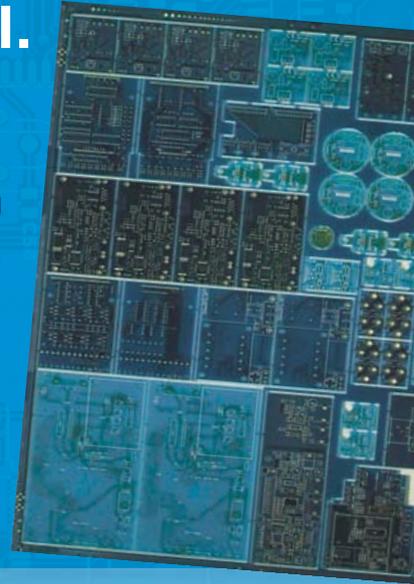
La référence Européenne pour les C.I. Prototypes et petites séries.

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

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

Nouveaux outils pour mettre vos dossiers en production plus vite

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



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

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

Prix avec TVA Française de 19.6% transport non compris

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

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

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

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

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

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

On demand – C.I. spéciaux minces et rigides

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

embarquez Linux ! (6)

Un démon sur le réseau

Benedikt Sauter [1]

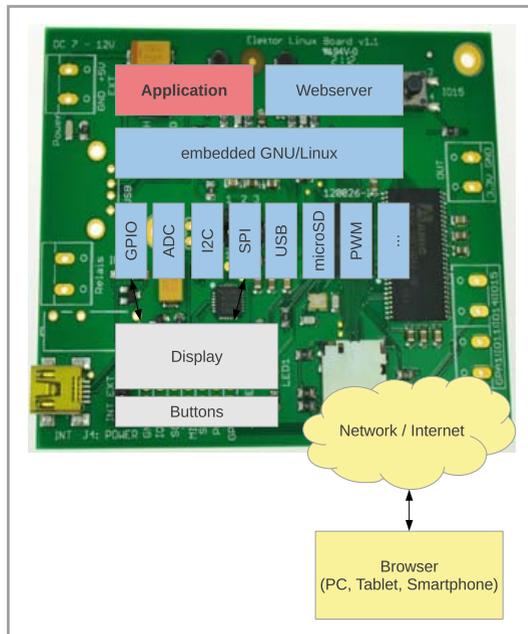


Figure 1.
Synopsis de l'application.

Au cours de cette série d'articles, nous avons abordé les thèmes essentiels de Linux, du chargeur de démarrage au noyau, en passant par l'installation de pilotes de périphériques. Le concept de démon que nous allons aborder ici va nous permettre d'être plus ambitieux, et de réaliser une application composée d'une « logique » en arrière-plan, d'une interface web pour sa configuration et son fonctionnement, et d'une unité de commande avec afficheur et boutons.

Messages entre applications

La **figure 1** décrit un scénario typique : le système Linux, une application, et un serveur web chargé de représenter l'interface sur les navigateurs de différents périphériques. L'afficheur et les touches sont optionnels, l'afficheur pouvant de plus être piloté par une application séparée. Dans ce scénario, l'application principale n'est qu'un processus exécuté en tâche de fond en même temps que de nombreux autres. Sous Linux, une telle application est appelée démon (*daemon* en anglais) ou service (p. ex. un service

La modularité de Linux en fait une plateforme parfaite pour qui souhaite une solution complète de mesure, de commande et de régulation. Les langages ne manquent pas pour écrire une application de commande et envoyer son interface à un PC, une tablette ou un smartphone. Après avoir montré comment ajouter la carte Linux-Elektor à un réseau local, nous pratiquerons ainsi avec une application serveur écrite en C et quelques applications écrites en *Lua*.

réseau). Une fois lancé, un démon travaille sans intervention de l'utilisateur. L'interface avec d'autres applications (p. ex. une interface web) peut se faire de plusieurs manières. Le terme « communication inter-processus » regroupe ces mécanismes d'échange entre programmes, processus ou fils. Voici quelques moyens de les mettre en œuvre : tubes, files d'attente FIFO, mémoire partagée, sockets, entrée/sortie standard, fichiers, bases de données, et autres programmes auxiliaires. Le moyen à utiliser dépend de plusieurs facteurs. Les applications ont-elles été écrites avec le même langage ? Le programme doit-il également pouvoir fonctionner sur d'autres systèmes d'exploitation ? Quel débit de données faut-il pour la communication entre les applications ?

Une méthode souvent employée est la communication par sockets. L'avantage avec les sockets est que les applications peuvent même échanger des données et des commandes à travers le réseau. Envoi et réception de messages sont de fait très simples avec les sockets, et la plupart des langages de programmation disposent

à cet effet de classes et de fonctions.

Nous optons donc pour les sockets. Et puisque la navigation sur l'internet repose elle aussi sur des sockets, nous en profiterons pour apprendre tous les moyens d'accéder directement à des données internet depuis la carte Linux-Elektor.

Connexion au réseau internet

Pour connecter notre carte à l'internet, nous avons besoin d'un routeur relié au réseau dont le rôle est de servir de passerelle aux autres participants du réseau. Un adaptateur USB/LAN relié à l'interface USB de la carte est bien entendu lui aussi nécessaire, et les pilotes de l'adaptateur réseau doivent être chargés [2].

L'ajout d'un participant à un réseau nécessite divers paramètres, comme l'attribution d'une adresse IP à la nouvelle machine, ainsi que la configuration de la passerelle et du serveur DNS. On effectuera ces paramètres soit manuellement, soit par DHCP (*Dynamic Host Configuration Protocol*). Avec le protocole DHCP, le routeur en charge du réseau local peut configurer automatiquement la carte Linux-Elektor (**fig. 2**). Pour tester une adresse IP attribuée par DHCP, vous pouvez lancer un *ping* vers, par exemple, le serveur web d'Elektor (**fig. 3**). Si aucun serveur DHCP n'est disponible, les paramètres de la connexion réseau doivent être entrés manuellement.

Adresse IP

L'attribution d'une adresse IP se fait avec la commande :

```
ifconfig eth0 <adresse_ip>
```

Remplacez simplement <adresse_ip> par une adresse IP disponible. Un *ping* vers le routeur permet de vérifier que l'adresse IP a bien été attribuée :

```
ping <adresse_routeur>
```

Ici aussi l'argument <adresse_routeur> doit être remplacé par l'adresse IP du routeur.

Adresse du routeur

Avant de pouvoir envoyer des paquets réseau via le routeur, il faut définir la route par défaut utilisée pour le trafic. GNU/Linux dispose pour cela de la commande :

```
root@gnublin:~# udhcpc
udhcpc (v1.17.3) started
Setting IP address 0.0.0.0 on eth0
Sending discover...
Sending select for 192.168.0.190...
Lease of 192.168.0.190 obtained, lease time 864000
Setting IP address 192.168.0.190 on eth0
Deleting routers
Adding router 192.168.0.1
Recreating /etc/resolv.conf
Adding DNS server 192.168.0.1
root@gnublin:~#
```

Figure 2. Configuration de l'adresse IP par DHCP.

```
root@gnublin:~# ping www.elektor.de
PING www.elektor.de (94.236.12.177): 56 data bytes
64 bytes from 94.236.12.177: seq=0 ttl=108 time=89.798 ms
64 bytes from 94.236.12.177: seq=1 ttl=108 time=76.505 ms
```

Figure 3. Un *ping* pour un bref dialogue avec le serveur d'Elektor.

```
route add default gw <adresse_routeur>
```

Même chose que ci-dessus, <adresse_routeur> est à remplacer par l'adresse IP de la passerelle. On pourra vérifier la bonne affectation de cette route en tentant de dialoguer avec un ordinateur quelconque du réseau internet, dans l'exemple ci-dessous avec le serveur web d'Elektor :

```
ping 94.236.12.177
```

Si le résultat est positif (pas de paquets perdus), on peut passer à la dernière étape.

Adresse du serveur DNS

Pouvoir utiliser directement un nom de domaine plutôt qu'une adresse IP pour naviguer sur l'internet exige un serveur DNS (*Domain Name System*). Le rôle d'un serveur DNS est de trouver l'adresse IP associée à un nom de domaine, par exemple à www.elektor.fr.

Modifions le fichier */etc/resolv.conf* à l'aide de l'éditeur nano afin de configurer un serveur DNS pour la carte Linux-Elektor (**fig. 4**) :

```
nano /etc/resolv.conf
```

Ajoutez l'adresse IP de votre routeur derrière l'élément *nameserver*. La plupart du temps, un routeur fournit aussi un service DNS ; si ce n'est pas le cas du vôtre, vous pouvez utiliser un de ceux disponibles sur l'internet. Celui de Google est idéal pour effectuer des essais, son adresse IP étant de plus facile à retenir : 8.8.8.8. Rappelons que sous nano un fichier s'enregistre avec Ctrl-o, et que cet éditeur de texte se ferme avec Ctrl-x. Vous devriez à présent être en mesure de



Figure 4.
Configuration du serveur DNS.

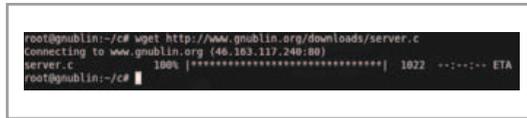


Figure 5.
Téléchargement du fichier *server.c*.

dialoguer avec n'importe quel serveur du réseau, à la fois par URL et avec la commande *ping*.

Programmer un démon de serveur

Servons-nous de cette connexion pour télécharger le fichier d'exemple *server.c*, un démon de serveur écrit en C. Reportez-vous à la **figure 5** pour l'adresse de téléchargement à passer en argument à *wget*, l'utilitaire que nous avons déjà utilisé pour récupérer divers fichiers depuis la console Linux. Si le compilateur de la carte suffit pour effectuer les premiers essais, il faudra revenir au PC et à la chaîne d'outils pour développer ce démon. Le **listage 1** reproduit son code source. Pour l'essentiel, un serveur simple est démarré et écoute le port 5000. Un participant du réseau qui contacte ce serveur reçoit automatiquement comme réponse l'heure actuelle du système. C'est à cette étape que l'on pourra plus tard ajouter un interpréteur de commandes, qui par exemple répondra ou agira différemment selon la commande envoyée par le client.

Commençons par compiler notre fichier C :

```
gcc server.c
```

Après compilation, démarrez le serveur avec :

```
./a.out
```

L'appel du programme bloque la console en cours, et toute action semble devenue impossible (à part forcer l'interruption du programme avec Ctrl-c). L'arrêt n'est qu'apparent, un Ctrl-z met le processus en sommeil et la console peut à nouveau servir à appeler des programmes ou

des commandes. À l'appel de Ctrl-z, la console affiche :

```
[1]+ Stopped ./a.out
```

Pour renvoyer le processus au premier plan, utilisez la commande *fg* (*fg* pour *foreground*) :

```
fg
```

La console est à nouveau bloquée, mais vous pouvez reprendre la main en suspendant une nouvelle fois le processus avec Ctrl-z. Avec la commande *bg* (*bg* pour *background*), le processus sera de nouveau exécuté en arrière-plan :

```
bg
```

Comme nous l'avons dit, lorsqu'un client se connecte à la « socket serveur », le serveur lui renvoie l'heure actuelle via la « socket client ». Nous pouvons vérifier cela depuis la console à l'aide du programme *telnet*, auquel nous passons en paramètre l'adresse IP du serveur ainsi que le port utilisé :

```
root@gnublin:~/c# telnet 127.0.0.1 5000
Tue Sep 27 21:34:49 2012
Connection closed by foreign host
```

Servez-vous de la commande *fg* si vous souhaitez renvoyer le processus de serveur au premier plan, par exemple pour le stopper.

Une requête peut bien sûr aussi être envoyée au serveur depuis une interface web. Et au lieu de l'heure, le serveur pourrait lancer une application quelconque (qui par exemple commanderait une LED). Nous avons déjà vu comment mettre ce mécanisme en pratique dans l'article précédent [2].

Utiliser une communication sécurisée

Jusqu'à présent, nous avons toujours utilisé un émulateur de terminal comme *picocom* pour accéder à la console de la carte. Avec notre nouvelle connexion, nous pouvons maintenant passer par le réseau pour ouvrir simplement et simultanément d'autres consoles, un énorme avantage lorsque l'on développe des applications. On se servira par exemple d'une console pour afficher le code source d'un programme dans

Listage 1 : Une application de serveur simple.

```

#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <time.h>

int main(int argc, char *argv[])
{
    int listenfd = 0, connfd = 0;
    struct sockaddr_in serv_addr;

    char sendBuff[1025];
    time_t ticks;

    listenfd = socket(AF_INET, SOCK_STREAM, 0);
    memset(&serv_addr, '0', sizeof(serv_addr));
    memset(sendBuff, '0', sizeof(sendBuff));

    serv_addr.sin_family = AF_INET;
    serv_addr.sin_addr.s_addr = htonl(INADDR_ANY);
    serv_addr.sin_port = htons(5000);

    bind(listenfd, (struct sockaddr*)&serv_addr,
        sizeof(serv_addr));

    listen(listenfd, 10);

    while(1)
    {
        connfd = accept(listenfd, (struct sockaddr*)
            NULL, NULL);

        ticks = time(NULL);
        snprintf(sendBuff, sizeof(sendBuff),
            "%.24s\r\n", ctime(&ticks));
        write(connfd, sendBuff, strlen(sendBuff));

        close(connfd);
        sleep(1);
    }
}

```

l'éditeur nano, et d'une autre console pour appeler le programme après chaque modification apportée au code, et ainsi pouvoir le tester en direct. Nous allons utiliser SSH (*Secure Shell*) pour établir une connexion réseau à la carte Linux-Elektor. Encore un avantage ici, personne ne pourra lire les données envoyées à la carte, puisque le protocole de communication SSH impose l'ouverture d'un canal crypté entre les deux parties.

SSH depuis Linux

Sous Linux, le programme ssh fait partie des outils standard. Pour vous connecter à distance depuis un ordinateur hôte, entrez la ligne suivante dans un terminal :

```
ssh root@192.168.0.190
```

Remplacez bien sûr l'adresse IP par celle attribuée à la carte. Une fois la connexion avec la carte établie, ignorez l'avertissement et répondez **yes** à la question posée par le programme (**fig. 6**). Vous avez maintenant accès à la console qui vous est désormais familière.

SSH depuis Windows

Sous Windows, vous pouvez utiliser l'émulateur de terminal PuTTY [3]. La **figure 7** reproduit sa fenêtre principale. Entrez l'adresse IP de la

carte, puis cliquez sur *Open*. Le programme vous demande si l'empreinte (*fingerprint*) de la clé du serveur distant est correcte (l'empreinte est une partie de la clé SSH qui sert à vérifier l'identité de l'ordinateur distant). Vous pouvez répondre **yes**. À l'invite d'ouverture de session, identifiez-vous

Listage 2: Script de démarrage.

```

#!/bin/sh

if [ ! -d /var/log/lighttpd ]
then
    mkdir /var/log/lighttpd
    chown -R lighttpd:lighttpd /var/log/lighttpd
    chmod 777 /var/log/lighttpd/
    touch /var/log/lighttpd/error.log
fi

modprobe asix
udhcpc

echo 3 > /sys/class/gpio/export
echo "out" > /sys/class/gpio/gpio3/direction
chown lighttpd:lighttpd /sys/class/gpio/gpio3/value

chown lighttpd:lighttpd /dev/lpc313x_adc
echo 1 > /dev/lpc313x_adc

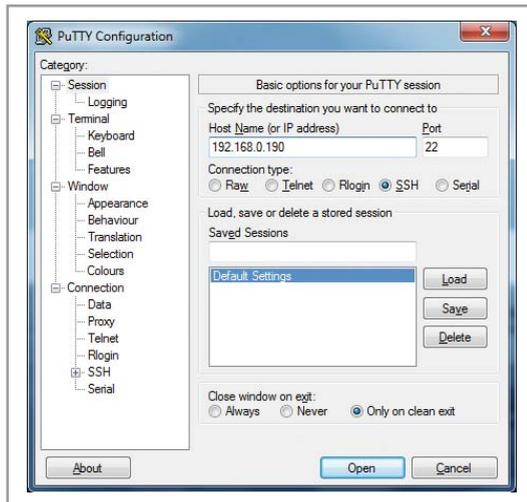
chmod 666 /var/log/lighttpd/error.log
/etc/init.d/lighttpd start

```

Figure 6.
Accès par SSH sous Linux.



Figure 7.
Accès par SSH sous Windows.



en tant que *root*. Validez avec la touche Entrée pour être connecté au système.

Lua, idéal pour des applications simples

Le langage C, qui nous a servi à écrire le programme de serveur précédent, est à la fois répandu, rapide et populaire. C n'est pas le seul langage de programmation, loin de là, à pouvoir être utilisé avec un système Linux. Le choix est vaste et se fait en fonction des avantages et inconvénients que présente un langage par rapport à l'objectif visé.

Le langage de script Lua est un langage interprété flexible, rapide et simple à utiliser. Lua tire sa popularité de son utilisation comme langage d'extension dans le développement des jeux

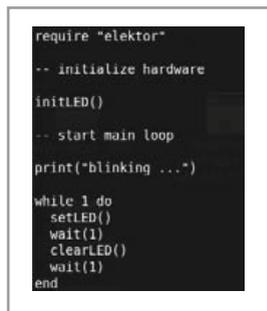
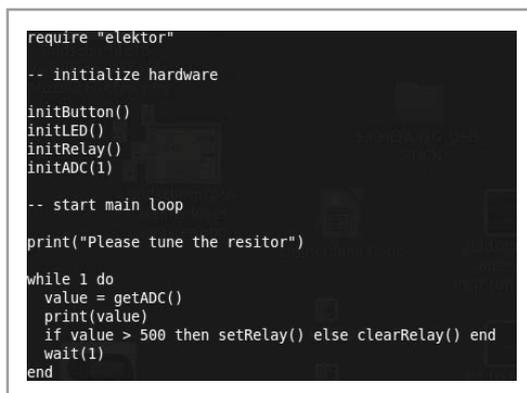


Figure 8.
Le programme d'exemple *blink.lua*.

Figure 9.
Le programme d'exemple *adc_relay.lua*.



vidéo. La taille de son interpréteur n'est que de 120 Ko, mais les possibilités de programmation qu'il offre sont néanmoins très grandes. L'accès aux fichiers, qui sous Linux englobe également l'accès aux périphériques ou aux fonctions matérielles, se fait avec les fonctions *io.open*, *io.read*, *io.write* et *io.close*. Benedikt Sauter a écrit une bibliothèque Lua pour la carte Linux-Elektor qui facilite l'utilisation des fonctions matérielles importantes (GPIO et convertisseur A/N).

L'avantage de Lua par rapport à C lorsque l'on souhaite développer rapidement une petite application, par exemple un programme de commande, est qu'il est possible d'écrire et d'exécuter directement sur la carte le code source à l'aide d'un des éditeurs de texte présents, nano ou vi.

Profitions une nouvelle fois de notre connexion réseau pour transférer les exemples Lua sur la carte SD :

```

mkdir lua
cd lua
wget http://www.gnublin.org/downloads/elektor.lua
wget http://www.gnublin.org/downloads/blink.lua
wget http://www.gnublin.org/downloads/button.lua
wget http://www.gnublin.org/downloads/adc_relay.lua
    
```

Le code en Lua reproduit sur la **figure 8** fait clignoter une LED. La première ligne charge la bibliothèque Elektor, l'instruction *initLED()* initialise la LED, et la boucle *while* l'éteint et l'allume sans fin.

Pour lancer le programme, tapez depuis la console :

```
lua blink.lua
```

L'application *button.lua* commande le relais de la carte par l'intermédiaire du poussoir IO15. Le code de l'exemple *adc_relay.lua* est reproduit sur la **figure 9**. La première instruction de la boucle *while* récupère la valeur de la tension lue par le convertisseur A/N. Le relais est fermé si cette valeur est supérieure à 500, ouvert sinon. Pour tester ce programme, reliez un potentiomètre à l'entrée analogique, comme nous l'avons expliqué dans le dernier article [2].

Les sites web d'introduction à Lua ne manquent pas [4], et il existe également de bons ouvrages sur le sujet [5].

Démarrer automatiquement un serveur web et une application

Il peut être contraignant d'avoir à lancer manuellement une application de serveur (et éventuellement d'avoir aussi à configurer le matériel) après chaque démarrage de la carte. Un bon moyen de contourner cet aléa est de regrouper les commandes nécessaires dans un script de démarrage. Le script du **listage 2** lance l'application de serveur que nous avons utilisée dans l'article précédent [5]. Recopiez l'adresse de la **figure 10** pour récupérer le fichier `start.sh` avec `wget`.

Pour que ce script soit lancé automatiquement au démarrage du système Linux, nous allons ajouter une entrée au script `/etc/rcS.d/S55bootmisc.sh` (le répertoire `/etc/rcS.d` contient les scripts de démarrage). Ouvrons donc ce fichier avec nano :

```
nano /etc/rcS.d/S55bootmisc.sh
```

À la fin du script sont présentes les entrées :

```
/bin/mkdir /var/run/sshd  
/usr/sbin/sshd
```

Ajoutez celle-ci :

```
/home/root/start.sh
```

Enregistrez le fichier avec Ctrl-o, puis quittez nano avec Ctrl-x. Le fichier `start.sh`, qui bien sûr doit être placé dans le répertoire `/home/root` spécifié, doit être rendu exécutable avec `chmod` (l'option `+x` ajoute un droit d'exécution) :

```
chmod +x /home/root/start.sh
```

Désormais, après chaque démarrage, la carte se connectera automatiquement au réseau et le serveur web sera lancé.

Accès par smartphone et tablette

Dans le dernier article, nous avons montré comment démarrer un serveur web léger et afficher une page web de démonstration dans un navigateur. La méthode est bien sûr identique avec un smartphone ou une tablette, si ce n'est qu'ici nous devons mettre en place un accès WLAN pour que les appareils mobiles puissent dialoguer avec la carte située dans le réseau. La **figure 11** montre l'interface HTML sur un



Figure 10. Téléchargement du script de démarrage pour le serveur web.



Figure 11. Accès depuis un iPhone.

iPhone. Le bouton permet de commander la LED de la carte.

Avant de conclure, notez que le wiki communautaire du projet GnuBlin [6] traite (en allemand et en anglais) des thèmes les plus divers, qu'ils concernent le matériel ou le logiciel. Y est en particulier expliqué comment envoyer et recevoir des données via les interfaces I²C, SPI et UART, et aussi comment transférer ces données dans l'espace utilisateur.

Le prochain numéro de cette série abordera les thèmes que vous aurez choisis. La sélection reste ouverte à l'heure où nous écrivons ces lignes, mais nous sommes impatients de vous satisfaire ! En attendant, et nous vous en reparlerons le mois prochain, nous travaillons à une carte d'extension qui offrira d'autres interfaces. Et voilà, maintenant vous aussi vous êtes impatients !

(120578)

Liens

- [1] sauter@embedded-projects.net
- [2] www.elektor.fr/120182
- [3] www.putty.org
- [4] www.lua.org
- [5] www.lua.org/pil
- [6] <http://wiki.gnublin.org>

retour aux sources (10)

hautes fréquences

Burkhard Kainka
(Allemagne)

La radio aura été le moteur de l'électronique et les hautes fréquences l'étincelle du moteur. Tubes électroniques et modulation d'amplitude furent les techniques pionnières, bien avant l'apparition des modulations VHF et FM. Aujourd'hui les semi-conducteurs règnent en maître et les fréquences utilisées sont de plus en plus hautes. Manipuler celles des ondes moyennes et courtes est néanmoins toujours excitant pour qui découvre l'électronique.

Construire une radio reste aujourd'hui un bon moyen de s'initier à l'électronique, à condition que le montage soit simple, sans composants spécialisés et sans danger pour l'utilisateur. C'est le cas de l'Audion à large bande de la **figure 1**, qui

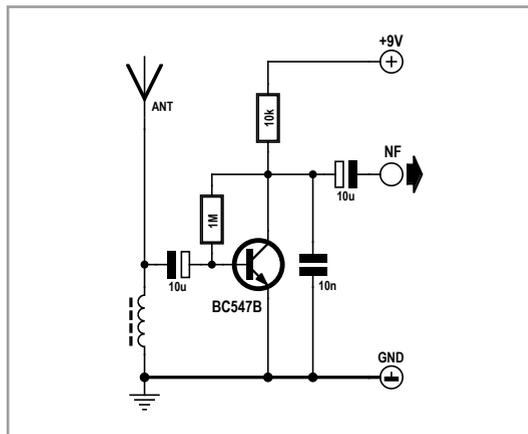


Figure 1.
Audion à large bande.

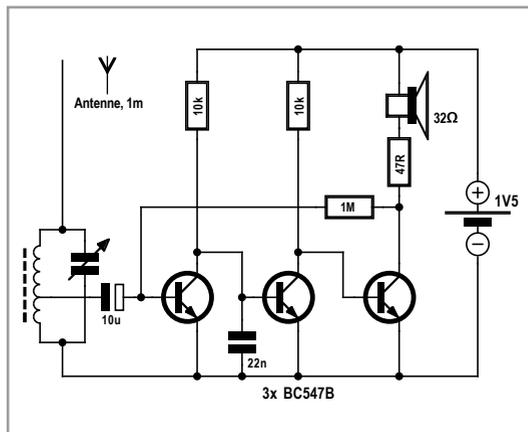


Figure 2.
Plus de sélection et d'amplification pour l'Audion.

démodule et amplifie des signaux HF. Ce récepteur des plus simple ramasse tout ce qui passe à portée d'antenne : émetteur en ondes moyennes proche, émetteur en ondes courtes très éloigné, ou bien sûr les perturbations HF qui baignent notre environnement. Il suffit d'une antenne filaire assez longue. L'idéal est une dizaine de mètres de fil suspendu, mais une antenne plus courte fera aussi l'affaire. La présence d'une liaison à la terre est également nécessaire. À la sortie BF, on branchera un casque de bonne sensibilité ou, mieux, un amplificateur et son haut-parleur.

L'Audion à large bande reçoit tout sans discrimination. On entendra donc pêle-mêle plusieurs émetteurs, dont le plus puissant dominera. Du choix de la bobine dépendra la plage de réception. On peut utiliser une inductance fixe : 300 µH pour la réception en ondes moyennes, 10 µH pour les ondes courtes, ou seulement 3 µH pour les bandes supérieures allant de 15 à 17 MHz. Pour construire vous-même le bobinage, enroulez 100 spires autour d'un barreau de ferrite ; pour les ondes courtes, entre 10 et 30 spires suffisent, avec un diamètre de 10 mm et sans noyau de ferrite. Le résultat dépend toutefois aussi de l'antenne.

Pour mieux comprendre, expérimentez avec plusieurs bobines (très) différentes. Les stations reçues changent en fonction du lieu de réception, de l'heure et de la bobine. Même avec ce récepteur simple, vous entendrez peut-être Radio Pékin, car dans les bandes décamétriques hautes la portée est très grande.

Remarquez que le circuit ne comporte qu'un seul amplificateur, celui qui amplifie le signal haute fréquence de l'antenne. Si cet amplificateur était parfait, nous n'entendrions rien du tout, car les fréquences en question sont situées bien au-delà du domaine audible. En fait, ce récepteur fonctionne grâce à la caractéristique d'entrée *tor-due* du transistor. Les demi-ondes supérieures du signal HF augmentent plus le courant de collecteur que les demi-ondes inférieures ne le diminuent, si bien que le courant moyen change en fonction de la tension d'entrée HF. Or dans l'émetteur de radiodiffusion, le signal BF à transmettre est utilisé pour moduler l'amplitude du signal HF (appelé porteuse). Et c'est ainsi que nous retrouvons ce signal BF en sortie du récepteur.

Les condensateurs de ce circuit jouent un rôle clé. Via la bobine d'entrée, le condensateur relativement gros placé à l'entrée court-circuite les signaux BF qui n'atteignent donc pas la base du transistor. Seuls les signaux HF y sont appliqués. Le circuit de collecteur est parcouru à la fois par des courants HF amplifiés et des courants BF démodulés. Les signaux HF sont toutefois court-circuités par le condensateur de 10 nF placé en sortie, et seules restent présentes les tensions alternatives BF audibles. Et voilà comment, avec trois fois rien, obtenir un récepteur doté d'une bonne sensibilité !

Le fonctionnement de la radio de la **figure 2** repose sur le même principe, mais les étages d'amplification sont ici plus nombreux et la sélectivité meilleure. L'entrée comporte un circuit d'accord réglable. Selon le dimensionnement de la bobine, la plage que peut capter la radio va des grandes ondes (GO) à la bande supérieure des ondes courtes (OC). La bobine de réception possède une prise intermédiaire à 10 % de l'enroulement. Grâce à elle, l'amortissement du circuit oscillant est faible et possède un facteur de qualité (ou de résonance) élevé (voir **l'encadré**) qui permet le filtrage de chaque émetteur. Avec trois transistors et une tension d'alimentation de seulement 1,5 V, cette radio est adaptée à une écoute au casque. Son circuit évoque celui de Tapir, le renifleur HF du numéro de juillet/août d'Elektor [1] qui lui aussi possède trois transistors, mais Tapir fonctionne comme un Audion à large bande.

Produire des signaux HF

L'émetteur HF le plus simple qui soit est le générateur d'étincelle(s). Un banal interrupteur suffit à

Circuits oscillants

Une bobine L et un condensateur C forment un circuit oscillant. Comme pour l'énergie mécanique d'un pendule, l'énergie électrique d'un circuit oscillant passe sans cesse de la bobine au condensateur ; ce processus se produit à une fréquence de résonance f caractéristique du circuit. Le circuit oscille de façon libre après avoir été excité par une brève impulsion de courant. Un circuit oscillant idéal ne perd aucune énergie, donc ses oscillations se maintiennent indéfiniment. En pratique elles sont amorties, car de l'énergie est perdue à travers la résistance ohmique du fil de la bobine, à travers le noyau de la bobine (énergie magnétique), et par rayonnement électromagnétique. Par simplification, l'ensemble des pertes peut être attribué à une résistance parallèle fictive R. La fréquence de résonance f est donnée par :

$$f = 1 / (2 \pi \sqrt{LC})$$

L'acuité de résonance d'un circuit oscillant est caractérisée par Q, le facteur de qualité. Q est donné par le rapport entre R, la résistance d'amortissement parallèle, et $R_L = 2 \sqrt{f L}$, la résistance inductive à la fréquence de résonance ; Q est aussi égal au rapport entre R et $R_C = 1/(2 \sqrt{f C})$, la résistance capacitive à la fréquence de résonance :

$$Q = R/R_L \text{ ou } Q = R/R_C$$

La tension d'un circuit oscillant excité à une fréquence variable par une source de courant alternatif (possédant une grande résistance interne) est maximale à la résonance. Moins les oscillations sont amorties par des pertes d'énergie quelconques, c.-à-d. plus le facteur de qualité est grand, plus la tension à la résonance est élevée. Sur la courbe de résonance, on peut déterminer des deux côtés de la fréquence de résonance les points pour lesquels la tension a baissé d'un facteur $1/\sqrt{2} = 0,707 = -3 \text{ dB}$. L'écart de fréquence entre ces points représente la largeur de bande b du circuit oscillant. Fréquence de résonance f , largeur de bande b et facteur de qualité Q sont reliés par $b = f/Q$.

Dans la plupart des cas, une conception soignée de la bobine permet d'atteindre un facteur de qualité Q de 100. Un circuit oscillant sera toutefois également amorti par le circuit auquel il est relié, ou encore par une antenne. Cet amortissement peut être atténué par un couplage lâche du circuit oscillant (obtenu avec un petit enroulement auxiliaire), par la prise d'une bobine, ou encore par un condensateur correctement dimensionné. Dans le cas d'une connexion directe à un amplificateur, la résistance de celui-ci doit être grande pour que l'amortissement reste faible.

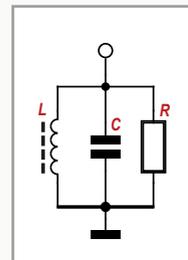


Figure 3.
Émetteur d'impulsions à
fréquence fixe.

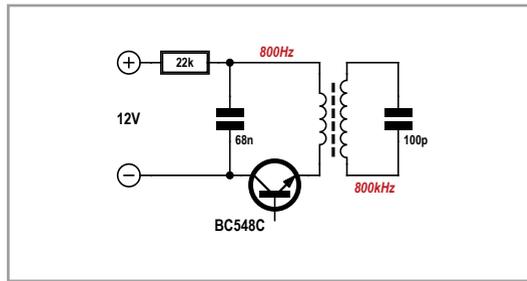


Figure 4.
Oscillateur sinusoïdal
avec couplage par
transformateur.

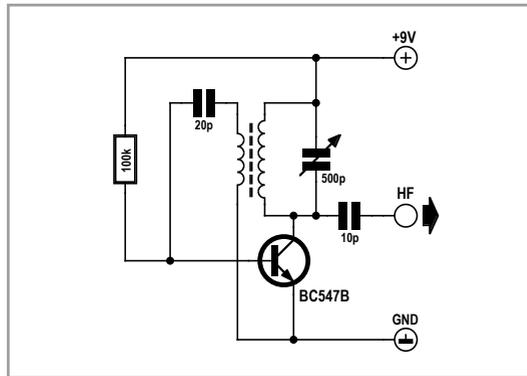


Figure 5.
Oscillateur pour petite
tension de service.

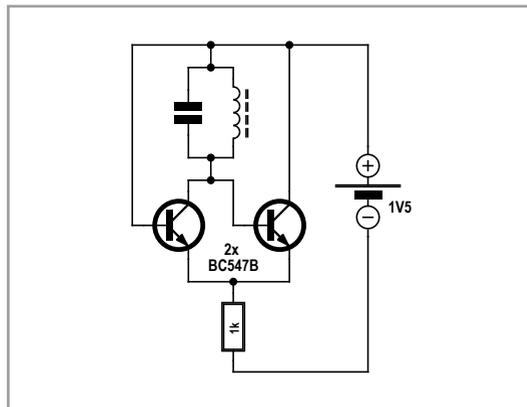
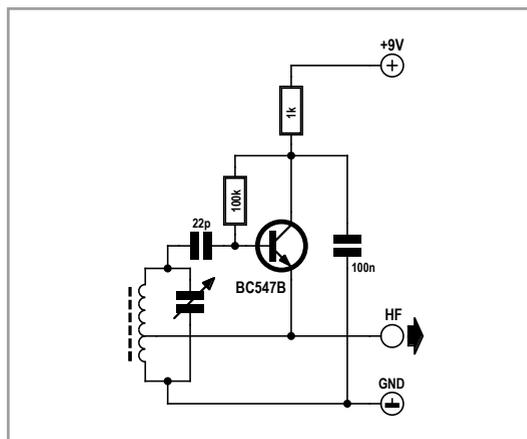


Figure 6.
Oscillateur avec une bobine
à prise.



en comprendre le principe. Le poste fera entendre un craquement lorsque vous actionnerez l'interrupteur d'une lampe électrique à proximité. Vous entendrez ce parasite sauf s'il est couvert par une station puissante qui émet à la fréquence sur laquelle est accordée la radio (quelque part dans les ondes moyennes).

D'où sort ce parasite ? Du signal de commutation dans l'interrupteur ; il présente un front raide qui comprend essentiellement des composantes du domaine des hautes fréquences. Les ondes moyennes sont très sensibles aux perturbations causées par les alimentations à découpage et autres perceuses... Le problème n'est ainsi pas de produire des signaux HF, mais bien l'inverse, à savoir éviter l'influence des signaux HF indésirables.

Sans songer à se transformer en station pirate, disposer d'un rien de puissance HF est parfois nécessaire, ne serait-ce que pour tester son propre récepteur. Le circuit de la **figure 3** est un mini « émetteur » qui fonctionne selon le principe des vieux émetteurs à arcs. Les « étincelles » périodiques sont ici produites par les oscillations de relaxation d'un transistor NPN (cf. le septième volet de cette série [2]) à une fréquence de 800 Hz. Chaque front de commutation raide excite le circuit oscillant, les oscillations produites s'affaiblissant rapidement (oscillations amorties). La fréquence d'émission est déterminée par les composants du circuit oscillant. Le barreau de ferrite bobiné d'une ancienne radio fournit ainsi un signal dans la gamme des ondes moyennes, 800 kHz par exemple. Une radio située près du bâton de ferrite produit un bourdonnement. Ne manque plus qu'un manipulateur de morse pour compléter le tout. Les émetteurs à étincelles maritimes d'autrefois fonctionnaient de la même façon, si ce n'est que la puissance était plus grande et les fréquences beaucoup plus basses.

Oscillateurs LC

La grande avancée est venue des tubes électroniques. Alors que les émetteurs à étincelles ne pouvaient produire que des oscillations amorties, les tubes étaient capables d'entretenir des oscillations non amorties. Si vous avez lu le volet consacré aux oscillateurs sinusoïdaux [3], vous devinez ce qui est nécessaire : un gain suffisant associé à une réaction adaptée.

L'archétype de l'oscillateur HF est l'oscillateur Meissner (ou Armstrong), dans lequel un enroulement de couplage fournit la position de phase correcte (**fig. 4**). Les premiers émetteurs de radiodiffusion suivaient également ce modèle : un tube, un circuit oscillant, une réaction, et l'oscillateur de puissance HF était prêt. Le câble d'antenne était en outre équipé d'un microphone à charbon, chargé également de la modulation.

L'oscillateur de Meissner de la figure 4 comporte un transistor. Le faire osciller n'est pas difficile. Si cela ne marche pas, il est probable que la position de phase soit incorrecte ; dans ce cas, il suffit d'inverser la polarité de la bobine de couplage.

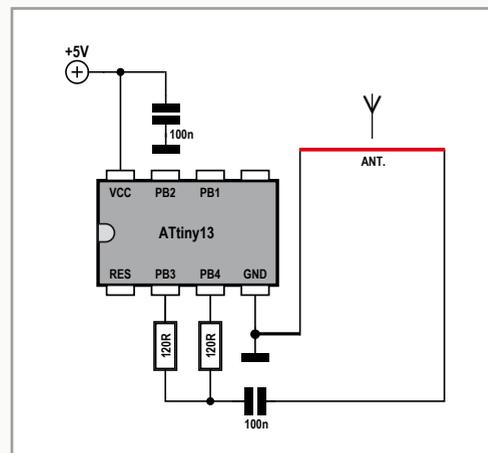
La fréquence peut être réglée avec le condensateur variable. L'amplitude dépend de nombreux facteurs, entre autres de l'amortissement de la bobine, de l'enroulement de réaction, et du condensateur de couplage. Stabiliser l'oscillateur passe parfois par quelques tâtonnements, car notre circuit a un défaut : des oscillations intermittentes peuvent apparaître pour des amplitudes trop grandes. La diode base-émetteur redresse la tension HF et charge négativement la base, jusqu'à ce que le transistor soit bloqué et que les oscillations décrochent. Il faut ensuite quelques microsecondes avant que la résistance de base ne ramène le transistor à son bon point de fonctionnement et que tout recommence.

L'oscillateur de la **figure 5** n'a pas ce problème puisque l'amplitude y est stabilisée de façon automatique. Le gain est élevé, donc la bobine n'a ici besoin ni d'enroulement de couplage, ni de prise, ni d'un grand facteur de qualité. Le circuit se prête bien au bricolage et démarre toujours sans problème. La réaction est obtenue à l'aide de deux transistors qui travaillent avec une tension très basse (0,6 V). L'ensemble du circuit fonctionne à partir d'une tension de service de 1 V et peut être utilisé avec des courants très petits, un avantage pour certaines applications.

Les fréquences vont du domaine BF à la plage VHF selon le dimensionnement du circuit oscillant, et ce circuit peut donc être assemblé et vérifié à peu de frais, p. ex. avec un petit générateur d'essai. L'inconvénient est que les capacités internes des transistors ont une influence sur la fréquence, et qu'elles dépendent aussi de la tension de service. Si ce circuit est donc très simple, il n'est toute-

Un générateur AM avec Tiny13

Disposer d'un petit générateur d'essai AM est utile lorsque l'on construit et développe un récepteur radio. Si la forme d'onde du signal convenable, il n'est en effet même plus nécessaire de régler la fréquence. Ce sera le cas si le générateur travaille sur une fréquence suffisamment basse et produit assez d'harmoniques. Notre



générateur AM à base d'ATtiny13 produit des impulsions courtes à une fréquence de 70 kHz. Il en résulte des harmoniques de grande amplitude sur toute la plage des ondes moyennes et longues. Le train d'impulsions est brièvement interrompu à intervalle régulier, et donne une modulation d'amplitude avec un son de 750 Hz. Une radio captera alors le signal de test d'un émetteur AM sur 70 kHz, 140 kHz, 210 kHz, et ainsi de suite.

Pour l'antenne, on formera par exemple une boucle de fil d'un diamètre de 10 cm. Elle produira un champ magnétique alternatif qui se couplera avec le barreau de ferrite du récepteur. Le générateur peut aussi servir à effectuer des mesures comparatives de la sensibilité, en cherchant simplement la distance à partir de laquelle le signal n'est plus perceptible. Un bon récepteur capte encore un signal clair à un mètre de distance.

Téléchargez le code source depuis la page [4] associée à cet article.

```
'ATtiny13 AM Generator
$regfile = "attiny13.dat"
$crystal = 1200000
$hwstack = 8
$swstack = 4
$framesize = 4

Config Portb = Output
Dim N As Byte

Do
  For N = 1 To 50   '70 kHz
    Portb = 255
    Portb = 0
  Next N

  For N = 1 To 50   'AM 750 Hz
    nop
    nop
  Next N
Loop

End
```

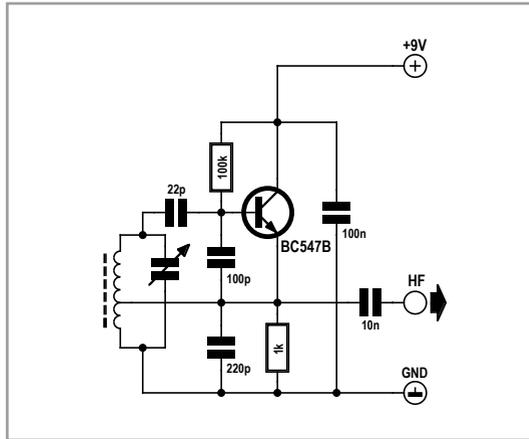


Figure 7. Oscillateur avec diviseur de tension capacitif.

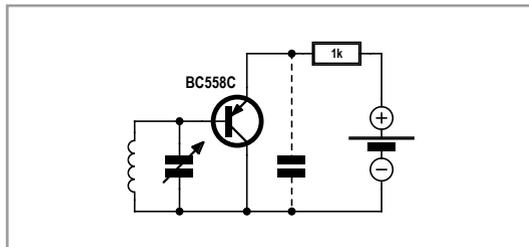


Figure 8. Oscillateur en collecteur commun avec couplage CC.

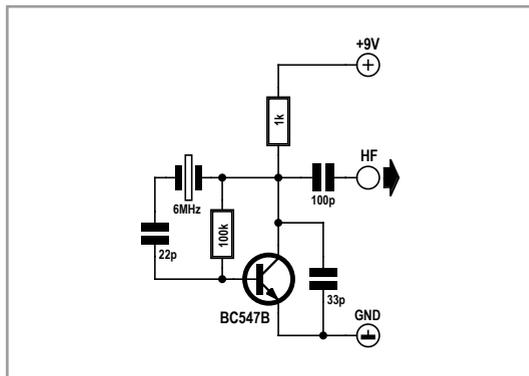


Figure 9. Oscillateur à quartz.

fois pas un modèle de stabilité du point de vue de la fréquence. Peut-être saurez-vous l'exploiter autrement, par exemple dans un dispositif de surveillance sans fil de l'état des piles ?

Une bobine avec prise permet également d'obtenir la relation de phase correcte pour un oscillateur (fig. 6). On peut également se servir d'un diviseur de tension capacitif (fig. 7). Le circuit peut être simplifié (fig. 8) en utilisant un transistor PNP monté en émetteur suiveur (collecteur commun) avec couplage CC direct. Le diviseur de tension capacitif de la figure 7 peut être omis si l'amortissement de la bobine est suffisamment faible, car les capacités internes du transistor joueront le même rôle. La tension de service peut

là encore être basse, p. ex. 1,5 V. Le couplage direct avec le circuit oscillant influence toutefois ici aussi la valeur de la fréquence, car les capacités de jonction changent selon la tension de service appliquée. Cet oscillateur simple met aussi en lumière les problèmes de stabilité que peuvent rencontrer certains étages d'amplification en collecteur commun. Ce qui ne devait être qu'un émetteur suiveur utilisé comme adaptateur d'impédance peut ainsi se transformer accidentellement en oscillateur. Amplificateur ou oscillateur, la ligne de partage entre les deux dépend par exemple du point de fonctionnement et de l'amortissement du circuit oscillant.

Oscillateurs à quartz

L'oscillateur à quartz est la meilleure solution pour qui a besoin d'une fréquence parfaitement stable. Un quartz est l'équivalent d'un circuit oscillant possédant un facteur de qualité extrêmement élevé. Un quartz permet donc de réaliser facilement un oscillateur stable sans bobine. Le circuit de la figure 9 est typique.

Si vous souhaitez construire une radio et expérimenter avec les ondes moyennes, vous devriez songer à faire vos essais avec un oscillateur modulé. Un tel émetteur de test n'aura pas à être relié à une antenne et ne dérangera pas votre voisinage, car l'énergie HF est ici transférée directement de bobine à bobine.

L'oscillateur d'un tel modulateur pour ondes moyennes doit toutefois être stable pour être utile. Comme le quartz reste très cher pour des fréquences aussi basses, on lui préfère souvent un résonateur céramique. Les fréquences de 500 kHz ou de 2 MHz sont faciles à obtenir, toutes deux étant situées en dehors de la gamme des ondes moyennes, mais des fréquences plus adaptées sont possibles. Nous avons p. ex. trouvé un résonateur céramique de 976 kHz dans la télécommande d'un vieux téléviseur. La figure 10 représente le circuit complet d'un oscillateur d'essai AM. Condensateur et résistance variables permettent un réglage fin. Si une station (probablement faible) est encore audible en arrière-fond, réglez simplement le circuit sur battement nul, p. ex. sur 981 kHz. La petite bobine d'émission à noyau de ferrite est directement couplée au barreau de ferrite de la radio. Après quelques modifications, ce circuit pourrait fonctionner en ondes courtes avec un quartz, et être exploité

sur des bandes de fréquences qui ne nécessitent pas de licence, p. ex. 13,56 MHz.

L'étage modulateur est monté en émetteur suiveur et module la tension de service de l'étage amplificateur final. Puisque les ondes courtes et moyennes sont encore en mono, les deux canaux d'entrée d'un signal stéréo ont été combinés. Le potentiomètre sert à obtenir la distorsion la plus faible possible et le meilleur son. L'étage d'amplification HF est volontairement réduit, car ici la puissance n'est pas le critère essentiel.

Audion ondes courtes avec réaction

Le récepteur d'ondes courtes à trois étages de la **figure 11** possède une réaction réglable. Le premier étage est un oscillateur semblable à celui de la figure 8, mais ici le point de fonctionnement peut être ajusté de façon arbitraire pour réduire le gain. L'astuce consiste à compenser les pertes du circuit oscillant en réglant le gain à la limite de l'oscillation propre n'apparaisse. En ce point le récepteur possède le plus grand gain et la meilleure sélectivité. L'étage oscillateur PNP en collecteur commun démodule le signal HF et constitue en même temps l'étage Audion. Les deux étages BF suivants fournissent la puissance nécessaire à l'utilisation d'un petit haut-parleur.

Que l'Audion puisse être configuré pour exploiter les oscillations dépend également de l'amortissement du circuit d'entrée via l'antenne connectée. C'est la raison pour laquelle sont présentes deux prises d'antenne. La connexion par le petit condensateur de liaison donne un couplage lâche et un amortissement faible. Quant à la connexion directe, elle convient aussi pour les antennes très courtes (une antenne longue rayonne en effet également de l'énergie HF, et atténuée donc le circuit oscillant). Un Audion habilement réglé peut avoir une très grande sensibilité. Ces récepteurs audio faisaient partie de l'équipement standard des débuts de la radio. Même un émetteur faible pouvait avoir une portée de plusieurs milliers de kilomètres. Il est possible de construire soi-même d'excellents récepteurs avec un minimum d'efforts, donc lancez-vous, en plus ce circuit vous fascinera !

Nous voici arrivés au terme de cette série de dix articles d'initiation. Au fil des prochains numéros, chaque fois que l'occasion s'en présentera, nous proposerons un retour aux sources de l'électro-

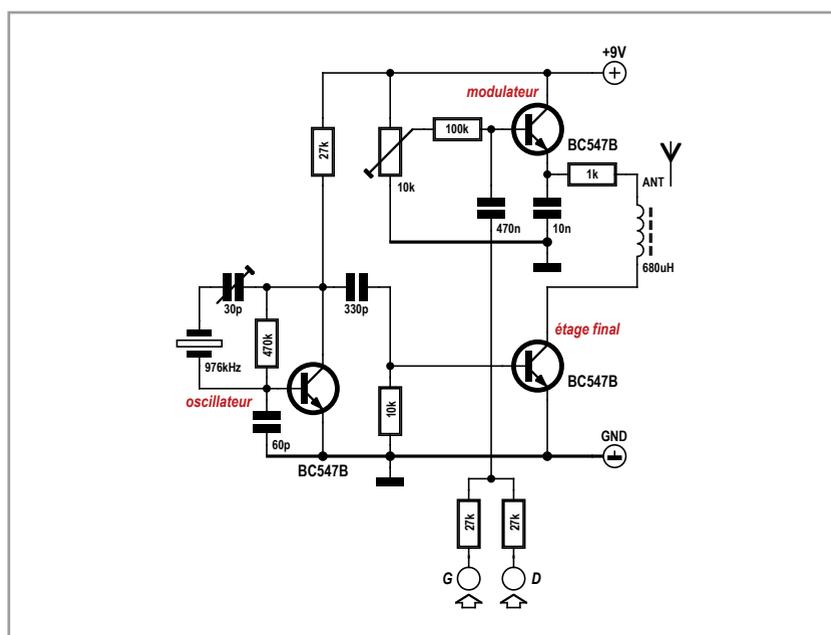


Figure 10. Le modulateur pour ondes moyennes. Nous explorerons également les amplificateurs opérationnels et quelques circuits logiques et numériques essentiels. Restez branchés !

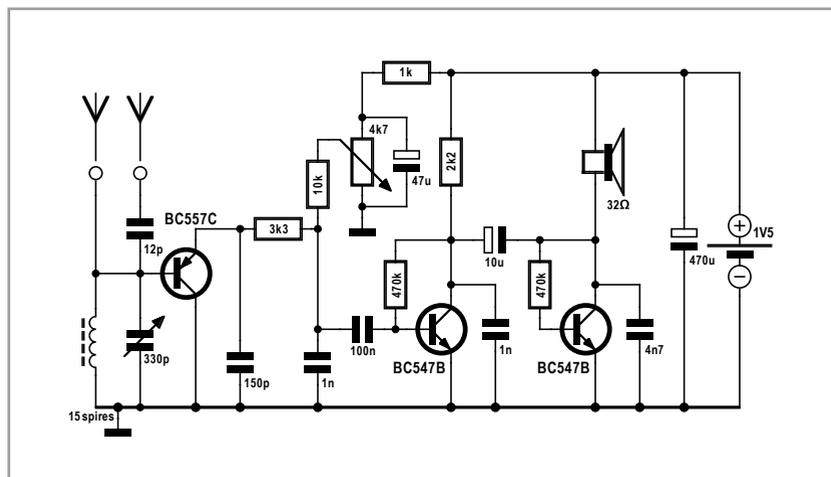
Figure 10. Le modulateur pour ondes moyennes.

(120010 – version française : Hervé Moreau)

Liens

- [1] www.elektor.fr/120354
- [2] www.elektor.fr/120007
- [3] www.elektor.fr/120009
- [4] www.elektor.fr/120010

Figure 11. Audion à réaction.



l'internet des objets



Tessel Renzenbrink (Elektor TTF)

L'internet des objets (IdO ou IoT pour *Internet of Things*) marquera la fusion des mondes physique et numérique. Des objets de la vie courante dotés d'un système informatique miniature se verront attribuer une adresse IPv6 qui permettra leur représentation numérique dans l'internet. Nous pourrions alors obtenir des informations sur ces objets grâce à l'internet, et même les commander à distance.

www.techthefuture.com

L'humain sera absent de l'IdO dans de nombreux cas. L'IdO est un réseau de machine à machine dans lequel capteurs et actionneurs communiquent entre eux pour interagir avec leur environnement.

Durant le festival PICNIC qui s'est tenu à Amsterdam en septembre, Florian Michahelles, Heico Sandee et Stefan Gessler ont animé l'atelier « ROBIOTS : *Robots Meet Internet of Things* ». J'ai pu discuter avec eux des défis technologiques et des risques liés à l'IdO.

Trois experts

Florian Michahelles est directeur des Auto-ID Labs de Saint-Gall (Suisse) dont les recherches portent sur les RFID en réseau. Il s'intéresse plus particulièrement à l'internet des objets.

Heico Sandee, directeur de programme au département de génie mécanique de l'université d'Eindhoven, est un spécialiste des systèmes de régulation. Il participe au projet RoboEarth, un partenariat multidisciplinaire entre roboticiens dont l'objectif est de créer un réseau web pour les robots.

Stefan Gessler est directeur de projet au département Services & Logiciels des laboratoires NEC. Il est membre du projet Florence, dont le but est d'améliorer le quotidien des personnes âgées à l'aide de robots bon marché et collabore au forum international IoT.

Objectif de l'IdO

Tessel : *Pourquoi avons-nous besoin d'un IdO ?*

Florian : Une des réponses possibles : simplement parce qu'il s'agit d'une évolution logique.

Heico : Pour exercer plus de contrôle sur notre environnement. Je suis tellement habitué à pouvoir communiquer avec mes semblables d'un simple mouvement de doigt, que je trouve insensé de ne pas pouvoir le faire avec les objets. Je veux pouvoir vérifier avec mon téléphone si j'ai bien pensé à fermer ma porte et même pouvoir la fermer à distance.

Stephan : L'IdO, repose sur la coopération et la communication. La communication est la condition préalable à la coopération. La collaboration entre objets se traduit par d'innombrables avantages. Pensez à un capteur qui détecte de l'eau sur un sol, il pourra fermer le robinet tout seul. Ou à une voiture qui indique aux véhicules derrière elle de ralentir parce qu'elle a détecté un danger sur la route.

Heico : Et cela ouvre de nouvelles possibilités. Dans le projet RoboEarth, nous créons un système dans lequel des objets, des robots et des humains échangent des informations. Mettons que vous reliez une machine à café à votre réveil. La machine se mettra en route dès que votre

réveil sonnera. C'est déjà possible avec certaines machines intelligentes, mais quand il n'y a plus de café ? C'est là que le robot entre en jeu. Parce qu'il peut se déplacer, il saura remettre du café là où il faut. Les robots donneront des bras et des jambes à l'internet.

Et des robots en réseau peuvent apprendre les uns des autres. RoboEarth développe une base de données dans le nuage à laquelle ont accès plusieurs types de robots. Quand un robot a résolu un problème, il peut ainsi ajouter sa solution à la base, et un robot qui rencontre le même problème peut télécharger les instructions idoines. Vous n'avez plus à envisager chaque problème possible dès la conception du robot, ni à l'équiper de toute une batterie de capteurs pour lui donner des capacités d'apprentissage.

Défis technologiques

Tessel : Quels défis technologiques attendent l'IdO ?

Stefan : Le plus grand sera de modéliser l'information, pour que chaque dispositif puisse communiquer et coopérer avec les autres. En ce moment, l'approche dite « en silo » est à la mode. Imaginez des dispositifs conçus pour exécuter une tâche particulière dans un certain champ d'application. Ce système est une boîte fermée qui comprend des capteurs, des actionneurs, et un programme. Pour une autre application, vous devez tout racheter, les capteurs, les actionneurs, mais cette fois-ci avec un logiciel et des protocoles différents. Ensuite vous devez réutiliser les données et les partager entre les dispositifs. Vous avez donc besoin d'une interface interopérable. C'est à cela que nous travaillons en ce moment.

Florian : Aux labos d'Auto-ID, nous cherchons à rendre plus efficaces le flux des marchandises et la chaîne logistique. L'idée est d'associer un code à barres ou un identifiant numérique RFID unique aux données qui décrivent le type d'un produit. Le code à barres est certes également utile aux consommateurs équipés de téléphones tactiles, mais d'après nous le consommateur veut obtenir autre chose que le poids ou les dimensions d'un produit, plutôt sa composition ou son usage. C'est à partir de ces données que nous devons définir une sémantique compréhensible par les robots. Dans l'IdO, les différents systèmes que l'on voit émerger pour différentes raisons doivent travailler ensemble.

Cela dit, la recherche de normes et de standards n'est pas tant un défi technique qu'un problème d'organisation et de motivation, parce que vous n'avez pas une seule entreprise qui fournit l'infrastructure, mais plusieurs. Comment les faire s'entendre autour d'une table ? Si une seule entreprise fournit les données, quel intérêt aurait-elle à les partager ? La bonne question est : quel est l'écosystème ? Quel intérêt commercial commun pour que chacun y trouve son compte ?

Heico : D'accord pour voir l'interopérabilité comme le plus grand des défis à relever, mais où est vraiment le problème ? Faire un choix et le respecter, est-ce si compliqué ?

Stefan : C'est plus que faire un choix, c'est prendre en compte tous les cas susceptibles d'apparaître.

Florian : Exact, c'est le point clé, comprendre tout ce dont nous avons besoin. D'un point de vue technique, il est difficile d'élaborer une solution adaptée à tous les cas possibles. Il est plus facile de se mettre d'accord sur, mettons, cinq cas, et d'écrire un standard pour eux.

Stefan : L'IdO va se développer de bas en haut. L'IdO ne sera pas fabriqué et mis sur la place publique du jour au lendemain, comme ce fut le cas de la télévision.

Infrastructure internet

Tessel : L'internet est la colonne vertébrale de l'IdO. Est-il assez solide ?

Stefan : Il nous faudra beaucoup plus de bande passante. Tout fonctionne pour l'instant, mais nous devons nous préparer à une avalanche de données. Leur croissance est exponentielle. Cinq milliards de gigaoctets de données ont été produits entre les débuts de la civilisation et le début de ce millénaire. En 2011, il n'aura fallu que deux jours pour créer le même volume, et en 2013 dix minutes suffiront.

La capacité du spectre radio est par essence limitée. Impossible d'ajouter



Florian Michahelles



Heico Sandee



Stefan Gessler



un nombre illimité de périphériques au réseau sans-fil.

Florian : Il nous faudra gérer la pénurie. Aujourd'hui, quand c'est engorgé, il faut un peu plus de temps pour télécharger, c'est tout. La plupart des données du trafic ne sont pas urgentes, mais cela pourrait être différent avec l'IdO. Pensez p. ex. à des applications médicales qui auraient besoin d'un accès permanent à certaines données. Gérer la pénurie, ce pourrait être donner la priorité à ces applications, ou faire payer un droit d'accès prioritaire. Je pense que nous serons toujours à la limite de la saturation de la bande passante, mais je ne pense pas que cela sera un obstacle à l'IdO.



Stefan : L'IdO pourrait contribuer à une gestion plus efficace du trafic. Il s'agit de prendre des décisions le plus tôt possible. Les données propres à un réseau local n'ont pas à être envoyées à un centre de données distant. Aujourd'hui nous distinguons le local et le nuage. Dans le futur nous pourrions avoir d'autres couches, différents niveaux de nuage. Les données seront organisées de façon hiérarchique, entre ce qui peut rester local et ce qui doit être accessible à un niveau régional ou mondial.

Dangers

Tessel : *Quels sont les dangers possibles de cette technologie ?*

Heico : La sécurité. Nous savons déjà ce que des pirates malveillants peuvent faire à l'internet. Les conséquences seraient encore plus grandes avec toutes sortes d'objets reliés au réseau.

Stefan : Chaque nouveauté peut aussi être utilisée à mauvais escient. Et chaque nouvelle technologie exerce une influence sur notre vie quotidienne. Beaucoup voient le changement comme une menace, donc nous devons aussi veiller à ce que l'IdO soit accepté par le public. Sécurité et respect de la vie privée seront sûrement des facteurs clés.

Florian : La transparence est un danger possible. En tant qu'humains nous comprenons de façon naturelle pourquoi il se passe telle ou telle chose. Avec cette composante numérique ajoutée à notre monde, nos sens pourraient perdre cette faculté de relier un fait à sa cause. Nous devons donc concevoir la technologie de façon à préserver ce modèle mental. Ne pas comprendre une situation fait peur, car cela signifie que vous être contrôlé



par quelque chose d'étranger. C'est un scénario noir, mais le plus sombre d'entre tous serait que l'internet des objets ne se réalise pas.

Elektor et l'IdO

Comme l'a justement souligné Stefan, l'IdO ne s'installera pas dans nos vies du jour au lendemain. Il s'agit plutôt d'une expansion illimitée de technologies et d'applications, dont beaucoup sont déjà exploitées. Elektor a même sans doute couvert plusieurs d'entre elles sans le savoir. Dans le seul numéro de novembre, j'ai compté cinq projets qui relèvent de l'IdO; aucun ne comporte pourtant de référence explicite à l'IdO.

Une des applications les plus célèbres de l'IdO est le compteur intelligent. Véritable terminaison nerveuse du réseau (électrique) intelligent, ces compteurs donnent la consommation à un niveau local et permettent au fournisseur et au consommateur de prendre des mesures raisonnées d'économies d'énergie. Le mois dernier, S. Mikkelsen a décrit page 48 l'installation d'un énergimètre S0 qui envoie les détails d'une consommation à une base de données locale ou en nuage.

Dans la série *Embarquez Linux !* de ce même numéro, B. Sauter a expliqué p. 68 comment installer une connexion réseau pour la carte Linux-Elektor. Et J. Nickel a présenté une bibliothèque C qui permet aux développeurs d'ajouter facilement des noeuds de microcontrôleurs à l'ElektorBus (p. 56).

N'oublions pas non plus J. Mertens, qui p. 36 revient aux fondements de l'internet en expliquant comment construire une carte NIC (*Network Interface Card*) avec un AVR à 8 bits. Connecté au port parallèle d'une caméra, le µC permet d'envoyer des vidéos à plusieurs destinataires d'un réseau. Et p. 28 dans l'article *Des périphériques maison pour votre téléphone Android*, D. Cuartielles et A. Goransson montraient comment poser les fondations de votre propre IdO. D'ailleurs, ils vous attendent pour la suite ! Avez-vous remarqué que dans ce numéro-ci, l'IdO est présent aussi en filigrane ?

Nous avons entre les mains les briques d'un édifice en cours de construction ?

(120632 – version française : Hervé Moreau)

Liens

- [1] www.autoidlabs.org
- [2] www.roboearth.org
- [3] <http://iot-forum.eu>



Jackaltac
com®

**PCBs
Muuuuch Cheaper...**

PCBs starting from **17.22 EURO***

* incl. VAT (23%), add delivery costs
(Example Germany) of 10.89 EURO,
min. number of ordered pcbs:
5 @ 100 mm x 100 mm

No-frills policy

www.jackaltac.com



Elektor Electronic Toolbox

**Disponible sur
App Store**

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

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

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



**MÉMOIRE PROFONDE
OSCILLOSCOPES
SIGNAUX MIXTES**
NOUVEAUTÉ Pico Technology



PORTABILITÉ & PERFORMANCES

pico
Technology

| PicoScope | 3204 MSO | 3205 MSO | 3206 MSO |
|-----------------------|-------------------------------|----------|----------|
| Canaux | 2 Analogique 16 Numérique | | |
| Bande passante | 60 MHz | 100 MHz | 200 MHz |
| Mémoire tampon | 8 Mé | 32 Mé | 128 Mé |
| Résolution (amélioré) | 8 bits (12 bits) | | |
| Générateur de signal | Générateur de fonctions + AWG | | |
| Prix | €785 | €1028 | €1270 |

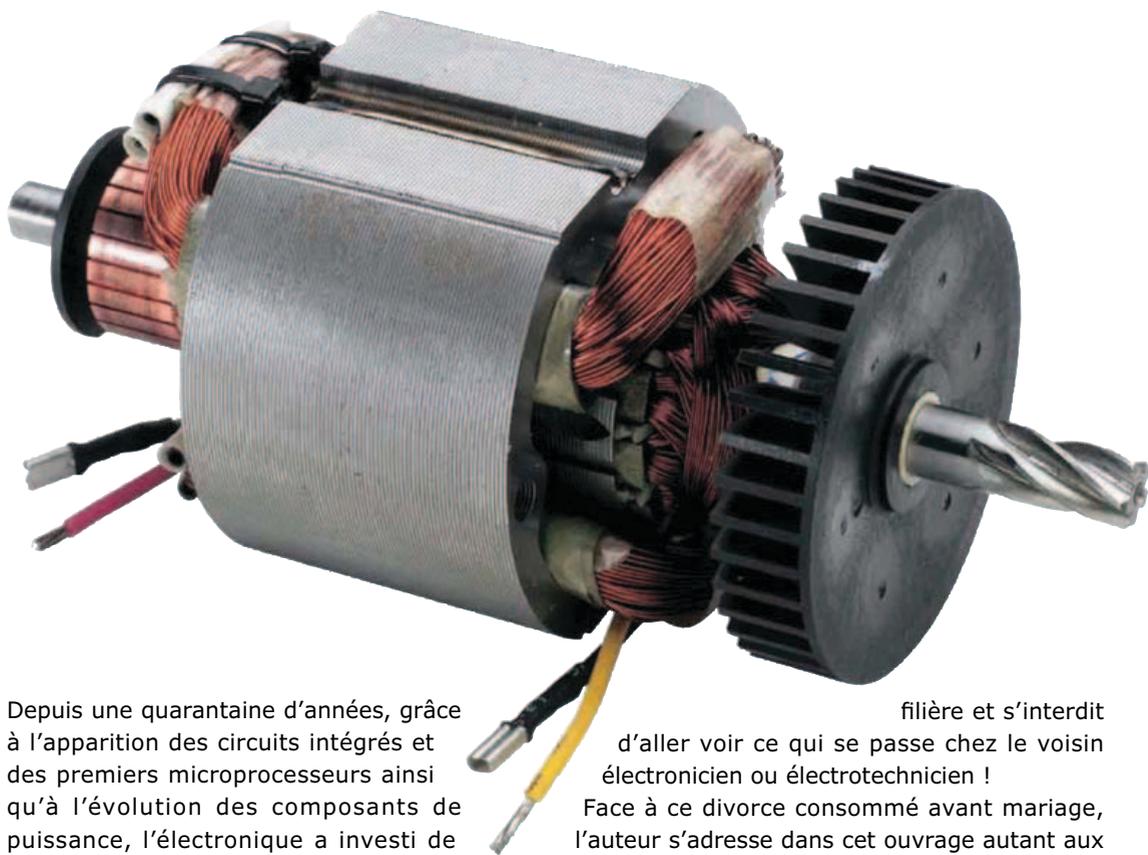
TOUS LES MODÈLES ONT DÉCLENCHEMENTS NUMÉRIQUES AVANCÉS, DÉCODAGE SÉRIEL (PC, SPI, RS232, CAN, LIN, FLEXRAY), TESTS DE LIMITE DE MASQUE, MÉMOIRE SEGMENTÉE, FILTRAGE NUMÉRIQUE, MISES À JOUR DU LOGICIEL GRATUITES ET 5 ANS DE GARANTIE

www.USBmso.com/PS174

mettez le doigt dans l'engrenage de l'électronique de puissance !

électrotechnique de puissance : démarrage, variation de vitesse, freinage

Gérard Guihéneuf



Tous les types de moteurs abordés

- Moteurs à courant continu
- Moteurs à courant alternatif monophasé (asynchrone, universel)
- Moteurs à courant alternatif triphasé (asynchrone)

Depuis une quarantaine d'années, grâce à l'apparition des circuits intégrés et des premiers microprocesseurs ainsi qu'à l'évolution des composants de puissance, l'électronique a investi de nouveaux domaines techniques comme l'électrotechnique : les hacheurs de puissance, les démarreurs-ralentisseurs progressifs et les variateurs de vitesse associés à des moteurs électriques à courant continu ou alternatif de plusieurs dizaines de kilowatts se sont démocratisés.

Curieusement, alors que l'électronique et l'électrotechnique fusionnent par le biais d'applications industrielles ou domestiques qui mettent en œuvre des moteurs électriques, les deux technologies demeurent des filières distinctes au sein de l'Éducation Nationale. Ainsi chacun (enseignant ou élève) suit sa voie dans sa

filière et s'interdit d'aller voir ce qui se passe chez le voisin électronique ou électrotechnicien !

Face à ce divorce consommé avant mariage, l'auteur s'adresse dans cet ouvrage autant aux électroniciens qu'aux électrotechniciens. Les trois chapitres fournissent une information vulgarisée et accessible (constitution, fonctionnement, caractéristiques, domaines d'utilisation des différents types de moteurs) mais aboutissent également à des réalisations électroniques simples et concrètes : les électroniciens découvriront le monde des moteurs électriques tandis que les électrotechniciens admettront que l'électronique n'est pas obligatoirement synonyme de complexité.

Trois chapitres

Le premier chapitre détaille en particulier les deux principes de variation de la vitesse des moteurs

Réalisations concrètes

Continu : variateur de vitesse pour mini-perceuse, variateur de vitesse pour train miniature

Monophasé : démarreur à contacteur statique à deux points de commande, variateur de vitesse pour moteur universel

Triphasé : démarreur inverseur statique

à courant continu : conversion alternatif/continu (redresseurs commandés par des thyristors) et conversion continu/continu (hacheurs à transistors IGBT). Les réalisations proposées permettent de mettre en pratique la théorie abordée : variateur pour mini-perceuse, variateur de vitesse pour train miniature.

Les moteurs du deuxième chapitre rapprochent le lecteur de l'électronique de puissance. L'auteur y passe en revue les différents moteurs à alimentation alternative monophasée : à induction, à bagues de déphasage, universel. La puissance des réalisations proposées s'exprime maintenant en kilowatts : démarreur à contacteur statique à deux points de commande pour moteur asynchrone monophasé à induction, variateur de vitesse pour moteur universel.

les moteurs électriques expliqués aux électroniciens

Le troisième chapitre s'intéresse au moteur électrique le plus utilisé dans l'industrie : le moteur asynchrone triphasé. Les différents principes de démarrage, de variation de vitesse et de freinage sont abordés : démarreur électromécanique à contacteurs, démarreur-ralentisseur, convertisseur de fréquence ou couplage des pôles pour la variation de vitesse, moteur frein, freinage par injection de courant...

Le lecteur qui réalisera le démarreur inverseur statique pour moteur asynchrone triphasé aura un exemple concret de la proximité entre électronique

3.11 Démarreurs-ralentisseurs progressifs électroniques

Principe de fonctionnement d'un démarreur progressif

Figure 3-40 - Pour que la tension fournie au moteur augmente progressivement pendant la durée du démarrage, l'angle d'amorçage α des thyristors décroît jusqu'à être nul pour que le moteur soit finalement alimenté sous sa tension nominale.

3.11.2 Démarrage et ralentissement progressifs avec le démarreur Altistart 01 (SCHNEIDER)

La figure 3-41 montre le schéma qui permet la commande marche-arrêt par deux boutons-poussoirs d'un moteur asynchrone triphasé alimenté par un démarreur-ralentisseur progressif Altistart 01. Décrit par des chronogrammes explicatifs, le fonctionnement de l'ensemble démarreur-moteur obéit à une procédure précise. Dans un premier temps, l'enclenchement du disjoncteur de moteur Q1 alimente le démarreur sans provoquer le moindre mouvement du moteur. Seule une diode électroluminescente verte située en face avant de l'appareil fait état de cette mise sous tension. Ensuite, il faut appuyer brièvement sur le bouton-poussoir S2 pour que l'angle d'amorçage α des thyristors décroisse jusqu'à être nul pour que le moteur soit finalement alimenté sous sa tension nominale.

2.2 Moteur asynchrone monophasé à induction

Tensions d'alimentation d'un moteur monophasé

Évolution du champ tournant dans le moteur sur une période :

Figure 2-06 - Le décalage dans l'espace des enroulements et le déphasage de leurs tensions d'alimentation induisent dans le moteur un champ magnétique tournant appelé vitesse de synchronisme du moteur.

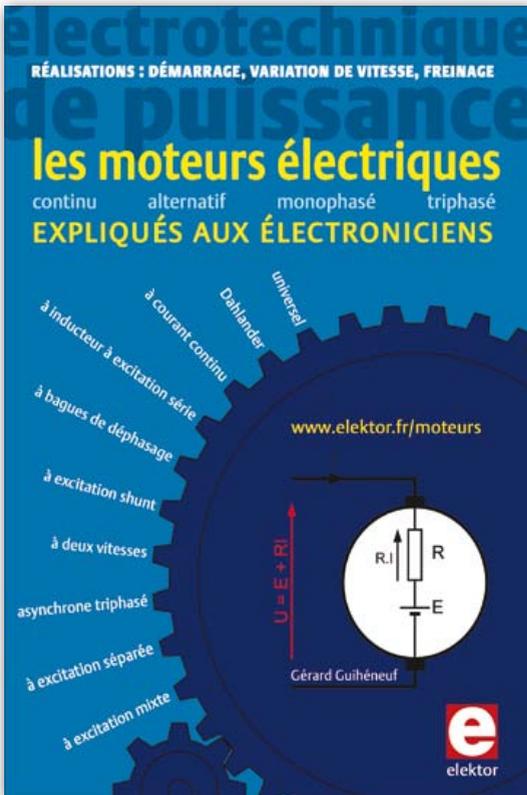
par rapport à ceux créés par l'enroulement PP et décalés géométriquement dans le moteur à cause de la position des bobinages. Désormais, à la lecture de la période décrite à la figure 2-06, il est possible d'affirmer que l'évolution des pôles magnétiques produits par les deux enroulements PP et PA décrit un champ magnétique qui tourne dans le sens antihoraire (exemple décrit par la figure 2-06).

Ce seul constat ne suffit pas à expliquer la rotation du rotor car celui-ci de par sa constitution n'est pas un aimant. Soumis à un champ magnétique tournant et variable, les conducteurs du rotor sont le siège de courants induits. Comme pour le moteur à courant continu, les conducteurs du rotor parcourus par ces courants induits et exposés à un champ magnétique subissent une force qui met le rotor en mouvement. Le sens de rotation du rotor obéit à la loi de Lenz selon laquelle les courants induits visent à annuler la cause qui leur a donné naissance. Il en résulte une rotation du rotor dans le même sens que celui du champ magnétique tournant. Cependant, le rotor tourne à une vitesse moindre que celle du champ tournant et

105

et électrotechnique : des portes logiques NON-OU alimentées sous 12 V commandent le sens de rotation d'un moteur de 1,5 kW alimenté sous une tension triphasée 3 × 400 V.

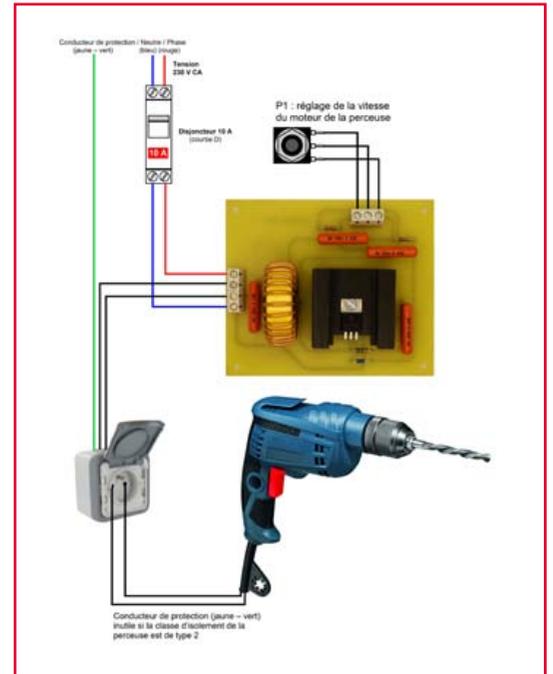
Enfin, le lecteur pourra mesurer l'évolution de ses connaissances grâce à de judicieux **questionnaires d'évaluation intermédiaire**, intégrés à chaque chapitre (corrigés à télécharger sur www.elektor.fr/moteurs).



Le sommaire ainsi que des extraits du livre sont disponibles gratuitement en ligne : www.elektor.fr/moteurs

Ils vous donneront une idée plus précise du contenu de ce livre.

120671



Ce nouveau livre, *Les moteurs électriques expliqués aux électroniciens* (ISBN 978-2-86661-188-0), ainsi que les autres livres de **Gérard Guihéneuf** sont disponibles sur le site www.elektor.fr/livres

A propos de l'auteur

Gérard Guihéneuf est titulaire d'un DUT Électrotechnique. Il enseigne dans un lycée professionnel nantais (BEP Métiers de l'Électrotechnique et Bac Pro Électrotechnique Énergie et Équipements Communicants). Passionné d'électronique et d'informatique appliquées, il a publié des articles dans des revues de vulgarisation ainsi que plusieurs ouvrages chez Elektor.

Autres titres de Gérard Guihéneuf publiés chez Elektor :

- **Comprendre et dimensionner les installations domestiques à énergies renouvelables**
ISBN 978-2-86661-170-5 | www.elektor.fr/nrj
Dossiers pédagogiques disponibles en ligne
- **Photovoltaïque - alimentation électrique autonome de sites isolés**
Réalisez vous-même l'électrification d'un abri de jardin, d'un garage ou d'un mobile home
ISBN 978-2-86661-181-1 | www.elektor.fr/nrjsol
- **Eau de pluie : récupération & utilisation**
Montages électroniques PICBASIC pour le recyclage des eaux de pluie
ISBN 978-2-86661-176-7 | www.elektor.fr/eau
- **Réalisez et programmez 12 applications pratiques pour maîtriser le PICBASIC PB-3B**
ISBN 978-2-86661-166-8 | www.elektor.fr/picbasic



Nouvelle édition revue et augmentée par l'auteur



ISBN 978-2-86661-189-7
format: 17 x 23,5 cm
240 pages

44,50€

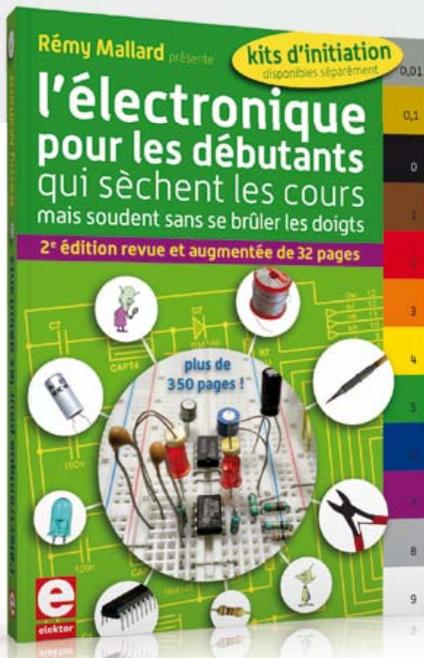
Ce livre est écrit par un spécialiste français de l'automatisation de l'éclairage de scène. Passionné à la fois de théâtre, de musique, de scène ET d'électronique, Benoît Bouchez consacre plus de la moitié de son ouvrage à la commande à distance, à l'automatisation, à l'utilisation des protocoles modernes tels DMX512 ou MIDI. Il en donne les éléments théoriques indispensables, puis les met aussitôt en pratique, à la portée de lecteurs désireux de comprendre et d'agir sur leurs installations.

Il présente également des protocoles encore plus récents comme RDM, ArtNet ou ACN. Conformément à la tradition des publications Elektor, les schémas des circuits électroniques sont dévoilés dans le livre, et dûment expliqués, les dessins des circuits imprimés peuvent être reproduits aisément. En outre le lecteur trouvera en téléchargement des programmes avec lesquels il pourra lui-même commander, tester, évaluer.

Le métier ne s'apprend qu'au contact de spécialistes, sur le terrain. L'expérience et le doigté ne se stockent pas dans des fichiers et ne s'impriment pas sur papier, mais ce livre contribuera à développer les talents d'éclairagiste de ceux qui le liront, amateurs ou professionnels. Il sera leur compagnon sur le chemin plein d'imprévus des innombrables possibilités offertes par les techniques numériques de commande de lumière.



Logiciel, matériel et informations complémentaires sur www.elektor.fr/dmx



ISBN 978-2-86661-186-6 - 352 pages - **42,50 €**

Kit n°1 : sirène - réf. 119016-71 - **22,50 €**

Kit n°2 : chenillard & thermomètre - réf. : 119016-72 - **22,50 €**

nouvelle édition revue et augmentée du livre l'électronique pour les débutants

deux kits d'initiation disponibles

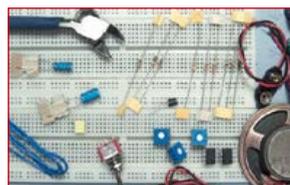
Voici le cadeau idéal pour partager votre passion de l'électronique avec vos enfants, petits-enfants, neveux... et autres geeks

Fin pédagogue, Rémy Mallard écrit pour les débutants dans un style inédit, et répond d'abord aux questions prosaïques du néophyte : quel fer à souder acheter ? Un multimètre à 5 € peut-il suffire ? Et bien d'autres interrogations trop souvent laissées en suspens.

L'auteur démystifie l'électronique en n'utilisant que ce qu'il vous faut de théorie pour aborder la pratique : identifier les composants et leur rôle, les récupérer, les tester et les ranger ; lire un schéma ; choisir ses outils ; mettre en boîte ses montages..

Les deux kits disponibles séparément permettent de réaliser, sur une plaque d'expérimentation sans soudure, quelques-uns des montages simples et ludiques présentés dans le livre.

nouveau kit !



Offre spéciale : livre + deux kits = **81,50 €** au lieu de 87,50 €
Informations complémentaires et commande :
www.elektor.fr/debut

HP-35 : révolution de poche (1972)

Il y a quarante ans, Hewlett-Packard sortait de sa poche la première calculatrice scientifique !



1

Avec la HP-35, Hewlett-Packard a présenté la première calculatrice scientifique de poche au monde. Les calculs laborieux et imprécis à grand renfort de règle à calcul et de table de logarithmes, tout d'un coup, ont pris fin. La HP-35 affichait racines carrées, fonctions trigonométriques et logarithmes d'une pression sur une touche — avec une précision incroyable de dix chiffres.

La règle à calcul est morte, vive la HP-35 !

« Et maintenant, il nous faut une machine d'un dixième de la taille, un dixième du prix, et dix fois plus rapide ! » [2] Telle fut, en substance, la réaction de Bill Hewlett à une démonstration interne de la machine à calculer de table HP-900A — un monstre de quelque 18 kg —, à prendre comme des félicitations à l'équipe

de concepteurs !

C'était en 1968. Avec la HP-900A, Hewlett-Packard défrichait une terre inconnue et élargissait le marché des machines à calculer électroniques. Toutefois, au prix de 4 900 dollars à l'époque (ce qui correspondrait à 33 000 dollars d'aujourd'hui [1]), le calcul électronique restait un rêve pour beaucoup d'ingénieurs et la règle à calcul leur outil principal.

Les choses changèrent environ quatre ans plus tard, le 1er février 1972, quand Hewlett-Packard fut en mesure, grâce à la technique Low Power Integrated Circuit [2] développée par Mostek, de présenter au monde la première calculatrice scientifique de poche. En dix-huit mois seulement, Tom Osborne et son équipe avaient réussi à satisfaire la demande de Bill Hewlett : une calculatrice technique et scientifique qui tient dans sa poche de poitrine. Même si seuls les ingénieurs

de grandes entreprises d'alors, capables d'aligner 395 dollars (environ 2600 dollars d'aujourd'hui) pour une calculatrice, pouvaient profiter d'une HP-35, cette innovation signalait l'arrêt de mort de la règle à calcul.

Matériel

La figure 1 montre cet appareil de 14,7 x 8,1 x 3,3 cm et 246,6 g. Sous l'afficheur à une ligne de 15 chiffres à 7 segments se trouvent un interrupteur marche-arrêt et un clavier à 35 touches. Ce nombre de touches, suivant le vœu de Bill Hewlett-Packard, a valu son nom au modèle.

L'afficheur à sept segments comporte devant chaque chiffre une petite loupe qui facilite la lecture tout en permettant de caser quinze chiffres dans la largeur disponible. Les afficheurs à LED, surtout alors, ne sont pas renommés pour leur frugalité, aussi la HP-35 est-elle équipée d'un bloc de trois accus au format AA dans un boîtier en plastique (figure 2). Les pôles plus et moins sont accessibles par deux plages dissymétriques sur le côté, ce qui évite les inversions de polarité. La calculatrice est livrée avec un étui ad hoc, un chargeur et un mode d'emploi — le tout dans un coffret solide et plutôt imposant. Malheureusement, les deux derniers articles manquent à ma collection (figure 3).

La figure 4 dévoile les entrailles de la HP-35, platine principale à droite, clavier et afficheur à gauche. La figure 5 représente le schéma de principe. La platine principale, de 7,0 x 7,5 cm, comporte cinq circuits intégrés MOS/LSI, dont trois ROM qui contiennent le logiciel. Les deux autres circuits intégrés hébergent les registres, l'unité arithmétique et le circuit séquenceur-temporisateur. La deuxième platine porte le clavier, l'afficheur et leur électronique de pilotage.

L'unité registres-logique se compose de cinq blocs : tampon des commandes et décodage, circuit temporisateur, sept registres de 56 bits, un additionneur-soustracteur et le décodeur d'affichage [5].

Le circuit de séquence-temporisation se charge

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

La HP-35 a été lancée à une époque passionnante. C'est en 1972 que commencent avec Apollo 16 et Apollo 17 les vols habités de la NASA vers la lune ; Mariner 9 envoie des images de Mars ; Pioneer 10 commence à étudier la ceinture d'astéroïdes de Jupiter. En septembre s'ouvre le pont routier d'Öland en Suède, un des plus longs d'Europe [3]. Et naturellement les percées technologiques des circuits intégrés à grande échelle, au début des années soixante-dix, ont donné une impulsion décisive au développement des calculatrices.

Années septante, années pétantes — et vous, que faisiez-vous pendant ce temps-là ?

du déroulement des processus internes. Il s'agit de la scrutation du clavier, de la synchronisation du système et du calcul des adresses des instructions [5].

Les trois puces de ROM contiennent des routines pré-programmées pour le calcul des fonctions mathématiques. Chaque ROM contient 256 instructions de 10 bits chacune. En plus, chaque ROM contient un décodeur. Il n'y a en tout qu'un espace mémoire de 7,5 Kbits, moins d'un kilooctet (!), pour caser la totalité du logiciel. Ce petit article de quelque douze cents caractères (octets) occupe plus de mémoire de la HP-35, infiniment plus (38 Koctets) si on prend le fichier de traitement de texte, sans les illustrations, bien sûr ! Comment Tom Osborne et son équipe ont-ils réussi à bourrer toute cette logique dans si peu d'espace ? C'est ce que nous allons voir.

Logiciel

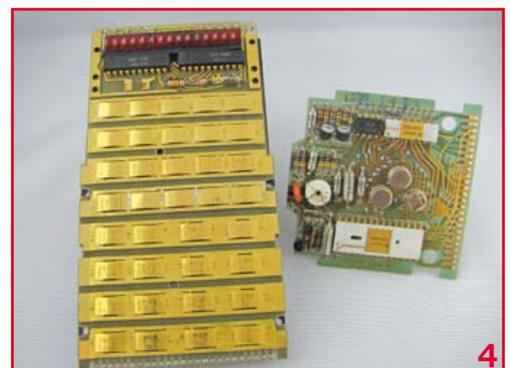
Deux points étaient essentiels pour la réalisation de ce logiciel avec ces ressources minimales en mémoire : l'adoption de la notation polonaise inverse (Reverse Polish Notation) pour la saisie et la mise en œuvre de l'algorithme CORDIC.

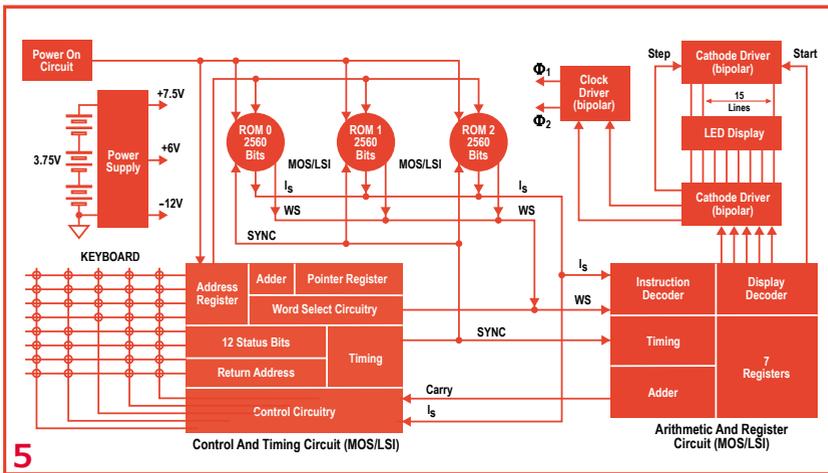
La notation polonaise inverse découle des travaux du mathématicien polonais Jan Lukasiewicz, qui a mis au point dans les années vingt (du vingtième siècle) la notation polonaise, un mode d'écriture sans parenthèses de la logique assertorique [6]. Dans ce mode d'écriture, on note d'abord les opérateurs, ensuite les opérands. C'est exactement l'inverse dans la HP-35 : on pousse les opérands sur une pile à quatre niveaux avec la touche Enter, deux fois plus large, puis on actionne la touche de l'opérateur ou des opérateurs souhaité(s). C'est pourquoi vous chercherez en vain la touche « égale » et les parenthèses ; elles sont inutiles en notation polonaise inverse. Dans les années soixante-dix, et en particulier après le lancement de la HP-65, première calculatrice programmable de poche, le débat sur les mérites respectifs de la notation polonaise inverse

(RPN) et de la notation algébrique (Algebraic Operating System) concurrente de Texas Instruments tourna à la guerre de religion, comme plus tard les querelles sur les langages de programmation BASIC, Pascal, puis C etc. Si vous cherchez une vue d'ensemble sur ce mode de saisie élégant, vous trouverez une foule de sources sur l'internet, par exemple [7].

La notation polonaise inverse permet deux sortes d'économies : elle dispense des touches « égale » et parenthèses et elle simplifie l'évaluation des expressions arithmétiques. Pour l'utilisateur, elle supprime la limitation du nombre de parenthèses imbriquées. En pratique, il faut tenir compte de la taille finie de la pile.

La deuxième caractéristique essentielle du logiciel de la HP-35 est la mise en œuvre des algorithmes CORDIC. Ce nom est l'acronyme de COordinate Rotation Digital Computer. Il s'agit d'une classe d'algorithmes itératifs puissants pour le calcul de fonctions trigonométriques et mathématiques, mis au point par Jack E. Volder en 1958 [8]. Volder réussit à concevoir, pour remplacer le calculateur de navigation du B-58, peu fiable, un calculateur numérique de position en temps réel. La caractéristique « temps réel » prend tout son sens si on sait que le B-58 était un avion supersonique.





Tom Osborne avait déjà mis en œuvre l’algorithme CORDIC avec succès dans la machine à calculer de table HP-9100A. C’est ainsi qu’il avait acquis de l’expérience dans l’exécution de divisions, multiplications et fonctions trigonométriques avec seulement des opérations d’addition et de décalage. La précision obtenue était comparable à celle des machines actuelles, mais il se présente facilement des écarts sur le dernier chiffre. Par exemple, la HP-35 donne pour $43 : 63,99999997$ au lieu de $64,00000000$.

Les bogues de la HP-35

Malgré tout le soin apporté à la programmation, plusieurs bogues se sont fauflés, pour être découverts seulement quand 25 000 exemplaires avaient déjà été vendus [8]. Par exemple, elle donnait $\ln(2,02) = 2$ au lieu du résultat correct $2,02$; $\arctan(0,0002) = 5,72957789310-3$ au lieu de $0,01145916$ [10]. Situation difficile, parce que faute d’internet et de mémoire flash, il n’était pas possible de publier une mise à jour, comme on le fait couramment aujourd’hui pour les calculettes d’un certain prix. Dans une réunion de crise, Dave Packard posa la question : « que deviennent les machines déjà vendues ? » Naturellement on écarta la réponse — devenue célèbre — d’un collaborateur : « motus » [9].

Au lieu de cela, tous les clients reçurent une lettre d’information et une proposition de réparation gratuite (voir [11] page 6, figures 5 et 6). On a dit que seulement un quart des machines en cause sont revenues pour réparation : la plupart des clients ont gardé leur machine avec la lettre originale et beaucoup ont commandé une HP-35 neuve avec le logiciel corrigé.

La marche inexorable des calculatrices de poche

La HP-35 a été un des plus grands succès de Hewlett-Packard, qui a reçu le 14 avril 2009 le prix très convoité IEEE Milestone in Electrical Engineering and Computing [12]. Le modèle suivant HP-45 fut proposé juste un an plus tard, et en 1974 apparut la HP-65, première calculatrice de poche programmable du monde, qui pouvait enregistrer ses programmes sur une carte magnétique. Au total, HP a développé plus de cent modèles, dont certains ont été utilisés par des gagnants de Prix Nobel. Bien que la vogue des calculatrices électroniques se soit estompée avec la généralisation des ordinateurs personnels, le développement continue avec de nouvelles machines avec des possibilités graphiques et algébriques. À côté du domaine scientifique, les financiers ont aussi bénéficié de calculatrices spécifiques pour leurs besoins : par exemple la Business Calculator HP-80 lancée en 1973. La calculatrice financière HP-12C a valu à Hewlett-Packard un autre record : présentée en 1981, elle est encore produite aujourd’hui, plus de trente ans !

(120454 – version française : Jean-Paul Brodier)

Sources et références

- [1] ‘What is a dollar worth?’ <http://www.minneapolisfed.org/>
- [2] Malone, Michael S.: ‘Bill & Dave’, Portfolio (Penguin Group) New York, 2007
- [3] <http://en.wikipedia.org/wiki/1972>
- [4] http://en.wikipedia.org/wiki/Integrated_circuit
- [5] HP Journal June 1971, page 2 et suivantes : <http://www.hpl.hp.com/hpjournal/pdfs/IssuePDFs/1972-06.pdf>
- [6] http://fr.wikipedia.org/wiki/Notation_polonaise_inverse
- [7] <http://h41111.www4.hp.com/calculators/uk/en/articles/rpn.html>
- [8] <http://fr.wikipedia.org/wiki/CORDIC>
- [9] <http://www.hpmuseum.org/hp35.htm>
- [10] <http://www.jacques-laporte.org/HP35%20ROM.htm>
- [11] <http://hhuc.us/2007/Remembering%20The%20HP35A.pdf>
- [12] http://www.ieee.org/about/news/2009/14april_1.html

Hexadoku casse-tête pour électroniciens

Ce jeu, c'est un peu comme le chocolat : on commence par un carré, ça fait saliver et finalement toute la tablette y passe. À cette différence près que notre Hexadoku ne cause pas de crise de foie, ne gêne pas les dents et ne fait pas de tâches quand on bave. Oubliez qu'il fait froid dehors, qu'il y a une poubelle à sortir, de la glace à racler sur le pare-brise. Détendez-vous ! Remplissez la grille selon les règles, envoyez-nous votre solution, et vous gagnerez peut-être l'un des quatre chèques-cadeaux Elektor mis en jeu. Après ça, vous serez d'aplomb pour reprendre vos cogitations électroniques.

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 envoyer la série de chiffres sur fond grisé.

Participez et gagnez !

Nous tirerons au sort l'une des réponses internationales correctes qui nous seront parvenues dans les délais ; son auteur recevra un chèque-cadeau Elektor d'une valeur de **100 €**. Nous offrons en outre 3 chèques-cadeaux Elektor d'une valeur de **50 €** chacun. À vos crayons !

Où envoyer ?

Envoyez votre réponse (les chiffres sur fond grisé) avec vos coordonnées par courriel, télécopie ou courrier avant le **1^{er} janvier 2013** :
Elektor c/o Regus Roissy CDG – Le Dôme – 1, rue de La Haye
BP 12910 – 95731 Roissy CDG
Courriel : hexadoku@elektor.fr

La solution de la grille du numéro de septembre 411 est **75E2B**

Le gagnant du **chèque-cadeau Elektor** d'une valeur de **100 €** est Steve Hasko, de Huntingdon, en Grande-Bretagne.

Les **3 chèques-cadeaux Elektor** d'une valeur de **50 €** chacun vont à : Michael Franke, d'Eching (Allemagne),

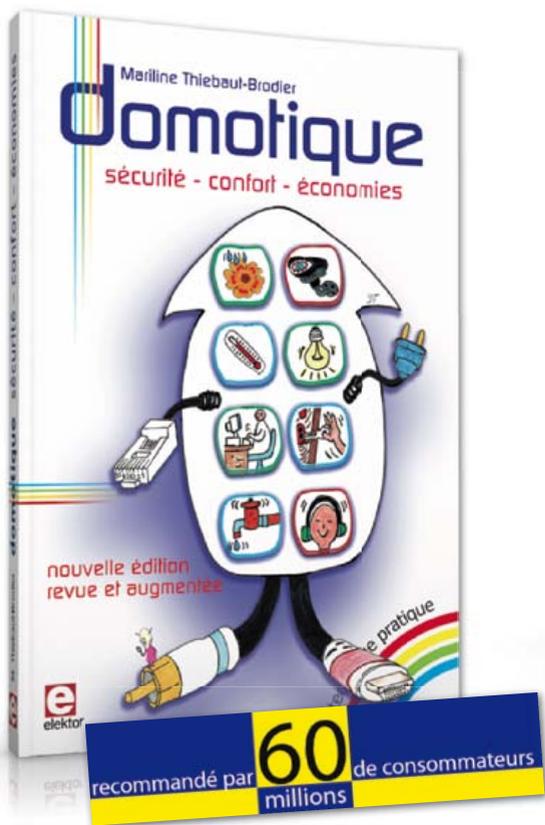
Jean-Paul Lagaisse, de Bavikhove (Belgique) et Franz Maurer, de Nimègue (Pays-Bas).

Bravo à tous et félicitations aux gagnants !

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

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



Conseillé par « 60 millions de consommateurs » dans son hors-série 1115 de novembre-décembre 2012

1 Domotique

La domotique, c'est l'électronique et l'informatique appliquées au logement. Elle améliore la vie quotidienne au moyen de dispositifs électriques et électroniques. Il peut s'agir aussi bien de motoriser et télécommander par exemple la porte du garage ou bien les volets, que réguler le chauffage, programmer les appareils électroménagers, simuler une présence, commander l'alarme à distance, arroser automatiquement, exploiter un réseau multimédia, etc.

Ce livre montre comment la domotique gère le fonctionnement des appareils et dispositifs électriques de la maison.

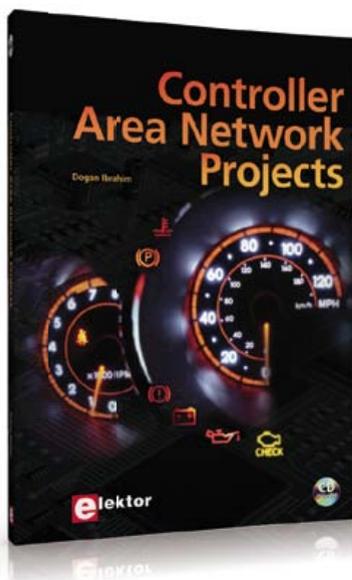
256 pages • ISBN 978-2-86661-182-8 • 33,00 €

livre avec CD-ROM

2 Controller Area Network Projects

Ce livre (en anglais) s'adresse à tous ceux qui souhaitent en savoir (beaucoup) plus sur le bus CAN et ses applications : étudiants, ingénieurs en exercice, ou amateurs motivés.

Il explique les principes de base des réseaux CAN et montre comment élaborer des projets avec des microcontrôleurs sur le bus CAN.



Pour en profiter pleinement, il faut quelques connaissances de base en électronique. La compréhension du langage de programmation C est utile dans les derniers chapitres du livre ; connaître au moins un microcontrôleur de la série PIC sera un avantage, notamment quand vous passerez à la réalisation de projets à base de microcontrôleurs qui utilisent le bus CAN.

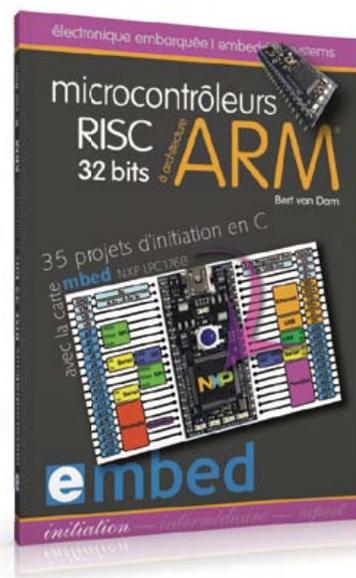
259 pages • ISBN 978-1-907920-04-2 – 34,50 €

35 projets d'initiation en C avec la carte mbed NXP LPC 17688

3 Microcontrôleurs RISC 32 bits à architecture ARM

La plate-forme mbed et son microcontrôleur ARM, le NXP LPC1768, sont conçus pour l'informatique en nuage ou cloud computing qui révolutionne le développement de logiciel : aucune installation de logiciel spécifique, il suffit d'un navigateur et d'un port USB. Vous programmerez et stockerez vos résultats sur l'internet, et y accéderez depuis n'importe quel PC, où que vous vous trouviez. Dans ce livre, il est question aussi bien du langage C, des bibliothèques mbed, d'exemples de programmes que du traitement de signaux analogiques, de capteurs, de moteurs etc.

232 pages • ISBN 978-2-86661-178-1 • 40,00 €



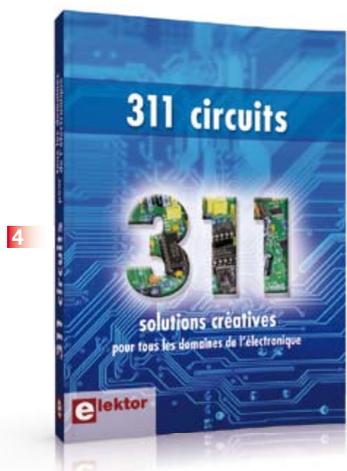
Solutions créatives pour tous les domaines de l'électronique

4 311 circuits

Cet ouvrage est un trésor : il réunit 311 schémas d'électronique analogique, logique ou numérique, des programmes, des liens vers des sites internet, des tableaux de caractéristiques de composants et des dessins de circuit imprimés. Il est le onzième volume de la collection « 300 circuits » (301... 302... 303... 304... 305... 306... 307... 308... 309... 310... 311 circuits). Ses deux tables des matières alphabétique et thématique vous permettent de trouver rapidement et facilement parmi les 311 articles pro-posés ceux qui répondront à vos besoins. Ces articles viennent des numéros doubles récents de la revue Elektor, publiés chaque année en été, et appelés numéros Hors-Gabarit, par allusion à leur contenu exceptionnellement riche. Ils forment un véritable catalogue d'idées, de trouvailles et d'astuces. C'est une source d'inspiration inépuisable, et à partir de laquelle chacun élaborera ses propres variantes qu'il combinera ensuite à sa guise avec d'autres circuits. Tous les domaines familiers et usuels de l'électronique sont abordés : alimentations, régulateurs et chargeurs • audio & vidéo • communication • hautes fréquences • informatique • jeux & modélisme • maison & automobile • mesure & test • processeur & contrôleur • robots et leurs accessoires.

448 pages • ISBN 978-2-86661-184-2 • 36,00 €

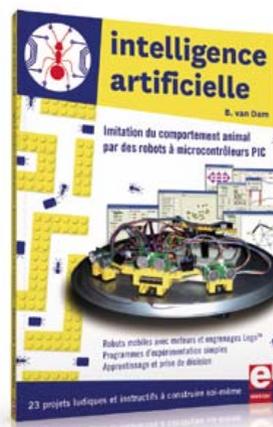
livres, CD & DVD-ROM, kits & modules



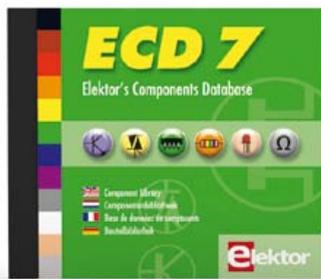
4



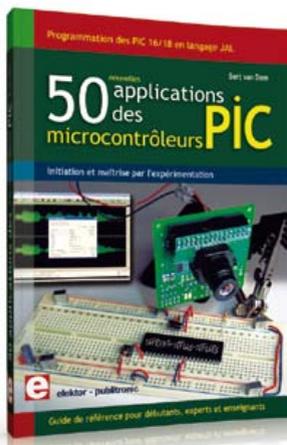
6



7



5



8



9

Base de composants interactive d'Elektor

5 ECD 7

Une quadruple banque de données (circuits intégrés, transistors, diodes et optocoupleurs) complétée par neuf applications satellites, au nombre desquelles on trouvera notamment de quoi calculer la valeur de la résistance associée à une diode zener, à un régulateur, à un diviseur, ou un multivibrateur astable, mais aussi le code de couleur de la résistance et de l'inductance. Le clou, c'est que vous allez pouvoir rajouter dans la base de données ce qui y manque encore, car elle est interactive ! Ainsi chaque utilisateur pourra lui-même rajouter des composants, en modifier les caractéristiques déjà enregistrées ou les compléter.

ISBN 978-90-5381-298-3 • 29,50 €

Une carte compacte et bon marché qui vous initiera tout en douceur !

6 Embarquez Linux !

Linux est partout, même dans certaines machines à café. Souvent, l'électronicien tenté d'adopter ce système d'exploitation est arrêté par sa complexité et par le prix des cartes de développement. Voici Linux pour les électroniciens, sous la forme d'une carte compacte et bon marché qui vous initiera tout en douceur !

Carte Linux Elektor (montée et testée)

Réf. : 120026-91 • 64,95 €

23 projets ludiques et instructifs à construire soi-même

7 Intelligence artificielle

Ce livre ne traite pas de théories abstraites, mais de pratique. Il s'adresse à vous, passionné d'électronique et de micro-informatique appliquée, et propose des circuits et des programmes simples. Expérimentez le biomimétisme sur vos propres robots mobiles, construits avec des pièces de Lego ! Il y est donc question de microcontrôleurs PIC, de programmation, de capteurs, de moteurs... mais aussi de mormions, de fourmis, de gnous, de vers, et d'autres bestioles « naturellement géniales ».

238 pages • ISBN 978-2-86661-179-8 • 43,50 €

Initiation et maîtrise par l'expérimentation

8 50 applications des microcontrôleurs PIC

Voici 50 projets instructifs et utiles pour vous initier au langage de programmation JAL et maîtriser les microcontrôleurs PIC16 et 18, avec des techniques universelles comme la commande de relais, ou le traitement des signaux émis par divers capteurs (y compris par exemple un codeur rotatif), la communication avec les bus I²C, SPI, RS232, USB, les afficheurs à 7 segments et même le bus CAN. C'est un ouvrage récréatif et pédagogique : assemblez et utilisez les projets proposés. Les explications claires, les schémas et les photographies vous

feront découvrir une activité enrichissante.

394 pages • ISBN 978-2-86661-177-4 • 45,00 €

Kit (circuits imprimés & composants)

9 TAPIR – Détecteur ultrasensible d'électromagnétique

Fin limier de la pollution électromagnétique qu'il traque et rend audible sur casque, le TAPIR est (aussi) un beau projet à construire, un kit avec tout ce qu'il faut, même le boîtier, qui est fait de quatre circuits imprimés ingénieusement assemblés. Le TAPIR, sigle de Totally Archaic but Practical Interceptor of Radiation (totalement archaïque mais pratique intercepteur de radiations), détecte et localise (en la faisant entendre) toute source de champ électrique E ou, avec l'antenne appropriée, de champ magnétique H.

Réf. : 120354-71 • 14,95 €

Informations complémentaires et gamme complète sur :

www.elektor.fr/e-choppe

Elektor/Publitronic SARL

1, rue de la Haye – BP 12910 – 95731 Roissy CDG Cedex

Tél. : +33(0)1.49.19.26.19 – Fax : +33(0)1.49.19.22.37

@ : ventes@elektor.fr

Composants

Raymond Vermeulen (Elektor)

régulation de ventilateur en fonction de la température

Si comme moi vous avez un faible pour les ventilateurs, vous apprécierez ces deux circuits intégrés qui assurent une régulation autonome de ventilation. Ils viennent à point nommé si vous avez un système gourmand en énergie et enfermé dans un coffret, ce qui impose une ventilation mécanique. Si vous voulez asservir la ventilation à la température sans charger d'une tâche supplémentaire le microcontrôleur présent, un régulateur séparé sera très pratique. Il suffit d'y raccorder un capteur et un ventilateur. Un détail notable est que ces circuits intégrés ne s'en tiennent pas à la commande de ventilateur standard à quatre broches avec modulation de largeur d'impulsion (PWM) à 25 kHz, mais hachent directement la tension d'alimentation à des fréquences beaucoup plus basses.

(120569 – version française : Jean-Paul Brodier)

MIC502

Le MIC502 de Micrel utilise une thermistance CTN ou CTP comme sonde de température, avec une deuxième sonde en option. Les sondes sont connectées aux entrées VT1 et VT2. Une tension comprise entre 30 % et 70 % de V_{dd} détermine un rapport cyclique de 0 % à 100 %. La plus haute des deux entrées prend la priorité, une caractéristique bienvenue si vous surveillez avec une sonde la température ambiante et avec une autre celle d'un composant à forte dissipation. Une tension appliquée à la broche V_{slp} commande la mise en sommeil de la puce si les entrées VT1 et VT2 sont à un niveau inférieur. Si la tension de VT1 ou celle de VT2 revient au-dessus du niveau, la puce est ré-activée. Le but premier est d'éviter que le ventilateur tourne avec un rapport cyclique trop bas. Le condensateur de la base de temps est connecté à la broche CF. La valeur de 100 nF est recommandée pour une fréquence de 30 Hz. Vous pouvez aussi bien choisir une fréquence plus élevée ; selon la feuille de caractéristiques, la plage s'étend de 15 à 90 Hz.

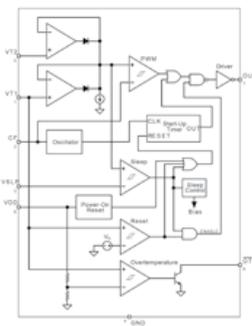


Figure 1 – Schéma synoptique du MIC502.

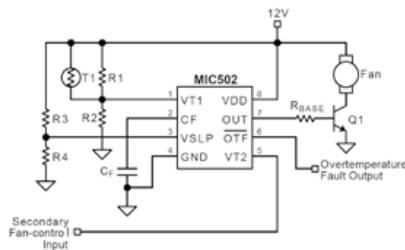


Figure 2 – Exemple d'application du MIC502.

Feuille de caractéristiques du MIC502 : www.micrel.com/_PDF/mic502.pdf

MAX6643

Le MAX6643 mesure la température au moyen d'un transistor connecté en diode. Trois réglages de température différents sont possibles. Le seuil de dépassement de température (broches OT1 et OT2, *Overtemperature Threshold*), fixe la température (entre 60 et 100 °C) au-delà de laquelle la sortie OT est activée. Le seuil de haute température (broches TH1 et TH2, *High Temperature Threshold*) fixe la température à partir de laquelle le rapport cyclique du signal MLI passe au cran supérieur. Le seuil de température basse (TL1 et TL2, *Low Temperature Threshold*) fixe la température le rapport cyclique descend d'un cran. Certains de ces seuils peuvent être réglés par le raccordement des broches correspondantes à V_{dd} ou à la masse (voir la feuille de caractéristiques). Une broche FULLSPD est disponible pour forcer le rapport cyclique à 100 %. Cela peut être utile si une température excessive est détectée. De plus, une entrée est disponible pour le signal tachymétrique du ventilateur, ce qui permet à la broche FANFAIL de signaler une défaillance éventuelle du ventilateur.

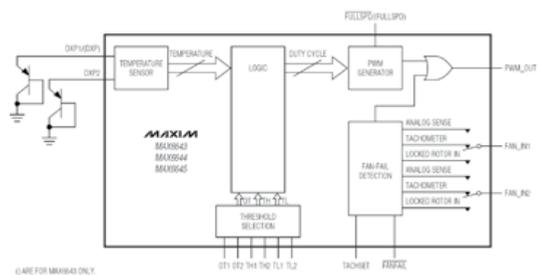


Figure 3 - Schéma synoptique du MAX6643.

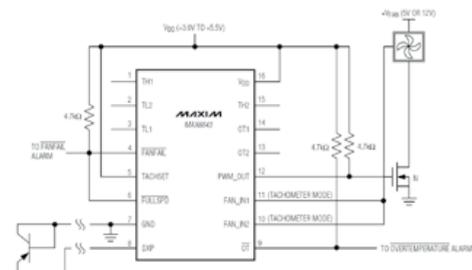
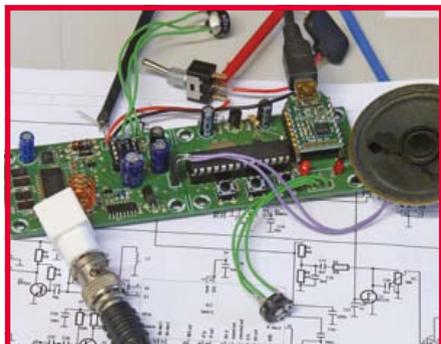


Figure 4 – Exemple d'application du MAX6643.

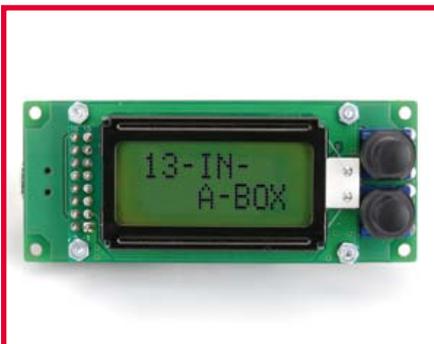
Feuille de caractéristiques du MAX6643 : <http://datasheets.maximintegrated.com/en/ds/MAX6643-MAX6645>.

Pour la première fois dans l'histoire d'Elektor, le magazine de janvier 2013 sera double. Un hors-gabarit, comme en été, au contenu considérablement enrichi !



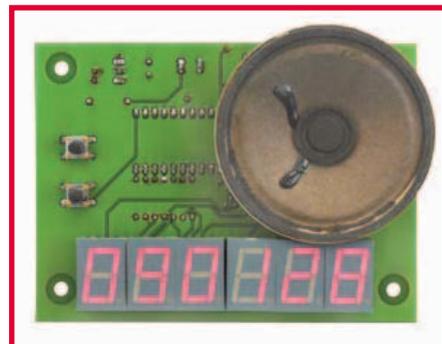
Scanner VHF pour la bande aviation

L'écoute des communications entre les pilotes et le contrôle aérien suscite l'intérêt de nombreux amateurs. Ce récepteur permettra aux *spotters* de suivre le trafic sur la bande VHF de 108 à 137 MHz réservée à l'aviation civile. Il est facile à construire, ne comporte qu'un seul réglage mécanique et peut être commandé depuis un PC via une connexion USB. Il fait appel à un unique circuit récepteur convertisseur VHF-FM, qu'il fait fonctionner en récepteur AM en utilisant habilement la sortie de détection de signal.



Multitesteur miniature

Imaginez que vous travaillez sur un projet à base de l'un ou l'autre microcontrôleur. Il y a des tas de choses à surveiller : tensions de sortie, niveaux logiques, impulsions de comptage et autres. Quel appareil utiliser ? Ce multitesteur réunit dans le même (tout petit) boîtier une belle gamme de fonctions : testeur logique, voltmètre, compteur d'impulsions, récepteur et transmetteur TTL/UART, fréquencemètre et... (à suivre).



Coucou numérique

Les pendules élect(ron)iques ont pris toutes les formes possibles et imaginables, et pourtant il en apparaît sans cesse de nouvelles, plus ou moins originales. En voici une qui va vous permettre à la fois de vous distraire, de vous singulariser et probablement de vous instruire. Comme son nom l'indique, elle appartient à la vaste famille des *cuculidés psittaciformes* à microcontrôleurs qui chantent tous les quarts d'heure la mélodie qu'on leur aura apprise.

Sous réserve de modifications. Le numéro double de janvier-février paraîtra le 16 janvier.

elektor  labs
est ouvert
24 heures sur 24
7 jours sur 7

Participez à l'élaboration
des projets sur
www.elektor-labs.com

The screenshot shows the homepage of **elektorlabs** with the tagline "Sharing Electronics Projects". It features a navigation menu (Home, News, Proposals, In Progress, Finished) and a search bar. The main content area is divided into sections: "wind speed and direction measurement" (with a video thumbnail), "Proposals" (listing "Switched 7805 Replacement THT" with a 4-star rating), "In Progress" (listing "dc motor driver" with a 5-star rating), and "Finished" (listing "Switched 7805 Replacement" with a 5-star rating). There are also sections for "About Elektor.LABS", "Create a Project" (with a "Create a new project or enter a proposal" button), "Challenges" (listing "Win a PSoC 5 Development Kit!"), and "Join a Project" (listing "Switched 7805 Replacement THT").

Une infinité d'applications une seule plate-forme

avec l'environnement de conception de systèmes le plus complet qui soit



NI LabVIEW est le seul environnement de développement qui offre autant de possibilités d'intégration matérielle ; au point de vous permettre de relever n'importe quel défi en matière d'applications de mesure et de contrôle. LabVIEW est au cœur de ce que l'on appelle la conception graphique de systèmes : une approche qui combine les avantages d'une plate-forme logicielle ouverte et d'un matériel reconfigurable à l'infini, pour que vous développiez votre système en un minimum de temps.

Le logiciel de conception de systèmes LabVIEW offre une intégration matérielle inégalée et vous permet de programmer de la façon dont vous pensez : graphiquement.



>> Concevez beaucoup plus rapidement vos systèmes en visitant ni.com/labview-platform/f

01 57 66 24 24

NATIONAL INSTRUMENTS France • 2 rue Hennepin • 92735 Nanterre Cedex, France • Tél. : (01) 57 66 24 24 Fax : (01) 57 66 24 14 • Société de droit américain - capital social 1.000,00 dollars • US - 11500 N Mopac Expwy, Austin-Texas USA - 10056236 - 344 497 649 • RCS Nanterre - SIRET B 344 497 649 00022 - APE 516 J - N.L.I. FR 57344497649

©2012 National Instruments. Tous droits réservés. National Instruments, NI, LabVIEW, et ni.com sont des marques de National Instruments. Les autres noms de produits et de sociétés mentionnés sont les marques ou les noms de leurs propriétaires respectifs. Pour plus d'informations concernant les marques de National Instruments, veuillez vous référer à la partie Terms of Use sur le site ni.com/legal. 08371

