

elektor

carte WiFi de commande à distance

pour rubans de LED, relais, moteurs, etc.



● de BASIC à **Python 2** | luxmètre céleste | modèle réduit perdu ?
chargeur d'accus li-ion | **FPGA 5** | réveil multifonction - réalisation
recyclez votre alim ATX ● pattes en l'air ● trio de LCR
1963 PSU à tubes ● Design Spark sur mesure 1^{ère} journée

L 19624 -420- F: 7,20 €



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 Hennape • 92735 Nanterre Cedex, France • Tél. : (0)1 57 66 24 24 Fax : (0)1 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 516J - N.I.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



eexplorez

le monde de l'électronique



**avec les cinq nouveaux
vaisseaux de la flotte Elektor**

elektor.magazine : le vaisseau amiral

elektor.labs : le drakkar

elektor.community : le paquebot

elektor.post : le catamaran

elektor.store : le cargo

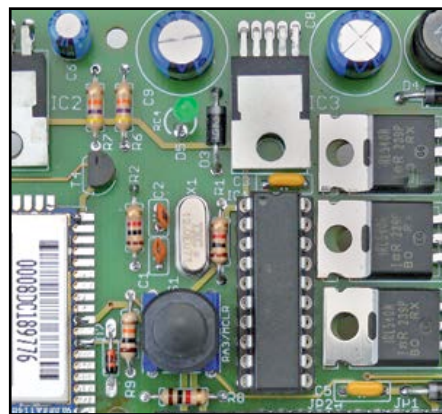
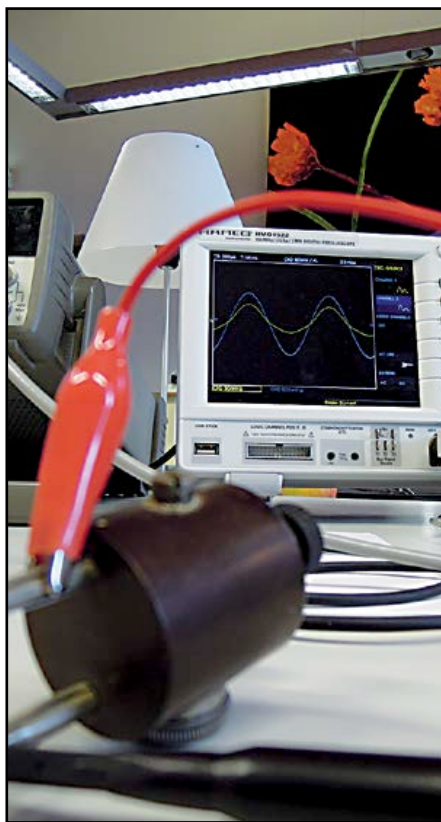


**Rejoignez la communauté Elektor :
embarquement immédiat !**



avec puce NFC Mifare programmable

Découvrez nos formules d'abonnement sur www.elektor.fr/abo



● communauté

6 de nous à vous

71 le monde d'Elektor

- Tube mystérieux
- (Ap)prendre les commandes

● industrie

8 tendances

● e-labs

12 LCR-mètres : *tRio grande*

Une idée née à la machine à café : comparer notre LCR-mètre à un modèle Hameg pro et à un modèle General Radio Company de 1960.

14 pattes en l'air !

Une carte SD devenue soudain invisible pour cause de configuration inappropriée des E/S inutilisées sur un FPGA

● e-pratique

**16 réveil multifonction
2^e partie**

Après avoir présenté le projet et étudié sa conception, voici les détails de l'électronique et la réalisation. Un montage aux possibilités étonnantes.

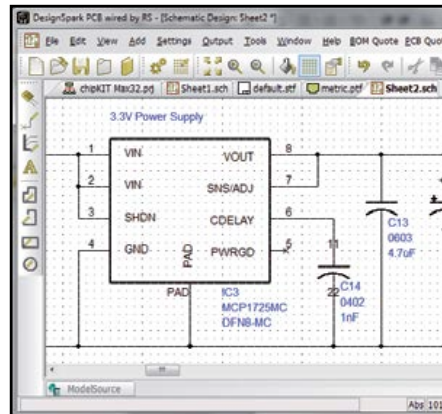
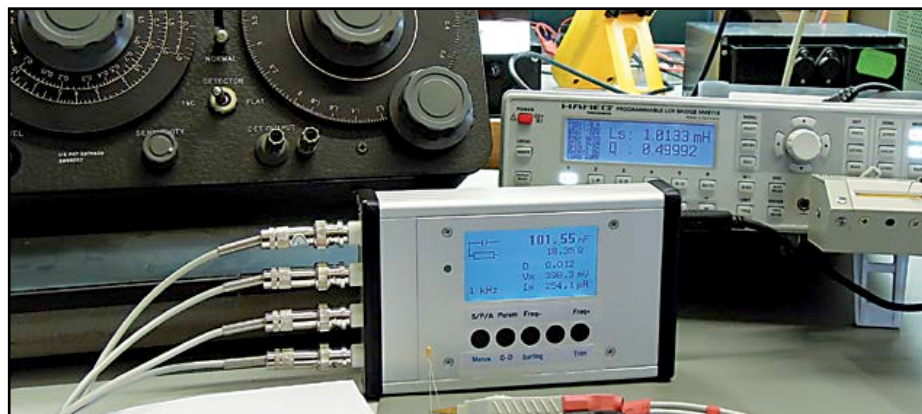
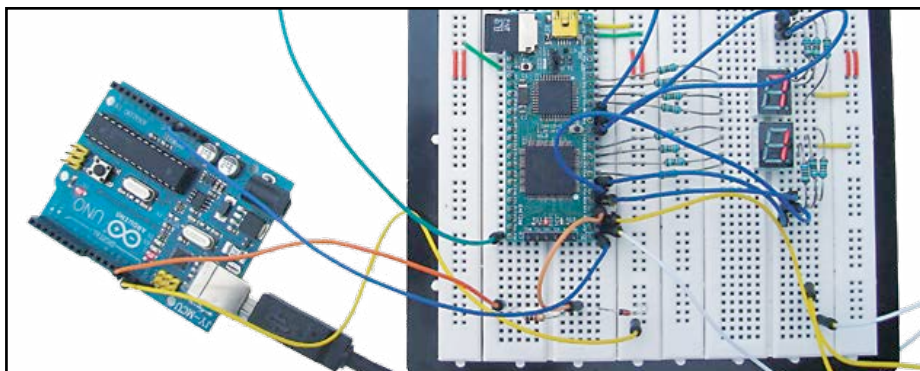
22 carte WiFi de commande à distance

Grâce à votre réseau WiFi, vous pouvez télécommander n'importe quoi et n'importe où dans votre maison !

32 chargeur d'accus lithium-ion sans microcontrôleur

La réutilisation d'accus lithium-ion récupérés est difficile parce qu'ils sont souvent chargés dans l'appareil lui-même. Construisons donc nous-mêmes un chargeur pour accus Li-ion récupérés (ou neufs).





36 modèle réduit perdu ? droit sur le site du crash

Radiogoniomètre et balise embarquée sont les mamelles du radiomodéliste aérien en détresse.

44 recyclez votre alimentation ATX avec simplicité et élégance

Une carte adaptatrice capable de convertir n'importe quelle alimentation ATX d'ordinateur en alimentation de paillasse, simple et élégante !

48 de BASIC à Python 2^e partie

Ouvrons les portes des bibliothèques graphiques de Python, un langage de programmation bien adapté à l'électronique.

56 luxmètre céleste

Mesure de lumière en occultation (totale) de soleil.

62 et l'homme créa sa puce 5^e partie 250 000 portes à tricoter avec des baguettes

Développer une application sur FPGA sous forme de réseau de symboles logiques, c'est possible, mais en pratique la méthode par programmation est souvent préférable.



● magazine

76 rétronique : alimentation de labo multiple à tubes Wandel & Golterman (vers 1963)

Remarquable alimentation à tubes, réglable, multi-tension, avec stabilisation électronique, fabriquée par Wandel & Goltermann entre 1920 et 1960.

80 hexadoku

casse-tête pour électroniciens

82 édition spéciale - été 2013

Ne manquez pas notre numéro double, avec encore plus d'idées, de projets, de réalisations, d'astuces et de trouvailles. À partir du 26 juin !

ELEKTOR / PUBLITRONIC SARL

c/o Regus Roissy CDG

1, rue de la Haye

BP 12910

FR - 95731 Roissy CDG Cedex

Tél. : (+33) 01.49.19.26.19

lundi, mardi et jeudi de 8h30 à 12h30

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 :

© 2013 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.

Plus ça change plus c'est... mieux

On parle beaucoup de la désaffection des électroniciens pour la conception et la production de circuits imprimés. Cette pratique figure pourtant toujours en bonne place parmi les centres d'intérêt de nos lecteurs. La réputation de qualité des cartes d'Elektor, établie depuis des décennies, n'y est sans doute pas étrangère. Il est loin le temps où nos cartes, dessinées à la main, au stylo Rotring et au ruban adhésif, étaient reproduites par photographie. Depuis belle



lurette tout cela est fait sur PC, à la souris, mais les débuts avaient été laborieux ; il fallait y croire, tant les outils informatiques étaient alors rudimentaires. Inutile de dire que cette innovation n'est pas à mettre au crédit des équipes dirigeantes de l'époque, mais de quelques pionniers hardis et visionnaires. Il en aura fallu, des nuits blanches, avant que l'informatique alors balbutiante parvienne à égaler le savoir-faire des virtuoses du crayon à papier et de la gomme.

Elektor a toujours dessiné ses propres circuits, mais en a sous-traité la fabrication à des professionnels. Nous ne fabriquons dans nos locaux que quelques-unes des cartes nécessaires pour les prototypes, et encore, le recours aux services spécialisés est de plus en plus fréquent, même pour les pièces uniques.

Le stockage et le conditionnement des cartes destinées à la vente sont eux aussi confiés à **elektorPCBservice**. De cette manière, ce service à nos lecteurs est assuré dans des conditions professionnelles de qualité et de diligence optimales, qu'il s'agisse des circuits imprimés nus ou de cartes assemblées prêtes à l'emploi (avec notamment les composants montés en surface, plutôt réservés aux robots ou aux doigts de fée). C'est grâce à cette récente amélioration de notre service de PCB que nous sommes en mesure de proposer à des conditions avantageuses des modules de compétition comme ceux du LCR-mètre publié récemment, une contribution d'origine française dont toute l'équipe est particulièrement fière.

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, Tim Uiterwijk, 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 online :	Daniëlle Mertens & Patrick Wielders

**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
+49 241 88 909-17
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
+31 6 1589 4245
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 OBSOLETE ELECTRONICS ENGINEERING INFORMATION**audioXPRESS**

vous connecte à



Nos annonceurs

**Beta Layout**www.pcb-pool.com 9**CadSoft**www.cadsoft.de 41**Eurocircuits**www.elektorpcbservice.com 81**HAMEG**www.hameg.fr 84**National Instruments**www.ni.com/labview-platform/f . . 2

Pour placer votre annonce dans le prochain numéro d'Elektor

veuillez contacter Mme Ilham Mohammadi par téléphone au (+31) 6.41.42.25.25
ou par courrier électronique : i.mohammadi@elektor.fr

Vos correspondants

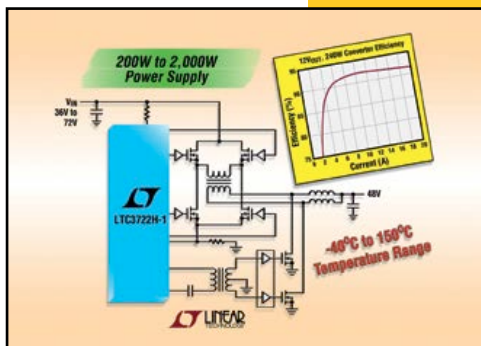
Nous sommes à votre service pour toute question relative à votre commande ou votre abonnement
par téléphone au (+33) 01.49.19.26.19 **lundi, mardi et jeudi de 8h30 à 12h30**
ou par courriel : service@elektor.fr

Contrôleur à pont et modulation de phase

Linear Technology Corporation annonce la version de classe H du LTC3722-1, un contrôleur DC/DC, à pont, à modulation de phase, avec commande manuelle ou ajustable (ZVS) de la commutation au zéro de tension. Le contrôle du retard de la fonction ZVS du composant et le réglage du redressement synchrone permettent d'optimiser le fonctionnement tout en réduisant la taille du transformateur et le niveau des interférences électromagnétiques, permettant d'utiliser ce composant pour des applications isolées, de forte puissance, jusqu'à plusieurs kilowatts.

La technique de commande ajustable au zéro de tension du LTC3722-1 est réalisée en mesurant la tension sur le pont de MOSFET et la tension d'entrée, ce qui permet au convertisseur de s'adapter à toute variation du courant dans la charge, de la température, des tolérances du composant, du retard des circuits de commande ou de la tension d'entrée. Les caractéristiques complémentaires incluent un contrôle à mode courant et fréquence fixe, le réglage de la limite d'intensité, programmation d'annulation du front montant, un démarrage progressif, une protection contre les courts-circuits et une compensation de pente programmable. Un régulateur 5 V, précis, à faible tension de déchet, fournit jusqu'à 15 mA pour l'alimentation des circuits auxiliaires.

La version de classe H fonctionne avec une température de jonction jusqu'à 150 °C.



Résumé des caractéristiques : LTC3722-1

- Commande de la commutation ajustable ou manuelle au zéro de tension
- Réglage du redressement synchrone
- Programmation de la durée de blocage du front montant
- Contrôle en mode courant
- Fréquence de fonctionnement fixe, programmable de 100 kHz et 600 kHz
- Synchronisation possible de la fréquence de fonctionnement de 100 kHz à 600 kHz
- Réglage de la limite du courant
- Démarrage progressif
- Régulateur LDO 5 V / 15 mA en sortie
- Programmation de la compensation de pente
- Protection contre les courts-circuits et les surcharges
- Fonctionnement sur la gamme de la température de jonction de -40°C à 150°C

Des centrales de mesure pour réduire efficacement la consommation d'énergie

Le coût de l'énergie augmente et chacun cherche de nouvelles solutions intelligentes pour réduire sa consommation d'énergie et surtout sa facture d'électricité. Pour cela, il est nécessaire de comprendre le système de consommation d'énergie des installations en collectant des données pour ensuite les analyser et prendre des mesures en conséquence. La nouvelle gamme de centrales de mesure MT de *GE Industrial Solutions* offre la possibilité de collecter précisément les données liées à la consommation telles que la tension, l'intensité, l'oscillation ou encore la fréquence. Il s'agit de simplifier la gestion des coûts tout en recherchant la plus grande efficacité énergétique des installations basse et moyenne tension. Disponible en trois versions (MTDIN, MT96 et MT144), la gamme MT s'adapte à diverses applications : mesure 3 phases adaptée aux applications où un montage sur rail DIN est préférable pour réduire l'encombrement, montages en façade et contrôle des alarmes (la consommation d'énergie peut être réglée sur des déclencheurs d'impulsions), mesure avancée pour montage en façade combinant un centre d'alarmes, un sous-compteur (qui mesure d'autres types de consommation comme le gaz ou encore l'eau), des convertisseurs de mesures et enfin un analyseur de qualité de la puissance.





L'ORIGINAL DEPUIS 1994
PCB-POOL
Beta LAYOUT

Le nouveau standard

Finition ENIG au lieu de l'étamage chimique sur votre carte

De l'or !

Sans supplément !

La meilleure finition : nickel-or chimique

Multicouche

Jusqu'à **50%**

de réduction de prix
pour les cartes
multicouche 2-5

www.pcb-pool.com

Beta

LAYOUT
create electronics

PCB-POOL est la marque déposée de Beta LAYOUT GmbH



eSTORE
Beta LAYOUT

Développer, assembler, souder



€ 36,50*

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

Reflow Controller



€ 129,00

Starter Kit UHF RFID Basic



€ 279,00

Le grand Beta-Reflow-Kit



€ 129,00

www.beta-eSTORE.com

Beta

LAYOUT
create electronics

eSTORE est la marque déposée de Beta LAYOUT GmbH

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.



elektor

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



Nouvelles caméras infrarouges

Les nouvelles caméras thermiques *Fluke* Ti105 et TiR105 complètent la gamme d'imageurs thermiques ultracompacts *Fluke*.

Compactes, faciles à utiliser et robustes, elles offrent la superposition d'une image plein spectre à l'image infrarouge, pour faciliter l'identification et la documentation, un raffinement inhabituel sur les caméras thermiques de cette catégorie de prix.

Dotée d'une gamme de mesure de température de $-20\text{ }^{\circ}\text{C}$ à $+250\text{ }^{\circ}\text{C}$ et d'une résolution infrarouge 160×120 , la caméra thermique *Fluke* Ti 105 permet aux professionnels dans les domaines électrique, mécanique, industriel et HVAC / R d'inspecter les équipements et de détecter rapidement les problèmes survenant sur les installations. Ces prises de mesures rapides et efficaces permettent de réaliser des interventions à distance et en toute sécurité, tout en gagnant du temps et en diminuant la probabilité d'arrêts non planifiés dans le cadre de maintenance préventive.

Le modèle TiR 105 est conçu pour le diagnostic de bâtiments : inspections et audits énergétiques, problèmes d'isolation, de fuites d'air, d'humidité, d'étanchéité ou des défauts de construction. Grâce au logiciel *SmartView* fourni, les caméras thermiques *Fluke* permettent de documenter rapidement les preuves visibles des problèmes détectés, de faire état de vos résultats et de personnaliser vos rapports en vue de réparations ou de mises à niveau.

Nouvelle génération de contrôleurs numériques de conversion d'énergie

L'ADP1051 est un nouveau contrôleur numérique de conversion d'énergie, destiné aux alimentations à découpage continu-continu isolées haute densité et ultracompactes. Il compte six sorties PWM programmables au moyen d'une interface graphique conviviale grâce à l'interface PMBus. Il permet de réaliser des topologies en pont complet à haut rendement ainsi que des variantes comme le circuit à clamp actif et la commande de redresseurs synchrones secondaires. Le rendement énergétique est optimisé dans toute la plage de charge au moyen d'une compensation active des temps de commutation, efficace en particulier pour les faibles charges. La consommation d'énergie du composant est inférieure à 100 mW, les pertes globales en mode veille basse consommation sont minimales. L'ADP1051 met en œuvre un jeu complet de commandes sur le PMBus, avec notamment la possibilité de relever avec précision les mesures de consommation en entrée et en sortie tout en permettant à l'utilisateur de prendre des décisions quant à la gestion de systèmes intelligents pour optimiser le rendement

énergétique. Encapsulé dans un boîtier QFN compact, le contrôleur ADP1051 convient idéalement aux modules d'alimentation continu-continu isolés ultra-compactes et haute densité, ainsi qu'aux alimentations



embarquées pour applications réseau, de communications et industrielles, pour des puissances de l'ordre de 100 W et plus. La programmation graphique permet aux utilisateurs de concevoir et d'enregistrer des paramètres en EEPROM interne haute

fiabilité. Ses multiples fonctions permettent de former un système robuste fonctionnant en modes parallèle et redondant, pour obtenir une grande fiabilité et une mise en parallèle aisée pour des besoins de puissance accrue. Ce contrôleur comprend une fonction de synchronisation maître/esclave, une protection contre les inversions de courant, une fonction de démarrage avec pré-polarisation, l'équilibrage précis du courant entre alimentations et une technique brevetée d'identification et d'inhibition sécurisée d'une alimentation défectueuse en mode parallèle. Le contrôleur numérique ADP1051 est conçu pour être associé aux circuits de commande de MOSFET ultra-rapides et aux isolateurs numériques iCoupler d'Analog Devices utilisés dans les systèmes d'alimentation isolés, ainsi qu'aux contrôleurs numériques de correction du facteur de puissance ADP1047 et ADP1048 qui équipent les systèmes de conversion AC-DC. Ces circuits facilitent la mise en œuvre de fonctions de gestion intelligente de la puissance qui permettent d'augmenter le rendement énergétique.

Nouveaux processeurs de reconnaissance d'images pour l'automobile

Toshiba a développé la série de processeurs de reconnaissance d'images Visconti afin d'accélérer la création de systèmes de vision utilisant des caméras pour des applications dans l'automobile. Les processeurs reconnaissent les files de trafic, les véhicules, les piétons, les panneaux de signalisation et encore bien plus d'éléments. Les composants des deux séries Visconti 2 et 3 comprennent des accélérateurs de traitement d'images qui détectent des piétons en temps réel.

Les composants Visconti 3 réduisent en outre les obstacles représentés par les développements logiciels pour les utilisateurs grâce à l'intégration d'un processeur ARM à 32 bits RISC Cortex-A9 MPCore (2 cœurs) pour les applications interfacées avec le processeur de reconnaissance d'images.

Les systèmes d'assistance au conducteur jouent un rôle croissant dans la sécurité des véhicules routiers. La prochaine impulsion arrivera dans l'Union européenne, où le programme Euro NCAP (*European New Car Assessment Programme* ou Programme européen d'évaluation des voitures neuves) exige des constructeurs automobiles l'installation de systèmes autonomes de freinage d'urgence à partir de l'an prochain. Cette décision touche initialement le freinage d'urgence pour éviter des collisions avec d'autres véhicules, mais sera étendue aux piétons en 2016.

Toshiba a commercialisé la série Visconti en 2004 et la série enrichie Visconti™2 en 2011. La nouvelle série Visconti 3 offre des améliorations au niveau des fonctions, de la performance et de l'efficacité du développement logiciel.

Principales caractéristiques

1. ARM Cortex-A9 MPCore

Pour améliorer la connectivité avec les systèmes de navigation automobile, les téléphones tactiles et les services de type *cloud*, et facilite aux constructeurs l'implantation de fonctions durant le développement logiciel.

2. Unité de calcul à virgule flottante en simple/double précision (FPU)

Les logiciels et les algorithmes de reconnaissance d'images utilisés par les processeurs de reconnaissance d'images embarqués doivent analyser de gros volumes de données provenant d'une source vidéo, trame par trame, en temps réel, et ceci avec des ressources limitées en terme de mémoire et de consommation. Certains algorithmes utilisés ne peuvent que difficilement réaliser cette tâche sans unité virgule flottante. C'est pourquoi les composants de la série Visconti 3 intègrent une FPU simple/double précision dans chaque cœur du processeur ARM.



Condensateurs ultra large bande

AVX lance une série de condensateurs ultra large bande conçue pour le blocage de courant continu de 16 KHz à 40 GHz. Les condensateurs de la série GX03 affichent des performances ultra large bande, de très faibles pertes d'insertion, d'excellentes valeurs de pertes de réflexion et les caractéristiques du diélectrique X7R. La série GX03 est idéale pour les applications de semiconducteurs pour communications, de sous-ensembles récepteurs optiques, d'amplificateurs de transimpédance et d'équipement de test.

« La série GX03 a été développée pour les applications qui exigent de plus hautes tensions que celles de nos autres condensateurs large bande » explique Larry Eisenberger, ingénieur chez AVX. De tension nominale 50 V_{CC} de -55°C à +125°C, les condensateurs ultra large bande GX03 sont en boîtier de taille 0603 ; complètement insensibles à l'orientation, ils utilisent le procédé de terminaison film mince de haute précision breveté d'AVX pour minimiser l'espace de carte occupé. Avec des terminaisons à revêtement nickel-étain et nickel-or, ils sont compatibles avec divers procédés de soudure, y compris le microcâblage pour les versions à finition nickel-or.



LCR-mètres : *trio grande*

Thijs Beckers
(Elektor)

Les idées nées à la machine à café ont souvent un potentiel d'espièglerie qui donne envie de les concrétiser illico. Voilà comment ça a commencé, juste après le bouclage le mois dernier du 3e article consacré à notre LCR-mètre de précision. Dialogue entre collègues de la rédaction et du labo :

« - On a comparé notre LCR-mètre à celui de Hameg, comparons-le aussi à un modèle historique !

« - Pourquoi pas ? À quoi penses-tu ?

« - J'ai dans ma collection Rétronique un modèle de 1960, beau comme un camion.

« - Allons-y ! »

Quelques jours plus tard, trois ponts de mesure d'impédance d'origines disparates se côtoient sur la paillasse du labo. Le modèle *Hameg* HM8118 [1], de *Rohde & Schwarz*, est un appareil professionnel (1600€ HT). Le modèle antique de la collection de Jan B. est un GRC 1650-A (*General Radio Company*) fabriqué en 1960, et qui à cette époque devait coûter 1.000 \$ [2]. Et enfin le LCR-mètre 0,05 % d'Elektor [3], qu'il est possible d'assembler pour environ 200 €.

C'est par le GRC que j'ai commencé. Respect aux ancêtres, il est plus vieux que moi ! Pour mes critères de 2013, c'est le moins facile à utiliser des trois — sans le manuel, aucune chance pour moi d'en tirer quoi que ce soit. Bilan :

- lire le mode d'emploi : 10 min ;
- apprendre à mesurer une capacité : 5 min ;
- mesurer (obtenir le « zéro » sur tous les cadrans) : 3 min.

Un usage quotidien de ce vénérable appareil permettrait de gagner 80 % du temps que j'ai mis. Il reste qu'avec le 1650-A, il faudra toujours compter une demi-minute pour mesurer un composant. Mais quel plaisir de manipuler ces robustes commutateurs rotatifs d'un autre âge ! Le HM8118 est (plus ou moins) *plug-and-play*. Le calibrage est automatique et dès que le composant (*Device Under Test*) est monté sur le bornier *Hameg* HZ181, ses caractéristiques sont affichées instantanément. En tout, 40 sec et je n'ai pas lu le mode d'emploi. Pour des mesures en série, si je compte trois secondes pour manipuler les composants, et deux pour le choix des réglages, il faut cinq secondes.

Tableau comparatif

DUT	GRC 1650-A	Hameg HM8118	LCR-mètre Elektor
R = 8,2 Ω	8,22 Ω	8,2458 Ω ($V_x=31,86$ mV, $I_x=3,862$ mA)	8,2379 Ω ($V_m=31,13$ mV, $I_m=3,779$ mA)
C = 100 nF	100 nF D: 0,0267	102,03 nF D: 0,01205 $R_s=19,20$ Ω ($V_x=364,9$ mV, $I_x=233,9$ μA, 1 kHz)	102,20 nF D: 0,013 $R_s=19,20$ Ω ($V_m=398,1$ mV, $I_x=256,0$ μA, 1 kHz)
C = 100 μF	92 μF D: 0,1	94,475 μF D: 0,10099 $R_s=170,27$ mΩ ($V_x=7,301$ mV, $I_x=4,312$ mA, 1 kHz)	92,400 μF D: 0,090 $R_s=155,4$ mΩ ($V_m=7,076$ mV, $I_x=4,091$ mA, 1 kHz)
L = 1 mH	1,255 mH Q: 0,63	995,85 μH Q: 0,49189 $R_s=12,722$ Ω ($V_x=52,60$ mV, $I_x=3,711$ mA, 1 kHz)	993,2 μH Q: 0,492 $R_s=12,681$ Ω ($V_m=51,18$ mV, $I_x=3,621$ mA, 1 kHz)



Le modèle d'Elektor enfin, avec juste un afficheur LCD, un bouton marche-arrêt et cinq boutons pour les menus : difficile de faire plus *plug-and-play* que ça ! Le calibrage semi-automatique avait été fait lors de l'assemblage du prototype que j'ai utilisé et j'ai donc pu passer directement à la mesure. En tout et pour tout, il m'a fallu 10 secondes, peut-être moins. Si pour faire des mesures en série vous utilisez des sondes Kelvin au lieu du bornier Hameg, la manipulation des composants durera forcément plus longtemps. Le bornier HZ181 (ou un modèle meilleur marché) est utilisable aussi avec le LCR-mètre d'Elektor sans que cela altère sa précision.

Le reste est dans le tableau. Pour le confort, il n'y a pas photo : les techniques ont changé ! Pour la précision, il n'y a photo qu'entre Hameg et Elektor. Là c'est le prix qui fait la différence. Le Hameg offre des fonctions que le modèle d'Elektor n'a pas, mais la précision de ce dernier pour un prix défilant toute concurrence n'est pas le moindre de ses atouts.

Précisons enfin qu'après plus de 50 ans, nul ne se serait attendu pour le GRC 1650-A à des résultats (encore) aussi présentables. Le monde

qui a produit des outils de cette qualité force le respect et l'admiration, mais c'était un univers fermé de spécialistes. À tout prendre, je préfère le monde d'aujourd'hui qui met de tels outils à la portée de tout le monde.

OFFRE DE LANCEMENT JUSQU'À LA FIN DU MOIS DE JUIN

Les cartes du LCR-mètre sont disponibles assemblées et testées, prêtes à l'emploi. Tous les composants y sont soudés. Il suffit de les monter dans un coffret et de les connecter à une interface USB pour commencer à utiliser l'appareil.

Une remise exceptionnelle de 15% est offerte à nos lecteurs sur l'achat de la carte principale et de la carte d'extension. Si vous les achetez séparément, **la remise est de 10%**.

(130166)

Liens

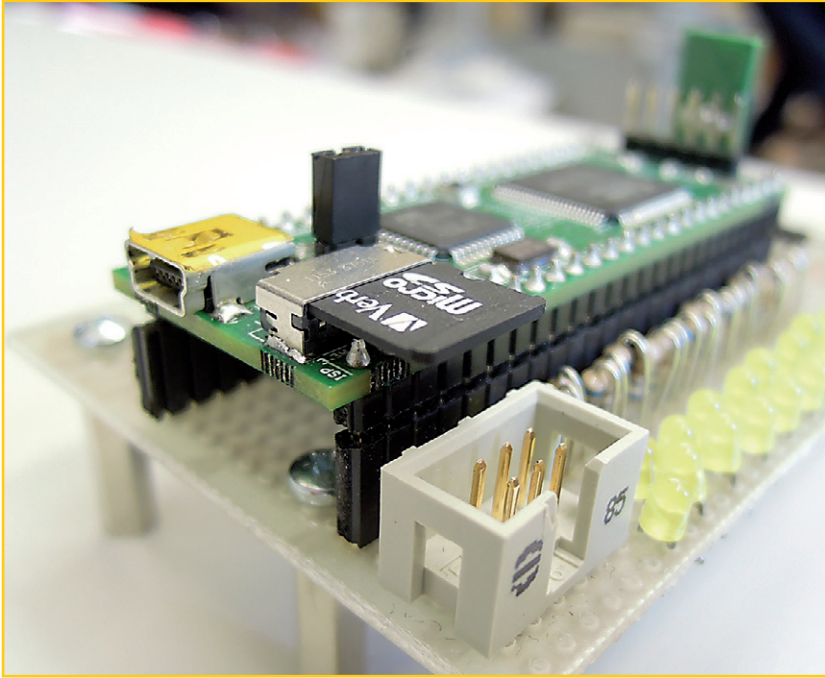
[1] www.hameg.com/13.0.html

[2] www.elektor.fr/075064

[3] www.elektor.fr/110758



pattes en l'air !



Thijs Beckers
(Elektor)

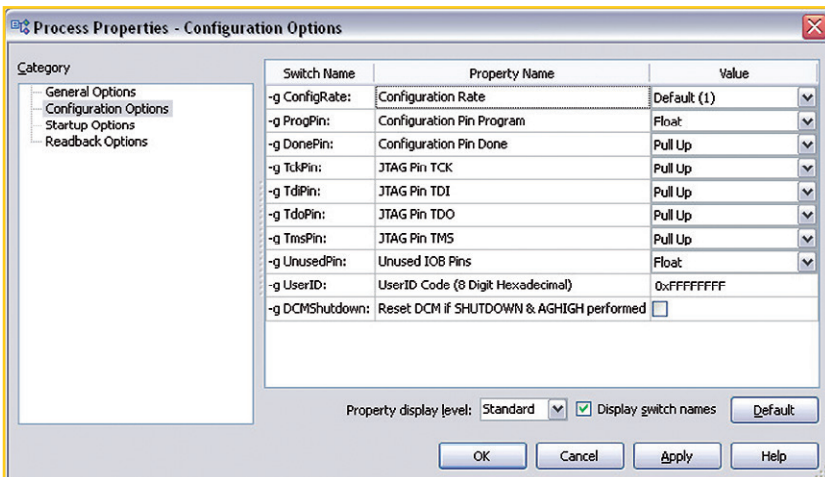
La carte d'expérimentation pour FPGA d'Elektor publiée dans le numéro de décembre 2012 [1] dispose d'une carte micro SD pour stocker les données de configuration du FPGA au démarrage, mais aussi d'autres données, internes ou externes. C'est en œuvrant sur une application pour sa carte FPGA que Raymond Vermeulen, un de

nos concepteurs, s'est trouvé confronté à des soucis majeurs avec la carte SD. Un jour, après une programmation du FPGA avec un micrologiciel flambant neuf, toute la carte devient invisible pour Windows. Raymond cherche la panne, débranche et rebranche le câble USB, révisé le logiciel, remesure tout, rien à faire. La carte SD serait-elle hors service ? Non, dans un autre lecteur, elle fonctionne. L'erreur est ailleurs, et ailleurs, c'est vaste !

On recherche plutôt deux fois qu'une, tous les défauts possibles y passent, soudure après soudure, mais tout est en ordre. Raymond refait toute la procédure et finalement reprogramme le FPGA. Enfin, c'est l'eurêka : quand on programme un FPGA, le logiciel de l'environnement de développement demande de valider un paquet de paramètres. Dans les options de configuration (cf. copie d'écran), on peut aussi traiter les broches d'E/S inutilisées. En fait, on doit ! Et Raymond lui-même l'avait bien expliqué à ceux qui se servent de la carte d'expérimentation pour FPGA d'Elektor : toutes les E/S inutilisées doivent être mises sur *float*, en l'air.

Sur la carte SD, comme les broches de données sont connectées à la fois au microcontrôleur et au FPGA, il est indispensable que le FPGA laisse ces broches flottantes quand il ne s'en sert pas. La suite de développement applique par défaut aux E/S inutilisées une résistance de rappel au niveau bas. Parfait, la plupart du temps, mais pas ici, justement : il faut changer ce réglage en *float* ! Le *setting* corrigé, le FPGA reprogrammé, la carte reprend vie et l'application de Raymond fonctionne aussitôt. « Faites ce que je dis, ne dites pas ce que fais ». Soulagé d'avoir trouvé, Raymond a proposé sa mésaventure comme anecdote à raconter dans la rubrique .LABS. À la prochaine...

(130036 – version française : Robert Grignard)



Liens

[1] www.elektor.com/120099

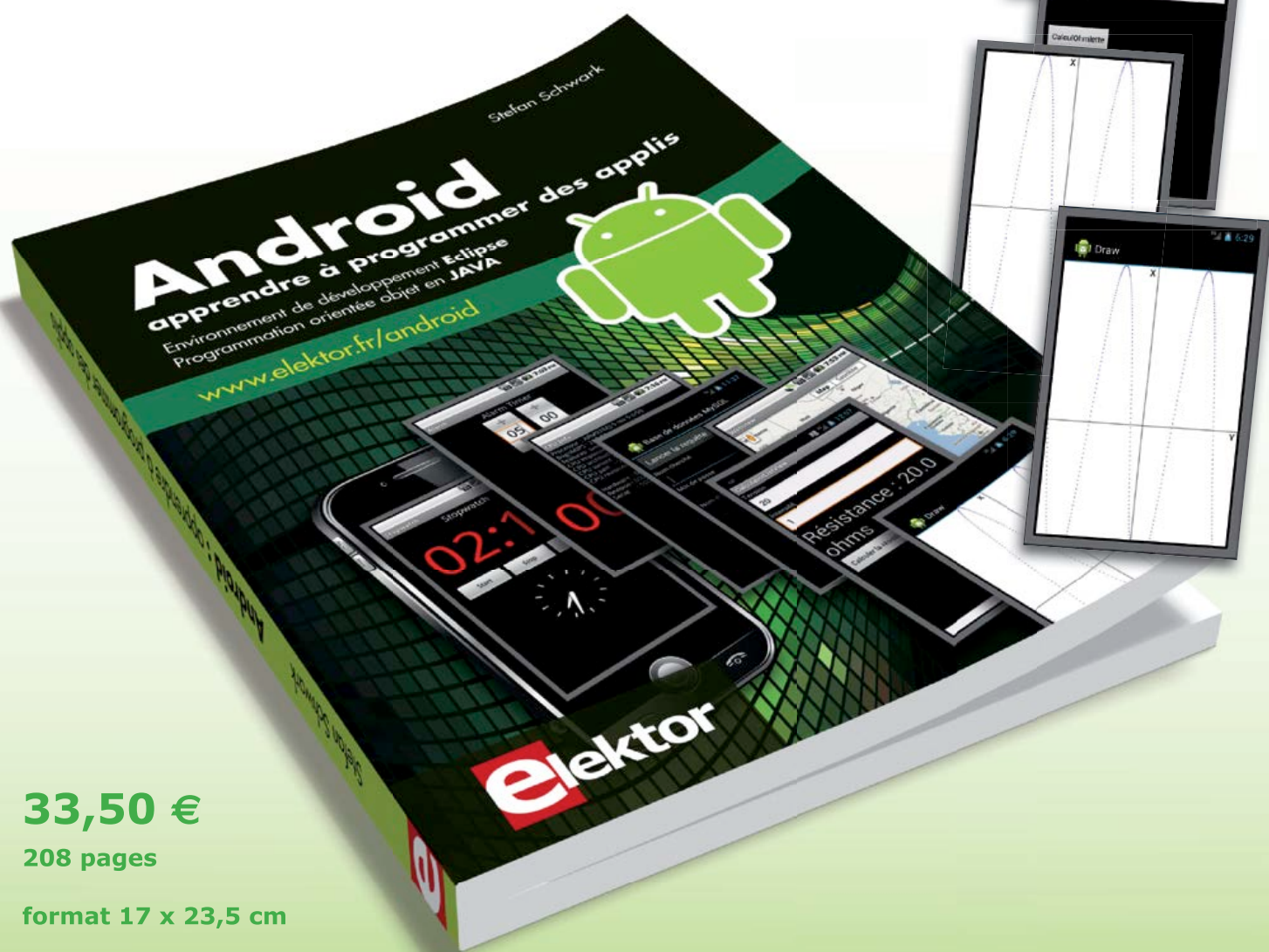


Télécharger des applications existantes n'est pas le seul moyen de **rendre votre téléphone tactile encore plus intelligent !**

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

C'est un ouvrage d'initiation, avec des exemples simples, variés et concrets, qui montre de façon progressive comment la combinaison de briques de code permet de créer toutes sortes d'applis dans l'environnement de développement Eclipse :

calculatrice simple, interrogation des capteurs, exploitation des données GPS, communication par l'internet etc.



33,50 €

208 pages

format 17 x 23,5 cm

ISBN 978-2-86661-187-3

elektor

www.elektor.fr/android

2^e partie

réveil multifonction en avant pour les schémas

Maintenant que nous avons présenté le projet et étudié sa conception, nous sommes fin prêts pour rentrer dans les détails de l'électronique de notre horloge.

Description du circuit – carte principale

Le schéma principal (**fig. 3**) donne l'impression d'une simple carte de commande industrielle, avec un minimum d'interfaces d'E/S. Vue de plus près,

la partie gérant l'audio est bien moins rudimentaire que sur une commande programmable ordinaire. Le choix du microcontrôleur a déjà été expliqué dans la première partie de l'article. L'AT89C5131 est un dérivé de la famille 8051 d'Intel et possède donc la plupart des caractéristiques des cœurs 8051, dont la configuration inexplicablement bizarre de leurs ports d'E/S. Le port 0, par exemple, possède des sorties à collecteur ouvert qui nécessiteront

Michael J. Bauer
(Australie)

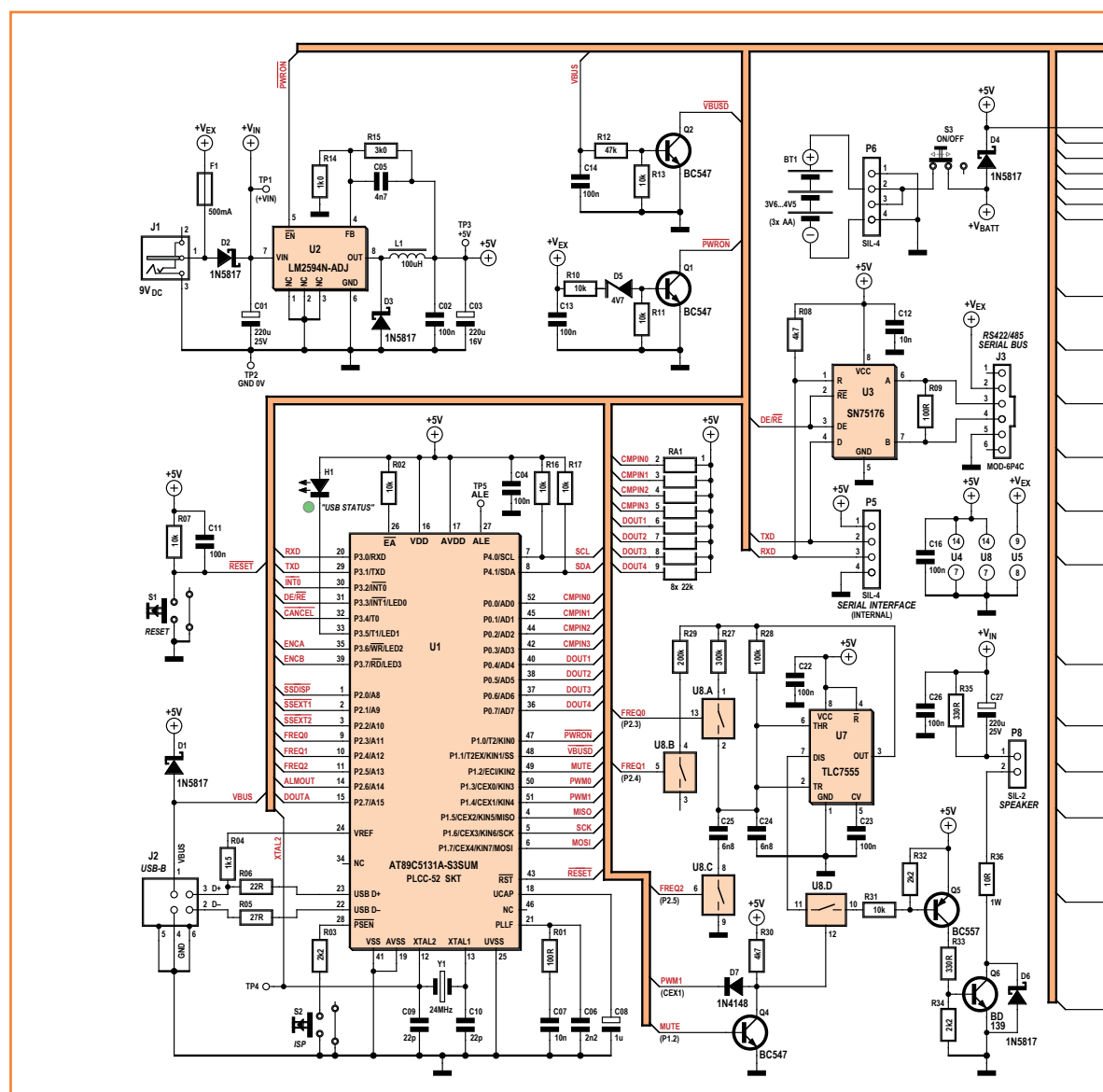


Figure 3.
Schéma de la carte principale.

l'utilisation de résistances de rappel. La valeur élevée des résistances de rappel internes ($\sim 25\text{ k}\Omega$) des autres ports limite le courant de sortie à une intensité qui ne permet pas de commander la plupart des charges qui nous intéressent.

Néanmoins, le 89C5131 se rattrape sur d'autres caractéristiques, p. ex. son cœur très vélocé. L'utilisation conjointe d'un quartz de 24 MHz et du mode X2 permet de cadencer le cœur du micro à 48 MHz ! La puce comporte aussi des périphériques : USB, temporisateur T2, PCA (pour les sorties MLI), SPI, et TWI (I²C).

L'AT89C5131 possède un bloc de mémoire flash séparé (4 KB) pour accueillir un chargeur de démarrage. Il est livré avec un chargeur de

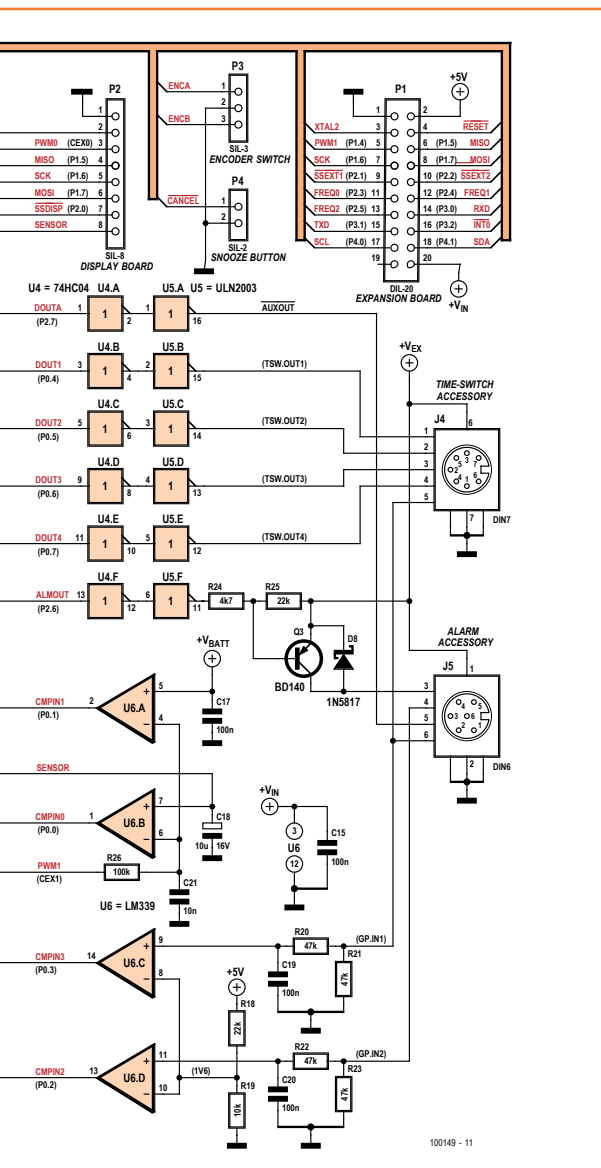
démarrage préprogrammé dans ce bloc, ce qui vous permettra de charger votre code dans la mémoire de programme (32 Ko) via le port USB sans recours à un programmeur séparé. Lorsque la broche PSEN# du processeur est maintenue à l'état bas (à l'aide du bouton ISP) lors d'une remise à zéro, c'est le chargeur de démarrage qui se lance à la place du code d'application. Les signaux TX et RX de l'UART sont disponibles sur le connecteur d'extension P1, ainsi que sur le connecteur à quatre voies P5 ; vous pourrez y relier un adaptateur RS-232. Ces signaux ont également été routés vers l'émetteur-récepteur RS-422/485 U3, un SN75176, pour rendre possible la prise en charge d'accessoires via un bus sériel. La version actuelle du micrologiciel du réveil n'en fait pas usage. En attendant que ce soit le cas, vous pouvez omettre U3 et J3.

Par ailleurs, la tension d'alimentation continue externe (+V_{EX}) est surveillée. Si elle descend en dessous d'environ 6 V, le transistor Q1 se bloquera et le signal PWRON# se trouvera à l'état haut à cause de la résistance de rappel intégrée. Le micrologiciel surveille ce signal et allume un indicateur (PF) lorsque la tension est insuffisante. Le signal PWRON# est aussi relié à la broche EN# du régulateur à découpage 5 V U2 (LM2594). Lorsque la tension externe 9 V disparaît, le régulateur 5 V est coupé, ce qui permet à l'USB ou à la batterie de prendre le relais.

La tension V_{bus} du bus USB est surveillée par le montage articulé autour du transistor Q2. Si cette tension est absente, le transistor Q2 se bloque et le signal VBUS# se retrouvera à l'état haut. Le micrologiciel adapte son fonctionnement suivant l'état de ce signal. L'état du signal V_{bus} peut également être utilisé par le micrologiciel pour commander une (ou plusieurs) sortie(s) de commande d'accessoire.

Logique d'E/S des accessoires

Six sorties logiques sont disponibles sur les connecteurs pour accessoires J4 et J5. Les quatre sorties commandées par l'horloge sont actives au niveau bas et capables de drainer chacune jusqu'à 50 mA. Elles ont été prévues pour piloter des optocoupleurs (entrées isolées) sur une carte de puissance à relais statiques. La sortie 9 V DC pilotée par alarme est active au niveau haut ; elle peut fournir jusqu'à 500 mA (limité par le fusible réamorçable F1). Comme les ports correspondants du micro ne sont pas en mesure de commander directement le circuit de pilotage U5 (ULN2803A),



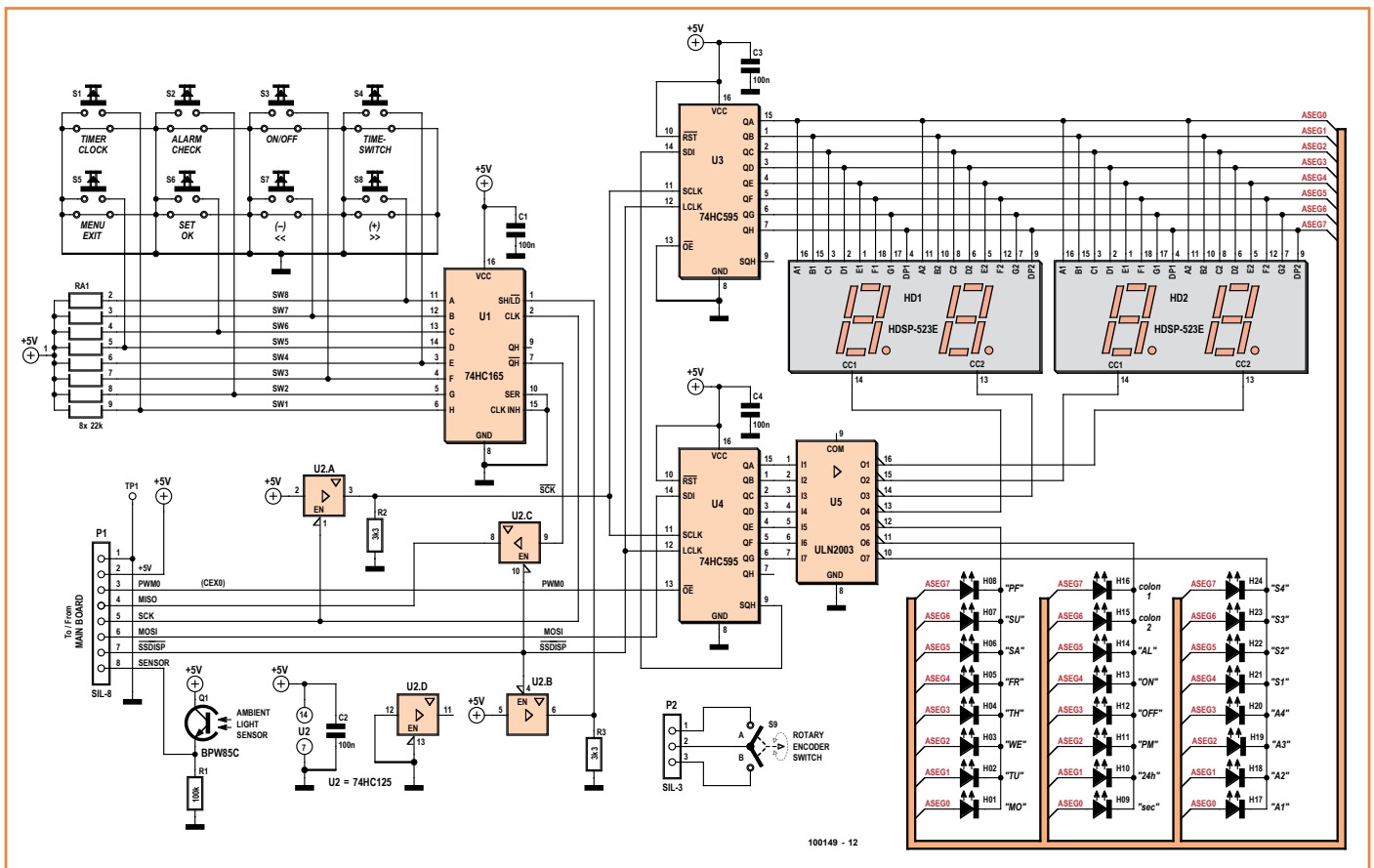


Figure 4.
Schéma de la carte
d'affichage.

ils ont été dotés du sextuple inverseur CMOS U4. Deux entrées logiques, réalisées autour des comparateurs U6C et U6D sont disponibles. Le seuil choisi est 1,6 V. Les entrées externes sont tamponnées et filtrées par des réseaux RC afin de protéger les entrées du micro des transitoires nocifs tels que les DES.

Entrées analogiques

L'AT89C5131 ne possède pas de CAN et ne fournit donc aucune entrée analogique alors qu'il nous en faut deux pour surveiller la luminosité ambiante et la tension de la batterie de secours. La conversion est obtenue à l'aide d'une astuce logicielle et d'une sortie MLI du micro utilisée pour produire une référence de tension variable. Un filtre passe-bas (R26, C21) retire la composante alternative du train d'impulsions (47 kHz) afin d'obtenir une tension continue commandée par le rapport cyclique de la MLI.

La tension de référence MLI attaque l'entrée inverseuse des comparateurs U6A et U6B, les tensions à mesurer étant reliées aux entrées non-inverseuses. En faisant évoluer périodiquement

la tension de référence, le micrologiciel est en mesure de déterminer les tensions. Ce dispositif est diaboliquement astucieux, mais sa précision est limitée par la résolution (ainsi que la linéarité et le bruit) des tensions de référence produites par MLI. Au mieux, la précision devrait être de 1%. À cause de la constante de temps du filtre, la tension de sortie met plusieurs millisecondes à se stabiliser après un changement du rapport cyclique de la MLI. Les conversions seront donc moins rapides qu'avec un convertisseur matériel, mais ici la vitesse n'est pas une contrainte.

Générateur audio et amplificateur

Un générateur de signal carré est réalisé autour du temporisateur CMOS U7 (TLC555). Sa fréquence est fixée par le réseau RC associé. Huit fréquences différentes (de 500 à 2000 Hz environ) peuvent être choisies à l'aide des sorties FREQ2, FREQ1 et FREQ0 du micro. C'est le quadruple interrupteur analogique U8 (CD4066B) qui permet de commuter les résistances R27 et R29 ainsi que le condensateur C25. Les fréquences ont été choisies pour être harmoniques.

L'amplitude est commandée par un signal MLI produit sur la broche CEX1 du micro qui commande la « découpe » via U8D du signal audio carré. C'est le rapport cyclique de ce signal qui détermine le niveau audio en sortie. La fréquence de la MLI est d'environ 47 kHz, bien au-delà du spectre audible et de la gamme du HP. Les 8 bits du registre de réglage du rapport cyclique donnent 256 valeurs. Cette plage de réglage suffit à notre application, mais 10 bits (ou plus) n'auraient pas été de trop. Le signal audio conserve sa forme carrée (modulée à 47 kHz), jusqu'au transistor Q6 qui pilote le HP. Notre amplificateur fonctionne en classe D. Le transistor Q5 permet de fournir à Q6 un courant de base suffisant pour commander le HP. L'interrupteur analogique U8 n'est pas capable de commuter des courants supérieurs à environ 2 mA. L'utilisation d'un amplificateur en classe D fournit non seulement plus de puissance qu'un amplificateur linéaire, mais présente également un autre avantage majeur pour notre application. Lorsque le générateur audio est désactivé (le micrologiciel met le signal MUTE au niveau haut), l'interrupteur MLI U8D est alors coupé et le transistor Q6 bloqué ; aucun courant ne traverse le HP, quelle que soit la quantité de bruit sur la tension d'alimentation ($+V_{IN}$). C'est un avantage énorme pour un appareil susceptible de se trouver à moins d'un mètre de vos oreilles lorsque vous cherchez le sommeil.

Toutes les deux millisecondes, le niveau audio est mis à jour par la routine *sound synthesizer* du micrologiciel. Celui-ci est capable de synthétiser une enveloppe d'amplitude pour les sons d'alarme. Des paramètres réglables par l'utilisateur permettent de spécifier l'attaque (pente), la tenue (durée), le relâchement (pente) et la pause (durée), de la même manière que sur les synthétiseurs analogiques des années 70 (pour lesquels l'auteur éprouve beaucoup de nostalgie). La durée de pause correspond à l'intervalle de silence entre deux répétitions d'un son d'alarme. (À ne pas confondre avec l'intervalle de dodo, réglable séparément.)

En plus de cela, le micrologiciel fournit un moyen de moduler la fréquence et/ou l'amplitude du signal audio avec des signaux modulateurs (virtuels) dont la période est réglable par pas de 2 ms (2, 4, 6, 8... ms), jusqu'à 500 ms.

En résumé, un circuit générateur bon marché, combiné à un micrologiciel intelligent, permet de fournir une vaste gamme d'effets audio attractifs (et d'autres moins attractifs !).

Description du circuit – carte afficheur

L'affichage se fait sur quatre afficheurs à 7 segments à LED. C'est une solution bon marché, facile à lire et qui permet de faire varier facilement la luminosité. Un afficheur LCD rétroéclairé blanc ou multicolore aurait certes été plus esthétique, mais son coût difficile à justifier et son approvisionnement incertain.

Vous trouverez en **figure 4** notre afficheur à quatre chiffres réalisé avec deux afficheurs à deux chiffres. Les modèles choisis (Avago HDSP-523X) sont disponibles en trois couleurs : rouge, vert ou jaune. Des composants au brochage compatible sont disponibles chez d'autres fabricants. Les différents indicateurs sont réalisés à l'aide de 24 LED rondes 5 mm à éclairage diffus de diverses couleurs.

Les afficheurs à 7 segments et divers indicateurs à LED sont multiplexés en une matrice à cathode commune de 7 x 8. Les cathodes sont pilotées par l'intermédiaire de U5 (ULN2003A), un ensemble de sept transistors Darlington avec des entrées compatibles TTL/CMOS.

Les anodes, quant à elles, sont pilotées par un registre à décalage de 8 bits. En regardant le schéma, les lecteurs fûtés auront remarqué qu'il n'y a pas de résistance de limitation du courant en série avec les lignes d'anode. Ce schéma utilise la résistance des sorties CMOS (env. 60 Ω) pour limiter le courant dans les LED. À l'état bas, les sorties de l'ULN2003 affichent une tension minimale d'environ 1 V, montage Darlington oblige. La tension de chute directe dans les LED est d'environ 2 V. Cela nous donne un total de 3 V qui, soustraits des 5 V de l'alimentation, laissent régner 2 V sur les résistances des sorties CMOS. Le courant maximal dans les LED est donc limité à environ 33 mA (2 V/60 Ω).

Rappelez-vous que les LED seront multiplexées en 8 ensembles et que leur luminosité sera réglée par MLI, le courant max peut donc sans risque être proche du maximum admissible par les LED. Le courant moyen dans les LED sera bien plus faible que ça : environ 2 mA dans des conditions de fonctionnement normales. La variation de luminosité est réalisée en appliquant un signal MLI de 47 kHz à rapport cyclique variable sur l'entrée *Output Enable* (OE#) du circuit de pilotage des cathodes U4. Nous disposerons de 256 valeurs différentes de luminosité. La valeur minimale (1/256) permet à l'afficheur d'être suffisamment lumineux pour être lisible lorsque l'éclairage ambiant est faible.

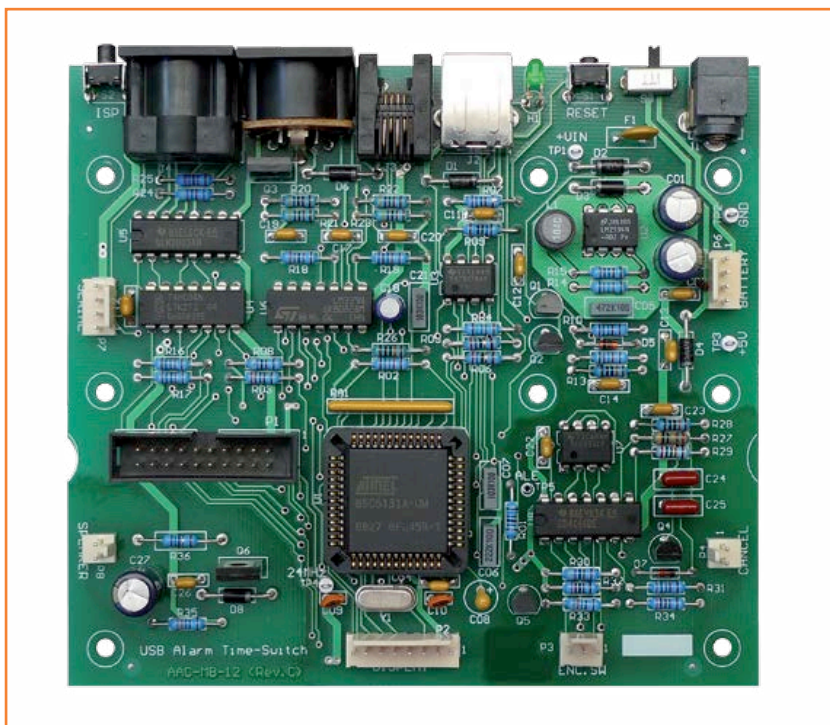
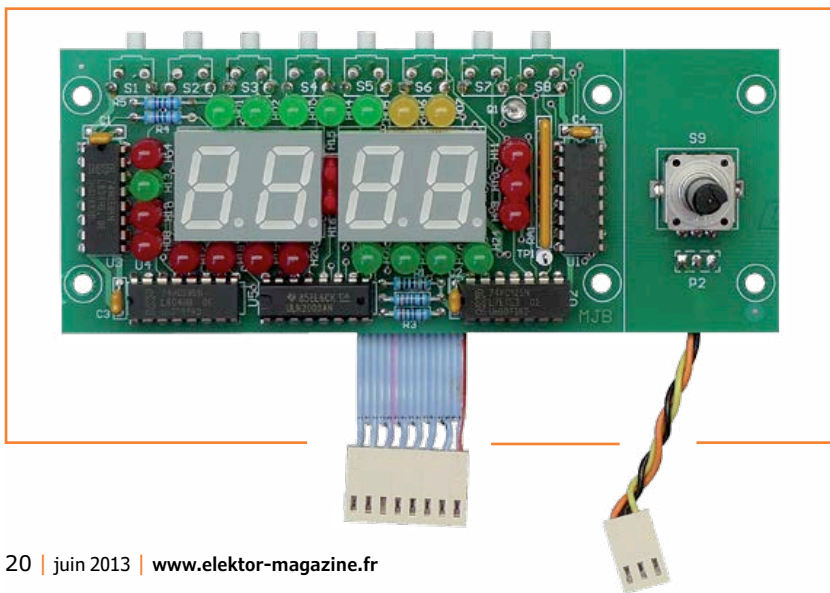


Figure 5.
La carte principale
assemblée, vue de dessus.

Au passage, ne soyez pas tentés d'utiliser des LED bleues ou blanches pour les indicateurs ; leur seuil de tension est trop élevé (autour de 4 V en général) pour permettre leur utilisation dans ce circuit.

Il nous faudra deux ports de 8 bits configurés en sortie pour l'afficheur à LED et un port de 8 bits en entrée pour les poussoirs. L'extension des E/S du micro est réalisée par des registres à décalage CMOS de 8 bits possédant des verrous parallèles pour les sorties et une logique série pour y transférer les données. Ces registres à décalage sont (presque) compatibles avec l'interface SPI du micro. Avec un soupçon de logique en plus,

Figure 6.
La carte d'affichage
assemblée, vue de dessus.



incarné par le quadruple tampon à trois états U2 (74HC125), les registres U3 et U4 (74HC595) de l'afficheur et le registre U1 utilisé pour les boutons (74HC165) seront faciles à raccorder au bus SPI. Vous remarquerez que les tampons U2A et U2B sont câblés en inverseurs et possèdent des résistances de rappel à la masse en sortie. L'inversion de la polarité de l'horloge du bus SPI permet de lire et écrire sur le bus en même temps.

L'utilisation du bus SPI permet de minimiser le nombre de signaux utilisés pour relier la carte d'affichage à la carte principale. Les registres des boutons et de l'afficheur partagent un même signal de sélection SPI (SSDISP#). Les registres de l'afficheur sont écrits et le registre des poussoirs lu lors du même transfert de deux octets sur le bus SPI. Puisque le registre des boutons ne fait qu'un octet, le micrologiciel ignore le deuxième octet lu. Le signal de sélection SSDISP# est inversé pour obtenir un signal de décalage/chargement SH/LD#. Lorsque le port d'entrée n'est pas sélectionné (SSDISP# à l'état haut), les données parallèles sont acceptées sur le verrou d'entrée. Lors d'un transfert SPI, SSDISP# est au niveau bas et SH/LD# au niveau haut, ce qui bloque le verrou d'entrée et active l'horloge série. En résumé, les données en entrée sont maintenues constantes lorsqu'elles sont lues.

Les registres de sortie (74HC595) sont constitués d'un registre à décalage de 8 bits à entrée sérielle et sortie parallèle doté d'un octuple verrou de type D. C'est la valeur présente dans leurs verrous qui pilote leurs sorties. Lors d'une impulsion sur le signal d'horloge des verrous (LCLK), le front montant entraîne le transfert des 8 bits contenus dans le registre à décalage interne sur les verrous de sortie. Tant que LCLK est à l'état haut, les bits du verrou de sortie ne changent pas. Les bits du registre à décalage interne changent lors d'un transfert SPI (lecture ou écriture), et ce peu importe l'état du signal de sélection SPI. En d'autres termes, les données du registre à décalage changent lors des transferts vers d'autres périphériques sur le bus SPI. Cela importe peu tant que le signal LCLK ne bouge pas lorsque ces données indésirables se trouvent dans le registre à décalage. En reliant LCLK au signal de sélection de l'afficheur SSDISP#, on obtient le comportement désiré.

La carte d'affichage possède un phototransistor (Q1 = BPW85C) qui permet de mesurer la luminosité ambiante. Le courant dans la résistance d'émetteur, et donc la tension à ses bornes, augmente avec l'intensité de l'illumination de la

jonction du transistor. La tension d'émetteur est surveillée par un comparateur sur la carte principale, à l'aide de notre technique approximative de conversion analogique/numérique présentée plus haut.

Un emplacement a été ménagé sur la carte d'affichage pour monter un codeur rotatif (S9). Ceux qui choisiront de ne pas utiliser le boîtier présenté ici voudront peut-être le placer ailleurs. Ce sera assez facile à faire puisqu'il suffira de couper la carte d'affichage en deux afin de séparer la petite partie supportant le codeur. C'est pour cette raison que le codeur est câblé séparément du reste de la carte (nappe à 8 conducteurs).

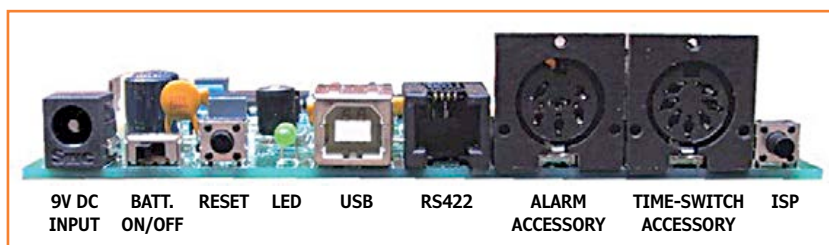
En interne, le codeur possède deux jeux de contacts qui s'ouvrent et se ferment lorsque l'on fait tourner son axe. Dotés de résistances de rappel au +5 V, les deux contacts (A et B) produisent des trains d'impulsions rectangulaires en quadrature de phase (c.-à-d. déphasés de 90°). Chacune des sorties produit 24 impulsions par tour de l'axe, mais le déphasage des deux signaux est différent (de 180°) en fonction du sens de rotation. Une routine du micrologiciel se charge d'éliminer le bruit des signaux et de suivre les mouvements de l'axe du codeur.

Assemblage

L'assemblage est décrit en détail dans un document intitulé *Assembly Instructions* que vous pourrez télécharger gratuitement depuis le site d'Elektor [1]. L'archive contient également la liste des composants. Par manque de place, nous ne pouvons reproduire ici que des photos. Vous pourrez jeter un coup d'œil à la carte principale assemblée (fig. 5), à la carte d'affichage (fig. 6), aux connecteurs disposés à l'arrière de l'horloge (fig. 7), ainsi qu'à la façon de monter la carte d'affichage dans le boîtier (fig. 8).

À vous de jouer !

Le code du micrologiciel est abondamment commenté et pourra servir d'exemple aux étudiants pour leurs projets de micrologiciels en temps réel, qu'il s'agisse de reproduire ou de modifier le réveil. Ce projet possède le potentiel pour donner naissance à plusieurs rejetons, et pas seulement des accessoires pour le réveil, mais bien des projets pouvant être utilisés indépendamment. Par exemple : une carte secteur à relais statiques à quatre voies, une lampe de chevet à LED avec variateur, une interface de commande par infrarouges (pour appareils A/V), une carte d'exten-



sion pour effets sonores PCM/MP3 (reliée via SPI au micro). Les trois derniers demanderont une mise à jour du micrologiciel pour pouvoir être utilisés avec le réveil.

Les membres de la communauté **ElektorLabs** [2] sont invités à développer une interface graphique sous Windows pour l'horloge présentée ici. Elle pourrait par exemple servir à éditer, sauvegarder et restaurer les différentes alarmes des plannings, changer les options, etc. Développeurs, à vous de jouer !

Tel qu'il est présenté ici, le réveil fonctionne tout seul et ne dépend pas d'un logiciel Windows.

Note : Suite à la publication de cet article dans l'édition anglaise d'Elektor en janvier-février et mars 2013, l'auteur a reçu assez de demandes au sujet du circuit imprimé pour justifier une production en petit nombre pour rendre service aux lecteurs intéressés par cette réalisation. Toutes les informations à ce sujet sont réunies sur son site [3].

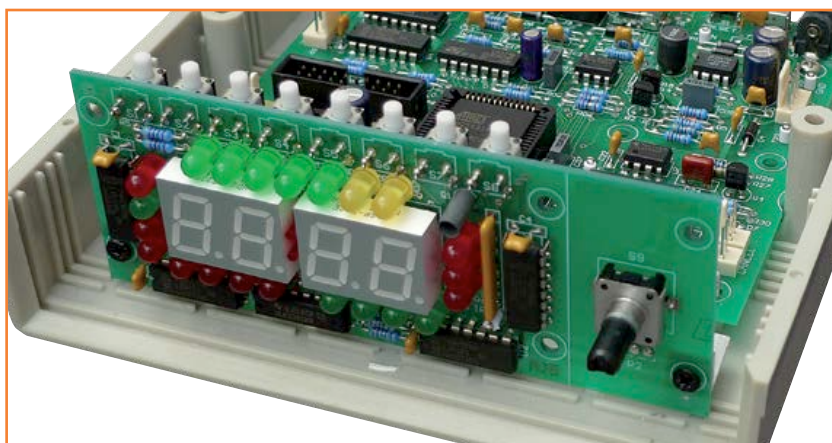
(120655 – version française : Kévin PETIT)

Liens

- [1] www.elektor.fr/100149
- [2] www.elektor-labs.com
- [3] www.mjbauer.biz/build_the_7-uP_alarm_time-switch.htm

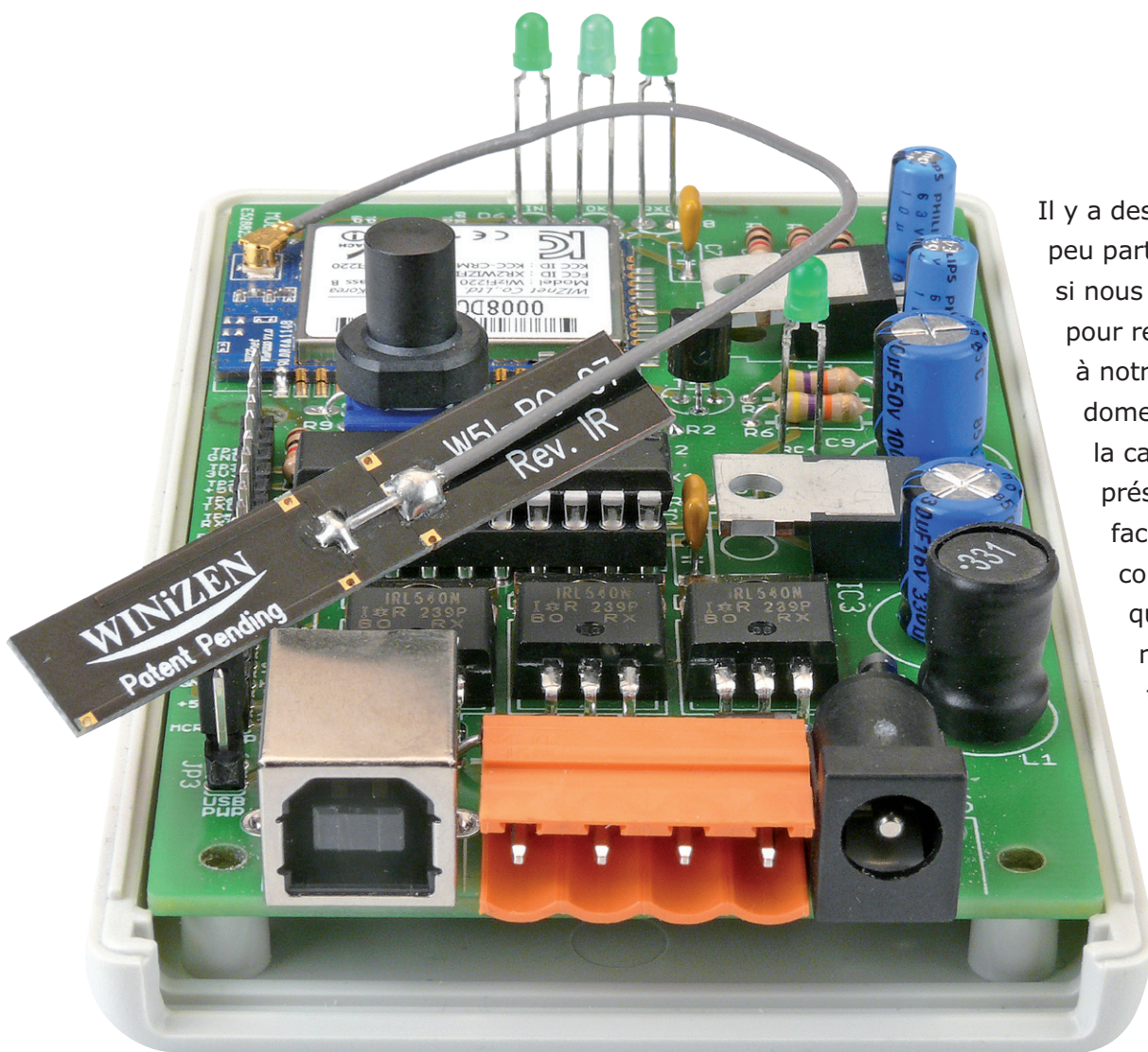
Figure 7.
Vous ne manquerez certainement pas de connecteurs.

Figure 8.
La carte d'affichage est prévue pour être disposée verticalement dans le boîtier Pactec CM6-225.



carte Wi-Fi de commande à distance

pour rubans à LED RVB, moteurs, relais ou autres, mais sans-fil



Il y a des modules Wi-Fi un peu partout maintenant ; si nous en profitons pour relier des appareils à notre réseau Wi-Fi domestique ? Grâce à la carte de commande présentée ici, il est facile de (télé-) commander n'importe quoi dans votre maison depuis votre téléphone par le Wi-Fi. Un ruban à LED RVB p. ex. ou autre chose.

Clemens Valens
(Elektor.Labs)

Tout a commencé il y a un an avec le *Home Automation system* [1] développé au labo Elektor par deux stagiaires, Koen et Jesper. Leur système comportait un module de commande de ruban à LED RVB par Ethernet (filaire donc). Sympa, mais pourquoi pas une connexion sans-fil, avec une application de commande dans un navigateur qui soit compatible avec la plupart

des navigateurs et règle la couleur du ruban avec des potentiomètres à glissière virtuels. Koen s'est attelé à la tâche sans arriver au bout avant la fin de son stage. Avant qu'il ne quitte Elektor, il m'a informé oralement des problèmes rencontrés, certains non résolus, puis le projet a pris la poussière sur mon bureau. Lorsque j'ai enfin trouvé le temps et le courage de m'y

plonger, j'avais tout oublié de ce que m'avait dit Koen. Grosso modo je suis reparti de zéro. Pour doser l'intensité des trois couleurs, Koen utilisait un AVR ATmega328 qui commande trois MOSFET en MLI. Pour la Wi-Fi, je pensais au module WizFi220 de WIZnet qui communique avec le micro simplement avec un lien sériel. Comme j'avais sous la main un *shield* Arduino doté justement de ce module [2] et que la carte Arduino Uno est basée sur un ATmega328, j'ai décidé que mon prototype serait une application Arduino. Il ne manquait qu'un deuxième *shield* avec des MOSFET pour commander le ruban à LED. Abracadabr... Arduino, le prototype est là (**fig. 1**) ! La suite ne serait que du code. Analyser les commandes HTTP reçues via le module Wi-Fi et y répondre correctement n'était pas compliqué ; c'est quand j'ai voulu changer la page web que j'ai rencontré les problèmes décrits par Koen. Le premier a été l'implantation des potentiomètres rectilignes : ils n'existent pas dans le standard HTML. En fouillant un peu sur la toile, j'ai découvert que HTML 5 les définissait, mais que peu de navigateurs les reconnaissent ; exit HTML 5. La solution de Koen consiste à utiliser les bibliothèques JavaScript JQuery et JQuery-UI [3]. Celles-ci permettent d'implanter des composants graphiques et autres joyeusetés dans des pages web. Le problème, c'est que ces bibliothèques sont en ligne, qu'il faut donc une connexion internet pour les utiliser et qu'elles sont trop volumineuses pour tenir dans la mémoire de programme du µC. Faute de mieux, je les ai utilisées en dépit de ces désagréments. En parcourant le code de la page web écrit par Koen, je me suis soudain rappelé un des problèmes dont il m'avait parlé : pour une raison encore inconnue, le module Wi-Fi ferme la connexion après la réception d'une commande, empêchant le navigateur d'en envoyer d'autres. On ne pouvait donc changer la couleur du ruban de LED qu'une seule fois, après il fallait redémarrer le module. Koen s'en dépatouillait avec un code JavaScript compliqué qui forçait le navigateur à changer de port avant qu'il n'envoie une commande. Son bricolage impliquait également d'utiliser la commande HTTP GET pour envoyer les couleurs, alors que cette commande, qui l'eût cru, sert à récupérer des données depuis un serveur, pas à en envoyer. Pour envoyer des données, il faut normalement utiliser la commande POST (ou PUT). Une solution plus élégante serait d'ajouter

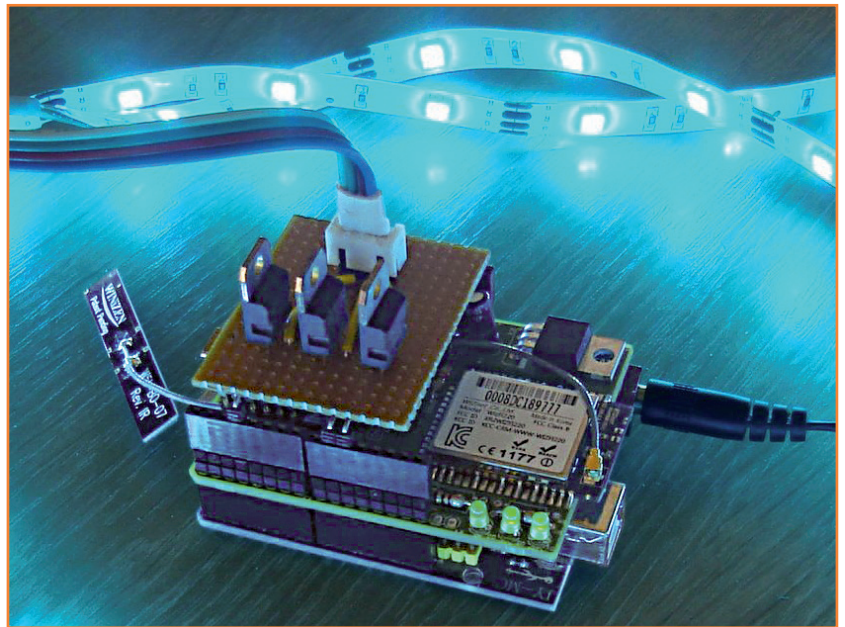


Figure 1.
Le prototype réalisé avec un Arduino Uno, un *shield* Wi-Fi Elektor (120306) et un morceau de plaque à essais.

la ligne Connection: close à la réponse du serveur à une commande GET. Du coup, je peux enlever le script qui contournait le problème, réduire la taille de la page et utiliser la commande POST, qui, à mon avis, convient mieux. Voici à quoi ressemble la commande POST telle qu'envoyée par Firefox (les données de couleur sont à la fin) :

```
POST / HTTP/1.1
Host: 192.168.2.15
User-Agent: Mozilla/5.0 (Windows
NT 6.1; WOW64; rv:17.0) Gecko/17.0
Firefox/17.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: keep-alive
Content-Type: application/x-www-form-
urlencoded; charset=UTF-8
Content-Length: 23
Origin: null
Pragma: no-cache
Cache-Control: no-cache

red=79&green=10&blue=20
```

L'étape suivante a été d'améliorer l'aspect de la page sur les appareils mobiles. Sur mon téléphone tactile, les potentiomètres sont si petits qu'il était presque impossible de les

manipuler avec précision. Ajouter la balise meta viewport à l'en-tête de la page a résolu ce problème :

```
<meta name='viewport'
content='width=device-width, user-
scalable=no' />
```

Avec cette ligne ajoutée au code HTML, la page occupe maintenant tout l'écran de mon téléphone Android tout en conservant une apparence correcte sur un PC. (Sur un iPad, elle n'occupe qu'environ un quart de l'écran et je n'ai pas cherché à améliorer ça.)

J'ai aussi optimisé le reste du programme afin qu'il occupe le moins de place possible en mémoire. J'ai gagné beaucoup de place en compressant la page web. On peut par exemple compresser la page avec *gzip* et ajouter la ligne "Content-encoding: gzip" à la réponse du serveur. La plupart des navigateurs, peut-être tous, savent quoi faire de pages web ainsi compressées. L'inconvénient, c'est qu'il est maintenant plus compliqué de modifier la page web ; je vous conseille de ne la compresser que lorsqu'elle ne changera plus.

Mon prototype est fonctionnel et compatible Arduino, tant du côté matériel que du logiciel. Si vous voulez les détails, visitez [4]. J'aurais pu m'arrêter là, mais je voulais améliorer le matériel : empiler trois PCB pour commander quelques LED ? Non ! En cherchant un micro avec moins de pattes, j'ai compris que je ne pourrais pas rester sur la plateforme AVR d'*Atmel* : les modèles qui convenaient sont difficiles à trouver. Par chance, quelques PIC18F14K50 de *Microchip* traînaient par là. Ce micro à 20 pattes avec USB me permettrait de configurer confortablement le module Wi-Fi depuis un PC [2]. Le chargeur de démarrage fourni gratuitement par *Microchip* était également une caractéristique intéressante qui faciliterait le développement d'un micrologiciel sans disposer d'un programmeur spécifique. Malheureusement, ce micro ne dispose que d'une voie MLI et il m'a donc fallu implanter dans le logiciel les trois voies MLI de commandes des couleurs.

L'adaptation pour ce PIC du code de l'AVR aurait dû être facile et directe. Eh ben non... j'avais un peu cherché les complications en voulant utiliser un chargeur de démarrage et l'USB, mais *Microchip* ne facilite pas les choses non plus. Avec mon super compilateur XC8 toutes options

que *Microchip* ne cesse d'encenser, je n'ai pas réussi à compiler le code USB ni bien entendu à le faire fonctionner. J'avais trouvé ce code dans la *Microchip Application Libraries v2012-10-15* et ce n'est qu'après plusieurs heures de bidouillage des paramètres du compilateur et des *pragmas* que j'ai lu quelque part sur l'internet que XC8 n'était pas (encore) compatible avec la bibliothèque de *Microchip*. La bonne blague ! Il me fallait donc un autre compilateur... que j'ai trouvé en moins d'un quart d'heure sur l'internet : un C18 officiel avec toutes ses options sans déboursier un centime. Quinze minutes plus tard, le code d'exemple du port série USB fonctionnait.

Voici venu le temps d'ajouter mon code Wi-Fi — cela s'est plutôt bien passé sauf pour logger les données dans la RAM segmentée du PIC — et de l'essayer. Et il fonctionnait... presque. Disons un peu. Je pouvais changer la couleur du ruban une fois ou deux, puis la communication Wi-Fi s'arrêtait. En commandant le ruban via le port série USB, tout fonctionnait bien. Après un peu de débogage, j'ai découvert que l'interruption USB empêchait apparemment l'interruption du port série de se déclencher, ce qui provoquait des pertes de données. Lassé de ces tracasseries, plutôt que de résoudre le problème de fond, j'ai décidé de désactiver le port USB en mode Wi-Fi. Si vous trouvez une meilleure solution, merci de m'en informer.

Assemblage

Mon second prototype à base de PIC fonctionnait maintenant comme prévu avec les composants de mon choix (**fig. 2**) ; il était temps de concevoir un circuit imprimé à monter dans un petit boîtier discret. Ce PCB a été conçu pour qu'il ne faille effectuer que peu de modifications sur le boîtier, surtout si vous utilisez la version bleue transparente qui permet de voir à l'intérieur la LED d'état. Dans ce cas, il n'y a aucun perçage, il suffira de couper les entretoises à l'intérieur. Plusieurs des 20 pattes du μC restaient inutilisées. Cela m'a incité à ajouter des fonctions, de sorte que, grâce au chargeur de démarrage, la carte deviendrait une plateforme polyvalente utilisable pour d'autres applications :

- contrôleur de (ruban à) LED RVB Wi-Fi
- contrôleur à trois voies pour moteurs ou relais par USB ou Wi-Fi (ou les deux)
- convertisseur USB / Série
- système de prototypage rapide avec Wi-Fi,

port d'extension et trois voies de commande de puissance

- *FlowBoard* compatible avec *FlowStone 3* [5]
- quelque chose auquel je n'ai pas pensé

Pour rendre ces différentes applications possibles, j'ai équipé la carte d'un régulateur à découpage permettant de l'alimenter efficacement à partir d'une tension continue entre 7 et 40 V. La

carte peut également être alimentée depuis le port USB, mais n'oubliez pas que, lorsqu'il transmet, le module Wi-Fi est gourmand. Il est alimenté par un gros régulateur linéaire 3,3 V (il consomme trop pour pouvoir être alimenté par le régulateur 3,3 V intégré au micro). Le module est accompagné de trois LED indiquant s'il est connecté à un point d'accès ou non, ou si des données sont en cours de transfert.

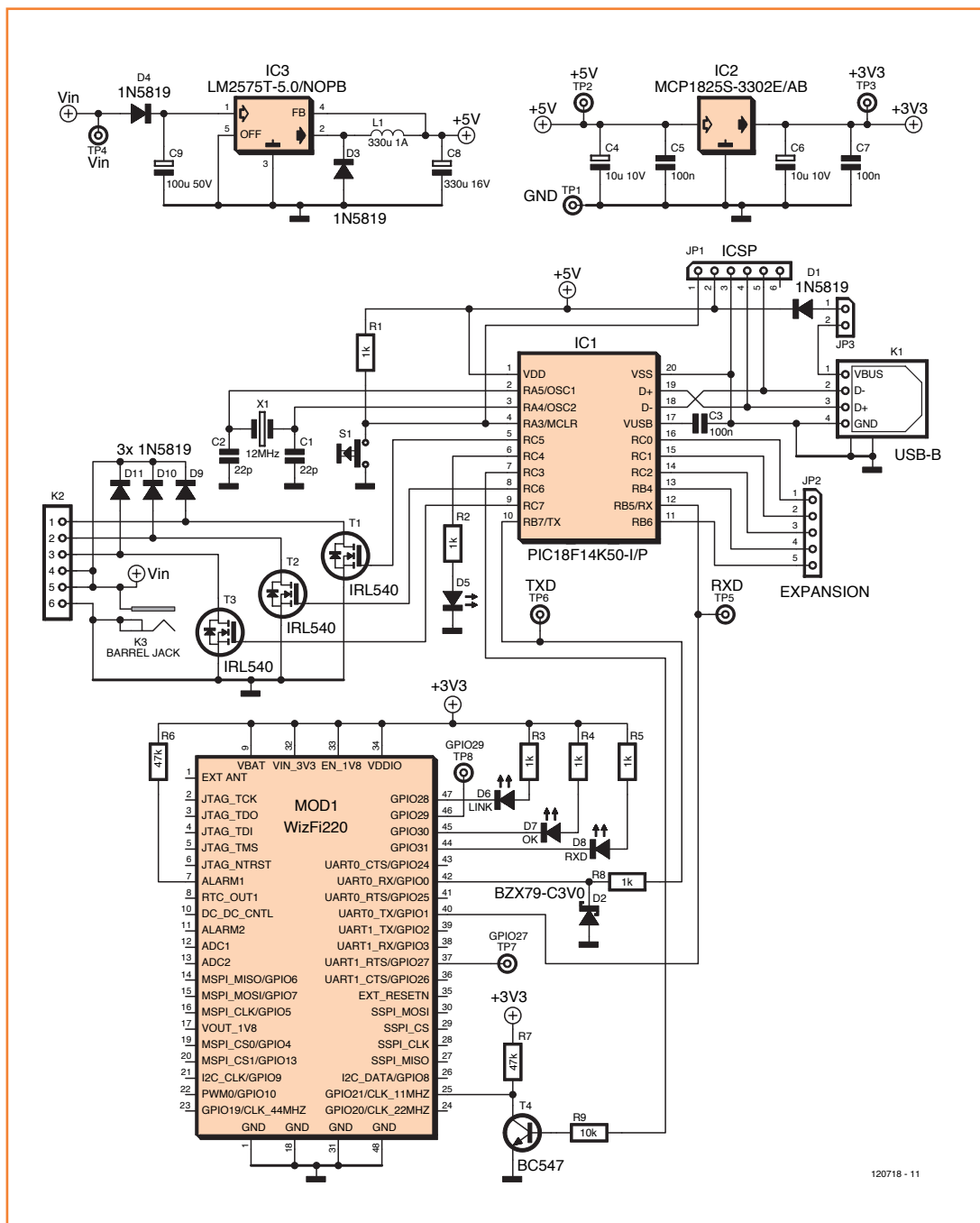


Figure 2.

Le schéma de notre carte de commande Wi-Fi. Avez-vous remarqué que les jacks sont toujours câblés dans le mauvais sens ? La patte centrale est censée être reliée à la masse. J'ai été vilain ici parce que ça simplifiait le routage du PCB.

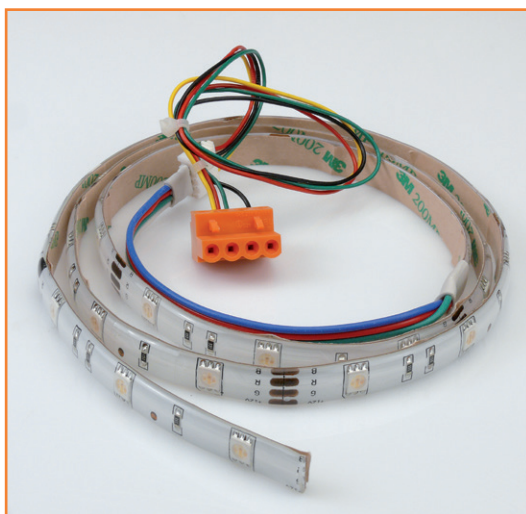


Figure 3.
Un ruban à LED RVB tout ce qu'il y a de plus standard. Le câble adaptateur à été récupéré sur une alimentation de PC.

Étant donné que le micro est alimenté en 5 V et le module Wi-Fi en 3,3 V, des convertisseurs de niveaux ont été ajoutés pour les entrées du module Wi-Fi. Le micro peut très bien utiliser des signaux 3,3 V en entrée.

Le micro est cadencé par un oscillateur à quartz à 12 MHz à cause de l'USB. Le poussoir relié à la patte de remise à zéro (MCLR) sert principalement à activer le chargeur de démarrage (l'entrée de remise à zéro externe doit être désactivée pour que cela fonctionne, fusible MCLR=OFF). Si vous n'utilisez pas le chargeur de démarrage, il servira à la remise à zéro ou une autre fonction que vous lui trouverez.

Les trois MOSFET, plutôt courants, possèdent une $R_{DS(on)}$ de 77 mΩ, peuvent commander jusqu'à 100 V et sont protégés par des diodes de roue libre afin de pouvoir les utiliser avec des charges inductives. Pour les connecteurs de sortie, j'ai choisi de classiques borniers à 4 voies au pas de 5,08 mm, conscient du fait que la plupart des rubans à LED que j'ai vus sont équipés de connecteurs plus petits. Comme je ne savais pas si ces connecteurs sont répandus ou encore si tous les rubans sont câblés de la même façon, j'ai opté pour une solution plus flexible. Vous pourrez aisément réaliser un adaptateur avec un câble d'alimentation de lecteur de disquettes par exemple (**fig. 3**).

Pour le connecteur d'alimentation, vous pourrez utiliser au choix un connecteur jack ou un bornier. La diode D4 apporte une protection rudimentaire contre les inversions de polarité. Un connecteur à 5 voies vous permettra d'utiliser les pattes libres du micro pour ce que vous voulez. Ce dernier sera complété d'un côté par

une rangée de points de test ; le connecteur de programmation *in situ* (ICSP) se trouvera de l'autre côté. Tous trois donnent accès aux pattes du micro ainsi qu'à toutes les alimentations. Il y a également une LED reliée à RC4 et un bouton. Grâce au chargeur de démarrage et à l'interface USB, le programme d'application sera facile à changer, ce qui fait de cette carte une excellente plateforme de prototypage rapide.

Tous les composants (sauf le module Wi-Fi) sont des composants traversants standard. Commencez par monter le module Wi-Fi qui, à cause de ses nombreuses connexions, sera peut-être un peu délicat à positionner. Vous remarquerez que les régulateurs de tension sont sur le ventre (**fig. 4**, ou alors côté cuivre). Si vous n'utilisez pas le boîtier suggéré et que vous les montez verticalement, vous pourrez facilement les doter d'un dissipateur (facultatif). Si vous souhaitez que la carte tienne dans le boîtier, il faudra monter les MOSFET sur le dos. Si vous voulez que les LED traversent le boîtier, vous devriez commencer par y percer les trous, ce qui facilitera l'ajustement de la longueur de leurs pattes. La LED D5 (reliée à RC4) et le poussoir ont été tous deux positionnés exactement sous une entretoise afin que vous n'ayez pas à vous embêter à mesurer avant de percer le boîtier. Ce dernier possède un emplacement à pile dans lequel vous pourrez voir les LED donnant l'état du module Wi-Fi, à condition de les monter côté cuivre.

Le poussoir devra être coiffé d'un capuchon de hauteur adéquate. Un modèle de 16 mm donne un ajustement presque parfait avec la face supérieure du boîtier.

Il vous faudra couper les quatre entretoises qui maintiennent le boîtier afin de pouvoir y loger le PCB. C'est facile à faire avec un gros foret. N'en enlevez pas trop sinon vous ne pourrez plus fermer le boîtier avec les vis. Vous devriez également retirer les deux languettes d'immobilisation de la pile.

Le module Wi-Fi possède un petit connecteur pour une meilleure antenne (**fig. 5**). Si vous l'utilisez, contentez-vous de la laisser pendouiller.

Programmation de la carte

Le micrologiciel téléchargeable pour ce projet [6] contient le chargeur de démarrage, l'application pour ruban à LED RVB et les réglages des fusibles dans un seul fichier HEX. Chargez-le dans le micro à l'aide d'un programmeur

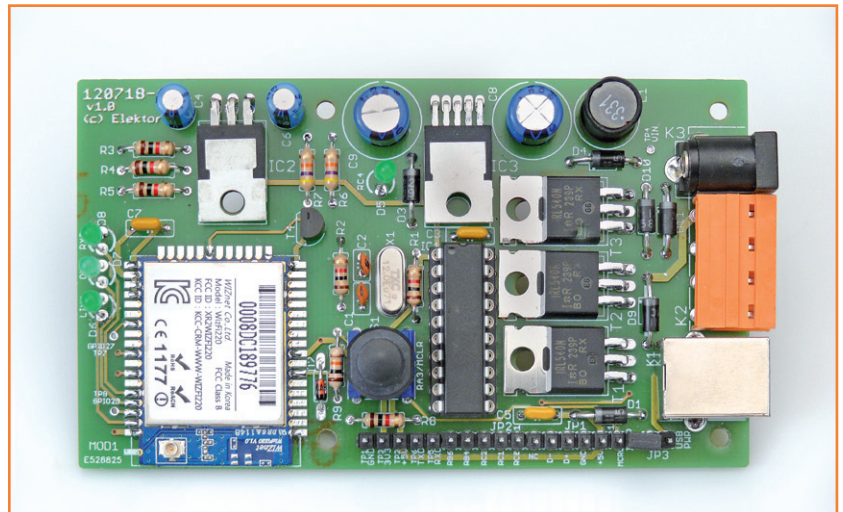
adéquat (PICKIT, ICD ou autre) et voilà. Une fois le chargeur de démarrage programmé dans le micro, vous pourrez y charger vos propres applications. Voici la procédure :

- hors tension, assurez-vous que JP3 n'est pas positionné
- reliez la carte à un port USB libre de votre PC
- maintenez S1 appuyé et positionnez JP3. J'ai remplacé JP3 par un poussoir normalement fermé (les micro-interrupteurs sont parfaits). Si vous suivez mon choix, appuyez brièvement sur le micro-interrupteur pour démarrer le chargeur de démarrage.
- le PC (*Windows*, *Linux* ou *Mac*) devrait détecter la carte comme un périphérique HID. Sur un PC *Windows*, il vous faudra pointer le système vers le fichier INF contenu dans l'archive la première fois.
- démarrez sur le PC l'outil *HIDBootloader* fourni dans l'archive (*Windows*, vous trouverez les versions *Linux* ou *Mac* sur le site de *Microchip*) (**fig. 6**). La carte devrait être détectée instantanément.
- sélectionnez votre fichier HEX puis cliquez sur le bouton de programmation
- pour sortir du chargeur de démarrage et démarrer l'application, cliquez sur le bouton *Reset* ou coupez puis rétablissez l'alimentation de la carte sans maintenir S1 enfoncé.

Configuration du module Wi-Fi

Dans [2] j'avais montré comment configurer le module *WizFi220* via un port série à l'aide de commandes AT. Cette technique est utile si p. ex. vous souhaitez que le micro reconfigure le module au vol. Elle permet également la configuration d'options qui ne sont pas accessibles autrement. La carte présentée ici possède un mode spécial qui vous permettra de configurer le module de cette manière. Le module Wi-Fi peut être configuré plus facilement si vous souhaitez seulement vous connecter à un réseau Wi-Fi existant. Dans ce cas, vous devrez mettre le module dans le mode *Limited Access Point* (LAP). La carte est aussi capable de le faire.

Quelle que soit la méthode, commencez par mettre la carte en mode USB vers série : reliez RC2 au +5 V (p. ex. avec un fil sur le connecteur d'extension) puis reliez la carte à un port USB libre de votre PC. La LED D5 commencera



à clignoter et vous pourrez envoyer des commandes AT sur le port COM virtuel créé (c'est à vous de trouver comment demander à votre système d'exploitation de vous donner son numéro ou son nom). Si le module Wi-Fi est connecté à un AP (c.-à-d. quand les LED *LINK* et *OK* sont toutes les deux illuminées) tapez "+++" dans votre programme d'émulation de terminal pour passer le module en mode commande (la LED *OK* devrait s'éteindre). Suivez maintenant la procédure décrite dans [2] pour configurer le module.

Si vous souhaitez placer le module en mode LAP, il vous faudra de la dextérité. Regardez attentivement clignoter la LED et habituez-vous à son rythme. Quand vous serez prêt, appuyez sur S1 alors que la LED est éteinte et maintenez-le appuyé pendant deux clignotements complets. Ne relâchez le poussoir que lorsque la LED s'éteint après le deuxième clignotement. Si vous l'avez fait correctement, les trois LED d'état

Figure 4.

La révision 1.0 de la carte assemblée. Pour la rev. 1.1, j'ai déplacé quelques composants pour améliorer l'incorporation dans le boîtier.

Figure 5.

Trois antennes 2,4 GHz, celle de *Winizen* (au-dessus) et deux autres aimablement fournies par les filles de 2J (www.2j-antennae.com).

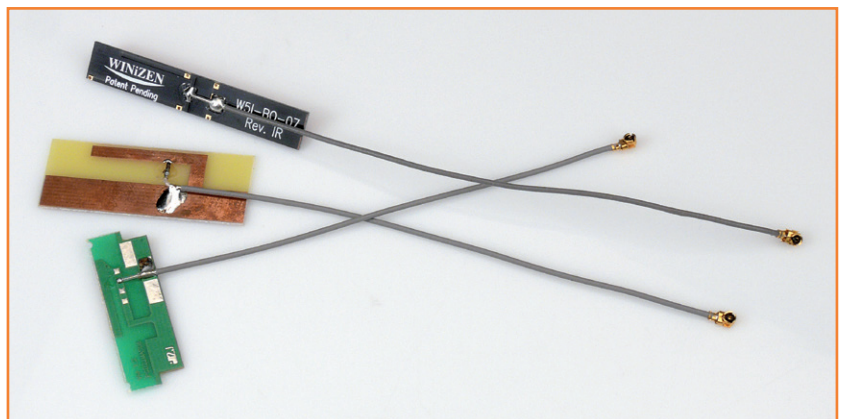


Figure 6.

Voici l'outil à utiliser pour charger une nouvelle application dans la carte de commande Wi-Fi.

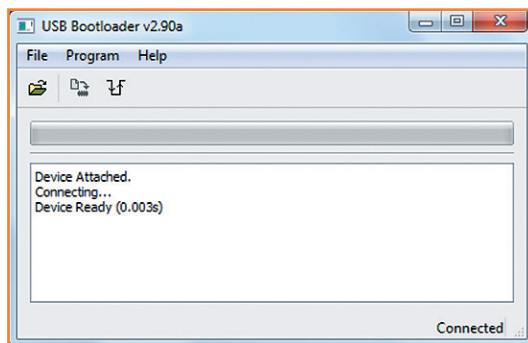


Figure 7.

La page de configuration du module WizFi accessible en mode *Limited Access Point* qui vous rendra la vie plus facile

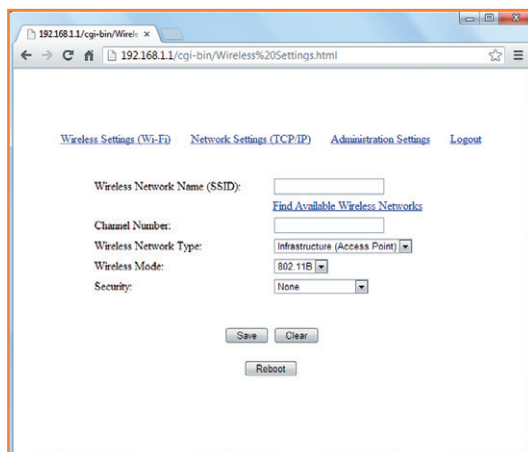
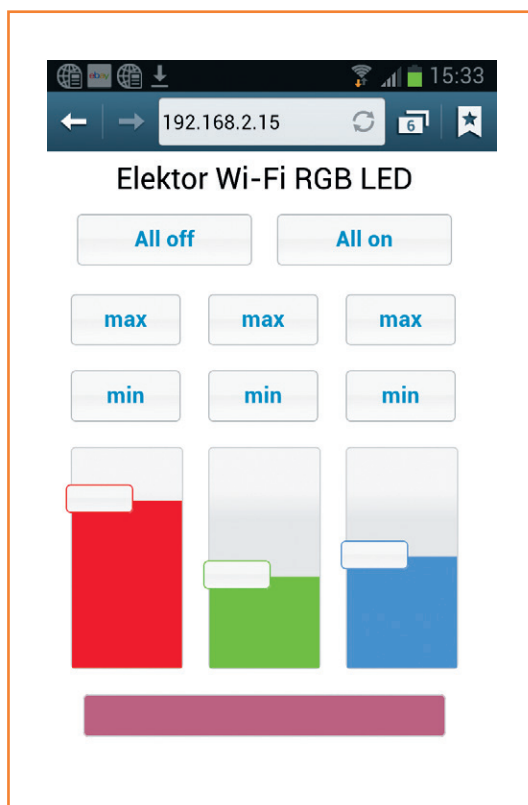


Figure 8.

La page de notre carte de commande Wi-Fi vue depuis un smartphone Android. Le bleu est-il à moitié allumé ou à moitié éteint ?



du module Wi-Fi devraient, après une bonne seconde, clignoter rapidement plusieurs fois : le module est en mode LAP. (Si vous attendez trois clignotements, le module chargera les paramètres de sortie d'usine ; un moyen rapide de vous sortir du pétrin.) Si vous avez relié la carte à un terminal, vous verrez alors le message suivant :

IP	SubNet	Gateway
192.168.1.1:	255.255.255.0:	192.168.1.1

[OK]

Assurez-vous maintenant que le module est à portée de l'AP que vous souhaitez utiliser. Avec un PC, un téléphone tactile ou une tablette, vous pourrez lister les points d'accès disponibles. Si tout se passe bien, vous devriez en voir un nommé "WizFiAPxxxx", xxxx étant un nombre. Connectez-vous à ce point d'accès — il est ouvert et vous n'aurez pas besoin de mot de passe — et pointez votre navigateur à l'adresse 192.168.1.1. Vous devriez voir une page similaire à celle de la **figure 7**. Cliquez sur lien "Find Available Wireless Networks". Une liste des réseaux à portée apparaîtra et vous pourrez sélectionner celui que vous voulez. Cliquez sur "Save And Continue". Cela vous ramènera au premier formulaire qui affichera désormais les détails relatifs au réseau sélectionné. En fonction des paramètres de sécurité de ce réseau, vous devrez peut-être alors rentrer un mot de passe. Cliquez enfin sur "Save" pour enregistrer la configuration. Cliquez sur le lien "Network Settings (TCP/IP)" et remplissez le formulaire. Je préfère utiliser une adresse IP statique pour le module de manière à toujours connaître son adresse mais ce n'est pas obligatoire. Dans la champ "S2W Connection method", tapez "1,1,,80". Cela transformera le module en une passerelle série à l'écoute sur le port 80, le port HTTP par défaut utilisé par les navigateurs web. Là aussi vous pouvez utiliser une valeur différente. Cliquez sur le bouton "Save" quand vous aurez terminé.

Le lien "Administration Settings" vous permettra de saisir un mot de passe pour le module. Je n'ai pas utilisé cette option.

Quand vous aurez terminé la configuration, cliquez sur le lien "Logout". Vous verrez apparaître le message "Rebooting..." et perdrez la connexion Wi-Fi. Le module WizFi redémarrera et tentera de se connecter directement au réseau choisi. Si tout se passe bien, vous verrez les LED *LINK* et *OK* s'illuminer ; cela veut dire que

vous pouvez maintenant vous connecter à la carte. Saisissez l'adresse IP du module dans votre navigateur et patientez jusqu'à ce que la page de la **figure 8** apparaisse. Une fois la page chargée, vous pourrez commencer à faire joujou. J'ai trouvé amusant de commander de l'extérieur l'éclairage dans ma maison avec mon téléphone. Ce n'est pas l'utilisation que je prévois pour cette carte, j'ai l'intention de la relier à la porte motorisée de mon garage pour utiliser mon *smartphone* comme clé à distance. Il me sera alors possible de donner l'accès à quelqu'un d'autre quand nous sommes en voyage. À vous d'imaginer maintenant d'autres applications pour cette carte !

(120718 – version française : Kévin PETIT)

Liens

- [1] Elektor Home Control :
www.elektor-labs.com/node/2325
- [2] shield Wi-Fi/Bluetooth :
www.elektor.fr/120306
- [3] JQuery(UI) :
<http://jquery.com> & <http://jqueryui.com>
- [4] la version Arduino :
www.elektor-labs.com/node/2373
- [5] FlowStone (dans ce même numéro) :
www.elektor.fr/130064
- [6] micrologiciel, fichiers Eagle et commandes :
www.elektor.fr/120718

Liste des composants

Résistances (5%, 0,25W)

R1,R2,R3,R4,R5,R8 = 1 kΩ
R6,R7 = 47 kΩ
R9 = 10 kΩ

Condensateurs

C1,C2 = 22 pF, céramique, 50 V au pas de 2,5 mm
C3,C5,C7 = 100 nF, Z5U, 50 V au pas de 5 mm
C4,C6 = 10 µF 63 V, radial au pas de 2,5 mm
C8 = 330 µF 16 V, radial au pas de 3,5 mm
C9 = 100 µF 50 V, radial au pas de 3,5 mm

Inductances

L1 = 330 µH 1 A au pas de 5 mm, p.ex. Würth Elektronik type 7447452331

Semiconducteurs

D1,D3,D4,D9,D10,D11 = 1N5819
D2 = zener 3 V, p.ex. BZX79-C3V0
D5,D6,D7,D8 = LED vertes 3 mm
IC1 = PIC18F14K50-I/P

IC2 = MCP1825S-3302E/AB

IC3 = LM2575T-5.0/NOPB

T1,T2,T3 = IRL540

T4 = BC547

Divers

MOD1 = WizFi220 avec antenne, Elektor 130076-92

JP1 = barrette 6 voies au pas de 2,54 mm

JP2 = barrette 5 voies au pas de 2,54 mm

JP3 = barrette 2 voies au pas de 2,54 mm

Cavalier pour JP3

K1 = embase USB-B

K2 = connecteur 4 voies coudé au pas de 5,08 mm, p.ex. MSTBA4

K3 = embase DC

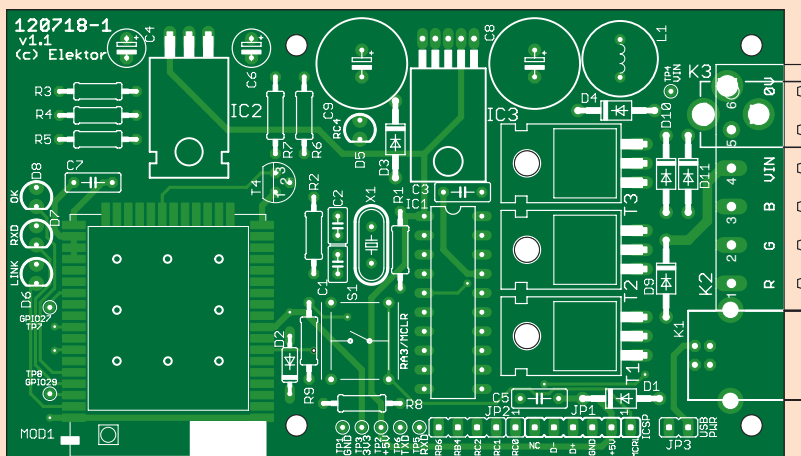
support DIP à 20 pattes pour IC1

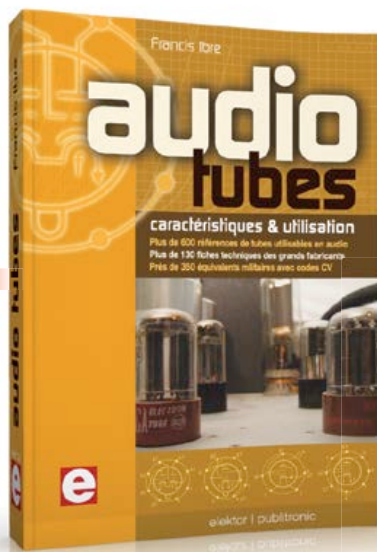
S1 = poussoir Multimec type RA3FTL6 avec capuchon type S09-16.0

X1 = résonateur à quartz 12 MHz en boîtier HC49/S

Boîtier Hammond type 1593QGY

circuit imprimé Elektor 120718-1





1

Caractéristiques & utilisation

1 Audio tubes

- Plus de 600 tubes utilisables en audio (préampli, puissance, redresseur) dans 9 tableaux descriptifs (type, équivalence...)
- Plus de 130 fiches techniques des grands fabricants
- Près de 350 équivalents militaires avec codes CV
- Recommandations pour interpréter les fiches techniques et en tirer pleinement parti (test, mesure, remplacement des tubes)
- Exemples concrets de calculs, de schémas et d'illustrations

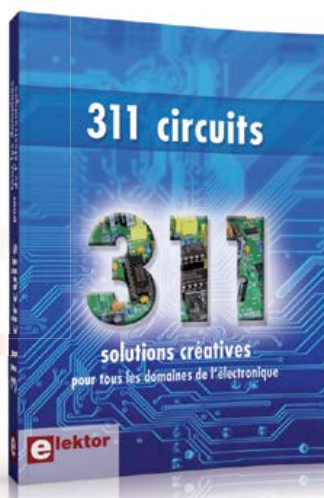
576 pages - ISBN 978-2-86661-174-3 - 60,00 €

Solutions créatives

pour tous les domaines de l'électronique

2 311 circuits

Cet ouvrage est un trésor : il réunit 311 schémas d'électronique analogique, logique ou numérique, des programmes, des liens vers des sites internet, des tableaux de caractéristiques de composants et des dessins de circuit imprimé. Il est le onzième volume de la collection « 3xx circuits ». Ses deux tables des matières alphabétique et thématique vous permettent de trouver rapidement et facilement parmi les 311 articles proposés ceux qui répondront à vos besoins.



2

Ces articles viennent des numéros doubles récents de la revue Elektor. 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.

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

Kit complet avec circuit imprimé, afficheur et microcontrôleur

3 Détecteur de rayonnement

Il ne faut guère plus qu'une photodiode PIN et un amplificateur de capteur adapté pour mesurer un rayonnement radioactif. Nous présentons ici un préamplificateur optimisé couplé à un compteur particulier : un microcontrôleur qui gère la durée de la mesure et affiche la fréquence des impulsions en « counts per minute ».

Réf : 110538-71 - 39,95 €

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

4 Embarquez Linux

Linux est partout, même dans certaines machines à café. Souvent, l'électronicien tenté d'adopter ce système



3



4

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)

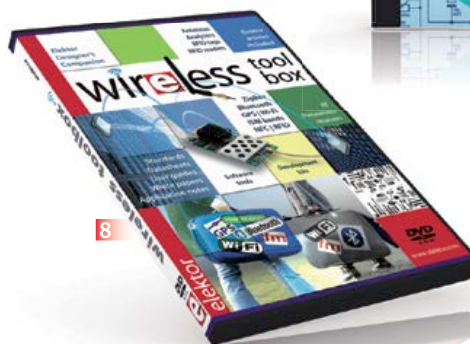
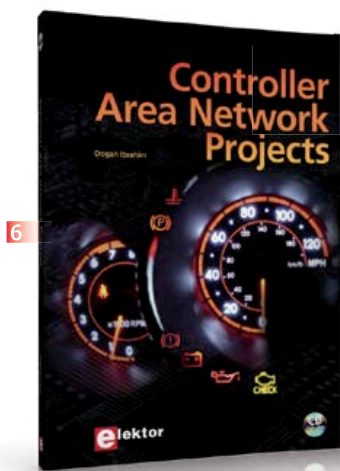
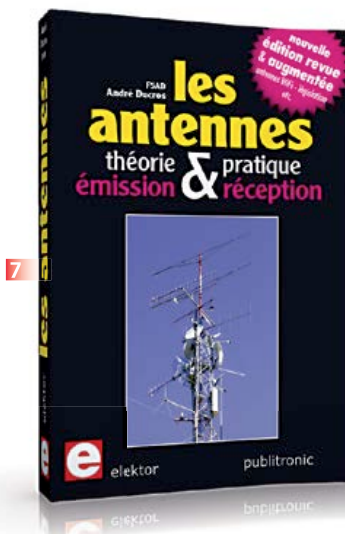
Réf. : 120026-91 • 64,95 €

Kit (circuits imprimés & composants)

5 Tapir – détecteur ultrasensible d'électrosmog

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 €



Avec CD-ROM

6 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 donne 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. **CD-ROM inclus** Ce livre comporte un CD-ROM gratuit qui contient une version de démonstration spéciale du compilateur mikroC. **ISBN 978-1-907920-04-2 • Langue : Anglais • 259 pages • 34,50 €**

théorie & pratique – émission & réception

7 Les Antennes

Cette bible des antennes devient l'ouvrage de référence pour les radioamateurs, les techniciens et les

ingénieurs. La première partie traite de la propagation des ondes dans l'espace et sur les lignes ainsi que des caractéristiques fondamentales des antennes (gain, rayonnement, courant, tension...). Cette étude théorique est suivie de réalisations pratiques, entre autres les antennes filaires, les antennes à gain, et les antennes à large bande et multibandes. La dernière partie est consacrée aux ultimes réglages : adaptation des impédances, appareils de mesure, conseils de sécurité (poussée du vent, résistance des matériaux, pylônes et haubans, foudre...).

470 pages – ISBN 978-2-86661-165-1 – 49,50 €

plus de 90 articles d'Elektor en français en bonus DVD-ROM Wireless

8 Toolbox

Sur ce DVD-ROM, vous trouverez des documents techniques et des outils pour libérer vos propres systèmes électroniques de leurs fils. Selon la distance à couvrir, le choix est vaste : quelques cm avec la communication en champ proche (NFC) et l'identification par radiofréquences (RFID), des dizaines de mètres avec Bluetooth, Wi-Fi et ZigBee, des milliers de km avec la réception de données par GPS. Le plus difficile est de s'y retrouver dans la jungle des normes. D'où l'utilité de ce DVD, avec

sa collection de normes, de protocoles standard et propriétaires (p. ex. MiWi de Microchip, SimpliciTI de Texas Instruments), sa revue des plages de fréquences utilisées (ISM p. ex.) et bien davantage.

ISBN 978-90-5381-268-6 – 32,50 €

Tous les articles d'ELEKTOR de l'année 2012 en français, anglais, allemand, espagnol et néerlandais

9 DVD Elektor 2012

Ce DVD-ROM réunit tous les articles d'ELEKTOR, le mensuel d'électronique et de micro-informatique appliquées, parus au cours de l'année 2012. Il contient non seulement le texte des articles ainsi que les schémas, mais aussi tous les dessins des circuits imprimés, sous forme de fichiers à haute résolution.

ISBN 978-90-5381-273-0 • 27,50 €

Informations complémentaires et gamme complète sur :

www.elektor.fr/e-choppe

Elektor/Publitrone SARL

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

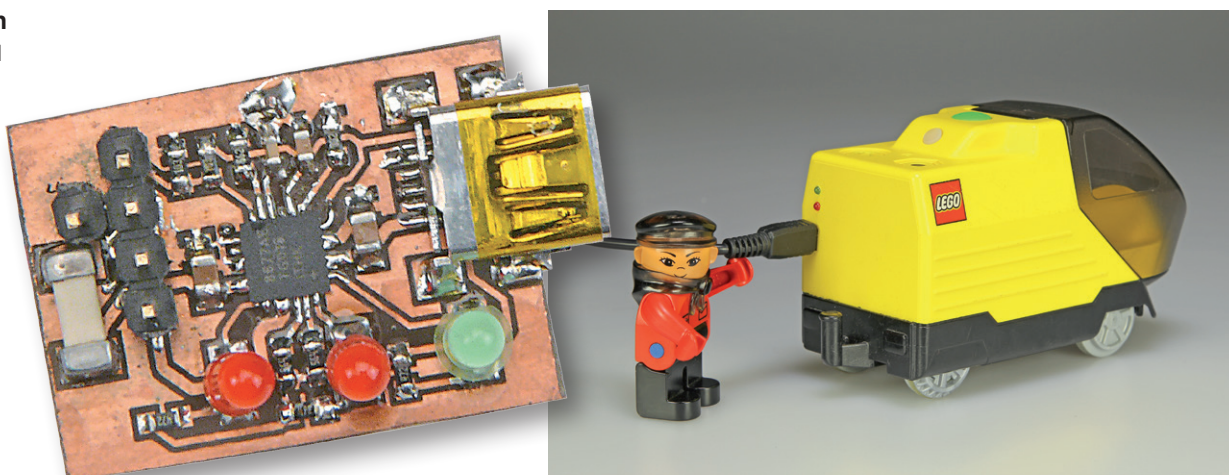
Tél. : +33(0)1.49.19.26.19 – Fax : +33(0)1.49.19.22.37

@ : ventes@elektor.fr

recyclage facile d'accumulateurs

Chargeur d'accus lithium-ion sans microcontrôleur

Fons Janssen
Maxim Integrated
(Pays-Bas)



La réutilisation des accus lithium-ion d'appareils réformés est difficile parce que ce genre d'accumulateur est souvent chargé dans l'appareil lui-même, si bien qu'il n'y a pas de chargeur extérieur disponible. Heureusement, il n'est pas difficile de construire soi-même un chargeur d'accus Li-ion récupérés (ou neufs).

Vous avez sûrement quelques vieux appareils qui traînent, alimentés par un accu Li-ion. Ces dernières années, ce type d'accu est utilisé dans de nombreux appareils portables, parce qu'il est facile à fabriquer dans toutes les formes et toutes les dimensions et qu'il offre une capacité relativement forte (par rapport aux NiMH et NiCd).

Qu'allez-vous faire de ce vieux lecteur MP3 ou de ce téléphone que vous avez remplacés depuis longtemps par des modèles meilleurs ? L'électronique est le plus souvent impossible à utiliser pour une autre application, mais l'accumulateur peut encore servir, ne serait-ce que dans un jouet. Inventif comme l'est tout électronicien, vous allez bien trouver le moyen de monter et raccorder cet accumulateur recyclé. C'est ainsi que l'auteur a

monté un accumulateur Li-ion dans un petit train Lego, en remplacement de trois piles bâtons (voir la photo du titre).

Schéma

Le chargeur proposé dans cet article est organisé autour d'un circuit intégré chargeur de Maxim, destiné aux accus Li-ion (MAX8677A, voir le schéma synoptique de la **fig. 1**). Le fonctionnement de ce circuit est complètement autonome, ce qui évite le recours à un microcontrôleur (et à un logiciel !). Il suffit de quelques LED pour signaler l'état du processus de charge.

Le circuit intégré est très souple et équipé d'un « *smart power selector* » (**fig. 2**), une sorte d'ajustage intelligent, composé de trois interrupteurs électroniques qui dirigent les courants de

charge et décharge dans les circuits adéquats. Ainsi le CI peut utiliser la source d'énergie disponible pour charger l'accumulateur aussi bien que pour alimenter l'application. Si l'application a besoin de plus de courant que le chargeur peut en délivrer, le CI peut appeler l'accu à la rescousse. Si aucune source de courant n'est disponible, il va de soi que l'application est alimentée par l'accumulateur.

Le CI peut être alimenté par un port USB via les broches 15 et 16 (USB). La consommation de courant est limitée à 500 mA (le maximum d'un port USB). L'alimentation peut aussi provenir d'un adaptateur secteur via les broches 2 et 3 (DC), auquel cas la limitation d'intensité peut être rehaussée jusqu'à 2 A.

Dans le circuit de chargeur présenté ici (**fig. 3**), nous utilisons l'entrée DC, ce qui nous donne plus de latitude pour fixer les différentes limites. La plage de tension d'entrée autorisée va de 4,1 à 6,6 V. Si la tension s'élève trop, le CI déconnecte son entrée pour éviter la surchauffe. Le CI peut survivre à des pointes de tension de 14 V au maximum.

L'indication de charge passe par les LED D1, D2 et D3. Les trois états suivants sont affichés :

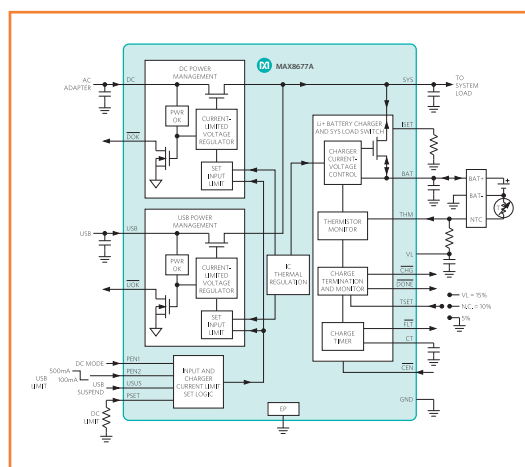


Figure 1.
Schéma synoptique des entrailles du MAX8677A.

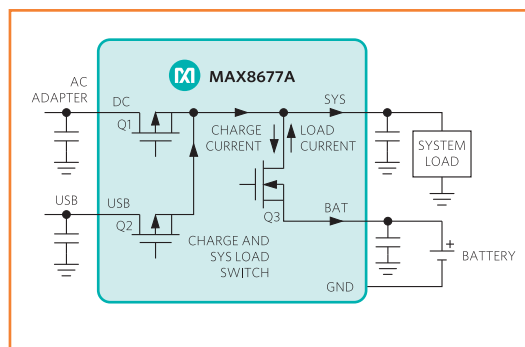


Figure 2.
Le « smart power selector » aiguille correctement les courants de charge et de décharge.

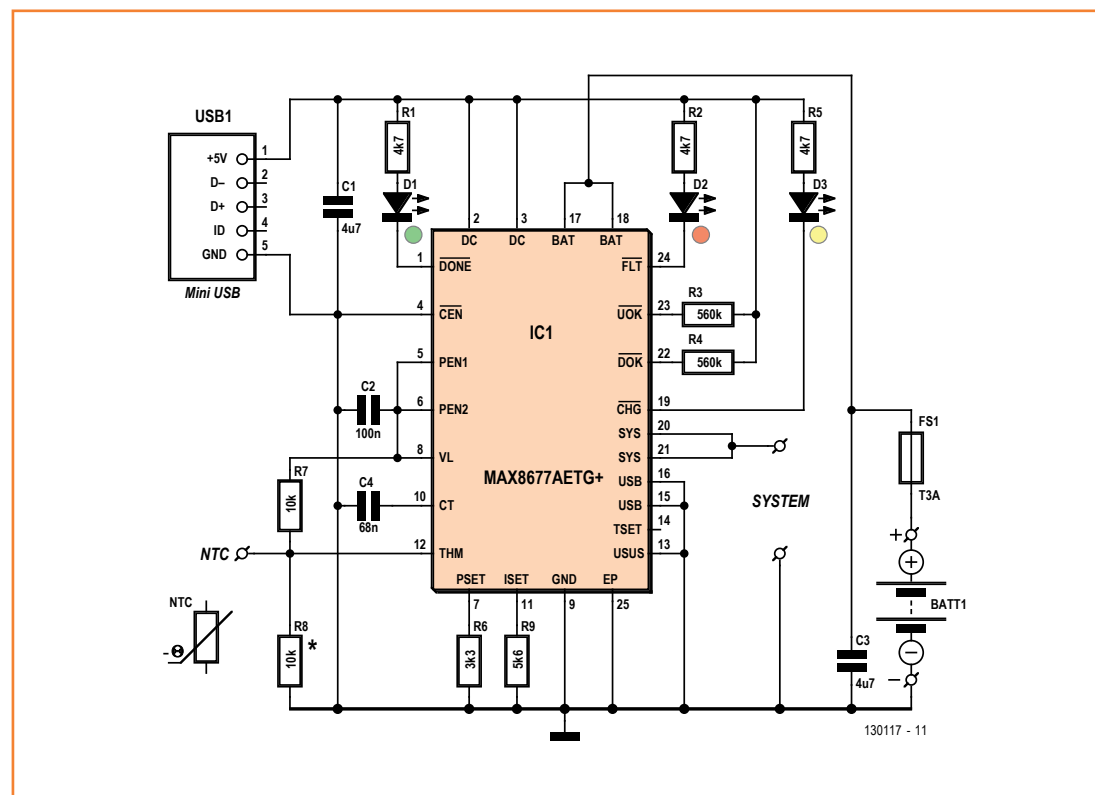
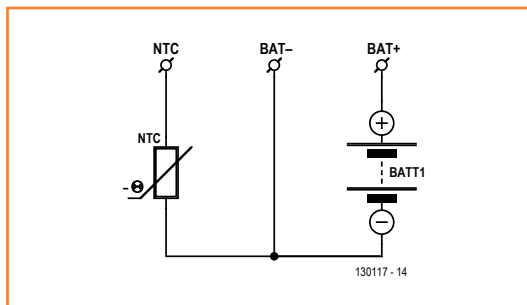


Figure 3.
Le circuit complet du chargeur est constitué principalement du MAX8677A et d'un connecteur mini-USB.

Figure 4.

La plupart des accu Li-ion à trois bornes ont une thermistance CTN connectée comme ci-dessus.



l'accu est en cours de charge (D3), l'accu est à pleine charge (D1) ou l'accu est défectueux (D2). On peut fixer à ce CI deux limites d'intensité : une pour le courant de charge maximal et une pour l'intensité d'entrée maximale. Il est évident que la première doit toujours être supérieure à

la deuxième. Si ce n'est pas le cas, le courant de charge n'atteindra jamais le maximum programmé et plafonnera au maximum du courant d'entrée.

Chaque limite est programmée par une résistance :

- Courant de charge maximal = $ICHGMAX = 3000/RISET = 3000/R9 = 3000/5k6 = 535 \text{ mA}$
- Courant d'entrée maximal = $IDC MAX = 3000/RPSET = 3000/R6 = 3000/3k3 = 909 \text{ mA}$

D'autres valeurs peuvent être choisies en fonction de la puissance délivrée par l'adaptateur secteur, de la consommation de l'application et du courant de charge choisi. Le CI supporte un courant de charge de 1,5 A.

Nous utilisons un connecteur mini-USB, ce qui permet l'alimentation par des chargeurs ordinaires. Cela nous assure aussi que la tension d'alimentation est de 5 V. Le courant maximal doit être accordé avec celui de l'adaptateur utilisé. Si cet adaptateur peut débiter au moins 1 A, le circuit fonctionnera parfaitement.

Thermistance ou pas ?

Les accu sont souvent équipés d'une thermistance CTN (à coefficient de température négatif), pour éviter la charge par une température trop haute ou trop basse. L'accu a pour cela trois bornes : un pôle positif BAT+, un pôle négatif BAT- et une connexion pour la thermistance (**fig. 4**). Faites bien attention, parce que certains accu à trois bornes ont une résistance ordinaire qui sert à l'identification. Sa valeur est fixe et ne varie pas avec la température de l'accu.

Quand on utilise la CTN, elle doit être connectée entre la borne THM et la masse (par la borne BAT-). D'autre part une résistance (R7) est connectée entre la borne THM et une tension de référence VL, ce qui constitue un diviseur de tension. Si la résistance de R7 est égale à celle de la thermistance à 25 °C, la tension sur la borne THM sera égale à 0,5 VL. Si la température monte (ou descend), la résistance de la CTN descend (ou monte) et avec elle la tension de la borne THM. Le CI n'autorise la charge que si la tension se situe entre 0,28 VL et 0,74 VL. Avec les thermistances courantes, ces valeurs correspondent à des températures comprises entre 0 °C et 50 °C. En l'absence de thermistance, il faut installer R8, ce qui fixe la tension de THM à 0,5 VL.

Liste des composants

Résistances

(CMS 0603, sauf mention contraire)
R1,R2,R5 = 4,7 kΩ
R3,R4 = 560 kΩ
R6 = 3,3 kΩ
R7 = 10 kΩ
R8 = 10 kΩ (seulement en l'absence de CTN dans l'accu)
R9 = 5,6 kΩ

Condensateurs

(CMS 0603, sauf mention contraire)
C1,C3 = 4,7 μF (CMS 0805)
C2 = 100 nF
C4 = 68 nF

Semi-conducteurs

D1 = LED verte, 3 mm
D2 = LED rouge, 3 mm
D3 = LED jaune, 3 mm
IC1 = MAX 8677AETG+ (TQFN 24 broches)

Divers

USB1 = connecteur mini-USB pour CI, montage en surface (p.ex. Molex 67803-8020, RS-Components réf. 720-6618)
FS1 = fusible CMS, valeur suivant l'accu (p.ex. LittleFuse nanofuse 3 AT, Farnell réf. 1596930RL)
circuit imprimé 130117-1 disponible sur [2]

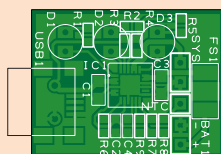


Figure 5.

Le circuit imprimé dessiné pour ce projet est aussi petit que possible, de façon à faciliter d'autant l'installation dans un appareil existant.

Astuces pour le raccordement

Les accumulateurs de téléphone sont équipés en standard d'un circuit qui les protège des surintensités et de la décharge profonde. Si vous voulez utiliser un élément séparé tiré d'une batterie d'ordinateur portable, vous devrez faire un circuit de protection vous-même. Le circuit d'origine de la batterie est destiné à protéger l'ensemble et ne peut pas servir pour un élément séparé. Un simple fusible (repéré par FS1 sur le schéma ; modèle monté en surface sur la platine) apporte une très bonne protection contre la surcharge et convient très bien pour les éléments séparés. Par contre, un simple fusible n'apporte aucune protection contre la décharge profonde. Si ce type d'accumulateur est déchargé trop profondément, il risque de rester dans cet état. Cela peut se produire si l'accu alimente trop longtemps une charge ohmique comme une ampoule à incandescence. La plupart des applications s'arrêtent quand la tension tombe en dessous d'une valeur déterminée, ce qui empêche que l'accumulateur continue à se décharger. C'est l'application qui décide si un élément séparé est utilisable avec seulement un fusible.

Réalisation

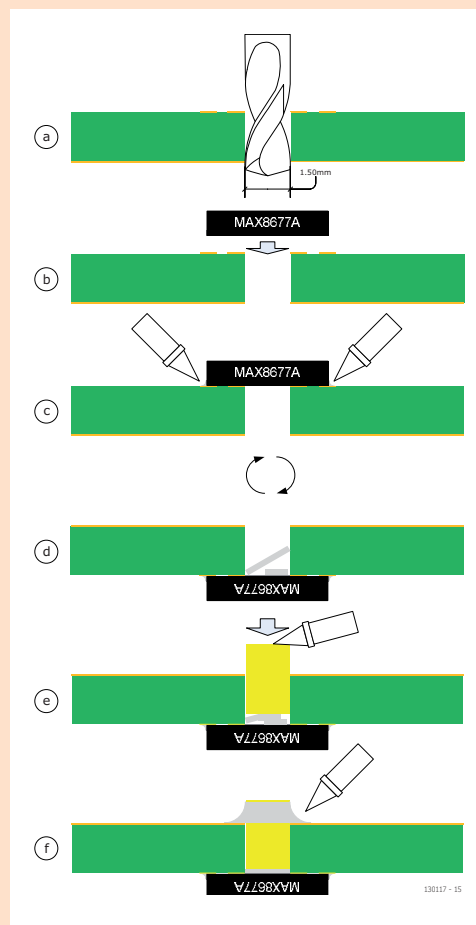
Nous avons conçu pour ce projet un circuit imprimé compact avec beaucoup de composants montés en surface (**fig. 5**). Les dimensions réduites permettent de monter facilement le circuit dans l'appareil. Les fichiers de la platine (réalisée avec *DesignSpark*) sont disponibles en téléchargement gratuit [2].

Le soudage des CMS exige une certaine adresse et de l'expérience. Le circuit intégré en boîtier TQFN exige un four à refusion, parce que les broches et la pastille du substrat se trouvent en dessous du boîtier de 4 x 4 mm. En fait, l'auteur a imaginé une méthode pour souder facilement le circuit intégré, voir l'encadré. Pour les LED et les connexions de l'accumulateur et de la charge, des trous sont prévus sur le circuit imprimé, ce qui facilite le raccordement par fils. Le connecteur mini-USB a deux pions en plastique qui tombent dans les trous correspondants pour assurer l'alignement. Si on ne veut pas monter de connecteur mini-USB, on peut utiliser les trous pour le raccordement de l'alimentation.

Attention : dans ce cas, l'alimentation doit se faire sous 5 V.

(130117 – version française : Jean-Paul Brodier)

Soudage manuel des TQFN



Le soudage du circuit intégré avec un fer à air chaud est possible pour un soudeur expérimenté, mais un four à refusion reste le plus pratique. La méthode décrite ici fonctionne aussi avec un fer à souder ordinaire, et même sur un circuit imprimé maison sans trous métallisés. Placez une pastille au milieu du circuit intégré, au lieu des neuf du dessin proposé. Percez cette pastille à un diamètre de 1,5 mm (a). Placez ensuite la puce (b) et soudez les contacts en périphérie (c). La tresse à dessouder permet de nettoyer la zone (l'auteur se sert d'une loupe binoculaire pour bien y voir). Si les contacts sont bons sur la face supérieure, retournez la platine

et laissez tomber quelques petits morceaux de soudure dans le trou (d). Prenez ensuite un petit morceau de fil de cuivre rigide (le fil de 2,5 mm² passe bien dans un trou de 1,5 mm), faites-lui à la lime une section bien plane à un bout. Introduisez cette extrémité dans le trou et chauffez le fil au fer à souder (e). Le morceau de fil de cuivre devient vite assez chaud pour faire fondre les morceaux de soudure au fond du trou, en conséquence de quoi le fil descend et vient au contact du substrat de la puce. Pour finir, soudez le morceau de fil au plan de masse du côté soudures de la platine (f). Vous avez maintenant une liaison électrique et thermique efficace entre le substrat de la puce et le plan de masse du circuit imprimé.

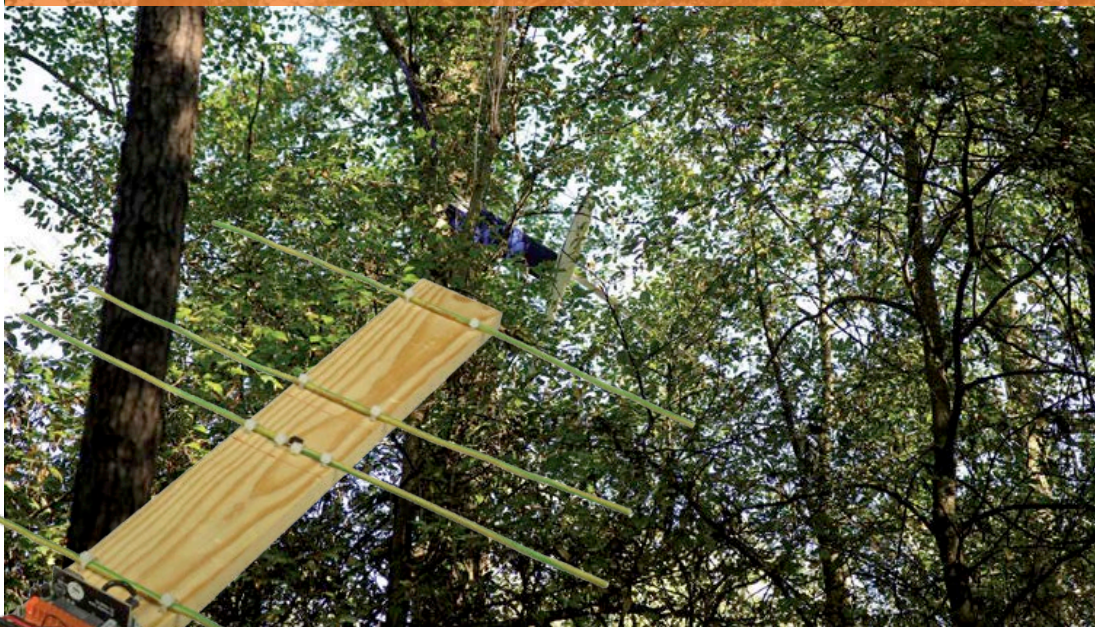
Liens

- [1] <http://datasheets.maximintegrated.com/en/ds/MAX8677A.pdf>
- [2] www.elektor.fr/130117

modèle réduit perdu ?

Robert Budniak
(Australie)

droit sur le site du crash



Tout radiomodéliste aérien sait qu'un jour ou l'autre, il risque de perdre le contrôle de son appareil et, s'il a échappé à sa vue, que le point de chute peut se révéler difficile à localiser. Hautes herbes, arbres, terrain accidenté ne facilitent pas la récupération. Une aide immédiate et efficace : le radiogoniomètre et une balise embarquée. Mieux encore : les construire soi-même.

Pour localiser un modèle réduit perdu, il existe bien des systèmes. J'ai voulu en construire un qui offre au moins les caractéristiques suivantes :

- émetteur léger pour l'avion
- batterie de secours en cas de rupture de connexion avec la source principale

- modules radio du commerce dans une bande UHF autorisée
- portée d'au moins 200 m
- récepteur portable assez directif pour localiser le modèle réduit.

J'ai réalisé un système d'émission et de réception qui répond à ces critères. Si vous êtes un as de la construction et du pilotage en vol mais que vous n'êtes pas très fort en électronique, vous pouvez toujours proposer ce projet comme activité en club.

La balise

Chaque pays dispose d'une série de radiofréquences allouées aux applications industrielles, scientifiques et médicales (ISM), telles que définies

Caractéristiques techniques

- réalisation à bas coût
- circuit imprimé TX en CMS, RX en technologie classique
- convient à la plupart des modules radio ISM 315/433 MHz
- indicatif personnel programmable sur chaque TX
- portée minimum : 200 m (600 pieds)
- portée maximum selon le terrain et les modules utilisés
- antenne directionnelle Yagi à 4 éléments sur le RX

par la norme EN55011 pour l'Union européenne, parfois combinées aux appareils à courte portée (SRD). L'une d'elles est la bande de 433,920 MHz $\pm 0,2 \%$ pour être précis. Aux États-Unis, la bande correspondante est celle de 315 MHz. En raison de la forte demande de modules simples et peu coûteux, plusieurs fabricants en ont mis sur le marché. Le module que j'ai utilisé dans ce projet opère en modulation d'amplitude par tout ou rien. Différentes firmes respectent une standardisation du brochage de ces modules qui nous libèrent du travail en radiofréquence (voir ma suggestion dans la liste des composants).

Le **figure 1** montre que le module RF se contente d'une broche DATA pour le débit binaire. Comme microcontrôleur, j'ai choisi un simple PICAXEE08M, aisément disponible. Il ne demande aucun appareil spécial pour le programmer et son langage s'apprend facilement. Que rêver de mieux pour des projets modestes en série limitée ?

Le code, visible dans le **listage 1**, ne compte que quelques lignes. Il produit trois salves à 500 Hz suivies d'une pause d'à peu près 2,5 secondes. On peut le copier, mais dans le cas d'une utilisation en groupe de plusieurs émetteurs proches, comme dans un club, mieux vaut modifier la ligne qui définit cette séquence.

La balise est normalement alimentée sur le récepteur de l'avion. Comme, lors d'un crash, souvent la batterie de l'avion se débranche, j'en ai ajouté une petite de secours, au LiPo, d'une capacité de 130 mA, comme sur les avions conçus pour le vol intérieur. On en trouve facilement auprès des fournisseurs en ligne.

D'habitude, on met des diodes pour sélectionner l'alimentation. Mais avec les tensions basses de ce projet, 5 V et 3,7 V, la chute, même limitée à 0,85 V d'une diode Schottky, abaisserait trop la tension. Une recherche m'a permis de découvrir des transistors pour petits signaux du type DTB123YK, T1 et T2, qui intègrent les résistances sur la puce. Ils marchent très bien en commutation des batteries, la chute de tension sur un transistor n'est que de 100 mV. L'inconvénient, c'est que quand la batterie principale est déconnectée, celle de secours prend d'office le relais. Or, il faut que la batterie de secours soit débranchée quand l'avion n'est pas en service, sinon elle se vide. Pour laisser en place l'émetteur balise en permanence – il est assez bon marché pour en avoir un dans chaque avion – je l'ai doté d'un port de charge combiné à un interrupteur marche/arrêt.

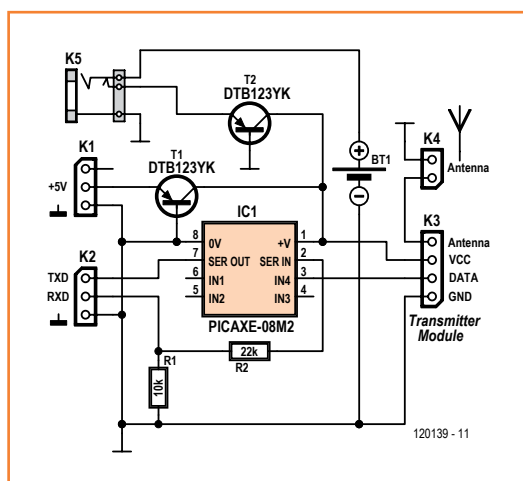


Figure 1.
Le schéma de l'émetteur,
guère plus qu'un
microcontrôleur PICAXE
programmé.

Listage 1. PICAXE TX code

```
main:
sound 4, (0,10,120,10,0,10,120,10,0,10,120,10,0,10)
high 4
pause 2300
goto main
```

J'ai donc utilisé une prise pour jack de 2,5 mm avec interrupteur mu par le jack correspondant. Si vous insérez le jack seul, la batterie est déconnectée. S'il est branché sur un chargeur, vous chargez la batterie.

Le chargeur

La **figure 2** montre une suggestion de chargeur. Les deux entrées commandent le mode. L'une sélectionne la tension régulée, 4,1 V ou 4,2 V avec

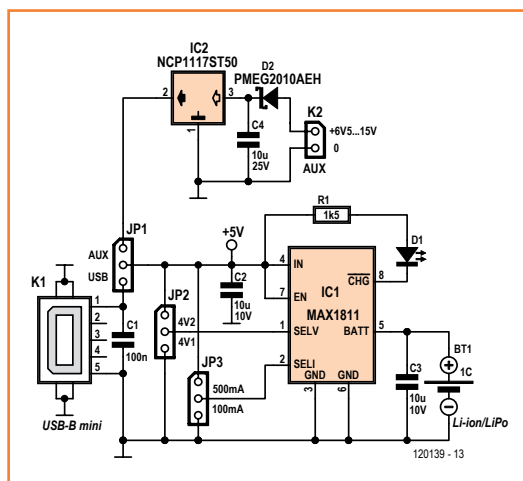


Figure 2.
Un exemple de chargeur
pour batterie LiPo/
li-Ion dans l'émetteur.
Selon la position de JP1,
l'alimentation est externe
(6,5 V à 15 V...) ou par USB.

Figure 3.
Construisez ce circuit inverseur si vous utilisez l'adaptateur FTDI TTL vers 232 entre les lignes TX/RX de l'émetteur et le microcontrôleur.

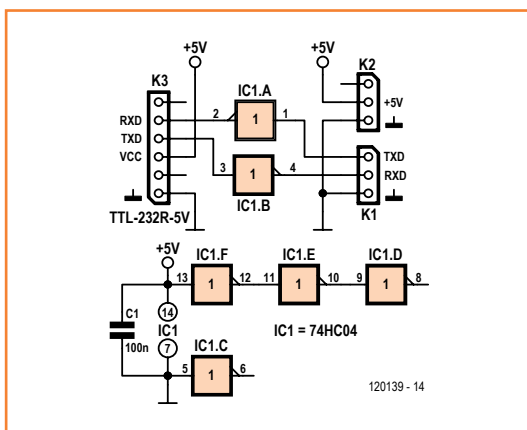
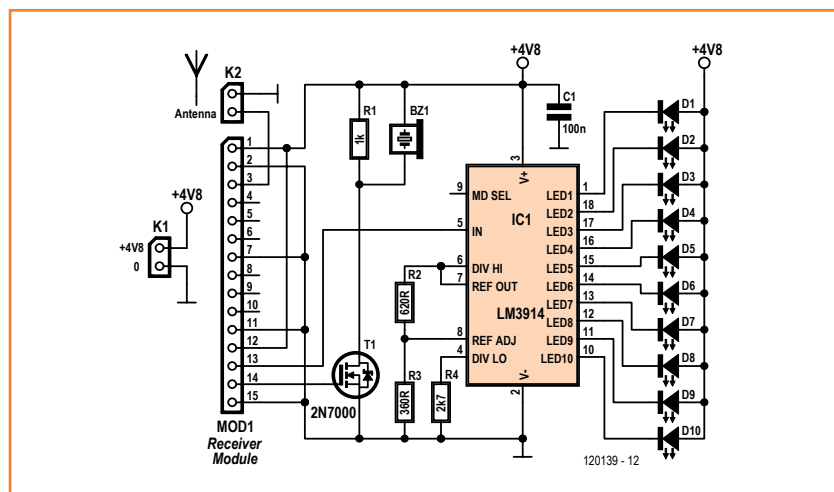


Figure 4.
Le radiogoniomètre retrouve une de nos vieilles connaissances, l'indicateur à colonne de LED LM3914. Le module récepteur à 433 MHz (315 MHz aux USA) est enfilé sur le connecteur MOD1.

le cavalier JP2 sur SELV. L'autre règle le courant de charge, 100 mA ou 500 mA avec le cavalier JP3 par SELI. Un avantage de la puce utilisée est d'amener la batterie à la décharge quasi complète avant recharge. L'entrée EN (de validation) n'est pas utilisée, il faut la fixer à l'alimentation. Le cahier de caractéristiques précise que la puce peut être alimentée par un port USB jusqu'à la limite inférieure de tension admise sur ce genre de port, 4,35 V. Sous des tensions plus hautes, le MAX1811 peut soutenir 6,5 V maximum, et en fort courant, le circuit limite la charge pour maintenir à un niveau sûr la température de la puce. Au cas où seule une source de tension plus élevée serait disponible, j'ai ajouté IC2, un régulateur de 5 V à faible chute. Le cavalier JP1 sélectionne la tension d'entrée pour le MAX1811 : AUX ou USB. Il ne faut pas brancher de tension à l'entrée du régulateur si la source sélectionnée est le port USB. Pour gagner de la place, le condensateur C2 à l'entrée du MAX1811 découple



aussi la sortie du régulateur.

Vous devrez graver dans le PICAXE le programme d'identification du modèle réduit. Avec un adaptateur FTDI TTL vers RS232 comme interface sur les lignes TXD et RXD du PICAXE, en combinaison avec l'éditeur de programmation du PICAXE, il vous faudra un inverseur supplémentaire (**fig. 3**).

Le radiogoniomètre

Le récepteur est constitué principalement d'un module complémentaire à celui de l'émetteur. La sélection du module récepteur demande un peu plus d'attention pour s'assurer qu'il fournit les bons signaux à notre circuit.

Dans la **figure 4**, la première sortie du récepteur est la broche 14, DATA, appliquée à un FET pour signaux faibles, T1 dont la sortie va au transducteur piézoélectrique Bz1, qui n'est pas un vibreur piézo. Ce que vous entendrez de ce transducteur, si tout va bien, ce sont les tonalités et les pauses de l'émetteur. À l'usage, vous reconnaîtrez votre indicatif personnel embarqué dans l'émetteur. La deuxième sortie du récepteur est la broche 13 RSSA. On y trouve une tension proportionnelle à la force du signal pour la commande automatique de gain (AGC) de ce genre de récepteurs. Nous l'appliquons à l'entrée du pilote de LED LM3914. Les tensions que j'ai mesurées sur le prototype varient entre 0,4 V et 2 V, elles deviennent les limites fixées pour le pilote de LED. Inutile de s'étendre sur la structure du circuit autour du LM3914, il a suscité un déferlement de projets à construire soi-même au cours des dernières décennies. Si vous désirez un tutoriel détaillé sur la question, pointez votre navigateur sur le lien [1], profitez du blog de Dave EEVBlog #204 et remettez-lui le bonjour de notre part.

Le récepteur est alimenté par des piles ou des accumulateurs AA ou AAA, mais trois piles alcalines pourraient suffire. Même si les spécifications du récepteur indiquent qu'il ne supporte pas plus de 5 V, mon module a été testé jusqu'à 7 V et il marche encore.

Il ne lui manque plus qu'une antenne directionnelle Yagi à quatre brins. Elle a été choisie pour sa simplicité de construction et ses propriétés de directivité, une qualité qui deviendra manifeste dans le mode d'emploi.

La construction

L'émetteur est assemblé sur un circuit imprimé à double face (**figure 5**) pour des CMS. On y implante des embases au pas de 2,54 mm pour

toutes les connexions extérieures sauf peut-être K3. Le tracé est disponible gratuitement par le lien [2]. Le PICAXE peut y être soudé directement, mais aussi inséré dans un support DIL-8. Ne soudez les deux transistors que si vous montez la version à batterie de secours. Ne branchez pas encore le module RF au connecteur K3.

À ce stade, vous pouvez tester le pilote de l'émetteur en connectant un transducteur piézoélectrique entre les broches data et masse qui vont au module RF. Quand vous mettez sous tension, vous devriez entendre votre indicatif.

On peut relier le module RF au circuit imprimé par un bout de câble plat à 4 conducteurs en utilisant embases et coupleurs IDC mâles et femelles. Autre solution : le coucher sur la carte du microcontrôleur comme à la **figure 6**, dans la version du laboratoire Elektor. Vérifiez bien que les soudures ne se touchent pas.

La dernière partie consiste à installer une antenne quart d'onde. Un morceau de fil rigide de 173 mm de long fera l'affaire, j'ai pris un des conducteurs d'un câble Ethernet. Avec un module émetteur à 315 MHz, la longueur de l'antenne passe à 240 mm, réminiscence : $(300/f) / 4$.

La carte du microcontrôleur, le module RF et la batterie de secours, s'il y en a une, on peut les envelopper dans un morceau de gaine thermorétractable.

Le tracé du circuit imprimé du **récepteur** est à la **figure 5**. Celui-ci est à une seule face et pour composants traditionnels, aucune difficulté donc, il n'y a qu'à y implanter et souder tous les composants. Le module radio se monte verticalement. La carte et le support pour trois (ou quatre) piles AA(A) peuvent prendre place sur la base de l'antenne, derrière le réflecteur. Il faut connecter les éléments du dipôle (radiateur) de l'antenne Yagi à l'entrée du module radio au moyen d'un morceau de câble coaxial de 50 Ω tel que du RG174/U ou /CP, le plus court possible pour éviter les pertes. Le câble audio blindé ne convient pas.

Le **chargeur**, en option, est construit sur la carte de la **figure 7**. N'oubliez pas les cavaliers.

L'antenne à 4 brins est du fait maison intégral. Internet foisonne de modèles variés et les forts en mécanique de votre club de modélisme sont probablement persuadés qu'ils peuvent fabriquer des antennes sophistiquées pour les membres.

Liste des composants

Émetteur

Résistances :

R1 = 10 k Ω 1%, CMS 0805

R2 = 22 k Ω 1%, CMS 0805

Semi-conducteurs :

IC1 = PICAXE-08M2, programmé

T1, T2 = DTB123YK

Divers :

K1, K2, K5 = embase à 3 broches au pas de 2,54 mm
K3 = embase à 4 broches au pas de 2,54 mm (en option, cf. texte)

K4, BT1, embase à 2 broches au pas de 2,54 mm module émetteur (sur K3), ASK, 433 MHz bande ISM, modèle approuvé, p.ex. Quasar Electronics type QAM-TX1 (433 MHz), Farnell/Newark # 1304024.

Aux USA, utiliser un équivalent à 315 MHz

accu LiPo, 3,7 V, 130 mAh

circuit imprimé 120139-1

Récepteur

Résistances :

R1 = 1 k Ω

R2 = 620 Ω 1%

R3 = 360 Ω 1%

R4 = 2,7 k Ω

Condensateur :

C1 = 100 nF, au pas de 5 mm ou 7,5 mm

Semi-conducteurs :

D1 à D10 = LED orange, 2,5x5 mm rectang. 20 mA

T1 = 2N7000

IC1 = LM3914

Divers :

BZ1 = (hors carte) transducteur piézo à fils Farnell/Newark # 1193640

K1, K2, (BZ1) = embase à 2 broches au pas de 2,54 mm

(MOD1) = embase à 15 broches, droite, SIL, au pas de 2,54 mm

MOD1 = récepteur AM Superhét, QAM-RX3 (433 MHz), RS Components # 742-4484. Aux USA utiliser un équivalent à 315 MHz

Coupleur de piles pour 3 ou 4 AA(A) cf. texte

circuit imprimé 120139-2

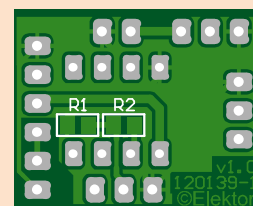
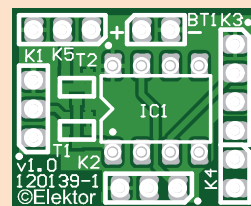
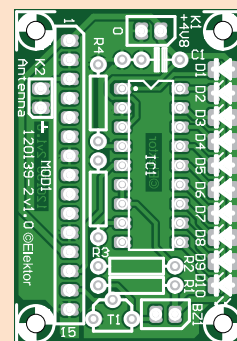


Figure 5.

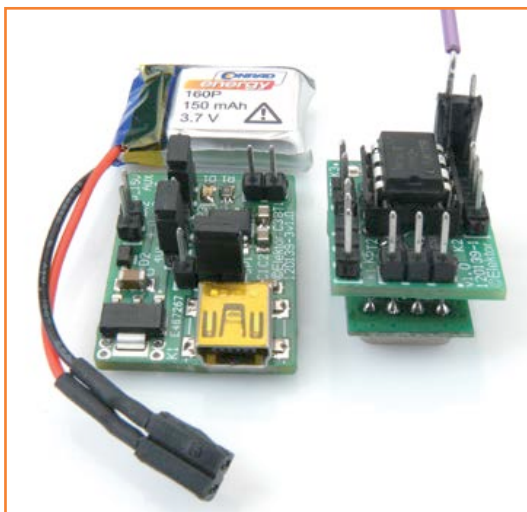
Le circuit imprimé de l'émetteur est à double face et garni de CMS ; celui du récepteur est à simple face pour composants traditionnels. La carte de l'émetteur est représentée ici à l'échelle 200 % de la vraie grandeur.



J'ai fabriqué les radiales à partir de fil de cintre, mais tout fil rigide conviendra, comme celui d'installation électrique. Tâchez de couper les radiales aux longueurs indiquées dans le dessin de la **figure 8**, ce sont celles pour 433 MHz. J'ai réalisé le prototype dans une plaque de plastique ondulé de 3 mm genre Corriboard et j'ai inséré les radiales dans les creux. L'espacement ne correspond pas exactement, mais en pratique, ça fonctionne.

De son côté, le laboratoire Elektor a construit son antenne Yagi sur une planche de bois de 570 x 53 x 12 mm avec des longueurs de fil

Figure 6.
Le circuit imprimé du chargeur LiPo à gauche, l'implantation de l'émetteur à droite. La carte PICAXE et le module émetteur peuvent se connecter sans câble, simplement en courbant et soudant les quatre fils à l'emplacement de K3.



électrique monobrin de 2,5 mm² csa (**figure 9**) positionnés exactement comme à la figure 8. La base est longue et assez large pour soutenir le circuit imprimé du récepteur et le coupleur de piles. Cette antenne présente un gain d'environ 7 dB pour permettre de localiser l'émetteur à une distance voisine de 300 m en zone habitée, comme le laboratoire a pu le mesurer dans et aux abords du château qui abrite la Rédaction. On connecte les éléments du dipôle d'antenne au récepteur par le câble installé précédemment. Les plus méticuleux du nanovolt voudront sans doute y ajouter un asymétriseur (*balun*) pour adapter le radiateur symétrique au câble coaxial, mais

ici, les pertes inévitables et le coût n'en valent pas la chandelle.

Les essais

Fort heureusement, aucun étalonnage n'est requis. On éteint provisoirement l'émetteur. En allumant le récepteur, on entend le bruit blanc avec des pics occasionnels. L'indicateur à LED ne montre que la première LED allumée.

On branche l'émetteur pour entendre l'indicatif, simultanément, la colonne lumineuse doit s'animer au rythme des tonalités de l'indicatif. Elle devrait atteindre le sommet quand émetteur et récepteur se rapprochent à environ 3 m l'un de l'autre.

À l'usage

Il est bon de s'entraîner un peu avant d'avoir à retrouver son appareil accidenté. Demandez à un ami d'aller cacher la balise dans un parc ou un jardin et essayez de localiser l'émetteur.

L'antenne est directionnelle et le bout étroit apporte la meilleure sensibilité. Lors de la recherche, tenez l'antenne droit devant vous et faites une rotation complète. Écoutez votre indicatif pour être sûr de chercher *votre* balise. Avancez dans la direction du maximum. Continuez ainsi en vous arrêtant de temps en temps pour vérifier votre orientation. La colonne lumineuse doit grimper à mesure que vous approchez.

Quand toutes les LED sont allumées, vous avez atteint le maximum. Retournez l'antenne pour pointer le réflecteur dans la direction que vous avez suivie. Maintenant, quand vous tournez l'antenne autour de vous, cherchez le minimum de signal.

Changez l'orientation verticale de l'antenne, parallèlement, perpendiculairement au sol ou entre les deux.

Les derniers mètres sont les plus délicats si la balise est perdue dans la végétation ou les broussailles. N'oubliez pas de scruter aussi la cime des arbres, on ne sait jamais où va se nicher un modèle réduit en perdition !

(120139-1)

Liste des composants

Chargeur LiPo (optionnel)

Résistances :

R1 = 1.5 kΩ CMS 0805

Condensateurs :

C1 = 100 nF CMS 0805 X7R
C2, C3 = 10 µF 10 V, CMS 0805, X7R
C4 = 10 µF 25 V, CMS 1206, Y5V

Semi-conducteurs :

D1 = LED, rouge, CMS 0805
D2 = PMEG2010AEH, Farnell/Newark # 1510673
IC1 = MAX1811ESA+, Farnell/Newark # 1593327
IC2 = NCP1117ST50T3G, Farnell/Newark # 2112617

Divers :

K1 = mini USB réceptacle type B, CMS
K2, (BT2) = embase à 2 broches au pas de 2,54 mm
JP1, JP2, JP3 = embase à 3 broches au pas de 2,54 mm, avec cavalier
circuit imprimé 120139-3

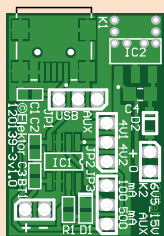


Figure 7. Le circuit imprimé du chargeur est à double face pour CMS.

Liens

[1] Dave's EEVBlog # 204 :

<http://goo.gl/IZdwTou>

http://www.youtube.com/watch?v=iIKGvHjD-QHs&feature=player_embedded

[2] www.elektor.fr/120139

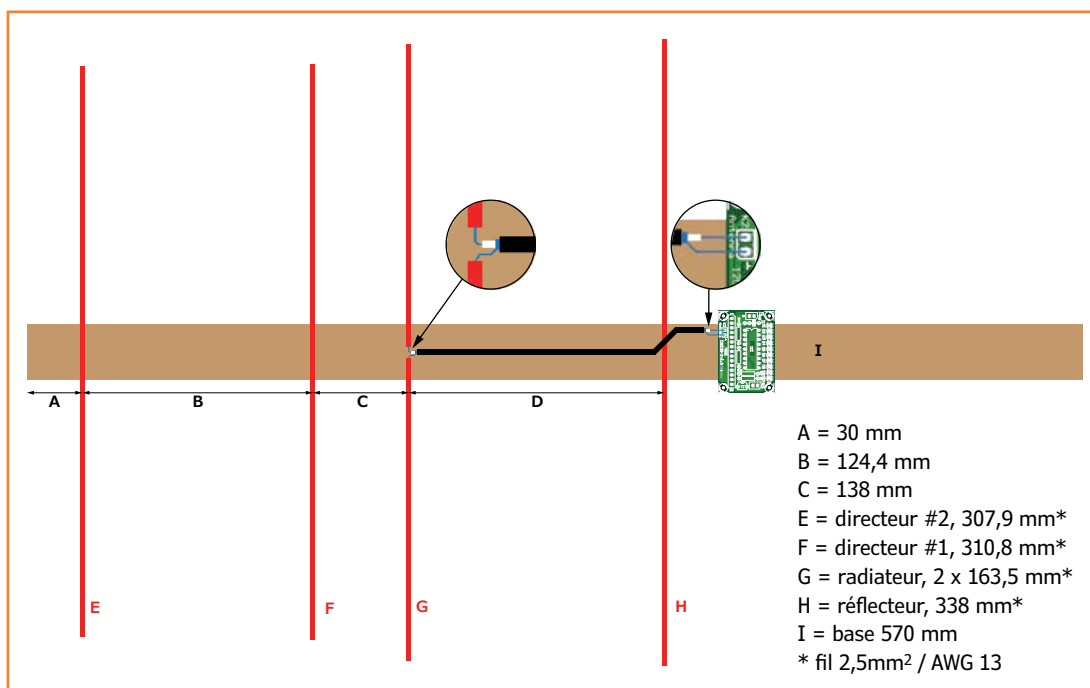


Figure 8.
Structure et dimensions de l'antenne Yagi construction maison. Son gain est d'environ 7 dB. Pour 315 MHz, l'espacement et les longueurs des éléments doivent être multipliés par 1,37.

Publicité

OBTÉNEZ LA DERNIÈRE VERSION !

Nouvelles fonctionnalités de la version 6.4:

- La simulation de schémas EAGLE dans LTspice IV
- L'affichage et la fonction de recherche des attributs pour le dialogue ADD- et REPLACE-
- L'importation de fichiers de design depuis P-CAD, Altium et Protel dans ACCEL-ASCII
- Un guide utilisateur et le pré-réglage (astuces sur les outils, raccourcis) optimisés

EAGLE
V6



www.cadsoft.de



Flowcode 5 pour concevoir

FLOWCODE5

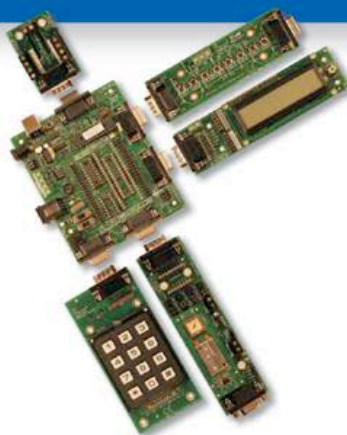


Flowcode 5 est l'un des langages de programmation graphique pour microcontrôleurs (PIC, AVR, ARM et dsPIC/PIC24) les plus avancés au monde. Son avantage principal est de permettre la création de systèmes électroniques et robotiques complexes même si l'on manque encore d'expérience.

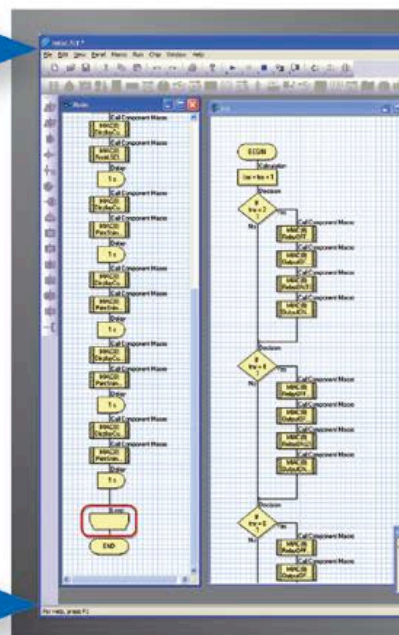
Flowcode est utilisé :

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

... en électronique



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



... pour la commande industrielle



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

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

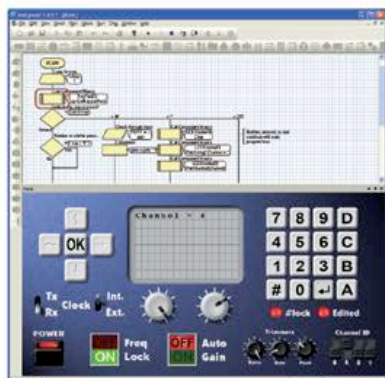
Outil de débogage FlowKit

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

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



voir et pour se former...



NOUVEAU dans Flowcode 5 :

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

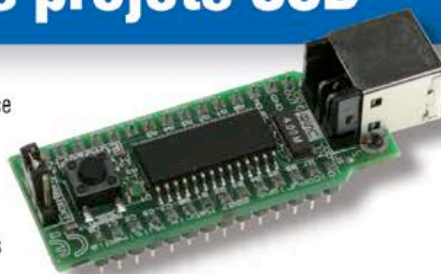
... en robotique

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



... pour les projets USB

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



Retrouvez les E-blocks et leur documentation sur :
www.elektor.fr/eblocks

recyclez votre alimentation ATX

avec simplicité et élégance

Ben Jordan
(États-Unis)

Voici une petite carte adaptatrice bien pratique, capable de convertir n'importe quelle alimentation ATX d'ordinateur en alimentation de paillasse. Une solution simple et élégante pour alimenter vos prototypes !

Caractéristiques

- sorties pour ± 12 V, 5 V, 5 V veille et 3,3 V
- aucune modification à apporter à l'alimentation ATX
- mises sous tension et hors tension de l'alimentation ATX faciles
- LED témoin de fonctionnement/défaut
- supporte des courants forts
- bornes de connexion pour chaque tension de sortie
- connecteurs femelles pour fils de type « crocodile »

Une chose est certaine en électronique, c'est que tout bricolage nécessite une bonne alimentation. Maintenant je ne sais pas ce qu'il en est de votre budget, mais le mien est plutôt... hum... disons tenu par une femme pour qui le mot alimentation est avant tout synonyme de courses à faire. Vous avez probablement déjà réalisé quelques alimentations à régulateurs linéaires, mais concevoir et construire une alimentation de laboratoire réglable et capable de délivrer des tensions élevées et des courants forts n'est pas trivial. Et puis si vous êtes comme moi, la plupart de vos montages comportent des circuits numériques et une section analogique sous basse tension, p. ex. des préamplificateurs audio.

En général je bidouille avec un microcontrôleur et quelques ampli-ops, donc le plus pratique pour moi est d'avoir un rail de 5 V pour le μ C (et son cortège de CI) et des rails de ± 12 V pour les AOP. Les échantillons avec lesquels je travaille (p. ex. le processeur DSP56367 de Freescale ou le μ C LPC2101 de NXP) ont de plus en plus souvent besoin de rails de 1,8 V et 3,3 V pour leur cœur interne et leurs E/S. Ce qui a priori veut dire construire une alimentation équipée de tous ces

rails à chaque projet. Plutôt casse-pieds. Heureusement, la plupart de ces tensions (sauf 1,8 V) sont disponibles sur les alimentations à découpage ATX des PC. Et des alims de PC, au fil des années j'en ai accumulé plusieurs. Elles possèdent toutes des sorties de 3,3 V, 5 V, 5 V veille, et ± 12 V. Je vous accorde que la régulation croisée n'est pas terrible sur les rails de 12 V, mais elle reste satisfaisante pour la majorité des AOP.

Réfléchir avant d'agir

Pourquoi, me direz-vous, ne pas se contenter de fixer des prises bananes directement sur l'alimentation, comme on le voit si souvent faire sur le Web ? Bonne question ! Voilà pourquoi :

- les alimentations ATX sont conçues avec des distances d'isolement et donc de sécurité très précises. Y ajouter des bornes de connexion, c'est prendre le risque d'entrer en contact avec une tension mortelle (côté CA d'une alimentation à découpage, les pics de tension peuvent dépasser 1000 V, et les bus sont sous +370 V CC).
- Votre alimentation fonctionne parfaitement, et si vous me ressemblez vous l'utiliserez pour alimenter des ampli-ops un jour, et une carte-mère d'ordinateur le lendemain. Pourquoi lui demander plus ?

Je voulais une conception plus élégante que ce genre de bricolage, avec prises de connexion et points de contacts pour pinces crocodile.

J'ai donc cherché à construire une carte qu'il suffirait de relier à une alimentation ATX pour en faire une alimentation de paillasse. J'ai commencé par écrire quelques spécifications. L'adaptateur devait :

- ne pas altérer l'alimentation ATX



- utiliser un connecteur ATX
- avoir une borne de connexion pour chaque tension d'alimentation de l'ATX
- associer à chaque tension de sortie une borne de retour reliée à la masse (GND)
- supporter de fortes intensités
- avoir un circuit de commutation qui utiliserait le contrôleur marche/arrêt de l'ATX

- avoir des LED témoin pour la mise en veille et le signal *Power Good*
- utiliser un montage traversant pour que vous aussi, cher lecteur, puissiez facilement l'assembler.

J'avais aussi envisagé l'ajout d'un galvanomètre ou d'un afficheur pour surveiller les tensions de

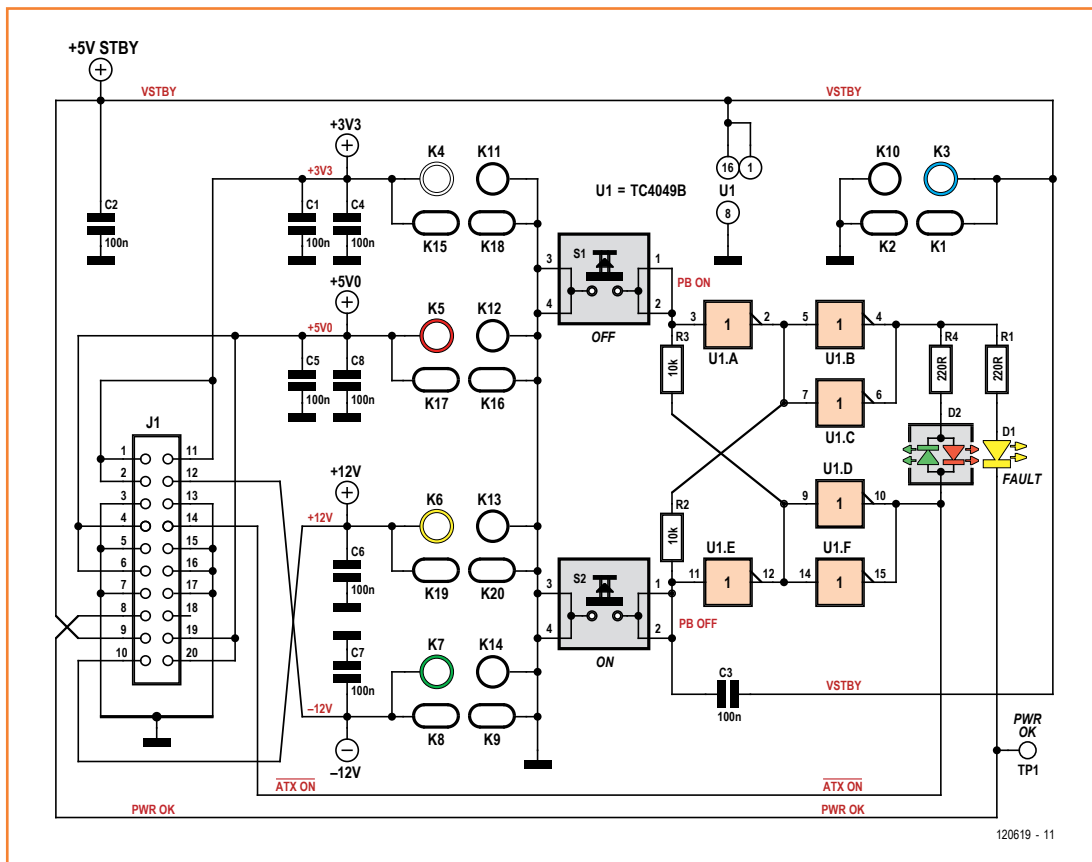


Figure 1.
Un schéma de principe qui montre aussi la simplicité de la carte.

sortie, mais ça aurait été céder à la manie des surcouches de fonctions plus ou moins utiles : après tout, les sorties sont parfaitement définies et plutôt bien régulées, et il est facile d'y insérer les pointes de touche d'un multimètre.

Un circuit simple

La **figure 1** montre la circuiterie de la carte. Le circuit de commutation est une simple bascule qui exploite deux portes d'un circuit CMOS sextuple NON 4049 alimenté par le rail ATX 5 V de la veille

Liste des composants

Résistances

R1,R4 = 220 Ω

R2,R3 = 10 k Ω

Condensateurs

C1 à C8 = 100 nF

Semi-conducteurs

D1 = LED, 3 mm, jaune

D2 = LED, 3 mm, bicolore à 2 broches

IC1 = TC4049B

Divers

K3 à K7,K10 à K14 = bornes de connexion pour fiches banane

J1 = connecteur ATX pour montage sur CI

S1,S2 = touches à effleurement

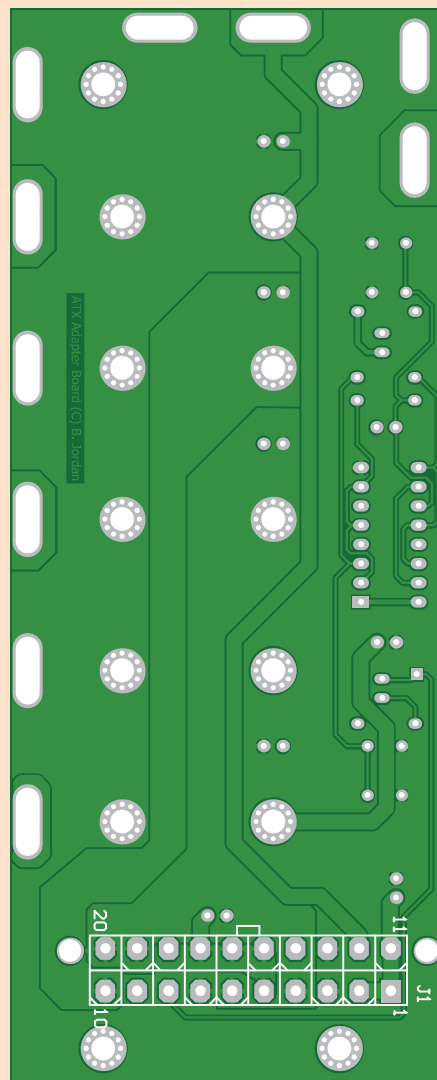
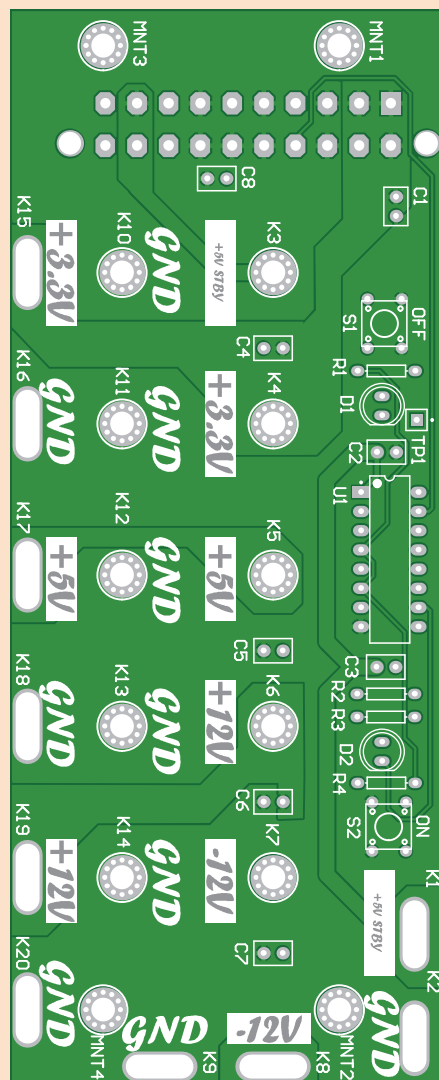


Figure 2.
Aucune erreur de montage possible avec cette sérigraphie. Notez que le connecteur ATX doit être monté côté cuivre de la carte.

(standby). Les autres portes du 4049 dirigent le signal de mise sous tension vers l'alimentation ATX et les LED. Appuyer sur S2 active l'alimentation, S1 la coupe. La LED bicolore D2 indique un mode veille (vert) ou sous tension (rouge). Les alimentations ATX ont une protection incorporée, mais aussi une sortie *Power Good* qui prévient la carte-mère lorsque l'alimentation est prête, ou en cas d'anomalie. La présence d'un défaut est signalée par la LED jaune D1 : elle s'allume lorsque *Power Good*, qui est un signal actif au niveau bas, passe au niveau haut. Tout clignotement continu est mauvais signe, et annonce en général un problème d'alimentation.

Les condensateurs aident au découplage des différentes sorties, à l'exception de C2, utilisé pour l'anti-rebonds du circuit marche/arrêt. Tous sont des condensateurs à céramique de 100 nF. R1 et R4 sont les résistances-talons des LED D1/D2, R2 et R3 forment les voies de réaction pour la bascule marche/arrêt.

J'ai vu que certains ajoutaient des résistances de charge minimale à leurs alimentations ATX converties. D'après mon expérience, cela ne sert à rien : le ventilateur est une charge minimale suffisante, et de toute façon la plupart des alimentations à découpage modernes sont conçues pour fonctionner en l'absence de charge. Cela dit, si une charge minimale vous rassure, vous pouvez ajouter une résistance de 10 ou 20 W (10 Ω suffisent en général) au rail de 5 V. Soudez-la entre les bornes GND et 5 V.

Conception de la carte

La carte adaptatrice possède une embase ATX, une circuiterie marche/arrêt, ainsi que des bornes de raccordement à espacement standard pour la connexion de fiches bananes simples ou doubles. Une couche de sérigraphie (**figure 2**) indique le placement des composants. Notez que l'embase pour l'alimentation ATX doit être montée face cuivre.

Nouveauté de cette carte, les connecteurs femelles métallisés disposés le long des bords qui permettent de se servir également de fils d'essai de type « crocodile » ; leurs tensions de sortie sont indiquées sur la sérigraphie. Le dessin de la carte au format PDF peut être téléchargé depuis [2].

Les trous métallisés K3 à K14 servent à fixer les terminaisons des prises banane / bornes de

connexion. Notez la couronne « trouée » qui les entoure, et qui encercle de même les trous de montage. Ces couronnes renforcent le serrage des vis et fournissent aux bornes de connexion un chemin conducteur de faible résistance à travers la carte.

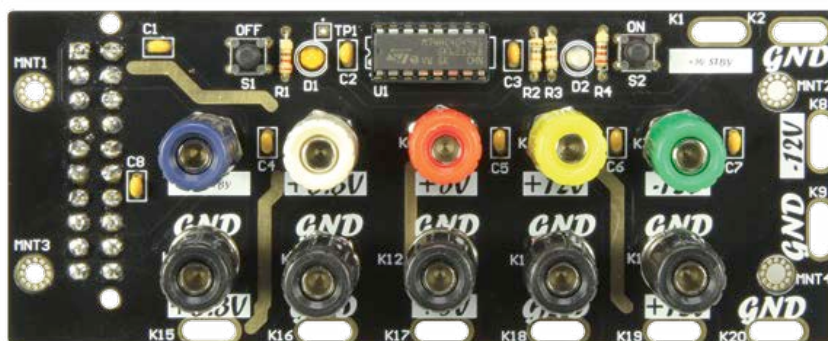
En pratique

La carte adaptatrice offre cinq bornes de connexion (5 V veille, 3,3 V, 5 V, +12 V et -12 V) de couleurs différentes, et dispose d'espacements standard de 0,75 pouce pour la connexion des fiches banane double. Je l'utilise au travail et à la maison pour alimenter plusieurs kits de développement. À mon travail, je l'ai reliée à une alimentation ATX pour alimenter en même temps trois cartes FPGA de chez Altium : une Nanoboard-II et deux Nanoboard 3000. Inutile de préciser qu'elle me simplifie la vie !

(120619 – version française : Hervé Moreau)

Liens

- [1] <http://jordandsp.com/ATX-bench-top-power-supply-adapter.php>
- [2] www.elektor.com/120619



J'en veux une !

Ce projet est disponible en nombre limité sous forme de kit complet : carte imprimée, ensemble des composants y compris les bornes de connexion, connecteur ATX, instructions de montage détaillées, schémas de principe et de montage, et gabarits de perçage pour un montage encastré. Visitez [1] pour plus d'informations sur le prix et les modalités de commande.

de BASIC à Python (2)

récit d'une découverte

Jean-Claude Feltes
(Luxembourg)



Dans la première partie, nous avons souligné les différences entre Python et BASIC, montré comment installer l'interpréteur, et exécuté quelques programmes. Nous poursuivons ici notre découverte de Python en ouvrant les portes de ses bibliothèques graphiques.

S'il est vrai que Python est un langage de programmation bien adapté à l'électronique, alors les électroniciens que nous sommes ne manqueront pas de vouloir visualiser des données. Une courbe sera en effet toujours plus facile à interpréter que des colonnes de chiffres. Inutile ici de réinventer la roue et de chercher à construire chaque portion de courbe « à la main », comme tout bon langage qui se respecte Python dispose pour cela de bibliothèques qui se chargent des détails ingrats.

Graphiques en 2D

La bibliothèque *matplotlib* est le module Python standard pour les tracés en 2D. Si la devise du langage Python lui-même est « Une méthode unique pour chaque tâche », il en va hélas autrement pour les modules supplémentaires. Un défaut qui m'a d'ailleurs coûté de nombreuses heures de travail. Le module *matplotlib* possède ainsi une interface procédurale simple, et une interface orientée objet compliquée. Les exemples que l'on trouve sur Internet ou ailleurs utilisent l'une ou l'autre de ces méthodes, et devant de tels codes on a vite fait de ne plus savoir à quoi s'en tenir.

L'interface *pyplot* simplifie elle aussi l'écriture des programmes. Celui du **listage 1** crée une oscillation sinusoïdale amortie, puis affiche la courbe résultante dans une fenêtre (**figure 1**).

La première ligne importe le module *pyplot* en tant qu'objet *plt* (ce qui permet de préfixer les fonctions de *pyplot* avec *plt* plutôt qu'avec un encombrant *matplotlib.pyplot*). La seconde ligne

importe 3 fonctions du module de calcul numérique *numpy*.

La fonction *linspace()* découpe un intervalle donné, puis retourne le résultat sous forme de tableau. Comme les fonctions de *numpy* peuvent aussi traiter les tableaux, il nous suffit de découper l'intervalle [0;7] en 1000 parties avec *linspace* pour ensuite calculer les valeurs de la fonction :

```
y= sin(5*x)*exp(-x)
```

Avec cette ligne, les 1000 valeurs calculées sont rangées dans le tableau *y*.

Pouvoir comme ci-dessus se passer d'une boucle *for* est une aubaine, et rend bien sûr le code plus clair et plus rapide à écrire.

L'instruction *plt.plot(x,y)* trace la courbe représentée par les points (x,y), mais il faut encore l'instruction *plt.show()* pour afficher le tracé obtenu. Pour représenter plusieurs courbes sur la même figure, il suffit d'appeler chaque fois la fonction *plot()* (**listage 2**).

La barre d'outils de la fenêtre d'affichage (**fig. 1**) permet de zoomer et de sauvegarder un tracé sous différents formats. Cette même barre affiche par ailleurs les coordonnées du point situé sous votre curseur lorsque vous survolez le graphique. Ajouter des objets graphiques (des *widgets*) à un tracé serait plus compliqué et nécessiterait une approche orientée objet.

Exemple : réponse en fréquence

Prenons l'exemple de la courbe de réponse d'un filtre passe-bas RC. Sa fonction de transfert *F* peut s'exprimer à l'aide de la formule du pont

diviseur avec impédance complexe selon :

$$F = 1 / (1 + j \omega R C)$$

La courbe de réponse correspond alors à la variation du module de F en fonction de la fréquence (**fig. 2**). Pas de problème pour le tracé, Python sait manipuler les nombres complexes.

Le programme (**listage 3**) commence par remplir le tableau f avec les valeurs obtenues par division logarithmique d'un intervalle de fréquences. Pour un partage linéaire, nous aurions utilisé la fonction `linspace()`. Ces valeurs servent ensuite à calculer les éléments des tableaux F (valeurs complexes de la fonction F) et $Fabs$ (valeurs absolues de F). Là encore, nul besoin de boucle `for` pour les calculs, les fonctions de `numpy` avalent les tableaux sans broncher.

La courbe est ensuite tracée, avec une échelle logarithmique pour l'axe des fréquences. Pour améliorer la représentation graphique, nous ajoutons de surcroît une grille :

```
ax.grid(True, which = "both", linestyle = "-")
```

Le paramètre `True` affiche la grille, `both` affiche la grille principale et la sous-grille (c.-à-d. les divisions de la grille principale), `linestyle` précise le style du trait.

Nous pourrions également représenter la réponse en phase avec les lignes de code suivantes :

```
# plot phi = f(f)
fig = plt.figure()
phi = angle(F)*180.0/pi
ax2 = fig.add_subplot(212)
ax2.plot(f, phi)
ax2.grid(True, which = "both", linestyle = "-")
```

```
ax2.set_xscale ("log")
ax2.set_xlabel("f/Hz")
ax2.set_ylabel("phi/degrees")
```

(Pensez à importer la fonction `angle` de `numpy`.)

Exemple : synthèse de Fourier

L'électronicien amateur de calcul numérique ajoutera au crédit de Python ses aptitudes mathématiques, puisque même des traitements complexes comme la synthèse de Fourier ne nécessitent que peu de code.

Illustrons cela (**listage 4**) avec la synthèse d'une onde carrée construite par pondération choisie de 30 harmoniques. L'approximation est

Listage 1 : Sinus.py

```
import matplotlib.pyplot as plt
from numpy import sin, exp, linspace

x=linspace(0.0, 7.0, 1000)
y= sin(5*x)*exp(-x)

plt.plot(x, y)
plt.show()
```

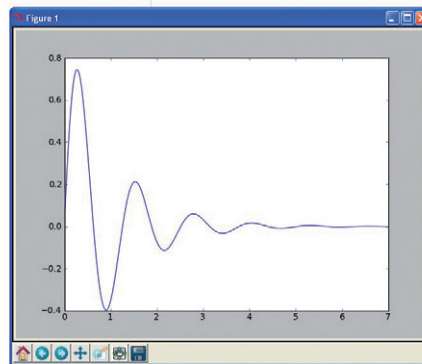


Figure 1.
Représentation graphique d'une onde sinusoïdale amortie (d'après listage 1).

Listage 2 : Multigraph.py

```
import matplotlib.pyplot as plt
from numpy import sin, exp, linspace

x=linspace(0.0, 7.0, 1000)
y1 = sin(5*x)*exp(-x)
y2 = y1* 0.5

plt.plot(x, y1)
plt.plot (x, y2)
plt.show()
```

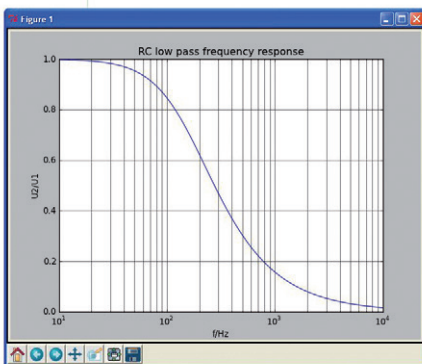
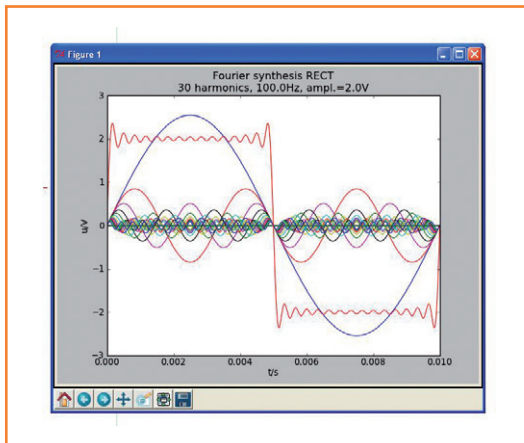


Figure 2.
Réponse en fréquence d'un filtre passe-bas RC (d'après listage 3).

Figure 3. Synthèse de Fourier d'une onde carrée (d'après listage 4).



excellente (**fig. 3**), et surtout obtenue au prix d'un programme relativement court. Le code permet d'augmenter ou de diminuer le nombre d'harmoniques.

To GUI or not to GUI ?

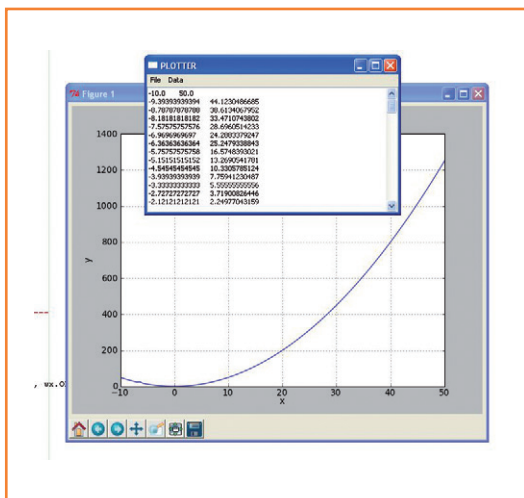
Les linuxiens purs et durs ont la réputation de considérer les interfaces graphiques comme des clicodromes et la queue de la souris comme celle du diable. Cliché ou réalité, les programmes bien conçus et simples à utiliser n'en restent pas moins toujours appréciés. Et honnêtement, comment se passer d'un menu pour la manipulation des fichiers ?

Quoi qu'il en soit, avec Python il est possible de combiner dans un programme des objets graphiques et non-graphiques. Celui du **listage 5** est un traceur de données. Les données lues sont d'abord affichées dans un terminal sous forme de nombres, puis représentées sous forme de courbe dans une fenêtre graphique (**fig. 4**).

Quelle bibliothèque graphique choisir ? Python nous laisse l'embarras du choix. Je me suis d'abord tourné vers *Tkinter*, un module fourni avec l'interpréteur et d'apprentissage facile. Les problèmes que je rencontrais avec la copie de graphiques dans le presse-papier m'ont toutefois finalement fait pencher pour *wxPython*. Comme autres bibliothèques graphiques pour Python, citons *PyQt* et *GTK*.

Notre traceur de données est un bon exemple d'utilisation des objets graphiques d'une biblio-

Figure 4. Traceur de données : affichage des données brutes dans un terminal et leur représentation graphique.



Listage 3 : RC.py

```
import matplotlib.pyplot as plt
from numpy import pi, linspace, log10, logspace
from numpy import complex, abs # these allow
vector operations

# EDIT
R = 10.0E3
C = 100.0E-9

# END EDIT

RC = R*C

# create f values equally spaced on a log scale
f = logspace ( 1, 4, 100) # 100 values from 10**1
to 10**4
```

```
# calculate F (complex) and absolute value Fabs
F = 1 / (1 + 1j* 2 * pi * f * RC)
Fabs = abs(F)
```

```
# plot Fabs = f(f)
fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot(f, Fabs)
ax.grid(True, which = "both", linestyle = "-")
ax.set_xscale ("log")
ax.set_xlabel("f/Hz")
ax.set_ylabel("U2/U1")
ax.set_title("RC low pass frequency response")
plt.show()
```

thèque GUI (*graphical user interface*). Le programme permet l’affichage et la modification des valeurs de mesure stockées dans un fichier texte. Ces valeurs peuvent être représentées sur un graphique XY à deux dimensions, et également dans deux colonnes de nombres correspondant aux valeurs X et Y.

Pour rendre les choses plus simples, nous avons évité la programmation orientée objet « à la

Python ». Le code commence par l’importation des modules nécessaires et la définition des fonctions. Le reste est le programme principal.

La première chose que fait ce programme principal est de créer une instance *app* de l’objet *App*. Nous créons ensuite un objet *Frame* qui sera la fenêtre de l’application. Elle possède un menu et une zone de texte pour l’édition et l’affichage des données. La zone de texte est en elle-même un

Listage 4 : Fourier.py

```

"""FOURIER SYNTHESIS FOR RECT VOLTAGE"""
#-----
# EDIT HERE

n = 30                # number of harmonics
nb_points=1000        # horizontal resolution
frequency = 100.0     #Hz
amplitude = 2.0       #V

# END OF EDIT AREA
#-----
print "Importing modules"
import matplotlib.pyplot as plt
from numpy import sin, exp, linspace, pi
from numpy import zeros
#-----
def calc_amplitude(amplitude, i):
    """ Calculate amplitudes of harmonics"""
    # take only odd harmonics
    if i % 2 == 0:
        ai = 0
    else:
        ai = (4/pi)*amplitude / i
    return ai
#-----
def calc_harmonics(nb_points, n):
    """ Calculate harmonics and resulting voltage
    returns
    uharm = array nb_points * n
    ug = array nb_points
    """
    # init arrays for resulting voltage and
    harmonics
    ug = zeros(nb_points)
    uharm = zeros((nb_points, n+1))

    # harmonics and total voltage
    for i in range(1,n+1):
        ai = calc_amplitude(amplitude, i)
        fi = frequency * i
        uharm[:,i] = ai * sin(2 * pi * fi * t )
        ug = ug + uharm[:,i]
    return uharm, ug
#-----
""" Main program"""

T=1/frequency

# equally spaced time array for 1 period
t = linspace(0.0, T, nb_points)

# plot harmonics
uharm, ug = calc_harmonics(nb_points, n)
for i in range(1,n+1):
    plt.plot (t, uharm[:,i])

# plot resulting voltage
plt.plot (t, ug)

s=str(n)+" harmonics, "+str(frequency)+"Hz,
ampl.="+str(amplitude)+"V"
plt.title("Fourier synthesis RECT\n"+s)
plt.xlabel("t/s")
plt.ylabel("u/V")

# make plot visible
plt.show()

```


Listage 5 : Dataplot.py

```
#!/usr/bin/env python
""" Plot data from file """
import wx
import os.path
import matplotlib.pyplot as plt

def create_menu(frame):
    # create menu
    menubar = wx.MenuBar()
    # main menus
    mnuFile = wx.Menu()
    mnuData = wx.Menu()
    menubar.Append(mnuFile, "&File")
    menubar.Append(mnuData, "&Data")
    # submenus
    m_Open = mnuFile.Append(-1, "&Open")
    mnuFile.AppendSeparator()
    m_Exit = mnuFile.Append(-1, "E&xit")
    m_Plot = mnuData.Append(-1, "&Plot")
    # attach menu to frame
    frame.SetMenuBar(menubar)

    # bind menu events to procedures
    frame.Bind(wx.EVT_MENU, OnExit, m_Exit)
    frame.Bind(wx.EVT_MENU, OnOpen, m_Open)
    frame.Bind(wx.EVT_MENU, OnPlot, m_Plot)

#-----
# Event handlers
def OnExit(event):
    frame.Close()

def OnOpen(event):
    # ask for filename
    dlg = wx.FileDialog(None, "Open data file", os.getcwd(), "", "*.*", wx.OPEN)
    dlg.ShowModal()
    filename = dlg.GetPath()

    # open file, get data and put it into textbox
    try:
        f = open(filename, "r")
        data = f.read()
        f.close()
        textbox.SetValue(data)
    except:
        wx.MessageBox("Could not open file!")
```

```
def OnPlot(event):
    # plot data
    x,y = fill_xy_with_values(textbox)
    plot_xy(x, y)
#-----
def fill_xy_with_values(textbox):
    # """ get values from textbox
    # returns arrays x, y and number of data points"""
    text=textbox.GetValue()
    lines=text.splitlines()

    x=[]
    y=[]
    for line in lines:
        columns = line.split()      #separator can be one or more " " or "\t"
        x.append (float(columns[0]))
        y.append (float(columns[1]))

    return x,y
#-----
def plot_xy( x, y):
    """ Plot arrays x, y with matplotlib"""
    plt.figure(1)
    plt.subplot(111)                # 1 row, 1 col, plot nb. 1
    plt.grid(True)
    plt.plot(x, y)
    plt.xlabel("x")
    plt.ylabel("y")
    plt.show()
#-----
# Main
#-----
app = wx.App()

# create frame
frame = wx.Frame(None, title='PLOTTER', pos=(350,300))

create_menu(frame)
# editor textbox for data
textbox=wx.TextCtrl(frame, style = wx.TE_MULTILINE)

# show frame and run event loop
frame.Show()
app.MainLoop()
```

petit éditeur : des valeurs peuvent y être ajoutées ou modifiées, les raccourcis habituels peuvent être utilisés pour interagir avec le presse-papier, et un clic droit fait apparaître un menu doté de fonctions d'édition.

L'instruction `frame.Show()` affiche la fenêtre, et `app.MainLoop()` démarre la boucle de réception des événements : un clic de souris activera p. ex. le gestionnaire d'événements, qui appellera alors la fonction associée à cet événement.

La première fonction, `create_menu`, définit les menus et leurs associations avec les gestionnaires d'événements (fonctions préfixées par `On`). Les fonctions `OnExit`, `OnOpen` et `OnPlot` sont appelées par menu ou activation d'une touche de clavier. `OnOpen` appelle le sélecteur de fichier `wx.FileDialog` ; le paramètre `os.getcwd` (**get current working directory**) demande à ce que le répertoire en cours soit le premier dossier de recherche.

Le fichier sélectionné est ouvert de façon provisoire pendant qu'une structure de contrôle `try/except` intercepte toute erreur d'ouverture. Si le fichier n'est pas fermé pour erreur, son contenu est lu dans la variable `data` puis affiché dans la zone de texte.

Copie, insertion ou modification sont possibles dans cette zone. Plutôt que de charger les données depuis un fichier, il est bien sûr possible de les saisir ici.

La fonction `OnPlot` est chargée de la représentation graphique des données. Elle appelle pour cela les fonctions `fill_xy_with_values` et `plot_xy`. La première lit chaque ligne de la zone de texte et l'insère dans un tableau appelé `lines`. Chaque ligne de ce tableau est ensuite traitée à l'intérieur d'une boucle `for` où `line.split()` sépare les valeurs X et Y. Ces deux valeurs sont respectivement ajoutées aux tableaux `x[]` et `y[]`, qui eux sont retournés sous forme de valeurs de fonction. La fonction `plot_xy` trace la figure correspondante (**fig. 4**). Aussi rudimentaire soit-il, ce petit programme peut faire beaucoup de choses, et son équivalent en Visual Basic aurait assurément demandé plus de temps de codage. Essayez, vous verrez par exemple qu'ajouter un menu pour la sauvegarde des données n'est pas difficile.

Bilan & perspectives

Après cet article, vous ne devriez plus douter de la simplicité et de la puissance de Python. Si le langage vous a séduit, vous n'aurez aucun mal à écrire des programmes plus complexes que

les exemples précédents. Comme nous l'avons montré, Python se prête bien à la capture des données et à leur traitement. Transformée de Fourier rapide ou autres filtres ne sont pas non plus insurmontables avec Python. Avec les bibliothèques graphiques, il est de surcroît simple d'enrichir un projet avec une belle interface. Dans le troisième volet, nous utiliserons Python pour programmer un simple système de commande et d'acquisition de données via le bus ElektorBus et RS-485.

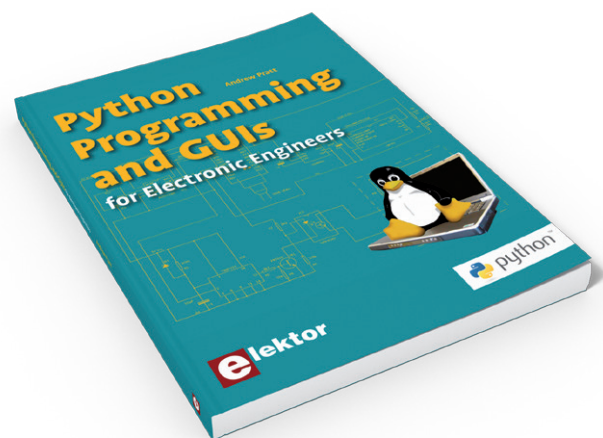
(120143 – version française : Hervé Moreau)

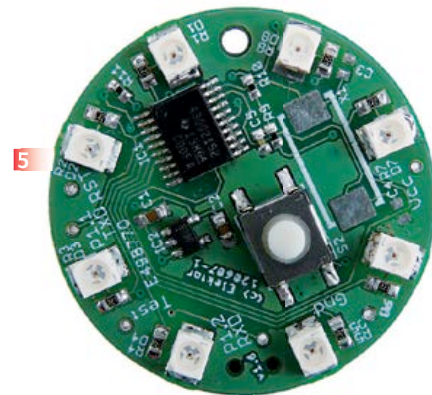
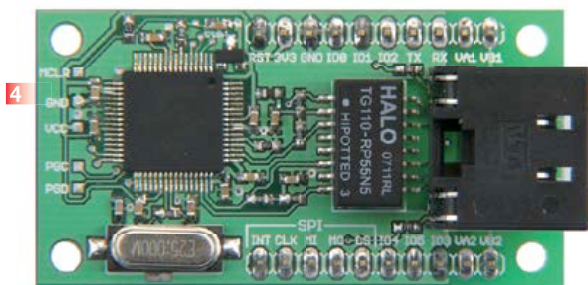
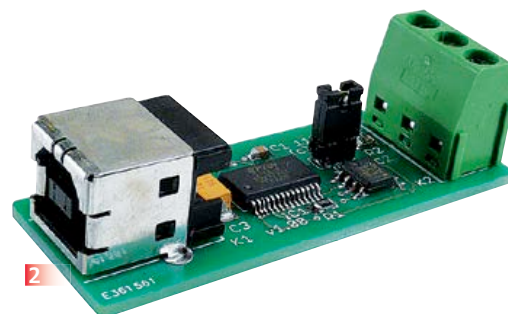
Liens & références

- [1] Listings etc. : www.elektor.fr/120143
- [2] Page de l'auteur (en allemand): <http://staff.itam.lu/feljc/home.html>
- [3] Documentation Python : <https://pypi.python.org/pypi/RPi.GPIO>
- [4] Tutoriels Python : www.awaretek.com/tutorials.html
- [5] Référence : Michael Weigend : Python gepackt
- [6] Introduction : J.M. Hughes : Real World Instrumentation with Python
- [7] Index des modules Python : <http://pypi.python.org/pypi>
- [8] Python pour l'électronicien : Andrew Pratt: Python Programming and GUIs for Electronic Engineers www.elektor.fr/python-programming

L'auteur

Jean-Claude Feltes enseigne l'électronique au Lycée Technique des Arts et Métiers de Luxembourg, une école publique qui forme des artisans, des artistes, ainsi que des techniciens supérieurs. Il se passionne pour l'électronique et l'informatique durant son temps libre (cf. [2]).





1 Andropod

Commandez vos montages

avec un smartphone ou une tablette Android

Écran tactile à haute définition, méga puissance de calcul, connexion réseau et fonctions de téléphonie sans fil : les téléphones tactiles et les tablettes Android seraient des centrales de commande presque idéales pour nos projets d'électronique s'il était plus facile de s'y connecter. Voici AndroPOD, votre interface série TTL et RS485. Décollage vertical garanti ! L'offre de matériel Android (par différents fabricants) et de logiciel est énorme et le code source du système d'exploitation est libre. Le puissant kit de fonctions logicielles ou framework d'Android offre accès à presque toutes les fonctions matérielles et permet de programmer des applications élégantes et conviviales.

Réf 110405-91

2 Le bus arrive

Le convertisseur USB/RS485 établit la connexion entre un port USB de votre ordinateur (portable) et un bus RS485 à deux fils. Un bornier à vis assure une connexion facile et fiable du bus RS485. Grâce au populaire circuit intégré FT232L et ses pilotes de FTDI, le convertisseur fonctionne non seulement avec toutes les versions actuelles de Windows, mais aussi

avec Windows CE, Windows Mobile, Linux et Mac OS X. Le module a été décrit dans une série d'articles d'Elektor qui traitent entre autres du protocole d'application, de mesure, de commande et de régulation, des platines expérimentales, etc.

Réf 110258-91

3 Module réseau universel

Connectivité interne les nuls

Le raccordement de sa propre électronique par une liaison internet est une fonction impressionnante. Ce module réseau universel - composé d'un circuit compact, d'une bibliothèque de logiciels libres et d'un microcontrôleur utilisable immédiatement comme serveur web - s'adresse même aux débutants. Quant aux électroniciens plus chevronnés, ils apprécieront à leur juste valeur les caractéristiques comme la communication SPI, PoE, et plus encore !

Réf 100552-91

4 Isolateur USB

Adieu parasites et boucles de terre. Débarrassez-vous des bruits causés par une boucle de masse et protégez votre PC des tensions extérieures. Entre appareils connectés, cet isolateur assure une séparation galva-

nique des lignes de bus et des lignes d'alimentation. Le circuit assure une séparation galvanique optimale autant des lignes de bus que des lignes d'alimentation entre le PC et l'appareil connecté par USB.

Le boîtier est fourni sans la découpe pour les connecteurs qui reste à faire dans les parois latérales.

Réf 120291-91

5 Boucle d'oreille à LED

La fantaisie et l'imagination sont des ressources précieuses aussi en électronique. Voici une petite création qui vous permettra de faire la fête... en boucle, et d'épater vos ami(e)s, et plus si affinités.

Caractéristiques :

- boucle d'oreille pilotée par microcontrôleur, avec motifs lumineux programmables
- carte ronde et légère, avec 8 LED symétriques
- très basse consommation (12 mW avec les LED multiplexées)
- circuit utile pour d'autres conceptions

Réf 120607-91

Informations et gamme complète sur :

www.elektorpcbservice.com

Luxmètre céleste

Mesure de lumière en occultation (totale) de soleil

Reinier Ott (Pays-Bas)

Assister à un phénomène stellaire comme une éclipse totale, c'est un événement rare, une expérience fantastique dont on garde la mémoire toute sa vie. Pour le photographier et obtenir des clichés à la mesure de ses espérances, il faut de la chance, une bonne préparation et un instrument comme celui que nous décrivons ici.

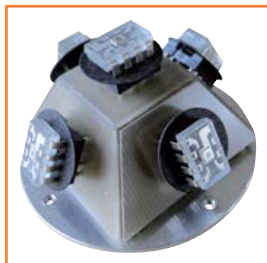


Figure 1.
L'assemblage des cinq capteurs de mesure sous des angles fixes.

L'occultation totale de soleil est très rare et le temps pendant lequel la lune éclipse le soleil n'est alors que de quelques minutes, tout au plus. Réussir des prises de vue dans ces circonstances exceptionnelles tient de l'exploit : d'abord, vous êtes en pleine lumière, sous une intensité d'éclairement de plus de 50 000 lux, puis le paysage s'estompe dans l'ombre de la lune et en peu de temps, la lumière du ciel tombe à moins d'un lux. Les différences de luminosité sont telles qu'un posemètre ordinaire ne peut plus rien pour vous. Si en plus vous avez fait le tour de la planète pour assister à cette éclipse, il ne s'agirait ni de la rater ni de s'affairer ailleurs au moment suprême.

Non seulement notre appareil de mesure doit posséder une dynamique suffisante, mais tout devra se passer automatiquement. Avec ce cahier de charges, on s'achemine probablement vers une centrale de mesure. N'oublions pas que les théâtres d'observation du phénomène ne se trouvent pas forcément dans un endroit du monde relié au réseau électrique : l'appareil sera alimenté sur une source d'une autonomie d'au moins 4 h.

Centrale de mesure compacte

L'instrument développé a pour mission de mesurer indépendamment, dans cinq directions différentes, la lumière du ciel pendant l'éclipse. En supplément, il comporte un thermomètre et un hygromètre. Pour concevoir et mettre au point cet instrument autonome que j'aime appeler mon « capteur d'éclipse », les E-blocks et le logiciel Flowcode de *Matrix Multimedia* m'ont été d'une grande aide.

À côté d'un écran LCD à deux lignes de 16 caractères et de trois témoins à LED, l'appareil dispose de trois poussoirs et d'un lecteur pour carte SD. L'interrupteur de mise en marche se trouve à l'arrière. Il servira aussi à la mise à zéro pour la création d'un nouveau fichier de données. Le **tableau 1** répertorie l'essentiel des caractéristiques du luxmètre céleste.

La batterie de capteurs de lumière se compose de cinq éléments identiques dont le convertisseur programmable lumière/fréquence TSL230 constitue la base. Chaque puce photosensible pointe vers le coin du ciel dont elle recevra la lumière : le zénith, le nord, l'ouest, le sud et l'est. L'appareil est construit de telle façon (**figure 1**) qu'il



Tableau 1.
Synoptique des spécifications de l'instrument de mesure

Principe de mesure	mesure de fréquence
Intervalle d'échantillonnage	fixe : 0,5 s
Intervalle d'enregistrement	1 s
Nombre de canaux	8
Intervalle de mesure par capteur	8 s
Types de capteurs	5x lumière (TSL 230) 1x température (LM35) 1x humidité relative (H1) 1x externe
Temps max. d'enregistrement	9,1 h ¹ (32 768 mesures)
Gamme d'éclairements	0,1 à 50.000 lux ²
Température	-25 °C à +50 °C
Humidité relative de l'air	20% à 100%
Entrée de mesure externe	5 Hz à 10 kHz (5 V)
Alimentation externe	Batterie ou adaptateur 9 V
Alimentation interne	6 V (4x AA alcalines)
Consommation de courant	80 mA (6 V)
Enregistrement des données	Carte SD/MMC
Processeur	PIC 18F4455
Poids total avec piles	976 g (capteurs 302 g)
Fixation au pied photo	filet 5/8"

¹ Possibilité de plus longue période

² Selon étalonnage (construction et filtres)

suffit de l'installer à l'aide du niveau d'eau et de l'orienter dans la bonne direction.

Pour éviter les erreurs de mesure dues à l'éclairage artificiel et mieux faire correspondre la sensibilité spectrale du TSL230 avec la lumière solaire, chaque capteur est coiffé d'un filtre bleu du type Wratten # 80A. Un écran en plastique noir placé sous chaque capteur empêche toute influence de reflets indésirables en provenance des autres capteurs.

La **figure 2** montre les cônes de sensibilité des capteurs. Les filtres en verre (**figure 3**) ont une ouverture effective de 25,4 mm (1 pouce) et une épaisseur de 2 mm.

La limitation de l'angle solide que les capteurs peuvent intercepter assure un minimum de chevauchement des cinq zones du ciel.

L'instrument se compose d'une console de commande et d'un ensemble de capteurs amovible. Un PIC 18F4455 trône au milieu de l'unité de commande. Ce processeur s'occupe de la communication avec les capteurs, de l'affichage, de l'enregistrement sur carte de mémoire SD, du clavier et des témoins à LED.

Par manque de place, le schéma du circuit complet n'est pas reproduit ici, mais est disponible gratuitement au téléchargement sur le site d'Elektor [1]. On y voit que la presque totalité des 40 broches du PIC ont été utilisées. Huit d'entre elles sont pour les canaux d'entrée indépendants, dont quatre servent aussi de sortie pour l'adressage des capteurs de lumière. L'affichage, le lecteur de carte et les trois boutons-poussoirs avec témoin à LED occupent 16 autres broches.

La sonde de température, un LM35, se trouve sur une platine séparée, dans l'ensemble des capteurs (**figure 4**). C'est un LM231 qui conver-

Figure 2.
La mesure de l'éclairement lumineux se fait dans 5 directions avec peu de chevauchement.

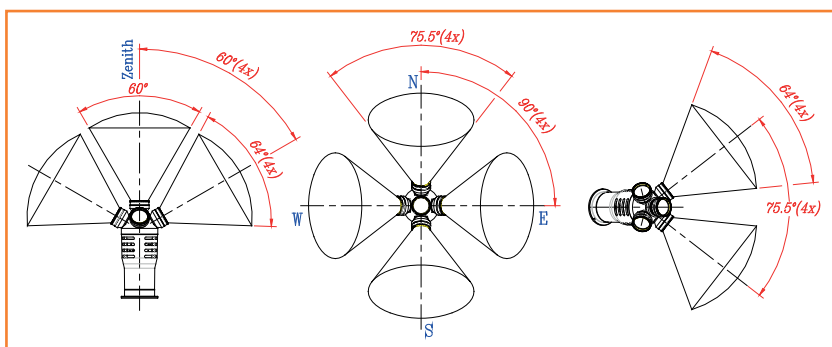


Tableau 2. Caractéristiques des capteurs

MultiPlex	1	2	3	4	5	6	7	0
Capteur	Lux_Z	Lux_N	Lux_E	Lux_S	Lux_W	Temp	Humid	Extrn
Signification	lum. zénith	lum. nord	lum. est	lum. sud	lum. ouest	temperature	humidité relative	Entrée aux.
Gamme automatique	oui	oui	oui	oui	oui	non	non	non
Msk (hex)	0x01	0x02	0x04	0x08	0x10	0x20	0x40	0x80

tit en fréquence la tension de la sonde pour la présenter au PIC en bonne et due forme. Quant au potentiomètre d'ajustage R11, il sert à régler la gamme de fréquences de travail. On améliore la résolution de la mesure de température en choisissant convenablement cette gamme, par exemple en faisant correspondre 5 kHz à 20 °C. Deux diodes en série, D1 et D2, permettent de mesurer également des températures sous le point de congélation. Il y a encore sur la même platine un capteur d'humidité relative, le Philips H1. Il agit comme condensateur variable, C6, et est associé à R4 pour former la constante de temps de l'oscillateur intégré au compteur binaire IC4. On calcule grosso modo la période ainsi obtenue par la formule : $T = 2,2 \cdot R4 \cdot C6$. La fréquence correspondante, IC4 se charge ensuite de la diviser pour la ramener entre 6 et 8 kHz pour un traitement optimal par le PIC.

J'ai également rédigé le logiciel pour PC, c'est le fichier « EclipseSens.exe » qui convertit les valeurs des capteurs, après correction d'étalonnage, vers les valeurs de degrés Celsius et de pourcentages d'humidité relative.

Le principe de mesure

Il y a huit canaux de mesure indépendants, tous effectuent des mesures de fréquence. Le pro-

gramme consacre exactement 0,5 s à échantillonner les signaux de mesure, puis 0,5 s à les traiter, les afficher sur écran et les enregistrer sur carte SD.

Chaque seconde, le programme évalue un capteur. Le résultat est multiplexé dans un bloc qui couvre précisément 8 s. Un masque de bits (Msk) sélectionne les canaux les uns après les autres. Durant la fenêtre de mesure de 0,5 s, on compte tous les flancs montants issus du capteur correspondant. À l'intérieur du Flowcode, on définit une petite routine en C (voir listage en fin d'article). Vous trouverez dans le **tableau 2** les caractéristiques des capteurs et les masques de bits qui leur sont attribués.

Changement de gamme automatique

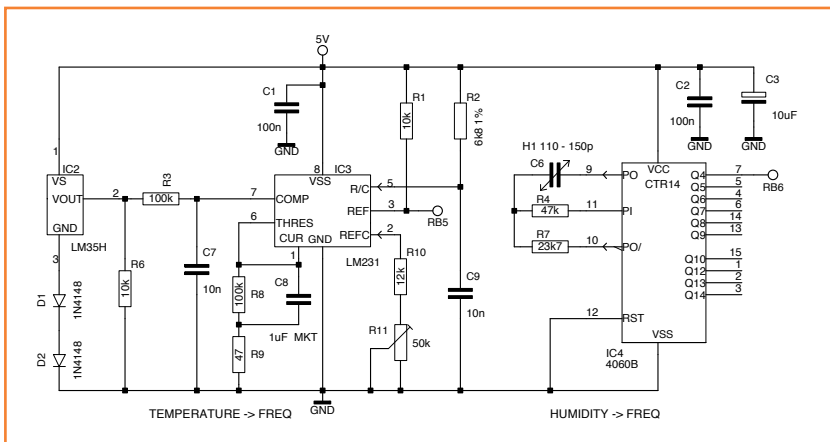
Nous l'avons déjà remarqué, la dynamique des valeurs de mesure de la lumière solaire et de celle du firmament est très vaste. Normalement, une mesure de fréquence n'offre qu'un pouvoir de résolution restreint. Le processeur aussi a ses limites en fréquence. Ce sont les échelles de sensibilité programmables du TSL230 qui nous sauvent la mise. C'est aussi la raison majeure du choix de cette puce de capteur.

Les caractéristiques du TSL230 montrent que quatre de ses broches servent à l'adressage de l'échelle de sensibilité (**tableau 3**). Que de possibilités ! Dans ce circuit, toutes les broches similaires des capteurs sont reliées en parallèle (**figure 7**), de sorte qu'ils réagissent tous simultanément au changement de gamme. Du coup, on réduit drastiquement non seulement le nombre de lignes de raccordement à la batterie de capteurs, mais aussi la taille du logiciel pour le PIC. Cette simplification améliore aussi la vitesse de mesure. Le seul inconvénient, c'est un relativement fort arrondissement des valeurs de mesure sur l'un ou l'autre canal quand apparaît une forte divergence d'éclairement entre les capteurs.



Figure 3.
Construction du capteur
avec ses filtres en verre.

Figure 4.
Le circuit qui rassemble les
capteurs de température et
d'humidité.



Il importe aussi de fixer des limites pour chaque TSL230 : elles sont de 200 Hz comme valeur plancher et de 4 kHz pour le maximum. Si ces valeurs sont dépassées en cours de cycle de mesure sur l'un des capteurs, au prochain temps mort, après la mesure de l'éclairement des cinq photomètres, l'instrument commutera sur une autre échelle de sensibilité, valable pour tous les capteurs de lumière.

Le lecteur de carte

Une carte de mémoire SD fonctionne sous 3,3 V. Comme l'instrument travaille en 5 V, il faut prévoir une conversion bidirectionnelle de la tension des signaux. Raison pour laquelle j'ai emprunté le principe du convertisseur de niveaux du MAX 3002 utilisé sur l'E-block EB037 (voyez cette partie de schéma à la **figure 8**). L'inconvénient, c'est qu'il n'est disponible que sous boîtier TSSOP et que souder ses 20 broches distantes de 0,65 mm demande... un certain entraînement.

On peut enregistrer directement les mesures sur une carte SD, mais il faut savoir que le formatage doit être en FAT16. On peut le réaliser avec la macro « PIC_FAT16.c » de Flowcode4. Vérifiez que vous disposez bien de la version 020710 ou plus récente.

La détection de la carte a lieu quand on appuie sur le bouton « INIT CARD ». S'il y en a une dans le lecteur, un nouveau nom de fichier est alors créé pour y enregistrer les données. Le nom de fichier est de la forme « ecl_#.txt » dans laquelle # est le numéro d'ordre tenu à jour dans l'EEPROM du processeur. Après chaque initialisation de la carte SD, ce nombre est incrémenté automatiquement d'une unité jusqu'à 255, après quoi on recommence à zéro.

Le protocole de mesure

Le format des données enregistrées sur la carte SD est comme ceci :

```
Eclipse data: 4298 1701 2429 3488 0 6.14
190 144 124 217 170 2429 3488 0 14.15 1948
1473 1296 2191 1724 2429 3488 0 22.15...
```

Pour être reconnu dans le traitement des données, le fichier commence toujours par le mot-clef « Eclipse data: ». Les valeurs de mesure s'écrivent alors à la file, séparées par un espace. La synchronisation de la série commence toujours

Tableau 3. Adressage du TSL 230						
S1	S0	SENSIBILITÉ		S3	S2	f ₀ SCALING (Divide-by)
L	L	Power Down		L	L	1
L	H	1x		L	H	2
H	L	10x		H	L	10
H	H	100x		H	H	100

Source : caractéristiques TSL230, Texas Instruments

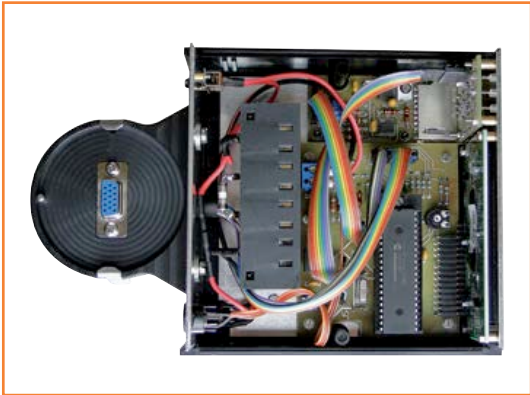


Figure 5.
Un coup d'œil sur l'intérieur de la console de commande et ...

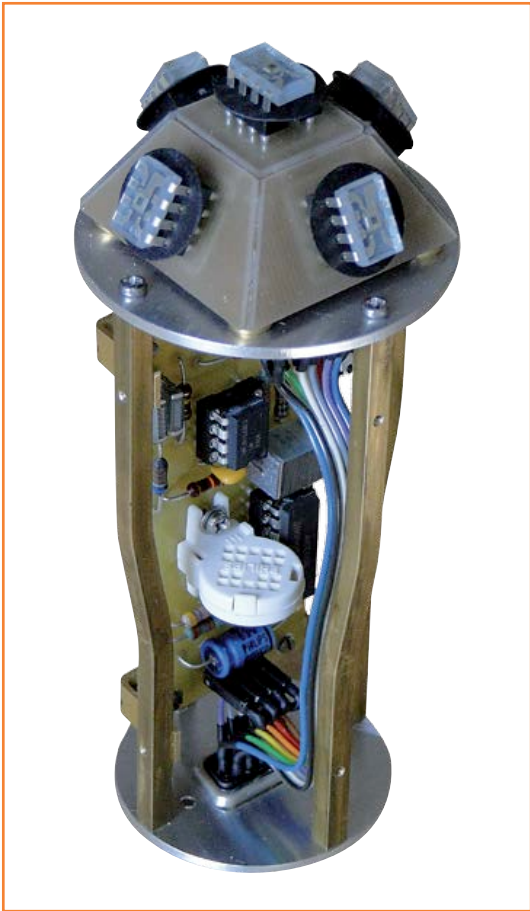


Figure 6.
.... petit coup d'œil sur l'intérieur de la partie capteurs.

Figure 7.

Voici l'image de l'une des cinq photocellules TSL230. Toutes les broches d'adresse des capteurs sont branchées en parallèle, la commande en est d'autant simplifiée.

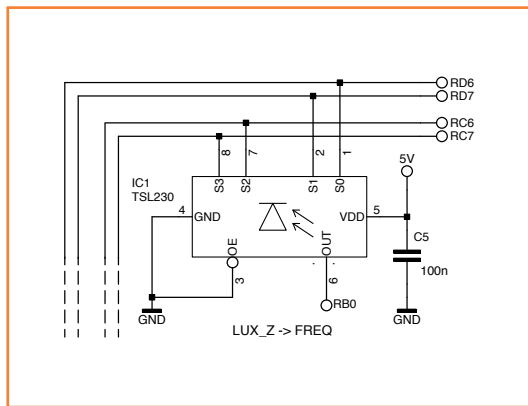


Figure 8.

Il y a conversion bidirectionnelle des niveaux de tension des signaux pour passer de 3,3 à 5 V et inversement.

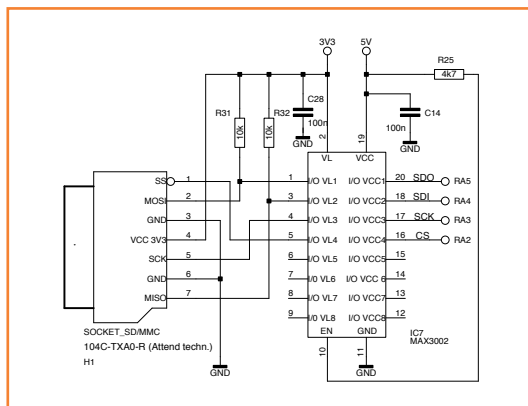
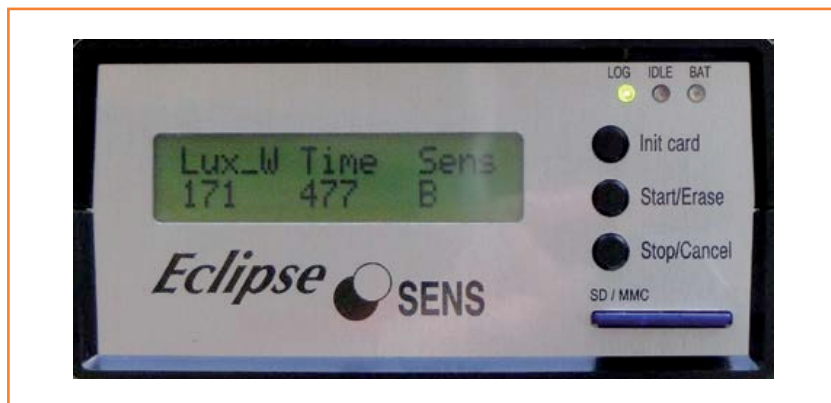


Figure 9.

Vue générale du panneau de commande et d'affichage.



par un nombre en « point flottant » : la valeur qui précède le point représente le nombre de secondes entières du temps de mesure, tandis que la valeur derrière le point est le code de la gamme d'éclairement. Viennent alors les valeurs des huit compteurs des capteurs et ainsi de suite.

Mode d'emploi

Après la mise en marche, un écran d'accueil apparaît brièvement. Puis, pendant un instant, les données du propriétaire, question d'augmenter les chances de récupérer l'appareil après une perte ou un vol. Bien entendu, il est possible de les modifier dans le programme *Flowcode*.

Ensuite, les mesures commencent automatiquement (cf. **figure 9**).

Signification :

- **Lux_W** : valeur de mesure du capteur (*counter value*), dans ce cas-ci, celle du photomètre ouest.
- **Time** : le nombre de secondes écoulées du temps de mesure, dans ce cas 477 s.
- **Sens** : indicateur de sensibilité des capteurs de lumière. A = la plus basse sensibilité, E = la plus haute.

Quand on allume l'instrument, après l'écran d'accueil, la LED jaune *Card Idle* s'allume pour signaler que la carte n'est pas encore initialisée. L'appareil effectue déjà des mesures, mais ne les mémorise pas.

Il faut appuyer sur le bouton « START/ERASE » pour lancer l'enregistrement des mesures. Le chronomètre de mesure commence alors à la seconde 0. Le processus s'arrête quand on pousse sur « STOP/CANCEL ».

La gestion de l'état des piles

Pour des appareils alimentés sur piles comme celui-ci, il est important d'en connaître l'état. Il n'y a pas de seconde session pour les éclipses ! Autant s'assurer de disposer de la tension suffisante pour effectuer des mesures pendant un certain nombre d'heures. Une LED rouge sur la console témoigne de l'état des piles.

Un autre aspect concerne la sécurité de fonctionnement du système d'enregistrement de données. Il

Tableau 4. Prochaines occultations totales du soleil

14-11-2012	Pacifique sud
3-11-2013	Afrique centrale
29-03-2015	Arctique, Atlantique nord et Spitsberg ¹
9-03-2016	Indonésie
21-08-2017	États-Unis

¹http://home.kpn.nl/nicole.franssen/SB_Eclips_Spitsbergen.htm

importe que la tension soit assurée à tout moment pendant l'inscription sur la carte SD. Le processeur veille donc à ce qu'aucune opération d'écriture ne soit effectuée quand la tension d'alimentation est trop basse. Le niveau de tension est continuellement mesuré et indiqué comme ceci :

$U_{bat} > 5,6$	LED éteinte	tension suffisante
$5,6 \geq U_{bat} > 5,3$	LED clignote	suffisante, mais vigilance requise
$5,3 \geq U_{bat} > 5,1$	LED allumée	insuffisante : Init card impossible
$U_{bat} < 5,1$	LED allumée	insuffisante : arrêt de l'enregistrement

Si la tension est trop basse pendant l'initialisation de la carte SD, l'avis « Battery too weak » s'affiche à l'écran.

Le dispositif pour la mesure de la tension des piles est extrêmement simple. La tension est appliquée, par l'intermédiaire d'un diviseur fixe de tension, à la broche RA0 du PIC, définie comme entrée analogique. Une diode Zener de 5,1 V aux bornes de la résistance qui va à la masse protège l'entrée d'une tension trop élevée quand l'adaptateur secteur est raccordé.

Le traitement des données

Les données enregistrées sur la carte SD sont des informations brutes, impropres à une interprétation directe. Le programme pour Windows ECLIPSENS.EXE sert à convertir les données. Il les recalcule pour fournir des valeurs d'éclairement en lux, de température en degrés Celsius et d'humidité en pourcents d'humidité relative. Il peut aussi montrer les courbes de mesure. La **figure 10** manifeste clairement la grande dynamique des courbes logarithmiques d'éclairement lumineux du firmament en lux. La routine de changement automatique de gamme du contrôleur PIC fait merveille en termes de précision. Lors de l'enregistrement de ces courbes, l'instrument était placé au voisinage d'un immeuble fortement éclairé. C'est la raison pour laquelle l'évolution est différente pour le canal du sud, la ligne en vert.

Il ne nous reste plus qu'à attendre la prochaine occultation totale du soleil. Les chasseurs d'éclipse connaissent sûrement par cœur le **tableau 4**.

(110647 - trad. Robert Grignard)

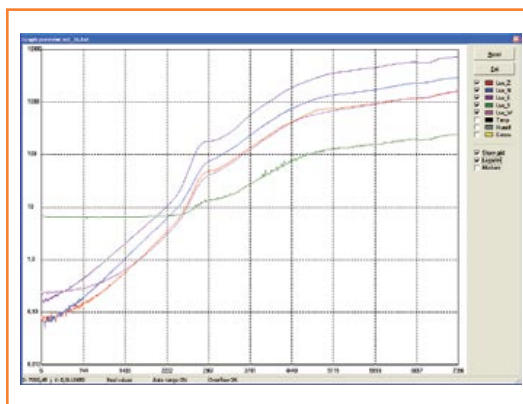


Figure 10.
Représentation graphique
des mesures pendant un
lever de soleil.

Liens

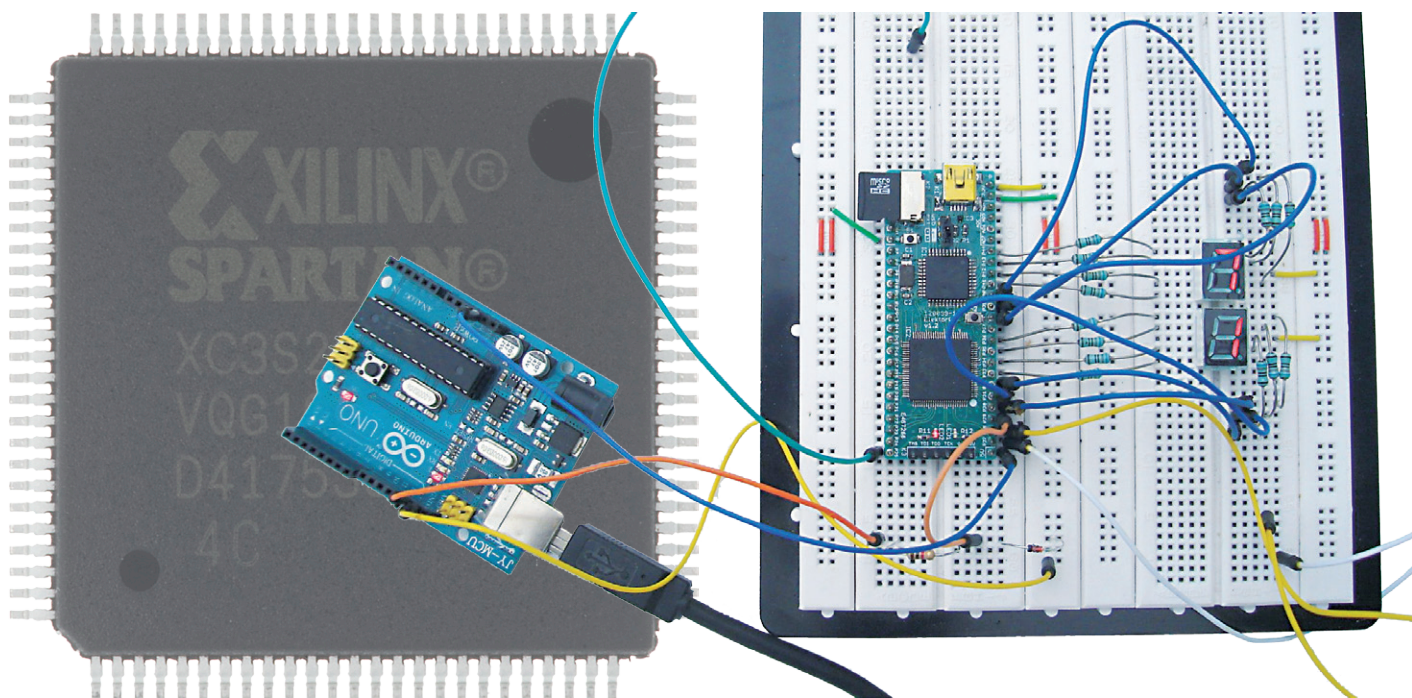
- [1] www.elektor.fr/110647
pour le téléchargement gratuit des codes source et hex, le logiciel de traitement des données pour WinXP et Win7, le schéma complet et d'autres information.
- [2] www.dutch.nl/rcott/eclips290306_1.htm
pour l'expérience acquise lors de l'occultation du soleil en 2006.

Listage

```
short SensorCount(unsigned char Msk, short SensorCount)
// Computing counter value of sensor
// INPUT : Msk = Mask to select number of sensor (1 to 8)
// INPUT : SensorCount = Counter value for raising pulse
// OUTPUT: SensorCount = new value after detecting raising pulse
{
    if ((FCV_SENSOR & Msk))
    {
        if ((FCV_SENSOROUD & Msk) == 0)
        {
            SensorCount++;
            FCV_SENSOROUD = (FCV_SENSOROUD | Msk);
        }
    } else {
        FCV_SENSOROUD = (FCV_SENSOROUD & (0xff-Msk));
    }
    return (SensorCount);
}
```

et l'homme créa sa puce (5)

250 000 portes à tricoter avec du code



Clemens Valens
(Elektor.Labs)

Développer une application sur FPGA sous forme de réseau de symboles logiques, c'est possible, nous l'avons vu, mais la méthode par programmation est souvent préférable en pratique. Les fonctions complexes se décrivent plus facilement sous forme d'algorithme que par un schéma. Tentons ensemble le développement d'une application de FPGA par programmation.

Dans le précédent épisode, pour tester une application, nous avons découvert VHDL et Verilog, des langages descriptifs du matériel. Cet ultime article va plus loin en proposant une application directement dans ces langages. Il m'a semblé fastidieux de poursuivre dans le bilinguisme. Mieux vaut choisir. Après plusieurs nuits blanches et sur l'avis de dizaines d'experts, j'ai choisi le VHDL. Les arguments ne manquent pas. L'un d'eux, le VHDL demande moins de temps de simulation parce qu'il est plus difficile d'arriver à un projet synthétisable. Valait-il mieux prendre le Verilog,

alors ? En fait, simuler demande souvent plus de temps que synthétiser, on en vient donc plus vite à bout en VHDL. Pour prendre une analogie avec la programmation des ordinateurs (fans de Verilog, s'il vous plaît, ne lisez pas ce qui suit), au lieu de flanquer du code au petit bonheur pour en faire après un algorithme qui marche dans le débogueur, on peut fort bien cogiter sur un algorithme et ensuite le déboguer pour voir s'il tient la route.

Comme application, j'ai choisi un décodeur DCF77. D'accord, c'est un classique, mais j'ai

mes raisons. Le signal radio de l'émetteur de cette horloge atomique, près de Francfort, on le reçoit facilement à peu près partout en Europe, avec un module bon marché. Le signal horaire est assez simple, c'est un enchaînement de 59 impulsions, une par seconde, qui peuvent avoir deux longueurs, 100 ms (0) ou 200 ms (1), elles représentent ensemble les bits de l'information horaire. La fin du code est marquée par l'absence de la 60^e impulsion, le bit 59. Un nouveau code est émis chaque minute. La date et l'heure transmises sont au format BCD et il y a quelques bits de contrôle pour vérifier la validité des données reçues. Voilà donc un signal adéquat pour faire ses premiers pas en VHDL (tout comme en Verilog, d'ailleurs), parce qu'il n'est pas trop compliqué de le reproduire avec une logique simple sur un afficheur à 7 segments. Pour ceux qui n'ont pas de récepteur DCF77 ou qui habitent trop loin de Francfort, j'ai même écrit un petit programme de simulation de DCF pour Arduino [5]. Il n'y aura donc pas de jaloux.

Commençons sur une note ingénue avec un projet fonctionnel de décodeur, en supposant un signal d'entrée normalement bon, conforme aux attentes et qui le reste dans le temps. Le projet n'en sera que plus simple.

Nous savons que le signal d'entrée est composé d'impulsions de 100 ms et 200 ms pour coder les zéros et les uns. Échantillons-le à 150 ms après le flanc avant de l'impulsion, nous verrons bien, comme à la **figure 1**, si c'est un 0 ou un 1. Profitons-en pour mesurer le temps entre deux flancs montants, de quoi détecter l'impulsion manquante qui marque le début du code horaire. Les bits découverts de cette manière, nous les introduisons dans un registre à décalage. Nous en sortons ensuite les champs BCD qui correspondent aux diverses valeurs de date et d'heure pour les appliquer à l'affichage à 7 segments. Du pseudo-code basé sur le C ressemblerait à ceci :

```
à faire pour chaque coup d'horloge
{
    compte = compte + 1;
    si (compte==150ms)
    {
        registre = (registre<<1) + signal;
    }
    si (flanc_montant(signal)==vrai)
    {
        si (compte>=1750ms)
        {
            restituer_contenu(registre);
        }
        compte = 0;
    }
}
```

À chaque coup d'horloge, le compteur est incrémenté. Quand il atteint 150 ms, le signal d'entrée est échantillonné et envoyé au registre à décalage. Quand un flanc montant est détecté dans le signal d'entrée, on vérifie si la valeur maximale est dépassée. Si oui, le registre à décalage est plein, on dispose alors des données complètes. On remet le compteur à zéro et c'est déjà tout. On pourrait en faire une usine à gaz, avec détection d'erreur et sécurité accrue, mais pas dans le cadre de cette introduction.

Comment faire ça en VHDL ?

C'est très facile, on peut simplement traduire le pseudo-code en VHDL. On obtiendra alors ce fragment :

```
1 process (clock) is
2 begin
3     if rising_edge(clock) then
4         counter <= counter + 1;
5         if (counter=t150ms) then
6             bits <= input & bits(58 downto 1);
7         end if;
8         if input_rise='1' then
9             if (counter>=t1750ms) then
```

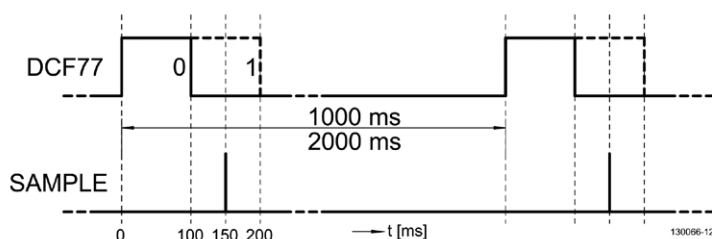


Figure 1.
Le signal de DCF77 avec les repères de l'instant d'échantillonnage. À cheval sur la dernière seconde, l'écart entre flancs montants passe à 2 000 ms, en guise de préambule.

```

10      data <= bits;
11      end if;
12      counter <= 0;
13      end if;
14      end if;
15  end process;

```

Il y manque sûrement les définitions et quelques subtilités de la syntaxe VHDL, mais l'algorithme est bien là. J'ai numéroté les lignes pour faciliter l'explication. Dans le code, counter est le compteur et bits, le registre à décalage de 59 bits de long. Commençons par la ligne 1.

Il s'agit ici d'un *Process*, c'est-à-dire qu'il sera exécuté par le FPGA. Pas de *Process*, pas d'activité. Il peut y en avoir plusieurs, tous exécutés simultanément. À l'intérieur de l'un d'eux, on travaille en principe du haut vers le bas. Le nôtre dépend du signal clock, ce qui veut dire que le *Process* ne s'exécute que quand ce signal change de valeur, il commence à la ligne 2 et se déroule jusqu'à la ligne 15.

La ligne 3 fait en sorte que tout ce qui s'exécute soit en synchronisme avec les flancs montants du signal d'horloge. C'est la fonction *rising_edge* qui nous l'assure. Dans beaucoup de code VHDL, nous rencontrons la construction

```

if clock'event and clock='1' then
..
end if;

```

qui a la même fonction que *rising_edge*, plus ou moins obsolète. Nous l'utilisons quand même. Il existe pareillement une fonction *falling_edge*. À la ligne 4, le compteur est incrémenté d'une unité. Simple, non ? Oui, mais il faut faire attention parce qu'utiliser la fonction d'incrément par le symbole « + » oblige à ajouter la bibliothèque *numeric_std* dans la liste des bibliothèques à utiliser. Nous verrons plus loin comment procéder. À la ligne 5, on compare la valeur du compteur à une constante qui correspond à 150 ms avec une fréquence d'horloge de 8 MHz, celle du FPGA. Le compteur arrive-t-il à 150 ms, la valeur actuelle du signal d'entrée est versée dans le registre à décalage bits (ligne 6). On le fait à partir de la gauche, à l'inverse du pseudo-code où le décalage avait lieu depuis la droite. Nous agissons ainsi parce que le code horaire démarre sur le bit de poids le plus faible, nous le présenterons alors dans le bon ordre. Après 59 impulsions,

le bit 0 est en position 0. L'opération de décalage paraît bizarre du fait que nous utilisons la fonction de liaison « & » qui relie simplement la partie gauche à la partie droite. La partie droite se compose des 58 bits, de 58 à 1, donc sans le bit 0, qui déménagent vers les positions 57 à 0. La partie gauche est le signal d'entrée d'un bit qui est maintenant le bit 58. Ne me demandez pas comment se débrouille ISE, ou plus exactement XST, ce qui compte, c'est que ça marche (et c'est le cas).

À la ligne 8, on observe s'il y a un flanc montant dans le signal d'entrée. Le fragment ne montre pas comment faire au juste – j'y reviendrai plus tard – mais quand il se présente, la ligne 9 vérifie si le compteur a dépassé la valeur de 1750 ms. Si oui, le contenu du registre à décalage est copié dans le registre de données pour traitement ultérieur. Ce n'est pas dans ce fragment-ci, voyez plus loin.

La ligne 12 met le compteur à zéro. Inutile de vider le registre à décalage, de toute façon, il se remplit chaque minute de nouvelles valeurs. Les lignes que je n'ai pas citées n'ont d'autre fonction que la conformité à la syntaxe du VHDL. Pour terminer ce fragment, il nous faut un détecteur de flanc et quelques questions de syntaxe pour les signaux d'entrée et de sortie du *Process*. J'ai aussi ajouté dans le projet définitif quelques signaux pour faire clignoter les LED sur la carte FPGA, de quoi s'assurer qu'il y a vraiment de la vie dans la boutique.

La détection des flancs

Détecter les flancs est une affaire fondamentale ici. J'avais commencé par comparer simplement la valeur courante du signal d'entrée à celle du coup d'horloge précédent. Elle n'était pas fiable et occasionnait des perturbations telles que la synchronisation se fourvoyait de temps en temps. La solution est manifestement d'ajouter une bascule qui permet de comparer la valeur précédente avec l'antépénultième, plutôt que l'actuelle avec la pénultième. Donc, ne pas comparer $t = n$ avec $t = n-1$, mais $t = n-1$ avec $t = n-2$.

Le lecteur attentif se demande sans doute pourquoi ne pas utiliser ici la fonction *rising_edge*. La réponse est que ça ne marche malheureusement pas. Si vous le faites, le synthétiseur XST croit qu'il s'agit d'un signal d'horloge et comme ce n'est pas vrai, il n'en sort pas et vous expédie un message d'erreur.

J'ai ajouté un détecteur de flanc, une fonction

en VHDL à voir dans le **listage 1**, en tant que module séparé, ce qui n'était pas indispensable. Le bonus, c'est qu'on pourra facilement l'utiliser dans un autre projet.

Le module débute par une instruction `library` suivie d'une instruction `use`. Elles servent à associer au module les fonctions standard et les signaux dont il a besoin. Si nécessaire, vous pouvez encore ajouter d'autres bibliothèques, comme `numeric_std` (pour le signe +). Contrairement à la plupart des langages de programmation, ces lignes ne sont pas valables pour la totalité du fichier dans lequel elles apparaissent, mais uniquement pour le bloc `entity` qui suit immédiatement avec les blocs `architecture` qui s'y rapportent. Chaque bloc `entity` doit donc normalement être précédé des instructions de bibliothèques.

Venons-en au bloc `entity`. Nous pouvons grosso modo le considérer comme le symbole que nous aurions utilisé dans un schéma. On y définit les entrées et sorties du composant dans la partie `port`. C'est là que se situent tous les signaux du type `std_logic` de la bibliothèque `std_logic_1164`, c'est dire qu'on a affaire à des signaux logiques qui peuvent prendre nombre de valeurs définies dans la bibliothèque : 0, 1, ou Z et encore bien d'autres. Les signaux terminés en `in` sont les entrées du module, ceux dont la désinence est `out` sont logiquement de sorties. Le bloc `entity` est suivi d'un bloc `architecture` dans lequel on détermine la fonction. Ce bloc porte un nom : `behavioral` et se trouve dans l'entité spécifiée `edge_detector`. Nous obtenons le nom `behavioral` quand nous demandons à ISE de faire un module VHDL. On trouve aussi souvent ici `rtl` (de *Register Transfer Level*, le niveau dans lequel le code VHDL est compilé), mais vous pouvez aussi lui attribuer vous-même un nom. Le détecteur de flanc est réalisé comme un `process` qui dépend du signal `clock`. Chaque fois que ce signal change de valeur, les instructions constituant le `process` sont exécutées.

La déclaration du `process` est suivie d'une liste de variables dont il a lui-même besoin, à l'exclusion des autres. Il n'y en qu'une ici, `history`, un registre à décalage de 3 bits. Cette variable est un vecteur, entendez par là qu'il compte davantage de bits. Remarquez que le vecteur est défini ici comme (1 to 3), tandis que le registre à décalage dans le décodeur DCF est défini comme (58 downto 0). Les deux sont valables, veillez

Listage 1. Module détecteur de flanc en VHDL.

```
library ieee;
use ieee.std_logic_1164.all;

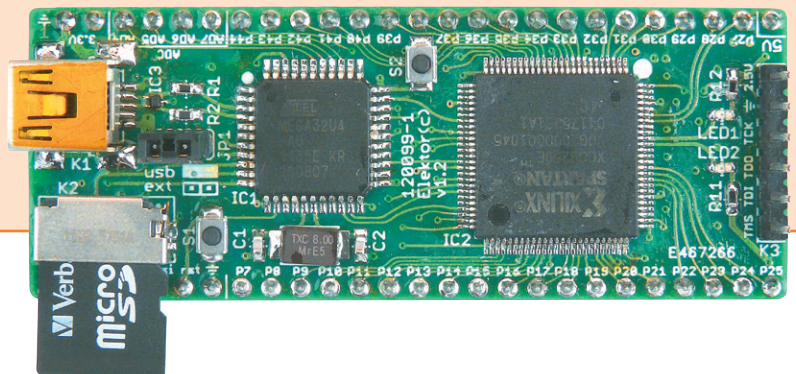
entity edge_detector is
  port (input : in std_logic;
        clock : in std_logic;
        rise : out std_logic);
end edge_detector;

architecture behavioral of edge_detector is
begin
  process (clock)
    variable history : std_logic_vector(1 to 3);
  begin
    if rising_edge(clock) then
      rise <= history(2) and not history(3);
      history := input & history(1 to 2);
    end if;
  end process;
end behavioral;
```

simplement à ne pas les mélanger par inadvertance quand vous devez les utiliser ensemble. Le détecteur de flanc ne réagit que sur un flanc montant du signal d'horloge. Il détermine alors la valeur du signal de sortie `rise` au moyen des bits 2 et 3 de `history` et place alors dans le

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

www.elektor.fr/120099



Listage 2. Le décodeur DCF77 avec détecteur de flanc.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity dcf77_decoder is
    port ( input : in std_logic;
          clock : in std_logic;
          data : out std_logic_vector (58 downto 0) );
end dcf77_decoder;

architecture behavioral of dcf77_decoder is

    component edge_detector is
        port (input : in std_logic;
              clock : in std_logic;
              rise : out std_logic);
    end component edge_detector;

    constant t1750ms: integer := 14000000; -- 1750 ms @ 8 MHz
    constant t150ms: integer := 1200000; -- 150 ms @ 8 MHz
    signal counter : integer := 0;
    signal bits : std_logic_vector(58 downto 0) := (others => '0');
    signal input_rise : std_logic := '0';

begin
    edge_detect: edge_detector port map (input => input,
                                         clock => clock,
                                         rise => input_rise);

    process (clock) is
    begin
        if rising_edge(clock) then
            counter <= counter + 1;
            if (counter=t150ms) then
                bits <= input & bits(58 downto 1);
            end if;
            if input_rise='1' then
                if (counter>=t1750ms) then
                    data <= bits; -- Transfer data.
                end if;
                counter <= 0; -- Clear counter.
            end if;
        end if;
    end process;
end behavioral;
```

registre à décalage la valeur actuelle du signal d'entrée, exactement comme dans le fragment du décodeur DCF77.

Vous pouvez constater que chaque bloc se termine par une instruction `end`, par exemple `end if`, éventuellement suivie du nom du bloc, pas mal de travail de saisie donc. C'est un des inconvénients du VHDL, il demande beaucoup de dactylographie.

La **figure 2** montre comment ISE voit le module détecteur de flanc sous forme de schéma. J'y vois quelque chose d'étrange, mais je me trompe peut-être. Disons simplement qu'ISE n'est pas un éminent dessinateur.

Si j'ai donné beaucoup d'explications sur ce module, c'est pour mettre en lumière la physiologie d'un module VHDL. Tous les modules se moulent dans le canevas de `library`, `entity` et `architecture`, le décodeur DCF n'échappe pas à la règle.

L'utilisation des modules

Nous voulons incorporer le détecteur de flanc au module DCF, alors, il faudra l'expliquer à XST d'une manière ou d'une autre. Il y en a plusieurs, nous utilisons ici la méthode des composants. Le **listage 2** indique comment les choses se mettent en place. Vous y trouverez le module complet de décodeur DCF, avec le bloc `entity` et son auxiliaire, le bloc `architecture`, plus les signaux d'entrée et de sortie, les variables locales et les constantes. Analysez-le à votre aise, prenez internet comme ouvrage de référence au besoin, ce n'est pas si difficile.

Importons le détecteur de flanc dans le module, en haut du bloc `architecture`, par une déclaration comme composant. Copions donc le bloc `entity` et remplaçons le mot `entity` par `component`. Il faut ensuite les connecter. Nous le faisons après le début du bloc `architecture` par l'instanciation d'un exemplaire, comme on dit. Il faut pour cela donner une étiquette, puis indiquer au moyen d'une table de correspondance des ports, ou `port map`, quels signaux (portes) du décodeur DCF (à droite dans `port map`, au milieu du listage 2) il faut relier aux signaux (portes) du détecteur de flanc (à gauche dans le même tableau de correspondance `port map`). Nous avons déjà les signaux d'horloge et d'entrée, puisqu'ils sont aussi des entrées ou des portes du décodeur DCF (voyez son bloc `entity`). Pour le

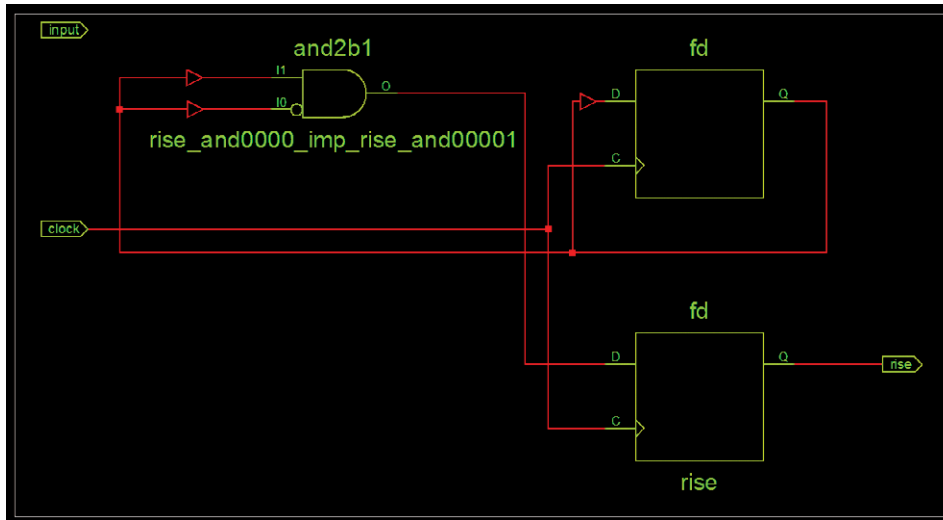


Figure 2.
Bizarre, le schéma qu'ISE dessine à partir de notre détecteur de flanc, présenté ici sur RTL Viewer (onglet *Design, Synthesize* - *XST* → *View RTL Schematic*). L'entrée du signal n'est reliée à rien, je ne comprends pas comment ça marche. Aurais-je fait une gaffe ?

signal de sortie *rise* du détecteur de flanc, nous devons ajouter un signal local du même type, je l'ai appelé *input_rise*. Nous pouvons dès lors l'utiliser dans notre process. Ce sera au cours d'une seule période d'horloge à 1 dès la détection d'un flanc montant dans le signal de DCF.

Si nous oublions de « *mapper* » un signal d'entrée d'un composant, nous risquons de recevoir un message d'erreur plutôt incompréhensible, qui tente de nous avertir que le signal oublié n'a pas

de valeur par défaut et qu'il peut donc rester en l'air. Vous voilà prévenu.

L'affichage

Je fais de mon mieux pour rester clair, mais du coup je suis plus long que prévu et mon chef n'aime pas ça. J'ai pourtant l'ambition d'expliquer en un seul article comment aboutir à un projet VHDL qui marche, sinon ce serait assez frustrant. Comme il me reste encore bien des choses à raconter, je fais l'impasse ici sur le code pour

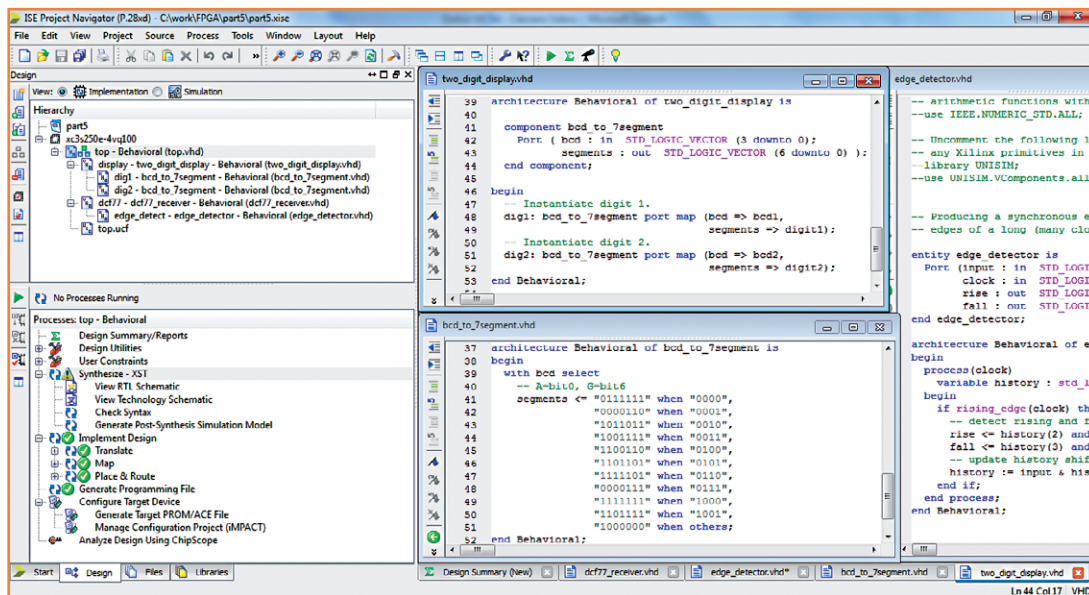


Figure 3.
Un écran d'ISE pour voir le code ainsi que la hiérarchie du projet et surtout les coches vertes révélatrices : on peut compiler jusqu'au fichier binaire.

l’affichage à 7 segments. J’ai déjà utilisé le même afficheur à deux chiffres dans la 3^e partie [3], où j’ai décrit un décodeur de BCD à 7 segments. Il s’agit d’un exercice courant dans les manuels, inutile de gaspiller de la place pour cela. Regardez son agencement dans la page du projet de ce numéro [5]. La **figure 3** vous en donne aussi le détail.

Toujours plus haut

Comme dans le précédent épisode, tout doit rejoindre le *top*, la différence, c’est qu’il sera constitué d’un module VHDL au lieu d’un schéma. Une information à communiquer à ISE par les *Design Properties*. Il faut créer d’abord un nouveau projet basé sur le précédent comme nous l’avons fait au début de [3], éliminer ensuite tous les schémas, mais conserver le fichier UCF. Ouvrir alors le *Design Properties* (p. ex. via le menu *Project*, en bas) et mettre *Top-Level Source Type* sur HDL et *Preferred Language* sur VHDL, si ce n’est pas déjà fait.

Nous pouvons à présent ajouter un nouveau fichier source en ouvrant le menu *Project* (ou d’un clic droit sur l’onglet *Design*). Choisir *New Source...*, sélectionner ensuite *VHDL Module*, lui donner un nom de fichier, par exemple *top*, et s’assurer que *Add to project* est coché. Un clic sur *Next*. Là, vous voyez un formulaire à remplir si, du moins, vous savez de quels signaux d’entrée et de sortie vous avez besoin, sinon ne remplissez rien. Si vous êtes prêt, cliquez sur *Next* et ensuite sur *Finish*.

Listage 3. Le niveau top du décodeur définitif de DCF77.

```
library ieee;
use ieee.std_logic_1164.all;

entity top is
    port ( dcf77_input : in std_logic;
          clk_in : in std_logic;
          hour_month : in std_logic;
          time_date : in std_logic;
          led1 : out std_logic;
          led2 : out std_logic;
          t_sample : out std_logic;
          digit1 : out std_logic_vector (6 downto 0);
          digit2 : out std_logic_vector (6 downto 0) );
end top;

architecture behavioral of top is

    component two_digit_display is
        port ( bcd1 : in std_logic_vector (3 downto 0);
              bcd2 : in std_logic_vector (3 downto 0);
              digit1 : out std_logic_vector (6 downto 0);
              digit2 : out std_logic_vector (6 downto 0) );
    end component;

    component dcf77_decoder is
        Port ( input : in std_logic;
              clock : in std_logic;
              tick : out std_logic;
              sync : out std_logic;
              data : out std_logic_vector (58 downto 0) );
    end component;

    signal data : std_logic_vector (58 downto 0);
    signal bcd1 : std_logic_vector (3 downto 0);
    signal bcd2 : std_logic_vector (3 downto 0);
    signal tick : std_logic;

begin
    display: two_digit_display port map (bcd1 => bcd1,
                                         bcd2 => bcd2,
```

ISE vous fabrique alors un fichier contenant un modèle pour y inscrire votre code VHDL. Ajoutez juste autant de fichiers que vous voulez faire de modules. Dans mon projet, il y en a cinq : *top*, le décodeur DCF77, le détecteur de flanc, le décodeur BCD vers 7 segments et l’afficheur à


```

digit1 => digit1,
digit2 => digit2);

dcf77: dcf77_decoder port map (input => dcf77_input,
                                clock => clk_in,
                                tick => tick,
                                sync => led2,
                                data => data);

process (clk_in) is
begin
    t_sample <= tick;
    led1 <= tick;
    if rising_edge(clk_in) then
        if time_date='1' then
            -- Show time.
            if hour_month='1' then
                -- Show hours.
                bcd1 <= data(32 downto 29);
                bcd2 <= "00" & data(34 downto 33);
            else
                -- Show minutes.
                bcd1 <= data(24 downto 21);
                bcd2 <= "0" & data(27 downto 25);
            end if;
        else
            -- Show date.
            if hour_month='1' then
                -- Show month.
                bcd1 <= data(48 downto 45);
                bcd2 <= "000" & data(49 downto 49);
            else
                -- Show day of month.
                bcd1 <= data(39 downto 36);
                bcd2 <= "00" & data(41 downto 40);
            end if;
        end if;
    end if;
end if;
end process;

end behavioral;

```

que s'opèrent les liaisons entre les pattes de la puce et le code VHDL. Il faut aussi que tous les signaux du fichier UCF soient présents, sinon gare aux remontrances d'ISE ! La nouveauté ici, c'est d'utiliser des vecteurs pour les broches des afficheurs. Nous indexons dans le fichier UCF le nom du vecteur, p. ex. *digit1(0)* avec le bit 0 du vecteur *digit1* de top.

Dans le bloc *architecture* de top, on fait appel à deux composants, *two_digit_display* et *dcf77_decoder*, on se servira d'un exemplaire de chacun. Avec un afficheur plus large de six chiffres, par exemple, nous pourrions opérer une instantiation de *two_digit_display*, à l'aide de trois étiquettes différentes, et les raccorder. Pour le raccordement des deux composants, j'ai défini plusieurs signaux supplémentaires qui ne sont pas nécessaires hors de top. Le signal *tick*, par exemple, sert à faire clignoter la LED1 à chaque bit reçu, mais comme cette LED n'est pas accessible sur une des broches de la carte FPGA, elle est branchée sur le signal *t_sample* qui est connecté à la broche P86 dans le fichier UCF. Ainsi pouvons-nous bien observer à l'oscilloscope le moment de

2 chiffres.

Vous découvrirez le *top model*, pardon, le module top dans le **listage 3**. Les noms des signaux d'entrée et de sortie de top, donc les signaux identifiés dans le bloc *entity* du module top, doivent correspondre aux noms dans le fichier UCF. C'est ainsi

l'échantillonnage par rapport au signal d'entrée de DCF77.

Le process de top ne fait rien de très excitant. En fait, c'est un multiplexeur qui, selon les niveaux de deux signaux de commande, *time_date* (P94)

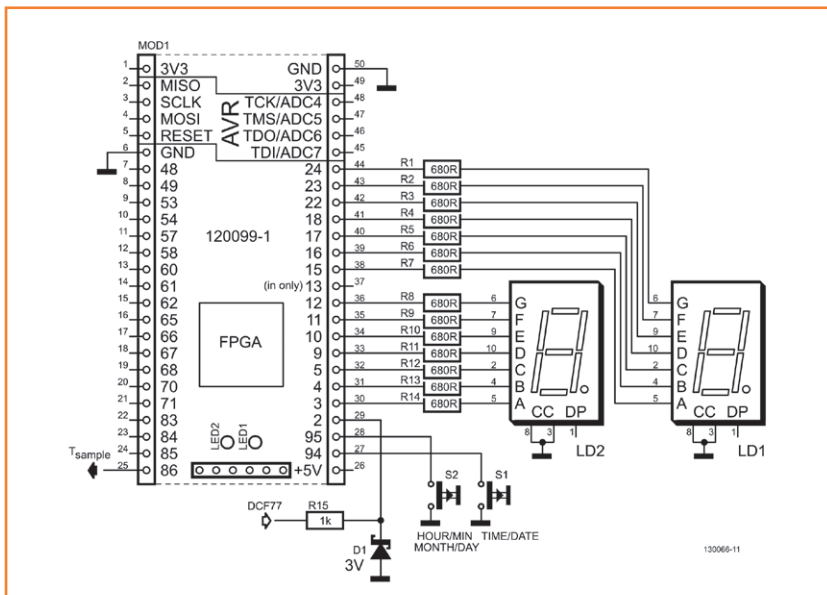


Figure 4.
Un schéma du décodeur DCF77 presque identique à celui de la partie 3 [3]. Comme le FPGA est vulnérable au 5 V, une diode zener limite le signal d'entrée à 3 V.

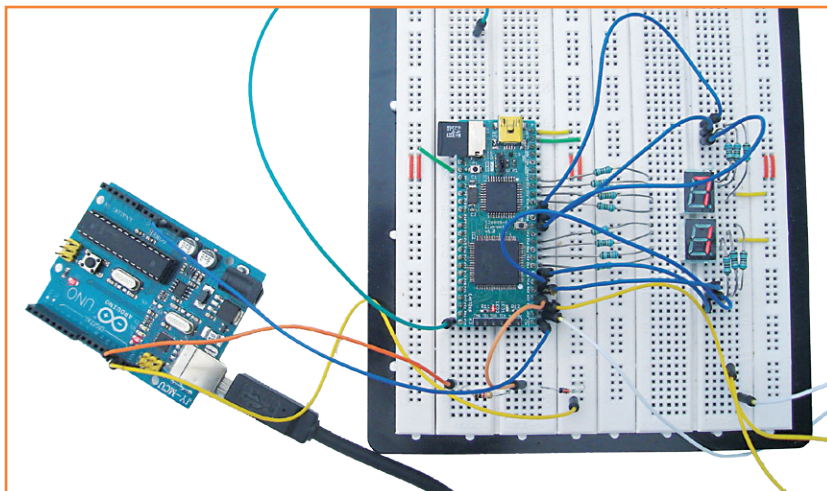
et hour_month (P95) donne à voir sur les deux chiffres à 7 segments, les heures, minutes, jour du mois et mois. Quelques couper-coller (« & ») remettent les bits dans le bon ordre à la sortie du grand registre à décalage pour en faire deux champs de 4 bits BCD que le décodeur traduira de BCD à 7 segments. Il est important de remarquer ici qu'il n'est pas nécessaire d'amener les vecteurs bcd1 et bcd2 explicitement aux afficheurs, le port map s'en est déjà occupé.

Lors de la synthèse du projet, vous serez gratifié de quelques avertissements. La cause en est qu'on n'utilise pas tous les bits du registre à décalage de DCF et que le bit 3 du vecteur bcd2 est toujours à 0, puisqu'on ne travaille pas sur des données à 8 bits. Seul le champ du millésime fait 8 bits, mais on ne l'a pas sorti. Ce que vous remarquerez aussi à la figure 3, c'est qu'une hiérarchie ordonnée s'est créée dans l'onglet *Design*, tout comme quand nous avons présenté le projet sous forme de schéma. Tout est donc comme avant. Nous avons vu que ISE dispose d'une visionneuse RTL (cf. fig. 2) capable de reproduire le code VHDL sous forme schématique et la boucle est bouclée. Un schéma, c'est du code VHDL et vice-versa.

Bon, c'est encore une sacrée tartine, ce dernier article ! Je me suis efforcé de transmettre les informations importantes, mais il vous faudra encore un certain effort personnel et de la persévérance pour mener à bien ce projet de construction personnelle. Bonne réussite !

(130066 – version française : Robert Grignard)

Figure 5.
Le circuit d'essai piloté par une carte Arduino pour simuler DCF77. L'affichage indique 17, l'heure simulée. La carte FPGA est ici alimentée par l'Arduino.



Liens

[1] 1^{re} partie : www.elektor.fr/120099

[2] 2^e partie : www.elektor.fr/120630

[3] 3^e partie : www.elektor.fr/120743

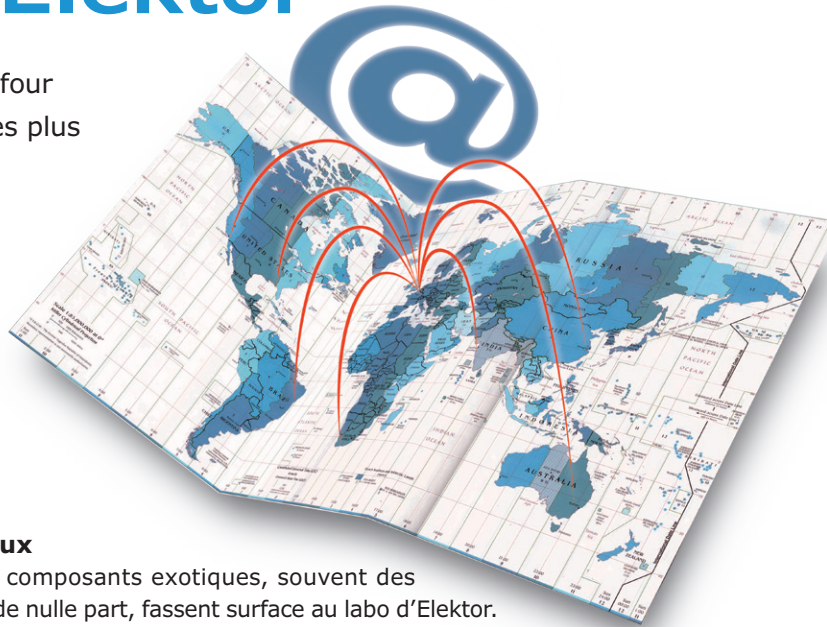
[4] 4^e partie : www.elektor.fr/130065

[5] 5^e partie : www.elektor.fr/130066

le monde d'Elektor

Le monde d'Elektor est un grand carrefour international où se croisent les voies les plus diverses et se créent les liens les plus improbables.

compilé par **Wisse Hettinga**



Tube mystérieux

Il arrive que des composants exotiques, souvent des antiquités, sortis de nulle part, fassent surface au labo d'Elektor. Ils sont alors le sujet d'interminables discussions, généralement assez fructueuses pour en dissiper rapidement le mystère. Cela n'a pas été le cas quand, il y a quelques semaines, un stagiaire a débarqué avec dans sa boîte à casse-croûte, bien calé entre les tartines, un tube dont les spécialistes ont attribué la paternité à Philips. L'inscription "I910" permettait de penser qu'il s'agissait d'un PM 1910 (PM = *photo multiplier*), mais nos collègues allemands et néerlandais, désireux d'en avoir le cœur net, ont sollicité l'aide de leurs lecteurs. Plusieurs d'entre eux, dont Christopher Kessler et Jan Swenker, ont

apporté de précieuses informations : pas moins de 6 pages dans le *Valvo Fotovervielfacher 1978-79* et une date de fabrication probable : 1978. Ces mêmes lecteurs ont attiré notre attention sur le livre *Hamamatsu Photomultiplier Tubes* de 1986, que l'on trouve sur l'internet, et qui donne un équivalent, qui porte la référence R1450... Bel exemple d'entraide au sein de la communauté d'Elektor !



(Ap)prendre les commandes

La théorie des systèmes de commande se concentre sur la manipulation des entrées d'un système pour modifier son comportement. Dans la pratique, les électroniciens connaissent bien la commande simple en boucle ouverte, comme p. ex. celle de moteurs pas à pas. Désormais il leur faut aussi apprendre à gérer la commande complexe en boucle fermée (p. ex. en robotique ou dans le modélisme aérien). Pour cela, ils ne retourneront pas à l'école, mais chercheront l'information sur l'internet, et ils se tourneront vers le logiciel et matériel *ouverts*. C'est ce qu'explique Brian Douglas, bien placé pour savoir de quoi il parle, puisqu'il est ingénieur des systèmes de commande. Sur son site YouTube, il met son savoir-faire au service d'une compréhension pratique de la théorie des systèmes de commande. Il tient également une rubrique chez nos confrères de Circuit Cellar, sur l'évolution technologique.

www.youtube.com/user/ControlLectures

www.circuitcellar.com/category/tech-the-future





1^{er} jour – DESIGNSPARK sur mesure

Neil Gruending

La fine équipe de RS Components a sorti la version 5.0 de DesignSpark PCB. Si comme moi vous aimez les programmes qui peuvent être configurés pour travailler de façon méthodique, vous apprécierez cette mouture.

Vous vouliez un programme sur mesure ? Suivez le guide ! Sous DesignSpark, tout peut être configuré globalement ou à l'aide de fichiers de styles et de modèles.

Bien démarrer

Avant toute chose, il faut comprendre que DesignSpark utilise des styles pour définir les règles de formatage relatives aux primitives

de conception comme les formes (rectangles, cercles, etc.), les polices ou les pistes. Comme dans un traitement de texte, chaque style a un nom, pour le référencer facilement. Un nom de style parlant évite d'avoir à vérifier ses attributs avant de s'en servir : un style nommé « Trou » conviendra si vous n'avez qu'un seul trou de connexion, mais un nom plus explicite est préférable : « Trou (0.45mmx0.95mm) » pour un perçage de 0,45 mm et une pastille de 0,95 mm. Libre à vous d'ajouter des styles personnalisés aux fichiers de styles (*technology files*), mais cet article se cantonne aux styles par défaut, p. ex. [Symbol Names], soit le nom du style utilisé pour les identifiants et le nom des composants. Vérifiez le chemin d'accès à ces fichiers de styles, défini dans *Settings/Preferences/General*. Par défaut, c'est : C:\Users\Public\Documents\DesignSpark PCB 5.0\Technology. Assurez-vous que ce chemin est correct et que le dossier *Technology* contient les fichiers .ptf et .stf, sinon DesignSpark ne les trouvera pas et sera difficile à utiliser.

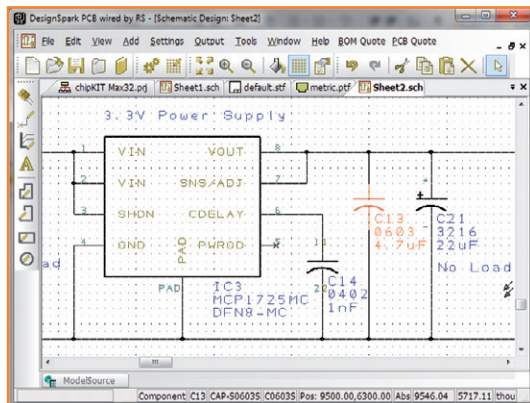


Figure 1.
Un schéma avec les paramètres par défaut.

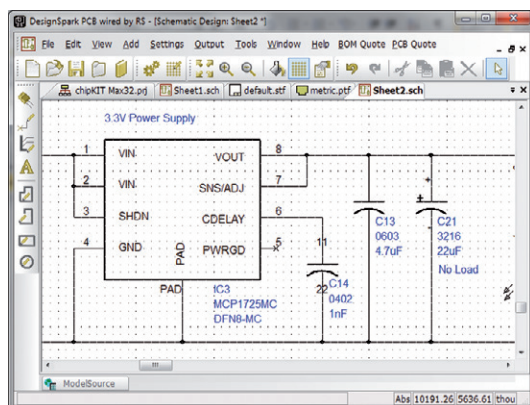


Figure 2.
Le schéma de la figure 1 avec la police Arial.

Fichiers de styles

Les fichiers de styles (*Schematic technology files*) permettent de configurer :

- le style des lignes utilisées pour les bornes et les nœuds
- le style des textes associés à des éléments comme les broches (police, taille, etc.)
- l'aspect de différentes lignes, p. ex. de documentation (continu, pointillé, etc.)
- l'aspect des lignes de connexion (continu, largeur de trait, etc.)

- les *nets* (lignes reliant des points équipotentiels), même s'il me semble préférable de le faire dans le schéma
- les classes de *nets* prédéfinies (masse, alimentation, etc.)
- les couleurs de divers éléments.

Allez dans *Settings/Design Technology/View/Colors* pour modifier ces paramètres.

Pour voir les paramètres par défaut, chargez un projet d'exemple, p. ex. chipKit Max32 (**fig. 1**).

Je trouvais la police un peu vieillotte, alors je lui ai préféré Arial, la police *True Type* standard. Après avoir joué quelques minutes avec les paramètres, j'ai obtenu la **figure 2**.

Comment entrer ces modifications dans un fichier de styles ? Ouvrez le fichier *default.stf* du dossier C:\Users\Public\Documents\DesignSpark PCB 5.0\Technology. Un schéma vierge s'affiche. Allez dans *Settings/Design Technology*. Dans l'onglet *Text Styles*, sélectionnez une police Arial de 80 pour les éléments *Net Names*, *Pin Names*, *Pin Numbers* et *Symbol Names*. J'ai également choisi Arial pour le style de texte *Normal*, mais avec une taille de 120. Dans le menu *View/Colors*, j'ai choisi le noir pour les champs *Pin Names* et *Pin Numbers*.

Une fois modifié, sauvegardez le fichier de styles. Pour l'utiliser, sélectionnez *default.stf* dans la fenêtre *Design Type* qui s'ouvre lors de la création d'un nouveau document.

Fichiers de modèles technologiques

Les fichiers *PCB Technology* sont l'un des meilleurs atouts de DesignSpark, car ils permettent de combiner en fichiers réutilisables les règles de base de conception et les empilages de couches. Supposez que j'ai défini un fichier pour des cartes bon marché à double couche, et un autre pour des cartes à 4 couches. Je peux alors choisir le fichier que je veux utiliser lorsque je crée une nouvelle carte. Cette fonction n'est pas répandue dans les autres logiciels de conception.

Les fichiers de modèles technologiques permettent de configurer :

- les unités (mm, mils, etc.) et la résolution
- les grilles, en particulier la grille de travail
- l'empilage des couches et les couleurs

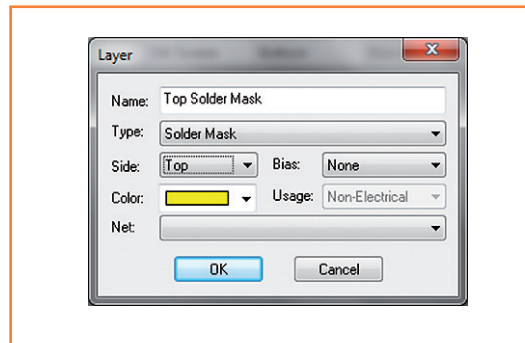


Figure 3.
Configuration d'une couche.

- l'espacement des pistes et les règles d'isolement
- le style des pastilles et des pistes, bien qu'en général vous définissiez le style des pastilles dans la bibliothèque de composants, et ici uniquement le style par défaut des pistes
- les classes de *nets* par défaut, mais d'ordinaire je les définis dans le schéma
- les règles d'auto-routage et de placement automatique
- les éléments de conception de base, comme les trous, contours de montage, etc.

Illustrons le bon usage des fichiers de modèles (*PCB Technology files*) avec l'exemple d'une carte à double face. Le plus simple est de partir d'un fichier préexistant. Sélectionnez le fichier C:\Users\Public\Documents\DesignSpark PCB 5.0\Technology\metric.ptf, et enregistrez-le sous un nouveau nom, p. ex. *my2layer.ptf*. Commençons par le menu *Settings/Units*. Il offre plusieurs unités et précisions ; pour moi ce sont toujours des mm et une précision de 4 chiffres après la virgule. Les préférences de grille se paramètrent dans *Settings/Grids*.

La configuration des couches se fait depuis l'onglet *Layers* du menu *Settings/Design Technology*. Certaines couches sont déjà définies par défaut : *Top Silkscreen*, *Top Copper*, *Documentation*, *Bottom Copper* et *Bottom Silkscreen*. J'ai ajouté les couches suivantes : *Top Paste*, *Top Solder Mask*, *Bottom Solder Mask* et *Bottom Paste*. Pour ajouter une couche, cliquez sur le bouton *Add* et sélectionnez les paramètres souhaités (**fig. 3**). Les boutons *Up* et *Down* de l'onglet *Layers* servent à modifier l'ordre d'empilage des couches. À la fin de votre configuration, vous devriez obtenir le paramétrage de la **figure 4**.

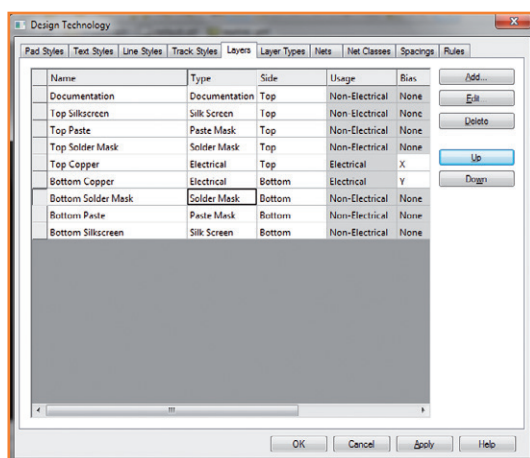


Figure 4.
Des couches bien empilées.

Passons aux règles d'isolement de l'onglet *Spacings*. Les isollements entre les différents types d'objets sont définis dans un tableau. Avec une simple carte double couche configurée pour une largeur de piste de 0,25 mm et un espacement de 0,25 mm, le tableau des espacements pourrait ressembler à celui de la **figure 5**.

N'oublions pas l'onglet *Rules*. Deux paramètres sont ici importants : *Annular Ring* (la largeur de la collerette qui entoure un trou de perçage) et l'espacement entre les composants. Dans l'onglet *Track Styles*, pour une carte 0,25 mm/0,25 mm j'utiliserais 0,25 mm pour les largeurs de piste minimale et normale du signal. La largeur normale de la piste d'alimentation reste à votre appréciation, mais pour sa largeur minimale je recommande 0,25 mm ; de cette façon les traces pourront être routées sur les pastilles. L'onglet *Pad Styles* enfin : pour une simple carte à deux

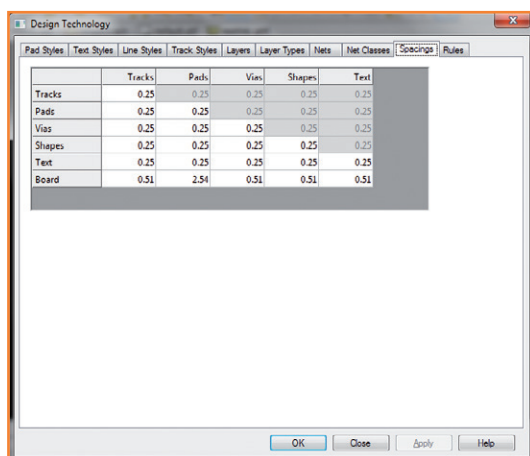


Figure 5.
Paramétrage des espacements.

faces, je choiserais un diamètre de perçage de 0,45 mm et une pastille de 0,95 mm. Vous pouvez définir et ajouter d'autres styles avec le bouton *Add Style*.

Vous disposez maintenant d'un ensemble de règles et de contraintes de conception faciles à réutiliser avec d'autres cartes à double face : créez une nouvelle carte de type *PCB Design*, et dans l'assistant sélectionnez votre fichier de modèle *my2layer.ptf*.

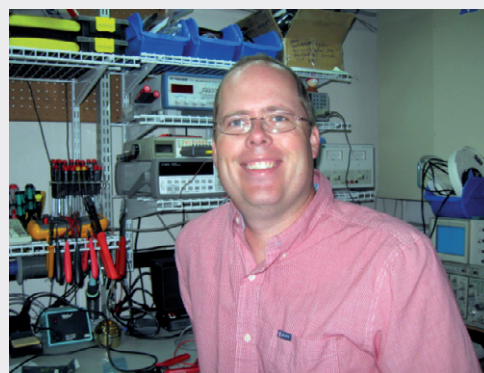
Conclusion

Nous pouvons encore améliorer notre environnement de travail en configurant également les bibliothèques de DesignSpark et en créant quelques modèles de documentation. DesignSpark est heureusement doté d'un vaste ensemble de bibliothèques qui facilitent la prise en main du programme.

(130172 – version française : Hervé Moreau)

L'auteur

Salut, je suis Neil Gruending. Dans mes activités d'ingénieur de conception électronique, j'ai utilisé au fil des ans bien des logiciels différents de CAO pour circuits imprimés. Je suis assez tatillon sur le choix de mes outils, car je recherche la plus grande efficacité possible afin d'accroître ma productivité chaque fois que possible. Je partage volontiers mes acquis sur mon site www.gruending.net et sur Twitter @ngruending.



détecteur à galène

Quand je commence un article, je cherche l'inspiration... sur mon bureau. Pour cette livraison de *Drôle de composant*, c'est une « diode à cristal » de type FRIHO D.R.P. qui a capté mon regard (Fritz Hofmann AG, Munich et Erlangen). Elle est citée dans un vieux *Radio Bauer Katalog* de 1926 (**fig. 1**) comme un des composants essentiels des antiques récepteurs radio à détection directe.

Je ne sais plus comment ni quand j'ai mis la main dessus. Dans mes jeunes années je passais beaucoup de temps à explorer la boutique de Quakkelstein à Vlaardingen, aux Pays-Bas. C'est vraisemblablement là que je l'ai trouvé et depuis lors il est resté sur ma table ou dans un tiroir. Ce composant fétiche nous ramène aux premiers jours de la radio. C'est un détecteur à galène – dit *cat's whisker*, moustache de chat – automatisé. En tournant le bouton, on change la position du petit fil (la moustache) sur le cristal de galène, jusqu'à trouver le point le plus favorable au fonctionnement de la diode : laisser passer le courant dans un seul sens. C'est comme ça qu'on écoutait la radio en ce temps-là. Pour obtenir un signal convenable, il fallait taquiner et ajuster la bobine, le condensateur variable et la diode de détection.

Que valaient ces diodes comparées à celles d'aujourd'hui ? Mesurer les caractéristiques d'une diode est vraiment simple. Pour obtenir une courbe correcte de la caractéristique IV, le minimum nécessaire comprend une alimentation, un voltmètre et un (milli)ampèremètre. La difficulté est d'obtenir un fonctionnement correct de la diode. On doit tâtonner un peu pour trouver le point où l'effet redresseur est à son maximum. Pour commencer, j'ai cherché à repérer ce point à l'ohmmètre, mais je suis vite arrivé à la conclusion que ça ne marcherait pas : les mesures variaient sans cesse. J'étais sur le point de laisser tomber quand j'ai décidé de mettre en œuvre un peu plus de matériel. Un générateur HP, un oscilloscope Hameg HMO1522 et une résistance de 330 Ω ont fini par donner quelques résultats prometteurs. Sur la **figure 2**, l'oscilloscope affiche un signal très flou, mais avec un peu d'imagination, vous pouvez voir comment la galène coupe le signal sinusoïdal d'entrée. La **figure 3** montre la caractéristique d'une diode moderne dans les mêmes conditions de mesure.

En regardant les résultats, je m'étonne du fonctionnement réel du détecteur FRIHO D.R.P. ! On trouve aujourd'hui une immense variété de diodes pour toutes sortes d'applications et dans une grande variété de boîtiers : diodes petits signaux, zener, varicap ou varactor, diodes tunnel... Toutes ont une sorte de parenté avec ce très vieux détecteur à galène.

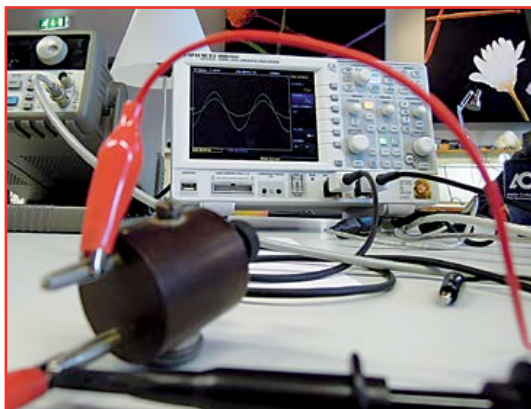
(130169 – version française : Jean-Paul Brodier)

Wisse Hettinga
(Elektor)

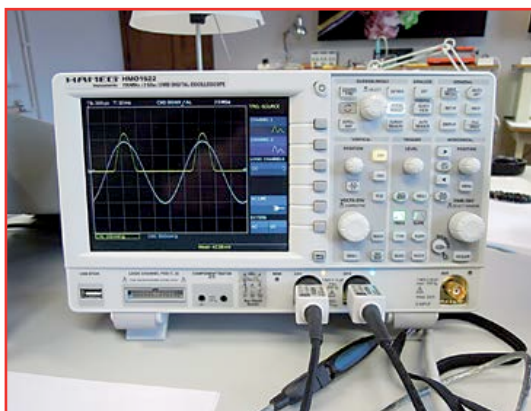
1



2



3



alimentation de labo multiple à tubes Wandel & Golterman NE-171 (vers 1963)

Le rêve de tout dépanneur radio

Jan Buiting

Donnez-moi un poste de radio à tubes construit entre 1920 et 1960 : il est probable qu'il sera défectueux du côté de l'alimentation, des commutateurs et du réglage de volume. Les défauts de l'alimentation peuvent être dus à des redresseurs épuisés (tubes ou sélénium), des transformateurs surchauffés, des fusibles coupés, des condensateurs électrolytiques desséchés, des nids

mesure quelques milliohms, ou si une résistance de puissance brûlée ou coupée se repère à l'œil ou au nez, mais les vrais problèmes se posent avec le fonctionnement intermittent d'une alimentation, du genre « ce vieux poste fonctionne correctement pendant vingt bonnes minutes, se tait, puis commence à ronfler, puis à fumer... et seulement le mercredi et le samedi soir, quand la tante Marie n'est pas là. » Ces problèmes, et de moins graves aussi, imposent la déconnexion complète de l'alimentation de la radio ou de l'amplificateur à tubes, et la prise en charge de toutes ses fonctions à la fois par un appareil reconnu en bon état. Pour rendre silencieuse la radio et découpler efficacement l'alimentation intégrée (ou ce qui en reste) il suffit de dessouder quelques fils (en général la tension de chauffage et la tension de plaque). C'est facile, mais si vous ne savez pas repérer ces fils, n'essayez pas de réparer quoi que ce soit « à tubes et vintage », tenez-vous-en à une électronique plus contemporaine. Ce qui est moins facile, c'est que vous avez besoin d'une alimentation en bon état pour toutes les tensions et courants nécessaires : 50 à 300 V CC pour les plaques, 6,3 V CA ou 4 V CC pour les filaments, -10 V à -100 V CC pour la polarisation négative des plaques. En un mot, une alimentation surdouée. Retour à une note positive : ce genre de matériel existe et les caractéristiques citées dans l'encadré ne sont pas un rêve. Ce sont celles de l'alimentation réglable multi-tension avec stabilisation électronique, modèle NE-171, fabriquée par la firme allemande *Wandel u. Goltermann*. Récemment on m'a donné une NE-171 à l'occasion de la liquidation d'un laboratoire d'électronique.



d'araignées, des crottes de souris, des résistances série ou shunt grillées. Vous ne pouvez pas éliminer un défaut avant de l'avoir localisé, ce qui révèle en même temps la cause du dysfonctionnement.

Exactement comme en algèbre, la solution des problèmes des vieilles radios (en fait, de tout échantillon d'électronique) passe par *primo* le remplacement d'un élément suspect par un élément reconnu bon et *secundo* le test. C'est facile si le primaire d'un transformateur d'alimentation

Made in Germany

Si les BM et les *Mercos* ont été pendant des décennies les symboles roulants de la qualité allemande, la marque *Wandel & Goltermann* a dominé le matériel électronique professionnel de test et mesure, depuis la fin de la Première Guerre

mondiale jusqu'à l'ère des circuits intégrés (c'est-à-dire avant l'absorption par Wavetek en 1999). Le matériel W&G se situait aussi dans le haut de la gamme des prix – et c'est toujours le cas sur eBay, un indicateur de la haute considération des collectionneurs.

Deux manuels

Bien qu'on trouve des informations diverses sur le NE-171 sur l'excellent site internet *Jogi's Röhrenbude* [1], rien ne remplace un manuel original imprimé. Coup de chance, j'ai hérité de deux manuels avec l'appareil. Le plus ancien est daté du 27 juin 1958. Il traite du lot de production « C », ce qui me permet de supposer que la conception originale est plus ancienne. La présentation de ce manuel est grossière, apparemment ronéotypé (chercher ce mot) à partir d'un original dactylographié (chercher ce mot). Une version postérieure

par un autocollant d'avertissement bien visible « *Nicht bestanden* » (refusé), mais je doutais que l'appareil soit en panne. Pour trois raisons au moins :

1. C'est un authentique produit *Wandel & Goltermann*.
2. Il venait d'un environnement où une manipulation et une maintenance correctes étaient assurées par des lecteurs d'Elektor anciens et actuels.
3. J'ai sur ma paillasse un transformateur variable complètement isolé pour assurer à l'appareil un démarrage en douceur après plusieurs années de stockage.

J'ai donc arraché l'autocollant et mis en marche. L'appareil a fonctionné immédiatement. Voilà deux scénarios qui expliquent qu'un appareil en état de marche soit condamné par un autocollant « Refusé » :

1. En 2006, lors d'une séance de contrôle de rou-



du manuel, datée de sept. 1962 par W&G, est en caractères d'imprimerie, sur papier glacé, imprimée et non ronéotypée. Ce livret orange brillant couvre les lots de production F, G et H. Mon exemplaire de NE-171 porte le numéro 16432H, ce qui situe sa date de fabrication en 1962, peut-être 1963.

Nicht bestanden (en juin 2006)

Malgré son âge, ma NE-171 est impeccable à l'extérieur, juste un peu poussiéreuse à l'intérieur. L'interrupteur marche-arrêt était couvert

tine, un agent de la *Commission des Étiquettes de Contrôle* a connecté son multimètre *Fluke 2012* à la sortie de tension réglable 50-300 V et mis en marche la NE-171. Voyant une saute de tension, l'agent inquiet a coupé le courant instantanément et apposé l'infamant autocollant rouge et blanc. Lors de l'inventaire annuel de l'équipement du labo, cette NE-171 a été déclarée « défectueuse » et garée, en attendant sa destruction. Sept ans plus tard, elle a été sauvée juste à temps par un technicien du labo, fan de *Rétronique*. J'ai reçu

un courriel, vous connaissez la suite.

2. Comme le 1. ci-dessus, mais au lieu de la saute de tension, aucune tension mesurable immédiatement. Nous verrons les explications plus loin.

Compacte et puissante

Il est certain qu'il existe de nombreuses alimentations haute tension réglables de différents fabricants, des années 1950 à 1970, toutes conçues pour l'utilisation avec du matériel à tubes, pour l'expérimentation au laboratoire ou pour le dépannage. Toutes sont encombrantes, comme la *Van der Heem 8619* décrite en avril 2007 d'Elektor [2]. Comparée à ce tank, la NE-171 W&G est un poids plume, tout en fournissant bien sûr un courant et une tension à peine moindres sur la sortie HT réglable. J'aime le format de la NE-171 – elle ressemble à un gros livre et c'est le compagnon parfait de beaucoup de mes activités de réparation sur les radios à tubes.

EL156 « à donf »

Du point de vue électronique, la conception de la

partie HT réglable de la NE-171 est conventionnelle. Seule une section du schéma est représentée ici. Nous avons un tube régulateur en série (Rö1), un amplificateur régulateur (Rö2) et un réseau de résistances, avec un potentiomètre pour régler la tension de sortie. La tension de référence de -85 V , fournie par un tube stabilisateur 85A2, est appliquée à la ligne commune du réseau de résistances inférieur.

L'essentiel n'est pas dans la conception basique, mais dans le choix des composants. Alors que beaucoup d'autres alimentations HT ont une paire d'EL34 (6CA7) comme ballast, on a ici une EL156 combinée à une pentode à pente très raide de type EF804S. Cette EL156 est bien notée d'un bout à l'autre : courant de cathode : 180 mA max. ; dissipation d'anode : 40 W max. ; tension d'anode : 800 V max. Ce composant est l'un des tubes les plus prestigieux mis au point par *Telefunken*, un sommet de leur expertise – admirez cette feuille de caractéristiques [3].

L'EL156 est également un grand tube de sortie audio.

Caractéristiques NE-171	
Tension continue 1 stabilisée, 6 gammes et réglage continu.	50 à 300 V
Courant de sortie maxi (DV2 = 0)	100mA
Résistance interne (175 V, 20 à 200 mA)	$\leq 2\ \Omega$
Précision de la tension de sortie	$\leq 1\% \pm 2$
Tension de ronflement (300 V; 100 mA)	environ 0,1 mVeff
Fluctuation de la tension de sortie ($\pm 10\%$ fluctuation secteur ; 300 V ; 100 mA)	environ $\pm 300\text{ mV}$
Tension continue 2 (50 mA max.) non stabilisée, flottante, pôle négatif commun avec Tension continue 1	environ 520 V
Courant de sortie maxi (TC1= 0)	50 mA
Résistance interne	1 k Ω
Tension d'ondulation (50 mA)	environ 0,5 Veff
Tension continue 3 stabilisée, flottante, pôle positif commun avec le pôle négatif de Tension continue 1	0 à -10 V / 0 à -100 V (commutable)
Courant de sortie maxi	
0 à -10 V	jusqu'au court-circuit (environ 1,5 mA)
0 à -100 V	jusqu'au court-circuit (environ 3,5 mA)
Ronflement (1 mA)	environ 20 μVeff
Régulation ($\pm 10\%$ fluctuation secteur)	environ $\pm 0,1\%$
Tensions alternatives	
3 sorties chauffage séparées, flottantes	4 V / 6,3 V 3 A
	6,3 V / 3 A
	18 V / 20 V 1 A
Liste des tubes	EL156, EF804S, ECC82, 85A2
Tension secteur alternatif	220 V, 45 à 60 Hz
Dimensions	140 x 315 x 249 mm
Poids	environ 10 kg (22 lbs)

La tension de sortie de la NE-171 se règle par la combinaison d'un commutateur de gammes à six positions à intervalle de 50 V et d'un potentiomètre de 0 à 50 V en plus de la valeur de la gamme. Dans mon cas, le commutateur de gammes semblait bloqué à la position 150 V, ce qui fait que je ne pouvais régler que des valeurs comprises entre 150 et 200 V. Il a suffi de libérer cet axe, à l'intérieur, le commutateur, la came et le bras à ressort étaient en état de marche.

La sortie réglable 50-300 V a un interrupteur marche-arrêt (rouge) séparé et un fusible réarmable calibré à 125 mA. Le dernier manuel NE-171 déconseille à l'utilisateur d'activer la sortie 50-300 V avant que l'appareil ait été sous tension environ une minute, sous peine de pointes de tension ou d'absence de tension. Vous savez maintenant pourquoi cet autocollant « *Nicht bestanden* » a été apposé, proba-



blement par quelqu'un de pas trop familier avec les tubes (« *Nicht verstanden?* » Pas compris ?) et sûrement pas un membre de l'équipe de ce labo. Aucun problème avec les trois condensateurs électrolytiques multiples de l'appareil. J'ai constaté que trois ponts redresseurs de type BY179 étaient installés à la place des B250C150 du schéma. La couleur verte des BY159 signale clairement qu'ils datent des années soixante-dix.

Je n'oublie pas une des grandes propriétés de la NE-171 : toutes les sorties sont flottantes par rapport au châssis.

NE-171 en tandem

Les alimentations réglables HT de deux NE-171 peuvent être connectées en parallèle ou en série, pour délivrer un plus fort courant (0 à 200 mA) ou une plus forte tension (100 à 600 V) de sortie. La configuration en parallèle nécessite le retrait d'une fiche à l'arrière de l'appareil et le branchement d'un câble de liaison. Le câble peut être de fabrication maison, à condition d'avoir pu mettre la main sur les fiches spéciales à cinq broches.

Utilisation pratique

Chaque fois qu'un poste de radio à tubes atterrit sur mon bureau, je déconnecte rigoureusement TOUS les circuits d'alimentation, condensateurs électrolytiques compris, et j'alimente les circuits radio par la NE-171 – après avoir consulté le schéma du poste, naturellement. D'abord, je laisse les tubes radio chauffer en appliquant la tension de chauffage pendant une quinzaine de minutes. Cela révèle les défauts de câblage des filaments et les filaments à coupures intermittentes.

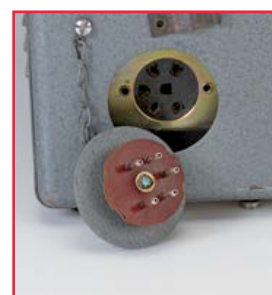
Ensuite, j'applique la tension des plaques, en démarrant prudemment à 50 V et en augmentant jusqu'à la tension nominale (habituellement 250 V). Dans huit cas sur dix, la radio fonctionne plus ou moins. Ensuite, je reforme les condensateurs électrolytiques, en démarrant à 50 V et en montant progressivement jusqu'à 300 V, avec l'extincteur à portée de main. Ensuite je remonte jusqu'aux entrées du redresseur. En cas de doute, un milliampèremètre est inséré dans la ligne du point milieu de la HT.

Toute la procédure peut se dérouler en toute confiance avec une alimentation fiable, ce qui est rassurant. Les réparations des sections BF, FI et HF terminées, je suis toujours un peu triste au moment de déconnecter la NE-171 et de rétablir les connexions de l'alimentation de la radio – pas à pas, bien sûr, et en mesurant les tensions avec une main dans la poche.

(130030 – version française : Jean-Paul Brodier)

Liens

- [1] www.jogis-roehrenbude.de/, chercher NE-171
- [2] Alimentation de labo HT, Rétronique, Elektor avril 2007, www.elektor.com/075036
- [3] www.hifitubes.nl/weblog/wp-content/telefunken-el156.pdf



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

Hexadoku casse-tête pour électroniciens

Puisque vous êtes adepte du jeu que nous vous proposons ici chaque mois, il y a de fortes chances pour que vous connaissiez des personnes qui elles-mêmes pratiquent le sudoku. Avez-vous déjà songé à leur faire découvrir nos grilles hexadécimales ? Il leur suffira de remplir la grille selon les règles, et de nous envoyer leur solution. Elles gagneront peut-être l'un des quatre chèques-cadeaux Elektor mis en jeu. Et deviendront peut-être des lecteurs assidus d'Elektor.

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 ran-

gée, 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 tirons au sort l'une des réponses internationales correctes reçues dans les délais ; son auteur recevra un chèque-cadeau d'une valeur de **100 €** à valoir sur des **circuits imprimés Eurocircuits**. Nous offrons en outre 3 chèques-cadeaux à valoir sur des livres d'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} juillet 2013** :
Elektor c/o Regus Roissy CDG – Le Dôme – 1, rue de La Haye
BP 12910 – 95731 Roissy CDG
Courriel : hexadoku@elektor.fr | www.elektor.fr/hexadoku

Les gagnants

La solution de la grille du numéro d'avril (417) est : **934CB**

Le gagnant des **100€** à valoir sur des circuits imprimés **Eurocircuits** est **David SMART** (Waterloo (Ia), États-Unis).

Les **3 chèques-cadeaux Elektor** d'une valeur de **50 €** chacun vont à :

Joseph REDING (Schweich, Luxembourg) | Karsten Krummeich (Dülmen, Allemagne) | Paul Baak (Voorburg, Pays-Bas).

Bravo à tous et félicitations aux gagnants !

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

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

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.

Professional Quality
Trusted Service
Secure Ordering



Elektor PCB Service at a glance:

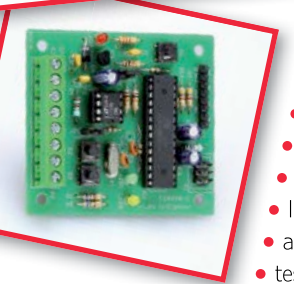
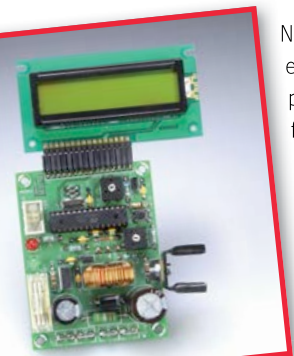
- ➔ 4 Targeted pooling services and 1 non-pooling service
- ➔ Free online PCB data verification service
- ➔ Online price calculator available
- ➔ No minimum order value
- ➔ No film charges or start-up charges

Delivery
from 2
working
days

●bientôt dans Elektor

numéro hors-gabarit | été 2013

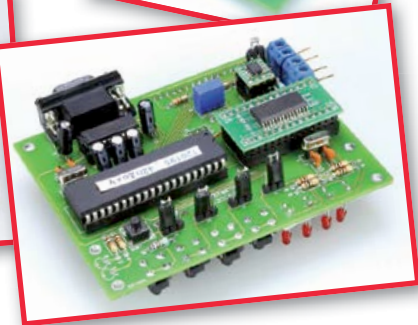
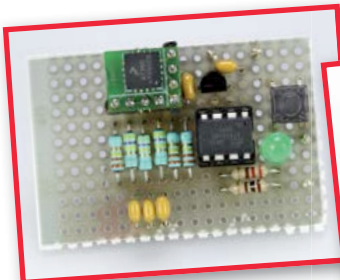
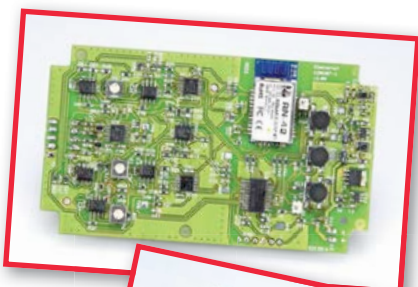
édition spéciale, avec encore plus d'idées, de projets, de réalisations, d'astuces et de trouvailles



Notre prochain numéro renouera avec la tradition estivale. Source d'inspiration inégalée, appréciée par les électroniciens du monde entier pour sa fécondité, sa diversité et sa stimulante créativité, cette édition sera un mélange d'articles de divers formats. Au moment de mettre sous presse le numéro de juin, les concepteurs et les rédacteurs sont encore la tête dans le guidon. Voici un avant-goût des articles sur lesquels ils travaillent et que vous retrouverez donc (probablement) au sommaire de l'édition 2013 du célèbre **numéro double d'Elektor**.

avant-première :

- testeur CAN
- commande de moteurs pas-à-pas sur ElektorBus
- allumage CDI pour motocyclettes
- limiteur de courant de démarrage
- ampli audio en classe D avec 555
- testeur de servos
- électrocardiogramme sur Android
- power pack USB
- horloge Arduino à tubes Numitron
- assistant de garage
- enregistreur de températures multicanal
- commande IR sous Android
- horloge numérique ultra-précise
- interface universelle de mesure de précision
- clé USB AVR multifonctionnelle



Informations préliminaires non contractuelles.
Le numéro de juillet-août paraîtra le 26 juin.

Publicité



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 3,99 €.



elektor labs

est ouvert
24 heures sur 24
7 jours sur 7

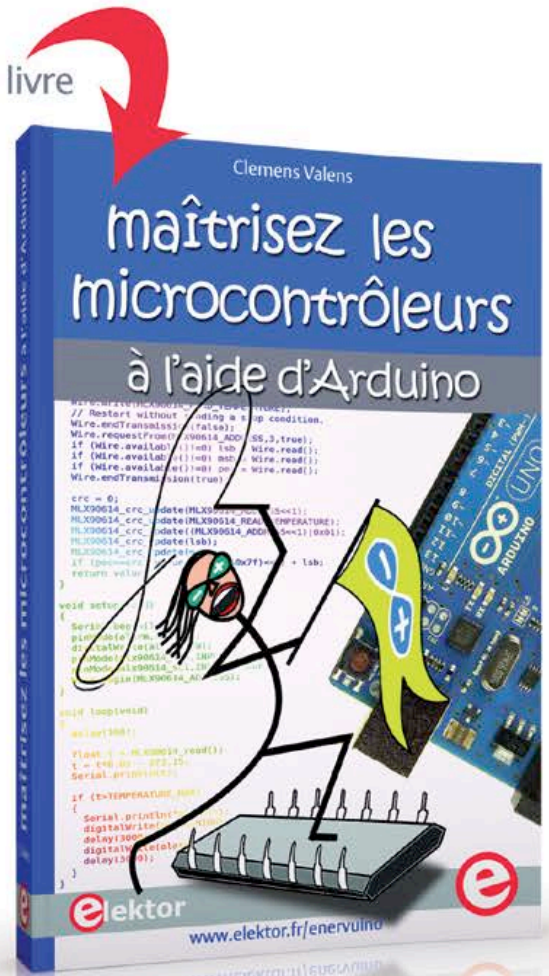
Participez à l'élaboration des
projets sur
www.elektor-labs.com

« Enfin une méthode à base de montages Arduino qui marche pour se débarrasser définitivement de ses amis et de sa famille... »
Et pour se retrouver enfin seul et libre de passer tout son temps à apprendre la programmation des microcontrôleurs. »

Ce livre
n'est pas
banal
pas superficiel
pas réducteur ni même simplificateur
pas barbant, pas soporifique
pas du genre « 30 applications rigolotes »
pas lâche, pas infantilisant
pas appuyé sur du matériel introuvable et/ou hors de prix
mais il est

original, par le fond et la forme
consistant, profond, complet
plaisant, souvent drôle, parfois hilarant
ou déconcertant, selon le tempérament de chacun,
généreux : il ne se contente pas de vous donner des envies,
mais donne aussi les moyens de les satisfaire,
courageux : il aborde les sujets laissés savamment
dans l'ombre par les autres, p. ex. les interruptions,
formateur, encourageant, exigeant et stimulant,
enraciné sur la plateforme de prototypage rapide Arduino,
la plus largement diffusée dans le monde

maîtrisez les microcontrôleurs à l'aide d'Arduino
352 pages – 10 chapitres – 13 réalisations inédites
sommaire et extraits : www.elektor.fr/arduino



logiciel gratuit téléchargeable

changez de loisirs devenez dresseur de puces !

elektor

À la fin du livre, le lecteur qui n'aura rien sauté sera capable
de mettre en œuvre n'importe quel microcontrôleur. Clemens Valens

ANALYSEURS DE SPECTRE

www.hameg.com | Great Value in Test & Measurement

HAMEG®

Instruments

A Rohde & Schwarz Company

Special été

Bénéficiez de 10% de remise pour l'achat d'un oscilloscope HMO et d'un analyseur de spectre HMS!

*Offre valable jusqu'à fin Octobre 2013



NOUVEAU!

Disponible par
upgrade firmware
A télécharger gratuitement
sur www.hameg.fr

HAMEG INSTRUMENTS augmente la fréquence de sa gamme d'analyseur de spectre série 1000 de 1 GHz à 1,6 GHz.

Avec l'achat d'un analyseur de spectre HMS1000, HMS1010 ou HMS1000E, les clients HAMEG feront l'expérience d'une augmentation immédiate de 60% de fréquence en plus, sans coût additionnel.

+60%

DE FRÉQUENCE EN PLUS

HMS1010 | HMS1000 | HMS1000E

de 1 GHz à 1,6 GHz


ROHDE & SCHWARZ

Rohde & Schwarz France | 9/11 rue Jeanne Braconnier
92366 Meudon-la Forêt | Tél : 01.41.36.10.00 | Fax : 01.41.36.11.11
www.rohde-schwarz.fr | contact.rsf@rohde-schwarz.com