

elektor

NUMÉRO DOUBLE : 132 PAGES – PLUS DE 20 RÉALISATIONS INÉDITES



éclairage annulaire à LED

● scanner de radionavigation aérienne | limiteur de transitoires
n'oublie pas ta brosse-à-temps | **FPGA : et l'homme créa sa puce**
bougie à LED bicolore | APU : alerte à la pile usée ● matériel libre
● NAGRA IV : LE magnétophone à bande ● des nouvelles du labo virtuel





DESIGNSPARK

NOUVELLE VERSION DESIGNSPARK PCB 4

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

À découvrir sur
www.designspark.com

DÉVELOPPÉ PAR



explorez



Les cinq nouveaux continents du monde d'Elektor

elektor.magazine : **le vaisseau amiral**

elektor.labs : **le drakkar**

elektor.community : **le paquebot**

elektor.post : **le catamaran**

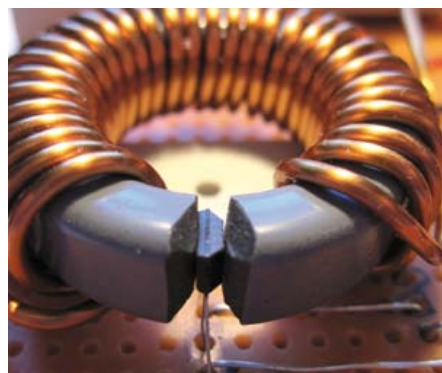
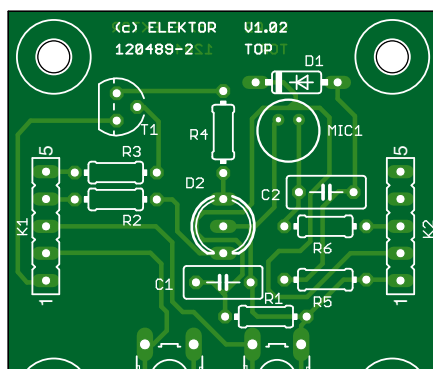
elektor.store : **le cargo**



Rejoignez la communauté Elektor :
embarquement immédiat !



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



● industrie

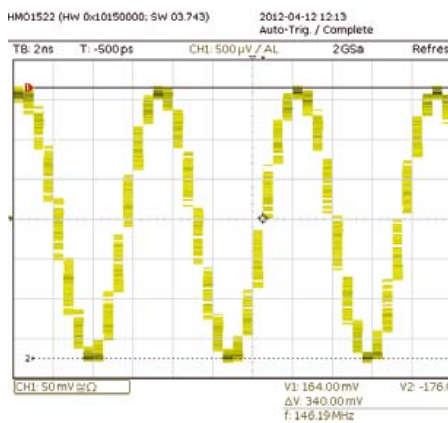
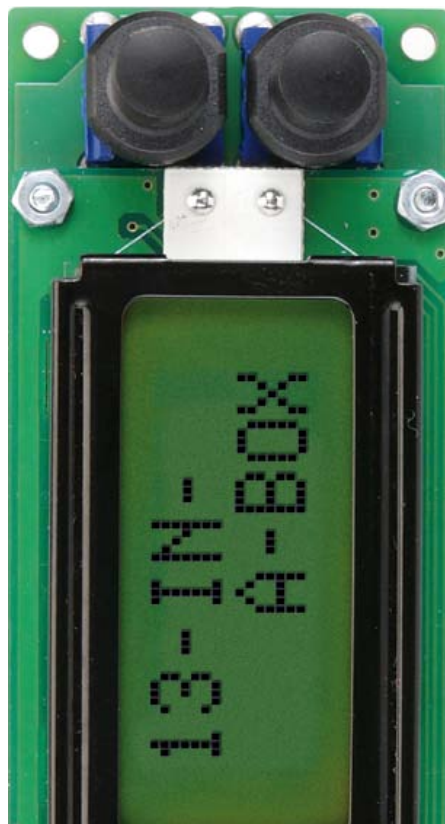
- 8 mesure électrique de proximité d'orage**
mille tonnerres de ... près
- 12 salon CARTES 2012**
430 exposants de 45 pays ;
une belle occasion de rencontrer
des développeurs passionnés
- 16 TI day sous le signe de l'innovation**
Elektor à l'escale parisienne
de la manifestation TI Day

● Tech the Future le futur en tête

- 116 matériel libre**
entretien avec l'un des pionniers
de l'*Open Source Hardware*

● pratique

- 28 la tension (de 5 V) nécessaire de Mr Cyclopède**
5 V à la force des mollets :
alim pour cyclistes
- 30 n'oublie pas ta brosse-à-temps !**
éducatif et divertissant :
pour les enfants qui n'aiment
pas leur brosse à dents
- 32 bougie à LED bicolore**
avec couleur réglable et détecteur
de courant d'air... et un ATtiny
- 34 limiteur de transitoires**
pour absorber le choc du courant
de démarrage
- 36 détecteur de présence capacitif**
pas de mécanique :
inusable et indérégable
- 40 horloge numérique sonore**
cloche ou coucou :
c'est le don qui fait la musique
- 44 détecteur d'humidité**
idéal pour ventiler une salle
de bains
- 45 antenne extérieure bande aviation**
idéal pour le scanner VHF
de la p. 76 de ce numéro
- 46 APU : alerte à la pile usée**
rudimentaire et salutaire
- 47 1 bouton + 1 LED = 2 fils seulement**
version améliorée
d'une astuce de câblage
- 48 CAN & BASCOM-AVR**
ouverture du compilateur BASIC
BASCOM-AVR sur le bus CAN
- 52 éclairage annulaire à LED**
cercle de lumière pour
photographie rapprochée



- 58 Gadgeteer : prototypage rapide façon Microsoft**
plateforme de prototypage rapide exploitant le .NET Micro Framework
- 68 mini-assistant de laboratoire**
multimètre logique et numérique pour tests ponctuels
- 72 testeur d'accu 1,2 V**
mesure de capacité par décharge
- 76 scanner de radionavigation aérienne**
portable et avec USB : tout le trafic dans la bande de navigation aérienne civile
- 84 HangTux**
jeu de pendu sur la carte **embarquez Linux !** d'Elektor
- 88 embarquez Linux ! (7)**
I²C, UART, RS485, extensions, mises à jour du noyau, de la chaîne d'outils et du compilateur C/C++

- 98 mettez le cap sur Arduino (4)**
surveillance d'arrosage de plantes
- 106 FPGA : et l'homme créa sa puce (2)**
250 000 portes pour une seule LED
- labs**
- 24 des nouvelles du labo virtuel**
Elektor a un nouveau cœur : Elektor dot Labs.
Comme les Shadoks, il pompe
• toujours plus de volts et d'ampères
• toujours plus haut et plus loin
- 66 E-labs : côté cuisine**
coup d'œil sous le couvercle d'une grosse marmite qui mijote au labo
- 115 tronçonner le tore**
pour confectionner une bobine sur un tore avec entrefer

● magazine

- 6 de nous à vous**
le réseau international d'Elektor
- 120 Nagra IV : LE magnétophone à bande**
un plaisir pour le reporter, vers 1968
- 124 défi énergie verte RL78 les gagnants**
méthodes de conception pour une meilleure efficacité énergétique
- 127 hexadoku**
casse-tête pour électronicien
- 130 bientôt dans Elektor**

● communauté

- 20 de vous à nous**
vos témoignages sur le rayonnement d'Elektor dans le monde entier

36^{ème} année, n°415/416

janvier-février 2013

ISSN 0181-7450

Dépôt légal : décembre 2013

CPPAP 1113 U 83713

ELEKTOR / PUBLITRONIC SARL

c/o Regus Roissy CDG

1, rue de la Haye

BP 12910

FR - 95731 Roissy CDG Cedex

France

Tél. : (+33) 01.49.19.26.19

du mardi au 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'œuvre dans laquelle elles sont incorporées (Loi du 11 mars 1957 -art. 40 et 41 et Code Pénal art. 425).

Certains circuits, dispositifs, composants, etc. décrits dans cette revue peuvent bénéficier de droits propres aux brevets; la Société editrice 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 editrice. La Société editrice 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 editrice 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 editrice 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.

Divers circuits d'hiver – an 01

Un numéro double d'Elektor en hiver, ça déroute, n'est-ce pas ? Comme de se retrouver en maillot de bain sur un glacier ou en skis sur la plage...

En tout cas, on s'est bien amusés à le fabriquer, même si, hélas, au bout il n'y a pas, comme en été, la carotte des vacances. Si nous l'avons fait, c'est surtout pour répondre à votre demande, comme à celle de nos

nombreux lecteurs inconditionnels de circuits petits, abordables, nombreux, variés, qui stimulent la pratique de l'électronique. C'est pourquoi la première moitié de ce numéro rappelle le style du traditionnel hors-gabarit de juillet-août. Les *petits vélos* sont réunis à l'avant, tandis que, pour équilibrer les 132 pages, on retrouve, à l'arrière, quelques *gros cubes*, c'est-à-dire des articles plus consistants et plus ambitieux.

L'électronique d'aujourd'hui, et surtout celle de demain telle que nous la concevons, sont incarnées par le trio d'articles *embarquez Linux, Arduino et FPGA*. D'autre part, la montée en puissance de *dot labs*, notre laboratoire virtuel sur **www.elektor-projects.com**, où se retrouvent les membres de la communauté d'Elektor, témoigne aussi de cet effort incessant d'adaptation de notre passion au monde, à la fois tel qu'il est et tel que nous voudrions qu'il soit. Les traditions ne sont jamais aussi vivantes que lorsqu'elles évoluent : innovons et bousculons ensemble !

Votre satisfaction sera notre récompense.

Denis Meyer



Notre équipe

Rédacteur en chef : Denis Meyer (redaction@elektor.fr)

Directeur éditorial : Wisse Hettinga

Rédaction internationale : Harry Baggen, Jan Buiting, Thijs Beckers, Eduardo Corral, Jens Nickel, Clemens Valens

Laboratoire : Thijs Beckers, Ton Giesberts, Luc Lemmens, Raymond Vermeulen, Jan Visser

Ont coopéré à ce numéro : Jean-Paul Brodier, Claire Dodé, 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
+31 46 4389417
f.tewalvaart@elektor.de



Netherlands

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



Spain

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



Italy

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



Sweden

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



Brazil

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



Portugal

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



India

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



Russia

Nataliya Melnikova
+7 (965) 395 33 36
Elektor.Russia@gmail.com



Turkey

Zeynep Köksal
+90 532 277 48 26
zkoks@bati.com.tr



South Africa

Johan Dijk
+27 78 2330 694
j.dijk@elektor.com



China

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

Notre réseau



VOICE COIL

CIRCUIT CELLAR
THE WORLD'S SOURCE FOR ORIENTED ELECTRONICS ENGINEERING INFORMATION



audio X PRESS



vous connecte à



Nos annonceurs



Beta Layout

www.pcb-pool.com11



Pico

www.usbmso.com/PS209131



DesignSpark

www.designspark.com 2



Reichelt

www.reichelt.fr27



Eurocircuits

www.elektorpcbservice.com57



Jackaltac

www.jackaltac.com71



National Instruments

www.nidays.fr132

Pour placer votre annonce dans le prochain numéro d'Elektor

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

Vos correspondants

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

mesure électrique de proximité d'orage par la technique de la bande RF étroite

Ruggero Leoncavallo,
ingénieur systèmes chez
ams AG

Les Américains conservent sur la météo et les accidents associés des rapports détaillés qui fournissent des archives à long terme au sujet des risques que la foudre fait encourir aux hommes. Depuis 1940, elle a fait 30 % de victimes de plus que les ouragans (d'après les données de la NOAA, le pendant américain de notre météo gauloise). La précision des prévisions météo permet de nos jours aux populations exposées de prendre des précautions en cas de risque d'orage violent, de sorte que le nombre de décès est tombé à moins de 50 par an en moyenne aux États-Unis. Cependant, chaque mort est un mort de trop. Pour les habitants de régions sujettes aux orages violents, les blessures et les dégâts causés par la foudre (détails sur www.struckbylightning.org) sont considérables. Comme si cela ne suffisait pas, des scientifiques chinois affirment qu'il est probable que le réchauffement climatique intensifiera l'occurrence des situations météo extrêmes, les quelques intempéries graves survenues ces dernières années ne seraient qu'un début (*China Meteorological Administration, July 30, 2007*).

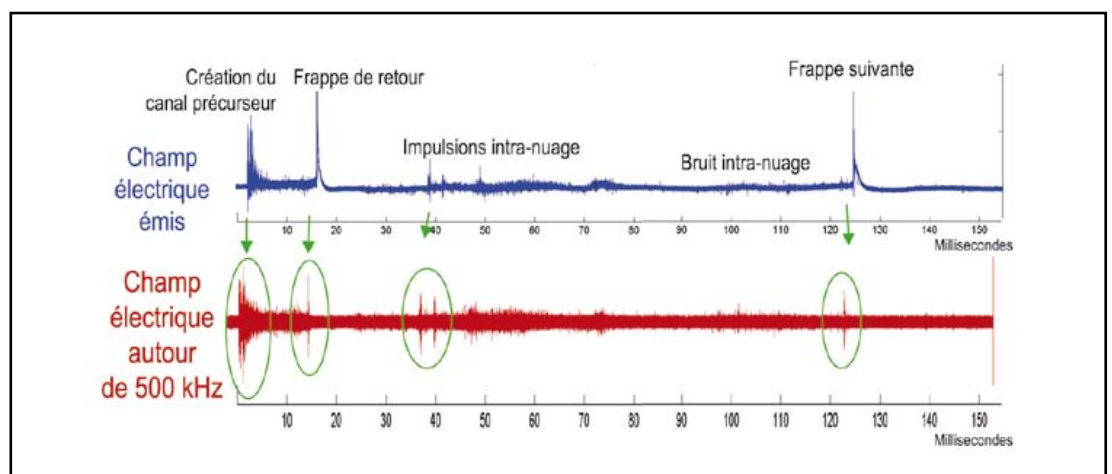
Aucun de nos sens ne nous permet de percevoir l'arrivée d'un orage. Pourtant, les morts et les dégâts attribuables à la foudre sont pour la plupart évitables, si l'arrivée d'un orage est annoncée

suffisamment tôt. Selon la règle des « 30-30 », si le temps entre éclair et tonnerre est de moins de 30 s, il faut se mettre à l'abri pour au moins 30 mn. Quand vous entendez le tonnerre, vous êtes probablement déjà en danger. Un délai de 30 s entre éclair et tonnerre signifie que l'on se trouve à une distance d'environ 10 km du point d'impact (la vitesse du son dans l'air étant de 300 m/s) ; c'est la distance maximale à laquelle les humains peuvent entendre le tonnerre dans un environnement calme sans obstacle physique gênant la propagation du son. S'il y a des obstacles et/ou un fort bruit ambiant (trafic, foule) cette distance peut être réduite à seulement quelques kilomètres. Pire encore, il est courant que la foudre ne frappe pas verticalement, mais diagonalement, sur des distances horizontales pouvant atteindre 10 km ; c'est la plus grosse limitation de la règle des « 30-30 ». Étant donné qu'on ne peut entendre le tonnerre qu'à une distance maximale de 10 km, il est clair que se fier à nos seuls sens augmente considérablement le risque encouru par les biens autant que les personnes.

Foudre et impulsion électromagnétique (IEM)

Dès le 19^e siècle, Alexander Stepanovic Popov a remarqué qu'il était possible de détecter la foudre

Figure 1.
Comparaison des signaux
émis avec ceux en bande
étroite pour un événement
de foudre.



à l'aide d'un simple récepteur radio AM ; il s'agissait alors du premier système électrique capable de prédire les orages. Concrètement, la foudre émet de l'énergie sous forme électromagnétique, depuis les très basses fréquences et jusqu'aux rayons X. Le spectre des ondes rayonnées possède une distribution de la forme $1/f$: les émissions sont plus fortes aux basses fréquences (quelques kHz), et s'affaiblissent à mesure que leur fréquence augmente. À l'aide d'un système simple comprenant un amplificateur, un adaptateur de bande et un filtre passe-bas, Popov était en mesure d'entendre le signal produit par la foudre.

Une technique similaire est toujours utilisée de nos jours dans les détecteurs de foudre pour le grand public. Les hautes instances de la météorologie ne reconnaissent ni la fiabilité ni la valeur de ces appareils portables. Ils peuvent néanmoins, quand les conditions sont favorables, détecter la foudre dans une petite zone. Ces appareils rudimentaires ne sont pas la panacée ; ils ne sont en effet ni capables d'estimer la distance du front de l'orage, ni de les discriminer des sources d'interférences telles que les fours à micro-ondes, les ballasts pour tubes fluorescents, les moteurs (de voiture ou autre) et les flashes d'appareils photo. De plus, ces systèmes sont basés sur des composants discrets et n'ont donc pas une consommation électrique optimale, ce qui souvent limite à quelques semaines la durée de vie des piles.

Ce qu'il faut pour une prévention fiable et utile, c'est un appareil capable d'estimer précisément la distance des orages dans un rayon de 30 km ou plus et de distinguer de manière fiable la foudre des autres sources d'IEM.

Utilisation de systèmes à bande étroite

Il existe deux types de foudre : nuage-à-sol et intra-nuage(s). Du point de vue électromagnétique, les énormes courants qui circulent pendant un orage produisent des signaux à large bande sur une gamme étendue de fréquences. La surveillance d'une telle bande est quasi-impossible avec un appareil mobile grand public. Par chance, depuis les expériences de Popov nous savons qu'un système à bande étroite peut capturer les signaux de la foudre. Mais quelle précision peut-on attendre d'un signal à bande étroite ?

Dans les faits, la foudre est une combinaison complexe de plusieurs événements : création d'un canal précurseur, impact de retour, activité intra-nuage et impact(s) suivant(s). La littérature scientifique (Le Vine, D.M., 'Review of Measurements of the RF Spectrum of Radiation from Lightning', 1987) nous apprend qu'il est possible de détecter la foudre avec un système à bande étroite, même si certains détails de la forme du signal sont perdus. Le signal capté à l'aide d'un récepteur à bande étroite calé sur 500 kHz ressemble de près au champ électrique émis (fig. 1).

au cul des vaches

Vu dans les Dernières Nouvelles d'Alsace du 16/12/2012



Au moment de mettre sous presse, nous recevions cette coupure de presse d'un fidèle lecteur, M. Claude Maechler, qui se souvient du **système de détection de vélage pour vaches** évoqué dans Elektor :

n°403, janv. 2012 | n°404, fév. 2012 | n°406 avr. 2012

Figure 2.
Structure interne du
détecteur d'orages AS3935.

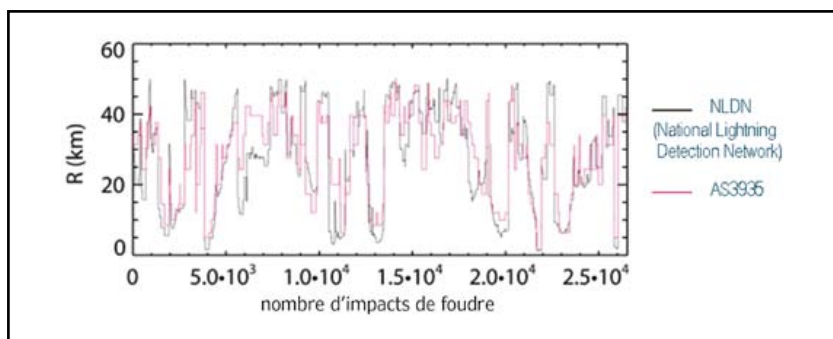
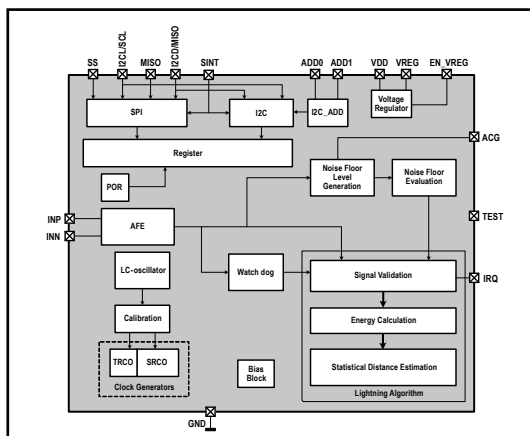


Figure 3. Comparaison
des données fournies
par l'AS3935 avec celles
du NLDN. Cela montre
que la sortie de l'AS3935
correspond assez fidèlement
aux données issues du
système sophistiqué à base
de radars du NLDN.

Le papier de Le Vine montre également que le pic d'émission de la foudre se situe autour de 5 kHz et que l'intensité décroît avec l'inverse de la fréquence. Par conséquent, si l'on règle trop haut la fréquence d'un récepteur à bande étroite, la faiblesse du signal reçu ne permet plus de le distinguer d'autres sources parasites d'IEM. Le signal est bien plus fort aux faibles fréquences ; cependant, aux très faibles fréquences, l'antenne devient trop encombrante pour être logée dans un appareil mobile. Un compromis s'impose entre force du signal et taille de l'appareil, les fréquences idéales se situant entre quelques centaines de kHz et quelques MHz.

Estimation fiable de la distance par reconnaissance logique de motif

Les vrais défis pour produire un détecteur fiable et utile sont :

- de pouvoir rejeter les signaux des autres émetteurs
- d'estimer précisément la distance de l'orage

Dans ce but, ams (précédemment *austriamicrosystems*) propose le circuit de détection d'orage

AS3935a, efficace pour la foudre intra-nuage(s) ou nuage-à-sol, dont les algorithmes analysent les signaux reçus et comparent leur forme à la forme typique d'une onde représentant un impact de foudre. De gros efforts de peaufinage de l'algorithme ont été faits afin d'obtenir un excellent équilibre entre le rejet des interférences liées aux émetteurs parasites et la reconnaissance de signaux émis par la foudre.

Un algorithme câblé dédié, implanté dans l'AS3935 est également capable, à partir de l'analyse de l'énergie du signal détecté par les étages RF de la puce, d'estimer précisément la distance. L'AS3935, dont la structure interne est donnée par la figure 2, surveille comme le fait le système de Popov, les bandes de basse fréquence (500 kHz – 2 MHz) afin de détecter la forte signature en $1/f$, caractéristique de la foudre. Le système comprend un étage d'entrée analogique (AFE), qui amplifie le signal capté par l'antenne et le ramène en bande de base, tout en le filtrant afin de préparer les calculs numériques qui suivent. L'algorithme de détection de foudre comprend trois étages : validation du signal, calcul de l'énergie et estimation statistique de la distance.

Le premier bloc vérifie la forme du signal et rejette les signaux d'origine humaine. Un signal de foudre typique (fig. 1) monte très rapidement et décroît doucement après le pic. L'algorithme flexible de l'AS3935 permet à l'utilisateur d'optimiser le compromis fait entre efficacité de la détection et réjection des perturbations.

Si le signal reçu est reconnu comme provenant de la foudre, le deuxième bloc en calcule l'énergie. Le résultat est ensuite analysé dans le dernier bloc qui évalue la distance du front à partir des données déjà collectées durant cet orage. L'estimation de distance est fournie en kilomètres avec une résolution entre 1 et 4 km. (L'estimation devient plus précise à mesure que l'orage se rapproche de l'instrument.)

Les performances de l'AS3935 ont été testées par le *Florida Institute of Technology* de Melbourne, EUA et comparées aux données officielles fournies par le *National Lightning Detection Network* (NLDN) américain, considéré comme la meilleure source en matière de surveillance de la foudre (fig. 3).

(120541 – version française : Kévin PETIT)



L'ORIGINAL DEPUIS 1994
PCB-Pool
Beta LAYOUT

Prototypes PCB et Petites Séries



Pochoir gratuit
avec chaque commande
"Prototype"



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

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

www.pcb-pool.com



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*

LED Blinker 6 LEDs Kit CMS



€ 6,00*

Le grand Beta-Reflow-Kit



€ 129,00*

Trousse à outils « Extended »



€ 149,00*

* TVA comprise, frais d'expédition en sus

www.beta-eSTORE.com

Beta
LAYOUT
create: electronics

Patrick GUEULLE
(France)

salon CARTES 2012



Pour sa 27^e édition qui s'est tenue du 6 au 8 novembre 2012 à Paris-Nord Villepinte, le salon CARTES accueillait cette année encore près de 430 exposants de 45 pays ; un bel achalandage malgré la montée en puissance des éditions délocalisées, CARTES Asia et CARTES America. Une occasion unique aussi de rencontrer des développeurs passionnés dont beaucoup se souviennent, au passage, d'avoir fait leurs premiers pas dans le métier... en lisant Elektor !

Apprivoiser le *sans contact*

Une telle complicité, si elle ne nous rajeunit pas, délie les langues et permet d'aborder franchement les sujets dérangeants sans les laborieuses dénégations officielles. Inutile de tourner autour du pot : une carte bancaire sans contact peut être lue à travers la poche de son porteur, révélant

Pas question, par contre, de lire (en la télé-alimentant) une telle puce passive à des distances de plusieurs mètres, mais intercepter son dialogue avec un lecteur légitime serait un tout autre cas de figure. Avec une puissance HF de 3 W et une simple antenne télescopique, des signaux Morse traversent en effet facilement l'Atlantique à la fréquence de 13,56 MHz, moyennant des algorithmes de réception SDR performants. Alors...

Les technologies radio ont d'ailleurs bien d'autres applications dans le monde des cartes à puce : ayant repris la branche santé de Xiring, **Ingenico healthcare e-ID** [2] a ainsi imaginé le *Witeo*, un lecteur scindé en deux : une base PC/SC classiquement reliée par USB à l'ordinateur hôte, et un porte-badges autonome que l'on garde sur soi. Dès que l'on s'éloigne de la base, la liaison sans fil se coupe, la carte étant alors considérée comme retirée du lecteur. Imaginée pour les cartes de professionnels de santé (CPS) en milieu hospitalier, cette idée concerne potentiellement n'importe quelle carte à puce ISO 7816 T=0 ou T=1, et donc de multiples domaines.

Le plus exceptionnel était la carte micro-SIM réalisée par le chinois **Sunward Telecom** [3], justement récompensée par un SÉSAME de l'innovation : en ajoutant une minuscule antenne RFID de 5 x 10 mm à un jeu de puces **INSIDE Secure** (frontal radio *PicoPulse* et élément sécurisé *VaultSecure*), c'est la totalité de la partie NFC d'un téléphone portable qui a été ainsi intégrée à la carte ! Un opérateur de téléphonie peut donc doter, à moindre coût, les téléphones

jusqu'à son
 identité ! En
 l'absence d'un
 étui blindé,
 n'importe quel
 lecteur RFID
 du commerce
 permet de faire
 très facilement
 l'essai, avec
 p. ex. l'étonnant
 logiciel libre
CardPeek
 [1], dont
 l'auteur reste
 prudemment
 anonyme.

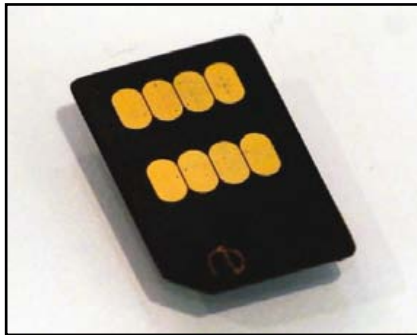
[illegible]

non-NFC de ses clients de modes d'émulation de carte et *Peer to Peer*, en attendant la fonction de lecteur, encore en développement.

Grands classiques revisités

Toujours chez **INSIDE Secure** [4], nous avons remarqué le VaultIC 420i, une EEPROM série pas comme les autres, mais interchangeable broche pour broche avec les versions I2C standard. Les EEPROM sécurisées, cousines des cartes à puce synchrones, ne sont pas précisément une nouveauté, mais là deux mémoires de capacité identique coexistent dans un boîtier : l'une ordinaire, et l'autre dotée de fonctions de cryptographie et de stockage sécurisé de données (certification FIPS-140-2 niveau 3). Une fois le basculement vers le bloc sécurisé effectué (par une commande spéciale envoyée sur le bus), une procédure d'authentification forte est obligatoire pour repasser au mode de base.

En digne héritier d'Atmel, INSIDE reste fort discret sur la mise en œuvre pratique de ces composants, et ne livre pas ses kits de développement à n'importe qui. Sans doute faudra-t-il attendre leur apparition dans des produits grand public pour examiner de plus près leur fonctionnement, et contourner ainsi cette *sécurité par l'obscurité* qui excite plus la curiosité qu'elle ne la dissuade... Encore dans le domaine des mémoires non volatiles, d'autres avancées technologiques se situent au plus profond du silicium : **Crocus Technology** [5] dévoile ainsi ses microprocesseurs pour cartes SIM et éléments sécurisés CT32MLU, dotés d'une technologie de mémoire dite *Magnetic Logic Unit*. Une version submicronique des mémoires à tores de ferrite, si l'on veut, mais basée sur le procédé CMOS. Dix fois plus rapide qu'une mémoire Flash pour un encombrement moindre et une sécurité accrue, c'est une aubaine pour les personnalisations et mises à jour par OTA (*Over The Air*), par exemple lors du téléchargement de nouvelles applications



(notamment de paiement mobile). Avec une capacité de 1,2 Mo dans un format nano-SIM, il y a de la place !

Clefs USB à la carte

Ces derniers mois, le concept **Plug-up** que nous présentions l'an dernier a fait du chemin (trophée *Seinari* de l'innovation 2012). Désormais supportée par des outils de développement libres (*public SDK*), cette clef USB déguisée en carte à puce (et aussi l'inverse !) s'ouvre enfin à tous les développeurs intéressés, sans mise de fonds initiale. Un peu comme la *BasicCard*, donc, mais la similitude s'arrête là, car

la parenté est plus évidente avec la *JavaCard*. Des spécifications détaillées sont d'ores et déjà en ligne [6] et les premiers échantillons enfin fonctionnels étaient distribués sur le salon. Technologie originale, à essayer impérativement : elle apporte l'équivalent (et plus encore) de l'arsenal sécuritaire de *Global Platform* à une large variété d'ordinateurs dépourvus de lecteur PC/SC (une situation encore bien trop courante...), sans installation de pilotes spécifiques. Le CCID des clefs USB, en somme.

Des API bientôt disponibles permettent d'y accéder depuis différents langages de programmation (et notamment Javascript, par exemple dans une page web), sous Windows (32 et 64 bits), Linux, ou Mac OS.

Les échanges d'*Application Protocol Data Units* (APDU) ne se font pas par l'équivalent d'un lecteur PC/SC, mais sur une interface *Human Interface Device* (HID) générique, comparable à celle d'un clavier ou d'une souris, exposée par le biais d'un *périphérique USB composite*. Un système d'exploitation particulier est donc implanté dans un microcontrôleur de carte à puce à connectivité USB 2 native : un ST23 (8 bits) en attendant un ST33 (32 bits), et le tour est joué !

Chez **Neowave** (www.neowave.fr), la famille *Weneo* de clefs USB multifonctions poursuit avec succès son parcours de certification sécuritaire, et introduit cette année le cryptage des partitions

de mémoire de masse *utilisateur*, avec mot de passe requis pour l'accès. Cela rappelle des produits comme le *SmartGuardian* de **Gemalto**, mais avec en plus des fonctions intégrées de *tag* sans contact et de lecteur PC/SC, plébiscitées dans les domaines du paiement (*Moneo*) et du transport en commun (*Calypso*) car elles facilitent le rechargement à la maison.

Terminal CB à domicile !

PayKnox, concept de paiement en ligne révolutionnaire développé par les équipes de TV-card, déjà à l'origine du système «mo-net» de rechargement à domicile du porte-monnaie Moneo, a été repris et amélioré par **Gemalto** sous le nom *PINPaid* [7]. Il s'agit carrément d'un terminal personnel de paiement par carte bancaire, avec clavier et écran, à la portée de toutes les bourses. Si le commerçant en ligne est affilié à ce programme (ce qui a évidemment un coût...), le client tape son code PIN chez lui dans les mêmes conditions de sécurité qu'en point de vente, et imprime un ticket en bonne et due forme après autorisation systématique de la transaction par la banque.



Ce moyen de paiement à distance hautement sécurisé mériterait d'être généralisé, y compris (rêvons un peu) pour ces transactions internationales qui échouent encore bien trop souvent, sans raison convaincante avancée ni par les banques ni par les organismes émetteurs de cartes. Cela éviterait peut-être aussi les débits récurrents, et fréquemment abusifs, de cartes utilisées en toute confiance pour un règlement en principe isolé... Mais la route sera longue !

Bricoler les tickets de bus ?

Les tickets sans contact en papier (rapidement biodégradables) sont une spécialité d'**ASK** [8]. Complémentaires des cartes à puce *Calypso* pour l'usage occasionnel, ils sont jetables et par conséquent récupérables en grandes quantités ! Si l'on en croit l'impressionnante collection exposée sur le stand de l'industriel, ils s'imposent progressivement et méritent donc tout l'intérêt de l'expérimentateur curieux.

Jusqu'alors assez réfractaires aux lecteurs sans contact courants (bien plus à l'aise avec les puces *Mifare UltraLight* du métro de Moscou par. ex.),

Quelques lignes de Basic...



L'écosystème *BasicCard* a cela de gratifiant pour l'amateur averti que quelques lignes de Basic lui suffiront pour s'introduire subrepticement (mais en tout bien tout honneur !) dans

les plus grandes applications exposées sur un salon aussi professionnel que CARTES...

Un kit logiciel gratuit [11], un lecteur CardMan 5321 [12] équipé de la dernière (c'est important) version de son pilote et de son API pour cartes synchrones, et en avant pour la lecture des tickets de transport en commun à SRT512 !

```
#Include CARDUTIL.DEF
#Include COMMERR.DEF
Declare Command &HFF &HCA UID(S$)
Declare Command &HFF &HA0 GEN(S$)
ComPort=102
REM Pour Omnikey 5321 avec driver 1.2.9.2 / XP (ou sup)
```

```
CLS:PRINT"LIA (c)2012 Patrick GUEULLE":PRINT
Call WaitForCard:ResetCard(S$):Call CheckSW1SW2
Print:Print
Call UID(P1P2=&H0000,Lc=0,S$,Le=8):Call CheckSW1SW2
Print «UID : <;
For F=1 TO Len(S$)
C$=MID$(S$,F,1):C=ASC(C$):C$=HEX$(C)
IF LEN(C$)=1 then C$=>>0>>+C$
Print C$;
Next F:Print:Print
S$=Chr$(&H01)+Chr$(&H00)+Chr$(&H01)
Call GEN(P1P2=&H0007,Lc=3,S$,Le=0):Call CheckSW1SW2
For B=0 To 15
S$=Chr$(&H01)+Chr$(&H00)+Chr$(&H0B)+Chr$(&H00)+Chr$(&H0
0)+Chr$(&H64)+Chr$(&H08)+Chr$(B)
Call GEN(P1P2=&H0005,Lc=8,S$,Le=0):Call CheckSW1SW2
For F=3 TO Len(S$)
C$=MID$(S$,F,1):C=ASC(C$):C$=HEX$(C)
IF LEN(C$)=1 then C$=>>0>>+C$
Print C$;
Next F:Print
Next B
S$=Chr$(&H01)+Chr$(&H00)+Chr$(&H02)
```

certaines modèles équipés d'un composant SRT512 (**STMicroelectronics**) se laissent finalement manipuler (voir encadré) avec le kit **BasicCard** sans contact de **ZeitControl** [9], basé sur un lecteur *Omnikey CardMan 5321* de **HID** [10]. Avec 512 bits d'EEPROM dont 64 sont occupés par des compteurs à écriture irréversible, nous voyons là au moins le même potentiel de bricolage qu'autrefois avec les Télécartes usagées : un identifiant unique (UID) inaltérable et un nombre appréciable de bits pouvant être lus, écrits, ou même effacés.

Il est assez passionnant d'étudier les spécifications *Intertic* (nullement confidentielles) pour comprendre comment l'utilisation de certificats cryptographiques générés par des modules de sécurité (SAM) est censée éviter les formes courantes de fraude malgré d'aussi larges possibilités d'altération du contenu des tickets.

(120590)

Liens

- [1] <http://code.google.com/p/cardpeek/>
- [2] www.ingenico.com
- [3] www.sunwardtel.com
- [4] www.insidesecond.com
- [5] www.crocus-technology.com
- [6] www.daplug.io
- [7] www.pinpaid.net
- [8] www.ask-rfid.com
- [9] www.zeitcontrol.de
- [10] www.hidglobal.com
- [11] www.basiccard.com
- [12] www.hitechtools.com

Votre Ticket

CARTE BANCAIRE

Paiement sur Internet par PINPaid
http://www.pinpaid.net/fr

A0000000421010
CB

dimanche 11 novembre 2012 19:27:14

PINPaid
Installation PINPaid
sur Internet
44252553100020

XXXXXXXXXXXX5997
3E152492CB812D66

001 001 000917

C

MONTANT
0,00 EUR

DEBIT

TICKET CLIENT A CONSERVER

Référence du panier : CHECK PAYMENT 11/11/2012 19:27:14
Référence du paiement : c79fa409-7c38-4 0584f

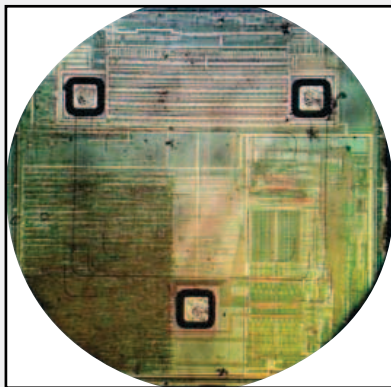
```
Call GEN(P1P2=&H0007,Lc=3,S$,Le=0):Call CheckSW1SW2
Print:Print
Call WaitForNoCard
```

```
00000004
2EAF0000
A9AB54E5
```

Voici un exemple de ce que cela peut donner, dans le cas bien réel d'un ticket n'ayant été présenté qu'une seule fois devant un valideur (en cas de correspondance, c'est un peu plus complexe !).

UID : 7460C350603002D0

```
100000A0
20120925
9995010C
00000000
00000000
00000004
FDFFFFFF
00000000
00000000
36070000
18C855C7
3500B622
73150F27
```



La puce SRT512 vue au microscope !

On identifiera facilement le certificat cryptographique en dernière position (le bloc de 32 bits n° 15), et les deux compteurs dans les blocs 5 et 6. L'un compte apparemment les validations du ticket, et l'autre (pré-grillé à la valeur 4) ses éventuels rechargements.

Juste trois lignes de code à changer pour procéder, cette fois, à l'effacement général (compteurs exceptés) de cette puce *vierge* à 1, pourvu que les bits de verrouillage (situés à l'adresse de bloc 255) n'aient pas été positionnés auparavant. Cela varie d'un opérateur de transport à l'autre...

```
S$=Chr$(&H01)+Chr$(&H00)+Chr$(&H0B)+Chr$(&H00)+Chr$(&H0
0)+Chr$(&H64)+Chr$(&H09)+Chr$(B) _
+Chr$(&HFF)+Chr$(&HFF)+Chr$(&HFF)+Chr$(&HFF)
Call GEN(P1P2=&H0005,Lc=&H0C,S$,Le=0)
```

Pour aller plus loin dans cette voie, il suffirait de remplacer les quatre valeurs &HFF par les octets *personnels* que l'on souhaite écrire, en faisant varier le numéro de bloc (B) entre 0 et 4, ou entre 7 et 15. De nouvelles aventures en perspective !

TI day sous le signe de l'innovation

Elektor à l'escale parisienne de la manifestation TI Days

Mariline
Thiebaut-Brodier

Ces journées de présentation organisées par *Texas Instruments* sont axées sur l'innovation. Les différents départements R&D de TI viennent présenter le fruit de leurs études ainsi que l'objet des recherches encore en cours. La matinée était consacrée à des présentations générales, six conférences techniques étaient proposées l'après-midi en parallèle ; il a fallu en choisir deux.

Un monde numérique indissociable de l'analogique

L'histoire de *Texas Instruments* a été liée aux composants analogiques, depuis le premier circuit intégré régulateur de tension analogique

dans les années 60, au premier pilote pour LED blanche dans les années 2000, en passant par le premier convertisseur analogique/numérique à 16 bits dans les années 1970.

Dans notre monde dit numérique, l'analogique n'a pas dit son dernier mot. Les systèmes numériques manipulent des données analogiques (signaux envoyés par les capteurs p. ex.). La part de TI sur le marché de l'analogique est de 18%. Avec 54% de son chiffre d'affaires, l'analogique reste pour TI un moteur d'innovation : applications plus intelligentes, plus fiables, plus vertes (diminution de la consommation), plus petites (réduction de la taille des boîtiers, augmentation du nombre de fonctions sur une même puce), plus confortables (commande sans fil, recharge sans fil).

Communication sans fil...

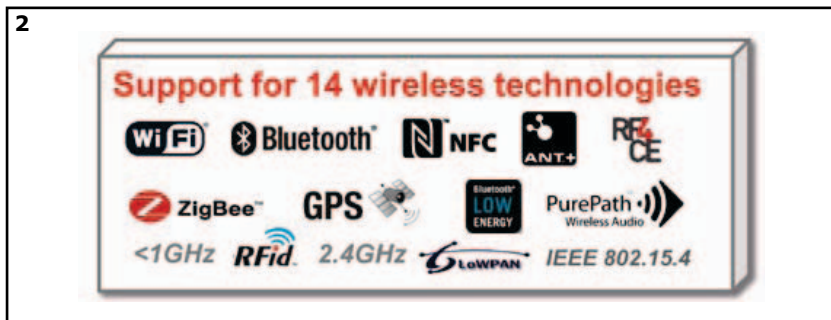
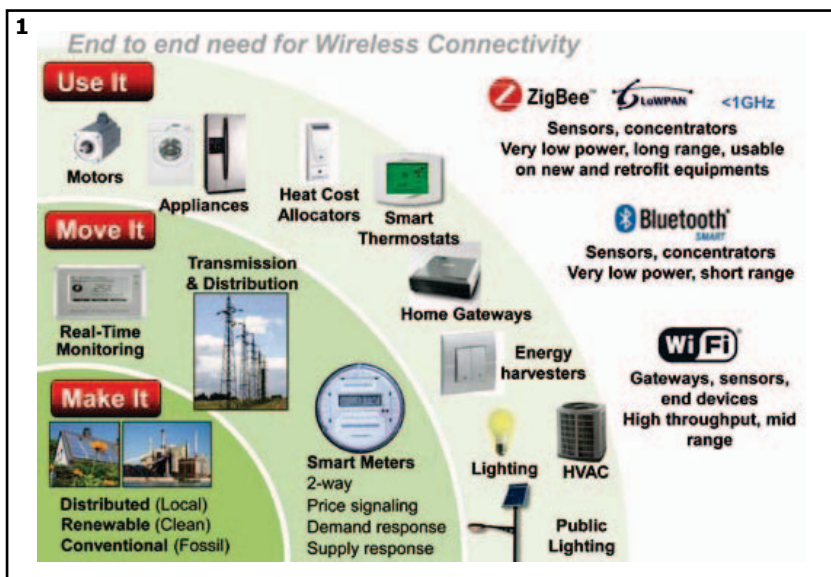
Pour répondre à l'explosion de la demande de communication bi-directionnelle sans fil, TI oriente ses recherches vers trois axes :

1. Contrôler tout et n'importe quoi à distance

Il peut s'agir p. ex. d'un capteur sur une roue de voiture qui envoie des informations (pneu usé...) pour plus de sécurité. Dans ce cas, le prix du capteur sera un critère de choix essentiel. Pour la connexion de la voiture au monde extérieur via *Bluetooth*, *Wi-Fi* etc., le concepteur portera son choix sur un composant qui converse dans plusieurs langues (protocoles) à des débits élevés.

2. Demande d'énergie « intelligente »

Dans la maison, l'amélioration de l'efficacité énergétique du matériel et l'évolution du comportement permettront de diminuer la consommation. Les fabricants de matériel électrique proposent



des dispositifs pour contrôler sa consommation en temps réel et l'optimiser, il s'agit de déplacer sa consommation à des moments où l'électricité est la moins chère. C'est le cas pour les abonnés au tarif heures creuses/heures pleines. Il s'agit de renforcer ce système pour éviter les pics de consommation et réguler la production, entre autres décentralisée (éolien, photovoltaïque...), en fonction de la demande. Dans cet objectif, ERDF déploie son compteur dit intelligent, *Linky*. Il donne des informations à l'utilisateur sur sa consommation, mais il en envoie également à ERDF. En fait le réseau électrique se transforme et devient « intelligent », on parle de *smart grid* (fig. 1). Toutefois, l'utilisateur ne dispose pas encore de systèmes faciles à installer, simples à mettre en œuvre, modifiables et personnalisables pour surveiller sa consommation (et sa facture) et agir sur ses appareils domestiques. Chaque fabricant de matériel électrique développe son propre réseau de surveillance (*WISER* pour *Schneider Electric* p. ex.) avec le protocole de son choix, d'où un problème d'interopérabilité entre ces systèmes.

3. Développement de l'internet des objets (Internet of Things)

À l'avenir, nombre d'objets seront dotés d'une adresse IP pour être interconnectés. Pour qu'une machine à laver dispose d'une adresse IP et envoie des informations, il faudra un composant qui permette d'ajouter cette fonction à un microcontrôleur à 8 bits. Pour répondre à ces différents cas, TI propose des composants :

- faciles à intégrer dans un montage existant pour ajouter une fonction de communication sans fil (i Wi-Fi/ CC3000, GPS/CC4000, 6LoWPAN/CC1180) ;
- compatibles avec jusqu'à 14 protocoles différents (fig. 2) pour la communication sans fil (famille WiLink 8.0).

Maison connectée

Pour établir une liaison sans fil, plusieurs protocoles sont disponibles. La grande question des concepteurs de domotique est : quel protocole de communication va émerger ?

À l'avenir, deux types d'entreprises voudront accéder au réseau d'une maison connectée :

- les fournisseurs de services qui disposent déjà de leur propre réseau (téléphonie/internet, électricité, gaz, eau) ;

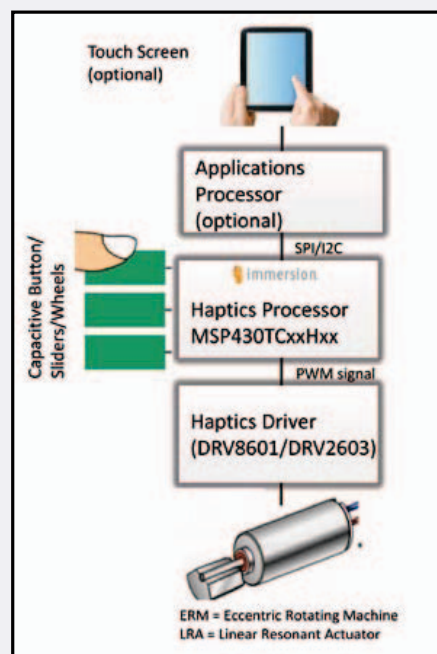
Surface capacitive et vibrante

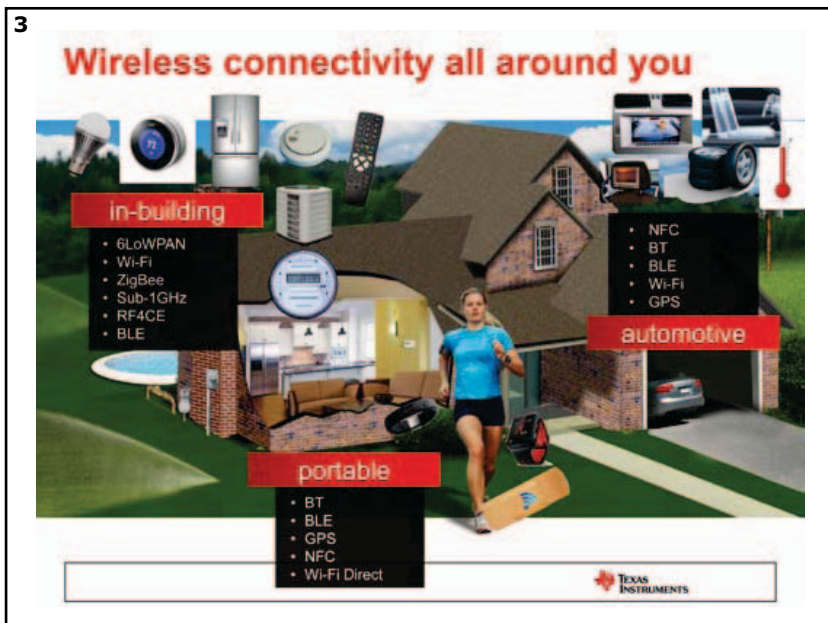
Les conférences de TI Day étaient complétées par des démonstrations du matériel TI ainsi que des stands de revendeurs et fabricants de matériel électronique. Les visiteurs pouvaient tester le kit d'évaluation du pilote DRV2603 dont le principe est le suivant : lorsqu'on appuie sur l'une des touches capacitatives, cela déclenche des vibrations.

La variation de capacité des touches sensibles est due à un élément conducteur : doigt, oreille, matériau organique ou inorganique (métaux, liquides). Le processeur MSP430 sur la carte traite le signal envoyé par les touches et commande la mise en mouvement de l'un des actionneurs mécaniques qui produit des vibrations. Il y a plusieurs types d'actionneurs :

- ERM pour *Eccentric Rotating Mass* : masse excentrée mise en rotation par un petit moteur à courant continu ;
- LRA pour *Linear Resonant Actuator* : bobine reliée à un ressort et placée dans un champ magnétique, parcourue par un courant (principe de la force de Lorentz) ;
- piézo : lame déformée lorsqu'elle est soumise à une tension.

Sur la photo ci-dessous, la masse de l'actionneur de type ERM est en mouvement. *Texas Instruments* propose des pilotes pour ces différents actionneurs (DRV8662, DRV2665/2667 pour piézo ; DRV8601 & DRV2602/2603/2604/2605 pour ERM/LRA).





- et les fournisseurs qui ne disposent pas de réseau, mais fournissent des services (société reliée à votre centrale d'alarme ou votre système de surveillance médicale).

La plupart des protocoles utilisés dans une maison (commande de volets/portes, chauffage, éclairage...) sont de type propriétaire (KNX, *io-homecontrol*, X2D...) et ne peuvent pas être interconnectés. En fait, le concepteur choisit le protocole adapté à ses besoins. Pour les liaisons sans fil, la portée et la consommation priment. Plus la portée est faible, plus la consommation l'est également ; un système Wi-Fi consomme plus qu'un *BlueTooth*. En outre, il faut savoir si l'on a besoin d'un réseau maillé de type *ZigBee* ou 6LoWPAN ou bien de liaisons point à point (*BlueTooth*, NFC, RF4CE). Notez également que plus la fréquence du signal radio diminue, plus la portée mais également la taille de l'antenne (comme la longueur d'onde) augmentent.

TI propose des modules qui simplifient la réalisation de systèmes connectés. Les composants disponibles prennent place à plusieurs niveaux :

- **interface personne-machine/serveur web** : plateforme WL1271 avec modules Wi-Fi et *BlueTooth* ou processeur AM335x avec port *Ethernet* p. ex. ;

- **passerelle** : processeur CC2530ZNP avec pile *ZigBee* ou carte *Stellaris Bluetooth* p. ex. ;
- **appareil final** : le processeur CC3000 *Sim-*

pleLink par exemple permet de connecter un appareil électroménager à un réseau Wi-Fi pour qu'il envoie des informations sur son état (alerte en cas de fuite d'un lave-linge, de coupure de courant sur un frigo...).

En 2020, le nombre d'objets interconnectés devrait être de 24 milliards (**fig. 3**). Essayez d'estimer le nombre d'objets radioémetteurs ou récepteurs présents dans une maison occupée par deux enfants et leurs parents ainsi qu'un chat. Chaque émetteur compte pour un, ainsi que chaque récepteur. Ainsi, trois volets télécommandés et leur télécommande constituent 4 objets *radio*. Vous en serez rapidement à plus de 60 objets susceptibles d'être un jour connectés : commandes d'ouverture/fermeture, alarme, éclairage télécommandé, commande de chauffage, station météo, matériel informatique sans fil, audio/vidéo sans fil, jouets radiocommandés, puce RFID du chat qui commande sa chaudière, etc. On est loin du protocole radio unique ! Ce développement de réseaux d'objets connectés soulève la question de la confidentialité des données, des attaques extérieures, des effets secondaires encore inconnus sur notre santé...

Communication sans fil, ni pile

La miniaturisation des semi-conducteurs va son train, celle des batteries est plus lente. *Texas Instruments*, comme tous ses concurrents, s'efforce donc de réduire la consommation de ses circuits et à la récolte d'énergie (*energy harvesting*), un domaine de recherche connexe, en pleine expansion. Il s'agit de récupérer, et éventuellement de stocker, de l'énergie, généralement gratuite et renouvelable, pour pouvoir se passer de piles. Plus de pollution, plus de panne due à une pile vide. Tout le monde connaît ce principe par la dynamo de vélo ou la lampe de poche à manivelle, mais le développement des nanotechnologies pousse les recherches vers des sources d'énergie faibles pour alimenter des systèmes miniatures : p. ex. les mouvements du cœur pour autoalimenter un stimulateur cardiaque, ce qui évite d'opérer le patient pour changer la pile. Ce mode d'alimentation rend totalement autonomes de petits capteurs sans fil dans des endroits difficilement accessibles.

Les techniques de récupération :

- **piézoélectricité** : récupération de l'énergie produite par un matériau piézoélectrique (sol parcouru par des piétons ou des

voitures par exemple) ;

- **thermoélectricité** : courant électrique créé par une différence de température (effet Seebeck) ;
- **énergie photovoltaïque** : les rayons du soleil ne sont pas toujours disponibles, dans ce cas il faut stocker ;
- **énergie cinétique** : l'énergie cinétique d'un ballon en mouvement sert à alimenter un capteur interne (vitesse par exemple) ;
- **énergie électromagnétique** : l'énergie produite par un champ magnétique est stockée dans un gros condensateur (*super cap*).

TI s'intéresse aux capteurs sans fil avec récupération d'énergie (**fig. 4**) et propose des capteurs de température sans fil à alimentation solaire pour un réseau basé sur le μC MSP430 (EZ430-RF2500-SEH).

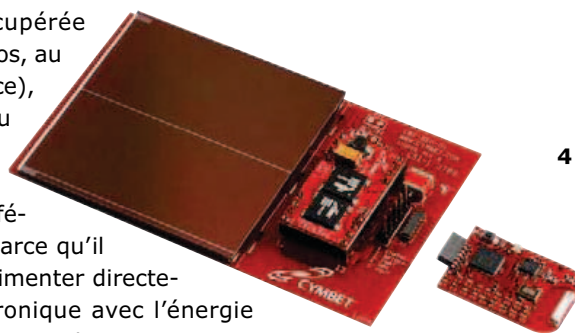
La consommation des composants alimentés

avec de l'énergie récupérée doit être faible (au repos, au démarrage et en service), comme c'est le cas du μC MSP430 qualifié d'*ultra low power*. TI propose également différents convertisseurs parce qu'il n'est pas possible d'alimenter directement un circuit électronique avec l'énergie récupérée : convertisseur-abaisseur continu-continu (TPS62120/62122), convertisseur élévateur (TPS61200).

La quantité d'énergie récupérée est par définition faible. Il reste à trouver des solutions pour la récupérer à grande échelle. La société *POWER-leap* p. ex. multiplie les expériences de trottoirs et sols piézoélectriques à travers le monde (trottoir à Toulouse, sol au Grand Prix de Monaco...).

120590

4



Ce livre en trois chapitres offre une information accessible et digeste : constitution, fonctionnement, caractéristiques, domaines d'utilisation, pour proposer aussi des réalisations électroniques simples et concrètes.

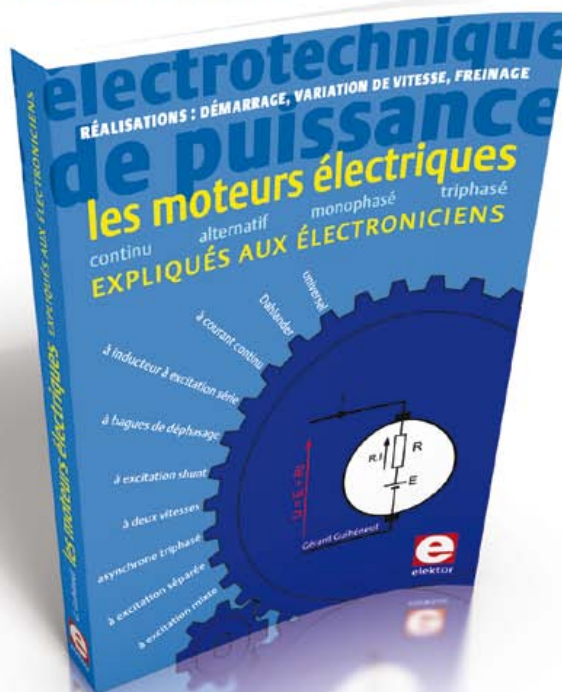
Le premier détaille les principes de variation de la vitesse des moteurs à courant continu : conversion alt./continu (redresseurs commandés par des thyristors) et conversion CC (hacheurs à transistors IGBT). Mise en pratique avec des variateurs pour mini-perceuse et pour train miniature.

Les moteurs à alimentation alternative monophasée (à induction, à bagues de déphasage, universel) du 2^e chap. font appel à une électronique de puissance. La puissance des réalisations proposées s'exprime en kW : démarreur à contacteur statique à deux points de commande pour moteur asynchrone monophasé à induction et variateur de vitesse pour moteur universel.

Reste le moteur électrique le plus utilisé dans l'industrie : le moteur asynchrone triphasé et ses différents principes de démarrage, de variation de vitesse et de freinage : démarreur électromécanique à contacteurs, démarreur-ralentisseur, convertisseur de fréquence ou couplage des pôles pour la variation de vitesse, moteur frein, freinage par injection de courant... Construisez le démarreur inverseur statique pour moteur asynchrone triphasé et découvrez la proximité entre électronique et électrotechnique : des portes NON-OU alimentées sous 12V commandent le sens de rotation d'un moteur de 1,5 kW alimenté sous 400 V en triphasé.

les moteurs électriques expliqués aux électroniciens - Gérard Guihéneuf

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



www.elektor.fr/moteurs

nouveau livre

le monde d'Elektor

compilation par
Wisse Hettinga
(Elektor)

Il ne se passe pas une minute sans qu'un passionné d'électronique ne découvre et réalise un projet électronique original. Tantôt pour le plaisir, tantôt pour le métier. Parfois l'amusement devient un commerce sérieux. Le monde d'Elektor, c'est un grand carrefour où se croisent les voies les plus diverses et se créent les liens les plus improbables. Ces pages sont les vôtres.



Cliché de femme

VERNON – Notre ami CJ (prononcer si djè), notre collègue de Circuit Cellar, nous parle des femmes. Ne cherchez pas le poster détachable en pages centrales ! Il s'agit des femmes ingénieures... D'après CJ, les ingénieurs des années 80-90 qui travaillaient avec des microcontrôleurs étaient tous des hommes blancs, moustachus, en chemises à col et des verres de lunettes comme des culs de bouteille.

Non, j'exagère. Il y avait déjà quelques types sympas (une dizaine ?), des personnalités enthousiastes, créatives et curieuses. Leurs goûts artistiques et leur style plus enlevé que la moyenne laissaient présager l'arrivée des talentueux électroniciens du 21^e siècle. Ceux-là mêmes qui ont créé les merveilles technologiques d'aujourd'hui. Et maintenant qu'on a tous les jouets dont on rêvait, on fait quoi ? Regardez bien leurs créateurs... Tous des hommes ! Depuis trop longtemps, les femmes sont aussi rares parmi les ingénieurs que les hommes dans une cour de récré à la maternelle, mais il semblerait que la donne s'équilibre. Quelques femmes sortent du lot.

En 2012, Circuit Cellar accueillait plusieurs innovatrices telles que :

- Ayse Kivilcim Coskun, *A Vision for 3-D Stacked Systems*, CC 264
- Hai (Helen) Li, *Embedded Inquiries*, CC 267
- Jan Axelson, *Debugging USB Firmware*, CC 268



Hai (Helen) Li

L'année 2013 s'annonce encore plus féminine, à suivre sur la page (toujours non détachable) :
<http://circuitcellar.com/category/female-engineers/>

Voulez-vous souder avec moi ?

- PUNE – Alors que les entreprises et les écoles européennes peinent à séduire de nouveaux candidats ingénieurs, il n'est pas rare de trouver des classes de plus de 100 étudiants en Inde. À Pune, au Bharati Vidyapeeth's College of Engineering for Women, de jeunes femmes étudient dans la joie et la bonne humeur les hautes technologies et l'électronique embarquée. Rien que dans cette ville, on trouve pas moins de 80 établissements de ce genre. Lorsqu'on voit dans quelles proportions ces jeunes et accortes recrues arrivent en Europe après leurs



études, on se dit qu'on en rencontrera forcément une tôt ou tard, et on se prend à rêver qu'on l'invitera : « Voulez-vous souder avec moi ? » .



Méto Radio Bravo

L'été dernier, nous vous proposons d'imaginer des circuits imprimés originaux, comme de créer un circuit fonctionnel à partir du plan de métro londonien. Yuri Suzuki, un concepteur de grand talent, s'est d'ailleurs engagé sur cette voie ténébreuse. Il a décalqué le plan du métro de Londres pour dessiner un poste de radio que l'on peut admirer et écouter au *Design Museum* de Londres. Sa collection vaut le détour.

L'affaire des robots tueurs

Ce n'est pas tous les jours qu'on appelle à débattre publiquement du bien-fondé d'une technologie qui ne sera peut-être envisageable que dans 20 ou 30 ans. C'est pourtant ce que proposent Human Rights Watch et la Clinique des



droits humains internationaux de la faculté de droit à Harvard, en publiant ce rapport : *Losing Humanity: The Case Against Killer Robots*. Affaire à suivre sur notre site satellite Techthefuture.com

Il est où le prof ?

Il y a quelques semaines, nous avons invité des enseignants hollandais à « abandonner leurs élèves », le temps de visiter le laboratoire d'Elektor. Pour rester en phase avec l'évolution de l'enseignement, rien de tel qu'un vrai contact humain. Cette agréable rencontre

a été enrichissante pour tous. Au fait, si vous passez dans le coin, envoyez-nous un petit message et nous vous accueillerons avec joie.



HP35 de mémoire de calculette

« Chers rédacteurs d'Elektor, Je me suis régalé avec votre récent article de *Rétronique* à propos de la calculette HP35. Quelle nostalgie ! J'ai été l'un des tout premiers utilisateurs d'un modèle qui à l'époque m'avait coûté 2 100 (feus) florins. C'est loin ! Je l'ai conservée et elle fonctionne toujours, à l'exception du bouton marche/arrêt devenu capricieux. Je possède aussi les modèles 31E, 71B, 18C, 28C et une 200LX. Connaîtrez-vous une bonne maison de retraite pour ces machines, ou un musée ? Chez moi, désormais, elles ne comptent plus guère que les pous-sières, mais je ne me résous pas à les jeter. »

Gerard Keijser, Pays-Bas

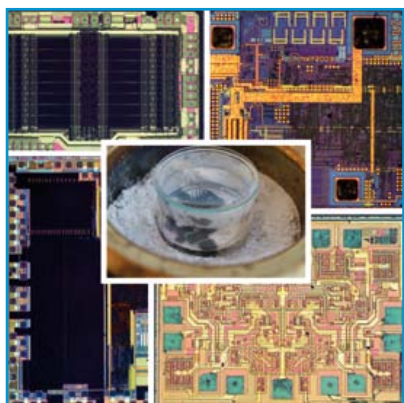
Réponse de la rédaction :

Vos réactions enthousiastes à propos de *Rétronique* nous réjouissent. Gerard a envoyé ses calculettes à Elektor, en échange d'un exemplaire du nouveau livre *Retronics*. Nous les avons rassemblées ici pour en faire profiter tout le monde.



Puces à la Russe

En Russie, la compagnie ZeptoBars déshabille chaque semaine une puce de son enveloppe en la plongeant dans un bain concentré d'acide sulfurique à 300 °C pendant une demi-heure. Et si ça ne suffit pas, ils terminent le boulot à l'acide nitrique ! La méthode est radicale mais donne un joli résultat dont on ne se lasse pas.



Les uns font circuler en toute transparence des photos à haute résolution de leurs circuits intégrés vedettes, d'autres cultivent sagement le mystère autour de leur architecture interne. C'est probablement là une forme de protection contre les tentatives d'ingénierie inverse, du moins au début de leur production. Nous ne voudrions pour rien au monde rater ces petites séances de voyeurisme électronique !

Chez Arduino, **pour une raclette ping-pong**

Chiasso – Si d'aventure vous passez par les bureaux d'Arduino à Chiasso, en Suisse, il y a fort à parier qu'on vous installe autour d'une table de ping-pong. Non pas pour jouer, mais parce que c'est la seule table assez grande pour accueillir 6 personnes. Elektor y était de passage en décembre pour un contrat de distribution mondiale des produits Arduino. Accord conclu au cours d'un joyeux repas.



De gauche à droite : Don qui essaye de convaincre Gianluca, Massimo qui fait semblant de relever son courriel, David qui tente d'échapper à la photo et Eduardo (l'initiateur de cette nouvelle coopération) qui présente son meilleur profil.



Le moment suprême de la signature.

À suivre...

du coq à l'âne... ou l'inverse !

Une conversation entre spécialistes à propos du transformateur de courant

Un intéressant échange de messages électroniques a suivi la publication de l'article « transformateurs de courant » dans le numéro d'octobre 2012 d'Elektor, page 40. Comme la rédaction figurait en copie de ces messages, nous partageons avec vous la remarquable progression, pour ne pas dire escalade, dans les propos de ces spécialistes.

Bonjour Jan,

En tant que concepteur de transformateurs de courant (TC) pour l'industrie aéronautique, j'ai trouvé fort intéressant l'article d'Ed Dinning. Je voudrais commenter la fig. 5. L'auteur montre la résistance de charge R_L du côté secondaire des diodes de redressement. Cette configuration entraîne une imprécision de la mesure liée à la tension de seuil des diodes qui varie avec la température. En tant que source de courant, le TC permet de placer la résistance de charge du côté des diodes où la mesure est effectuée, et de l'affranchir ainsi de la chute des diodes. J'ai exploité cette astuce pour connecter ma résistance de charge dans *Emergency Load Generator Meter* fig. 2 de l'article paru en février dans l'édition anglaise d'Elektor. L'utilisation de diodes Schottky réduit l'exigence de puissance sur le TC. J'ai aussi présenté les avantages des transformateurs à noyaux toroïdaux dans l'édition de mai 1996 de *Popular Electronics*, p. 53 sous le titre *All About Toroidal Transformers*.

Bien à vous,

BR, Chuck Hansen (USA)

Bonjour Chuck,

Je suis d'accord avec votre suggestion concernant la position de la résistance de charge pour éviter l'imprécision de la tension de seuil en fonction de la température. L'objectif de cet article était de proposer un instrument de mesure simple et isolé, accessible au bricoleur amateur, en dépit de ses petites imperfections aux abords du bas d'échelle. L'usage de diodes Schottky d'1 A les rend d'ailleurs pratiquement invisibles avec un courant de 50 mA. On aurait également pu améliorer la précision en réduisant la densité de flux, mais le noyau aurait été plus difficile à trouver ou à recycler. Par simple curiosité, quelle densité de flux employez-vous avec vos toroïdaux ? Lorsque je travaillais pour un fabricant de contacteurs, il y a bien longtemps, nous utilisions des alliages spéciaux qui opèrent à 0,01 T.

BR, Ed Dinning (UK)

Bonjour Ed,

Bonne idée de recycler d'anciens transfos. Difficile de changer le rapport de tensions de transfos du commerce quand, pour un meilleur couplage de flux, le secondaire est à l'intérieur de la bobine et le primaire à l'extérieur. Ajouter quelques spires extérieures n'est malaisé, car la fenêtre du noyau est remplie à ras bord pour réduire le poids et donc le coût. En aéronautique, le poids est une priorité absolue. J'ai donc pris l'habitude de pousser la densité de flux au-delà des normes usuelles. Les exigences varient d'après l'usage du TC mais avec du fer au silicium (B_{max} 16 kG ou 1,6 T), je limite la densité de flux à pleine charge à 3 200 gauss (0,32 T). Ici les concepteurs parlent toujours de densité de flux en lignes/pouces carrés. Avec une protection différentielle (deux TC connectés hors phase via un circuit de protection de charge) les TC ne peuvent pas provoquer de mise en sécurité inutile lors d'un courant de défaut dans le pire des cas, celui d'un courant de retour en provenance de la zone protégée par les TC. Je limite la densité de flux au courant d'erreur dans le cas le plus défavorable, en permettant encore des transitoires jusqu'à 1,4 T à la température maximale du noyau. Avec la charge spécifiée au secondaire et le courant primaire nominal, l'erreur d'angle de phase ne doit pas dépasser 20 min et les erreurs de phase de tous les TC dans la zone de protection différentielle ne doivent pas différer de plus de 5 min par rapport à la moyenne des erreurs. La différence d'amplitude de courant dans le secondaire entre les deux TC ne doit pas dépasser 0,25 % de la moyenne des amplitudes de leurs courants au secondaire.

Si le TC est utilisé comme protection de courant excédentaire ou comme limiteur avec un régulateur de tension à alternateur, la section du noyau que j'utilise dépend de la résistance de charge du circuit plus le nombre de diodes de redressement. Pour la métrologie et l'instrumentation, les normes militaires et aéronautiques définissent que l'impédance de charge à 400 Hz doit être $1,48 + j2,09 \Omega \pm 10\%$. Nous utilisons surtout des noyaux toriques en fer à 3 % de silicium pour nos TC, mais d'autres alliages existent pour des applications spéciales : le Satmu métal (alliage spécial à grains orientés de 65% de fer-nickel, recuit dans un champ magnétique, développé par *Telcon Ltd* en Grande-Bretagne, les concepteurs à l'origine du Mu métal) ; *Deltamax Round Orthonal* ; poudre métallique *Supermalloy* pour des fréquences > 2 kHz.

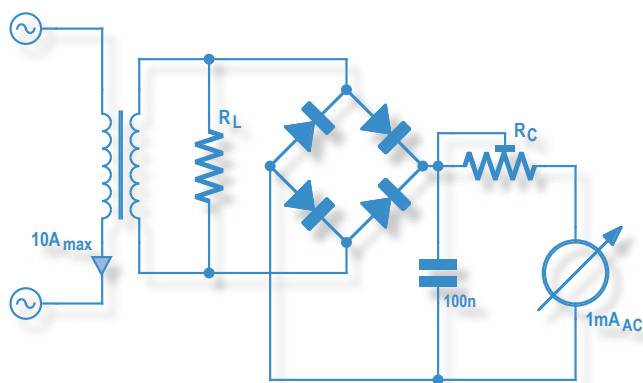
BR, Chuck Hansen (USA)

Bonjour Chuck,

Les cours de magnétisme étaient bâclés dans les écoles britanniques, d'abord en pouces puis en cgs (centimètre, gramme, seconde). C'est seulement en allant à la fac, où ils utilisent le système international (pas le MKS), que tout s'éclaire. Il y a ici une société qui essaye d'abolir le centimètre.

Bonne idée quand on voit comment l'école a bousillé le système métrique. Nous on bobine les transfos en commençant par le 1^{aire} pour faciliter les modifs. Souvent, on bobine aussi le 1^{aire} et le 2^{aire} côte à côte par sécurité (lignes de fuite et distance d'isolement). Faciles à modifier, mais délicats à régler. Exception faite des transfos audio et des alims à découpage, où le 1^{aire} et le 2^{aire} sont imbriqués par sections pour un couplage optimal. J'ai été dans l'aérospatiale aussi. Un des projets utilisait du *supermendur* (fer-cobalt) et tournait à 2 T / 400 Hz avec refroidissement forcé. Il fallait porter des protections à cause de la magnétostriction. C'était pour un transformateur/redresseur sur un avion de chasse. Un projet plus intéressant portait sur des convertisseurs de fréquence sur avions militaires de transport, à des fréquences inhabituelles (380 à 620 Hz) avec refroidissement par kérosène. Il fallait passer de 3 phases à 18 phases. On simplifiait peut-être les alternateurs et améliorait le facteur de puissance, mais ça compliquait drôlement le bobinage et l'électronique ! Délibérément, j'ai toujours évité tout ce qui concerne l'opérateur j et le facteur de puissance dans les articles Elektor.

BR, Ed Dinning (UK)



Bonjour Ed,

Je suis d'accord avec vous pour l'opérateur j. Quand nous avons commencé les tests sur les TC de mesure, on m'a autorisé à utiliser une résistance bobinée de 2,56 Ω pour la charge. Et ça n'a rien changé aux mesures.

BR, Chuck Hansen (USA)

Bonjour,

Il m'est venu une idée d'amélioration en regardant le deuxième exemple cité dans l'article « transformateurs de courant » du numéro d'octobre 2012. On peut y voir la sortie du transformateur de courant branchée à la résistance de charge (comme il se doit). Cependant, le redresseur en pont prend le signal de la tension de la résistance de charge et à cause de la tension de seuil du pont, la sortie n'est plus tout à fait linéaire, jusqu'à disparaître complètement avec un courant trop faible.

Une meilleure solution serait de redresser d'abord le courant de sortie du secondaire et puis d'appliquer le courant redressé à la sortie du pont avec la résistance de charge. Comme la sortie du transformateur est un courant et agit comme une source de courant, ce même courant doit passer par la résistance de charge du pont redresseur (le transformateur doit seulement fournir une tension légèrement supérieure pour dépasser le V_f de la diode et permettre le passage du courant du secondaire. Ainsi, le courant qui traverse la résistance de charge est unidirectionnel et un simple filtre RC avant le galvanomètre suffit à lisser et adapter le signal. L'instrument répond alors aux signaux les plus faibles et on n'a plus de soucis à cause de la tension de seuil du pont redresseur. Vous avez mentionné les effets de chute de tension du pont redresseur dans le dernier paragraphe et je suis d'accord avec les maths et la méthode du reste de l'article.

Merci pour cette présentation très didactique à propos des transformateurs de courant. J'espère que cela éclairera les ingénieurs et les passionnés qui se posaient les mêmes questions.

KR, Jim Mettler, Triad Magnetics

des nouvelles du labo virtuel

Clemens Valens
(Elektor .Labs)

Elektor a un nouveau cœur : *Elektor dot labs*, haut-lieu de la fécondation des idées, de la gestation des projets et de la mise au monde de circuits mûrs, débogués au labo. Nous travaillons sur des protocoles d'optimisation dont vous, lecteurs, bénéficiez en ligne et dans la revue. C'est nouveau, tout le monde n'est pas encore convaincu, forcément, le changement et l'innovation dérangeant, mais ce cœur bat. Et comme les Shadoks, il pompe. De plus en plus fort.

Toujours plus de volts et d'ampères

L'un des premiers projets postés sur .LABS n'en était pas un. Il s'agissait d'une commande pour une alimentation de laboratoire. Pas une alim de tous les jours, non, l'auteur du premier message voulait 40 V au lieu des 30 V max. habituels. Apparemment il ou elle n'avait pas les capacités de la concevoir lui-même (ou elle-même) et s'est donc adressé(e) à nous. Et comme l'une de nos missions est de concevoir pour vous, nous avons relevé le défi.

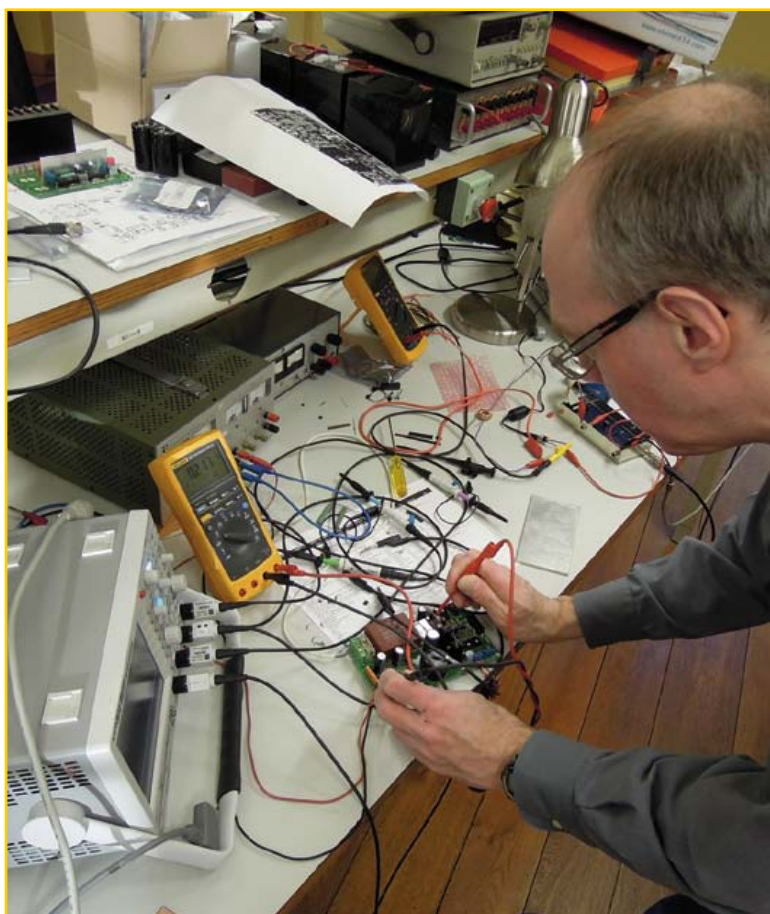
Force est d'admettre après quelques mois que la mèche est un peu longue. Le projet est encore loin d'être achevé, mais il avance, et quand Ton (prononcer « *tonne* ») Giesberts, mon collègue chargé de le mener à bien, m'a montré où il en était, j'ai eu la surprise de constater que... ça marche ! Pas encore à la perfection, mais, à ce stade, quelques petites oscillations et des traces de bruit ici ou là,

c'est gérable. Ce cher Ton nous mitonne une nouvelle méthode entièrement de son cru. Au lieu de recourir à un régulateur à tension de sortie variable, utilise un régulateur à tension fixe et lui fait croire qu'il est un régulateur variable. Ou quelque chose de ce genre. Ça vous intéresserait, des précisions sur son circuit ? Tous les tenants et les aboutissants sont en ligne sur .LABS, allez-y. Vous trouverez également dans ce numéro-ci une double page d'avant-première.

Une des raisons du ralentissement de ce projet a été le délai de livraison d'un condensateur à très faible ESR qu'il a fallu faire venir de l'autre bout du monde. Le croirez-vous, il y a 1,36 milliard de Chinois, mais pas un seul qui ait pu nous livrer ce condensateur. Après des mois de recherches, nous avons fini par en trouver un, mais ce brave homme, au moment de glisser le précieux composant dans l'enveloppe, le laisse échapper ; le condo sans ESR rebondit sur la table, roule vers la table voisine et de là tombe dans un bac qui contenait, en vrac, quelques milliers de condensateurs ordinaires dont rien ne permet de le distinguer. On patientera en se racontant d'autres histoires de ce genre.

L'initiateur du projet demandait 2 A pour cette alim, Ton trouve qu'il en faut 3 A. Non, il vient encore de changer d'avis, ce sera 6 A. Voilà pourquoi il faudra encore patienter avant de voir apparaître son alim dans le magazine.

Suivez la progression de Ton en ligne :
www.elektor-labs.com/120437



Toujours plus haut et plus loin en ballon

Je ne sais pas pour vous, mais mon monde à moi a 3 dimensions. Ou plutôt quatre, si je prends le temps. Et au-delà ? Mathématiciens et physiciens s'escriment sur le sujet depuis des lustres. Et si je ne me trompe pas, ils en sont à 26. C'est trop subtil pour moi. Déjà que j'ai raté la 5^e dimension en 1967 ! Oui, elle est là depuis tout ce temps, sous nos yeux, ou plutôt sous nos oreilles : en 1967, un groupe américain appelé *The 5th Dimension* lance le tube *Up, Up and Away* qui bientôt emportera cinq *Grammy Awards*. Et qui en est l'auteur ? Jimmy Webb, hé hé. Vous me voyez venir ? La 5^e dimension, c'est tout bonnement le *world-wide jimmy web*. Vous rigolez, moi je suis sérieux. Pas convaincus ? Allez sur le site Elektor dot LABS (oui, c'est mon nouveau dada) et dites-moi ce que vous voyez ? Une proposition d'un certain *ale* concernant une commande de ballon météorologique. Ballon, internet, 5^e dimension, ça colle, non ?

Examinez ce projet d'*ale*, qu'il appelle *Sounding Balloon Main Unit Controller* et mérite le détour.

C'est un système qui transmet par liaison radio FM six signaux analogiques plus des données de localisation. Ces canaux permettent de renvoyer au sol des paramètres tels la température, les rayonnements infrarouge et ultraviolet, le niveau de charge de la batterie (à quoi bon ?) ou la pression atmosphérique. Comme il faut qu'il puisse monter haut et loin dans un ballon, ce système devra rester léger et bon marché.

À propos de ballon, saviez-vous qu'Elektor a publié un livre qui est à la fois le premier livre au monde apte à la communication NFC (avec même une puce vierge que VOUS pourriez programmer) et un beau livre sur les ballons. C'est le premier livre à cinq dimensions jamais publié.



Entrez dans la 5^e dimension :

www.elektor-labs.com/9121102594



Les préférés de la Rédaction

Sur .LABS, la Rédaction a retenu quelques projets pour une publication prochaine. Seulement, dans certains cas, c'est l'impasse car l'initiateur du projet ou *original poster* (OP) a disparu de la circulation (la x^e dimension, sans doute...). C'est pourquoi nous vous prions instamment de ne pas abandonner vos propositions sur le site .LABS. Et jetez un coup d'œil dans le courrier reçu à l'adresse avec laquelle vous vous êtes inscrit ; nous nous en servons pour communiquer avec vous, notamment en vue d'une publication.

Voici les projets retenus :

**EDITOR'S
CHOICE**

Laser TV



Le projet de TV laser TV posté par *hpt* est une tentative de projection d'une image vidéo au moyen de 30 miroirs rotatifs montés sur moteur de tête VHS. Des essais ont été menés. Le principal défi est la synchronisation de phase de la plaque supérieure, et l'OP lance un appel à collaboration pour programmer la PPL du moteur sous BASCOM (ou une autre solution similaire par logiciel).

www.elektor-labs.com/9120502218



Analog Theremin

Le Theremin est considéré comme le premier ou l'un des premiers instruments de musique élect(ron)iques. On en joue en déplaçant les mains entre deux antennes. L'internet fourmille de circuits qui délivrent un détestable signal carré. L'OP D.Philips voudrait construire un Theremin analogique de qualité, qui produise du sinus. Pouvez-vous l'aider ?

www.elektor-labs.com/9120502157

Waterbed energy saver

Avez-vous un lit à eau comme l'OP *Schwabix* ? Lui trouve que le chauffage de son matelas consomme trop et envisage donc de couper le thermostat durant la journée, puisque le lit n'est pas utilisé. En début de soirée, le chauffage doit être enclenché automatiquement assez tôt pour que la température ait retrouvé un niveau suffisant au moment où *Schwabix* se couche. Il aimerait qu'on l'aide et qu'on vienne en discuter avec lui.

www.elektor-labs.com/9121102686



Portable Radio/Media Player

rsavas est le pseudo de l'OP qui a conçu ce qu'il appelle *Portable Radio/Media Player*. C'est une radio, un lecteur de MP3/AAC/WMA (clé USB ou carte SD) et il se connecte à un PC au moyen du codec USB (*PC Stereo/sound card*). Il est doté d'un sélecteur de source, d'une commande de volume, d'un ampli pour casque et d'un ampli de 20 W en classe D.

Ses caractéristiques sont actuellement limitées par le logiciel et l'initiateur du projet cherche de l'aide.

www.elektor-labs.com/9120502222

elektor labs

Sharing Electronics Projects

Home News Proposals In Progress Finished
Search

The PSoCcamorph

Contribute and win!

Proposals	In Progress	Finished
<div>Active Popular</div>  <p>Pic dev board, (with programmer)</p> <p>★★★★★</p>	<div>Active Popular</div>  <p>Mains Gate: Programmable Relay & Energy Monito...</p> <p>★★★★★</p>	<div>Active Popular</div>  <p>Switched 7905 replacement</p> <p>★★★★★</p>

About Elektor.LABS

Elektor Labs

Create a Project

Create a new project or enter a proposal

Get help, feedback & votes from other visitors, and maybe you will get Elektorized too!

Challenges

Win a PSoC 5 Development Kit!

All challenges...

www.elektor-labs.com



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



+49 (0)4422 955-333



+49 (0)4422 955-360

**Qualité professionnelle
à des prix réduits !**

Technique des mesures

Technique d'éclairage

UNI-T.



Thermomètre infrarouge

Mesure de la température sans contact !

- Plage de mesure : -18°C ... 380°C
- Distance à la taille du point : 10:1
- Dissolution : 0,1°C
- Exactitude de mesure : $\pm 2^\circ\text{C} / \pm 2\%$

Bestellnummer:

UT 300B

21,95

Multimètre numérique 5 en 1

combine les fonctionnalités de mesure environnementale avec celles d'un multimètre

- Affichage 3 4/5, 4000 comptes
- Niveau sonore, intensité d'éclairage, humidité de l'air, température et fonctionnalités d'un multimètre
- Protection de surcharge dans toutes les plages
- Fonction de mesure de la valeur relative
- Sécurité: EN 61010-1 ; CAT III 600 V

PeakTech



Bestellnummer:

PEAKTECH 3690

66,95

Unité d'alimentation compacte-laboratoire, 100 W

- Écran LCD illuminé, 4 chiffres grande taille
- Tension de service : 230 VAC / 50 Hz
- Tension de sortie : 1 - 36 VDC
- Courant de sortie : 0,05 - 3 A DC
- Rendement : 85 %
- Sécurité : EN-61010-1



Manson

Bestellnummer:

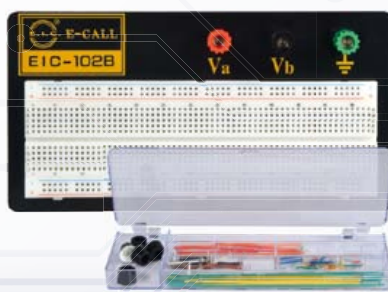
NSP 3630

99,95

Besoin de l'atelier

Expériences

Outils de développement



Kit pour débutants :

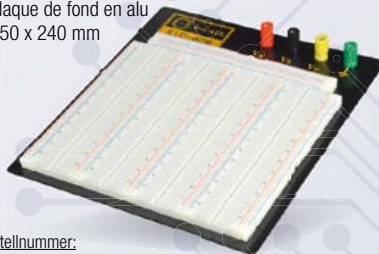
- 3x bornes de raccordement
- 630/200 contacts
- dont kit de 70 fils jarretières

STECKBOARD START

9,75

Tableau de prises

- 2560/700 contacts
- Bornes de raccordements 4 pôles
- Plaque de fond en alu
- 250 x 240 mm



Bestellnummer:

STECKBOARD 4K7V

14,95

Kit de fils jarretières

140 pièces, divers coloris

Bestellnummer:

STECKBOARD DBS

3,65

RM 2,54, 39 rangées

1,5 mm/stratifié de papier
Largeur : 100 mm
Trou : 1 mm
Revêtement Cu 35 μ



Bestellnummer:

H25SR050

0,60

50 mm

H25SR150

1,45

150 mm

H25SR075

0,90

75 mm

H25SR160

1,50

160 mm

H25SR100

0,98

100 mm

H25SR200

1,85

200 mm

Technologie réseau et PC

Alimentation en courant électrique

Technologie satellite et TV

Blocs à découpage pour LED



- IP64, pour applications intérieures et extérieures
- Protection de surcharge par limitation de courant, auto-récupération
- Protection de court-circuit, de surcharge et de surtension

Série LPC - Courant de sortie fixé

- Entrée : 90 - 264 VAC



Bestellnummer:

	U (V)	I (mA)	URW (mV)	P (W)
MW LPC-20-350	9,85	9,0 - 48	350	200
MW LPC-20-700	9,30	9,0 - 30	700	200
MW LPC-35-700	14,30	9,0 - 48	700	200
MW LPC-35-1050	15,55	9,0 - 30	1050	200

➔ Plus de blocs d'alimentation :

<http://rch.it/7C>

Série LPV - Tension de sortie fixée

- Entrée : 90 - 264 VAC ; 127 - 370 VDC



Bestellnummer:

	U (V)	I (mA)	URW (mV)	P (W)
MW LPV-20-12	10,95	12	0 - 1,67	120
MW LPV-20-24	10,95	24	0 - 0,84	150
MW LPV-35-12	16,25	12	0 - 3	120
MW LPV-35-24	16,25	24	0 - 1,5	150

Série HLN - Configuration réglable

- Circuit électrique PFC intégré
- Tension de sortie et limitation de courant réglables
- EN61000-3-2, classe C, EN55015, EN61347-1, EN61347-2-13
- Entrée : 90 - 264 VAC ; 127 - 370 VDC



Bestellnummer:

	U (V)	I (A)	URW (mV)	P (W)
MW HLN-40H-12A	36,85	12	0 - 3,33	150
MW HLN-40H-24A	36,85	24	0 - 1,67	150
MW HLN-60H-24A	39,95	24	0 - 2,5	150
MW HLN-80H-12A	44,95	12	0 - 5	150
MW HLN-80H-24A	44,95	24	0 - 3,4	150

la tension (de 5 V) nécessaire de Mr Cyclopède

5 V à la force des mollets

Philip Jaschewski
(Elektor)

Le cycliste de 2013 part en vadrouille sur son vélo avec une foule d'accessoires électroniques dans ses fouilles. La tentation est forte de les alimenter à partir du réseau électrique du bord, car, en 2013, oui Monsieur, il y a un réseau électrique sur les vélos. Ne reste donc qu'à trouver un convertisseur. Le voici.

Le 28 mars 1983, dans la «Minute nécessaire de Monsieur Cyclopède», Pierre Desproges demandait « *Comment reconnaît-on un homme d'une femme ?* » [3]. En fait, la vraie question était « *Comment reconnaît-on la joconde du jocond ?* », mais ce n'est pas le sujet de cet article. Ce qu'il nous faut ici, c'est un circuit pour mettre tout le monde d'accord autour d'une même tension continue de 5 V obtenue à la force des mollets à l'aide d'une bonne vieille dynamo.

Une dynamo fournit habituellement une puissance de 3 W environ sous 6 V. Les unes, clas-

sement, oscillent de quelques dizaines à quelques centaines de Hz. Il faut donc commencer par la redresser avant de la stabiliser à 5 V.

Voyez la **figure 1** : un régulateur classique et un redresseur. Et pourtant, les circonstances ne sont pas tout à fait banales : la tension d'une dynamo, à vide ou quand la charge est faible, augmente linéairement avec la vitesse de rotation. Autrement dit, même à vitesse modérée, la tension sur le redresseur atteint aisément 6 V. Avec de bons mollets, à fond de train, vous dépasserez les 12 V.

siques, sont montées contre une roue et entraînées par friction, les autres, plus récentes, sont incorporées au moyeu. Dans un cas comme dans l'autre, la tension est alternative, à une fréquence qui peut

Comme la courbe de cette tension est à peu près sinusoïdale, la tension aux bornes de C1 atteint, en l'absence de charge, sa valeur de crête. IC1 accepte certes 25 V à l'entrée, mais il vaudrait mieux se protéger. C'est la diode zener D1 qui veille à écrêter la tension.

À l'entrée, il y a l'inverseur bipolaire S1 qui commute entre l'éclairage et l'alimentation des accessoires. Eh oui, voir ou nourrir, il faut choisir. Il y a probablement moyen de tirer plus de la dynamo que ses 3 W de puissance nominale, mais ce ne sera jamais assez pour alimenter et l'éclairage et les accessoires.

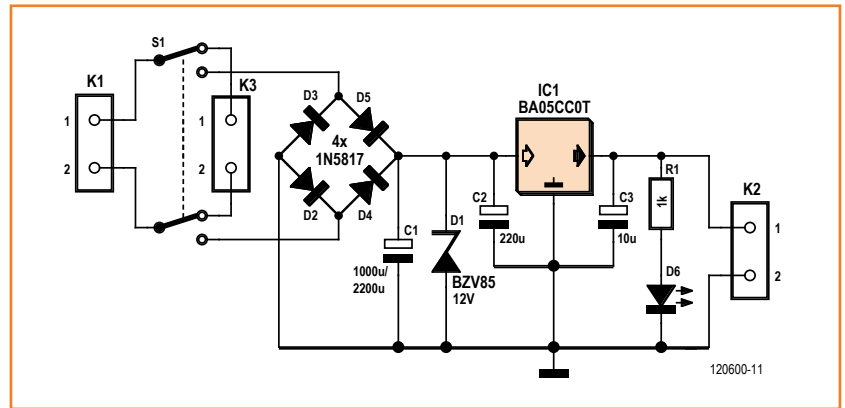
Caractéristiques techniques

- sélecteur éclairage-accessoires
- tension de sortie : 5 V
- courant : min. 0,5 A
- convient pour alimenter/recharger divers appareils portables

Pour ne rien perdre de la tension en charge, le régulateur IC1 n'est pas un 7805 ordinaire, mais un modèle à faible chute (LDO) ; il régule bien, même quand la tension d'entrée n'est que faiblement supérieure à la tension stabilisée. Pour donner 5V, le 7805 ordinaire exige au moins 8 V à l'entrée. Les variantes améliorées se contentent de 7 V. Le LM2931T, compatible broche à broche, se débrouille déjà avec seulement 5,6 V. On peut aussi gratter un peu de marge avec des diodes Schottky SB140 ou 1N5817 au lieu d'un redresseur ordinaire au silicium.

Les dessins du circuit imprimé sont téléchargeables gratuitement [1]. Vous pouvez aussi commander un circuit prêt à l'emploi, de facture professionnelle.

La réalisation ne devrait poser aucune difficulté selon le modèle de la **figure 2**. Pour tester le circuit, commencez par l'alimenter par un transfo de 6 V. Quand vous le monterez sur le vélo, disposez le circuit dans un boîtier en plastique étanche, car l'électronique et l'humidité ne font bon ménage. Sur certains vélos, la dynamo n'a qu'un fil, l'autre conducteur étant le cadre. Dans ce cas, le circuit imprimé sera relié lui aussi au cadre du vélo (p. ex. par la borne K1-2). Attention, ceci exclut tout contact électrique entre la masse de la sortie K2-2 et une pièce métallique du vélo.



Tout ça, Desproges ne le savait pas quand il posait la grave question. « *Comment reconnaît-on un homme d'une femme ?* » et répondait : « *À la forme de son... vélo !* »
Étonnant, non ?

Figure 1.
Le circuit de l'alimentation cyclopédique est d'une simplicité exemplaire.

(120600 – version française : Monsieur Cyclopède)

Liens

- [1] www.elektor-magazine.de/120600
- [2] <http://eagleup.wordpress.com>
- [3] <http://www.ina.fr/divertissement/humour/video/CPC83050880/sachons-reconnaitre-la-joconde-du-jocond.fr.html>
ou <http://goo.gl/SLwfj> pour aller plus vite

Liste des composants

Résistances :

R1 = 1 kΩ

Condensateurs :

C1 = 1000 µF/35 V au pas de 5 mm, ø 10 mm
C2 = 100 µF/35 V au pas de 2,5 mm, ø 6 mm
C2 = 10 µF/16 V au pas de 2,5 mm, ø 5 mm

Semi-conducteurs :

D1 = diode zener 24 V, 1 W
D2 à D5 = 1N5817 ou SB140*
IC1 = BA05CC0T, LDO, 1 A, TO220*
LED1 = LED verte, 5 mm

Divers :

K1 à K3 = bornier à vis encartable à 3 points, pas de 5 mm
S1 = inverseur bipolaire, 24 V/1 A
circuit imprimé 120600-1

* cf texte

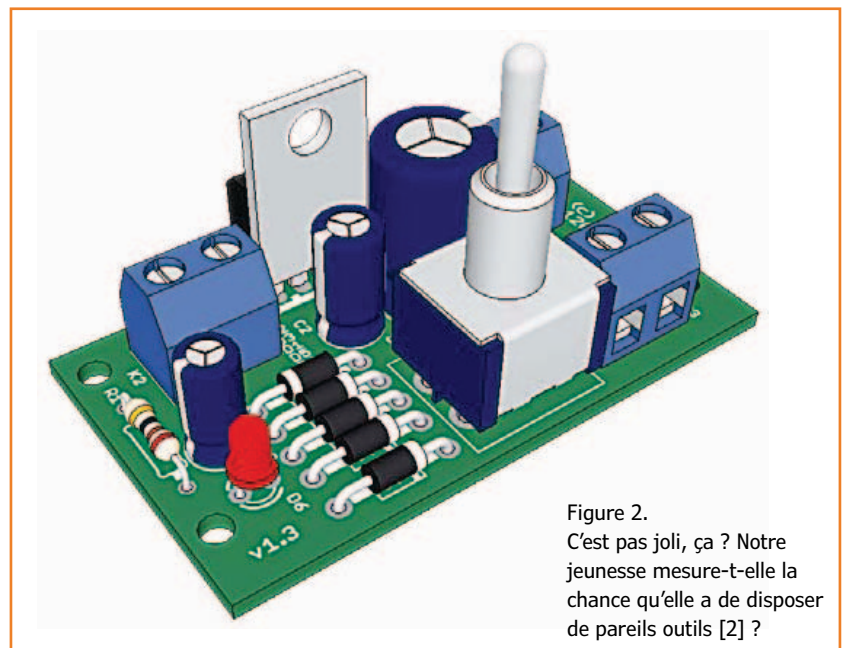
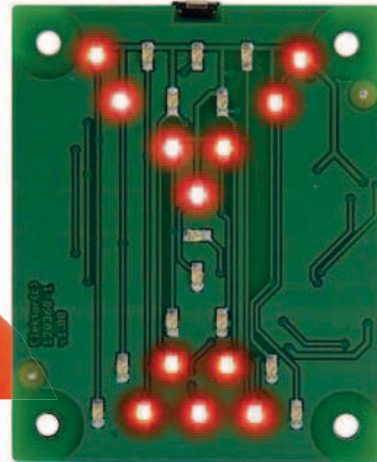


Figure 2.
C'est pas joli, ça ? Notre jeunesse mesure-t-elle la chance qu'elle a de disposer de pareils outils [2] ?



n'oublie pas ta brosse-à-temps !



temps à consacrer au brossage de leurs dents. Voici donc la *brosse-à-temps*, accessoire éducatif et divertissant. Programmeurs en C ou fans d'AVR, ce sablier à LED vous intéressera aussi !

La brosse-à-temps s'inspire d'un sablier classique et son fonctionnement ne nécessite donc a priori aucune explication. Certainement pas pour vos enfants qui savent d'instinct allumer la télé et le routeur Wi-Fi ou télécharger des jeux sur votre téléphone tactile.

Appuyez sur le bouton pour démarrer le minuteur. Les LED s'allument successivement pour imiter l'écoulement du sable. Un bip discret est émis toutes les 30 s et quand le temps est écoulé. Une LED = 10 s.

Le circuit est un exemple typique de petit micro-contrôleur qui remplace avantageusement la ribambelle de composants qu'auraient nécessités les classiques CMOS, TTL ou NE555. Un ATtiny13

Jere Manner (Finlande)

Quoi de plus authentique qu'un bon vieux sablier ? Bien que silencieuse et entièrement dépourvue d'électronique, la douce et insaisissable chute de sable semble fasciner la plupart des enfants. Peut-être à cause de leur perception innée d'un temps qui s'écoule. Difficile de le savoir, tout comme il est difficile de comprendre pourquoi la plupart des enfants n'arrivent pas à respecter l'heure d'aller au lit ou le

Liste des composants

Résistances :

(CMS 0805)

R1 = 1 kΩ 1%

R2 = 10 kΩ 1%

R3, R4 = 75 Ω 1%

R5 à R16 = 120 Ω 1%

Condensateurs :

C1, C2, C3 = 100 nF 25 V, CMS 0805

Semi-conducteurs :

D1 à D24 = LED, rouge, CMS 0805, p.ex. KPHCM-2012SURCK chez Kingbright ; Farnell 1686067

D25 = LED, verte, CMS 0805, p.ex. KPHCM-2012CGCK chez Kingbright ; Farnell 1686075

IC1 = ATtiny13A-SSH, boîtier SO-8, programmé, Elektor 120368-41

IC2, IC3 = 74HC164D, boîtier SOIC-14

T1 = BC850C, boîtier SOT-23

Divers :

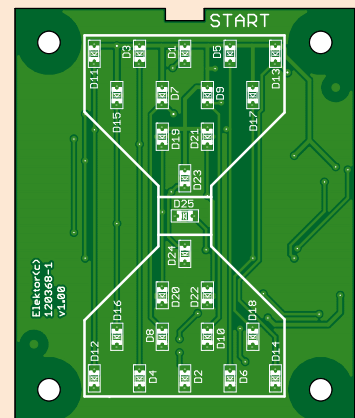
BT1 = pile bouton CR2032 et porte-piles ; Farnell 2064725

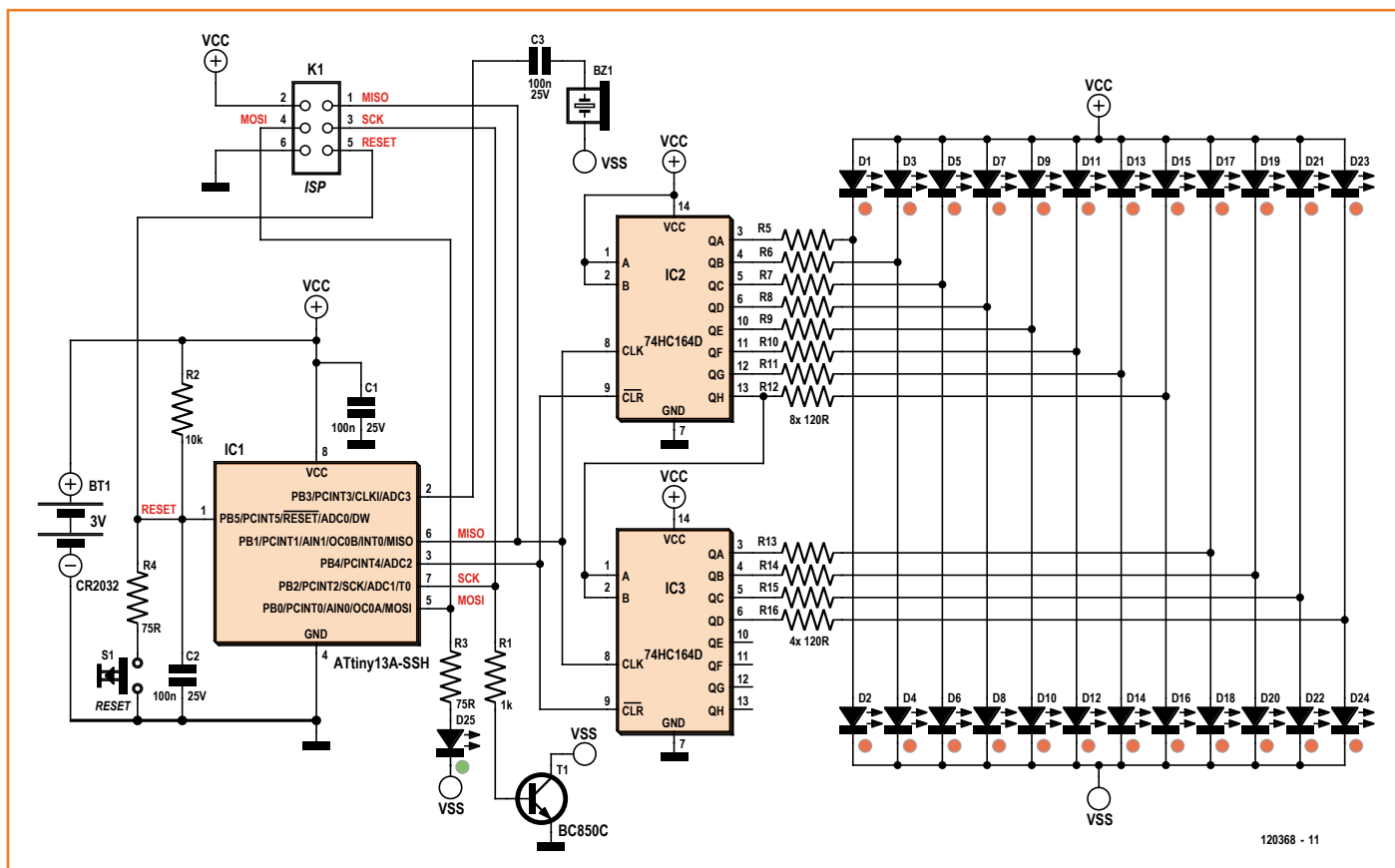
BZ1 = buzzer piézo, Farnell 1192551

K1 = embase à 6 broches (2x3), pas de 2,54 mm

S1 = bouton-poussoir à actionnement latéral, Farnell 1761633

circuit imprimé 120368-1, voir www.elektor-magazine.fr/120368





exécute un microprogramme qui contrôle le signal d'horloge nécessaire aux registres à décalage IC2 et IC3, ainsi qu'au transducteur piézo. Les registres 74HC164D pilotent une matrice de LED (CMS) via les résistances-talons R5 à R16. Le choix de petits composants et le dessin du circuit imprimé donnent un joli montage, léger et compact.

Le projet initial de l'auteur a été remanié au sein d'Elektor Labs. Points mineurs, mais essentiels : ajout de R4 et d'une embase ISP (programmation *in-situ*) afin que les amateurs de programmation puissent modifier le circuit et graver leurs propres puces à leur guise ; révision du code source en C du programme, et changement de quelques composants dû à une rupture de stock locale. Reste la capacité critique de la pile bouton CR2032 prévue. Celle-ci ne tiendra pas longtemps à cause du courant de LED. Remplacez-la de préférence par un modèle plus puissant, des piles AA par exemple. Le code source peut être téléchargé gratuitement [1].

(120368 – version française : Hervé Moreau)

Trucs & astuces d'Elektor Labs

Collez le porte-piles à la platine pour éviter de l'arracher lors d'un changement de pile ☹.

Programmez le contrôleur **avant** de le souder à la carte, ou utilisez une alimentation externe. Lors de la programmation, la tension de la pile CR2025 chute de 3,0 V à 2,3 V → initialisation du µC, mises en veille → situations imprévisibles ☹.

Un boulon ou une vis fixé(e) à un angle de la platine fera un support au sablier.

Liens

[1] www.elektor-magazine.com/120368

À vous de jouer !

Dessinez un boîtier et fabriquez-le par impression 3D.

Renforcez l'alimentation pour l'adapter au courant de LED.

Ajoutez un interrupteur au mercure (cf. « alarme pour moto », www.elektor.fr/120106) pour lancer le minuteur comme un vrai sablier.

Partagez vos réalisations sur www.elektor-labs.com !



bougie à LED bicolore

avec couleur réglable et détecteur de courant d'air

Vous trouvez les bougies à LED aussi attrayantes qu'un feu rouge ? Un ATtiny et une LED bicolore vous feront changer d'avis !

Jörg Trautmann (Allemagne)

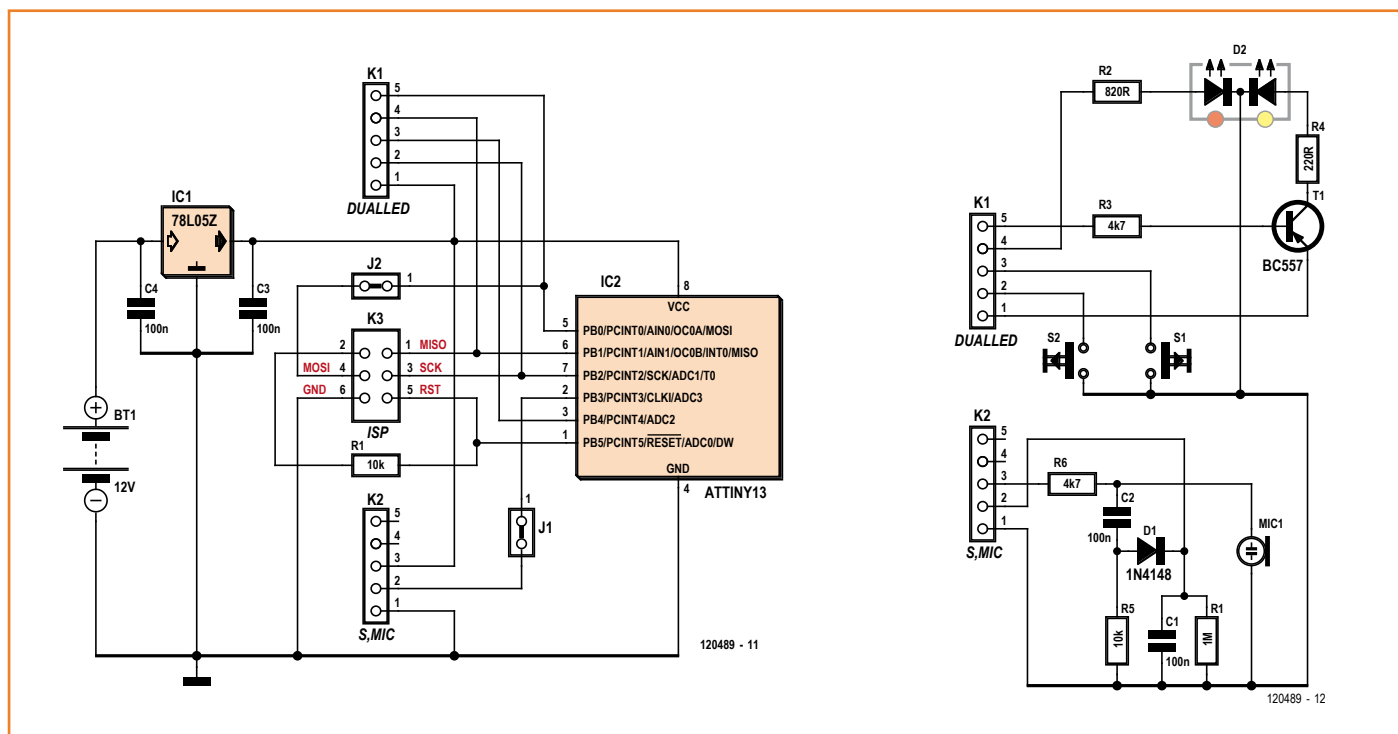
Il existe d'innombrables modèles de bougies à LED. Le plus souvent conçues de façon simple, elles contiennent les mêmes puces que celles utilisées dans les cartes de vœux musicales. La mélodie enregistrée module la luminosité de la LED pour reproduire avec plus ou moins de réussite le vacillement d'une flamme de bougie. Certains modèles plus chers sont équipés d'un détecteur de courant d'air qui permet de « souffler » la bougie. Pourtant, aucune de ces bougies ne nous leurre : la couleur de leur flamme est soit d'un jaune trop froid, soit d'un orange vif irréaliste. Même les LED dont la luminosité est commandée par le rythme d'une musique sont loin de donner l'illusion d'être de vraies bougies. Pour paramétrer correctement la couleur de

flamme de notre bougie à LED, il nous faut les caractéristiques spectrales des LED et des flammes de bougie. La flamme d'une bougie normale émet un rayonnement de longueur d'onde d'environ 600 nm, le spectre lumineux d'une LED jaune va de 565 à 590 nm, et celui d'une LED rouge de 625 à 740 nm. Un moyen de combler l'intervalle entre 590 et 625 nm est de combiner les intensités de chaque LED. Rien de plus pratique pour cela que d'utiliser une LED bicolore jaune et rouge.

Microcontrôleur minuscule

Le circuit de la bougie exploite un ATtiny13 [1]. Ce microcontrôleur lilliputien est doté de deux sorties MLI (OC0A et OC0B) qui permettent de modifier la luminosité de la LED bicolore. La LED rouge consomme peu de courant, car elle est la moins exploitée des deux. Elle a donc pu être

Figure 1.
Les deux faces du circuit.



reliée directement au microcontrôleur par une résistance-talon R2 de 820 Ω . Son courant est ainsi maintenu sous les 5 mA maximum que peut fournir le microcontrôleur. La LED jaune, la plus active, consomme plus que ces 5 mA admissibles. Elle est donc alimentée par le transistor PNP T1. La valeur de 220 Ω choisie pour R4 garantit que le courant qui la traverse n'excèdera pas 15 mA. Les deux boutons servent à régler la couleur de la flamme. Tant que l'on appuie sur S1, la composante rouge augmente et la jaune diminue. La touche S2 a l'effet inverse. Le nombre de pas de réglage du rapport cyclique est 256, soit une résolution de 8 bits.

La valeur de ce rapport impulsion-pause est modifiée par un générateur de nombres aléatoires. La variation maximale d'amplitude est de 35 % pour la LED jaune, de 10 % pour la rouge. Ce sont ces variations aléatoires qui donnent son réalisme à la couleur de la flamme. Le scintillement qui en résulte est faible, assez naturel. Avoir trouvé le réglage idéal est bien, le retrouver est encore mieux. Lorsque la bougie est mise hors tension, la dernière valeur réglée est donc enregistrée dans l'EEPROM du microcontrôleur (ses 64 ko suffisent !) et sert de valeur par défaut à la prochaine mise sous tension.

Le capteur de courant d'air est construit avec une capsule de microphone à électret (Mic1). La tension de sortie du microphone est lissée par D1 et C1, puis amenée à un convertisseur A/N. R1 sert de résistance de décharge. Le niveau de déclenchement a été choisi de façon à ce qu'un faible courant d'air ne suffise pas à influencer la pseudo-flamme. Au-delà de ce seuil, les LED scintillent et deviennent plus faibles. Elles s'éteignent si la pression d'air agit plus de deux secondes. Souffler brièvement sur le capteur ranime la bougie. La constante *Trigger_value* doit être adaptée à la sensibilité du microphone, une valeur élevée diminuant la sensibilité du capteur. Pour faciliter la modification de cette valeur ou du programme, l'interface ISP (programmation *in situ*) est reliée au connecteur K3.

La bougie peut être alimentée par deux piles bouton CR2032. Comme la tension d'alimentation de l'ATtiny13 ne doit pas dépasser 5,5 V, le circuit utilise un régulateur de tension LDO (*Low Drop-out*) 78L05. Le courant de repos est d'environ 1 mA, donc retirez les piles si la bougie n'est plus utilisée pendant quelque temps.

Montage sur deux circuits

Nous avons réparti le circuit sur deux circuits imprimés afin de garder l'ensemble aussi compact que possible. Les deux parties se relient à l'aide d'un connecteur. Deux cavaliers sont à souder sur le circuit « microcontrôleur » ; le circuit « LED » n'en a pas besoin. Voilà, rien de plus excitant à signaler côté assemblage. Le μ C est déjà programmé, mais il peut également l'être par l'utilisateur. Le programme est téléchargeable gratuitement depuis le lien donné en [2].

(120489 – version française : Hervé Moreau)

Liens

[1] www.atmel.com/Images/doc2535.pdf

[2] www.elektor-magazine.fr/120489



Liste des composants

Circuit principal

Résistance :

R1 = 10 k Ω

Condensateurs :

C3 = 100 nF

C4 = 100 μ F

Semi-conducteurs :

IC1 = 78L05

IC2 = ATtiny13-20PU d'Atmel, programmé, 120489-41

Divers :

K1, K2 = barrette à broches, 5 contacts

K3 = support DIL, 2x3 contacts
circuit imprimé 120489-1

Circuit à LED

Résistances :

R1 = 1 M Ω

R2 = 820 Ω

R3, R6 = 4k7 Ω

R4 = 220 Ω

R5 = 10 k Ω

Condensateurs :

C1, C2 = 100 nF

Semi-conducteurs :

D1 = 1N4148

LD1 = LED bicolore jaune/rouge

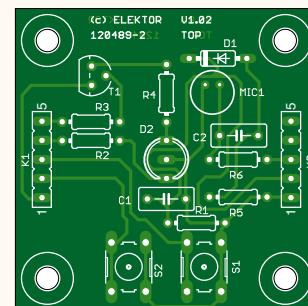
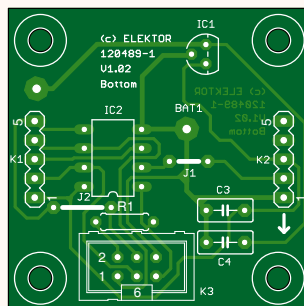
T1 = BC557

Divers :

Mic1 = capsule de microphone à électret

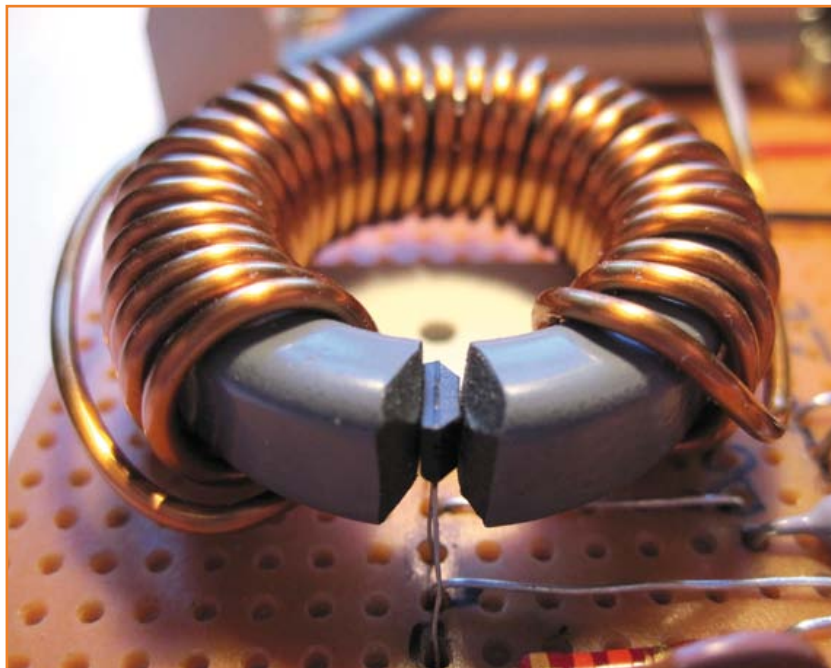
S1, S2 = poussoirs

K1, K2 = embase à 5 contacts
circuit imprimé 120489-2



Wilfried Wätzig
(Allemagne)

limiteur de transitoires



Lors de la mise en marche de certains appareils, tels que des moteurs ou des transformateurs, des causes physiques peuvent donner naissance à des pics de **courant** qui risquent de faire sauter le fusible en amont ou basculer le disjoncteur du moteur, empêchant le dispositif de fonctionner. On évite ce souci avec un limiteur de courant de démarrage qui offre une grande résistance pour absorber les courants transitoires initiaux. Après le démarrage, cette résistance est court-circuitée et l'appareil retrouve ainsi la pleine tension du secteur.

Le circuit présenté ici forme un limiteur d'appel de courant qui, relié directement au secteur, surveille l'afflux de courant vers l'appareil au moyen d'un bobinage et d'un capteur à effet Hall. La fabrication de cette bobine demande un minimum de doigté. Elle se construit sur un tore de ferrite, un FT 114-74 p. ex., d'un diamètre extérieur de 30 mm, diamètre intérieur de 20 mm pour une hauteur de 6 mm. Il faut d'abord pratiquer dans ce noyau une fente à air d'environ 1,5 mm de largeur par découpage, opération suffisamment pointue pour que nous y consacrons une petite illustration dans les Bruits de labo de ce numéro. L'entrefer doit permettre d'insérer parfaitement un capteur à effet Hall TLE4935L (cf. photo). On bobine sur le tore 30 spires de fil de cuivre émaillé de 1,2 mm d'épaisseur. Le courant qui parcourt cet enroulement engendre dans l'entrefer de l'anneau un champ magnétique que détecte le capteur placé à cet endroit. Il s'agit d'une version numérique de la sonde avec trigger de Schmitt et sortie à collecteur ouvert. Lorsque le champ

magnétique dépasse une certaine intensité, le capteur commute sa sortie à la masse. Avec les valeurs données ici, l'opération a lieu quand le courant dans la bobine dépasse 1 A. Comme la surface sensible du capteur à effet Hall, avec son 1 mm², est beaucoup plus petite que les 30 mm² de la section du tore, elle ne peut intercepter qu'une fraction du champ magnétique produit. Un noyau de ferrite de section transversale plus petite serait encore préférable.

Parcourons le circuit. La tension du réseau à basse tension (230 V~) appliquée sur K1 arrive sur le limiteur de courant. L'appareil utilisateur qu'il faut protéger d'un excès de courant d'appel se branche à la sortie, sur K2. Le transformateur TR1, forme avec le redresseur en pont B2 et le régulateur de tension IC2, l'alimentation du circuit de commande, elle fournit du 12 V stable.

L'ensemble de la self L1 et du capteur à effet Hall IC1 mesure le courant dans la charge branchée en



sortie. Aussi longtemps que le relais de puissance ne court-circuite pas les résistances R1, R2 et R3, elles absorbent les pointes de courant.

Tant que le courant dans la bobine ne dépasse pas une certaine valeur, la tension à la sortie du capteur est établie à 12 V environ par le diviseur de tension R4, R6 et R7. La tension sur la base de T1 est alors suffisamment haute pour faire conduire le transistor. Lors de la mise sous tension de la charge et tandis qu'un fort courant parcourt la bobine, le capteur Hall commute et T1 bloque. En raison de la tension alternative du secteur, la sortie du capteur produit un signal rectangulaire à 50 Hz qui est inversé par T1. Le redresseur en pont B1 qui suit se met à charger C5. Avec les valeurs indiquées pour R9 et C5, le délai avoisine 60 ms avant que le Darlington T2 ne conduise et ferme le relais K1. C'est pendant ce temps que les résistances de puissance R1, R2 et R3 absorbent les pics de courant qui se produisent.

La résistance de protection est divisée en trois composants de puissance de 1,5 Ω chacun. Supposons un courant de 20 A : chaque résistance doit alors dissiper momentanément une puissance de 600 W. Un fusible thermique F1 a donc été prévu, il doit être placé en contact thermique étroit avec au moins une des résistances de puissance. Si jamais le circuit de commande connaît un raté, on est alors assuré que l'ensemble du circuit sera mis hors tension. Sachez qu'il ne faut jamais souder un fusible thermique, mais bien le fixer mécaniquement en contact avec l'organe à surveiller.

Attention, il y a danger de mort !

Il faut impérativement empêcher toute possibilité de contact avec ce circuit : il doit donc être placé sous boîtier selon les normes VDE (VDE 0100, VDE 0701). En outre, les conducteurs porteurs de la tension du secteur doivent être maintenus à une distance suffisante du circuit de commande !

(100759 – version française : Robert Grignard)

détecteur de présence capacitif

inusable et indéréglable

Martin Jepkens,
Thomas Scherer et
Daniel Wunsch
(Allemagne)

La réalisation mécanique d'un système fiable de détection de la présence du conducteur sur la plate-forme d'un gyropode n'est pas à la portée du premier venu. En revanche, un détecteur purement électronique, avec capteur de proximité capacitif, ça on sait faire ici. Et une fois qu'il sera au point, il aura bien d'autres usages.

Trois mordus qui travaillaient à leurs propres versions d'un Wheelie, le gyropode d'Elektor, transporteur personnel semblable aux scooters Segway discutaient de la détection de la présence (ou de l'absence) du conducteur sur l'engin. Un dispositif quelconque doit éviter le démarrage sans conducteur et l'obliger à s'arrêter quand le conducteur descend (volontairement ou non). Il doit donc indiquer infailliblement si le conducteur est sur la plate-forme ou pas.

Segway recourt pour cela à des contacts mécaniques actionnés par le poids du conducteur. Les amateurs du monde entier s'engagent dans cette voie, mais pour eux, parvenir à des niveaux de performance et de fiabilité semblables à ceux des systèmes mécaniques industriels n'a rien d'évident. Les contacts prémontés sont chers ou pas étanches, leur mécanisme est instable et souvent assez moche. C'est ainsi que notre trio germain a eu l'idée de détecter la présence du conducteur par méthode capacitive.

Méthodes capacitives

Les premiers essais ont tenté de tirer profit de la flèche de la plaque supérieure et d'actionner un microrupteur lorsque la tôle ployait sous le poids du conducteur. La course d'un microrupteur de bonne qualité est réduite : 1,5 mm à peine. Il faut donc un réglage mécanique de la position du microrupteur. Autrement dit, une galère.

Pour des électroniciens, l'image de la flexion d'une plaque évoque quelque chose. C'est ça, un condensateur ! Couplée à une deuxième plaque fixe, la plaque qui fléchit constitue un condensateur variable parfait. Il suffirait donc de mesurer la variation de capacité lorsqu'une personne monte sur la plaque supérieure ou en descend, et le tour est joué. L'avantage est double, car les indispensables ajustements peuvent être à la fois électroniques et automatiques. L'ajustement automatique est possible si l'on utilise un (petit) microcontrôleur pour la mesure de la capacité. Jusque-là, tout va bien, mais le diable est dans les détails. Une capacité se mesure de différentes façons : celle des plaques pourrait déterminer la fréquence d'un multivibrateur astable (premier procédé détaillé ci-dessous) ; on pourrait aussi mesurer le courant de charge du « condensateur » ; ou encore, utiliser un procédé fiable et éprouvé, légèrement plus complexe et, qui plus est, breveté, mais qui donne des résultats stables et fiables (cf. QTouch ci-dessous).

Le MVA à microcontrôleur de la **figure 1** est très simple : C_x , la capacité à mesurer, est chargée et déchargée via la sortie, broche A, et une résistance R. L'entrée d'un microcontrôleur usuel (broche B) présente un phénomène d'hystérésis de bascule monostable, ce qui simplifie le microprogramme : quand la tension à l'entrée, broche B, est au niveau « bas », alors la sortie, broche A, est au niveau « haut » - quand la tension à l'entrée, broche B, est au niveau « haut », alors la sortie, broche A, est à nouveau au niveau « bas ». L'entrée B du microcontrôleur voit ainsi



un signal rectangulaire dont la fréquence dépend de C_x . Il ne reste qu'à mesurer sa fréquence ou la période du signal à l'aide d'une fonction de temporisation du microcontrôleur et on obtient une mesure de la capacité. Lorsque la plateforme ploie sous poids d'une personne, l'écart entre la plaque haute et la plaque inférieure fixe se réduit. Par conséquent, la capacité et la fréquence baissent. Et ça marche.

Nous avons d'abord essayé avec une variante microcontrôlée d'un multivibrateur astable pour lequel on mesurait la fréquence ou les temps résultants. C'était un premier essai KISS (*keep it simple, stupid*). Simplicité toute théorique que nos multiples expériences infructueuses nous ont conduits à abandonner. La surface de la contre-plaque servant de capteur est d'environ 15 cm² et avec un écart de 1 mm entre les plaques, on atteint seulement quelques dizaines de pF. Pour que la fréquence qui en résulte reste suffisamment basse, la résistance R doit être de l'ordre du mégohm. L'entrée ressemble alors à une entrée ouverte : la moindre interférence crée des brouillages et on perd à la fois en fiabilité et en sensibilité, au point de rendre l'ensemble inutilisable.

Pourquoi réinventer la roue ? La société *Quantum* a breveté sous le nom de *QTouch* un principe fiable. En absorbant cette entreprise il y a quelques années, *Atmel* a mis la main sur *QTouch* et le propose pour ses propres microcontrôleurs. Rien ne s'oppose à ce que nous reprenions pour nous cette technique éprouvée.

QTouch

Dans le terme *QTouch*, le « Q » n'est pas uniquement l'abréviation de *Quantum*, mais aussi le symbole de la charge électrique - et c'est bien ce dont il s'agit dans le système *QTouch*. Le principe consiste à transborder la charge électrique C_x du pavé de détection dans un plus gros condensateur. Le plus étonnant dans l'affaire est sa simplicité ! En théorie, il suffit de remplacer la résistance R de la figure 1 par un condensateur C_L et on arrive au circuit de principe de la **figure 2**.

Le principe est simple : à partir d'une faible capacité C_x , on charge une capacité C_L plus grosse, de façon cyclique, jusqu'à ce que C_L soit pleine. Le tout, en comptant le nombre de cycles nécessaires. Une fois pleine, C_L est déchargée et on recommence. Le **pseudocode** de l'encart illustre ce principe simple. La seule condition imposée au microcontrôleur est qu'il puisse configurer ses broches en entrée ou en sortie en cours de fonc-

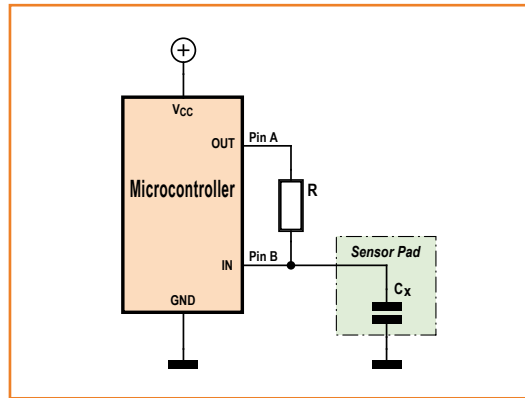


Figure 1.
Circuit de principe d'un multivibrateur astable à microcontrôleur dans lequel la fréquence est fonction de C_x , la capacité du capteur.

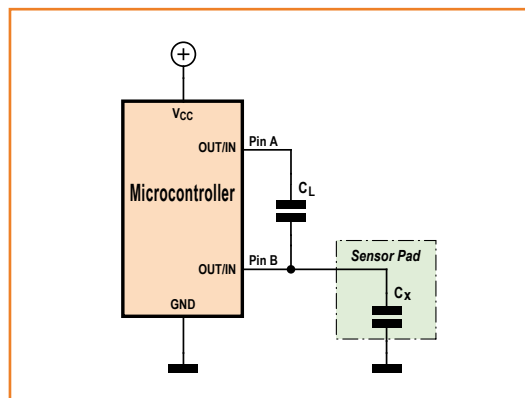


Figure 2.
Circuit de principe d'un *QTouch* avec microcontrôleur. Par rapport à la figure 1, R a été remplacé par C_L et l'entrée est commutée aussi en sortie par intermittence.

tionnement. La capacité C_L étant nettement plus grosse que celle de C_x , l'impédance apparente sur la broche B lors de la mesure du niveau est bien moindre que celle de la figure 1. L'immunité aux parasites du système *QTouch* est donc appréciable.

Pseudo-code QTouch

Au départ, C_L et C_x sont déchargés (ligne 2).

- 1 Pin A et Pin B = out 'Broches A et B en sortie
- 2 Pin A et Pin B = low 'Décharge de C_L et C_x
- 3 cycles = 0 'Initialiser le compte-cycles
- 4 Pin B = in 'Broche B en entrée
- 5 Pin A = high ' C_L est un peu chargé par C_x
- 6 cycles = cycles + 1 'incréméntation du compte-cycles
- 7 If Pin B = low then goto 1 'Si C_L est chargé, recommencer
- 8 Pin A = in 'Broche A passe à haute impédance
- 9 Pin B = out 'Broche B en sortie
- 10 Pin B = low 'On décharge C_x
- 11 goto 4 'Cycle suivant

Ligne 7, si C_L est chargé (pin B = low), la variable « cycles » a atteint une valeur inversement proportionnelle à C_x ($C_x = C_L / \text{cycles}$).

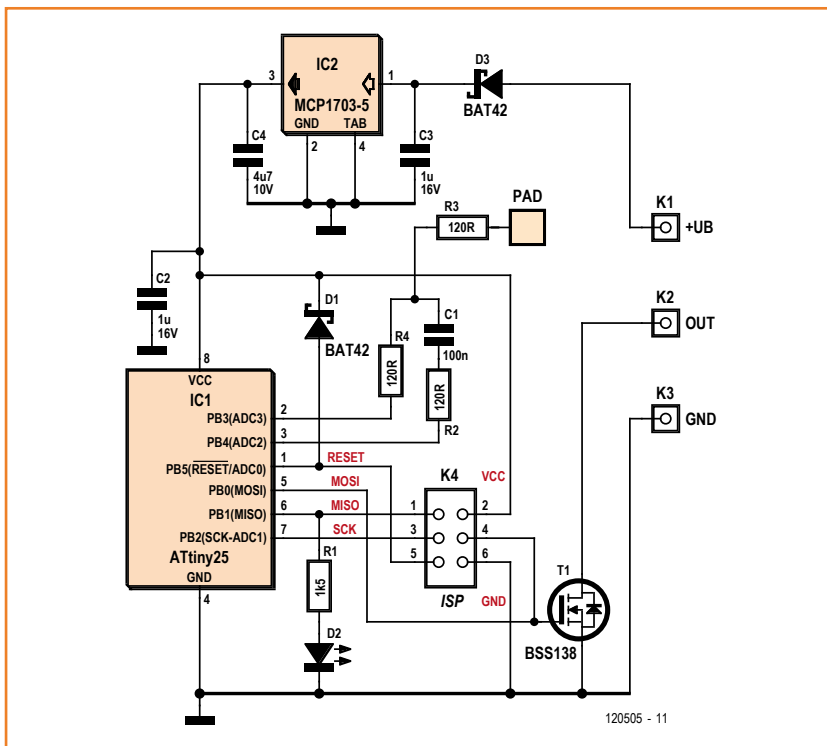


Figure 3.
Le circuit du détecteur de présence capacitif selon le procédé QTouch.

Circuit et montage

Le circuit réel de la **figure 3** n'est pas bien plus compliqué que le principe exposé par la figure 2. Le condensateur C1, dont la capacité de 100 nF est nettement supérieure à celle du pavé de détection, fait office de C_L . Trois autres résistances, R2, R3 et R4, limitent les courants. La LED sur PB1 s'allume lorsque la capacité du pavé dépasse la valeur seuil. Le transistor T1 est commandé en même temps en ouverture via PB0. La sortie à drain ouvert du circuit fonctionne comme contact de travail et peut débiter sans peine ses 100 mA. La diode D1 protège la broche de réinitialisation d'une trop grande tension. Pour que le circuit puisse être monté comme module complet, p. ex. sur le *Wheelie* d'Elektor, on a ajouté d'office une stabilisation de l'alimentation (IC2). La protection contre les inversions de polarité est

assurée par D3. La tension d'alimentation peut ainsi être comprise entre 7 et 14 V.

Regardez, sur la **figure 4**, le prototype monté ! La partie sensible, autour de C1, est entourée de plans de masse. Le pavé, qui fonctionne comme deuxième plaque de condensateur de détection du gyropode, est placé au dos de la carte (**fig. 5**). La raison ? C'est le dos de la carte qui sera tourné vers le marchepied du conducteur. Les autres composants se trouvent sur l'autre face pour qu'ils ne gênent pas la mise en œuvre. C'est pour cette raison que nous avons soudé une version CMS de K4, la barrette de six contacts destinée à la programmation du contrôleur. La forme du pavé n'a pas une importance décisive, mais il est conseillé de lui conserver à peu près la même aire.

Une fois le montage, la programmation et les essais terminés, vaporisez un vernis transparent sur le circuit si vous le posez sur un véhicule. N'oubliez pas de protéger K4 ! Le vernis évitera des incidents dus, par exemple, à la condensation. Ensuite, une fine couche de caoutchouc, d'environ 0,6 mm d'épaisseur, au dos du circuit servira de tampon si la plaque supérieure venait à fléchir sous le poids du conducteur au point de risquer d'endommager le pavé. Sur la face libre, du côté des composants, on en colle une couche un peu plus épaisse (1 mm, p. ex.). Cette face ainsi protégée par son caoutchouc est collée sur un support métallique solidement fixé, en prenant évidemment soin d'orienter vers le haut le pavé de détection. Et le montage fonctionne quasiment comme une butée pour la plate-forme du conducteur.

Variantes et remarques

Commençons par le microprogramme : il a été écrit en Bascom et occupe à peine plus d'un tiers de la mémoire de l'ATtiny25. Les fichiers, source et hexadécimal, sont disponibles gratuitement sur le site d'Elektor [1]. Le principe du programme est le suivant : après l'allumage, la première étape est une mesure de la capacité du pavé pour calibration. En effet, cette capa-

Les auteurs

Martin Jepkens est ingénieur en électrotechnique et directeur de la société ME-Engineering, spécialisée en automatisation de processus industriels. Il construit et optimise actuellement un scooter gyropode durant ses loisirs.

Thomas Scherer travaille depuis plus de 30 ans comme auteur pour Elektor. Il travaille actuellement à un gyropode

de sa conception dont l'ampleur et le niveau de complexité le ravissent.

Daniel Wunsch a développé autrefois des appareils ménagers et travaille aujourd'hui chez un constructeur automobile de renom. Il est le père du gyropode amateur le plus confortable à ce jour.

citée dépend de sa position concrète de montage. On a fixé à 2/3 de la valeur initiale la valeur du seuil à partir duquel le capteur de proximité est considéré par le code comme actionné. Ce seuil peut facilement être modifié dans le programme bien commenté. Une hausse de la valeur seuil rend le circuit plus sensible. Il ne faut tout de même pas le rendre trop sensible s'il doit rester fiable et insensible aux parasites.

On peut évidemment adapter le code à d'autres microcontrôleurs et à de plus petits pavés (p. ex. pour des touches à effleurement). Avec un plus petit pavé, la capacité est plus petite. Il est donc nécessaire de réduire d'autant la valeur de C1 (proportionnellement) de façon à éviter un trop grand nombre de cycles et le dépassement de capacité d'une variable (*overflow*). On pourrait aussi brancher plusieurs pavés à un microcontrôleur. Théoriquement, un tel montage nécessiterait une broche A commune à tous les pavés et autant de broches B que de pavés.

Si le montage est utilisé comme interrupteur à pédale de Wheelie, il en faut deux, un pour chaque pied. Vérifiez également que la plate-forme se déforme suffisamment et qu'elle fléchit d'environ 1 mm lorsque l'on y applique un poids de plus de 15 kg. Cette masse peut sembler faible, mais l'engin doit pouvoir démarrer sans que l'on ait à appuyer dessus de tout son poids. Vérifiez également que le capteur de proximité est bien calibré automatiquement au démarrage. Ne posez pas le pied sur la plate-forme au moment précis de la mise en marche. Le circuit peut aussi fonctionner sans problème en 3,3 V, à condition de monter une version 3,3 V d'IC2.

La technique QTouch est brevetée, mais Atmel, le détenteur du brevet, propose une bibliothèque QTouch libre de droits [2]. L'utilisation de la bibliothèque d'Atmel aurait cependant impliqué l'utilisation du C et rendu impossible la mise en œuvre des contrôleurs les plus petits, disposant de trop peu de mémoire. Le droit des brevets n'est pas applicable à l'utilisation dans le domaine privé. En revanche, nous déconseillons fortement tout usage commercial de notre circuit !

Pour la fixation de la platine, on préférera un collage à une fixation par vis pour que l'espace entre le circuit et la plate-forme reste bien dégagé. Pour d'autres usages, une fixation par vis peut parfois être plus appropriée. On omettra alors aussi la couche de caoutchouc posée sur la face des composants.

(120505 – Claire Dodé)

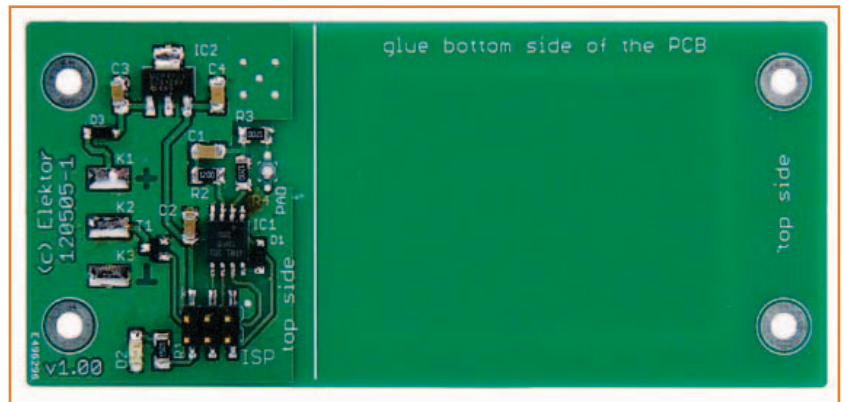


Figure 4.
Le prototype du détecteur de présence capacitif (côté composants).

Liste des composants :

Résistances :

(CMS 1206)
R1 = 1,5 kΩ
R2 à R4 = 120 Ω

Condensateurs :

(CMS 1206, cér.)
C1 = 100 nF/25 V
C2, C3 = 1 µF/16 V
C4 = 4,7 µF/10 V

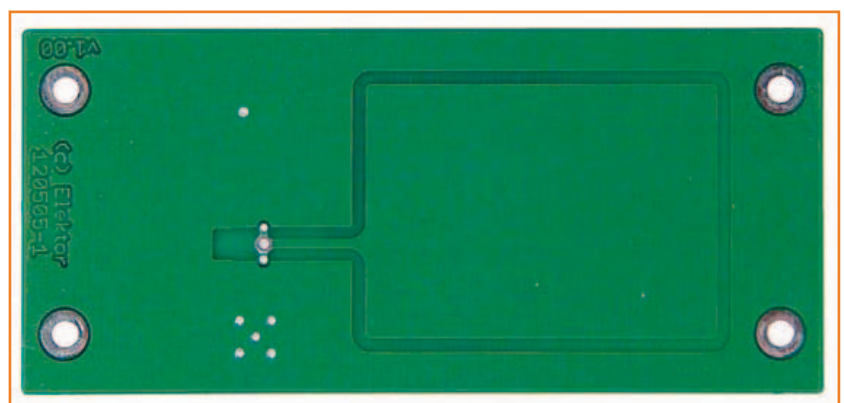
Semi-conducteurs :

D1, D3 = BAT42W, Schottky, SOD-123
D2 = LED rouge, 1206
IC1 = ATtiny25, SOIC-8, programmé 120505-41
IC2 = MCP1703-5, SOT-223
T1 = BSS138, SOT-23

Divers :

K4 = barrette mâle CMS 2x3, au pas de 2,54 mm
circuit imprimé 120505-1

Figure 5.
Dos du circuit imprimé du détecteur capacitif montrant comment la surface sensible (le pavé) est réalisée.



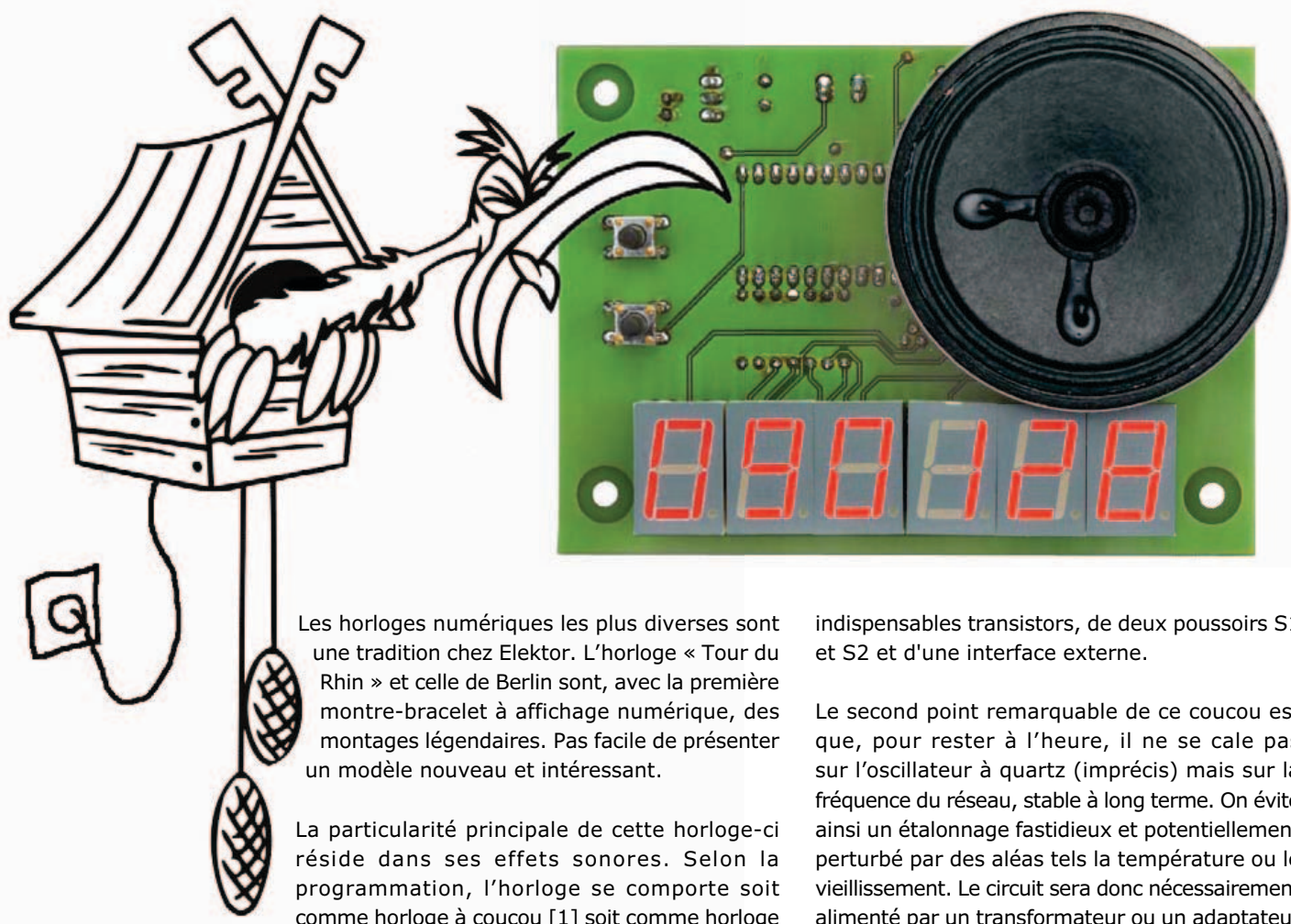
Liens

- [1] www.elektor-magazine.fr/120505
- [2] www.atmel.com/microsite/atmel_studio6/default.aspx

horloge numérique sonore cloche ou coucou ?

Martin Ossmann
(Allemagne)

Encore une horloge ? Celle-ci est certes de construction simple, mais annonce les quarts d'heure et les heures pleines par un carillon, un chant de coucou ou n'importe quel autre son de votre fabrication.



Les horloges numériques les plus diverses sont une tradition chez Elektor. L'horloge « Tour du Rhin » et celle de Berlin sont, avec la première montre-bracelet à affichage numérique, des montages légendaires. Pas facile de présenter un modèle nouveau et intéressant.

La particularité principale de cette horloge-ci réside dans ses effets sonores. Selon la programmation, l'horloge se comporte soit comme horloge à coucou [1] soit comme horloge de clocher. Chaque quart d'heure est alors signalé par un, deux, trois ou quatre coups de cloche, auxquels s'ajoute, aux heures pleines, une autre cloche pour indiquer l'heure du jour.

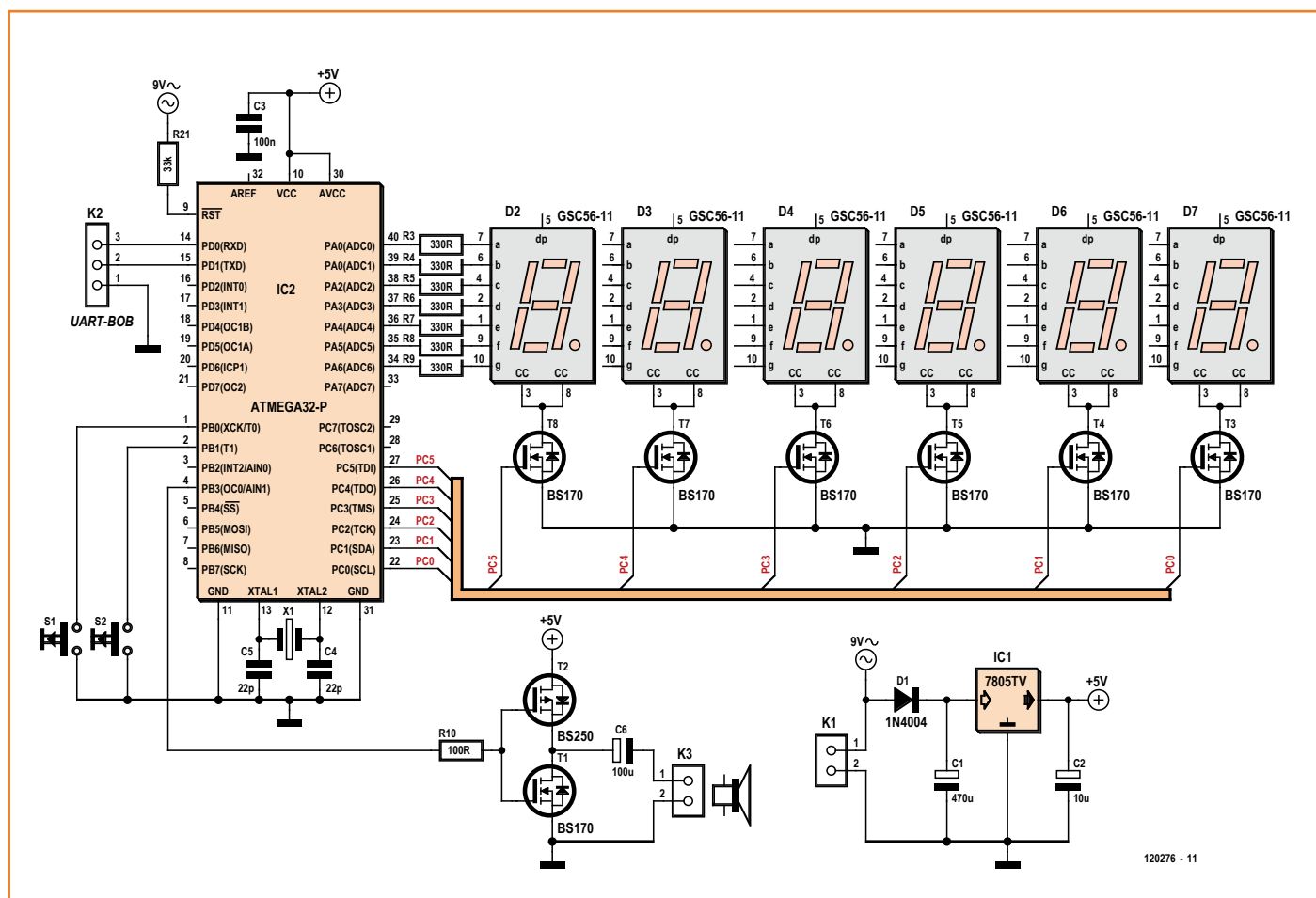
Le circuit

Le cœur de ce coucou (**fig. 1**) est un microcontrôleur ATmega32 d'Atmel, cadencé par un quartz de 16 MHz, et flanqué d'un régulateur de tension 5V (IC1), d'un amplificateur de signal MLI, d'un afficheur à LED à six chiffres avec les

indispensables transistors, de deux poussoirs S1 et S2 et d'une interface externe.

Le second point remarquable de ce coucou est que, pour rester à l'heure, il ne se cale pas sur l'oscillateur à quartz (imprécis) mais sur la fréquence du réseau, stable à long terme. On évite ainsi un étalonnage fastidieux et potentiellement perturbé par des aléas tels la température ou le vieillissement. Le circuit sera donc nécessairement alimenté par un transformateur ou un adaptateur secteur. La tension alternative parvient via R1 jusqu'à l'entrée d'interruption INTO (PD2) du contrôleur.

Le reste de l'alimentation est simple : trois composants seulement. D1 assure le redressement mono-alternance. La tension continue pulsée est en partie lissée par C1, si bien que le régulateur de tension fixe 7805 délivre une tension continue utilisable de 5 V pour le reste du circuit.



Entrées et sorties

L'ATmega est incapable de fournir un courant de forte intensité. Il faut donc, pour piloter le haut-parleur, un étage de puissance constitué simplement de deux MOSFET (T1 et T2), l'un, un BS250, à canal P - et l'autre, un BS 170, à canal N. Le volume sonore obtenu est respectable.

Le format d’affichage de l’heure sur six afficheurs à LED à sept segments est hh:mm:ss. Ici aussi des transistors de type BS170 à canal N (T3 à T8) amplifient le courant des ports du contrôleur. Le courant à travers les segments est limité par R3 à R9.

Les touches S1 et S2 permettent de régler l'heure. À partir du mode normal, lorsque l'on appuie sur la touche S2, on passe en mode de réglage et le chiffre des dizaines de l'heure clignote. Avec le poussoir S1, on peut alors régler ce chiffre. Appuyer à nouveau sur S2 pour passer au chiffre des unités de l'heure, qui se met à clignoter à son tour, et ainsi de suite. On règle

ainsi tous les chiffres les uns après les autres.
Une dernière pression sur S2 repasse l'horloge
en mode normal.

Sons WAVE

Pour la sortie analogique, on utilise Timer0 en mode MLI rapide. La fréquence de la porteuse, dont la largeur d'impulsion est modulée, est de $16 \text{ MHz} / 256 = 62,5 \text{ kHz}$, nettement au-delà de la limite audible. La lecture du son de carillon, à raison de 11 025 échantillons par seconde, est commandée par le Timer1 que l'on fait compter sans cesse jusqu'à 1451 ($= 16 \text{ MHz} / 11025 \text{ Hz}$). Les échantillons qui forment le son de cloche, codés sur 8 bits, sont enregistrés en mémoire de programme dans le tableau sound. Le **listing** donne une idée de la routine d'interruption pour Timer1. Pour lancer la lecture des échantillons sonores, il suffit de mettre à 0 la valeur de `samplePtr`. Le deuxième son de cloche, plus aigu, est obtenu tout bêtement par une lecture accélérée des mêmes échantillons.

Figure 1.
Un microcontrôleur et 6
afficheurs à LED.

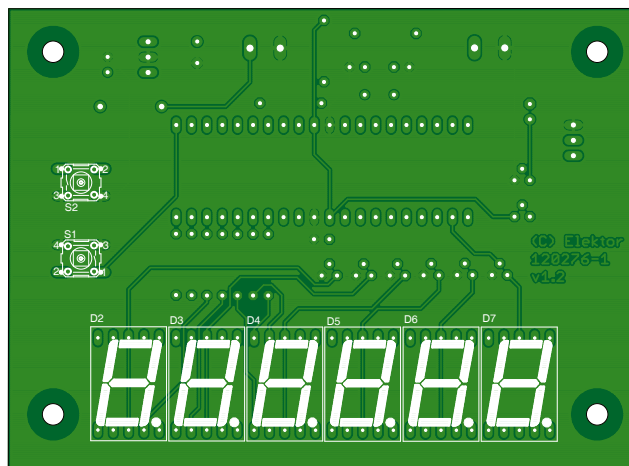
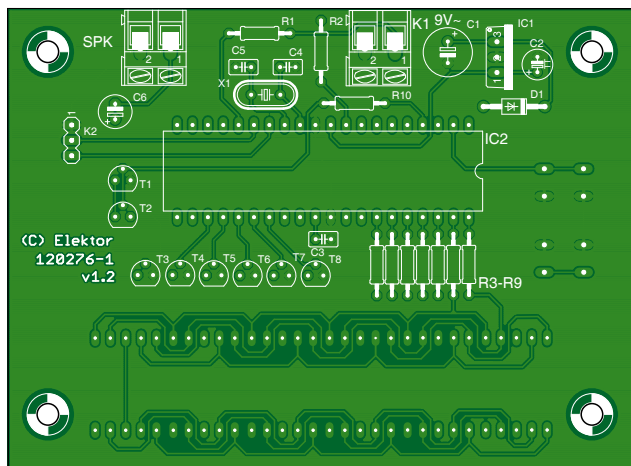


Figure 2.
Implantation du circuit
imprimé double face.
Afficheurs et poussoirs
ne sont pas montés du
même côté que les autres
composants.

L'intérêt principal de ce montage est de vous permettre d'utiliser vos propres sons (coucou, ...). Il suffit de les enregistrer en stéréo à 11 025 échantillons/seconde dans un fichier WAVE qu'un programme en Java convert1.jar permet ensuite de convertir en un fichier d'en-tête (include) à utiliser lors de la compilation du projet de l'horloge. Ouvrez les fichiers batch du programme, ils expliquent comment faire. Pour stocker un son, l'ATmega32 dispose de 24 Ko, ce qui correspond à environ 2,1 s (= 24000/11025).

Interface

Les deux liaisons USART du microcontrôleur, TxD et RxD (PD0 et PD1), sont disponibles sur K2. À ce connecteur d'interface série vous connecterez p. ex. un convertisseur USB/TTL.

Pour un couplage direct avec l'interface RS232 d'un PC, il faut un adaptateur de niveaux.

L'interface a deux fonctions, elle donne l'heure et permet de la régler. On utilisera pour ça un émulateur de terminal (p.ex. *HyperTerminal*) à 19 200 bauds. L'heure est actualisée chaque seconde ; pour le réglage, on entre l'heure souhaitée (en commençant par les heures) et on tape la lettre « s ». Il s'affiche par exemple :

```
Clock 09:01:52 Type 's' to set clock to: 00:00
Clock 09:01:53 Type 's' to set clock to: 00:00
...
```

Si l'on saisit à présent « 0845 » sur le terminal, on obtient :

Liste des composants :

Résistances :

R1 = 10 kΩ
R2 = 33 kΩ
R3...R9 = 330 Ω (ou réseau de 7 résistances en boîtier DIL)

Condensateurs :

C1 = 470 μF/16 V
C2 = 10 μF/16 V
C3 = 100 nF
C4, C5 = 22 pF
C6 = 100 μF/10 V

Semi-conducteurs :

D1 = 1N4004

LD1 à LD6 = afficheur à 7 segments SC56-11SRWA (rouge) (Kingbright)

IC1 = 7805

IC2 = ATmega32P

T1, T3 à T8 = BS170

T2 = BS250

Divers :

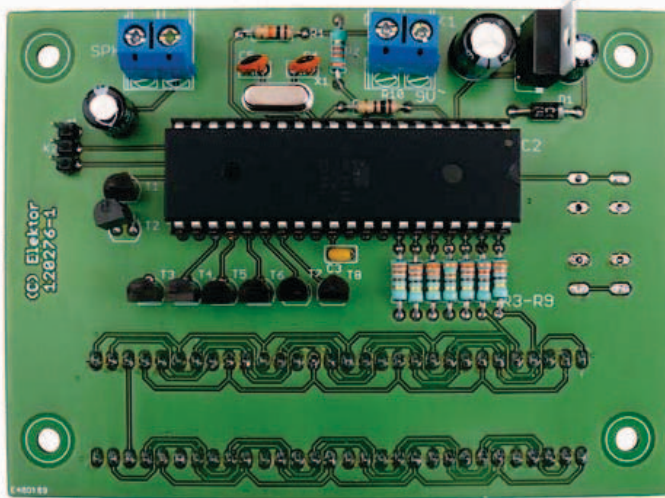
X1 = quartz 16 MHz

S1, S2 = poussoirs

K1, K3 = bornier encartable bipolaire

K2 = embase à 3 points

support CI 40 contacts



Clock 09:01:56 Type 's' to set clock to: 08:45
 Clock 09:01:57 Type 's' to set clock to: 08:45
 ...

disponible gratuitement sur le site d'Elektor [2].

(120276 – version française : Claire Dodé)

Une seule pression sur « s » et la nouvelle heure est fixée :

Clock 08:45:00 Type 's' to set clock to: 08:45
 Clock 08:45:01 Type 's' to set clock to: 08:45
 ...

Liens

- [1] www.fuglar.no/galleri/lyder/Cuculus.canorus.mp3
- [2] www.elektor-magazine.fr/120276

Même si l'ATmega32 est théoriquement programmable *in situ*, cette interface n'a pas été prévue sur le circuit imprimé. Pour programmer le contrôleur (p. ex. pour de nouveaux sons), il faut donc le retirer du support et le placer dans un programmeur.

Un jeu d'enfant

Grâce au circuit imprimé double face de conception claire (**fig. 2**) proposé par Elektor [2], ce coucou numérique est facile à monter. Pas de CMS, tous les composants sont traversants, l'implantation est un jeu tentant.

Les afficheurs et les deux poussoirs sont placés au dos du circuit imprimé.

Si vous voulez pouvoir ajouter de nouveaux sons, le contrôleur devra être monté sur support. L'ATmega32 est disponible préprogrammé auprès d'Elektor, mais vous pouvez aussi le programmer vous-même, puisque le code source en C est

Routine d'interruption

```
prog_int8_t sound []={
137, 138, 140, 135, 138, 124,
.....
..... 129, 128, 124, 127,
} ;

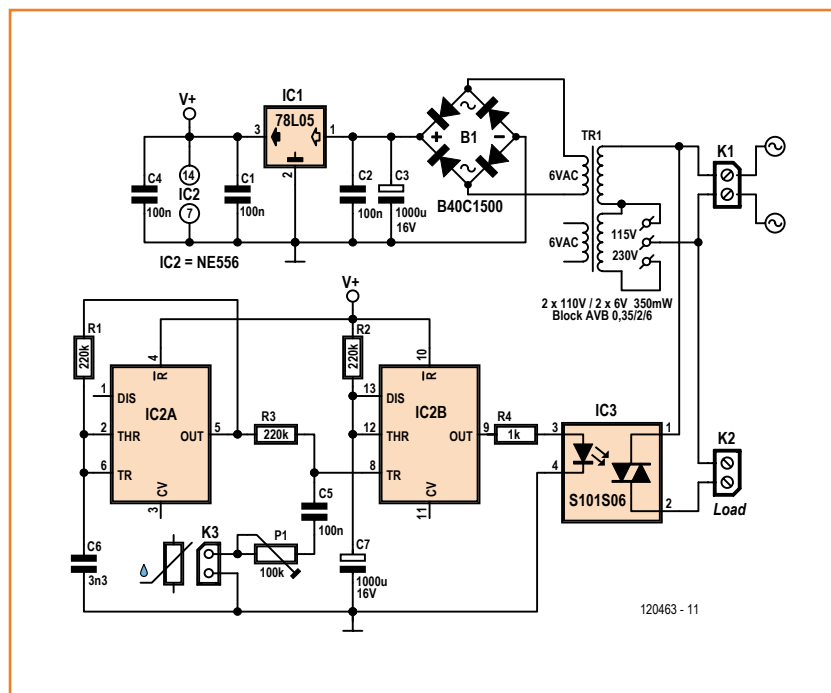
#define Nsamples 18000

ISR(TIMER1_OVF_vect) {
if ( samplePtr<Nsamples){
OCR0=pgm_read_byte( sound+samplePtr);
samplePtr++ ;
}
```


détecteur d'humidité

idéal pour ventiler une salle de bains

Leo van der Linde
(Nouvelle-Zélande)



Ce circuit a été conçu pour mettre en marche un ventilateur de salle de bains aussitôt que l'humidité dépasse un certain seuil. Le circuit de détection consiste en un double temporisateur CMOS, un capteur résistif et un relais.

Le temporisateur IC2A, avec R1 et C6, est monté en oscillateur 1 kHz. Le signal de sortie est appliqué à travers une résistance (R3) au capteur d'humidité relié à K3 et à l'entrée de déclenchement du deuxième temporisateur IC2b. Pour bloquer toute composante continue, le capteur est mis en série avec un condensateur tandis qu'une résistance ajustable permet d'en régler la sensibilité.

En raison de la courbe de résistance exponentielle (<20 kΩ à 75% d'humidité relative, <100 kΩ à 93% d'humidité relative et <150 kΩ à 95% d'humidité relative), la résistance du capteur d'humidité augmente et finit par atteindre une valeur assez élevée pour que le signal de sortie de IC2A déclenche le second temporisateur IC2b, monté en monostable redéclenchable. Et c'est ainsi qu'est mis en marche le ventilateur au moyen du relais. Tant que l'humidité est forte, le ventilateur reste en action. Quand elle diminue, IC2b finit par ne plus recevoir de signal de redéclenchement. Il passe alors quatre minutes avant que le ventilateur s'éteigne.

À la mise sous tension du circuit, le ventilateur tourne pendant 4 minutes puis s'arrête en attendant que le taux d'humidité dans la salle de bain dépasse le seuil que vous aurez réglé.

Pour votre sécurité, l'alimentation est assurée en basse tension par un transformateur, un redresseur, un condensateur et un régulateur, de manière à éliminer tout risque de tension dangereuse sur le capteur.

Grâce au petit circuit imprimé, vous construirez en un tournemain ce circuit qui ne comporte que des composants faciles à manipuler. Vérifiez la position du cavalier de sélection de la tension du secteur.

Liste des composants

Résistances :

R1,R2,R3 = 220 kΩ
R4 = 1 kΩ
P1 = 100 kΩ aj. hor.

Condensateurs :

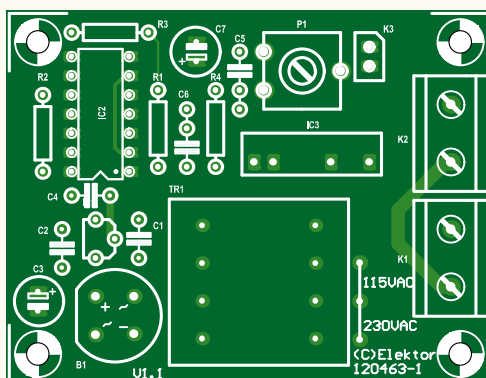
C1,C2,C4,C5 = 100 nF
C3,C7 = 1000 µF/16 V rad.
C6 = 3,3 nF

Semi-conducteurs :

B1 = redresseur B40C1500
IC1 = 78L05
IC2 = NE555
IC3 = relais S101S06V (Sharp)

Divers :

Tr1 = transfo encartable, sec. 2 x 6V, 350 mW (p. ex. Block AVB 0,35/2/6)
K1, K2 = bornier à vis bipolaire au pas de 7,5 mm
K3 = embase à 2 points pour le capteur
capteur Hygrosens SHS A2 (ou autre)
circuit imprimé 120463-1 (voir www.elektor.nl/120463)



Pour pouvoir être utilisé dans une salle de bains, ce circuit doit être monté dans un boîtier étanche au ruissellement. Avec un peu de chance, vous trouverez de la place dans le boîtier du ventilateur. La sonde doit être à l'air libre pour capter directement l'humidité de la salle de bains (si vous faites un trou dans le boîtier pour passer les fils, rebouchez-le au moyen de silicone). Faut-il encore préciser que la tension d'alimentation d'origine du ventilateur doit être appliquée

au connecteur K1, tandis que le ventilateur sera raccordé à K2. Réglez P1 de sorte que le ventilateur s'allume lorsque l'humidité relative est supérieure à environ 90%.

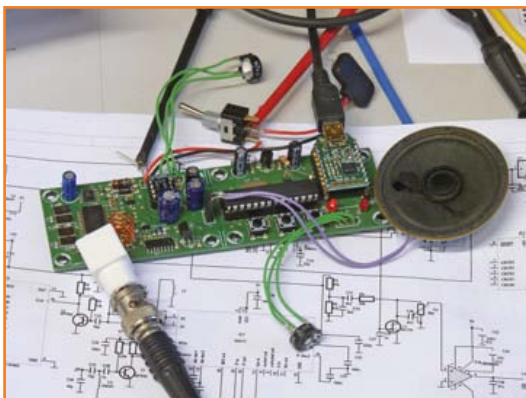
Comment savoir ? Pour une fois, on pourra faire ça... au pif, tout bêtement !

(120463)

antenne extérieure bande aviation pour le scanner VHF

Gert Baars (Pays-Bas)

Cette antenne fouet donne d'excellents résultats, associée au scanner pour la bande aviation civile (108 à 137 MHz) décrit dans ce numéro, p.76. À fabriquer avec du câble coaxial normal, RG58 ou plus fin, le RG174.

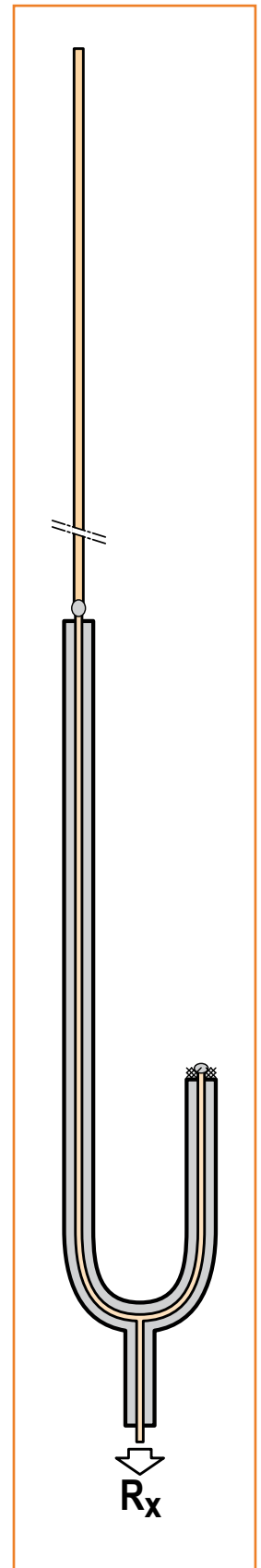


L'antenne se compose d'un brin en $\lambda/2$ dont on transforme l'impédance en 50 Ω à l'aide de deux morceaux de câble coaxial, l'un de 12 cm de long, l'autre de 46 cm. Comptez un centimètre

de plus pour les raccords. Le plus petit morceau est court-circuité à une extrémité en y soudant l'âme au blindage. L'autre bout est connecté en parallèle sur le morceau de câble plus long et celui de liaison vers l'entrée du scanner, où le branchement se fait normalement avec une fiche adéquate. De l'autre côté du long morceau de coaxial, on soude, à l'âme du câble, le brin d'antenne fouet de 117 cm, constitué par exemple d'un morceau de fil électrique monobrin d'un millimètre d'épaisseur auquel on peut encore attacher l'antenne intérieure.

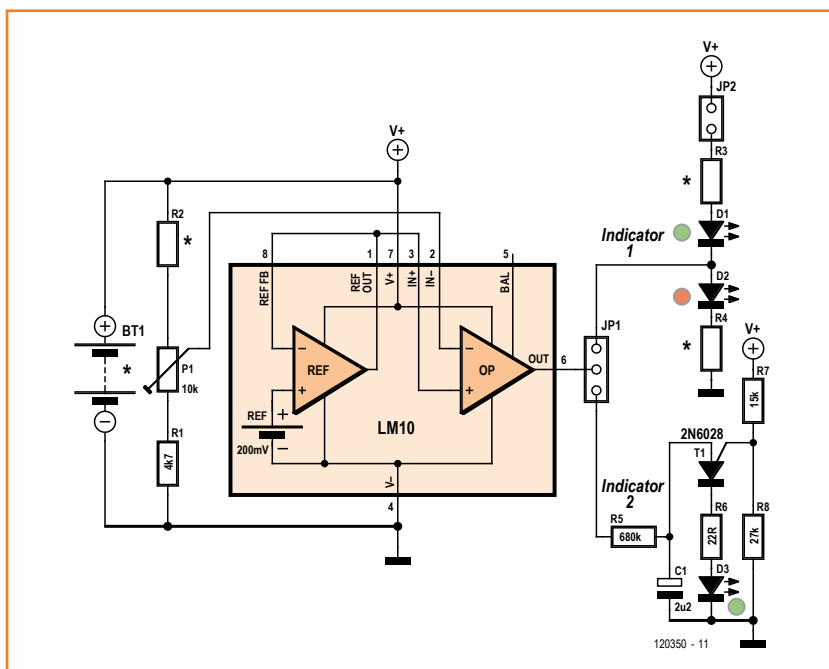
Pour l'usage à l'extérieur, on peut glisser le tout dans un tuyau en PVC ou un bambou.

(10678 – version française : Robert Grignard)



APU : alerte à la pile usée (pas si) rudimentaire et salutaire

J.F. Verrij (Pays-Bas)



Comme son nom l'indique, APU émet un signal quand y a p'us d' jus. C' est un circuit qui ne paye pas de mine, mais qui pourra rendre de grands services s'il est utilisé à bon escient.

On peut régler en le seuil de déclenchement selon la pile ou l'accu utilisé. En résumé, il consiste en une référence de tension, un comparateur et un voyant. Mais ceci mérite un examen rapproché.

Le circuit intégré LM10 offre une source de tension de référence de 200 mV, tamponnée par un amplificateur opérationnel dont la tension de sortie est comparée par un deuxième amplificateur opérationnel du LM10 à une fraction de la tension de la pile surveillée ou de l'accumulateur. La sortie du comparateur (broche 6), haute quand la tension surveillée passe sous le seuil fixé avec P1, commande le voyant, à une ou deux LED selon votre choix.

[lire la suite page 51]

Liste de composants

Résistances :

R1 = 4k7
R2 = cf texte
R3, R4 = cf texte
R5 = 680 k Ω
R6 = 22 Ω
R7 = 15 k Ω
R8 = 27 k Ω
P1 = 10 k Ω aj.

Condensateurs :

C1 = 2 μ 2/16 V

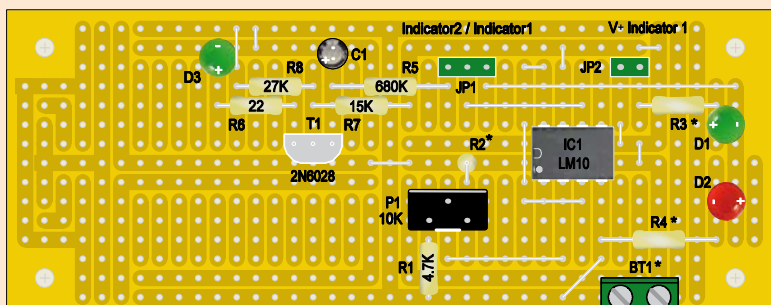
Semi-conducteurs :

D1 = LED verte

D2 = LED rouge
D3 = LED rouge, haut rendement
T1 = 2N6028
IC1 = LM10

Divers :

circuit d'expérimentation Elex



1 bouton + 1 LED = 2 fils seulement

version améliorée

Klaus Jürgen
Thiesler (Allemagne)

Un circuit est d'autant plus fiable que son câblage est simple. C'est en partant de ce postulat qu'en mai 2012 je vous avais montré comment n'utiliser que deux fils pour connecter un poussoir et sa LED témoin, au lieu des trois habituels [1]. À l'époque, j'avais déjà en tête une variante plus économe en énergie, applicable également aux microcontrôleurs qui ne fonctionnent que sous 2,1 V. Voici cette version, avec son courant de repos encore plus faible lorsque le bouton est ouvert.

Resituons l'affaire en deux mots : relier un bouton-poussoir et une LED témoin à un μC nécessite d'ordinaire au moins trois fils, même quand le fil de masse est commun aux deux composants. Avec mon circuit, le besoin en fils est réduit d'un tiers.

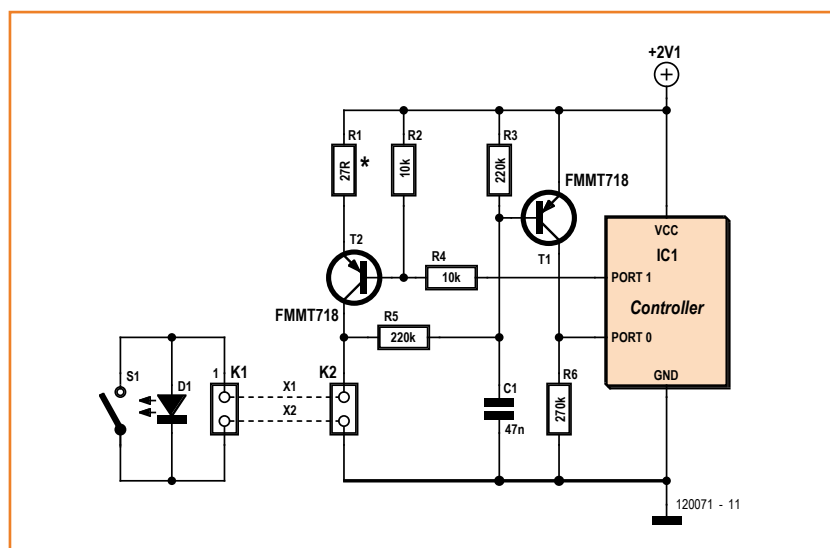
Le truc

L'astuce consiste à faire circuler par la LED, même quand elle doit rester éteinte, un courant de quelques μA : la différence de potentiel aux bornes de la LED est faible, la lumière émise reste imperceptible. Lorsque la LED doit briller, elle est cette fois-ci traversée par un courant de quelques mA, mais la tension à ses bornes n'est guère plus haute. Il y règne donc en permanence une tension comprise entre 1,1 V et 1,7 V. Appuyer sur le bouton met la LED en court-circuit et provoque une excursion de tension que le μC peut exploiter.

Le circuit

Par rapport au circuit du mois de mai, celui-ci utilise deux transistors PNP à très faible tension de saturation U_{CEsat} . Ces composants sont inhabituels, mais bon marché et faciles à trouver, p. ex. chez Reichelt. Le courant de repos (2,5 μA) de la LED passe par R3 et R5. La LED brille lorsque la sortie *Port.1* du μC est au niveau bas. T1 agit alors comme une source de courant constant : la chute de tension aux bornes de R1 vaut 0,3 V. Avec une valeur de 27 Ω pour R1, il passe par la LED un courant d'intensité limitée d'environ 10 mA.

Côté poussoir, R3 et R5 sont dimensionnées de façon à ce que T1 soit conducteur quand un court-



circuit est établi en X1 par S1. *Port.0* est alors placé au niveau haut. Le passe-bas R5-C1 atténue le niveau de bruit sur la ligne X1. Le courant de repos est uniquement le courant qui traverse la LED. S1 ouvert, le circuit ne consomme que 2,5 μA .

Adaptation

De nombreuses applications nécessitent une tension d'alimentation autre que 2,1 V. Les μC fonctionnent sous 3,3 V et moins. Avec une V_{CC} de 1,8 V, prévoyez 8,2 k Ω pour R4, et 160 k Ω pour R5. Sous 3,3 V, utilisez 22 k Ω pour R4, et 470 k Ω pour R5.

Le circuit sera plus efficace avec une tension V_{CC} de 2,1 V et des LED rouges. La tension directe élevée des LED vertes et jaunes réserve leur emploi aux tensions plus hautes. Les LED bleues et blanches ne sont pas recommandées. Certains types de LED dites *low-current* peuvent consommer nettement moins de courant avec une résistance R1 plus grosse, p. ex. 2 mA avec 180 Ω .

(120071 – version française : Hervé Moreau)

[1] www.elektor.fr/110572

CAN & BASCOM-AVR

Mark Alberts (Pays-Bas)

Le compilateur BASIC BASCOM-AVR [1] est ouvert sur le vaste monde, même sur celui du bus CAN (*controller area network*). Conçu pour la communication dans l'automobile, ce bus est utilisé dans bien d'autres branches industrielles. Elektor a publié en avril 2009 [2] son *CANtrôleur pour l'automobile* ; je me suis précipité et j'ai fait l'acquisition de deux modules, car on y trouve un AT90CAN32. Le grand avantage du contrôleur CAN incorporé à un processeur AVR est la légèreté du logiciel et la grande disponibilité du processeur. Le circuit imprimé évoqué est plutôt complet, mais pour les tests, il sera difficile de se passer de port sériel. Elektor propose des interfaces USB/TTL sous forme de câbles (080213-71 [3]) qu'il est possible d'interconnecter avec TX1 et RX1. Avantage supplémentaire, la tension d'alimentation est fournie par l'interface USB. Si vous préférez, au lieu du câble USB-TTL, vous pouvez aussi utiliser un bon vieux MAX232, mais il faudra un adaptateur secteur pour alimenter votre circuit.

Pour ma démonstration, il faut au moins deux cartes, reliée par le bus CAN (K2). J'utilise le programmeur USB-AVRprog d'Elektor [4], mais tout autre programmeur ISP avec un connecteur à 6 broches fera l'affaire.

Selon la configuration standard des fusibles du contrôleur, c'est l'oscillateur interne et le diviseur par 8 qui sont activés. Pour la communication CAN, il faut une horloge bien stable, c'est pourquoi il y a un quartz de 12 MHz sur le circuit. Voici ce que donne le \$PROG BASCOM :

Message in a CAN

Pour faciliter l'utilisation de CAN, quelques commandes CAN spécifiques sont venues enrichir BASCOM. Le traitement du code est effectué au moyen d'interruptions. C'est ainsi que le processeur est le moins sollicité.

Cela nous mènerait trop loin de rentrer ici dans le protocole CAN, je me contenterai donc d'évoquer

Note : Cet article assez condensé suppose chez le lecteur une certaine familiarité avec le sujet.

Code source du programme de démonstration CAN

```
On Canit Can_int          ' define the CAN interrupt
Canreset                  ' reset can controller
Canclearallmobs           ' clear all message objects
Canbaud = 125000          ' use 125 KB

Config Canbusmode = Enabled      ' enabled,standby,listening
Config Canmob = 0 , Bitlen = 11 , Idtag = &H0120 , Idmask = &H0120 , Msgobject = Receive , Msglen = 1 ,
    Autoreply = Disabled          'first mob is used for receiving data
Config Canmob = 1 , Bitlen = 11 , Idtag = &H0120 , Msgobject = Disabled , Msglen = 1 ' this mob is used for sending data

Cangie = &B10110000          ' CAN GENERAL INTERRUPT and TX and RX
Print #2 , "Start"

Do
    If Pinc <> Bdil Then      ' if the switch changed
        Bdil = Pinc          ' save the value
        Bok = Cansend(1 , Pinc) ' send one byte using MOB 1
        Print #2 , Bok       ' should be 0 if it was send OK
    End If
Loop

Can_int:
_can_pageok = Canpage        ' save can page because the main program can access the page too
Cangetints                   ' read all the interrupts into variable _can_mobints
```

ces commandes. Le choix du débit approprié (*baudrate*) est essentiel. C'est *CANBAUD* qui permet de fixer ce paramètre. Veillez à ce qu'il soit le même pour tous les appareils CAN interconnectés. Saviez-vous que c'est parce qu'ils utilisent des débits différents que l'on trouve parfois plusieurs réseaux CAN différents sur une même auto ?

CAN fait appel à des trames de données que le processeur peut traiter comme des objets au moyen de *message objects*. Le AT90CAN32 connaît 15 MOB différents. Selon les caractéristiques attribuées à un Mob, nous déterminerons si nous envoyons ou si nous recevons des données. C'est ainsi aussi que l'on choisit entre des trames de 11 bits ou 29 bits. Chaque MOB pourra être paramétré indépendamment des autres MOB.

Par exemple : Config Canmob = 0 ,
Bitlen = 29 , Msgobject = Receive , Msglen = 8
, Idtag = &H0000 , Idmask = &H0000 ,
Autoreply = Disabled

Le format de MOB numéro 0 est de 29 bits et nous indiquons que nous allons recevoir un message.

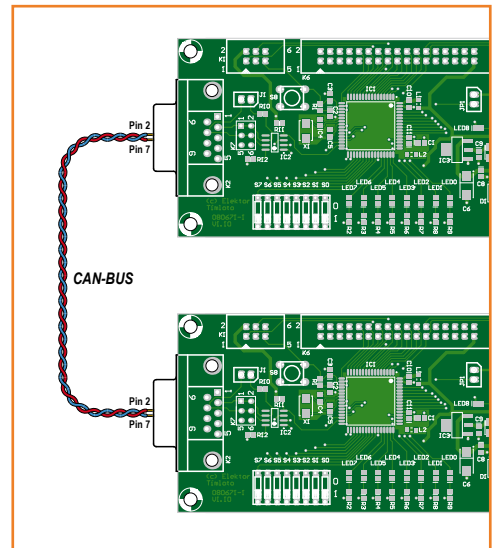
Le message à recevoir ou à envoyer comptera 8 octets au plus. Des données supplémentaires peuvent être envoyées dans d'autres MOB.

L'IDTAG permet de filtrer les ID à recevoir, tandis qu'IDMASK permet de définir une plage. De cette façon, il est possible d'attendre un ID spécifique et rien d'autre. Pendant ce temps, tout autre ID que celui-là sera ignoré. Une telle précaution évite de surcharger le système. Il ne s'agirait pas de laisser interrompre le processus principal par des messages qui ne nous intéressent pas.

Exemple :

IDTAG=&H0123 , IDMASK=&H0123 : filtrer ID &H0123.

IDTAG=&H0123 , IDMASK=&H0120 : filtrer ID &H0120-&H0123



```

For _can_int_idx = 0 To 14                                ' for all message objects
  If _can_mobints._can_int_idx = 1 Then                    ' if this message caused an interrupt

    Canselpage _can_int_idx                               ' select message object

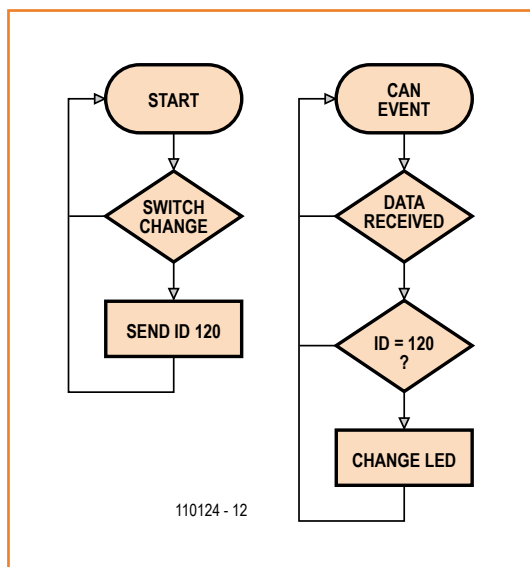
    If Canstmob.5 = 1 Then                                  ' we received a frame
      _canid = Canid()                                     ' read the identifier
      Print #2 , Hex(_canid)

      Breceived = Canreceive(porta)                        ' read the data and store in PORTA
      Print #2 , "Got : " ; Breceived ; " bytes"          ' show what we received
      Print #2 , Hex(porta)

      Config Canmob = -1 , Bitlen = 11 , Msgobject = Receive , Msglen = 1 , Autoreply =
        Disabled , Clearmob = No                           ' reconfig with value -1 for the current MOB
    ElseIf Canstmob.6 = 1 Then                             'transmission ready
      Config Canmob = -1 , Bitlen = 11 , Msgobject = Disabled , Msglen = 1 , Clearmob =
        No                                                  ' reconfig with value -1 for the current MOB and do not set ID and MASK
    End If
  End If
Next
Cansit1 = 0 : Cansit2 = 0 : Cangit = Cangit               ' clear interrupt flags
Canpage = _can_pageok
Return

```


Ordinogramme CAN
message object 0 et 1
 MOB 0 est utilisé pour recevoir des données
 MOB 1 est utilisé pour transmettre l'état des interrupteurs DIL en cas de changement



Lors de l'envoi d'un message, IDTAG est utilisé pour en fixer l'ID. En revanche, lors de l'envoi, IDMASK ne sert à rien.

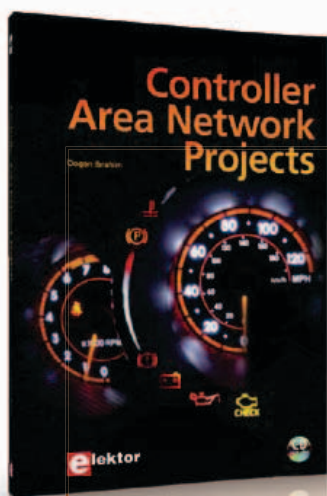
Comme plusieurs MOB différents peuvent être actifs en même temps, ils sont susceptibles de déclencher en même temps une interruption. La

routine d'interruption CAN scrute donc les MOB pour voir lesquels ont demandé une interruption. Si c'est le cas, la fonction *CANRECEIVE* lit le message et active de nouveau le MOB. Important : un MOB qui a servi est remis en circulation. À défaut de quoi on ne recevrait plus de nouveaux messages. Il n'est pas nécessaire de redéfinir tous les paramètres.

La fonction *CANRECEIVE* stocke dans une variable spécifique les données reçues et retourne le nombre d'octets reçus. Par exemple : `received=CANRECEIVE(portA)`

Le code de l'interruption doit être aussi court que possible. Lecture de données et sauvegarde. La lecture d'un drapeau pourra être faite dans le programme principal, de même que le traitement des données. Dans l'exemple imprimé, on relève la présence de commandes PRINT alors que celles-ci, de même que toute autre commande lourde, devraient être bannies de la routine d'interruption.

L'envoi de données se fera avec *CANSEND*. On indiquera quel MOB utiliser, on spécifiera une variable et, facultativement, le nombre d'octets à envoyer.



Ce livre s'adresse aux étudiants, aux ingénieurs en exercice, aux amateurs et à tous ceux qui souhaitent en savoir plus sur le CAN et ses applications.

Ce livre vous enseignera les principes de base des réseaux CAN et il vous montrera comment développer des projets avec des microcontrôleurs qui utilisent le bus CAN.

- Apprendre les principes théoriques du bus CAN utilisé dans l'industrie automobile
- Apprendre les principes, le fonctionnement et la programmation des microcontrôleurs
- Concevoir des projets complets à base de microcontrôleurs programmés en langage C
- Développer des projets avec bus CAN complets et réels
- Apprendre les principes des systèmes OBD utilisés pour déboguer les circuits électroniques des voitures

Vous apprendrez comment concevoir les nœuds du bus CAN avec des microcontrôleurs, créer un bus

CAN, écrire des programmes dans un langage de haut niveau et enfin échanger des données en temps réel sur le bus. Vous apprendrez également à assembler le matériel et à l'interfacer avec des LED, des écrans à cristaux liquides et des convertisseurs A/N.

Pour lire ce livre, il faut posséder quelques connaissances de base en électronique. La compréhension du langage de programmation C est un plus dans les derniers chapitres du livre ; connaître au moins un microcontrôleur de la série PIC sera un avantage, en particulier si vous prévoyez de réaliser les 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 ; ce compilateur supporte les microcontrôleurs répandus comme PIC, dsPIC, PIC24, PIC32 et AVR. En outre cette version spéciale comprend une bibliothèque CAN avancée, avec des fonctions intuitives et faciles à utiliser, pour programmer aisément et confortablement des réseaux CAN.

Exemple :

ok= `CANSEND(0 , ar(1) , 4)` 'envoie 4 octets du tableau AR vers MOB 0

Si l'envoi s'est bien passé, cette fonction retourne un '0'. Le MOB est reconfiguré par la routine d'interruption.

L'Auto-CANtroller est doté d'interrupteurs DIP et de 8 LED. Nous avons composé un programme à charger dans les deux contrôleurs (téléchargeable gratuitement [5], voir aussi l'ordinogramme). Ce programme attend un ID &H0120 avec un octet de donnée. Cette donnée est visualisée par les LED. Puis c'est le tour des interrupteurs DIL dont l'état est lu puis transmis au moyen de ID &H0120. Vous observerez que l'ID est identique. C'est possible puisqu'il est impossible d'envoyer de message au sein d'un même contrôleur.

Changez la configuration des interrupteurs DIL sur une carte et vous verrez sur l'autre carte le changement affiché par les LED et vice versa.

(110124 – version française : Pacôme Tabennèse)

Commandes	
CANRESET	Remise à zéro du contrôleur CAN
CANCLEARALLMOBS	Effacement de tous les objets messages
CANCLEARMOB	Effacement d'un objet message
CANBAUD	Paramétrage du débit
CONFIG CANBUSMODE	Configuration du mode
CONFIG CANMOB	Paramétrage d'objet message
CANGETINTS	Lecture d'interruption
CANSELPAGE	Sélection d'un message
CANID	Lecture ID CAN
CANRECEIVE	Lecture de données
CANSEND	Envoi de données

Liens

[1] www.mcselec.com

[2] www.elektor.fr/080671

[3] www.elektor.fr/080213

[4] www.elektor.fr/080083

[5] www.elektor.fr/110124

alerte à la pile usée

[suite de la page 46] Avec le voyant 1, la LED rouge s'allume quand la pile est presque à plat. La verte fait office de témoin de fonctionnement. Avec le voyant 2, c'est un UJT programmable et une LED à haut rendement qui produisent des éclairs visibles de loin même en plein jour (l'auteur l'utilise sur des modèles réduits télécommandés de bateaux, ça lui évite de se mouiller). Ce petit monde est facile à caser sur un morceau de circuit à pastilles, ou une circuit d'expérimentation ELEX. Sur le prototype photographié, il y a les deux voyants. On n'en utilise évidemment qu'un seul à la fois, choisi au moyen d'un cavalier.

Reste le calcul du diviseur de tension. On part de la tension de seuil de la pile ou de l'accu à surveiller. Pour du LiPo p. ex., c'est 3,3 V par élément. L'accu ne sera pas entièrement déchargé ce qui permettra p. ex. de ramener un bateau au port avant que la tension n'atteigne des valeurs dangereusement basses. Prenons deux cellules LiPo et tension minimale U_{\min} de $2 \times 3,3 \text{ V} = 6,6 \text{ V}$. Avec du NiCd, on descendrait sans risque à 1,1 V par cellule. Avec 6 cellules, cela donnerait aussi $U_{\min} = 6,6 \text{ V}$.

Le calcul de R2 donne ceci :

$$R2 = (48,5 \times U_{\min} - 14,7) \text{ k}\Omega$$

Soit pour $U_{\min} = 6,6 \text{ V}$:

$$R2 = (48,5 \times 6,6 - 14,7) = 305,4 \text{ k}\Omega.$$

La valeur normalisée la plus proche est 330 kΩ. À la place de l'accu ou de la batterie à surveiller, branchez provisoirement une alim de labo qui délivre précisément U_{\min} (6,6 V), puis réglez P1 pour que le voyant s'allume tout juste.

Le calcul de la valeur de la résistance-talon des LED (R3 et R4) est le suivant :

$$R_{\text{série}} = (U_{\text{alim}} - 2) / I_{\text{LED}}$$

Pour simplifier, nous considérons ici que la tension de seuil des LED rouge et verte est uniformément de 2 V, alors qu'en réalité elle n'est pas la même. Avec des LED à haut rendement (3 mA), nous aurions donc 1533 Ω, soit 1,5 kΩ en valeur normalisée.

(120350)

éclairage annulaire à LED

cercle de lumière pour photographie rapprochée

Vincent Himpe (USA)

L'éclairage annulaire est une source utilisée avec les *webcams*, microscopes et appareils photo. Il se monte autour de l'objectif pour fournir un éclairage uniforme quand le sujet est très près. Les sources annulaires sont utilisées pour le portrait ou la photo rapprochée, ou pour éclairer des objets sous une loupe binoculaire. Nous allons construire nous-mêmes un éclairage annulaire puissant à partir d'un kit de composants.



Les éclairages annulaires courants utilisent un tube fluorescent circulaire et se trouvent de ce fait limités à une puissance fixe, sauf à recourir à des gradateurs pour tube fluo, complexes et coûteux. La disponibilité de LED blanches de forte puissance permet d'envisager une meilleure solution. L'éclairage présenté ici met en œuvre un assemblage de 36 LED blanches alimentées par un convertisseur-élévateur à courant constant programmable, conçu spécialement pour les LED.

Anatomie d'une LED

Les LED utilisées dans ce projet sont du type CLA1B de *Cree*, en boîtier PLCC4 à monter en surface (**fig. 1**). Avec leurs dimensions de 3,2 par 2,8 mm seulement, elles peuvent produire jusqu'à 13,0 lumens avec un angle de disper-

sion de 120 °. Le courant direct maximal est de 80 mA, mais nous les exploiterons bien en dessous du maximum.

Pilote

Pour attaquer les LED, nous utilisons un circuit intégré pilote *Micrel MIC3289*, convertisseur-élévateur à modulation de largeur d'impulsion qui délivre une forte tension de sortie à partir d'une faible tension d'entrée (**fig. 2**). La puce mesure le courant tiré de la sortie et corrige sa tension de sortie de façon à faire circuler le courant prévu dans la charge, ici la chaîne de LED. Alors qu'un convertisseur-élévateur usuel produit une tension de sortie fixe, ce composant produit une tension variable pour imposer un courant fixe à travers la charge. Comme les LED sont commandées en

courant, c'est la solution idéale.

Le MIC3289 offre d'autres caractéristiques embarquées, conçues précisément en vue de la commande de LED. Parmi elles, une commande en courant à 16 paliers suivant une échelle logarithmique, proche de celle de l'œil humain qui perçoit la luminosité suivant une courbe logarithmique. La plus grande partie du circuit est intégrée dans la puce, aussi seules une résistance de mesure du courant et une inductance externes sont nécessaires. La puce commute à très haute fréquence (1,2 MHz nominal), ce qui permet à l'inductance d'être minuscule. Une broche de commande numérique permet à l'utilisateur de fixer la luminosité et l'état marche-arrêt du convertisseur-élévateur.

Le convertisseur-élévateur est capable de produire une très haute tension, susceptible de détruire la chaîne de LED si quelque chose se passe mal. C'est pourquoi un détecteur de tension incorporé arrête le convertisseur avant les dégâts. Deux versions du pilote sont disponibles : tension de 16 V ou 24 V en sortie. Comme le seuil de tension directe d'une LED blanche peut s'élever à 3,8 V et que nous en avons six en série, il nous faut une version 24 V. Un coup d'œil rapide à la feuille de caractéristiques nous montre qu'il s'agit du type MIC3289-24YD6. Le MIC3289 capte un courant sous forme d'une tension aux bornes d'une résistance, reliée à la broche d'entrée FB (Feed Back). La sortie maximale correspond à une tension de 250 mV sur cette entrée.

Le *shunt* de 1 Ω produit une tension de 250 mV pour un courant de 250 mA. Partagé entre les six chaînes de LED, il donne environ 42 mA par

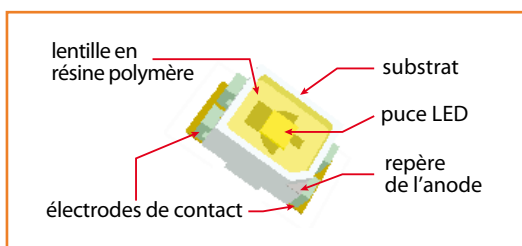


Figure 1.
L'anatomie d'une LED.

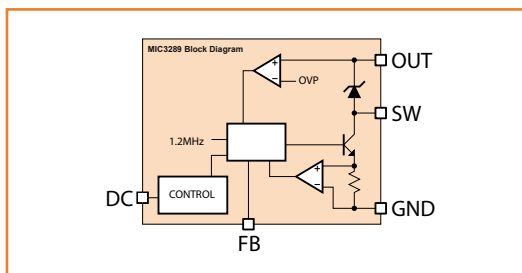


Figure 2.
Les entrailles du circuit intégré pilote de LED.

chaîne, nettement en dessous du maximum de 80 mA spécifié pour les LED. Le MIC3289 est capable de délivrer bien plus de 500 mA, donc vous pouvez jouer avec la luminosité maximale en abaissant jusqu'à 0,5 Ω la valeur du *shunt*. Prenez garde à ne pas amener le MIC3289 à la limite du rabatement thermique. La puce est munie d'un circuit de détection interne pour protéger son transistor commutateur, si bien que vous ne risquez pas vraiment de la détruire.

Microcontrôleur

Le circuit imprimé de la **figure 3** est tiré du livre *Mastering Surface Mount Technology* publié par Elektor. Le circuit intégré U1 se charge de l'élévation de tension, avec l'inductance L1 et le capteur d'intensité R8. La batterie de condensateurs formée par

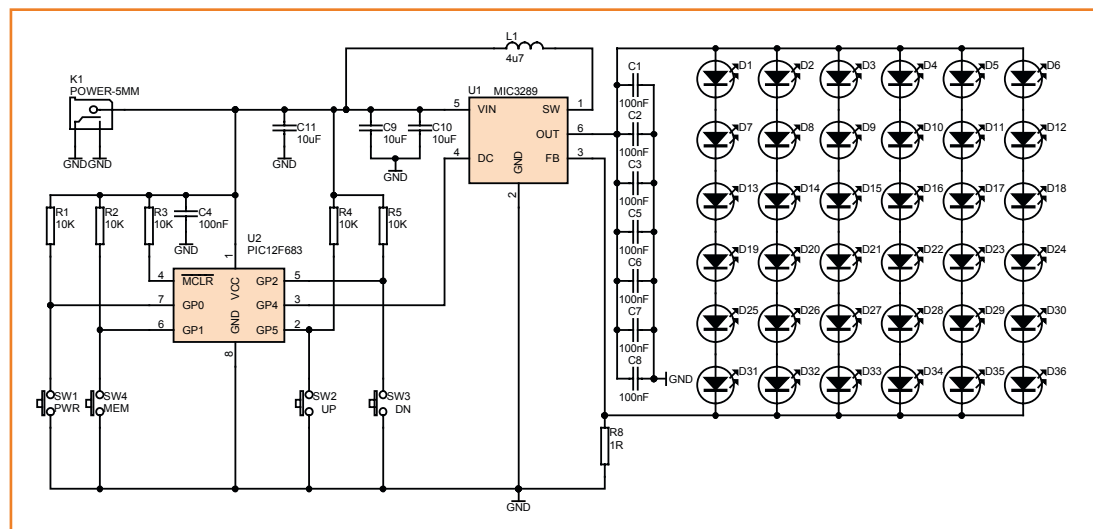


Figure 3.
Le cœur du circuit est U1, les composants qui l'entourent servent « seulement » à commander (U2, SW1 à 4), découpler (C1 à 11), agir (D1 à 36) et compléter (R1 à 5).

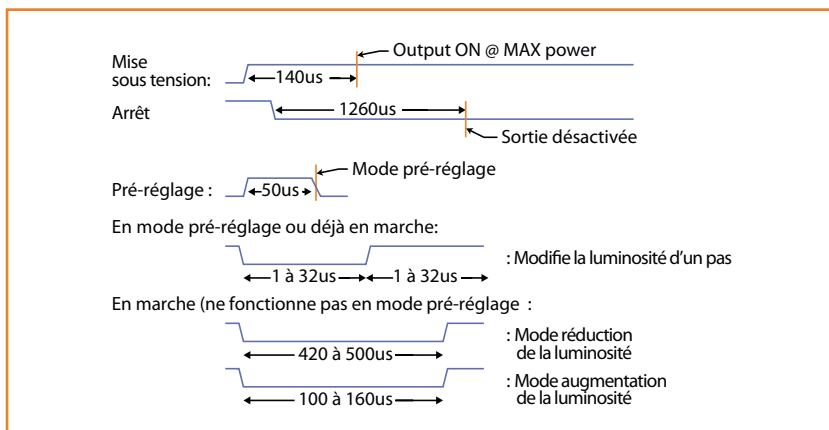


Figure 4.
Chronogramme des
commandes du MIC3289.

C1, C2, C3 et C5 à C8 applique le lissage adéquat à la tension de sortie du MIC3289. Les six chaînes de LED sont simplement connectées en parallèle. Le microprocesseur U2 est utilisé pour produire le train d'impulsions de commande du MIC3289. Quatre interrupteurs avec leurs résistances de polarisation sont reliés aux entrées numériques. La fonction anti-rebonds est assurée par le logiciel. Les boutons permettent le réglage, la commande marche-arrêt et la mémorisation d'un pré-réglage. Chaque fois que les boutons UP et DOWN sont actionnés, l'intensité lumineuse est modifiée

en conséquence. Le bouton STORE sauvegarde en EEPROM la luminosité courante.

Le circuit entier peut être alimenté via le connecteur K1 par un bloc secteur standard 5 V / 1 A.

Synchronisation des instructions

Le protocole de communication est unique en ce qu'il se contente d'un seul fil. Dès que le MIC3289 détecte un changement de niveau sur la broche DC, il démarre un système temporisateur interne. Quand le niveau de la broche DC passe de bas à haut, une temporisation de 140 µs démarre. Si aucun autre changement ne se produit, le MIC3289 lance son régulateur et applique aux LED la puissance maximale, telle qu'elle est définie par le *shunt*.

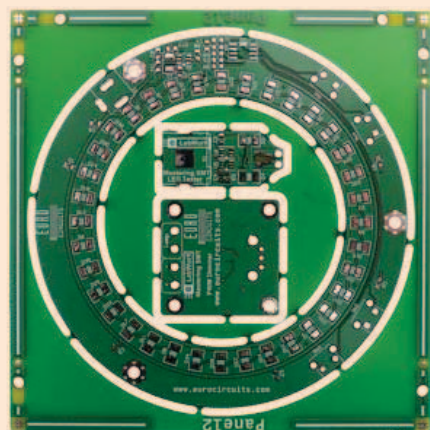
Si le niveau de la broche DC change pendant les 50 premières microsecondes de la fenêtre de 140 µs, le MIC3289 entre en mode programmation. Pour couper la sortie, tirer la broche DC au niveau bas pendant plus de 1260 µs. Cela permet d'utiliser le MIC3289 comme un simple interrupteur marche-arrêt. Passer simplement la broche DC au niveau haut et l'y laisser. Le convertisseur démarrera au bout de 140 µs. Passer la broche au niveau bas arrêtera le convertisseur au bout de 1260 µs. Quand un signal pulsé est appliqué à la broche DC pendant les 50 premières microsecondes après le passage au niveau haut (en supposant qu'elle était au niveau bas auparavant), des commandes plus savantes peuvent être envoyées au contrôleur. Après le passage en mode programmation, un autre temporisateur interne démarre. Si le MIC3289 voit un niveau bas pendant 100 à 160 µs, il l'interprète comme l'ordre d'augmenter la luminosité. Si le niveau bas se maintient pendant 420 à 500 µs, c'est l'ordre de diminuer la luminosité. Chaque impulsion de 1 à 32 µs commande un pas dans la direction programmée. La direction programmée et le niveau de luminosité sont retenus aussi longtemps que la sortie est active.

Si vous coupez l'alimentation du MIC3289, la luminosité est fixée au maximum et le sens de variation à la diminution. Cela permet de programmer un pré-réglage pendant la mise sous tension. Tirez simplement la broche DC au niveau haut et, pendant les 50 premières microsecondes, envoyez un train de zéros logiques pendant une durée maximale de 32 µs avec le nombre de pas que vous voulez commander dans le sens de la descente. Laissez la broche au niveau haut et le MIC3289 commencera à alimenter les LED, 140 µs après la dernière impulsion.

Projet bonus : testeur de LED

Le kit *LED Ring Light* est accompagné d'un petit gadget pratique pour tester la polarité des LED CMS. C'est un petit projet agréable pour tester votre adresse au soudage de CMS. Exercez-vous en appliquant la pâte à braser avec l'écran, en alignant les composants [1] correctement avant de les souder sur la platine.

Projet bonus : gradateur MLI pour ampoule



Le deuxième projet bonus est un gradateur MLI pour charges résistives ou inductives telles qu'ampoules ou moteurs. Il tire profit des compétences acquises avec le testeur de LED. Sa plage de tension d'entrée va de 5 à 24 V et il offre une fonction de démarrage progressif. Un potentiomètre détermine le rapport cyclique. La liste des composants est disponible en [1].

Jusqu'à 460 lumens avec un angle de dispersion de 120 °

Autrement dit (voir la **figure 4**) :
pour activer la sortie :

- Tirez la broche DC au niveau haut pendant plus de 140 μ s.
- Si la sortie est déjà active :
- Tirer la broche DC au niveau bas entre 100 et 160 μ s passe le mode de réglage à AUGMENTATION.
- Tirer la broche DC au niveau bas entre 420 et 500 μ s passe le mode de réglage à DIMINUTION.
- Tirer la broche DC au niveau bas entre 1 et 32 μ s modifie la luminosité d'un pas.
- Tirer la broche DC au niveau bas plus de 1260 μ s désactive la sortie.

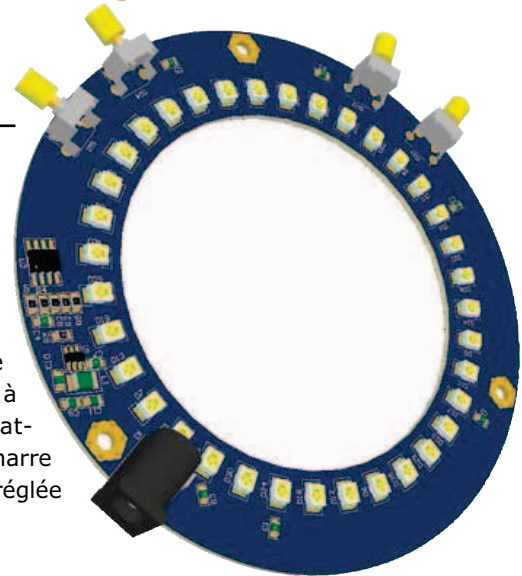
Pour prérégler à la mise sous tension :

Dans les 50 premières micro-secondes après le passage au niveau haut, appliquer le nombre de pas requis (impulsions de 1 à 32 μ s séparés par des temps d'attente de 1 à 32 μ s). Le pilote démarre en mode DIMINUTION, intensité réglée au maximum.

Exemples

Exemple1

Nous voulons la moitié de la luminosité à la mise sous tension. Nous tirons DC au niveau haut et envoyons ensuite huit impulsions de 16 μ s au niveau bas, séparées par des pauses de 16 μ s,



Publicité

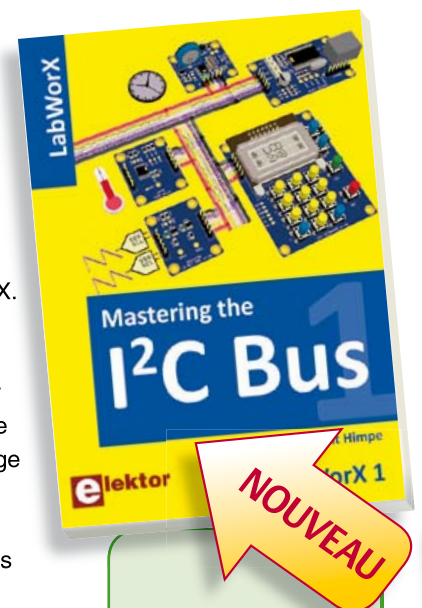
LabWorX 1 Mastering the I²C Bus

➡ Nouveau livre d'Elektor en anglais

Le livre « Mastering the I²C Bus » est le premier opus de la nouvelle collection LabWorX. Il vous emmène explorer le bus I²C (*Inter Integrated Circuit*) et ses applications. Outre le protocole du bus (événements, adressage, synchronisation...), l'auteur s'intéresse également aux applications pratiques et à la conception d'un système entier à bus I²C. En effet il propose un pilote I²C en pseudo-code, c'est-à-dire dans un langage de programmation imaginaire que tout programmeur pourra transposer dans son langage préféré, ainsi que des F.A.Q. sur le matériel et le protocole.

Les catégories de puces compatibles I²C les plus courantes sont traitées en détail (entrées/sorties parallèles, contrôleurs LED & LCD, convertisseurs A/N & N/A, mémoires EEPROM...).

Grâce aux seize mini-circuits (*labsticks*) proposés en complément du livre, le développement d'un prototype sera rapide. La « sonde I²C » permet notamment de commander différents composants I²C depuis un PC (via un port USB). Tous les exemples de programme et les logiciels pour PC nécessaires peuvent être téléchargés de la page internet réservée à LabWorX.



NOUVEAU

248 pages
Format 17 x 23,5 cm
ISBN 978-0-905705-98-9
34,50 €

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

www.elektor.fr/e-choppe

après quoi nous laissons la broche DC au niveau haut. Au bout de 140 μ s, le pilote alimente les LED à mi-intensité. Nous avons choisi la durée de 16 μ s, au milieu de la fourchette des durées admises. Comme le mode par défaut est la mise sous tension à puissance maximale, les huit impulsions en mode DIMINUTION réduisent la luminosité de 8 pas, la moitié des 16 pas disponibles.

Exemple 2

En marche, nous voulons augmenter la luminosité d'un pas. C'est fait par une impulsion de durée comprise entre 100 et 160 μ s, pour passer en mode AUGMENTATION, suivie par une impulsion de 16 μ s, pour sauter un pas.

Exemple 3

En marche, nous voulons diminuer la luminosité d'un pas. Nous appliquons une impulsion de durée comprise entre 420 et 500 μ s pour passer en mode DIMINUTION, puis une impulsion de 16 μ s pour corriger d'un pas.

Microprocesseur

Nous avons choisi pour piloter le MIC3289 un petit microcontrôleur PIC12 en boîtier à huit broches, avec EEPROM incorporée. Le programme scrute les quatre poussoirs et détermine quelles commandes envoyer au MIC3289. Les boutons UP

et DOWN déclenchent l'envoi des commandes appropriées pour régler la luminosité dans le sens désiré. Le microcontrôleur se souvient du mode (AUGMENTATION/DIMINUTION) dans lequel se trouve le MIC3289 et de sa position sur l'échelle, si bien qu'il n'est pas nécessaire de commander le changement de mode à chaque fois (même s'il est plus prudent de le faire).

Le compteur de pas interne « passe par derrière », c'est-à-dire que quand le maximum est atteint il revient au minimum et inversement. Le composant aux commandes (le microcontrôleur dans notre cas) peut l'éviter en tenant à jour la position sur l'échelle et en ignorant les ordres qui entraîneraient un dépassement de l'échelle quand l'un ou l'autre taquet est atteint. C'est ce que fait le micrologiciel mis au point pour ce projet. Le PIC retient la position sur l'échelle et empêche le « passage par derrière ». Le bouton MEM enregistre la luminosité actuelle dans l'EEPROM interne du microcontrôleur. Le bouton PWR bascule entre marche et arrêt. Quand le circuit est activé, le niveau de luminosité stocké dans l'EEPROM est appliqué.

Dessin de circuit imprimé

Un kit de composants complet est disponible pour ce projet (voir encadré), accompagné par quelques bonus. Le circuit imprimé est en forme d'anneau, ce qui permet de le monter autour de l'objectif de l'appareil photo ou du microscope utilisé. Toute l'électronique est installée sous le circuit, sauf éventuellement les boutons. Trois trous de fixation permettent de monter le système sur l'objectif. Vous pouvez tendre des élastiques, avec des colliers en Rilsan ou un gabarit à ressort découpé sur mesures pour les accrocher. Les mêmes trous peuvent être utilisés pour monter, si vous voulez, un diffuseur fait de plexiglas dépoli. L'assemblage de la platine est facile. Commencez par les composants passifs, résistances et condensateurs, suivis par les LED. Ensuite, installez les circuits intégrés et finissez par les boutons-poussoirs traversants et le connecteur d'alimentation.

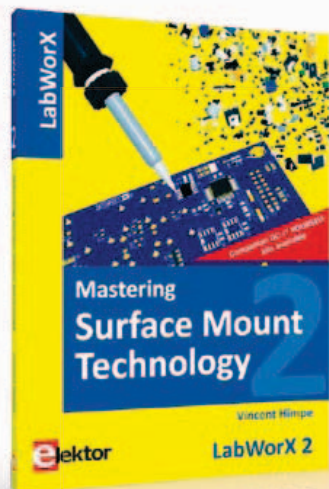
Que la lumière soit et envoyez-nous maintenant des gros plans bien éclairés de vos circuits !

120701 – version française : Jean-Paul Brodier

Liens

[1] www.elektor-magazine.fr/120701

[2] www.elektor.fr/labworx



Livre Mastering Surface Mount Technology

Si vous voulez en savoir plus sur ce projet ou le soudage des CMS, nous vous recommandons notre livre *Mastering Surface Mount Technology*, le deuxième de la série LabWorX d'Elektor. Ce livre vous emmène dans un cours accéléré sur les techniques, trucs et savoir-faire en vue d'introduire la technologie CMS dans vos méthodes de travail. Visitez www.elektor.fr/labworx pour plus d'information.

Kit disponible !

Un kit de composants pour la construction de l'éclairage annulaire est proposé par notre partenaire Eurocircuits. Visitez www.elektor.com/ringlight pour commander votre kit et recevoir deux mini-projets en prime (un gradateur MLI et un testeur de LED). Le kit est fourni complet avec des circuits imprimés de grande qualité, tous les composants, y compris le microcontrôleur programmé et — à titre gracieux pour Elektor seulement — un écran de sérigraphie pour l'application sans histoires de la pâte à braser.

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

Gadgeteer : prototypage rapide façon Microsoft

Clemens Valens
(Elektor.LABS)

Début 2011, *Microsoft* lançait *.NET Gadgeteer*, une plateforme de prototypage rapide exploitant le *.NET Micro Framework* (NETMF). Comme tout système de prototypage rapide, celui-ci doit aider le non-spécialiste à créer ses propres systèmes électroniques en lui masquant la complexité matérielle. À la différence des autres plateformes, *Microsoft* ne vend pour l'heure aucun matériel. Seuls ont été publiés le code source du logiciel et des spécifications détaillées.

Adapter le cadre d'applications NETMF à un matériel reposant sur un microcontrôleur différent selon chaque fabricant de cartes n'est pas une mince affaire, aussi la sortie de la première carte compatible avec les spécifications *Gadgeteer* aura-t-elle demandé du temps. Deux ans plus tard, plusieurs fabricants tiers proposent quelques cartes-mères et bon nombre de modules d'extension.

Gadgeteer ?

Un système *Gadgeteer* est composé d'une carte-mère et d'un certain nombre de cartes d'extension, ou modules. Ceux-ci se connectent à la carte via des *sockets*, en fait des embases mâles renforcées à 10 broches (2x5), et via des câbles en nappe. Ces embases, apparemment identiques, sont toutes différentes : à chaque *socket* est associé un identifiant qui précise le type de module qui peut y être connecté (**tableau 1**). Le type est identifié par une lettre, et un *socket* peut gérer plusieurs types. Le type A fournit par exemple trois entrées analogiques, un port d'entrée/sortie à usage général (GPIO), et trois broches « interdites » (cf tableau). Le type P a trois sorties PWM, deux broches GPIO, et trois broches interdites. Pour tous les *sockets*, la broche 1 reçoit le 3,3 V, la 2 le 5 V, et la 10 la masse. La complémentarité des types A et P permet de les combiner en un *socket* de type AP. Les deux types de *socket* uniquement GPIO, à savoir X (3 broches GPIO) et Y (7 broches GPIO), peuvent être combinés

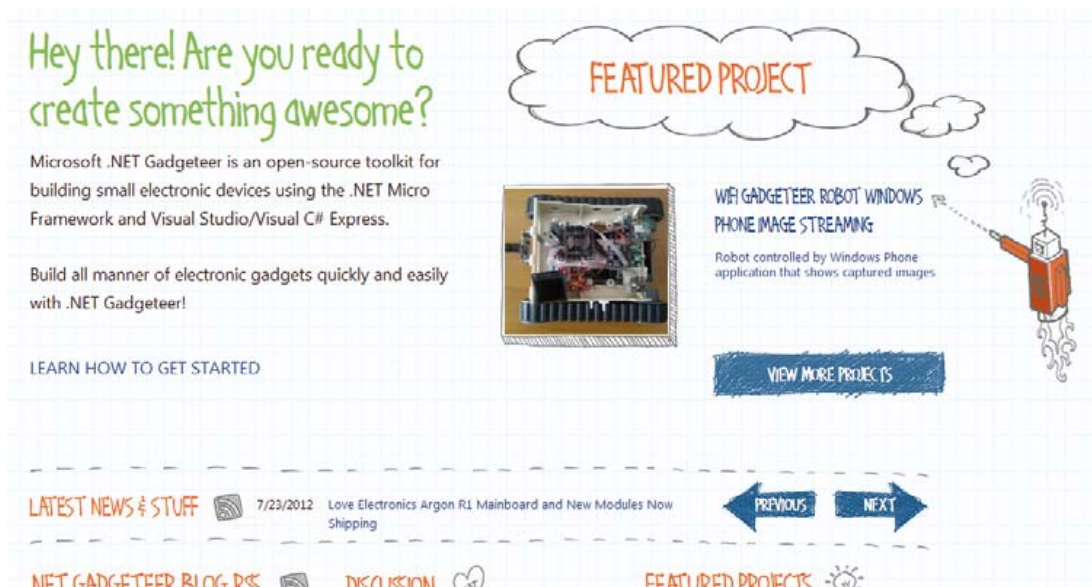
avec la plupart des autres types. C'est pourquoi la majorité des *sockets* sont identifiés par une combinaison de X et/ou de Y, et d'une ou plusieurs autres lettres. La fonction des sept broches disponibles du *socket Z*, appelé *Manufacturer Specific* (MS), n'est pas spécifiée.

Les modules possèdent eux aussi des *sockets* identifiés par des lettres, et ne se connectent qu'à des *sockets* de type identique. Un *socket* de module identifié par un astérisque (*) peut être connecté en chaîne à l'aide du protocole *Daisy-Link*. Cette interface utilise deux bus, un bus I²C, et un bus voisin à un seul fil pour l'initialisation et les interruptions.

Pour augmenter les capacités du matériel existant, les spécifications *Gadgeteer* autorisent aussi l'emploi des *shields*, en gros des cartes filles empilables les unes sur les autres.

Si *Microsoft* n'a pas spécifié de processeur particulier, c'est que la plateforme est censée être indépendante du matériel. Les cartes-mères du moment semblent exploiter des ARM, mais il est probable que d'autres microcontrôleurs prendront le relais. Les différences de bas niveau sont gérées par les bibliothèques de pilotes (DLL) fournies par le fabricant du matériel. Dans l'ensemble, *Gadgeteer* semble s'inspirer de l'architecture PC, si ce n'est qu'à la place d'un PC sous *Windows* nous avons une carte à microcontrôleur sous NETMF, et *DaisyLink* à la place du bus USB. L'utilisateur crée un système en reliant à la carte-mère les modules dont il a besoin. Sa configuration est ensuite facilitée





tée par un outil graphique qui permet de dessiner les modules et leurs connexions. L'outil peut aussi s'occuper du câblage (**fig. 1**) (on regrettera juste qu'il ne sache pas reconnaître automatiquement un système à l'aide d'une caméra). Un clic sur un bouton et *hop*, l'outil produit un *framework* pour le système, y compris les objets logiciels pour toutes les fonctions exportées par les modules (**fig. 2**). L'utilisateur doit bien sûr d'abord installer les pilotes. Une fois le *framework* prêt, il peut alors commencer à programmer. Fin des choses simples.

Common Language Runtime ?

La carte-mère *Gadgeteer* est à ce jour une carte à microcontrôleur assez puissante, dotée d'un processeur à 32 bits, de 390 ko de mémoire de programme, et de 64 ko de RAM (d'après la documentation du *NETMF Porting Kit*). Cette mémoire sert en particulier à l'exécution du *Common Language Runtime* (CLR), la machine virtuelle qui exécute le code du programme utilisateur. S'en sortir avec seulement 256 Ko est possible avec le bien nommé *TinyCLR* — un CLR qui occupe moins de mémoire de programme. De l'espace est

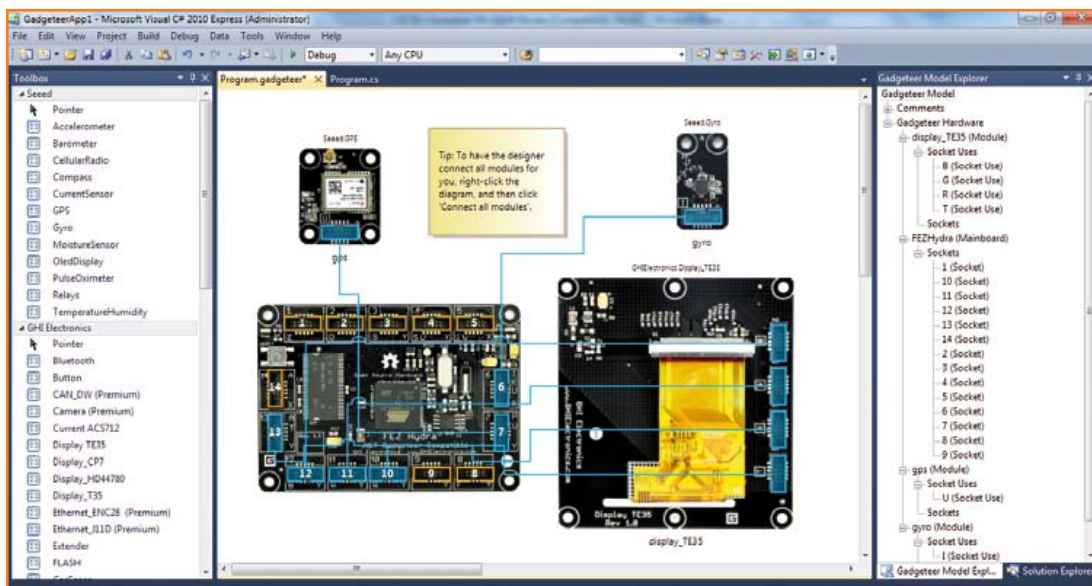
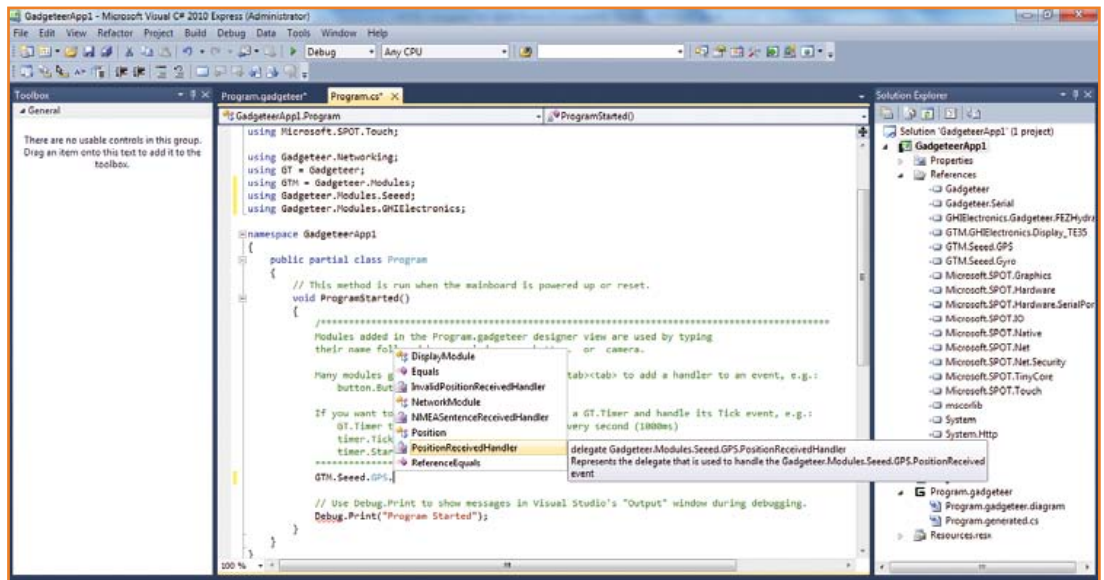


Figure 1.

Le système que j'ai réussi à créer après avoir installé les logiciels nécessaires, les SDK, et agrandi la fenêtre de droite. J'ai laissé le programme connecter les modules. La boîte d'outils de gauche contient les modules installés pour l'occasion ; les sockets disponibles sur la carte sont visibles à droite.

Figure 2.
Le programme créé pour mon système. Espace de noms, classe partielle... saisir tout ça demande une certaine expérience. Lorsque vous ajoutez une commande (ici pour le module GPS), l'IDE propose une liste d'options pour la compléter. Je le redis, avoir un peu de barbe dans ce domaine est indispensable !



toutefois également nécessaire pour l'application utilisatrice, et une carte mère typique disposera donc de 512 Ko de mémoire de programme, et de plus de 64 Ko de RAM. Le *Porting Kit* (PK) cité plus haut permet d'adapter le CLR à un matériel maison, mais malgré les exemples fournis l'exercice reste réservé aux audacieux. Quant aux aventuriers, ils pourront tenter d'être (parmi) les premiers à porter le CLR sur un μ C non pris en charge. Actuellement le PK gère les ARM, les *Blackfins* d'*Analog Devices*, ainsi que les architectures SH-2 de *Renesas*. Le CLR ressemble assez à un système d'exploitation, mais il s'agit en réalité d'une sorte d'interpréteur de commande qui compile à la volée le code de l'application utilisatrice. Le CLR prend

en charge les besoins des applications en temporisateurs, gestion de la mémoire et des exceptions, *multi-threading*, etc., d'où des programmes compacts. Un programme qui recourt à ces fonctions est dit géré (*managed*) par le CLR. Un programme autonome est lui qualifié de non géré (*unmanaged*). *Microsoft* revendique une vitesse de traitement d'environ 550 appels de méthodes gérées par MHz de fréquence d'horloge du μ C (méthode signifie ici « fonction »). Difficile de traduire cette unité en quelque chose d'intuitif, par exemple en fréquence maximum de changement d'état des broches.

Le CLR du NETMF ne comprend pour le moment que les programmes écrits en C# (qui se prononce « si sharp » et dont les fichiers ont pour extension .cs). Coder en C# revient à coder en Java ou en C++ si l'on considère que ces langages sont tous orientés objet. Ceux qui ne maîtrisent pas la programmation objet ne seront sans doute pas emballés par cette restriction. Comme outil de développement, téléchargez gratuitement Visual C# 2010 Express de *Microsoft* (j'ignore si la version 2012 est supportée). Il vous faudra également installer le SDK .NET, ainsi que *.NET Gadgeteer Core* et *.NET Gadgeteer Builder Templates*. Vous devrez aussi installer les pilotes propres à votre matériel.

Écrire l'application pour le système utilisateur est relativement facile, même si nous sommes loin de la simplicité du paradigme initialisation/boucle d'Arduino (voir encadré). Citons le site de Gadgeteer : « *.NET Gadgeteer utilise .NET Micro*

Figure 3.
Microsoft a rédigé des spécifications détaillées pour le matériel *Gadgeteer*. Les angles de la carte doivent être arrondis et les trous de montage sont bien définis.

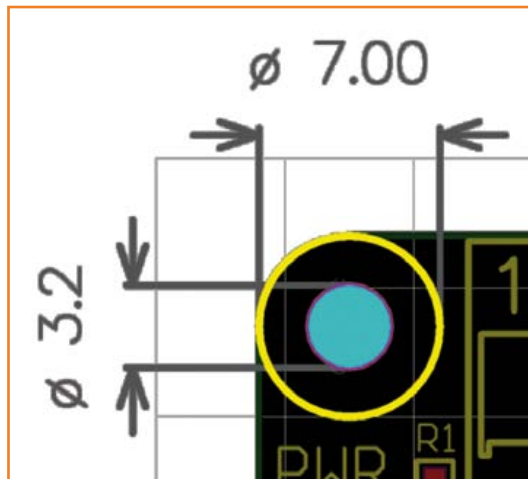


Tableau 1. Types de socket Gadgeteer.

Légende :

GPIO Broche d'entrée/sortie à usage général ; fonctionne sous 3,3 V.

(G) Broche pouvant aussi être utilisée comme GPIO, en plus d'une autre fonction.

(OPT) Type de socket supporté en option par une carte mère ou un module.

[UN] Un module ne doit pas être relié à cette broche s'il utilise ce type de socket.

[MS] Broche spécifique à un fabricant.

! GPIO gérant interruption et forçage logiciel au niveau haut.

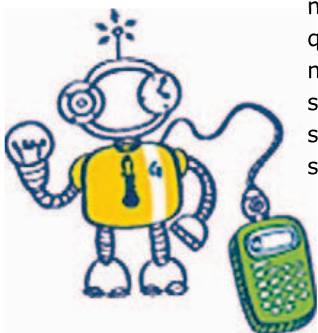
type	lettre	br. 1	br. 2	br. 3	br. 4	br. 5	br. 6	br. 7	br. 8	br. 9	br. 10
3 GPIO	X	+3,3V	+5V	GPIO!	GPIO	GPIO	[UN]	[UN]	[UN]	[UN]	GND
7 GPIO	Y	+3,3V	+5V	GPIO!	GPIO	GPIO	GPIO	GPIO	GPIO	GPIO	GND
Analog In	A	+3,3V	+5V	AIN (G!)	AIN (G)	AIN	GPIO	[UN]	[UN]	[UN]	GND
CAN	C	+3,3V	+5V	GPIO!	TD (G)	RD (G)	GPIO	[UN]	[UN]	[UN]	GND
USB Device	D	+3,3V	+5V	GPIO!	D-	D+	GPIO	GPIO	[UN]	[UN]	GND
Ethernet	E	+3,3V	+5V	[UN]	LED1 (OPT)	LED2 (OPT)	TX D-	TX D+	RX D-	RX D+	GND
SD Card	F	+3,3V	+5V	GPIO!	DAT0	DAT1	CMD	DAT2	DAT3	CLK	GND
USB Host	H	+3,3V	+5V	GPIO!	D-	D+	[UN]	[UN]	[UN]	[UN]	GND
I2C	I	+3,3V	+5V	GPIO!	[UN]	[UN]	GPIO	[UN]	SDA	SCL	GND
UART+ Handshaking	K	+3,3V	+5V	GPIO!	TX (G)	RX (G)	RTS	CTS	[UN]	[UN]	GND
Analog Out	O	+3,3V	+5V	GPIO!	GPIO	AOUT	[UN]	[UN]	[UN]	[UN]	GND
PWM	P	+3,3V	+5V	GPIO!	[UN]	[UN]	GPIO	PWM (G)	PWM (G)	PWM	GND
SPI	S	+3,3V	+5V	GPIO!	GPIO	GPIO	CS	MOSI	MISO	SCK	GND
Touch	T	+3,3V	+5V	[UN]	YU	XL	YD	XR	[UN]	[UN]	GND
UART	U	+3,3V	+5V	GPIO!	TX (G)	RX (G)	GPIO	[UN]	[UN]	[UN]	GND
LCD 1	R	+3,3V	+5V	LCD R0	LCD R1	LCD R2	LCD R3	LCD R4	LCD VSYNC	LCD HSYNC	GND
LCD 2	G	+3,3V	+5V	LCD G0	LCD G1	LCD G2	LCD G3	LCD G4	LCD G5	BACK-LIGHT	GND
LCD 3	B	+3,3V	+5V	LCD B0	LCD B1	LCD B2	LCD B3	LCD B4	LCD EN	LCD CLK	GND
Manufacturer Specific	Z	+3,3V	+5V	[MS]	[MS]	[MS]	[MS]	[MS]	[MS]	[MS]	GND
DaisyLink	*	+3,3V	+5V	GPIO!	GPIO	GPIO	[MS]	[MS]	[MS]	[MS]	GND

Framework pour rendre l'écriture du code de vos systèmes aussi facile que l'écriture d'une application pour un environnement de bureau, le Web, ou Windows Phone. » (Hé, à quand remonte l'écriture de votre dernière appli Windows Phone ?) Dès le départ, l'utilisateur est confronté à un programme par défaut plutôt imposant, auquel doivent être ajoutés le code d'initialisation et les gestionnaires d'évènement. Rien à dire si on est ingénieur logiciel, mais le concepteur de matériel lambda se sentira sans doute un peu perdu au début. Cela dit, avec un peu de ténacité, chacun

devrait pouvoir déployer rapidement une application simple sur son système. Eh oui, le Gadgeteer sérieux se doit de « déployer une solution gérée sur son système » plutôt que de « re-flasher sa carte à tout-va » comme n'importe quel bidouilleur de base.

Conclusion

Microsoft cherche à créer un écosystème fondé sur son NETMF qui engloberait PC, téléphones tactiles et cartes à microcontrôleurs, et assez séparé du matériel pour que le développeur



logiciel soit à l'aise sur n'importe quel système. Objectif légitime et cohérent, d'ailleurs visé par d'autres sociétés. *Gadgeteer*, plateforme de prototypage rapide exploitant NETMF, s'inscrit dans cette logique. Le développeur logiciel peut néanmoins aussi se tourner vers *Linux* pour l'embarqué, ou *Android*. Les sockets en sont absents — même s'il est facile de les ajouter — mais l'écosystème est beaucoup plus vaste, et les cartes s'y comptent ici par centaines, voire par milliers, sans parler de quelques téléphones tactiles vraiment bien conçus.

Estampillée « technologie Microsoft », *Gadgeteer* attirera probablement un public assez nombreux, et il sera intéressant de suivre le

rapprochement entre *Gadgeteer* et *Windows Phone*. Pour l'heure, à en juger par la faible activité des forums NETMF et *Gadgeteer*, le lancement paraît poussif : quelques sujets seulement au moment d'écrire cet article (début 12/2012), un dernier message vieux d'un mois, et des *news* datées de juillet 2012...

(110738 — version française : Hervé Moreau)

Liens

- [1] Page d'accueil de *Gadgeteer* : www.netmf.com/gadgeteer
- [2] .Net-MF pour électroniciens : www.elektor.fr/120033

Arduino contre Gadgeteer

Toutes deux plateformes de prototypage à code source ouvert destinées aux non-spécialistes, *Arduino* et *Gadgeteer* sont de vraies fausses jumelles. D'abord le public visé : *Arduino* vise les utilisateurs pas forcément électroniciens et/ou programmeurs, tandis que *Gadgeteer* a pour cible les programmeurs sans compétences *hardware* ; d'où une programmation simplifiée autant que possible côté *Arduino*, et un environnement de développement classique indispensable pour programmer *Gadgeteer*. Différence marquée côté matériel également. *Arduino* a été conçue pour que tout amateur puisse construire sa propre carte à l'aide des schémas de référence publiés sous licence *Creative Commons*, tandis que *Gadgeteer* se contente de spécifier la conception de sa carte — ce qui se trouve dessus est laissé au fabricant — tout en imposant qu'elle puisse exécuter NETMF.

Autre différence intéressante, le paradigme de programmation. Si, sans s'en vanter, *Arduino* recourt comme *Gadgeteer* aux techniques de la programmation orientée objet, l'approche entre les deux diffère : boucle d'attente mono-thread vs traitement *multi-thread* des événements. Un programme *Arduino* (appelé *sketch* ou croquis) exécute une boucle sans fin pour contrôler la carte. Utiliser les interruptions pour détecter et gérer les événements est bien sûr possible, mais peu y recourent. Pas de boucle dans un programme *Gadgeteer* (appelé *solution*), le programme principal (le *thread*) peut être en attente pendant que les événements sont traités par les gestionnaires d'interruption.

Arduino est adapté aux applications de contrôle simples, dénuées d'interactions utilisateur compliquées, tandis que *Gadgeteer* convient mieux à des systèmes plus complexes, pilotés par menus. *Arduino* et *Gadgeteer* sont de fait des plateformes plus complémentaires que concurrentes.

Né avec des lauriers sur la tête ou des chrysanthèmes aux pieds ?

.NET Micro Framework (NETMF) tire son origine de SPOT (*Microsoft's Smart Personal Object Technology*), un projet initié en 2003 pour rendre polyvalents les appareils électroniques grand public. Les références à SPOT sont visibles dans les programmes NETMF. SPOT exploitait les services réseau *MSN Direct* pour envoyer par ondes radio FM des données à des périphériques du type montre, unité de navigation GPS, ou même machine à café. Les informations provenaient de chaînes payantes : bulletin météo, horoscope, cotations boursières, etc. La réception de SMS en provenance de *Windows Live Messenger* était aussi possible. Dépassé par d'autres technologies, le service *MSN Direct* a été fermé en janvier 2012.

Les périphériques *Windows SideShow* utilisent eux aussi NETMF. Couplée au *Volet Windows* (et à ses gadgets, un lien avec *Gadgeteer* peut-être ?), la technologie *Windows SideShow* pilote divers dispositifs auxiliaires d'affichage reliés à un PC *Windows*. Ces dispositifs peuvent alors accéder à des informations ou médias, même (et surtout) PC éteint. Le 10 juillet 2012, *Microsoft* a malheureusement invité les utilisateurs des versions Vista et 7 à désactiver le *Volet Windows*, parce que certains gadgets peu sûrs peuvent exécuter un code malveillant...

[Source: Wikipedia]

Spécial: Économisez 50% Spécial: Économisez 50%

Fêtez le 25ème Anniversaire de *Circuit Cellar*



\$25 Papier ou Numérique :: **\$50** Les deux éditions

Fêtez le 25ème Anniversaire de la date depuis laquelle *Circuit Cellar* propose à ses lecteurs une analyse éclairée de la technologie à la base de l'électronique enfouie.

Allez rendre visite à www.circuitcellar.com/el912 pour bénéficier de ces superbes offres.

OFFRE BONUS! OFFRE BONUS! OFFRE BONUS! OFFRE BONUS! OFFRE BONUS!

Abonnez-vous aujourd'hui et recevez également, outre votre abonnement, **le Numéro Spécial du 25ème Anniversaire!**



Flowcode 5 pour concevoir

FLOWCODE5

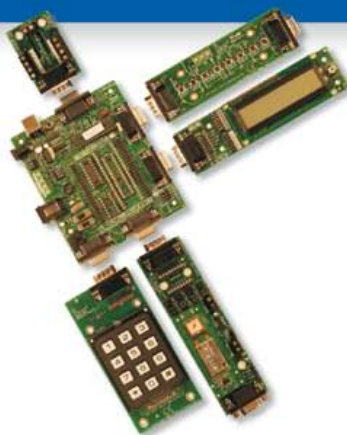


Flowcode 5 est l'un des langages de programmation graphique pour microcontrôleurs (PIC, AVR, ARM et dsPIC/PIC24) les plus avancés au monde. Son avantage principal est de permettre la création de systèmes électroniques et robotiques complexes même si l'on manque encore d'expérience.

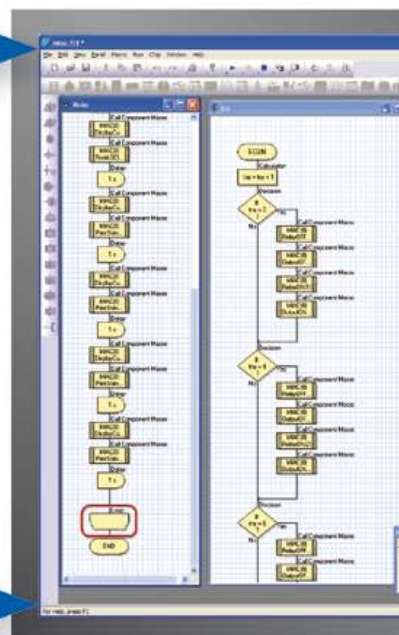
Flowcode est utilisé :

- 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.

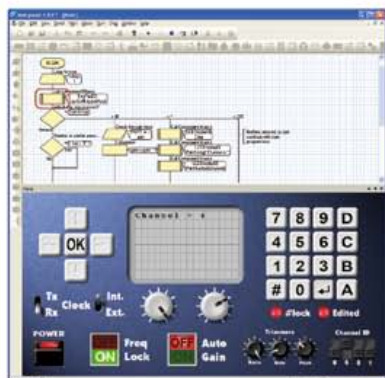
Outil de débogage FlowKit

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

Qu'est-ce c'est qui mijote là ?

... ça sent un peu le brûlé, c'est normal

Transfo d'isolation pour 115 V et 230 V ; l'empreinte reste la même pour des modèles avec tension ou courant de sortie différents. C'est pratique.

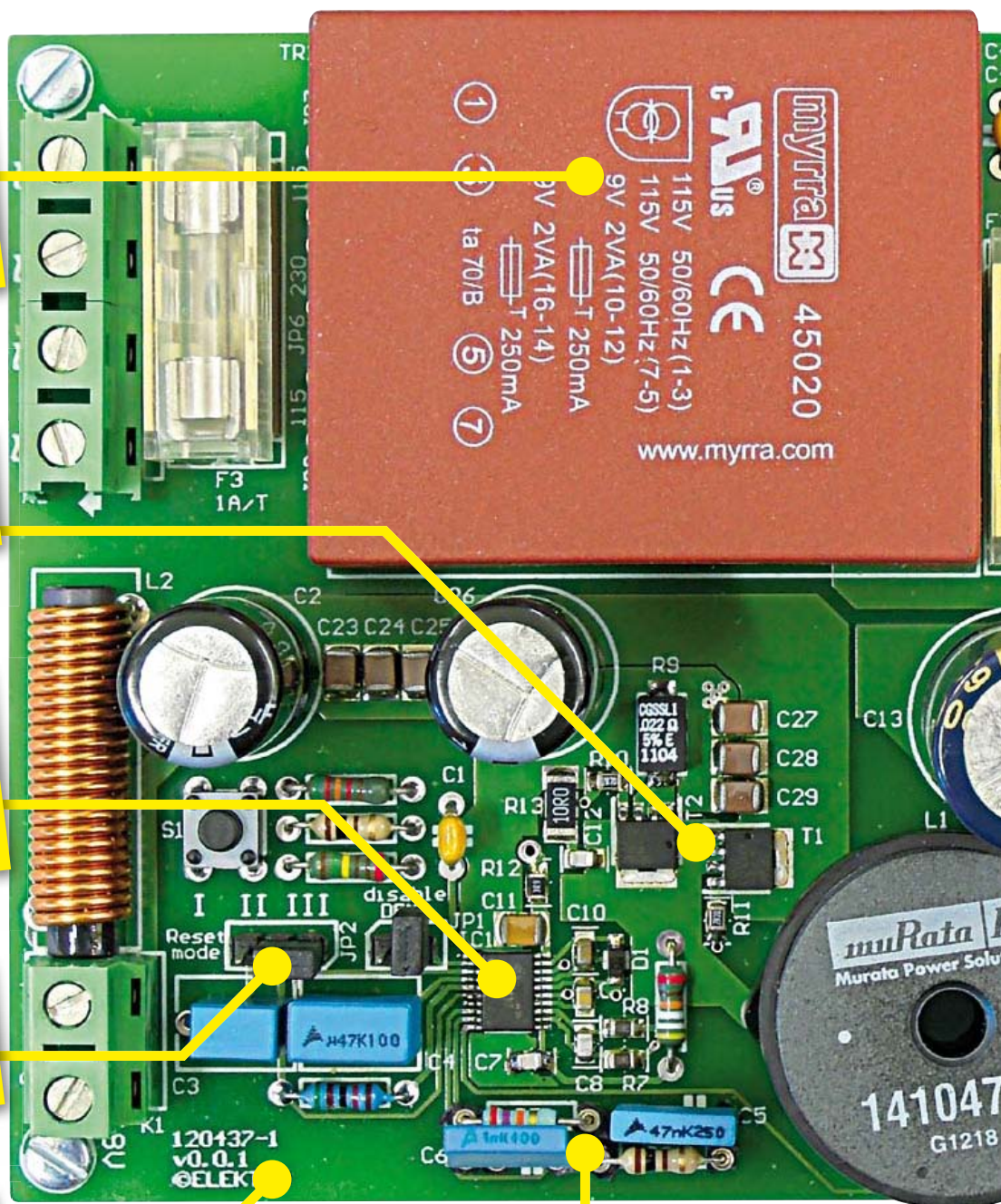
T1 et T2 sont des FET de puissance particuliers de NXP, avec une $R_{DS(on)}$ extrêmement faible.

IC1 et IC3 ne sont pas faciles à souder à la main, mais ça reste faisable, de préférence à l'air chaud. Avec un four à refusion, c'est encore mieux.

Le mode de remise à zéro est déterminé par la position de JP2 (hiccup, latch-off, cycle-by-cycle)

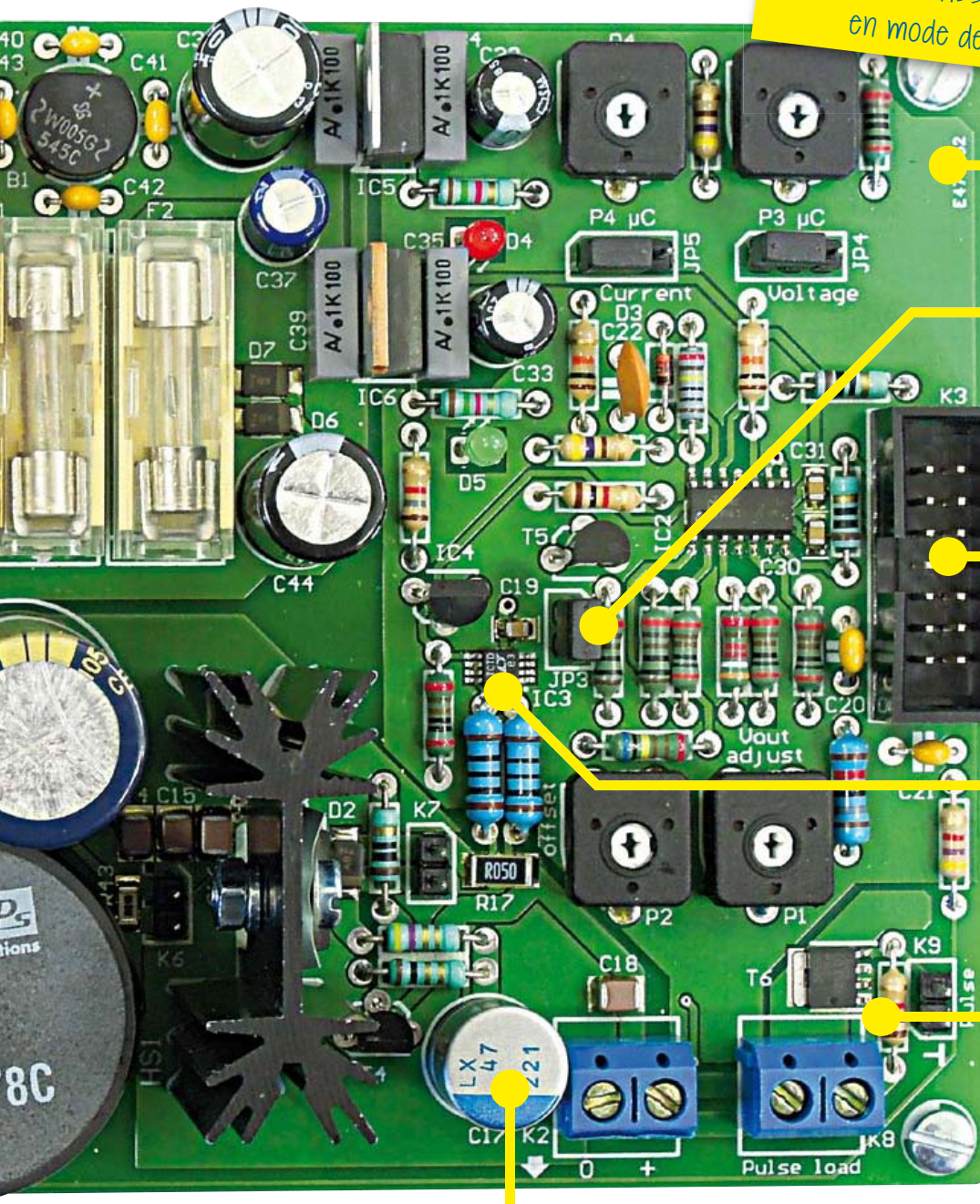
Vu le numéro de cette version du circuit imprimé : v0.0.1, on n'y est pas encore.

Pour le réglage fin du filtre de ré-injection (C5, C6), Ton s'est fié à l'expérimentation.



Au labo d'Elektor, dans les casseroles de notre concepteur à cinq étoiles Ton Giesberts, un nouveau projet d'envergure prend forme. C'est une alimentation de labo à découpage qui mijote. Pour simplifier, on l'appelle SMPSU (switch-mode power supply). Ton a bien voulu nous laisser soulever le couvercle... goûtez un peu, ça promet, non ?

Thijs Beckers (Elektor)



« Au fond, c'est une alimentation à convertisseur abaisseur en mode de courant. »

Circuit imprimé à 2 couches

JP3 permet de choisir entre une limitation de courant par le matériel ou par le logiciel.

Logiciel et conception en cours pour l'interface vers le microcontrôleur. Pas moyen d'en savoir plus pour l'instant. Cachotier, le cuistot !

On dirait qu'il y a un court-circuit interne entre les broches NC d'IC3? Investigations en cours...

T6: section à l'étude avec des tests de charge impulsionnelle... ça dépote.

Le seul fabricant qui garantisse les caractéristiques imposées par le concepteur à C17 est Nichicon. Gare à la disponibilité ?! Les résultats des premières mesures de résistance série équivalente ESR de C17 ($0,4 \times \Omega$) n'étaient pas concluantes, tout bêtement à cause d'une boucle de masse.

mini-assistant de laboratoire

multimètre logique et numérique



Thomas Ücok
(Allemagne) &
Jens Nickel (Elektor)

La mise au point d'applications à microcontrôleurs passe toujours par des tests ponctuels. « L'UART débite-t-il effectivement quelque chose ? Quel est le niveau de la broche PC2 ? Ai-je configuré correctement mon temporisateur pour produire la fréquence voulue ?... » Pour toutes ces tâches et bien d'autres, notre lecteur Thomas Ücok a mis au point un multitesteur sur la base du module *Minimod* d'Elektor. C'est le point de départ d'un projet d'Elektor, encore en chantier.

La table d'un électronicien concepteur est habituellement trop petite. Dès qu'il y a quelque chose à tester rapidement, il faut d'abord faire de la place pour l'oscilloscope, le géné BF et le reste. Il faut ensuite démarrer les appareils et les régler. Pour les petits tests, toute cette mise en œuvre est inutile, comme le montre le module *Minimod* [1] d'Elektor, un multitesteur maniable et facile à utiliser (**fig. 1**).

Le menu offre les fonctions suivantes :

- Testeur de niveau logique 5 V/3 V
- Mesure de fréquence jusqu'à 7 MHz
- Testeur de niveau CMOS 5 V (haut/bas/ouvert)
- Mesure de tension analogique de 0 à 5 V
- Compteur d'impulsions (front montant)
- Compteur d'impulsions (front descendant)
- Récepteur de test UART
- Émetteur de test UART
- Lecture du numéro en ROM d'un capteur 1-Wire
- Mesure de durée d'impulsion négative (max. 45 ms)
- Mesure de durée d'impulsion positive (max. 45 ms)
- Générateur rectangulaire 1 kHz, 10 kHz, 100 kHz et 500 kHz
- Mesure de la durée d'impulsion d'un signal de commande de servo-moteur

Au travail : un nouveau multitesteur

Malheureusement, le *Minimod* n'est plus disponible. Toutefois, nous avons trouvé l'idée de Thomas Ücok si bonne que nous avons décidé de développer un nouveau *multitesteur*, sans la carte *Minimod*.

La grande illustration montre un premier prototype. Les résultats de test sont affichés sur un mini-écran à cristaux liquides de deux lignes de huit caractères. Le testeur est alimenté, via une fiche USB, par un PC de table ou portable, ou bien par une petite alimentation USB. Deux touches suffisent pour sélectionner et activer toutes les fonctions.

Pour être capable de supporter jour après jour l'environnement impitoyable du laboratoire, un appareil de mesure ne peut se passer d'un coffret robuste. Pour la première fois, nous avons conçu pour notre projet un coffret sur mesure qui peut être réalisé par une imprimante 3D. La **figure 2** représente un prototype de ce coffret.

Le schéma

La **figure 3** représente l'état actuel du schéma qui transcrit quelques-unes des bonnes idées de Thomas Ücok. L'organe central est un ATmega328P monté de façon classique, avec quartz de 16 MHz, alimentation 5 V et connecteur ISP pour la programmation. L'écran à cristaux liquides est piloté en mode 4 bits. Le potentiomètre P1 règle le contraste, le cavalier JP1 allume à la demande le rétro-éclairage de l'écran.

L'interface USB sert avant tout à l'alimentation du circuit. Elle permet aussi de télécharger un nouveau micrologiciel, si on utilise le chargeur *USBaspLoader* [2]. Cet outil assure la communication USB par logiciel, ce qui nous dispense d'ajouter au circuit un composant convertisseur série/USB.

Le bus I2C pilote une puce EEPROM 24C512 qui peut être utilisée pour stocker les résultats des mesures.

Le seul composant inhabituel du circuit est le commutateur audio/vidéo MAX4584 [3], piloté lui aussi par le bus I2C. Ce commutateur applique le signal à mesurer (amené par exemple par une pointe de touche) à différentes entrées du microcontrôleur. En effet, il n'y a pas dans la famille AVR de broche qui assure toutes les fonctions de mesure et d'entrée-sortie. On utilise d'une part la broche PC7/ADC7 pour la mesure des niveaux TTL/CMOS, la mesure de tension et le comptage d'impulsions, d'autre part la broche PD4/T0 pour



Figure 1.
Le *multitesteur* de Thomas Ücok est basé sur le module *Minimod*.

la mesure de fréquence. Cette broche sert aussi comme sortie série de l'UART.

L'entrée analogique passe par un diviseur de tension (R1, R2, R3). Le diviseur polarise l'entrée à 1 V, ce qui la place dans la zone interdite de la plupart des familles TTL. C'est ainsi qu'on peut détecter les entrées ouvertes lors de la mesure des niveaux TTL. En mode « analogique », on mesure toujours 1 V tant qu'aucune autre tension n'est appliquée.

La mesure de fréquence nécessite une entrée de compteur (T0). Or la broche PD4/T0 est déjà utilisée par l'écran à cristaux liquides. C'est pourquoi le circuit présenté exploite un petit truc :

Figure 2.
Ce coffret pour le *multitesteur* a été fabriqué par impression 3D.



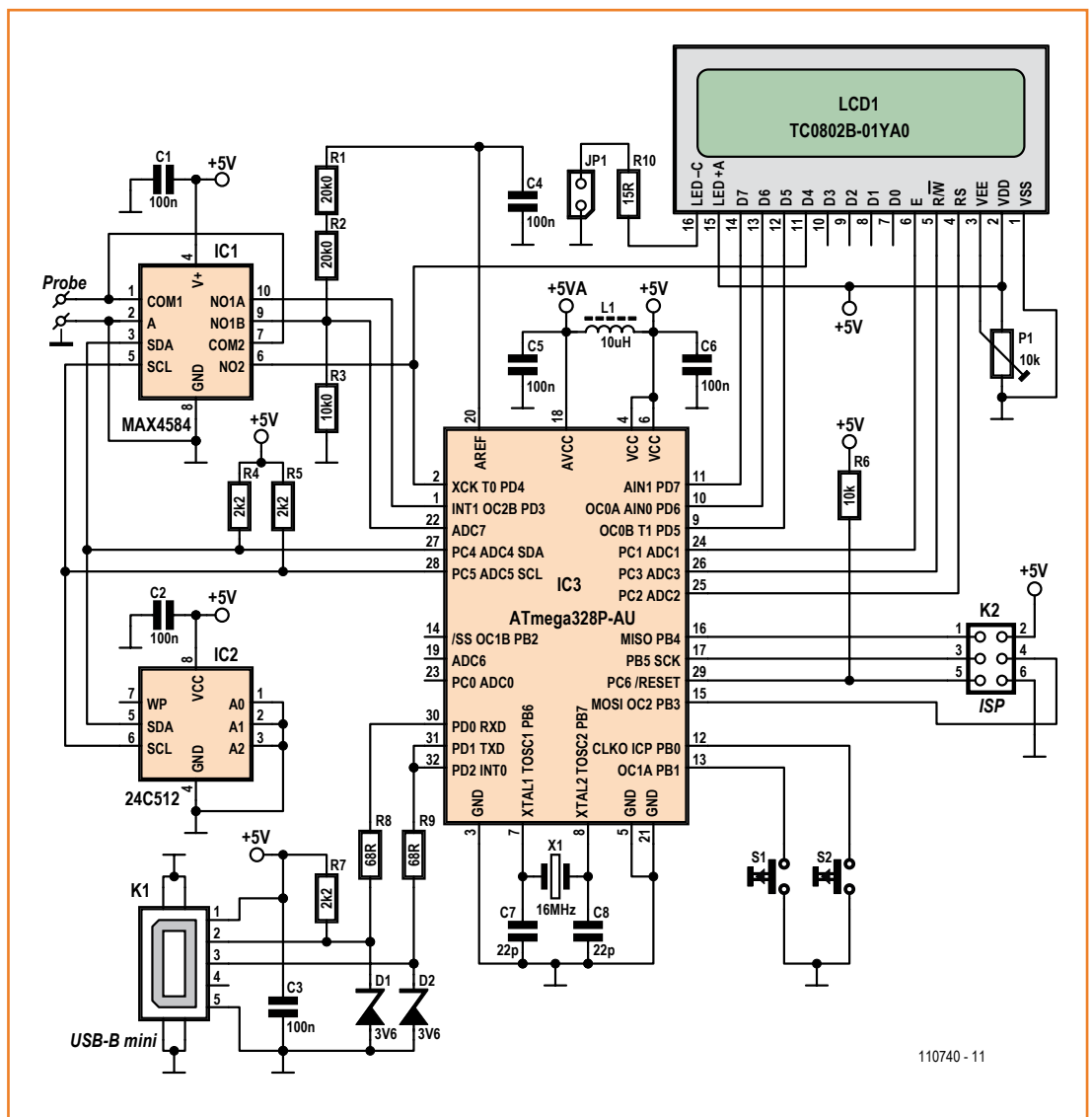


Figure 3.
La base du circuit est un ATmega328 (compatible broche à broche avec l'ATmega88 bien connu).

pendant la mesure, il n'y a pas d'affichage, ce qui laisse le temps de re-configurer la broche en entrée de compte.

La platine

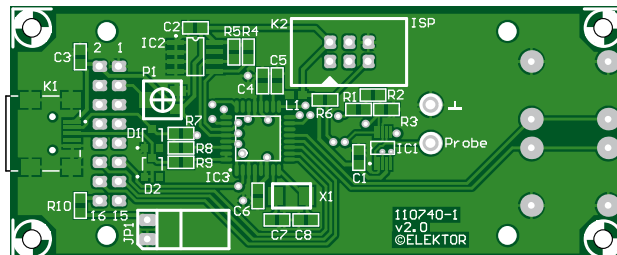
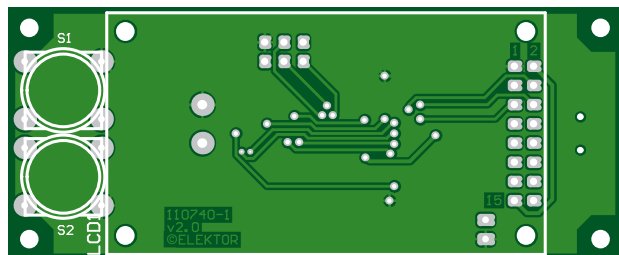
Le prototype de circuit imprimé est à double face (fig. 4). Les touches et l'écran sont les seuls composants d'une face. Naturellement, on peut aussi relier l'écran par des fils, mais l'enfiler est plus élégant. On soude pour cela une embase HE10 pour circuit imprimé au dos de l'afficheur. En face, la platine du *multitesteur* est munie d'une barrette à deux rangées de 8 broches rondes. Ces broches ont un côté plus fort (0,7 mm), mais c'est l'autre qui est soudé dans les trous du circuit imprimé. Le cavalier JP1 est monté à plat,

ce qui économise de la place. Pour fixer l'écran, nous utilisons quatre vis M2 et douze écrous. L'entrée du signal et la masse passent par deux picots de 1,3 mm soudés du côté composants, on peut ensuite les plier à l'horizontale, de façon à ce que ce côté du module reste plat. Le connecteur ISP n'est monté qu'en cas de besoin.

Logiciel

Le prototype actuel met en œuvre le logiciel de l'auteur, écrit en BASIC Bascom. Le logiciel offre de nombreuses possibilités d'extension et d'amélioration.

Des informations sur l'état d'avancement du projet sont disponibles sur notre site eLabs [4] où il est toujours en chantier. Vous pouvez télécharger



le dessin de circuit imprimé, la liste des composants et naturellement le logiciel. Les modes d'emploi du multitesteur disponibles avec ce logiciel sont décrits en détail.

Plus que jamais, pour ce circuit en cours d'élaboration, nous recevrons avec plaisir vos commentaires, propositions d'amélioration, développements maison et autres avis.

(110740 – version française : Jean-Paul Brodier)

Liens

- [1] www.elektor.fr/090773
- [2] www.obdev.at/products/vusb/usbasploder.html
- [3] www.maximintegrated.com/datasheet/index.mvp/id/2080
- [4] www.elektor-labs.com/9120802389

Figure 4.

La plupart des composants se trouvent d'un côté de la platine, l'autre ne porte que l'afficheur et les touches.



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 €.





PCBs Muuuuch Cheaper...

PCBs starting from **17.22 EURO***

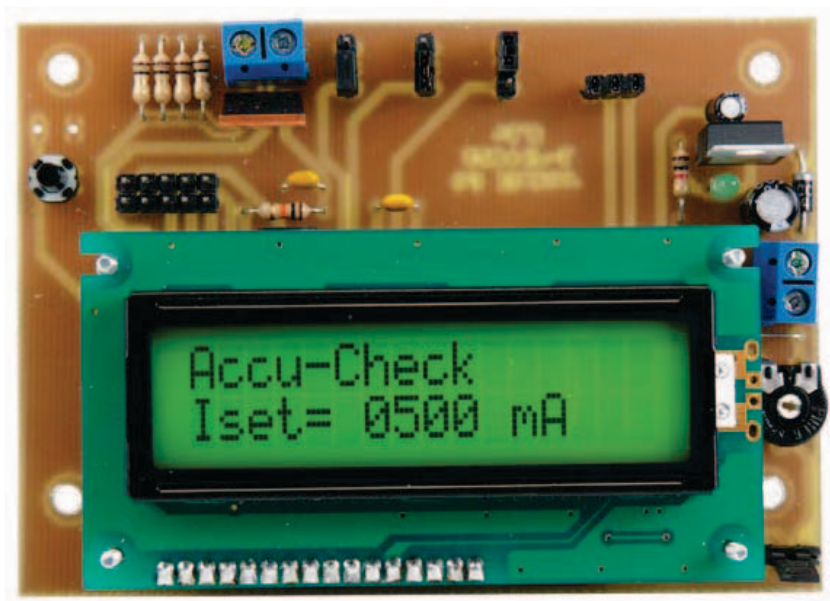
* incl. VAT (23%), add delivery costs
(Example Germany) of 10.89 EURO,
min. number of ordered pcbs:
5 @ 100 mm x 100 mm

No-frills policy

www.jackaltac.com

testeur d'accu 1,2 V

mesure de capacité par décharge



Martin Ossmann (Allemagne)

Plutôt colossale, ma collection d'accus de 1,2 V. La vôtre aussi ? Vous en avez sûrement de tous types, de tous âges, gaillards les uns, moribonds les autres ! Comme moi, faites le tri avec ce petit appareil infallible.

Pour ausculter un accumulateur, il n'y a qu'à bien le charger, puis examiner sa décharge. Avec cet appareil de mesure de la décharge, vous pourrez déterminer si sa capacité s'est améliorée après quelques cycles de charge/décharge ou s'il est arrivé irrémédiablement en fin de vie. Au cœur du circuit, il y a le MOSFET T1 et les résistances de shunt R6 à R9. Le transistor est piloté en modulation de largeur d'impulsions (MLI) de manière telle que le courant moyen prévu de décharge de l'accu (sur K2) circule continuellement dans le *shunt*. On mesure le courant de décharge et la tension de la cellule d'accumulateur. On intègre alors le courant consommé par le *shunt* pour déterminer, en mAh, la charge Q initiale emmagasinée. À partir de 1,1 V jusqu'à la tension finale de 1,0 V, le courant est réduit linéairement de sa valeur maximum jusqu'à 0. Les paramètres actuels de temps, tension, courant et charge sont tous affichés sur écran LCD et transmis également par l'interface série pour une collecte de données sur PC.

Le circuit

Non, le circuit n'est pas particulièrement compliqué, la **figure 1** le confirme. Toutes les mesures, les calculs et l'affichage des résultats sont vaillamment pris en charge par le contrôleur IC1, un ATmega88. Il doit son pouls à un quartz à 11,0592 MHz. Un bloc secteur fournit l'alimentation, entre 9 et 12 V, pour un LM7805 qui stabilise la tension à 5 V, tandis que IC2, un LM385-2V5, délivre la tension de référence de 2,5 V pour le convertisseur A/N. Le signal rectangulaire sur la broche 12 (PD6/PWM C0A) sert à produire une tension négative pour le contraste du LCD, réglable par P1. Le cavalier JP1 permet aussi d'activer l'éclairage d'arrière-plan de l'écran.

La MLI sur la broche PB1 est convertie par C2 et R3 en une tension continue de commande pour le MOSFET T1. Le courant de décharge est mesuré aux bornes du *shunt* R6 à R9 et réglé par T1 pour correspondre à la valeur programmée.

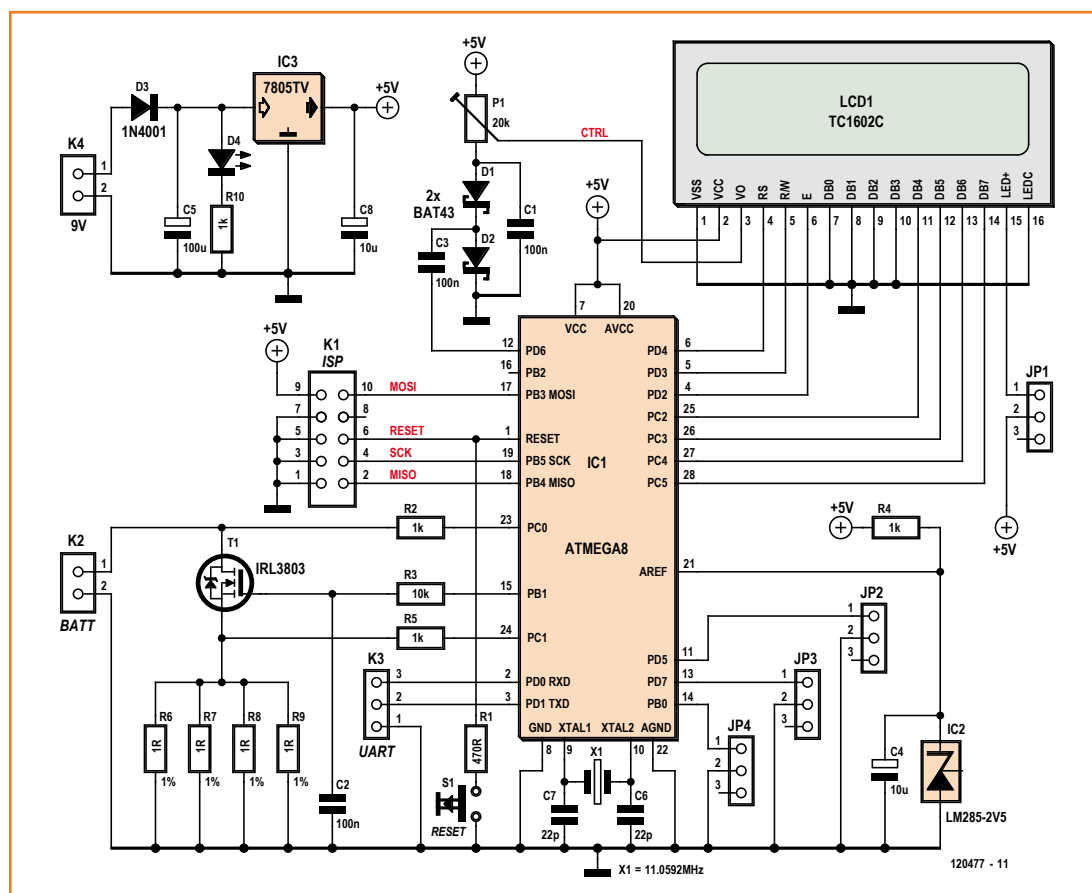


Figure 1.
Le testeur d'accu de 1,2 V
avec T1 comme instrument
de décharge.

Les canaux analogiques ADC0 (PC0) et ADC1 (PC1) sondent la tension et le courant. On règle le courant de décharge entre 50 mA et 1 A au moyen des cavaliers JP2 à JP4 en se référant au **tableau 1**.

Les lignes PD0 et PD1 du contrôleur forment l'interface sérielle, on peut y brancher par exemple un convertisseur USB/TTL et, sous 19 200 bauds, lire tous les paramètres au format :

T=00043 sek U=1.396 V I=0.502 A
Iset=0.500 A Charge=0003 mAh

Le circuit imprimé porte également une interface de programmation ISP, il est donc possible de charger par son intermédiaire un nouveau micrologiciel pour le contrôleur.

Construction et mise en œuvre

Pour ce projet, nous proposons via l'échoppe le circuit imprimé seul et le contrôleur

préprogrammé. Vous gardez toutefois la liberté de tout faire vous-même et télécharger comme d'habitude le tracé des pistes et l'ensemble du logiciel [1].

Pour la construction du testeur d'accu, il y a peu de choses à expliquer : la platine de la **figure 2** est à simple face, il n'y a pas de CMS, tous les composants sont à fils ; il y a sept ponts de câblage, dont un sous le quartz à installer sur la

Tableau 1. Position des cavaliers de réglage du courant

Courant (mA)	JP-4	JP-3	JP-2
1000	mis	mis	mis
500	mis	mis	omis
250	mis	omis	mis
100	mis	omis	omis
50	omis	mis	mis

Liste des composants

Résistances :

R1 = 470 Ω
 R2, R4, R5, R10 = 1 k Ω
 R3 = 10 k Ω
 R6 à R9 = 1 Ω , 1 %/0,5 W
 P1 = 20 k Ω aj. horiz.

Condensateurs :

C1 à C3 = 100 nF
 C4, C6 = 10 μ F/10 V, radial au pas de 2 mm/5 mm
 C5 = 100 μ F/10 V, radial au pas de 2,5 mm/6 mm
 C7, C8 = 22 pF

Semi-conducteurs :

D1, D2 = BAT43
 D3 = 1N4001

D4 = LED rouge 3 mm
 T1 = IRL3803
 LCD1 = TC1602C
 IC1 = ATmega88-20 (programmé : 120447-41)
 IC2 = LM285-2V5
 IC3 = 7805

Divers :

LCD1 = LCD 2x16, p. ex. Elektor 120061-71
 K1 = embase à 2x5picots 2,54 mm
 K2, K4 = domino encartable au pas de 5,08 mm
 K3 = embase à 3 picots 2,54 mm
 JP1 à JP4 = embase à 3 picots 2,54 mm
 S1 = bouton-poussoir
 circuit imprimé 120447-1

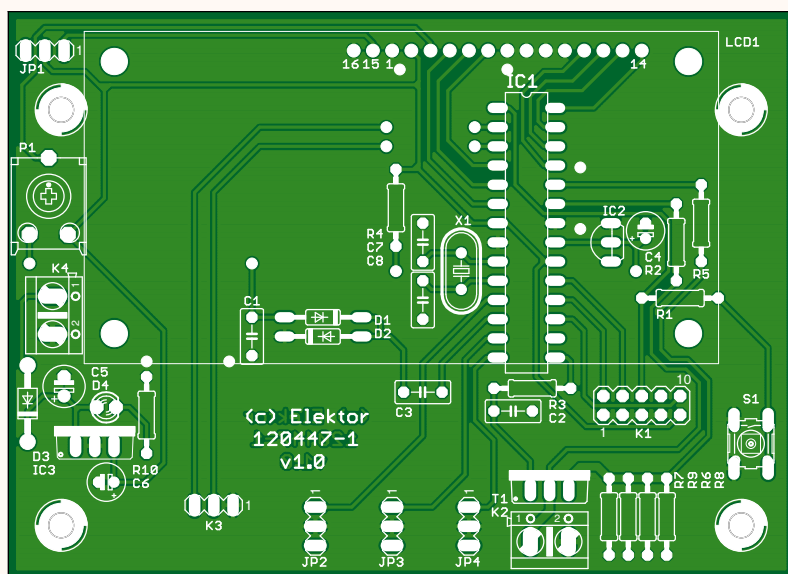
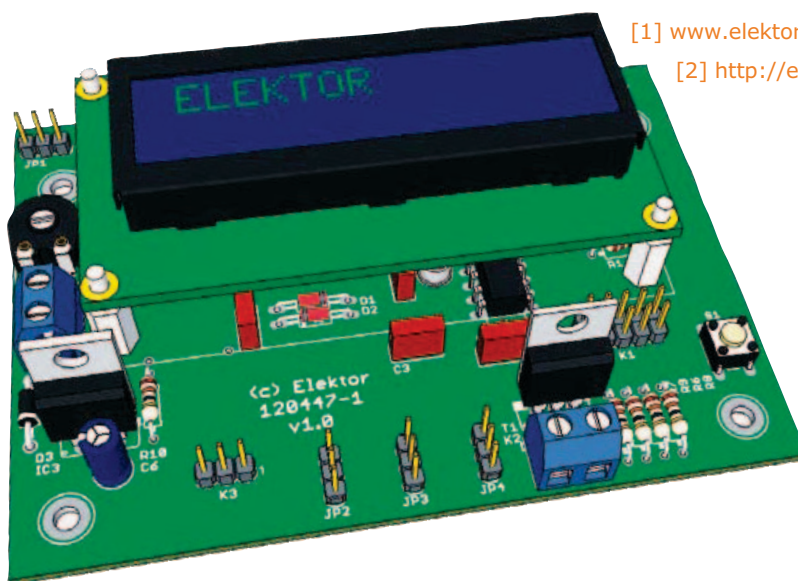


Figure 2.
Le circuit imprimé du
testeur d'accumulateur

Figure 3.
Le circuit imprimé assemblé
selon SketchUp, fichier en
3D créé par EagleUp [2]
que l'on peut également
télécharger du site d'Elektor.



face cuivrée. C'est par là qu'il faut commencer le soudage. L'écran se monte en impériale sur le circuit imprimé, mais de façon à laisser l'accès à l'interface utilisateur, aux raccords, aux boutons et aux cavaliers.

Comme on dispose de la référence du LM385, il n'y a pas d'étalonnage. Les fusibles du contrôleur sont à programmer pour travailler avec un quartz externe. Le transistor T1 requiert un petit radiateur pour encaisser la puissance maximale de $1 \text{ A} \times 1,3 \text{ V} = 1,3 \text{ W}$. La tolérance des résistances de *shunt* R6 à R9 ne devrait pas dépasser 1 % si vous voulez des mesures de courant dignes de confiance.

Raccordez à K2 un support de pile adapté et insérez-y l'accumulateur de 1,2 V récemment chargé. Appuyez sur le bouton S1 pour une mise à zéro qui relance le temporisateur et initialise l'intégration de la décharge. Le reste se fait tout seul. Le circuit règle le débit sur la valeur prévue. Durant les premières heures, la tension décroît de 1,4 V à 1,2 V. Puis elle reste relativement stable au niveau de 1,2 V, comme on s'y attend. C'est ce qui fait durer la décharge. Finalement, la tension recommence à baisser. Quand on atteint 1,1 V, l'appareil réduit le courant de décharge. À 1,0 V, il le ramène à zéro pour éviter une décharge profonde. Au bout du processus, on peut lire la capacité de l'accumulateur en bas, à gauche de l'écran.

(120447 – version française : Robert Grignard)

Liens

- [1] www.elektor-magazine.fr/120447
- [2] <http://eagleup.wordpress.com>

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

deux kits d'initiation disponibles

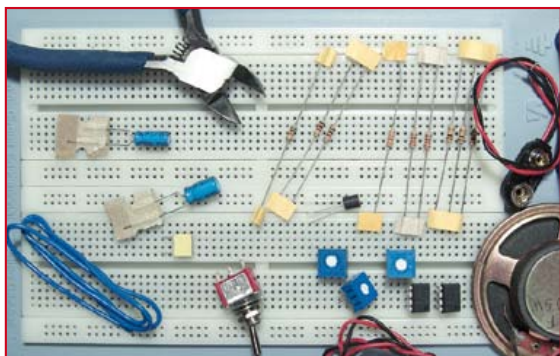
Le cadeau idéal pour partager votre passion de l'électronique avec vos enfants, petits-enfants, neveux... et autres *geeks*.

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

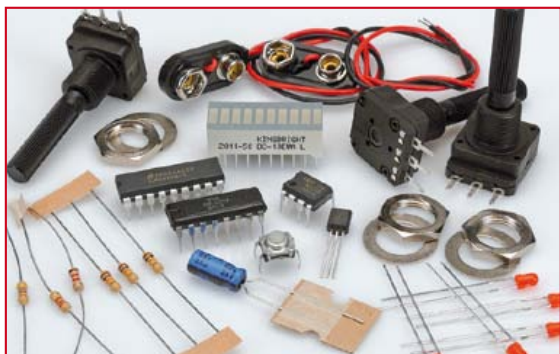
L'auteur démystifie l'électronique en n'utilisant que ce qu'il vous faut de théorie pour aborder la pratique : identifier les composants et leur rôle, les récupérer, les tester et les ranger ; lire un schéma ; choisir ses outils ; mettre en boîte ses montages...

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

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



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



ISBN 978-2-86661-186-6
édition revue et augmentée
352 pages - **42,50 €**



informations complémentaires et commande :
www.elektor.fr/debut

scanner de radionavigation aérienne

Pour les amateurs d'aviation et en particulier les observateurs du trafic aérien aux abords des aérodrômes, suivre les communications entre les avions et la tour de contrôle revêt un grand intérêt. Ce récepteur-ci vous permet d'écouter sans difficulté tout le trafic dans la bande de navigation aérienne civile. Il est facile à construire, ne demande qu'un seul réglage et vous pouvez le piloter de votre PC par USB.

Gert Baars (Pays-Bas)

« Schiphol Approach, KLM123 with information delta, descending to flight level 70 just passed artip inbound spy, speed 250 knots... »

Il y a toujours quelque chose d'excitant à suivre la communication entre un avion et le sol. Il ne s'agit d'habitude que d'informations de routine sur le relevé de la position ou d'indications pour le pilote qui va décoller ou se poser, mais ces voix vous procurent un certain frisson, un peu comme si vous étiez vous-même dans le cockpit à côté du pilote.

Il y a pas mal d'amateurs qui écoutent les échanges verbaux de navigation aérienne parce qu'ils les trouvent fascinants, mais aussi des observateurs, pour découvrir quel appareil est en approche ou prêt à décoller.

Bien sûr, des scanners de radiotéléphonie aérienne existent prêts à l'emploi et ne coûtent pas trop cher, mais c'est compter sans le plaisir de le fabriquer soi-même et de réaliser des expériences avec lui. Le modèle présenté ici est de construction simple, mais son microcontrôleur offre une belle série de fonctions supplémentaires ; il permet surtout la commande et la programmation par ordinateur des fréquences utiles.

Réflexions

Pour garder au circuit une certaine simplicité, nous allons limiter le scanner à la bande de radionavigation aérienne civile de 108 à 137 MHz. Cette gamme était à l'origine divisée en canaux de 25 kHz, elle s'est ensuite modernisée pour accepter des sauts de $25/3 = 8,33$ kHz, mais en pratique, cette subdivision est rarement en usage. Notre scanner se programme avec une précision de 1 kHz, le décalage ne dépassera donc jamais 330 Hz. La bande passante requise en modulation d'amplitude avoisine 6 kHz. Nous savons ainsi à quelles exigences le récepteur devra satisfaire.

Pour ne pas nous encombrer d'un clavier et d'un écran supplémentaires, utilisons l'ordinateur comme terminal. Le scanner communique par une liaison série USB avec un PC sur lequel tourne un programme d'interface via *hyperterminal*. Ainsi, depuis le PC, on programme les fréquences et différents autres réglages.

Côté récepteur, nous avons choisi un système relativement simple avec une unique puce, un récepteur superhétérodyne FM à un seul changement de fréquence, utilisé comme récepteur AM

portable et avec USB

Caractéristiques techniques

- Programmable sous Windows
- 100 mémoires programmables
- Plage de réception de 108 à 137 MHz, AM
- Sensibilité de 0,2 μ V pour 6 dB de rapport s+B/B
- Vitesse de balayage : 5 canaux/s
- Le scanner peut programmer lui-même les fréquences trouvées
- Accord manuel possible par boutons up/down

en prenant le signal de force de champ comme détecteur d'amplitude. Avec pareil circuit, il apparaît une fréquence image qu'il faut atténuer. Le plus simple pour y arriver, c'est de laisser une grande distance entre la fréquence de réception et la fréquence image, ce qui demande de choisir une fréquence intermédiaire (FI) élevée. Avec la puce choisie pour le récepteur, le maximum se situe aux alentours de 25 MHz. On peut aussi prendre 27 MHz, mais les filtres de bande correspondants sont souvent chers et difficiles à dénicher. Nous avons opté pour deux filtres de bande bâtis sur des quartz à 27 MHz, il en faut quatre en tout, mais, à 0,50 euro

la pièce, ce n'est pas un drame. L'avantage d'une FI à 27 MHz, c'est que la fréquence image se situe entre 54 et 83 MHz, juste sous la bande de radiodiffusion FM.

On réalise souvent le réglage numérique de fréquence sur les récepteurs par *PLL* ou *DDS*, mais pour faire simple, il est aussi possible de choisir pour ce projet une solution moins élaborée, comme la boucle à asservissement de fréquence (*FLL*). Le contrôleur mesure la fréquence de l'oscillateur à fré-

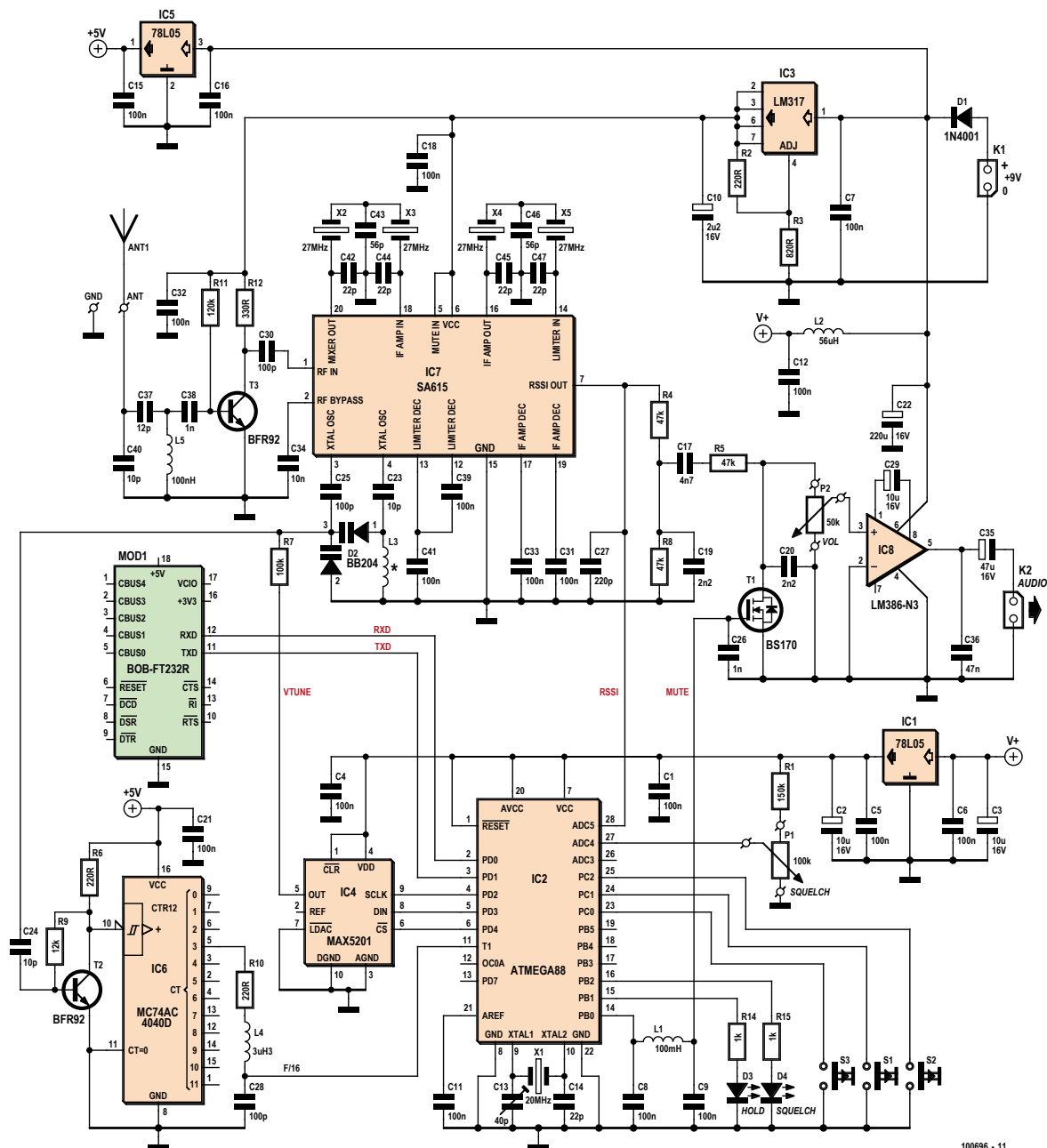


Figure 1.

Le schéma du récepteur : en haut, la section analogique, en bas, la section numérique. Trois stabilisateurs intégrés délivrent des tensions d'alimentation à toute épreuve pour les différentes parties du montage.

quence variable (VFO), la compare à celle précisée dans le logiciel et la corrige par le convertisseur N/A. Des essais ont montré que cela marche parfaitement si la résolution du CNA est suffisante pour obtenir les petits sauts de fréquence voulus. L'entrée du compteur du contrôleur ne peut pas gérer de très hautes fréquences, il a fallu intercaler un diviseur derrière l'oscillateur. La solution qui en résulte n'utilise que relativement peu

de composants. Le scanner ne totalise que trois boutons-poussoirs, un réglage de volume et un de silencieux (*squelch*). Le choix des fréquences et des réglages s'opère sur le PC, après quoi le scanner redevient portable et autonome.

En supplément, le scanner dispose d'un mode dans lequel il peut rechercher seul les fréquences et les programmer. Il faut pour cela régler le

silencieux, démarrer en mode de recherche et attendre que le balayage soit complet. Le scanner écoute d'initiative une à une toutes les fréquences dans la bande aéronautique et les programme en mémoire s'il y trouve un signal.

Composante réceptrice

Le récepteur dont le schéma est à la **figure 1** est construit autour d'une puce du type NE615 (SA615). Elle constitue une version améliorée de la combinaison de deux anciens circuits, le NE602, un mélangeur avec VFO, et le NE604, un amplificateur FI avec limiteur et détecteur FM. Comme la détection FM ne nous intéresse pas, de nombreuses broches sont inutilisées. Mais la puce offre une sortie RSSI (*received signal strength indicator*) qui délivre une tension proportionnelle à la force du signal. Nous la partageons entre la sortie audio, agrémentée de filtres et d'une atténuation du souffle et, pour le contrôleur, la détection de présence du signal, le balayage s'arrête alors pour reconnaître la station.

On améliore la réception en soumettant le signal d'antenne à un filtrage et une amplification avant de le mélanger avec le VFO, l'oscillateur local, accordé par une double diode à capacité variable. Il en résulte un signal à FI sur 27 MHz à filtrer par deux cellules à quartz (X2 à X5) pour façonner une bande passante de 6 kHz. La FI n'est pas exactement sur 27,000 MHz, mais plutôt à 26,998. On pourra la modifier dans les réglages initiaux, mais à l'origine, la FI est à 26,998 MHz. Les quartz doivent avoir un type de fondamentale identique, mais un décalage de quelques kHz n'est pas inconciliable, on le rattrapera au besoin dans l'initialisation.

Le signal du VFO est envoyé à travers l'étage amplificateur T2 au 74AC4040 (IC6) configuré en diviseur par 16. La plage va de $f_{HF} - f_{IF} = 81$ à 110 MHz. L'entrée du compteur du contrôleur n'accepte que 8 MHz au maximum, une division par 14 est donc un minimum indispensable.

Le signal de sortie audio de la broche 7 d'IC7 atteint, par quelques résistances et le FET de silencieux T1, le réglage de volume P2, lequel l'envoie au petit amplificateur audio IC8, un LM386, capable d'attaquer un petit haut-parleur de 4 ou 8 Ω .

La combinaison de signaux analogiques (HF) et numériques rend nécessaire une stabilisation de tension soignée. C'est pourquoi on n'y trouve pas moins de trois régulateurs. IC3 approvisionne la

Bandes aéronautiques

Quand on veut construire un scanner pour bandes dites aéronautiques, il faut commencer par regarder dans quelles gammes de fréquences elles se trouvent. Ces bandes comportent principalement la plage de fréquence pour l'aéronautique civile : VHF 108 à 137 MHz et celle pour l'aéronautique militaire : UHF 230 à 400 MHz, dans lesquelles on utilise la modulation d'amplitude (AM). Mais il existe aussi des bandes HF entre 3 et 30 MHz en SSB principalement pour les communications aéronautiques longue distance.

Notre scanner se limite à la bande civile. Elle sert aux communications entre les contrôleurs du trafic aérien (ATC) et les aéronefs, entre les avions, mais aussi entre les avions et les bureaux des compagnies aériennes ou les véhicules au sol.

partie HF en 6 V bien propre. La fraction numérique se voit dotée de son propre régulateur de 5 V, c'est IC5, alors que le diviseur IC6 est nourri sur le côté par IC5, un 78L05.

Composante à microcontrôleur

Le microcontrôleur IC2, un ATmega88, ne manque pas de besogne. Il doit mesurer la fréquence du VCO divisée par 16. Il dispose pour ce faire d'un compteur dont le port est activé pendant 16 ms, de manière à obtenir une mesure en kHz. Le pilotage du VCO revient à IC4, un CNA à 16 bits MAX5201. C'est bien le nombre de bits qu'il nous faut. Une gamme de fréquences [en kHz] de $29\,000 / 2^{16}$ produit environ 440 Hz. Avec une bande passante de 6 kHz, la précision de la syntonisation est acceptable. Deux caractéristiques importantes ont prévalu dans le choix du CI : l'excursion totale de l'amplificateur de sortie et sa commande sérielle.

La capacité de mémoire de l'EEPROM autorise, en plus des réglages, l'acquisition d'une centaine de fréquences. C'est plus que nécessaire, mais permet également des sélections dans différentes banques de fréquences, par exemple 5 de 20 fréquences.

Deux entrées du CAN du microcontrôleur numérisent la force du signal et le réglage du potentiomètre de silencieux (broches 28 et 29). Ces valeurs sont comparées en interne et le résultat décide quand stopper le balayage ou le relancer. Les LED D3 et D4 donnent des informations notamment sur le niveau du silencieux et l'attente pendant le balayage. Il y a aussi trois boutons-poussoirs, S1 à S3, pour bloquer un canal pendant le balayage, par exemple. La passerelle USB/série BOB-FT232R [1] déjà utilisée

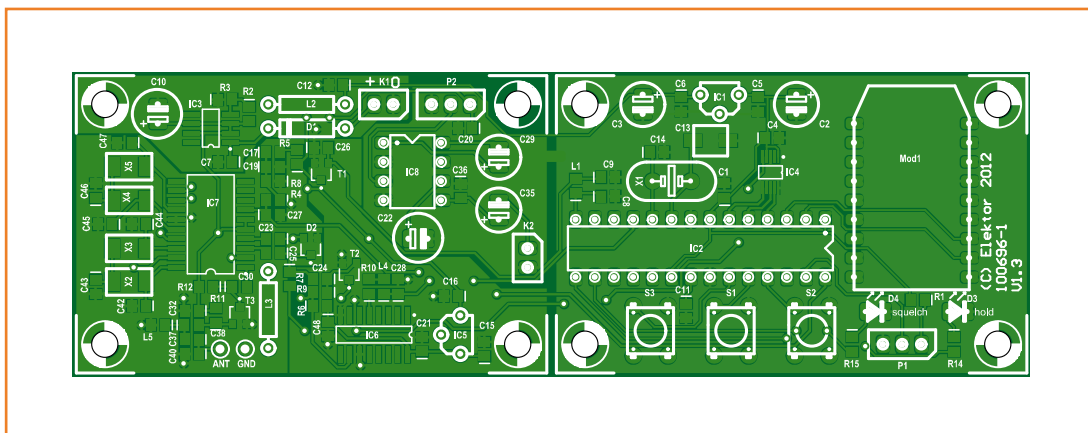


Figure 2.
On peut scinder en deux le circuit imprimé, tout dépendra du boîtier retenu.

dans divers projets d'Elektor est connectée aux broches 2 et 3 du microcontrôleur. Une place lui est réservée sur le circuit imprimé. Ce convertisseur permet la liaison avec un PC au moyen d'un câble USB.

Composante logicielle

C'est bien sûr le logiciel qui fait fonctionner le scanner comme il se doit. Dans son parcours principal, il lit toujours dans l'EEPROM une fréquence. Il faut en déduire la fréquence pour le VFO. Mais en raison du système utilisé, selon le principe de la FLL, on ne peut pas se contenter d'envoyer une fréquence. Avec le CNA à 16 bits, 16 mesures de 16 ms sont nécessaires. Cela ralentit le balayage, mais simplifie le schéma du scanner en comparaison avec une puce de PLL ou de DDS.

Après accord sur une fréquence, le logiciel vérifie s'il y a un signal sur la sortie RSSI. Si oui, la

sortie de silencieux est mise au niveau bas et le signal audio envoyé vers le haut-parleur. Si le signal disparaît, la fréquence sera encore écoutée un certain temps, réglable, avant que le balayage ne reprenne en lisant une nouvelle valeur dans l'EEPROM pour s'accorder dessus.

Pendant l'attente et l'écoute d'une fréquence, il n'y a pas de changement d'accord. Au cours du balayage, on vérifie le tampon sériel et l'état des boutons poussoirs. C'est alors que l'on peut lire le terminal. Pousser sur S1 permet de sauter un canal et de marquer la fréquence d'un B dans l'EEPROM, une manière de bloquer la fréquence si le scanner a tendance à y rester accroché, à cause de la présence d'une porteuse, par exemple. Ce blocage, on pourra le libérer avec le terminal.

Le scanner possède un certain nombre de réglages standard qui sont lus de la mémoire de programme et inscrits dans l'EEPROM, comme la FI. On garde ainsi le loisir de la changer, depuis le terminal également.

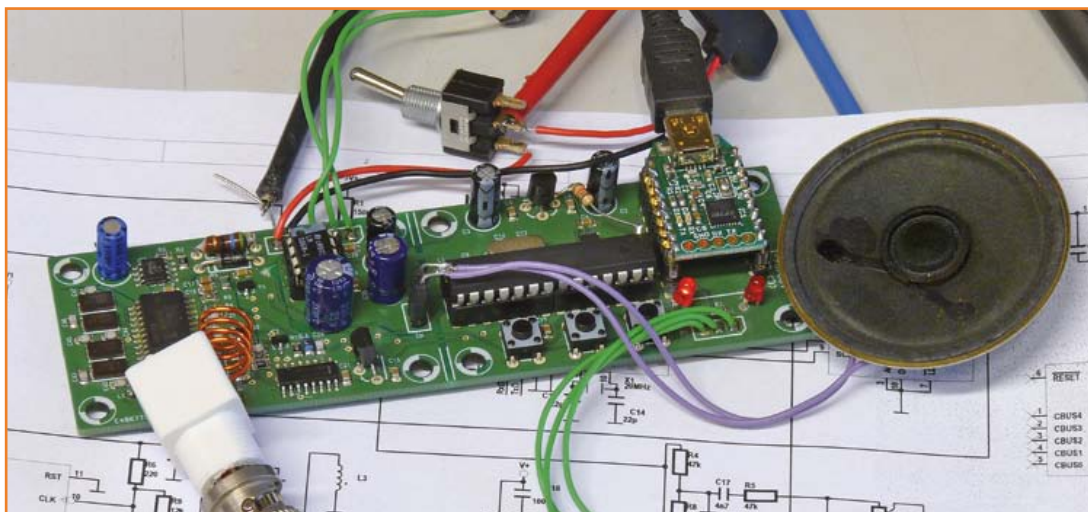


Figure 3.
Même si ce prototype n'a pas l'apparence escomptée, il fonctionne parfaitement.

Construction

La **figure 2** reproduit le circuit imprimé conçu pour le scanner de radionavigation. Il comporte une section analogique et une numérique que l'on peut ou non scinder, suivant la manière de le mettre en boîte. En cas de découpe, il faudra, pour relier ces sections, prévoir 5 conducteurs plus une ligne de masse.

Le mieux est de commencer par l'implantation des CMS, c'est la besogne la plus ardue. Après quoi, on installera les supports de puces et les composants ordinaires à souder. Les deux potentiomètres trouvent place sur le boîtier et se branchent sur la platine au moyen de petits fils isolés.

La self L5 est la seule à bobiner. On la réalise en enroulant 4 spires de fil de cuivre argenté Ø 1 mm sur la queue d'un foret de 7 mm. Étirez ensuite le bobinage pour atteindre la longueur de 7 mm et soudez-le sur le circuit imprimé. Glissez-y ensuite un petit morceau de mousse de polystyrène pour éviter les vibrations. Pour finir, installez le PIC programmé, le LM386 et la passerelle USB/série BOB-FT232R.

Pour le choix du boîtier, tenez compte du fait qu'à côté du circuit imprimé, il faut de l'espace pour les potentiomètres, le haut-parleur et un bloc de 9 V, pile ou accu. Les boutons-poussoirs doivent être actionnés de l'extérieur, au moyen de tubes d'allonge, p.ex. et les LED visibles au dehors, peut-être en laissant les fils plus longs. Il faudra aussi un connecteur BNC pour l'antenne. À l'extérieur, on peut se servir comme antenne d'un brin d'environ 60 cm de long. À la maison, on utilisera de préférence une antenne plus sensible, comme celle qui est décrite ailleurs dans ce numéro.

Réglage

À côté du quartz, un condensateur d'ajustage de 40 pF peut caler avec précision l'accord sur 20 MHz. On peut réaliser l'opération en mesurant en sortie du 74AC4040 la fréquence et l'ajuster, au moyen de la commande T, sur une fréquence de réception fixe. Il faut alors régler cette valeur, moins la FI, le tout divisé par 16. Une fois l'accord exact trouvé, on vérifie que la FI est correcte dans la configuration. L'ajuster au besoin par la commande I. Au début, tournez le condensateur d'ajustage au milieu de sa plage.

Liste des composants

Résistances : (CMS 0805)

R1 = 150 kΩ
R2, R6, R10 = 220 Ω
R3 = 820 Ω
R4, R5, R8 = 47 kΩ
R7 = 100 kΩ
R9 = 12 kΩ
R11 = 120 kΩ
R12 = 330 Ω
R14, R15 = 1 kΩ
P1 = pot. 100 kΩ lin.
P2 = pot. 50 kΩ log.

Condensateurs : (CMS 0805, sauf mention contraire)

C1, C4 à C9, C11, C12, C15, C16, C18, C21, C31 à C33, C39, C41, C48 = 100 nF
C2, C3, C29 = 10 µF/35 V radial
C10 = 2,2 µF/63 V radial
C14, C42, C44, C45, C47 = 22 pF
C17 = 4,7 nF
C19, C20 = 2,2 nF
C22 = 220 µF/16 V radial
C23, C24, C40 = 10 pF
C25, C28, C30 = 100 pF
C26, C38 = 1 nF
C27 = 220 pF
C34 = 10 nF
C35 = 47 µF/16 V radial
C36 = 47 nF
C37 = 12 pF
C43, C46 = 56 pF
C13 = ajustable 8,5 pF à 40 pF (Murata TZB4P400AB10R00)

Inductances :

L1, L5 = 10 nH (SMD 0805)
L2 = 56 µH
L3 = 4 spires fil Cu argenté Ø 1 mm, l = 8 mm, Ø int. 7 mm
L4 = 3,3 µH (CMS 0805)

Semi-conducteurs :

D1 = 1N4001
D2 = BB207 (TO236)
D3, D4 = LED 3 mm à faible courant
T1 = BS170F (SOT23)
T2, T3 = BFR92A (SOT23)
IC1, IC5 = 78L05
IC2 = ATMEGA88-20PU (programmé : EPS 100969-41)
IC3 = LM317 (SOIC-8)
IC4 = MAX5201AEUB+
IC6 = 74AC4040 (SOIC-16)
IC7 = SA615D (SOIC-20)
IC8 = LM386 (DIL-8)

Divers :

X1 = quartz 20 MHz
X2 à X5 = quartz 27 MHz (en fondamentale !) (p.ex. Citizen CS1027.000MABJ-UT)
USB-FT232R BOB 110553-91 [1]
S1 à S3 = bouton-poussoir 6 mm (Multicomp MC32830)
2 embase à 2 broches
support DIL à 28 contacts pour IC2
support DIL à 8 contacts pour IC8
2 x embase à une rangée de 10 broches pour USB-FT232R BOB
haut-parleur miniature 4 of 8 Ω
circuit imprimé 100696-1

Mode d'emploi

Lors de la mise en marche du scanner, l'écran suivant est envoyé au terminal.

```
***** Airbandscanner startup *****

Keyfunctions:

H          -help (this page)
S          -show stored frequencies
B <fn>     -block on/off, fn is freq.number
T F(6)     -tune to frequency(Khz)
R          -scanner run/stop
F fn F(6)  -store frequency(Khz) fn=freq.nr.
I Fif(5)   -Fif (Khz)
N <nf>     -#frequencies to scan (1-100)
W <Tw>     -scanning waittime (x 0.5 sec.)
M <fn>     -memory start from fn=freq.nr.
P <nr>     -#passes (2-15) for autom. search.

current settings:

Fif        = 26998
Twait      = 2
#Freqs.    = 20
Mem.start  = 1
#Srch.pas  = 5

Fn 1
```

Le scanner est alors en mode de balayage et le numéro de la fréquence est visible au bas de l'écran. Pour donner des commandes, il faut saisir d'abord R <Entrée>. Le scanner s'arrête, on peut donner des commandes d'une lettre. La plupart d'entre elles se comprennent aisément, il suffit de les essayer. Après S <Entrée>, les fréquences s'affichent en tableau. On peut régler la première de ces fréquences avec la commande M et leur nombre par la commande N. Ces réglages déterminent aussi la première fréquence et le nombre

total pour un balayage. On peut aussi composer des banques de fréquence de la même manière. C'est toute une banque qui sera alors balayée, réglée par M et N, par exemple de la fréquence 1 jusqu'à la 20. Du terminal, on peut la modifier en indiquant par exemple des fréquences 40 à 59. On peut ainsi répartir en banques les 100 fréquences.

Le scanner est capable de composer lui-même un jeu de fréquences. Si par exemple on a introduit un certain

La commande T permet de fixer la gamme de fréquences couverte entre 108 000 et 137 000. Si le terminal indique une fréquence trop élevée, il faut raccourcir un peu, par compression, la petite self L2 du VFO ; si la fréquence est trop basse, l'opération inverse s'impose : étirer la petite self. Après la commande T, la fréquence affichée devrait être exactement la même que celle que vous avez demandée.

Les codes source et hexadécimal sont disponibles gratuitement au téléchargement sur la page du projet [2]. Les candidats constructeurs peuvent aussi y commander le contrôleur préprogrammé et le circuit imprimé correspondant.

(100696 – version française : Robert Grignard)

nombre de fréquences, disons de 1 à 40, qui ne doivent pas être écrasées, on peut alors régler 40 pour M et N. Le potentiomètre de silencieux doit avoir été réglé au mieux pour un usage normal.

Allumez le scanner tout en appuyant sur S2, la LED D3 (*Hold*) s'allume. Le scanner va maintenant balayer quelques fois les 137 000 – 108 000 / 25 = 1 160 canaux. Le nombre de fois est réglable par la commande P. Ensuite, les fréquences trouvées seront programmées sous les numéros 40 à 80. La LED D3 s'éteint. Relancer le scanner permet de voir le résultat au moyen de la commande S. Au cours d'un essai à une distance d'environ 50 km d'un aéroport, avec P=6 et 20 fréquences, en une demi-heure nous avons trouvé automatiquement à peu près toutes les fréquences, la tour, l'approche, les départs, le radar et d'autres d'utilisation courante. Elles sont immédiatement triées en fonction des plus employées et mises en mémoire à partir de la 1^{re} place.

On peut aussi utiliser le scanner comme récepteur ordinaire avec la commande T. Tapez T 122400<Entrée>, le récepteur se syntonise dessus et sa fréquence apparaît sur le terminal, suivie de la force du signal, entre 1 et 9.

Modes

Scanner :

Après la mise en marche du scanner, le balayage s'effectue à partir de la mémoire. La LED D4 (*Squelch*) s'allume dès qu'un signal dépasse le seuil du silencieux et D3 s'allume quand le balayage s'arrête pour une période d'écoute. Quand le signal disparaît, D4 s'éteint, mais l'écoute reste possible pendant l'attente. La LED D3 ne s'éteint qu'à l'issue de la période d'attente, quand le balayage reprend. Pour rester bloqué sur la fréquence, il suffit de ramener à zéro le seuil du silencieux.

Au cours du balayage, le numéro de la mémoire de fréquence est continuellement envoyé au terminal et si un signal permanent interrompt le parcours, on peut l'en empêcher en poussant sur S3, ce qui le stigmatise en mémoire en plaçant un B derrière sa fréquence. Seul le terminal pourra l'effacer.

Récepteur :

Après l'allumage, maintenir enfoncé pendant au moins 1 s le bouton S3 fait passer le scanner en mode récepteur. La syntonisation s'opère alors au moyen des boutons S1 vers le bas et S2 vers le haut de gamme ; le silencieux fonctionne normalement. La LED D3 s'allume quand on est arrivé en bout de gamme. Lors d'un démarrage en mode récepteur, la fréquence de départ est la plus basse.

Le bouton S3 gagne alors une autre compétence : celle de faire monter la fréquence d'accord jusqu'à la découverte d'un signal supérieur au niveau de silencieux. La recherche s'arrête et la LED D3 s'allume. On appuie de nouveau sur S1 pour continuer la recherche, ce dont témoigne l'extinction de D3. Arrivée en bout de gamme, la recherche reprend depuis la fréquence minimale.

Mode de recherche :

Après l'allumage, maintenez le bouton S2 enfoncé pendant au moins 1 s, la LED D3 s'allume et le récepteur se met à rechercher des fréquences dont l'activité est variable. Vous filtrez ainsi les signaux constants, la bande est scrutée par pas de 25 kHz pour trouver les émetteurs qui ont une activité intermittente.

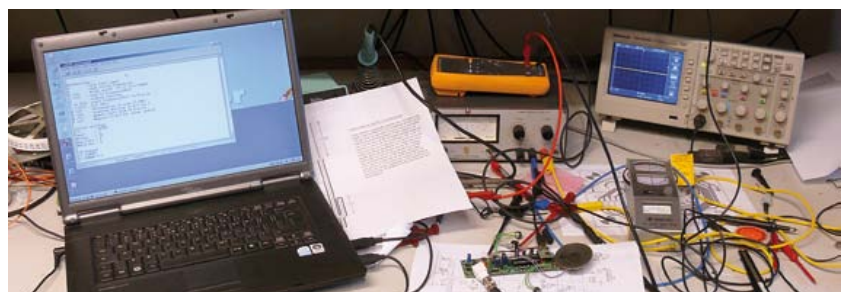
Le nombre de parcours de la bande à effectuer se règle par la commande P sur le terminal. Une passe de balayage prend 7 minutes. Si P=2, l'opération dure 14 minutes. Une bonne solution, c'est de procéder à 6 balayages, donc en 42 minutes ; le mieux, c'est avec P=15, mais alors il vous faudra attendre une heure trois quarts.

Quand la recherche est terminée, les fréquences trouvées sont programmées en mémoire, puis la LED D3 s'éteint et il faudra rallumer le récepteur. Il y aura N fréquences programmées à partir de la position M en mémoire (cf. commandes M et N). Avant de lancer une recherche avec le scanner, il faut convenablement régler le potentiomètre de silencieux à l'aide de la LED D4. Après rallumage, on balaye les résultats. Ils ne seront inscrits en EEPROM qu'après l'extinction de la LED D3, donc une coupure prématurée n'a pas d'effet.

Liens

[1] www.elektor.fr/bob

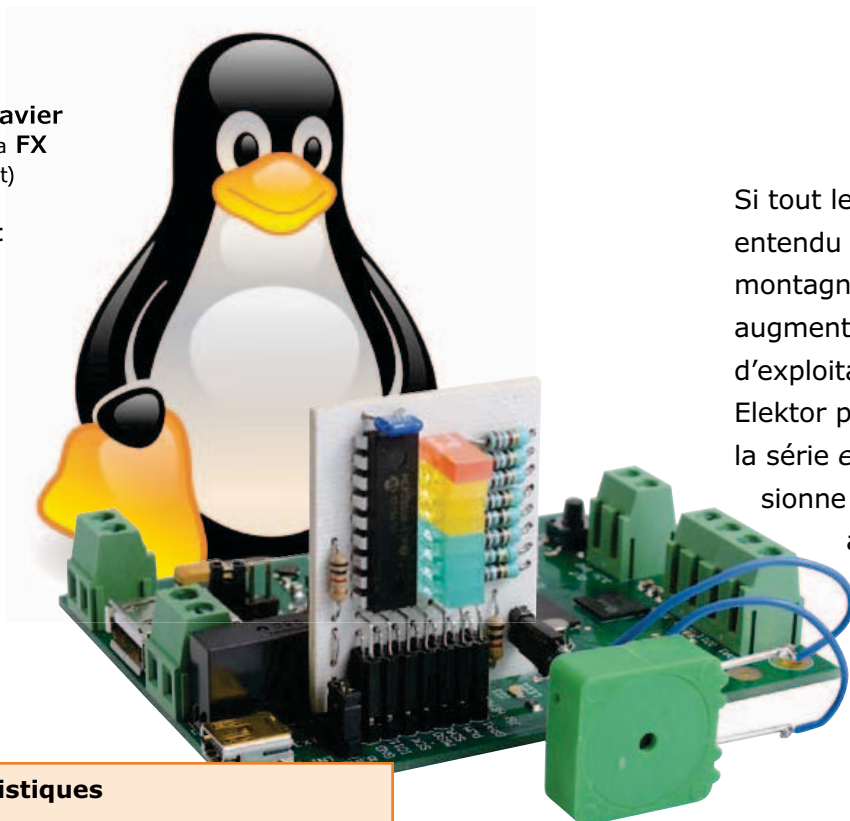
[2] www.elektor.fr/100696



HangTux

jeu de pendu sur la carte embarquez Linux ! d'Elektor

François-Xavier
Maurille aka FX
(Nieul-le-Dolent)
&
Kévin Petit
(Cambridge)



Caractéristiques

- Plusieurs dictionnaires disponibles
- Possibilité de choisir le niveau de difficulté
- Affichage du nombre d'erreurs sur un VU-mètre

d'Elektor. Il a commencé par un projet de batterie électronique qui lui tenait à cœur, mais qu'il a dû mettre de côté parce qu'il n'obtenait pas les performances attendues ; c'est alors qu'est née l'idée de ce *jeu de pendu*.

Tous les potaches savent que le jeu du pendu consiste à faire deviner, lettre par lettre, un mot au joueur. Celui-ci doit commettre le moins possible d'erreurs dans le choix des lettres. Chaque fois qu'il en propose une qui ne figure pas dans le mot à deviner, on dessine un morceau de potence ou un membre du pendu. Si le joueur trouve le mot avant la fin du dessin, il gagne. Si le dessin

Dans le cadre de son stage au labo d'Elektor [1], une des tâches de François-Xavier a été d'étudier une application pour la carte Linux

Si tout le monde (ou presque) a déjà entendu parler de Linux, il reste des montagnes de préjugés à déplacer pour augmenter la diffusion de ce système d'exploitation. Depuis plusieurs mois, Elektor prend sa part de cet effort avec la série *embarquez Linux !* [3] qui passionne les lecteurs. Voici une petite application concrète de la carte Linux associée, pour montrer que le manchot n'est pas si farouche qu'on peut le craindre.

est fini avant que le joueur ne devine, celui-ci est pendu haut et court.

Ici, plusieurs niveaux de difficulté sont disponibles et, renommée internationale d'Elektor oblige, différents dictionnaires vous permettront de jouer au choix dans votre propre langue ou dans une autre. L'interface utilisateur est un terminal Linux et tout a été fait pour rendre le jeu facile et agréable à utiliser.

Le matériel

Afin d'améliorer l'interface avec l'utilisateur et de montrer les possibilités de la carte *embarquez Linux !*, FX a décidé de développer une petite extension (**fig. 1**). Pour représenter *le pendu*, l'idée initiale était de dessiner avec des LED le contour du manchot Tux, la mascotte de Linux qui n'est décidément pas un pingouin. Sympathique, mais irréalisable en raison du nombre

élevé de LED et de la forte consommation de courant. Comme il fallait ménager la possibilité d'alimenter la carte par le port USB, le choix s'est porté sur une simple barre de LED (comme on les utilise pour la fonction de VU-mètre) qui permette d'afficher le nombre d'erreurs.

Comme le nombre de lignes d'entrées/sorties polyvalentes (GPIO) disponibles sur le connecteur d'interface de la carte Linux lui paraissait limité, FX a opté pour une solution d'extension de ces lignes avec une puce communiquant par le bus SPI. Les modules SPI et GPIO de la carte Linux seront donc utilisés.

La sortie MLI de la carte a également été rendue accessible via deux broches (MLI et masse) sur la carte d'extension pour produire, par exemple, du son sur un résonateur, mais cette fonction n'est pas réalisée dans la version présentée ici. Excellente occasion pour vous de vous exercer. La puce d'extension choisie est un MCP23S08 de *Microchip*, un modèle à 8 bits, simple d'utilisation, très abordable et disponible en boîtier DIP18, ce qui a facilité le prototypage. Le dialogue avec le MCP23S08 consiste à échanger des mots de 24 bits sur le bus SPI. Le premier octet permet d'adresser la puce et de préciser si l'on souhaite effectuer une opération de lecture ou d'écriture. Ici il vaudra toujours 0100 0000 ; l'accès se fera toujours en écriture et les deux bits d'adresse A1 et A0 valent tous deux 0 (les broches 4 et 5 du circuit sont reliées à la masse). L'octet suivant correspond à l'adresse du registre dans lequel on souhaite écrire. Notre carte d'interface n'utilise que les registres IODIR (*I/O Direction*) permettant de configurer les broches en sortie et OLAT (*Output Latch*) chargé de verrouiller le niveau logique désiré pour les broches de sortie. Le signal CS de sélection de la puce est commandé par la broche IO11 de la carte Linux.

Le logiciel

Le logiciel qui anime ce projet est écrit en langage C. Pour rattraper le temps perdu avec le projet initial de batterie choisi comme application de la carte *embarquez Linux !*, l'auteur a en fait recyclé un projet antérieur, non sans quelques améliorations comme les listes de mots personnalisées ainsi que les niveaux de difficulté.

Pour que le jeu puisse être joué en plusieurs langues, FX a choisi de stocker les mots dans des dictionnaires différents, un par langue. Ceux-ci prennent la forme de fichiers texte à l'extension .dict, contenant un mot par ligne (100 mots maxi-

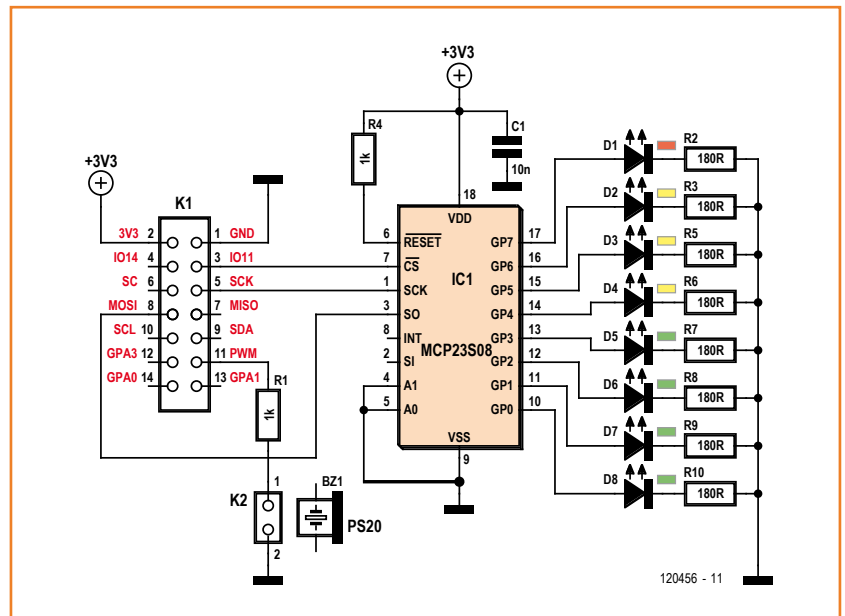


Figure 1.
Schéma de la carte d'affichage.

mum par dictionnaire). Pour créer vos propres dictionnaires, il vous suffit de placer votre fichier .dict dans le répertoire *dictionary*. Le fichier *def.dict* est utilisé pour stocker le dictionnaire par défaut. Pour commander les lignes GPIO de la carte *embarquez Linux !* d'Elektor, plusieurs solutions existent. La première (et la plus rudimentaire) consiste à utiliser la commande *echo* de l'interpréteur de commandes afin d'écrire les valeurs désirées dans l'entrée adéquate du système de fichiers. Utilisé depuis un programme C, ce n'est ni élégant, ni performant, ni pratique et surtout pas très fiable. On pourrait également utiliser la fonction *fprint* sur ce même fichier, mais ce n'est guère mieux.

Liste des composants

Résistances :

R1, R4 = 1 kΩ
R2, R3, R5 à R10 = 180 Ω

Condensateurs :

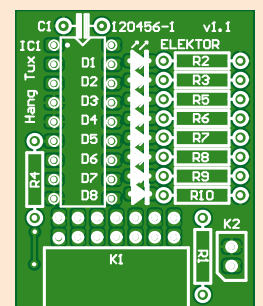
C1 = 10 nF

Semi-conducteurs :

D1 = LED rectangulaire rouge 5 mm
D2 à D4 = LED rectangulaire jaune 5 mm
D5 à D8 = LED rectangulaire verte 5 mm
IC1 = MCP23S08

Divers :

P1 = barrette sécable à deux points au pas de 2,54 mm
K1 = barrette sécable double rangée coudée au pas de 2,54 mm, 7 x 2 points



Il existe en revanche sous Linux un fichier spécial nommé `/dev/mem` qui permet d'accéder directement à la mémoire du système, à condition, bien sûr, que votre programme ait les permissions adéquates. L'utilisation courante qui en est faite depuis un programme C consiste à ouvrir ce fichier et à en effectuer une projection en mémoire à l'aide de la fonction `mmap`. Ce faisant, vous obtenez un pointeur qui, si vous le déréférez, vous permettra d'accéder à n'importe quel emplacement de la mémoire. Voici par exemple comment écrire la valeur `val` à l'emplacement mémoire `offset` :

```
*(ptr + offset) = val;
```

Avec cette technique, plus performante et plus sûre que les deux précédentes, FX accède au registre `IOCONFIG` de configuration des GPIO. Le positionnement à l'état haut ou bas des différentes E/S est obtenu de la même façon, en accédant aux registres correspondants.

Côté SPI, c'est l'appel système `ioctl` qui est utilisé sur le fichier spécial correspondant au bus SPI (/

`dev/spi0`), notamment pour configurer le nombre de bits par mot. L'écriture de données sur le bus est effectuée par l'appel système `write`.

Pour ce qui est des graphismes, on choisira une des deux formes disponibles : un homme ou bien alors, Linux oblige, le manchot Tux (voir le fichier `graphic.c` du code source [2]).

Mise en route et utilisation

Côté matériel c'est très simple puisqu'il suffit de relier, hors tension bien sûr, la carte d'extension à la carte *embarquez Linux !* d'Elektor par l'intermédiaire du connecteur d'extension.

Avant de lancer le logiciel pour la première fois, il vous faudra créer le fichier spécial `/dev/mem` lui permettant d'accéder aux registres GPIO. Pour ce faire, il vous suffit de taper les commandes suivantes dans un terminal :

```
mknod -m 660 /dev/mem c 1 1
chown root:kmem /dev/mem
```

Vous pourrez ensuite, toujours depuis un terminal, lancer le jeu. Placez-vous tout d'abord dans

Listing 1. Algorithme du jeu

Précondition : - carte d'extension correctement configurée

Début

SI aucun dictionnaire n'a été choisi ALORS

Charger le dictionnaire par défaut " dictionnaire/def.dict "

Fin SI

Boucle TANT QUE l'utilisateur veut jouer

Choisir un mot au hasard parmi ceux du dictionnaire chargé et le masquer suivant le niveau

Boucle TANT QUE le mot n'a pas été trouvé OU que le nombre d'erreur n'a pas atteint 7

Demander une lettre au joueur

SI la lettre n'a pas été proposée auparavant ALORS

SI la lettre est présente dans le mot ALORS

Afficher la lettre dans le mot

SINON ALORS

Incrémenter le nombre d'erreurs

Fin SI

Fin SI

Rafraîchir l'affichage et le VU-mètre en fonction des erreurs

Fin Boucle

Afficher un message suivant si le mot a été trouvé (GAGNÉ) ou non (PERDU)

Proposer une autre partie

Fin Boucle

Fin

le dossier où vous aurez mis le binaire compilé ou bien téléchargé à partir de notre site [2]. Vous pourrez ensuite lancer le jeu à l'aide de la commande `./hang_tux`.

Une fois lancé, le menu principal s'affiche et permet de démarrer une partie, de changer de dictionnaire ou de niveau de difficulté ou enfin de quitter le programme.

Lorsque l'utilisateur choisit de commencer une partie, le dictionnaire est chargé dans un tableau en mémoire (si ce n'est pas déjà fait) et un mot est choisi aléatoirement. En fonction du niveau de difficulté, le mot est plus ou moins masqué. La partie peut alors commencer. Le joueur propose une lettre qu'il tape avant de valider son choix à l'aide de la touche Entrée. Si cette lettre avait déjà été choisie, le message *key already pressed* apparaît. Dans le cas contraire, le logiciel vérifie si la lettre apparaît dans le mot. Si c'est le cas, le joueur peut saisir une autre lettre. Dans le cas contraire, des messages amusants seront affichés, le compteur d'erreurs incrémenté, l'homme ou le manchot sera (re)dessiné et l'état du VU-mètre mis à jour. La partie s'arrête soit par pendaison après sept erreurs (**fig. 2**) soit quand le joueur gagne en devinant le mot caché. Dans un cas comme dans l'autre, il y a fort à parier que la plupart des joueurs voudront remettre ça, encore et encore... Vous trouverez l'algorithme du jeu dans le **listing 1**.

Dans le menu principal, l'utilisateur peut choisir le dictionnaire qu'il souhaite utiliser. La présence du fichier `X.level` indique que l'utilisateur a choisi le niveau X. Grâce à ce fichier, le jeu se souvient du niveau choisi par l'utilisateur.

Modeste et encourageant

Nous espérons que ce modeste jeu vous aura montré qu'il est facile de développer pour la carte *embarquez Linux !* d'Elektor et vous encouragera à concrétiser vos propres projets. La place de Linux dans le monde de l'électronique ne cesse d'augmenter, alors ne loupez pas le coche et rendez-vous sur Elektor Projects [4] !

(120456)

Liens et références

- [1] www.elektor.fr/120609
 [2] www.elektor.fr/120456
 [3] www.elektor.fr/120026
 Embarquez Linux (1), Elektor n°407 mai 2012
 [4] <http://www.elektor-projects.com/>



Figure 2.
Capture d'écran du terminal
après une partie perdue.

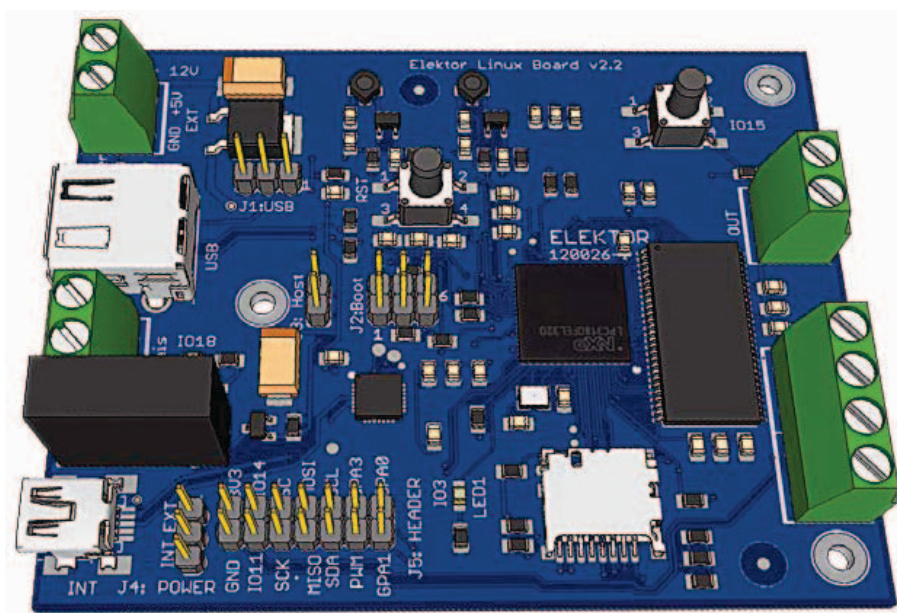


embarquez Linux ! (7)

I²C, UART, RS485

Benedikt Sauter [1] Chers lecteurs linuxiens, nous vous avons promis de consacrer cet article (en principe le dernier de la série) à des sujets choisis par vous. Or vous avez été nombreux à répondre avec pertinence à notre questionnaire, comme en témoigne indirectement la longueur de cet article. Nous passerons en revue les outils offerts par Linux pour développer un programme en C, puis nous verrons comment configurer et exploiter des interfaces série comme I²C et UART, souvent indispensables au fonctionnement d'un projet.

Modèle 3D de la carte Linux-Elektor, dessiné avec l'outil EagleUp [14].




```
wget http://gnublin.org/downloads/eldk-eglibc-i686-arm-toolchain-qte-5.2.1.tar.bz2
```

Pour extraire l'archive téléchargée dans le répertoire racine, entrez :

```
sudo tar xjf eldk-eglibc-i686-arm-toolchain-qte-5.2.1.tar.bz2 -C /
```

Pour pouvoir lancer le compilateur depuis n'importe où, nous allons écrire un script shell qui affectera automatiquement à la variable d'environnement PATH le répertoire de la nouvelle installation. La commande ci-dessous crée et ouvre le fichier *set.sh* dans le dossier */opt/eldk-5.2.1* :

```
sudo gedit /opt/eldk-5.2.1/set.sh
```

Copiez-y les lignes de la **figure 1**, profitez-en pour noter la façon dont les chemins d'accès ont été ajoutés à la variable PATH, puis enregistrez le fichier. Ce script est à appeler avant chaque compilation sur le PC d'un programme écrit pour la carte Linux-Elektor. Respectez l'espace entre le point et la barre oblique lorsque vous l'appellez :

```
. /opt/eldk-5.2.1/set.sh
```

Le nouveau compilateur de la chaîne d'outils peut alors être lancé :

```
arm-linux-gnueabi-gcc -v
```

Environnement de développement C/C++

Plusieurs lecteurs ont souhaité disposer d'un environnement de développement intégré de type Eclipse pour la programmation en C/C++. Voici en quelques étapes simples comment écrire un programme en C ou C++ sur l'ordinateur et le tester sur la carte.

Les outils nécessaires :

- la console de la carte Linux-Elektor (accès par UART ou par réseau)
- un système de fichiers en réseau (pour l'accès à partir du PC de développement)
- un environnement de développement, en option bien sûr.

Dans une arborescence de répertoires Linux, les périphériques sont représentés par des fichiers.

Un support de mémoire distant pourra donc être monté dans cette arborescence comme le seraient un dossier ou un sous-dossier. Pour ce montage, Linux offre (bien sûr) plusieurs possibilités, par exemple NFS, FTP, ou encore Samba.

Dans le dernier article [4], nous avons utilisé SSH pour établir une connexion réseau à la carte. SSH est un programme client-serveur qui permet d'accéder par réseau à la console de la carte. SSH est comparable à Telnet, mais plus sécurisé puisque le protocole de communication SSH crypte la liaison client-serveur. Autre avantage de SSH, la simplicité avec laquelle se font les échanges de fichiers entre client et serveur. Côté utilisateur, nous nous servons du programme *scp* (*Secure Copy Protocol*).

À supposer que la carte Linux possède l'adresse IP 192.168.0.190, la commande suivante transfère directement le fichier *test.c* du PC au système de fichiers de la carte Linux :

```
P1=/opt/eldk-5.2.1/armv5te/sysroots/i686-eldk-linux/usr/bin/armv5te-linux-gnueabi/
P2=/opt/eldk-5.2.1/armv5te/sysroots/i686-eldk-linux/bin/armv5te-linux-gnueabi/
export ARCH=arm
export CROSS_COMPILE=arm-linux-gnueabi-
export PATH=$P1:$P2:$PATH |
```

Figure 1.
Les variables d'environnement pour la nouvelle chaîne d'outils.

```
scp test.c root@192.168.0.190:/root
```

Et pour transférer tout un dossier :

```
scp -r myproject root@192.168.0.190:/root
```

Le premier argument de la commande *scp* est le nom du fichier ou du dossier à transférer. Dans le cas d'un dossier, l'argument *-r* demande à *scp* d'effectuer une copie récursive des sous-dossiers et fichiers présents. Vient ensuite le nom de l'utilisateur, suivi d'un @ et de l'adresse IP. Après le double-point est indiqué le répertoire de destination sur la carte Linux.

Vous pourriez ainsi effectuer le transfert de tous vos programmes compilés, mais il y a plus élégant encore !

Le linuxien avisé préférera en effet *sshfs* à *scp*. Pour l'installer sur votre PC, lancez :

```
sudo apt-get install sshfs
```

Le programme *sshfs* va nous servir à monter le système de fichiers de la carte Linux sur celui du PC. Plus besoin alors de copier à la main les fichiers sur la carte SD de la carte Linux-Elektor. Comme pour SSH ou *scp*, une connexion réseau doit bien sûr être présente.

Sur le système de fichiers du PC, créons le point de montage auquel nous accrocherons le système de fichiers distant. Comme point de montage, nous choisissons un dossier vide, à créer avec un gestionnaire de fichiers graphique ou, notre méthode préférée, avec *mkdir*, depuis la console.

Plaçons-nous d'abord dans le répertoire utilisateur :

```
cd ~
```

Créons le dossier vide qui servira de point de montage :

```
mkdir elektor-linux-board
```

Nous sommes prêts à monter le système de fichiers de la carte Linux sur le PC :

```
sshfs root@192.168.0.190:/
elektor-linux-board
```

Le système demande une nouvelle fois notre mot de passe d'utilisateur. Nous sommes alors supposés pouvoir explorer l'ensemble du système de fichiers à l'aide des commandes *ls* et *cd*, mais pour certains répertoires nous nous heurtons à un « Permission denied ». Logique, les droits d'accès du système de fichiers sont ceux de l'utilisateur *root*, et nous travaillons sur le PC en tant que simple utilisateur, une identité déterminée en substance par notre UID, ou *User ID*, soit notre identifiant utilisateur. Pour naviguer sans restriction dans le système de fichiers que nous venons de monter, nous devons donc nous connecter en tant qu'administrateur (ou utilisateur *root*). Inutile pour cela de changer de session, nous pouvons lancer une console qui dispose des droits *root* :

```
sudo /bin/bash
```

Promus administrateur, nous pouvons désormais naviguer librement dans le système de fichiers monté, y copier-coller des fichiers ou en supprimer.

Écrire sur le PC des programmes destinés à la carte est maintenant très simple : déposez tout bonnement le dossier de votre projet dans le système de fichiers. Une page du Wiki [5] du site GnuBlin explique comment développer des programmes C/C++ sous Eclipse.

Avant chaque nouveau projet, une bonne habitude à prendre est d'ajouter un utilisateur sur la carte Linux-Elektor :

```
adduser elektor
```

Vous n'avez ensuite plus qu'à monter le répertoire */home* de cet utilisateur au point de montage *elektor-linux-board* :

```
sshfs elektor@192.168.0.190:/home/elektor
elektor-linux-board
```

Si vous ne partagez votre PC et la carte avec personne d'autre, les deux utilisateurs recevront alors automatiquement l'identifiant UID 1000, et il n'y aura jamais aucun problème de droits d'accès dans ce répertoire. Néanmoins et bien sûr, rien n'interdit d'exécuter un programme en tant que *root* dans certains cas.

Mise à jour du noyau

Le noyau sert d'interface entre les applications et le matériel. Des modules de noyau trop anciens ou inexistantes peuvent ne pas reconnaître un matériel récent ou particulier, aussi doit-on parfois télécharger puis compiler une version plus récente du noyau.

Pour la carte Linux-Elektor, la source première des nouvelles versions est l'archive du noyau du projet GnuBlin. Des modules ont été ajoutés depuis la parution du premier article de cette série, notamment CAN, SPI et I²C pour le dialogue avec les périphériques.

Le code source est conservé dans un système de gestion de versions. La structure est claire et l'accès aux fichiers simple, les développeurs peuvent donc y déposer rapidement leurs codes modifiés. La gestion des archives est prise en charge par le logiciel GIT, qu'il vous faudra installer avant de pouvoir accéder à un dépôt GIT :

```
sudo apt-get install git-core
```

Pour « cloner » l'archive du dépôt GIT, c'est-à-dire en obtenir une copie locale, placez-vous dans un dossier quelconque et lancez :

```
git clone http://code.google.com/p/
gnublin-develop-kernel/
```

Le téléchargement peut prendre un certain temps. Une fois terminé, le curseur clignote et attend une nouvelle entrée. Pour « construire » la dernière version stable du noyau pour la carte, placez-vous dans le répertoire du noyau 2.6.33 (ou, si vous aimez les frissons, expérimentez avec la version 3.3.0 du noyau) :

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

Ce dossier contient un fichier de configuration standard :

```
make elektor_defconfig
```

Lancez l'outil de configuration du noyau si vous préférez activer ou désactiver vous-même certains modules :

```
make menuconfig
```

Compilez enfin le noyau avec la commande (sans oublier d'appeler le script *set.sh*) :

```
make zImage
```

Si le noyau a été créé sans erreur, transférez-le sur la carte SD en remplaçant dans la commande ci-dessous LABEL_SD_CARD par le nom du dossier correspondant :

```
sudo cp arch/arm/boot/zImage /media/
LABEL_SD_CARD/
```

Les modules se compilent avec la commande :

```
make modules
```

Installez ensuite ces modules dans un répertoire temporaire :

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

Pour finir, copiez sur la carte SD les modules stockés dans ce répertoire :

```
sudo cp -r /tmp/lib /media/
LABEL_SD_CARD/
```

Outils I²C

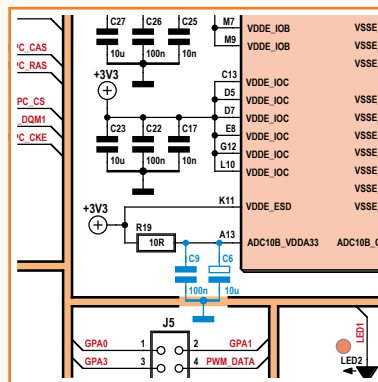
Dans le cercle des utilisateurs de microcontrôleurs, I²C est depuis des décennies un bus très apprécié. La gamme des composants qu'il dessert va du convertisseur A/N au duplicateur de ports d'E/S, en passant par le capteur de température. Pour l'utiliser sous Linux, nous devons activer dans le noyau les pilotes correspondants :

Device Drivers → I2C support → I2C

Hardware Bus support → I2C bus support
for Philips PNX targets

Des mesures de tension plus précises

Après avoir vu dans le cinquième volet de la série [15] comment se servir du convertisseur A/N pour mesurer des tensions, quelques lecteurs nous ont fait remarquer que la tension de 3,3 V de la carte, utilisée comme tension de référence par le convertisseur A/N, n'était pas assez stable pour obtenir des mesures précises. Un remède simple consiste à placer derrière R19 deux condensateurs de 10 µF et 100 nF, et à les relier à la masse (**fig. 8** et **fig. 9**). Cet ajout stabilise un peu plus la tension et devrait permettre d'obtenir des mesures beaucoup plus précises.



En configuration standard du noyau, le pilote est déjà présent.

Les dépôts Debian contiennent des programmes pour tester I²C. Après avoir téléchargé le paquetage correspondant, nous le convertirons au format d'archive *tgz* afin de pouvoir copier son contenu sur la carte SD.

Commençons par télécharger le paquetage *i2c-tools* sur le PC :

```
wget http://ftp.de.debian.org/debian/
pool/main/i/i2c-tools/i2c-tools_3.0.2-5_
armel.deb
```

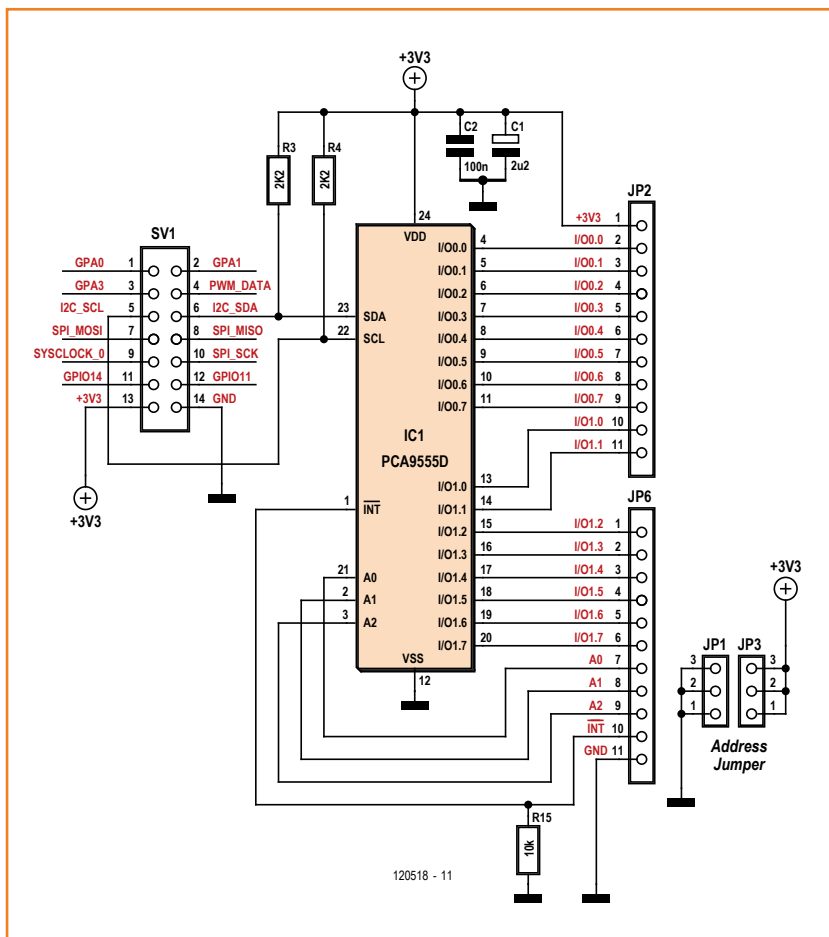



Figure 2.
Le schéma du duplicateur PCA9555 pour I2C.

Utilisons `alien -t` pour le convertir au format `tgz` :

```
alien -t i2c-tools_3.0.2-5_armel.deb
```

Le dossier où nous nous trouvons contient maintenant un fichier `i2c-tools-3.0.2.tgz`. Copions-le sur la carte SD :

```
cp i2c-tools-3.0.2.tgz /media/  
LABEL_SD_CARD/
```

Une fois la carte Linux redémarrée, nous devons décompresser l'archive `tgz`, puis créer les fichiers de périphériques qui serviront à tester I2C. Commençons par l'archive qui contient les outils Linux pour I2C :

```
cd /  
  
tar xvzf i2c-tools-3.0.2.tgz
```

Pour créer les fichiers de périphériques, nous utiliserons à nouveau la commande `mknod` [6]. Les numéros majeurs correspondant aux pilotes de périphériques chargés en mémoire sont donnés par :

```
cat /proc/devices
```

Vous devriez trouver le numéro majeur 89 à l'entrée correspondante. Adaptez les commandes suivantes si ce n'est pas le cas :

```
mknod /dev/i2c-0 c 89 0  
mknod /dev/i2c-1 c 89 1
```

Une fois les fichiers de périphérique créés et les programmes installés, vous pouvez brancher le premier périphérique I2C et dialoguer avec lui. Pour nos premiers essais, nous avons utilisé le composant PCA9555, un duplicateur de ports que nous avons relié au connecteur d'extension à 14 broches de la carte (**fig. 2**).

Nos essais nécessitent une alimentation de 3,3 V qui peut être prélevée sur le connecteur d'extension. Pour que les lignes du duplicateur puissent être mises au 3,3 V ou à la masse, fixez un cavalier directement sur les entrées A0-A2 du duplicateur. L'adresse du composant sera ainsi définie. Si la puce est correctement interconnectée et reliée à la carte Linux-Elektor via SDA et SCL, l'espace d'adressage I2C du bus peut être scanné avec l'outil `i2cdetect` :

```
i2cdetect 1
```

La sortie est reproduite sur la **figure 3**. L'adresse 0x6e provient de l'interface du LPC3131. Puisque dans notre circuit toutes les lignes d'adresse du PCA9555 ont été mises au niveau haut, l'adresse du composant est 0x27.

Pour vérifier rapidement et simplement le bon fonctionnement du bus I2C, nous avons relié trois LED au port 0 du duplicateur (**fig. 4**). Utilisons le programme `i2cset` pour envoyer un court message I2C via la ligne de commande. Définissons d'abord l'ensemble des broches d'E/S en tant que sorties :

```
i2cset 1 0x27 0x06 0x00
```

Puis plaçons les sorties au niveau haut :

```
i2cset 1 0x27 0x02 0xff
```

Les LED brillent, tout fonctionne.

Encouragés par la réussite de ce test effectué depuis la console, nous allons maintenant écrire un petit programme C qui commandera nos LED via le bus I²C.

I²C en C

Le LPC3131 offre deux interfaces I²C physiquement distinctes ; sur le connecteur à 14 broches, seul le second bus est disponible.

Les fichiers de périphérique correspondants sont respectivement `/dev/i2c-0` et `/dev/i2c-1`. L'accès au matériel se fait de la façon habituelle : ouverture du fichier de périphérique, puis accès au périphérique associé, en lecture ou en écriture. Le programme C du **listage 1** commande l'allumage des LED. Pour compiler son code sur le PC, entrez :

```
arm-linux-gnueabi-gcc -o i2ctest
i2ctest.c
```

Pour transférer le fichier compilé sur la carte Linux, vous avez maintenant le choix : la carte SD, `scp`, ou `sshfs`. Vous pouvez également compiler le fichier directement sur la carte Linux avec son compilateur `gcc`.

Pour appeler `i2ctest` depuis n'importe quel répertoire sans avoir à spécifier chaque fois son chemin d'accès complet, copiez le programme dans `/usr/bin`:

```
cp i2ctest /usr/bin
```

Pour lancer `i2ctest` sur la carte, entrez comme paramètre le nom du fichier de périphérique, une information destinée au bus I²C :

```
i2ctest /dev/i2c-1
```

Les LED s'allument et s'éteignent selon les instructions du code.

Étudions le code source (**listage 1**) : la fonction `open()` crée un descripteur de fichier pour le fichier de périphérique. La fonction `ioctl()` (un appel système qui sert à la configuration des pilotes de périphériques) passe au pilote l'adresse de périphérique actuelle du composant demandé. Si l'adresse est définie, un message I²C est envoyé au périphérique à l'aide de la fonction `write()`. Nous retrouvons ensuite la définition en tant que sorties des broches d'E/S

```
root@gnublin:~# i2cdetect 1
WARNING! This program can confuse your I2C bus, cause data loss and worse!
I will probe file /dev/i2c-1.
I will probe address range 0x03-0x77.
Continue? [Y/n]
  0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  27  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  6e  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
root@gnublin:~#
```

du duplicateur, comme vu plus haut avec `i2cset` :

```
buffer[0] = 0x06;
```

```
buffer[1] = 0x00;
```

La boucle sans fin `while (1)` contient les instructions pour allumer et éteindre les LED (avec une pause de 100 ms entre chaque appel).

On dialoguerait de la même façon avec d'autres composants présents sur le bus I²C. Les programmeurs avancés pourront se tourner vers la bibliothèque `libi2c` [7] et son accès très flexible au bus I²C. Cela dit, les fonctions `ioctl`, `write` et `read` du système l'exploitent déjà !

UART depuis la console

Nous avons vu comment installer un convertisseur USB/RS-232 dans le quatrième volet de la série [6]. Sous Linux, les interfaces UART peuvent être nommées de différentes manières. Le dialogue avec les interfaces UART du microcontrôleur se fait généralement avec les fichiers

Figure 3.
Sortie du programme
`i2cdetect`.

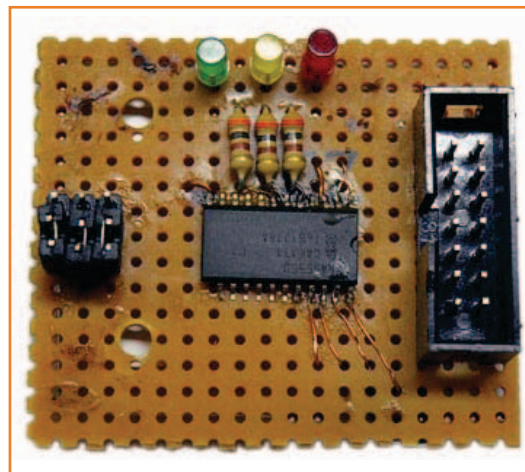


Figure 4.
Trois LED pour un petit test.

Listage 1 : Commande d'un duplicateur d'E/S I2C

```
#include <stdio.h>
#include <fcntl.h>
#include <linux/i2c.h>
#include <linux/i2c-dev.h>

//Slave Address
#define ADDR 0x27

int main (int argc, char **argv)
{
    int fd;
    char filename[32];
    char buffer[128];
    int n, err;

    if (argc == 0) {
        printf("usage: %s <device>\n", argv[0]);
        exit(1);
    }
    sprintf(filename, argv[1]);
    printf("device = %s\n", filename);

    int slave_address = ADDR;

    if ((fd = open(filename, O_RDWR)) < 0) {
        printf("i2c open error");
        return -1;
    }
    printf("i2c device = %d\n", fd);

    //prepare communication
    if (ioctl(fd, I2C_SLAVE, slave_address) < 0) {
        printf("ioctl I2C_SLAVE error");
        return -1;
    }

    write(fd, buffer, 2);

    //Port-0 GPIO as Output
    buffer[0] = 0x06;
    buffer[1] = 0x00;
    write(fd, buffer, 2);

    n = 0;
    while (1)
    {
        buffer[0] = 0x02; /* command byte: write
        output regs */
        buffer[1] = 0x00; /* port1 data */
        write(fd, buffer, 2);

        usleep(100000);
        buffer[0] = 0x02; /* command byte: write
        output regs */
        buffer[1] = 0xff; /* port1 data */
        write(fd, buffer, 2);

        printf("%d\n", n++);
        usleep(100000);
    }
}
```

de périphérique */dev/ttyS0*, */dev/ttyS1*, etc. La console standard, celle de l'invite de connexion et de tous les messages du système, a ainsi pour descripteur */dev/ttyS0*. Les interfaces USB/UART auront quant à elles des fichiers de périphériques appelés */dev/ttyUSB0*, */dev/ttyUSB1*, etc, mais quels que soient les noms attribués par le système aux interfaces UART, leur accès se fait toujours de la même façon. Ces fichiers de périphérique confèrent aussi une certaine indépendance à la plateforme Linux : on pourra par exemple développer ou modéliser entièrement sur PC une application USB/UART, puisque les convertisseurs USB/UART y seront aussi enregistrés en tant que */dev/ttyUSB0*, etc. Et si le programme est sur le PC, on peut le déplacer sur la carte Linux, de là lancer une compilation croisée, puis l'appeler et l'exécuter avec les paramètres adéquats.

Un fichier de périphérique absent peut être recréé avec *mknod*, toujours en chargeant préalablement le pilote correspondant :

```
mknod /dev/usb/ttyUSB0 c 188 0
mknod /dev/usb/ttyUSB1 c 188 1
mknod /dev/usb/ttyUSB2 c 188 2
```

Lors du quatrième volet, nous avons utilisé l'émulateur de terminal *microcom* pour communiquer avec l'adaptateur USB/série. La commande *microcom* prend pour paramètres le nombre de bauds et le nom du fichier de périphérique. Une fois lancé, l'émulateur affiche les caractères reçus : tout caractère alphanumérique entré au clavier est automatiquement transmis à *microcom* via l'interface série.

Évidemment, ce n'est pas un programme de ce genre que nous voulons utiliser maintenant. Ce que nous voulons, c'est envoyer et recevoir des données depuis la console. Ce n'est pas difficile, nous pouvons également accéder au fichier de périphérique `/dev/ttyUSB0` avec les simples instructions d'écriture et de lecture `cat` et `echo`. Nous devons toutefois d'abord trouver un moyen de paramétrer la vitesse de transmission. Linux dispose pour cela du programme `stty`. Pour définir par exemple un débit de 38400 bauds, on lancera :

```
stty -F /dev/ttyUSB0 38400
```

Nous nous sommes appuyés sur les exemples du lien [9] pour construire notre programme. La configuration de base se fait avec la structure de données `struct termios options` déclarée en début de programme. La fonction `tcgetattr` modifie la configuration courante contenue dans la structure de données qui lui est transmise. Le débit des caractères entrants est paramétré avec la fonction `cfsetispeed`, celui des caractères sortants avec `cfsetospeed`. L'initialisation des bits de départ, de stop, et de données se fait aussi à cet endroit du code. La fonction `tcsetattr` peut alors transmettre la nouvelle configuration au pilote.



Nous voici prêts à envoyer une suite de caractères :

```
echo "Salut les pingouins" > /dev/ttyUSB0
```

Et pour attendre des caractères :

```
cat < /dev/ttyUSB0
```

Jouez avec ces commandes pour bien les comprendre ! L'utilisation de protocoles simples, capables de transmettre et recevoir des caractères ASCII affichables, permet des essais et un travail rapides. Pour transmettre des valeurs binaires, nous avons besoin d'un court programme écrit en C.

UART en C

L'étape à franchir n'est pas si longue qu'il y paraît. Comme avec la communication établie depuis la console, nous devons d'abord configurer l'interface série, en particulier le débit de la transmission. La méthode de configuration que nous avons adoptée dans le programme en C du lien [8] (« Listage 2 ») est typique.

L'envoi et la réception des données fonctionnent comme avec les périphériques standard : ouverture préalable du descripteur de fichier avec `open()`, envoi des données avec `write()`, et réception avec `read()`. Dans notre exemple, la machine distante attend comme commande une valeur codée sur 6 octets. La variable `buffer[0]` reçoit par exemple un 0.

Le programme attend à son tour une réponse sur 6 octets de la part du périphérique connecté. La réception est ici traitée par une boucle qui lit six caractères. À l'intérieur de cette boucle, le programme attend l'arrivée du prochain caractère avec l'instruction `read` (appel bloquant). Cette façon de procéder n'est pas optimale, puisque les données ne sont transférées qu'à un débit de 19200 bauds. Durant l'exécution de cette boucle, notre rapide microcontrôleur ne fait quasiment rien d'autre qu'attendre bras croisés l'arrivée des caractères, mais accapare néanmoins un précieux temps de calcul.

En général, on préfère donc utiliser ici une interruption, c'est-à-dire accéder à la mémoire de réception dès notification de la présence de nou-

Sous les feux de la rampe, Benedikt Sauter répond aux questions d'Elektor sur le stand Farnell/element14 lors de l'Electronica 2012 à Munich. Benedikt présente sa carte dans la vidéo du lien [13].

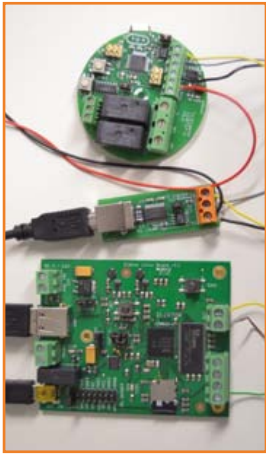


Figure 5.
Le module Relaiss pour tester RS-485 avec l'ElektorBus.

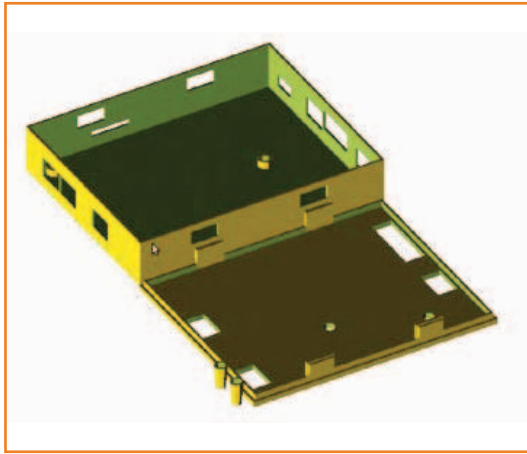


Figure 6.
Un boîtier qui attend son imprimante 3D.

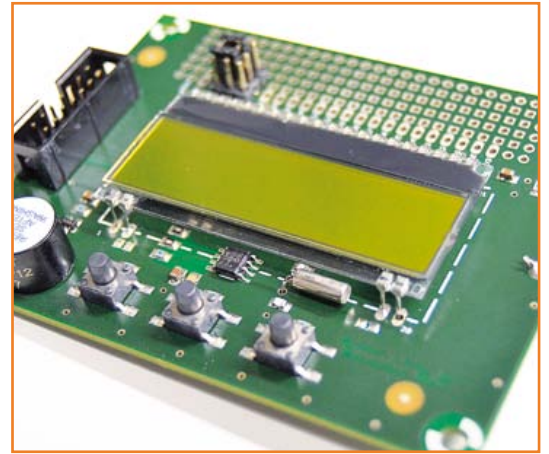


Figure 7.
La nouvelle carte, prochaine vedette du prochain numéro.

velles données. Avec un système d'exploitation comme Linux, le déroulement du programme s'avère alors bien plus économe en ressources : le programme fera autre chose si aucune donnée n'est présente, ou sera mis en pause via un appel système ; le programme ne sera réactivé qu'à l'arrivée de données, qu'il pourra alors traiter. Le troisième programme du dossier de téléchargement [8] implante une méthode de traitement par interruption. Le programme ressemble en grande partie à l'exemple précédent, si ce n'est qu'ici c'est une fonction `signal_handler_IO` qui est appelée après réception des données.

RS-485 avec le bus ElektorBus

Maintenant que nous savons comment travailler avec une interface UART, voyons comment envoyer et recevoir des octets via une interface RS-485. En lieu et place d'un convertisseur USB/RS-232, c'est cette fois-ci un convertisseur USB/RS-485 que nous devons relier à l'interface USB de la carte. Vous n'en avez pas ? Procurez-vous celui d'Elektor (lien [8], numéro 110258-91), le circuit est monté et testé. Le pilote de sa puce FTDI est déjà présent dans le noyau, référez-vous au quatrième volet de la série si vous avez oublié comment l'activer.

Comme machine distante, nous utiliserons la carte Relaiss [10], un module de commande de charges équipé de deux relais (fig. 5). Le programme *Listage 2* doit être adapté au protocole ElektorBus : débit de 9600 bauds, et variable

LENGTH initialisée à 16, entre autres petites modifications. Les 16 octets sont affectés au tableau `Buffer` avant l'envoi du message RS-485. La suite d'octets suivante commute le relais 1 du module :

```
170,0, 0,5,0,10, 96,1,0,0, 0,0,0,0, 0,0  
(coller)  
170,0, 0,5,0,10, 96,0,0,0, 0,0,0,0, 0,0  
(décoller)
```

La suite d'octets pour le relais 2 est :

```
170,0, 0,5,0,10, 0,0,96,1, 0,0,0,0, 0,0  
(coller)  
170,0, 0,5,0,10, 0,0,96,0, 0,0,0,0, 0,0  
(décoller)
```

Notez que le lien [11] vous offre un tour complet d'ElektorBus, agrémenté de nombreuses informations.

Perspectives

Vous pouvez désormais habiller votre carte Linux-Elektor avec un boîtier fabriqué par impression 3D (fig. 6). Son modèle 3D peut être téléchargé aux formats STL et SCAD depuis le site Thingiverse [12]. Rien à visser, corps et couvercle s'assemblent par simple encliquetage !

Votre série consacrée à l'embarqué sous Linux se termine ici, mais l'aventure se poursuivra puisque nous sommes loin d'avoir épuisé toutes nos idées

de projets. En attendant, nous avons conçu une nouvelle carte Linux-Elektor (**fig. 7**), enrichie de fonctions très utiles :

- afficheur (2 x 16 caractères)
- trois touches servant d'unité de commande ou de commandes par menus
- 16 entrées/sorties numériques supplémentaires
- horloge temps réel (RTC), avec alimentation par pile du système donnant l'heure
- vibreur sonore pour une sortie acoustique des messages
- laboratoire pour circuits d'extension.

Toutes ces nouveautés seront présentées en détail dans le prochain numéro !

(120518 – version française : Hervé Moreau)

Liens

- [1] sauter@embedded-projects.net
- [2] www.gnublin.org
- [3] www.elektor.fr/120180
- [4] www.elektor.fr/120578
- [5] <http://en.gnublin.org/index.php/Eclipse>
- [6] www.elektor.fr/120181
- [7] <http://opensource.katalix.com/libi2c/>
- [8] www.elektor-magazine.fr/120518
- [9] www.tldp.org/HOWTO/Serial-Programming-HOWTO/
- [10] www.elektor.fr/110727
- [11] www.elektor.com/elektorbus
- [12] www.thingiverse.com/thing:29314
- [13] www.element14.com/community/community/events/electronica
- [14] <http://eagleup.wordpress.com>
- [15] www.elektor.fr/120182

Publicité

Nouvelle édition revue et augmentée par l'auteur



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

44,50€

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

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

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



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

mettez le cap sur Arduino (4) surveillance d'arrosage de plantes



David Cuartielles (Suède/Espagne)

Invité à la *Rencontre d'Experts* organisée par Elektor, j'ai passé une journée entière à discuter avec d'autres experts et l'équipe d'Elektor de la manière de concevoir les projets et les sujets présentés dans la revue. Durant l'une de nos tables rondes, Jan, le rédacteur en chef de l'édition anglaise, nous met sous le nez un assortiment de composants disparates et lance à la cantonade : « Imaginez qu'un samedi après-midi vous rangez votre bazar et restiez avec ces composants sur les bras. Qu'en feriez-vous ? »

La discussion qui a suivi n'a pas donné grand-chose, mais de retour chez moi, j'ai repensé à tous ces composants oubliés. J'en ai des tas, moi aussi, souvent achetés pour tel ou tel projet, mais qui n'ont pas été utilisés dans le prototype final. J'ai ouvert un des cartons et y ai trouvé un afficheur à matrice de points et quelques boutons. J'ai repensé au défi de Jan et imaginé d'en faire un genre d'horloge. J'ai donc acheté à la boutique du coin une puce horloge temps réel puis assemblé le tout. Le résultat, que je vais décrire, est visible sur la photo ci-dessus.

Le matériel

Pour les expériences de ce mois-ci, il vous faudra :

- une carte Arduino Uno
- un shield de prototypage
- de la barrette sécable (mâle)
- un câble USB
- deux pushoirs
- un afficheur à matrice de points avec contrôleur HT1632 ; j'ai utilisé le modèle 32x16 de Sure

- une horloge temps réel (RTC) DS1302
- un résonateur à quartz de 32 kHz
- un support pour batterie CR2032
- un connecteur pour pile 9 V
- une pile de 9 V (pour l'Arduino) + une pile CR2032 (pour la RTC)
- un boîtier capable de contenir le tout

Remarque : J'ai construit ce projet dans une boîte en carton, mais je pense qu'il y a mille autres boîtiers possibles, de la boîte à casse-croûte au boîtier plastique de qualité pro... soyez créatifs !

Le cahier des charges : faire quelque chose d'utile

Je me demandais quoi faire d'utile avec quelques composants qui traînent chez soi quand j'ai pensé à la cage d'escalier de l'immeuble que j'habite, agrémentée par quelques plantes partagées avec les autres occupants. Quand je dis agrémentée, c'est théorique. Traduisez : personne n'arrose ces plantes régulièrement, elles ont donc soit trop d'eau, soit pas assez. Celles qui survivent n'ont pas fière allure, les unes pourrissent sur pied,

les autres sont desséchées. On est des braves gens et si nous n'arrosions pas ces plantes, ce n'est pas parce qu'on s'en moque, c'est parce que chacun pense que les autres le feront. J'ai donc eu l'idée d'un compteur de temps, ou un chronomètre si vous préférez, qui fournirait une indication visuelle de la dernière fois que les plantes ont été arrosées par « quelqu'un ».

Il ne fallait surtout pas que le projet soit complexe et je n'ai retenu que le facteur humain de ce problème d'arrosage, abandonnant l'idée de mesurer l'humidité de la terre ou quoi que ce soit d'autre de ce genre. Je ne voulais pas non de connexion à l'internet pour me rappeler d'arroser les plantes (ce que fait le projet Botanicalls [1] qui tweete un message quand une plante a cruellement besoin d'eau).

Juste un dispositif qui permette aux voisins de savoir la dernière fois que quelqu'un s'était occupé des plantes. J'ai alors pensé aux horloges utilisées pour les joueurs d'échecs. Après chaque mouvement, ils appuient sur un bouton de l'horloge pour passer la main à l'adversaire. J'ai imaginé une horloge qui offrirait le même type de fonctionnement : quand quelqu'un arrose la plante, il ou elle appuie sur un bouton et passe ainsi la responsabilité à quelqu'un d'autre. Un deuxième bouton permettrait d'afficher depuis combien de temps la plante n'a pas été arrosée. Il ne me restait qu'à rassembler les composants dont je pensais avoir besoin (fig. 1).

Commande de l'afficheur

Il n'y a finalement pas tellement de différences entre d'une part la commande d'une modeste LED de 5 mm comme celles que nous utilisons tous dans de petits projets et d'autre part la commande des énormes afficheurs lumineux que l'on voit maintenant dans les grandes artères des grandes villes. N'importe quel bricoleur qui fait clignoter sa première LED se dit ensuite qu'il pourrait faire la même chose avec deux, dix cent ou mille LED.

Il se trouve que ce n'est pas si simple que ça, car les microcontrôleurs ne disposent que d'un nombre limité de pattes qui ne peuvent fournir qu'une quantité limitée de courant. L'avantage d'un afficheur à matrice de points est justement de rassembler dans un boîtier plastique plusieurs LED dont les anodes (ou cathodes) ont été mises en commun ; cela les rend faciles à manipuler. Sur certains modèles, une puce de commande est

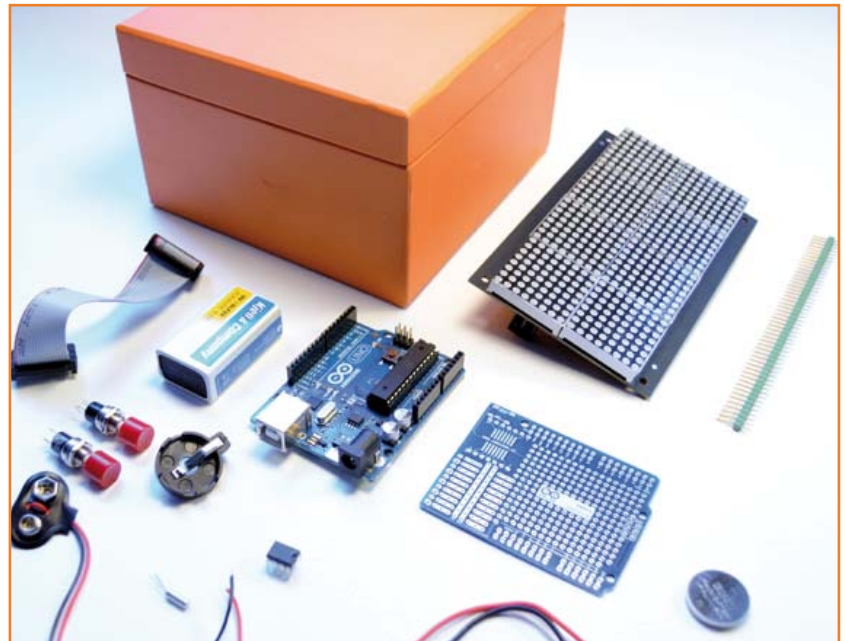
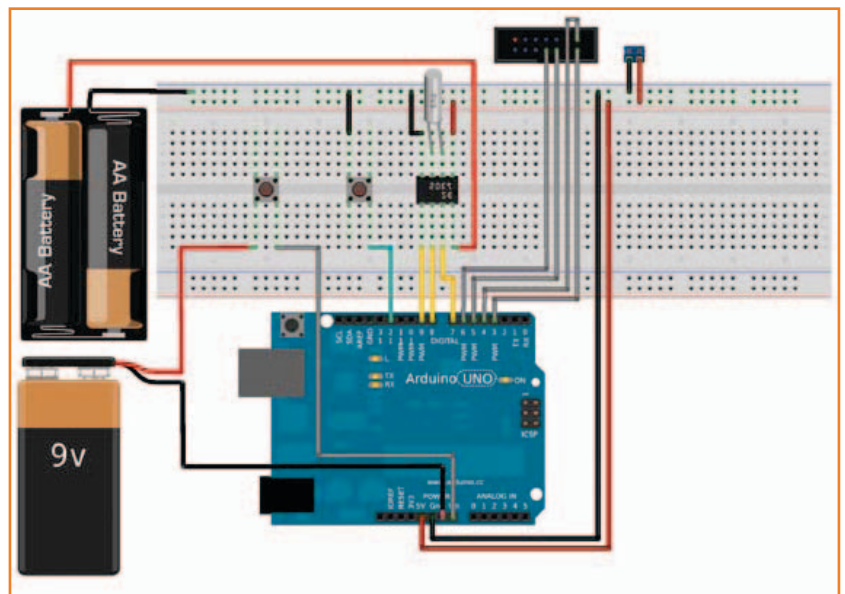


Figure 1.
Les composants de notre surveillant d'arrosage

même intégrée dans le boîtier. Vous trouverez le schéma de notre projet en **figure 2**.

Je possède quelques afficheurs de la marque Sure qui viennent de l'Arduino Store, prévus pour quelques expériences pour mes cours de l'an dernier, mais je suis sûr que vous pourrez les trouver chez d'autres vendeurs. Ils sont bien documentés en ligne et je vous recommande de mettre quelque chose de ce genre à votre répertoire. Cet afficheur commande 16x32 LED R/V, soit 512 LED bicolores, rouges ou vertes ou les deux à la

Figure 2.
Le schéma du projet



fois, ce qui donne de l'orange. Il est facile de mettre ces modules en cascade ; j'ai vu des projets en commander quatre d'affilée, cela fait 2048 LED ! Pour ce projet, puisque je souhaite n'afficher qu'une durée, je n'utiliserai qu'une seule des couleurs. Mais que cela ne vous empêche pas d'en faire plus. Jetons ensemble un coup d'œil aux possibilités de cet afficheur.

Installation de la bibliothèque de commande de l'afficheur

J'ai assemblé cette bibliothèque à partir du code de plusieurs auteurs, remanié en conformité avec le style d'une bibliothèque *Arduino*. J'ai également ajouté quelques petits exemples. La bibliothèque est compatible avec les versions 1.0 et ultérieures de l'EDI *Arduino*. Tout est prévu pour utiliser une horloge temps réel (RTC) assemblée

Listing 1.

```
#include <fonts.h>
#include <HT1632c.h>
#include <images.h>

HT1632c display(6, 5, 3, 4);

void setup() {
    display.setup();
}

void loop() {
    display.text("Hej", 5, 5);
}
```

Listing 2.

```
#include <fonts.h>
#include <HT1632c.h>
#include <images.h>

HT1632c display(6, 5, 3, 4);

void setup() {
    display.setup();
}

void loop() {
    display.image(Arduino_logo, 0, 0, 32, 16);
}
```

Listing 3.

```
#include <fonts.h>
#include <HT1632c.h>
#include <images.h>

#define heart_icon_width 17
#define heart_icon_height 16

unsigned char PROGMEM heart_icon[] = {
    0x00, 0x1e, 0x00, 0x3f, 0x80, 0x7f, 0xc0, 0x7f, 0xe0, 0x7f, 0xf0, 0x7f,
    0xf8, 0x3f, 0xfc, 0x1f, 0xfe, 0x0f, 0xfc, 0x1f, 0xf8, 0x3f, 0xf0, 0x7f,
    0xe0, 0x7f, 0xc0, 0x7f, 0x80, 0x7f, 0x00, 0x3f, 0x00, 0x1e };

HT1632c display(3, 4, 5, 6);

void setup() {
    display.setup();
}

void loop() {
    display.image(heart_icon, 0, 0, heart_icon_width, heart_icon_height);
}
```


sur le *shield* de prototypage (**fig. 3**).

Il vous faudra télécharger la bibliothèque depuis le site d'Elektor [4] et l'installer. Rappelez-vous que l'installation d'une bibliothèque dans l'EDI se fait en créant un dossier nommé *libraries* dans votre *sketchbook* et en y décompressant le fichier téléchargé. Vous devrez ensuite redémarrer l'EDI pour que la bibliothèque soit visible dans les menus.

Remarque : l'EDI *Arduino* est sur le point d'inclure un système permettant d'installer automatiquement les bibliothèques à partir du fichier compressé que l'on télécharge sur l'internet. Cependant, alors que j'écris cet article (décembre 2012), la fonction n'est pas encore disponible. En plus de la bibliothèque, vous disposerez d'une série d'exemples vous permettant de :

- vérifier que l'afficheur fonctionne correctement
- afficher de simples messages textuels
- faire défiler le texte
- charger des images

Pour accéder aux exemples, c'est très facile puisqu'il suffit d'utiliser l'entrée *File-> Exemples->HT1632c* du menu.

Pour utiliser la bibliothèque dans vos programmes, il vous faudra inclure trois fichiers d'en-tête : *fonts.h*, *HT1632c.h* et *images.h*. Il vous faudra également fournir au constructeur les numéros des pattes auxquelles est relié l'afficheur. L'ordre à respecter sera le suivant : *Data*, *Write clock*, *Chip Select* et *Clock*.

Configuration de l'afficheur

Ici, l'afficheur est relié aux E/S numériques 3, 4, 5 et 6. La bibliothèque permet d'utiliser n'importe quelle E/S numérique pour envoyer des données aux LED. Je vous rappelle que vous pourrez également mettre en cascade plusieurs afficheurs. Je n'ai fait d'essais qu'avec deux, ça fonctionnera sans problème avec trois ou quatre ou plus. L'afficheur est alimenté directement par le régulateur de l'*Arduino* ; lors de mes tests, je n'ai pas eu l'impression qu'il surchargeait la pile. Si vous prévoyez d'en utiliser beaucoup dans un seul projet, assurez-vous que votre source d'énergie tienne la route !

Maintenant, jetons un coup d'œil au **listing 1** qui correspond à l'exemple « *simple_text* » fourni avec la bibliothèque.

Par défaut, la méthode *text* utilisera la couleur

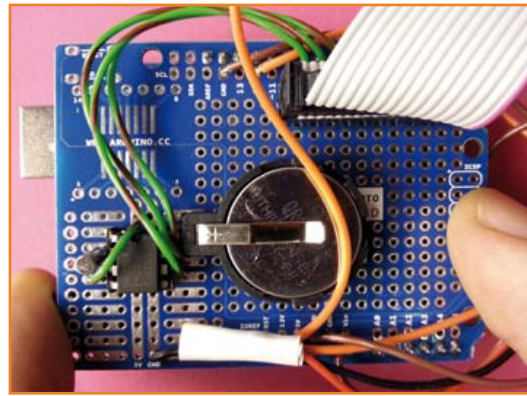


Figure 3.
Le shield de prototypage avec les composants



Figure 4.
Wahou ! Notre « écran » affiche le logo Arduino avec l'exemple « *Simple_image* ».

orange. Un quatrième paramètre vous permettra de choisir la couleur entre BLACK, GREEN, RED et ORANGE.

Afficher une image simple

Il est possible d'afficher des images de basse résolution sur ces afficheurs. L'image devra être stockée en mémoire programme. C'est ce que fait le code du **listing 2** correspondant à l'exemple « *Simple_image* » en chargeant l'icône par défaut fournie avec la bibliothèque : le logo *Arduino*.

Le résultat est visible sur la **figure 4**.

Pour afficher vos propres images, vous pourrez utiliser *Gimp*, un logiciel libre d'édition d'images qui tourne sur les principaux systèmes d'exploitation, pour transformer votre image en un tableau de nombres destiné à la mémoire programme de l'*Arduino*. Voici comment procéder :

- ouvrez l'image et redimensionnez-la à la taille de l'afficheur (32 colonnes x 16 lignes)
- faites la pivoter de 90° dans le sens horaire
- exportez-la en XbitMap
- ouvrez le fichier xbm produit et copiez le tableau

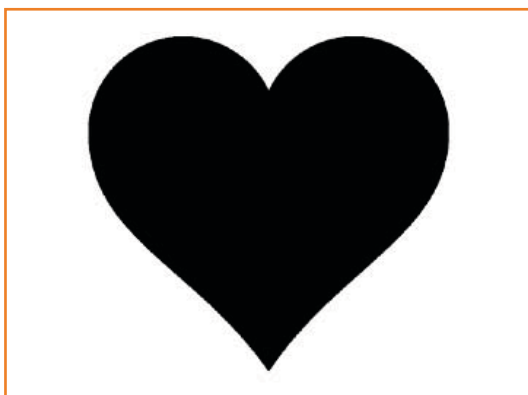


Figure 5.
Une image de cœur avant
le traitement permettant
de l'inclure dans un sketch
Arduino

collez-le dans votre programme en le faisant précéder de `unsigned char PROGMEM`, en d'autres termes, mettez-le dans la mémoire de programme pour ne pas consommer de la RAM inutilement.

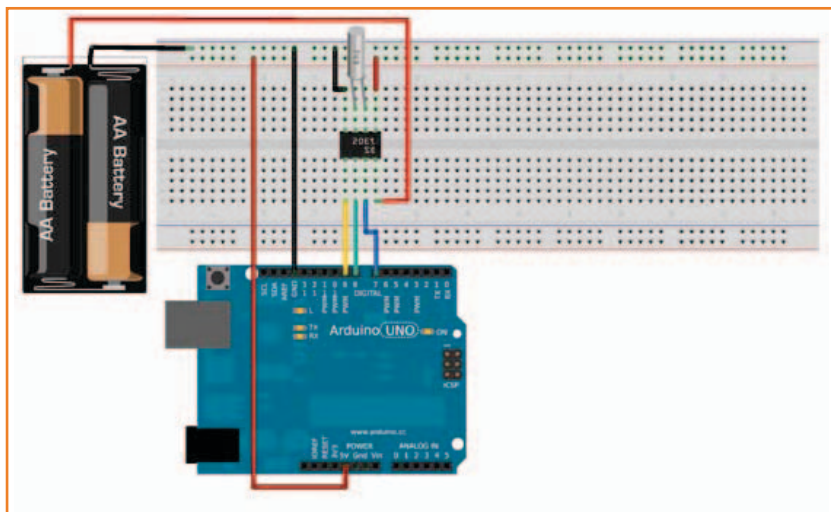
L'application de ces étapes sur l'image de la **figure 5**, vous donnera le tableau du **listing 3**.

Remarque : le fichier `xbm` est affiché un peu différemment sur l'afficheur. Vous devrez donc permuter la largeur et la hauteur quand vous utiliserez vos propres images. Dans mon cas, l'icône faisait 16x17, mais j'ai saisi 17x16 (largeur x hauteur).

Tuyau : vous pouvez ajouter plusieurs images ou icônes à votre programme ; il suffit de leur donner des noms différents.

La bibliothèque de commande de l'afficheur permet de faire beaucoup de choses, mais, pour ce projet, nous n'avons besoin que d'afficher un peu de texte et peut-être quelques images

Figure 6.
Voici comment relier la RTC
à l'Arduino



simples. N'hésitez pas à explorer les possibilités de la bibliothèque ; c'est un logiciel libre et les noms des fonctions devraient être suffisamment explicites. Passons maintenant à l'utilisation de l'horloge temps réel.

Les puces d'horloge temps réel

En utilisant un résonateur à quartz à 32 kHz, les RTC peuvent mesurer le temps avec une très bonne précision. Étant donné qu'elles doivent compter le temps en permanence, elles doivent rester sous tension, alimentées par une batterie. Pas d'inquiétude, elles ne consomment que peu et fonctionneront longtemps sur une batterie de 3 V, parfois même plusieurs années.

La puce choisie pour mon projet est une DS1302 de *Maxim*, une RTC très courante et bien documentée par la communauté *Arduino*. Je ne pourrais que vous conseiller de lire l'article écrit par l'utilisateur *Arduino* Krodal sur l'*Arduino Playground* [2] ; vous y trouverez également un code explorant en profondeur les capacités de la puce. Cependant, et par souci de simplicité, j'ai choisi d'utiliser une bibliothèque de Matt Sparks [3]. Jusqu'ici nous n'avons utilisé que quatre pattes de l'*Arduino Uno* pour relier l'afficheur ; la RTC nécessite trois E/S numériques et j'ai choisi les E/S 7, 8 et 9. Le schéma de la **figure 6** vous montrera comment relier la RTC à son quartz et à l'*Arduino*.

Installation de la bibliothèque RTC et réglage de l'heure

Vous pouvez obtenir une copie de la bibliothèque RTC depuis le site d'Elektor [4] ou depuis le projet github de l'auteur [3]. Installez-la en la décompressant dans votre dossier *libraries*, comme expliqué plus haut pour la bibliothèque HT1632c, et jetez un coup d'œil aux exemples.

Il vous faudra exécuter l'exemple `set_clock.pde` au moins une fois pour mettre à l'heure votre RTC. Tant que cette dernière restera reliée à sa batterie, vous n'aurez pas besoin de le refaire. N'oubliez pas de modifier la ligne de la fonction `setup` qui contient l'heure à programmer. Le **listing 4** montre la partie de `set_clock.pde` qu'il vous faudra modifier.

Utilisation de l'EEPROM pour stocker les heures

L'ATMega 328 de l'*Arduino Uno* possède 512 octets d'EEPROM, une mémoire non volatile, pour conserver des données hors tension. Cette

Listing 4.

```
[...]

void setup() {
    Serial.begin(9600);

    /* Initialisation d'une nouvelle puce en désactivant la protection en écriture
     * et remettant à zéro le drapeau d'arrêt de l'horloge. Ces méthodes n'ont pas
     * tout le temps besoin d'être appelées. Référez-vous à la datasheet du DS1302
     * pour plus de détails */
    rtc.write_protect(false);
    rtc.halt(false);

    /* Création d'un objet Time pour régler la date et l'heure */
    /*   Tuesday, May 19, 2009 at 21:16:37.           */
    Time t(2009, 5, 19, 21, 16, 37, 3);

    /* Set the time and date on the chip */
    rtc.time(t);
}

[...]
```

Un projet fait uniquement avec de la récup'

partie de la mémoire est couramment utilisée par les concepteurs pour stocker la configuration de base d'un appareil.

Dans notre cas, nous utiliserons l'EEPROM pour stocker la dernière date où quelqu'un a arrosé les plantes (et appuyé sur le bouton). Il nous faudra au total trois octets : l'année, le mois et le jour. Comme nous souhaitons afficher le nombre de jours que les plantes ont passé sans eau, il nous faudra aussi donner à notre programme la capacité de calculer la différence de temps entre la date actuelle et celle stockée en mémoire. Le **listing 5** vous montrera comment stocker quelque chose en EEPROM.

Un bouton pour stocker l'heure...

Nous utiliserons un bouton pour signaler au micro que la date doit être enregistrée. Après une action sur ce bouton, la date est lue depuis la RTC et stockée dans l'EEPROM. J'ai utilisé l'E/S 12 pour ce bouton et l'ai configurée en INPUT_PULLUP (**listing 6**) pour ne pas avoir à ajouter une résistance externe.

Après avoir stocké la date, l'afficheur vous remer-

ciera d'avoir pensé aux plantes avec le message « *nice!* », et le programme s'arrêtera. Pour faire quoi que ce soit d'autre, il faudra redémarrer le programme. Je pense que c'est un bon moyen d'éviter que l'EEPROM ne soit lue ou écrite trop souvent.

En lisant le code complet disponible sur le site d'Elektor, vous remarquerez que j'ai ajouté une

Listing 5.

```
#include <EEPROM.h>

int val;

void setup() {
}

void loop() {
    val = analogRead(A0);
    EEPROM.write(0, val);
}
```




Figure 7.
« Hé, ça fait trois jours que
j'ai pas eu d'eau »

petite animation pour remercier mes voisins
d'avoir arrosé les plantes.

... et un autre pour l'alim

Le plus important dans ce projet est sans doute la
manière dont est gérée la batterie. La meilleure

façon d'épargner la batterie est de laisser le sys-
tème éteint. Avez-vous remarqué dans la liste du
matériel la présence de deux boutons alors que
jusqu'ici le code ne fait usage que d'un seul ?
Le deuxième bouton, il faudra le maintenir appuyé
pour que le système soit alimenté. Les microcon-
trôleurs démarrent rapidement ; ici la première
chose que vous verrez à l'écran est le logo *Arduino*
et, après quelques secondes, le nombre de jours
écoulés depuis le dernier arrosage des plantes.
La présence de ce bouton implique que pour stoc-
ker la date, vous appuyiez sur les deux boutons
en même temps. Tandis que le premier mettra le
système sous tension, le deuxième déclenchera
la sauvegarde de l'heure de la RTC en EEPROM
(fig. 7).

Le mot de la fin

Sous couvert d'une improvisation à partir de
composants qui tombaient sous la main, nous
avons expérimenté avec des circuits de com-
mande capable de piloter des centaines de LED
ainsi qu'avec les horloges temps réel. Prochaine
étape : mon prototype remplit-il sa fonction ?
Je vais le mettre dans l'escalier et on verra bien
si « nos » plantes se portent mieux.
C'est à vous, lecteurs, de vous approprier cette
proposition pour en faire autre chose à votre guise.
Une boîte dotée d'un afficheur et de deux boutons,
ça donne des idées. Horloge ou jeu ? C'est vous qui
voyez. Pourquoi pas ajouter un buzzer et quelques
boutons et en faire une petite console de jeux ?
N'hésitez pas à nous présenter votre chef-d'œuvre
sur www.elektor-projects.com.

(120714 - version française : Kévin PETIT)

Listing 6.

```
[...]

int saveTimePin = 12;
int saveTimeButton = LOW, saveTimeButtonOld = LOW;

void setup() {
    [...]
    pinMode(saveTimePin, INPUT_PULLUP);
}

void loop() {
    [...]
    saveTimeButton = digitalRead(saveTimePin);
    if(saveTimeButton == LOW && saveTimeButtonOld == HIGH) {
        // la date présente dans t sera utilisée pour la sauvegarde
        EEPROM.write(0, t.yr); // position 0 - year
        EEPROM.write(1, t.mon); // position 1 - month
        EEPROM.write(2, t.date); // position 2 - day
        // on efface l'afficheur
        display.cls();
        // et on affiche un message de remerciement
        display.text("nice!", 2, 5);
        // on arrête l'exécution ici
        while(true) {};
    }
    saveTimeButtonOld = saveTimeButton;
}
```

Références

- [1] Le projet Botanicalls : <http://botanicalls.com>
- [2] Explications de Krodal's sur la RTC DS1302 :
<http://arduino.cc/playground/Main/DS1302>
- [3] La bibliothèque RTC de Mark Sparks :
<https://github.com/msparks/arduino-ds1302>
- [4] Téléchargement de la bibliothèque et du
programme : www.elektor.fr/120174

Remerciements

Merci aux gars de l'Arduino Store qui m'avaient fait cadeau
de l'afficheur.

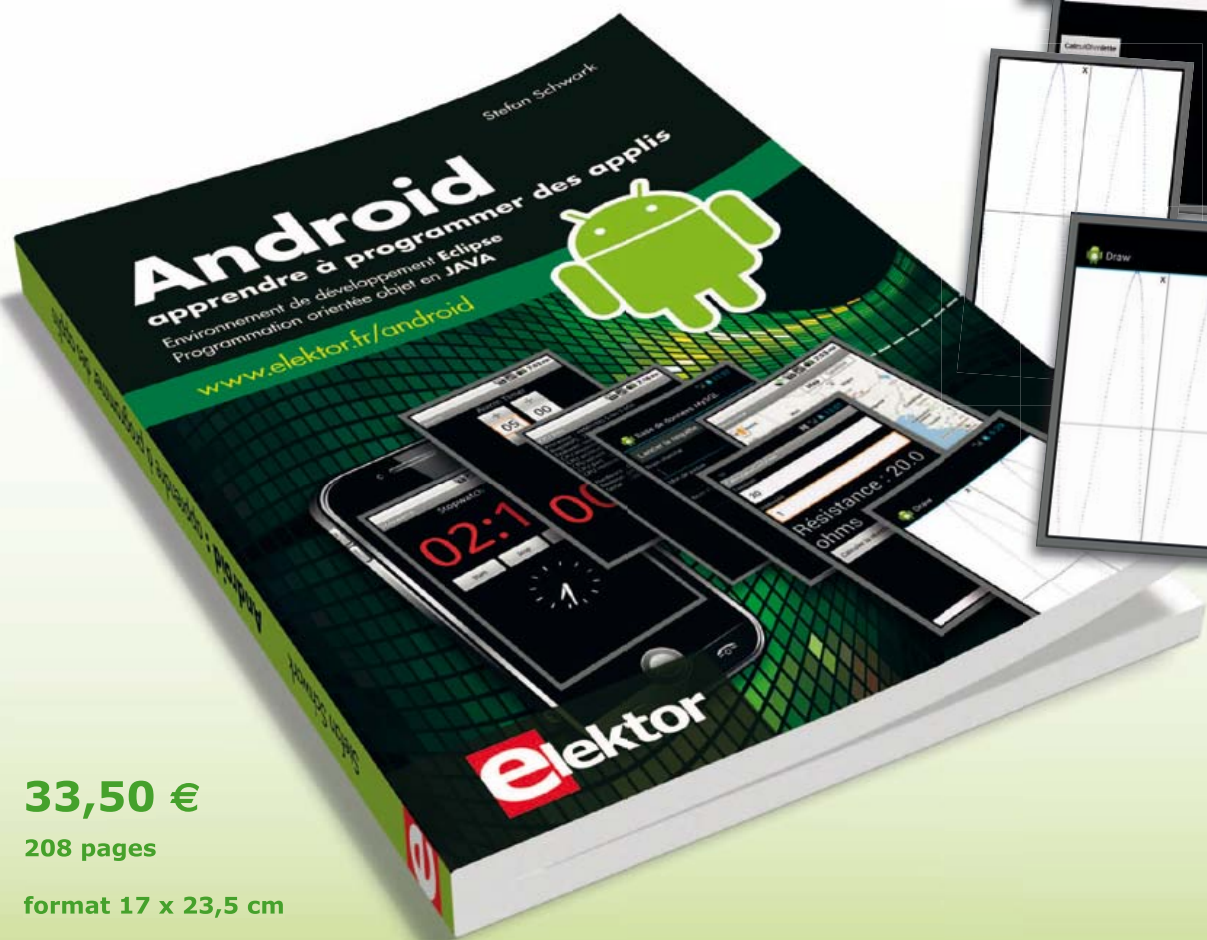


Télécharger des applications existantes n'est pas le seul moyen de **rendre votre téléphone tactile encore plus intelligent !**

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

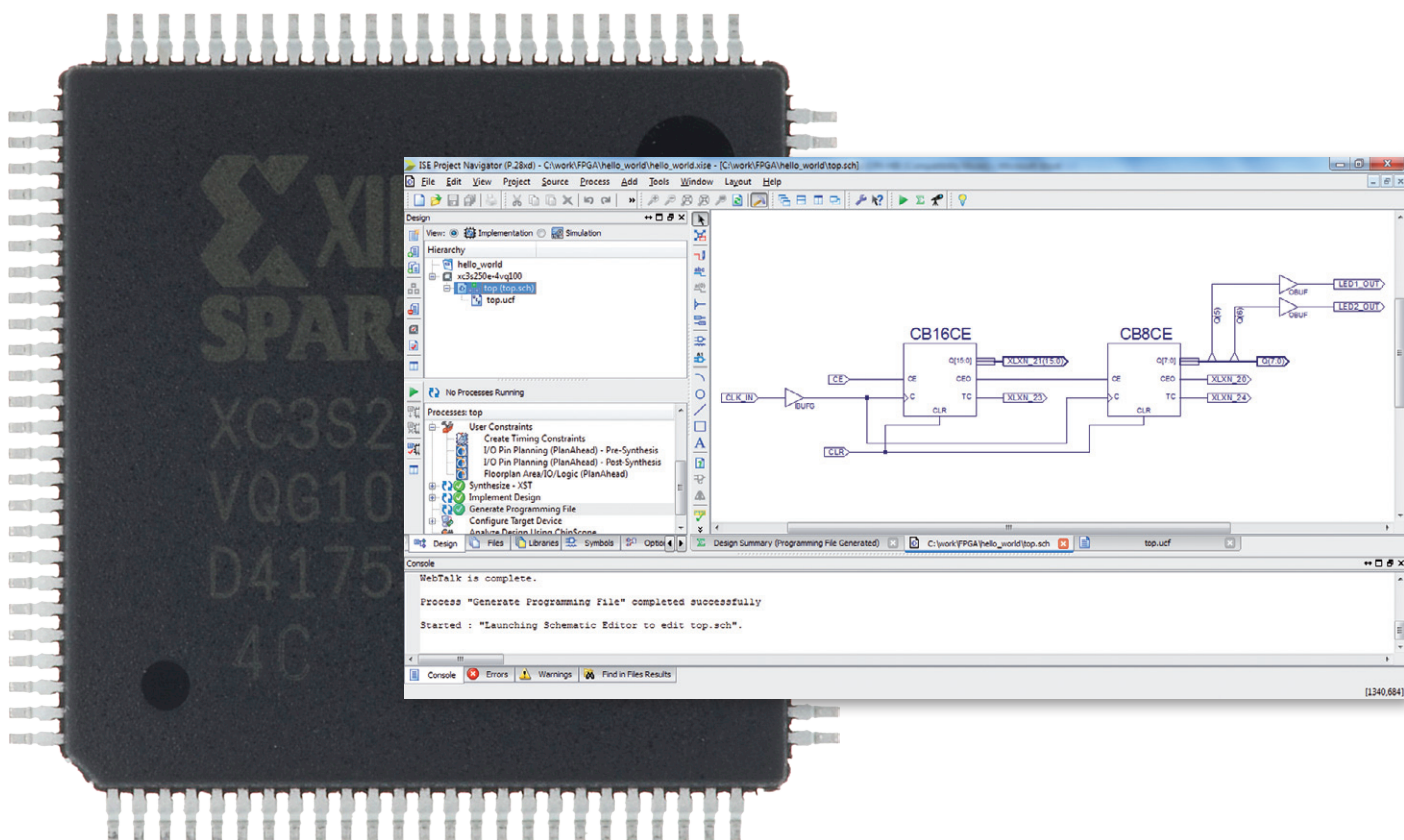
ISBN 978-2-86661-187-3

www.elektor.fr/android

elektor

et l'homme créa sa puce (2)

250 000 portes pour cligner d'une LED



Clemens Valens & Raymond Vermeulen
(Elektor)

C'est quoi un FPGA ? Et ça sert à quoi ? Nous avons répondu à ces questions dans l'article précédent et vous n'avez plus peur des FPGA. Peut-être même avez-vous déjà leur carte d'expérimentation d'Elektor. Comme il s'agit de composants complexes, les outils pour les programmer le sont aussi. Nous allons donc faire un bout de chemin avec vous à leur découverte. Les premiers pas sont aisés, mais quand on veut approfondir le sujet, il faut y investir pas mal d'énergie.

Et d'abord, installer le logiciel...

De quoi s'y mettre tout de suite, nous vous avons conseillé de demander à *Xilinx* le DVD gratuit pour l'avoir aujourd'hui sous la main. Avec lui, pas de querelle de systèmes d'exploitation,

il en existe un pour *Windows* et un pour *Linux*. Vous pouvez tout aussi bien le télécharger, mais autant les fabricants de FPGA aiment bourrer leurs composants de portes, autant ils raffolent de verser à torrents les octets dans leurs outils :

le téléchargement compte pas loin de 6 Go. Aussi, vérifiez si le *Download Manager* de *Xilinx* s'entend bien avec votre navigateur (**fig. 1**). J'ai eu du mal avec Firefox, probablement un paramètre quelque part que j'avais réglé de travers, mais ça a bien marché avec *Internet Explorer*. Ce qui est *cool* avec le *Download Manager*, c'est qu'il reprend le téléchargement là où il a été interrompu, on y gagne joliment du temps, parce qu'en revanche, la vitesse n'est pas sa qualité première.

Je n'ai pas de PC sous *Linux*, je continue dans la version *Windows*, mais je gage qu'il ne doit pas y avoir de grosse différence d'installation pour les utilisateurs de *Linux*. Glissez le DVD dans l'ordinateur ou prenez le fichier TAR chargé et lancez *xsetup.exe* au cas il n'aurait pas démarré tout seul. Cliquez sur *Next*, acceptez les deux *Licence Agreements* et sélectionnez *ISE WebPack* (**fig. 2**). *Next* de nouveau et par précaution, laissez toutes les coches en l'état. En principe, il suffirait de ne cocher que la case Acquérir ou Gérer une clé de licence, mais qui peut dire, quand vous en saurez davantage, ce que vous aurez encore envie d'approfondir. Le plus étonnant, c'est que cela n'a pas la moindre influence sur le volume de l'installation, un bon 12 Go. Un clic sur *Next*, choisissez une place pour tout installer, reclic sur *Next*, examinez le résumé et cliquez sur *Install*. Suivant la vitesse de votre PC, vous avez maintenant largement le temps de vaquer à une autre occupation, chez moi, ça a pris environ trois quarts d'heure.

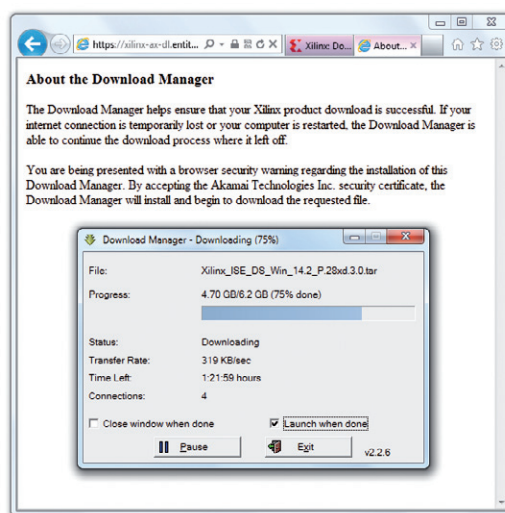


Figure 1.
Le *Download Manager* de *Xilinx*. Si vous ne le voyez pas, c'est qu'il ne fonctionne pas.

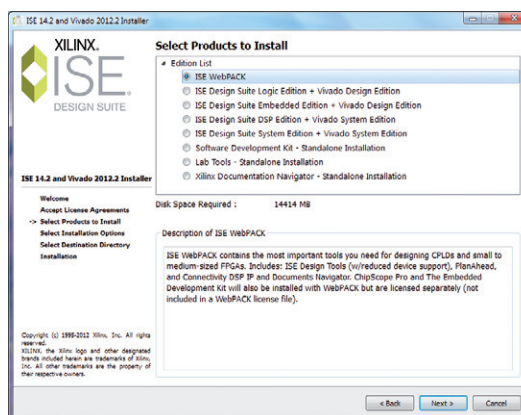


Figure 2.
L'installateur ISE qui va installer le *WebPACK ISE*.

Le moment est venu de demander une licence. Choisissez dans le *License Configuration Manager* (**fig. 3**) *Get Free ISE WebPack License* et cliquez sur *Next*. Voici une fenêtre de données à utiliser pour la licence. On clique sur *Connect Now* et on se trouve sur le site de *Xilinx* où il faut ouvrir un compte si on ne l'a pas encore fait, auquel cas on se retrouve éloigné de la page de la licence. Pour y revenir vite, le plus simple est de cliquer, dans le *Licence Configuration Manager*, sur l'onglet *Acquire a Licence*. On revient alors dans la figure 3 pour y faire ce qui vient d'être expliqué. Pour arriver à la page de demande de licence, il y a un tas de formulaires à remplir. Complétez-les comme vous le sentez, puis choisissez *ISE Design Suite : WebPACK License* et cliquez sur le bouton *Generate Node-Locked License*. Apparaît alors un récapitulatif avec un bouton *Next* et finalement un télégramme de félicitations accompagné

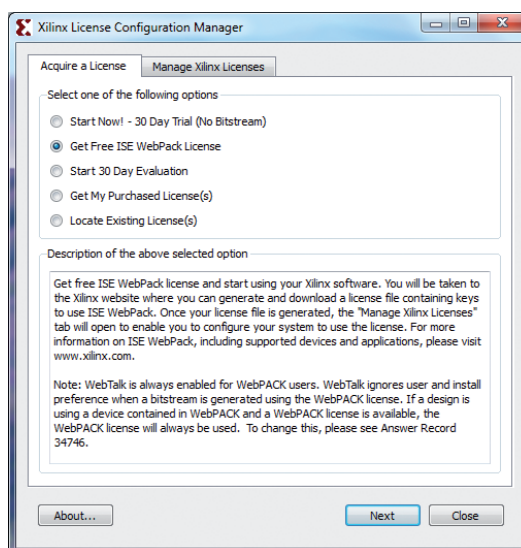


Figure 3.
Il faut obtenir une licence pour pouvoir utiliser le *WebPACK ISE*.

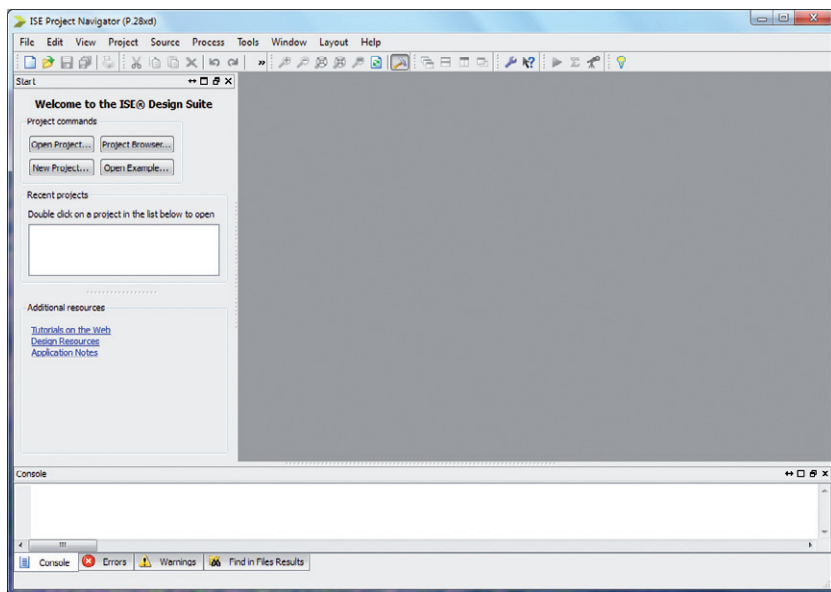


Figure 4.
Le *Project Manager ISE*
est encore totalement vide.

de l'avis que le fichier de licence a été envoyé à l'adresse de courriel que vous avez déclarée lors de l'ouverture de votre compte. Déposez ce fichier quelque part sur le PC de manière à le retrouver et retournez dans le *License Configuration Manager*. Cliquez ce coup-ci sur l'onglet *Manage Xilinx Licenses* et ensuite sur le bouton *Copy License...* Naviguez vers le fichier de licence et cliquez sur *Open*. Opération qui devrait se solder par : « *License installation was successful* ». Si c'est effectivement le cas, fermez le manager d'un clic sur *Close*. Vous conviendrez avec moi que c'est une installation qu'on ne réalise pas en vitesse entre deux portes.

Avant de commencer...

Si les outils ISE sont installés et que la licence est acceptée, nous pouvons enfin commencer. Si tout va bien, l'option *Xilinx Design Tools* est arrivée dans la liste du programme. Ouvrez-la, puis ouvrez *ISE Design Suite 14.2* (le numéro de version pourrait avoir changé entre-temps), ouvrez alors *ISE Design Tools* et lancez le *Project Navigator*. Dans mon menu, il y en a deux, une version pour 32 bits et une pour 64 bits, choisissez celle qui convient à votre PC. Vous verrez alors la fenêtre de la **fig. 4**.

Cliquez sur *New Project...* pour ouvrir le *New Project Wizard*. Complétez-le comme à la **fig. 5**, mais avec un nom compréhensible pour vous et mettez-le à un endroit pratique. Dans le champ *Description*, écrivez quelque chose, c'est une bonne habitude à prendre. Notre première appli-

cation FPGA, nous allons la réaliser sous forme de schéma, c'est plus familier à la plupart d'entre nous. La pure programmation de FPGA en langage VHDL ou *Verilog*, nous la gardons encore au chaud. Comme *Top-level source type* choisissez donc *Schematic*. Du fait que les applications de FPGA sont souvent assez vastes, on les organise de manière hiérarchisée. Cela veut dire qu'il y a plusieurs niveaux et, selon la profondeur à laquelle on a recours, il y a davantage de détails. Le niveau le plus élevé, dit *top-level*, n'est en réalité qu'un diagramme fonctionnel. Chaque bloc est alors réparti en morceaux maniables à des niveaux inférieurs, jusqu'à ce que la fonction soit définie en détail. Pour notre première application, nous allons faire clignoter une LED sur la carte d'expérimentation de FPGA, nul besoin de niveaux multiples, travaillons directement dans le *top-level*. Cliquez sur *Next* pour demander à l'Assistant de montrer la deuxième page (**fig. 6**). La plupart des champs peuvent garder les valeurs par défaut, sauf les champs *Family*, *Device*, *Package* et *Speed*. Ils doivent correspondre à la puce sur la carte et dans ce cas-ci, elle appartient à la famille *Spartan 3E*, type XC3S250E sous boîtier VQ100. Comme *Speed*, on choisira -4, c'est la version standard, -5 est la version rapide. C'est inscrit sur la puce, si vous regardez bien, vous verrez 4C en dessous. Cliquez alors sur *Next* pour consulter le résumé du projet. Il n'y a pas grand-chose à voir ; d'ailleurs, à ce moment, vous ne pouvez plus rien y changer. Aussi, cliquez sur *Finish*.

Prenez le temps de regarder ce qui se passe avant d'ouvrir, via *Project* → *New Source...* le *New Source Wizard* (**fig. 7**). Choisissez *Schematic* comme *Source Type*, comme nom de fichier, prenons *top* et contrôlons que *Add to project* est coché. Encore un clic sur *Next* pour obtenir un écran de résumé que nous quitterons par *Finish*. Après quelques hésitations et clignotements, la mise en forme de l'ISE change brusquement pour montrer une page blanche sur laquelle on peut dessiner le schéma. À gauche de la feuille, il y a aussi eu des changements et nous y voyons une liste d'options. Sous cette liste, une kyrielle d'onglets, plus qu'on ne pourrait en voir. Cliquez quelques fois sur la flèche à droite de l'onglet *Options* pour en afficher d'autres. L'onglet *Design*, nous l'avions à l'écran avant de lancer le *New Source Wizard*. Comme vous pouvez le voir, le schéma *top* a été ajouté au projet.

La Genèse est dans le cambouis !

... le dessin d'un schéma

Bon, le but est de faire visiblement clignoter une LED sur la carte à FPGA. Visiblement, ça veut dire à une fréquence que l'œil peut distinguer, tout au plus une dizaine de hertz. Le FPGA est rythmé à 8 MHz par le contrôleur. Il nous faut donc diviser le signal des tas de fois par deux, c'est ce que les bascules font le mieux ! 8 MHz divisé 22 fois par 2, ça doit donner un peu moins de 2 Hz. Vérification : $8\,000\,000 / 2^{22} = 1,9\text{ Hz}$. Juste ce dont nous avons besoin.

Nous devrions maintenant dessiner 22 bascules dans le schéma, mais nous pouvons aussi prendre un compteur. Un clic sur l'onglet *Symbols*, à côté de l'onglet *Options* dans le coin inférieur gauche de l'ISE. Choisir ensuite *Counter* dans la liste *Categories*. La liste *Symbols* donne à voir tous les compteurs que nous pouvons utiliser. Il n'y en a pas à 22 bits, nous devrons en réunir deux plus petits en série. Le premier compteur dans ma liste s'appelle *cb16ce*, ça ressemble à 16 bits, commençons avec lui.

Placer les symboles

Sélectionner le compteur *cb16ce* d'un clic et le traîner à la souris vers le schéma, tandis qu'un contour du symbole suit le curseur de la souris. Cliquer n'importe où dans le schéma pour y déposer le compteur. Et voilà qu'apparaît effectivement un compteur à 16 bits avec un tas d'entrées et sorties en plus. Pour savoir à quoi elles servent, un clic du bouton gauche de la souris sur le symbole. Dans le menu qui s'affiche, choisir *Symbol*, puis *Symbol Info*. L'aide d'ISE s'ouvre à la bonne page qui nous apprend tout sur son compte. Entre autres que la broche CLR doit être basse et la broche CE haute pour pouvoir compter. On y voit aussi que la broche CEO peut servir d'horloge à un autre compteur. Donc, nous n'avons pas besoin des sorties Q0 à Q15. Comme second compteur, nous prendrons un *cb8ce*.

Dans l'article précédent, nous avons clairement attiré l'attention sur le fait qu'entre les blocs logiques de FPGA et le monde extérieur, il faut nécessairement passer par des blocs d'E/S. C'est valable ici pour la LED et le signal d'horloge, mais aussi pour les signaux CE et CLR que nous

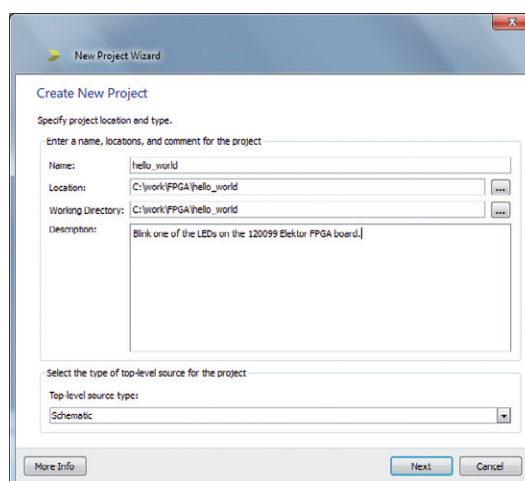


Figure 5.
Pour commencer un nouveau projet, on utilise l'Assistant *New Project*.

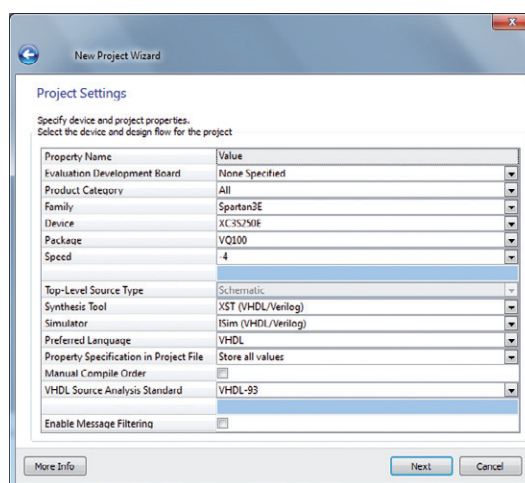


Figure 6.
Il est primordial de sélectionner le bon FPGA.

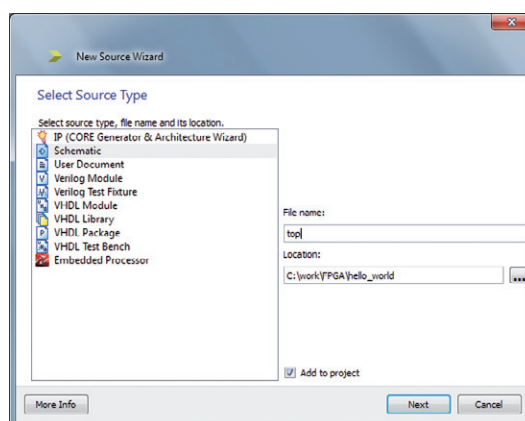
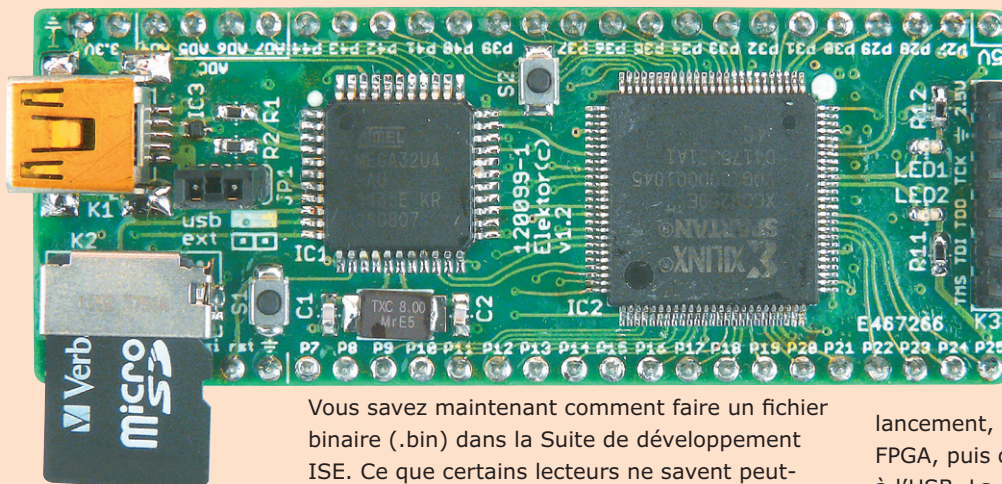


Figure 7.
Un assistant vous aide à ajouter des fichiers à votre projet.

voulons commander en passant par des broches. ISE dispose pour réaliser tout cela de symboles spéciaux, classés dans la catégorie E/S, mais il y a des subtilités auxquelles il faut être attentif. Les signaux d'horloge font partie d'une classe à

Le micrologiciel du microcontrôleur : la substantifique moelle

Raymond Vermeulen



Vous savez maintenant comment faire un fichier binaire (.bin) dans la Suite de développement ISE. Ce que certains lecteurs ne savent peut-être pas encore, c'est comment le faire atterrir sans encombre dans le cœur du FPGA. Là, on s'adresse au talent acquis par le microcontrôleur sur la carte. Rien moins que son micrologiciel, adroitement mis au point au Laboratoire Elektor. Pour ceux que la chose intéresse, le code source et le code hexadécimal sont disponibles sur notre site [1]. Le logiciel est d'ailleurs à source ouverte et nous invitons les lecteurs expérimentés à y ajouter les fonctions complémentaires qu'ils jugeront utiles. Ce micrologiciel a été développé en partant du principe qu'il n'est pas indispensable de sans cesse réinventer la roue. Pour certaines fonctions, il existe déjà du code source ouvert à tous, il y en a même des bibliothèques pleines. Nous en avons d'ailleurs fait nos choux gras. Raison pour laquelle je tiens à remercier entre autres **Dean Camera** de la librairie LUFA [3] et **ChaN** de (petit) FatFs [4]. Sans ces codes, la mise au point de notre projet aurait duré beaucoup plus longtemps et n'aurait sans doute pas comporté autant de fonctions. De la librairie LUFA, nous avons utilisé la version 120219 et, de celle de petit FatFs, la version R0.02a.

Pour mettre à jour le micrologiciel dans le microcontrôleur, il ne faut pas d'ISP. Il y a un chargeur d'amorce DFU intégré à la puce, c'est une partie du projet LUFA. Sous Windows, on peut facilement effectuer une mise à jour du

micrologiciel avec le programme « Atmel Flip » [5].

Normalement, le chargeur d'amorce est écrasé lors du démarrage de la carte. Mais si on pousse sur S1 puis sur S2 pour ensuite lâcher S1 puis S2, le chargeur se lance. Le pilote correspondant fait partie du fichier d'installation d'Atmel Flip.

Mais retournons à l'application proprement dite. Après le

lancement, on commence par programmer le FPGA, puis on exécute tout ce qui se rapporte à l'USB. La raison en est que les opérations en USB de LUFA sont gérées par interruptions. On a pensé qu'il serait bon de régler les affaires dont la chronologie est critique avant d'activer les interruptions. Cela m'a aussi permis de garder séparées les deux fonctions principales. Il faut bien comprendre que la carte micro SD ne peut pas être montée en même temps sur le PC et localement.

La première phase concerne l'initialisation du matériel. On dispose des E/S nécessaires, comme les {2 :0} broches pour configurer le FPGA en mode *SerialSlave*. On passe aussi Hswap au niveau haut pour débrancher les résistances de polarisation haute des E/S pendant la configuration. On s'adresse à la carte micro SD, elle est identifiée et apprêtée à l'usage (des versions différentes demandent des variantes dans la communication). On monte la partition localement sur la carte micro SD en utilisant des fonctions de Petit FatFs. Si le fichier config.bin est bien sur la carte, on l'ouvre. On passe la ligne Prog_b au niveau bas pendant 1 ms pour mettre à zéro le FPGA, puis c'est la ligne Init_B qui devient haute et le transfert de données commence. Les blocs de données de la carte mémoire sont lus et écrits dans le FPGA en passant par le bus USART commandé par des fonctions de la librairie LUFA. Pendant l'émission de ces blocs de données, on vérifie

après chaque octet si la ligne Init_B n'est pas devenue basse, ce qui signifierait qu'une erreur a été détectée. Quand l'ensemble du fichier a été transmis, on vérifie si le signal « Done » est haut pour signaler que la configuration s'est déroulée correctement. On démonte alors la carte mémoire, on éteint les périphériques SPI et USART et les broches utilisées sont remises à haute impédance (Hi Z).

Pour utiliser localement les FAT32 et 16 sans recourir à un système d'exploitation, on se sert de Petit FatFs. Il se trouve dans \FPGA_board_Config_v1_0\local_fat dans le fichier zip à télécharger via [1]. On trouve dans pff.h les directives « define » utilisées pour choisir la gestion de la FAT32. Seule la partie « read » est activée parce qu'il n'est pas nécessaire d'écrire sur la carte dans cette phase. MMC.c est une composante d'un exemple d'application que l'on trouve sur [3]. La fonction à bas niveau existante est en grande partie remplacée par des fonctions de la librairie LUFA.

Dans la deuxième phase, une boucle sans fin démarre pour la gestion de la communication USB, avec un port COM virtuel et une carte micro SD comme périphérique de stockage de masse. Il faut pour cela réinitialiser le matériel, donc adresser de nouveau la carte mémoire, identifier et préparer la communication SPI. Le résultat de la gestion d'erreurs de la première phase est utilisé pour choisir une chaîne de caractères qui commandera le port virtuel vers le PC. La carte micro SD est alors traitée par le système d'exploitation comme une clé USB, ce qui signifie que les commandes SCSI sont envoyées sur le bus USB et que le système de fichiers ne fonctionne plus localement, mais

à distance sur le PC. Le code pour utiliser à cette fin une carte micro SD ne se trouvait pas normalement en LUFA, mais je l'ai trouvé sur internet. J'ai dû l'adapter pour qu'il s'accorde à la version LUFA 120219. Il y a encore du code devenu superflu dans le fichier FPGA_board_Config.h qui sera sans doute éliminé prochainement.

La librairie LUFA sert surtout à traiter tous les événements USB. Mais elle contient tout un tas de fonctions à bas niveau pour la commande de divers périphériques. On trouve aussi dans le paquet LUFA des profils matériels pour communiquer avec des cartes spécifiques. J'aurais aussi pu en ajouter un qui convenait pour cette carte-ci, mais dans le feu de l'action, j'ai abandonné la compatibilité au profit d'une programmation plus rapide et plus facile. Il y a aussi des projets de démonstration LUFA. Ce micrologiciel se base sur une démonstration de stockage de masse virtuel. Pour d'autres informations sur le fonctionnement détaillé de LUFA, la documentation est disponible sur [3].

Pour le débogage, on utilise un port COM virtuel qui envoie au PC les messages d'erreur. Il nous a bien servi au cours de la mise au point. Celui que ça intéresse peut l'utiliser dans le même but, mais moyennant quelques aménagements, il peut aussi communiquer dans l'autre sens.

Pour une notation simple et fonctionnelle, on peut utiliser des directives « define ». Elles marchent bien, sans surcharger la fonction. Je les ai utilisées pour configurer des broches et les lire, la fonction reste bien lisible.

Il y en a par exemple dans FPGA_board_Config.h :

```
#define Release_FPGA_reset()      DDRB&=~(1<<DDB4)
#define Init_B_status()          ((PINC >> PC6) & 0x01)
```

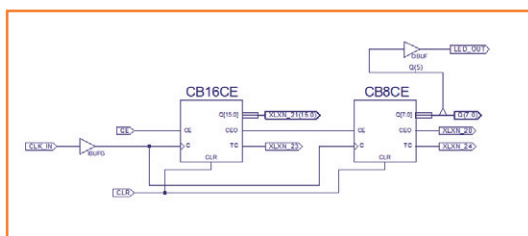
Et dans une fonction de FPGA_board_Config.c on l'utilise aussi comme ceci :

```
while (!Init_B_status()) {;;}
```

ou :

```
Release_FPGA_reset();           // release the Prog_b line
[/code]
```


Figure 8.
Le schéma de notre
clignotant.



Avant de planer, il faut aller au charbon

part, ils ont leur propre symbole, une telle entrée s'appelle *ibufg*. C'est un tampon qui veille à ce que le signal d'horloge arrive directement sur une seule des lignes spéciales du FPGA. Les autres entrées peuvent aussi se raccorder par des tampons, on les nomme *ibuf*. En fait, si vous lisez convenablement l'aide sur ce symbole, vous comprendrez qu'il vaut mieux ne pas le positionner vous-même, mais laisser faire ISE. Ce que vous pouvez entreprendre, c'est de placer un marqueur d'E/S, une sorte d'étiquette. Dans notre petit projet, nous n'allons pas placer de symbole *ibuf*, mais seulement des étiquettes d'E/S. Les sorties, en revanche, nous pouvons les envoyer à l'extérieur explicitement par l'intermédiaire d'un tampon. Il en existe de différentes sortes, qui permettent de spécifier le type de sortie. Pour la LED, nous pouvons utiliser un simple *obuf*. Après cette explication, plaçons dans le schéma un *obuf* à proximité de la sortie Q du compteur *cb8ce* et un *ibufg* près de l'entrée d'horloge du compteur *cb16ce*.

Le tire-ligne

Passons maintenant aux liaisons. C'est ce qu'on fait en cliquant sur le bouton *Add Wire* qui se trouve dans la partie supérieure de la boîte à outils, entre le schéma et l'onglet *Options*. Survolez du curseur le bouton, vous verrez sa fonction. Après un clic, le curseur se transforme en croix pour vous aider à viser juste en cliquant sur les broches dans le schéma. Le mode par défaut est *Autorouter* (voyez l'onglet *Options*), il suffit alors de cliquer sur les extrémités à joindre. ISE se charge de tracer les fils de manière (relativement) propre. Vous pouvez en déplacer des segments à la souris pour embellir le schéma. Reliez de la même façon C et C ainsi que CLR et CLR sur les deux compteurs. Reliez aussi CEO de

cb16ce avec CE de *cb8ce*. Connectez la sortie de *ibufg* à l'entrée C de *cb16ce*. Laissez libre l'entrée de *obuf*, on s'en occupera plus tard.

Marqueurs et étiquettes

L'étape suivante consiste à ajouter les étiquettes d'E/S à l'aide du bouton *Add I/O Marker*, par exemple. Si l'option *Add an automatic marker* est activée, vous pourrez dans certains cas cliquer directement sur une broche pour y placer une étiquette. Si ça ne marche pas, dessinez d'abord un fil ou un segment de bus avec *Add Wire*. D'un double clic sur un segment, vous le terminez à la place voulue. En tout cas, dessinez un segment de bus à la sortie Q[7:0] de *cb8ce*, parce que nous en avons besoin tout de suite.

Les étiquettes d'E/S obtiennent automatiquement un nom. Vous pouvez le modifier avec le bouton *Add Net Name*. Inscrivez d'abord le nom du nœud (onglet *Options*) avant de le baptiser d'un clic. Renommez de cette manière tous les nœuds qui doivent être liés à une broche : CLR, CE, CLK_IN et LED_OUT. Renommez le bus à la sortie Q[7:0] de *cb8ce* jusqu'à Q(7:0) (quand vous utilisez des crochets, ISE en fait des ronds). Q(7:0) signifie que sur le bus circulent les signaux Q0 à Q7.

Le branchement du bus

Reste à connecter l'entrée de l'*obuf* à la sortie Q5 (nous divisons par 2^{22} , souvenez-vous) du compteur *cb8ce*. ISE accepte la méthode suivante. Un clic sur le nœud *Add Bus Tap* et dans *Options* choisir *Bottom* (ou quelque chose d'autre) pour en obtenir la bonne orientation. Vous pouvez maintenant installer dans le schéma une sorte d'entonnoir dont le côté large fait face au segment de bus Q(7:0). Reliez ensuite par un bout de fil la sortie de l'entonnoir à l'entrée de *obuf*. Appelez-la Q(5). À présent, votre schéma ressemble sûrement à la **fig. 8**.

Avant de poursuivre par la labellisation des broches du FPGA, il serait sage de voir s'il n'y a pas de défaut à notre projet. Normalement, ce serait le moment de soumettre le projet à une simulation, mais il est tellement simple et cet article déjà si long que nous allons provisoirement faire l'impasse sur cette étape. Voyons si nous pouvons *synthétiser le projet*, c'est-à-dire demander à ISE de le traduire en portes standard et bascules disponibles dans sa bibliothèque. Les programmeurs auront compris que c'est l'équivalent de la compilation d'un programme.

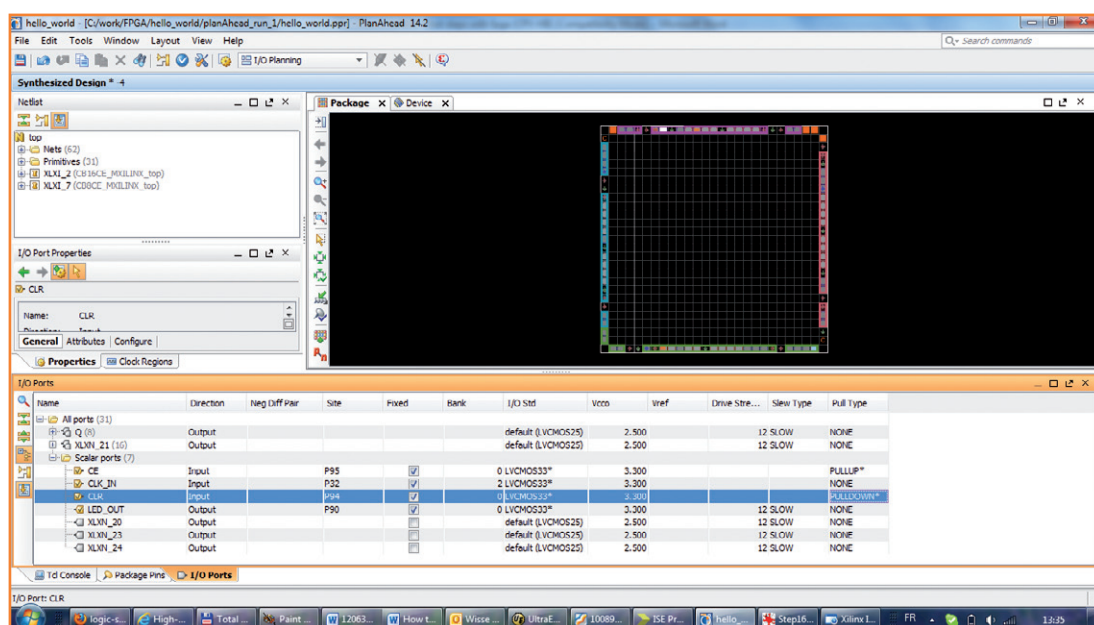


Figure 9.

Ce bel outil graphique permet l'attribution de signaux aux broches du FPGA. Il est aussi possible de travailler sur un fichier UCF dans un éditeur de texte.

La synthèse

Pour synthétiser, on passe par l'onglet *Design* et on double-clique sur *Synthesize – XST*. Plein de choses s'affichent dans la fenêtre *Console* et un icône pivotant indique qu'il y a du travail en cours. Si tout se passe bien, une coche verte se manifestera à gauche de *Synthesize – XST*. La console vous apprendra aussi que le projet consomme 24 bascules.

L'attribution des broches

Attribuons maintenant des broches aux étiquettes. Nous disposons pour cela de l'outil *PlanAhead* à lancer depuis l'onglet *Design*. Pour cela, il faut ouvrir *User Constraints*, après quoi nous choisirons, d'un double clic, *I/O Pin Planning (PlanAhead) Post-Synthesis* parce que nous venons à peine de synthétiser. ISE demande si vous voulez ajouter un *User Constraint File (UCF)* au projet et puisque c'est le cas, cliquez sur OK. Selon la vélocité de votre PC, cela peut durer un moment avant que *PlanAhead* ne soit prêt à l'emploi, c'est un outil assez costaud.

Dans l'onglet *I/O Ports* de *PlanAhead* (en bas) nous retrouvons les étiquettes de notre schéma. Ouvrez *Scalar ports*, vous y verrez les étiquettes auxquelles nous voulons lier les broches. Vous pourriez utilement vous reporter aux figures 7 et 8 de l'article précédent pour savoir quelles broches utiliser. Les LED sont sur les broches 90 et 91. Cliquez dans la rangée *LED_OUT* et dans la colonne

Site. Une liste s'ouvre dans laquelle vous choisissez P90 (ou P91 si vous préférez). Cliquez ensuite dans la colonne *I/O Std* et choisissez LVCMOS33 (le seul standard utilisable sur notre carte). Parfait, le reste, l'outil le remplira. Au tour de la broche *CLK_IN*. En principe, l'horloge entre toujours sur notre carte FPGA par la broche 32, choisissez donc P32 dans la colonne *Site*. Mettez le standard d'E/S sur LVCMOS33. Il reste les signaux CE en CLR. Vous pouvez leur choisir vous-même des broches, à condition qu'ils arrivent sur l'un des connecteurs K4 ou K5. J'ai choisi P94 (CLR) et P95 (CE). Le standard d'E/S est de nouveau LVCMOS33. Pour nous assurer que ces entrées ne vont pas bloquer le circuit, nous activerons aussi les résistances de polarisation haute et basse sur ces broches. Cliquez dans la colonne *Pull Type* pour donner à CE une résistance PULLUP et à CLR une PULLDOWN (fig. 9). Fermons cette configuration en cliquant sur le bouton *Save Constraints* et retournons à l'onglet *Design* d'ISE.

L'implémentation

Nous allons implémenter le projet : ça veut dire quoi ? Traduire en blocs logiques et trucs du genre. En avant ! Un double clic sur *Implement Design* et on attend qu'une coche verte apparaisse. Si jamais il y a des erreurs ou des avertissements, faudra aller voir sous l'onglet *Errors* ou bien *Warnings*. Mais si vous avez tout fait aussi bien que moi, il n'y aura ni erreur ni avertissement.

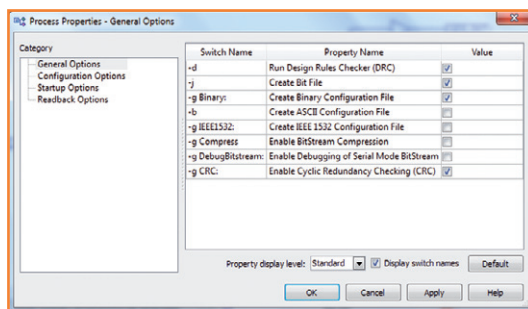


Figure 10.
Les options du *bitstream*, première page.

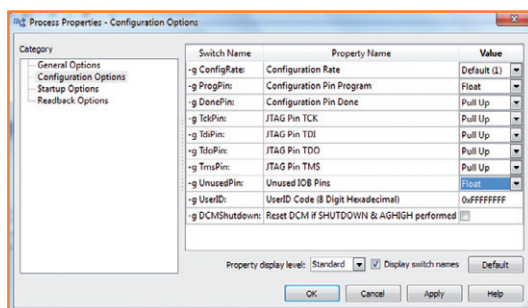


Figure 11.
Les options du *bitstream*, deuxième page.

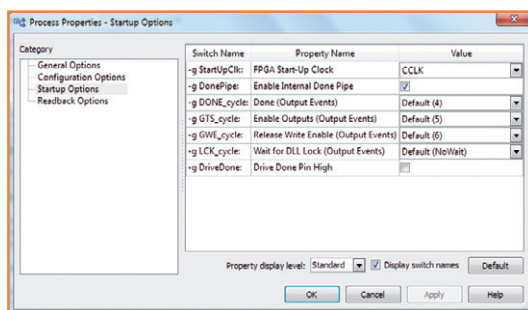


Figure 12.
Les options du *bitstream*, troisième page. Il n'est pas nécessaire de modifier la page quatre.

pour produire le flux binaire. Si tout va bien, après un certain temps, une coche verte apparaît. Dans le dossier du projet (regardez dans la barre de titre d'ISE), vous trouverez le fichier *top.bin*.

... pour programmer le FPGA

Introduisez la carte micro SD dans son support sur la carte que vous branchez sur un port USB libre du PC pour que son système d'exploitation détecte un lecteur externe. Copiez-y le fichier *.bin* qui vient d'être produit. Renommez-le en *config.bin*. Débranchez comme il convient ce lecteur externe et appuyez sur S1, juste à côté du support de carte micro SD sur la carte d'expérimentation. Si tout s'est bien déroulé, une des LED se met à clignoter.

Devoir à la maison

Branchez aussi l'autre LED de la carte FPGA pour qu'on la voie clignoter, mais d'une façon différente.

(120630 – version française : Robert Grignard)

Liens

- [1] www.elektor.fr/120630
- [2] www.xilinx.com/support/download/index.htm
- [3] www.fourwalledcubicle.com/LUFA.php
- [4] http://elm-chan.org/fsw/ff/00index_p.html
- [5] www.atmel.com/tools/FLIP.aspx

1 Générer le flux binaire...

On y est presque ! Mais il nous faut encore produire un fichier pour programmer le FPGA. Il s'appelle le *bitstream* et il y en a de toutes sortes et pour tous les goûts. Assurons-nous de produire le bon, cliquons du bouton droit de la souris sur *Generate Programming File* sous l'onglet *Design*, après quoi nous choisirons les *Process Properties* dans le menu qui surgit.

Choisir dans la liste *Category*, *General Options* et cocher *Create Binary Configuration File* (fig. 10). Choisir la catégorie suivante, *Configuration Options*, et mettre les valeurs pour *Configuration Pin Program* et *Unused IOB Pins* sur *float* (fig. 11). Cocher l'option *Drive Done Pin High* dans la catégorie *Startup Options* (fig. 12). Les *Readback Options* peuvent rester telles quelles. Cliquer OK pour fermer les *Process Properties* et double-cliquer sur *Generate Programming File*

La carte d'expérimentation de FPGA montée et testée, prête à l'emploi, est disponible chez Elektor au prix de 59,95 euros seulement, frais de port en sus.
www.elektor.fr/120099

tronçonner le tore

Dans ce numéro vous avez peut-être déjà remarqué un article intitulé *limiteur d'appel de courant* pour lequel il faut confectionner une bobine sur un tore avec entrefer. Voici la préparation de ce spécimen :

Thijs Beckers
(Elektor)

(120717)



Commencez par trouver le bon tore de ferrite, bobinez le fil et immobilisez le tout dans un étau en serrant modérément. Intercalez des chutes de tôle d'aluminium pour éviter la morsure des mâchoires de l'étau sur le cuivre ou la ferrite.



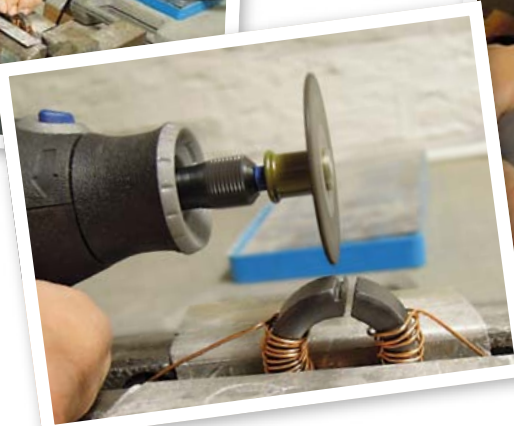
Sortez votre mini-disqueuse avec un disque à tronçonner le métal, et commencez à tronçonner le tore en surveillant l'angle du disque. Vous portez bien sûr des lunettes de sécurité. Gare aux escarbilles !



Ayez la main légère ! Il nous a fallu bien 3 ou 4 mn pour la 1^{re} coupe. La vitesse de rotation doit être élevée, sans toutefois dépasser la limite tolérée par le disque.



Repositionnez le tore dans l'étau pour être à main pour le second trait de coupe. Attention, le tore est peut-être chaud. Comme il a perdu une partie de sa rigidité, il est plus cassant.



Ne forcez rien, prenez votre temps. Un disque à tronçonner supporte mal les torsions et ne doit pas être utilisé pour abraser. Une pression malencontreuse risque de le briser.



Ça y est enfin ! Ne touchez pas le toron tout de suite, il est peut-être brûlant. Et manipulez-le prudemment, il est maintenant nettement plus cassant que lorsqu'il était entier.

matériel libre



Sébastien Bourdeauducq et Milkymist.

Tessel Renzenbrink
(TTF)

Les logiciels libres ont fait leurs preuves. Ancrés dans deux des plus grandes qualités humaines, la curiosité et la collaboration, ils répondent aux besoins d'accès, de partage, de modification et d'amélioration du code source. Ces logiciels permettent à tout individu de développer son propre code à partir de solutions déjà écrites. Cette méthode atténue la redondance, réduit les coûts, abaisse les barrières et rend la programmation plus agréable. Avec un système basé sur Linux sur plus de la moitié des téléphones mobiles, on peut sans conteste affirmer que les logiciels libres se sont fait leur place.

La mentalité « libre » s'étend aujourd'hui à la communauté *hardware* et l'on assiste à l'émergence de matériel volontairement libre de droits, dit *Open Source Hardware* ou OSHW.

On désigne ainsi un matériel informatique dont la conception est mise en accès libre pour tout un chacun. Elle peut être partagée, modifiée et,

évidemment, utilisée pour construire un matériel informatique.

Pour optimiser le partage et la production, la communauté OSHW travaille actuellement à la standardisation. Elle développe des outils de collaboration et des logiciels de conception. Et, inspirée par les logiciels libres, elle publie des licences qui

régissent les droits et restrictions de l'utilisateur.

Opérant aux confins de l'OSHW, Sébastien Bourdeauducq est fondateur du projet libre Milkymist [1]. Il est également l'organisateur de plusieurs conférences sur le thème de l'OSHW, tel le rassemblement *Exceptionally Hard & Soft Meeting* qui a eu lieu à Berlin du 28 au 30 décembre 2012. Lors d'un entretien, Sébastien m'a parlé de la conception d'une puce libre de droits et des applications spatiales du code *Milkymist* qui se retrouve en orbite terrestre basse.

Libres de droits

Tessel : Comment en êtes-vous arrivé à vous impliquer dans le matériel libre ?

Sébastien : J'ai toujours adoré comprendre comment tout fonctionne. J'ai débuté l'électronique et la programmation alors que je n'étais qu'un enfant. Mon premier contact avec le libre date de 1999, lorsque je suis passé à Linux, mais j'avais déjà modifié du code publié dans des magazines, des livres et dans BBS [*Bulletin Board Systems*], alors qu'il n'était pas explicitement déclaré *open source*. Frustré des restrictions de *Windows* qui empêche d'accéder au cœur du système d'exploitation, j'ai vraiment apprécié Linux.

Le premier de mes projets de « matériel libre » date de 2002 : c'était un système de communication infrarouge pour la calculatrice TI89, avec une licence GFDL, la licence de documentation libre GNU. Quelque chose de très simple. Ensuite, je me suis attaqué au développement d'un pilote sans fil pour *FreeBSD* (un système d'exploitation libre de type UNIX). C'était un mélange entre matériel et logiciel car je devais comprendre comment fonctionnait le matériel sans fil fourni sans documentation. Aujourd'hui, mon plus grand projet OSHW est *Milkymist*, lancé en 2007 (**encart**) pour ouvrir la conception des puces, quelque chose que peu de projets libres abordent, même aujourd'hui.

Innovation nouvelle manière

Tessel : Pourquoi des puces libres de droits ?

Sébastien : La première raison est éthico-philosophique : presque toute l'électronique qualifiée aujourd'hui de libre est basée sur des puces qui sont des boîtes noires que les gens non seulement ne comprennent pas, mais qu'ils n'essayent pas de comprendre. Débridons le savoir !

La seconde raison relève de la sécurité : ce

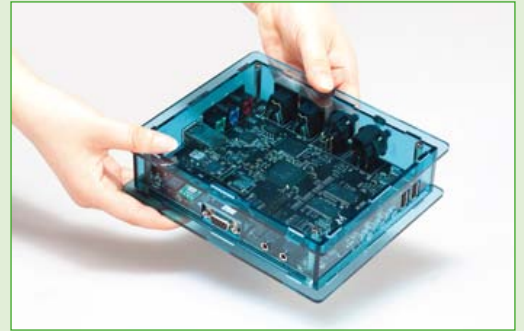
Milkymist

Milkymist One est un synthétiseur vidéo autonome d'effets psychédéliques et interactifs pour la scène. Il est en source ouverte autant que possible, tant pour le logiciel que pour le matériel.

La plate-forme

Milkymist dispose en particulier d'un système sur puce (*System-on-Chip* ou SoC) basé sur un FPGA développé sur mesure et elle est complétée par un code source en HDL libre,

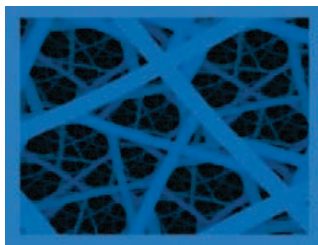
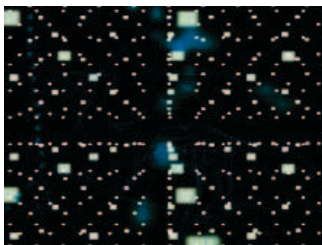
langage de description de matériel. À l'exception du processeur Lattice Mico32, la plupart des composants du SoC dépendent de la licence publique générale GNU. La programmation et les fonctions du SoC sont décrites en détail dans l'édition de juillet-août 2012 d'Elektor [6]. Le logiciel de synthèse vidéo *Flickernoise* est fortement inspiré par *Milkdrop*, un *plugin* libre de droits pour le lecteur audio *Winamp* qu'il agrmente d'effets visuels. Comme la plupart des projets libres de droits, *Milkymist* a bénéficié de contributions nombreuses [1].



Milkymist One

matériel peut comporter des portes dérobées. Une telle backdoor est une manipulation du matériel ou du microprogramme de la puce qui, à l'insu de ceux qui l'utilisent, donne à un tiers non autorisé un accès à la puce, pour la désactiver ou la reprogrammer. Les portes dérobées peuvent être aménagées dès l'étape de production. Elles sont considérées comme une menace réelle, en particulier dans le matériel informatique militaire. Enfin, la troisième raison était le souhait d'intensifier l'innovation. Et ça marche bien, mais hors de la communauté OSHW. Le code *Milkymist* a fait son chemin dans des projets scientifiques majeurs comme la station spatiale internationale (ISS) et les accélérateurs de particules.

En 2009, le *Jet Propulsion Laboratory* de la Nasa (JPL) a utilisé le code du contrôleur de mémoire SoC de *Milkymist* pour son programme de communication et de navigation spatiales (*Space Communications and Navigation* - SCA-N) [2]. Le 15 octobre 2012, Sébastien recevait la lettre suivante :



Cher Sébastien,
c'est avec plaisir que je vous remercie pour votre travail libre de droits dont nous sommes servis pour développer la radio logicielle du *Jet Propulsion Laboratory*, embarquée dans le *SCaN Testbed*. Celui-ci est à présent installé sur la station spatiale internationale. Au cours des derniers mois, il a passé avec succès les tests et affectations initiales, y compris l'utilisation du contrôleur dynamique de ram conçu par Greg Taylor à partir de vos travaux. On peut donc affirmer que dans l'espace, désormais, plusieurs millions de bits passent dans le code que vous avez écrit.

Encore merci,
James P. Lux
Co-Principal Investigator, SCaN Testbed

Tessel : vous avez également participé aux projets de matériel libre du CERN, le Centre Européen de Recherche Nucléaire. De nombreux instituts de recherche gardent leurs travaux secrets. Pourquoi le CERN a-t-il opté pour l'OSHW ?

Sébastien : Eh bien, les instituts de recherche, comme les chercheurs, ne sont pas tous semblables. La plupart d'entre eux ont certes un réflexe de secret – parfois légitime, parfois par désintérêt, et les éditeurs scientifiques privilégient le cloisonnement par intérêt financier. La communauté scientifique de la physique des particules me semble un peu plus ouverte que les autres, peut-être parce qu'ils font de la recherche fondamentale sans retombées commerciales ou militaires immédiates.

L'initiative du CERN au niveau du matériel libre est également portée par quelques employés acquis à la cause de l'OSHW. Ils ont créé un répertoire en ligne de matériel en source libre (*Open Hardware Repository*) où les ingénieurs en électronique peuvent collaborer autour de matériel libre utilisé en physique expérimentale [3]. Et en 2011, ils ont publié la licence *CERN Open Hardware*. [4].

Exceptionally Hard & Soft Meeting

Tessel : Pourquoi organisez-vous l'Exceptionally Hard & Soft Meeting (EHSM) ?

Sébastien : Certains des thèmes de la plupart des conférences d'OSHW et de *hackers*, y compris les trois co-organisées avant l'EHSM [5], manquaient de consistance. L'essentiel de la technologie dans le domaine matériel est simple et évite soigneusement ce qui est cher et complexe. Ça aussi c'est frustrant. Ma motivation initiale est donc la même que pour *Milkymist*. Avec l'EHSM, nous visons plus loin : notre intervenant principal est un adolescent de 17 ans qui a construit un réacteur de fusion nucléaire.

EHSM cherche également à impliquer les institutions scientifiques et la recherche. Le CERN y participe d'ores et déjà. Devraient s'y joindre aussi des chercheurs en physique quantique, avec peut-être un atelier de pratique expérimentale en physique quantique. Et nous avons déjà pris contact avec un autre institut berlinois très avancé en recherche sur les matériaux.

Nous voudrions aussi que ces conférences ne soient plus quasi-exclusivement masculines.

Tessel : OSHW diffère du mouvement des logiciels libres dans la mesure où la question de brevets s'ajoute à celle des droits d'auteur. Que pensez-vous des brevets, nuisent-ils à l'OSHW ?

Sébastien : les brevets portent sur des applications tellement ciblées et étroites qu'ils nuisent à simplement tout, y compris à l'OSHW. À l'origine, l'une des raisons d'être des brevets était d'encourager les inventeurs à publier leurs travaux en leur fournissant une protection légale avant que le brevet ne tombe dans le domaine public. Ce qui pourrait être vu comme une forme de système libre de droits puisqu'une fois le brevet arrivé à son terme, l'invention tombe dans le domaine public. Mais le système actuel de brevets est un lamentable échec, un champ de bataille corrompu. En particulier les brevets sur des non-innovations absolues (pensez au brevet du rectangle à coins ronds d'*Apple*). Seules y trouvent leur compte

les grandes entreprises qui copient allègrement tout en s'accordant à ne pas se gêner mutuellement. Les nouveaux, en revanche, sont des cibles faciles ; il s'ensuit que le système des brevets étouffe l'innovation au lieu de la promouvoir.

L'avenir de l'OSHW

Tessel : Quels seraient les avantages si le monde passait aux sources libres ?

Sébastien : D'abord une foule d'occasions d'apprendre. Du point de vue de l'économie, je ne sais pas. Les innovations libres et l'intensification d'une coopération étroite et simplifiée entre individus plus nombreux contrebalanceraient-elles les avantages d'une conception secrète ? Je l'ignore, mais j'aime expérimenter et c'est aussi pourquoi je m'engage dans des projets comme *Milkymist*. Je vois que l'approche OSHW attire aujourd'hui des individus très doués, et ça, c'est positif.

Tessel : Quel sera le rôle futur de l'OSHW ? Remplacera-t-il le matériel non libre ?

Sébastien : Pas pour l'instant, car l'essentiel du matériel est manufacturé par des entreprises comme *Apple* ou *Intel* qui n'ont rien à faire de l'OSHW et détiennent un savoir-faire considérable qu'elles ne sont pas prêtes à divulguer. D'autre part, la plupart des adeptes de l'OSHW – du moins ceux qui ne viennent pas à l'EHSM – n'ont pas l'intention de créer du matériel à un niveau industriel. J'espère qu'à long terme la situation s'améliorera, mais ce sera long et difficile et les mentalités devront changer.

(120643 – version française : Claire Dodé)

Liens

- [1] <http://milkymist.org/>
- [2] <http://spaceflightsystems.grc.nasa.gov/SOPO/SCO/SCaNTestbed/>
- [3] www.ohwr.org/
- [4] www.ohwr.org/projects/cernohl/wiki
- [5] <http://ehsm.eu/>
- [6] www.elektor.fr/110447



appel

devenez **auteur** chez Elektor !

Vous avez une grande expérience de l'électronique ou d'un domaine connexe (robotique, micro-informatique, audio, domotique...) et vous avez envie de partager votre savoir : **publiez votre livre aux éditions Elektor !**

Nous vous accompagnerons dans l'écriture de votre ouvrage et nous chargerons de la mise en page, de l'impression, de la commercialisation en librairie en France et dans les pays francophones et bien sûr sur l'internet (Amazon).



La relation éditeur-auteur est encadrée par un contrat d'édition. Vous serez rémunéré en droit d'auteur, à savoir un pourcentage du prix de vente pour chaque exemplaire vendu.

Lancez-vous !

Un jour vous avez été un électronicien débutant, aujourd'hui vous serez un auteur débutant. Vous avez des connaissances à partager, une passion à communiquer. Vous pouvez en tirer de la satisfaction et ... des revenus. Chaque année, Elektor paie à ses auteurs des centaines de milliers d'euros de droits. Pourquoi pas à vous ?

Présentez-nous votre projet par courriel en nous envoyant une description.

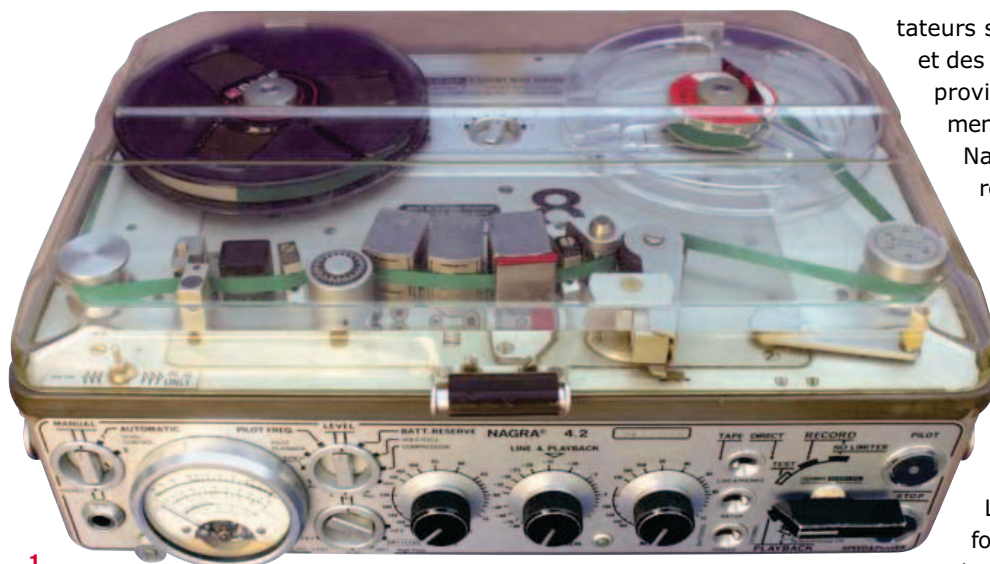
Sur la page www.elektor.fr/auteur vous trouverez des renseignements complémentaires qui guideront votre démarche.

Allez-y et le deuxième pas sera fait.

Le premier, c'est celui qui vous a conduit à lire jusqu'ici.

écrivez un livre technique

Nagra IV : LE magnétophone à bande un plaisir pour le reporter, vers 1968



1

Peter Beil (Allemagne)

Il a été un temps le plus encensé des magnétophones portatifs au monde, et pourtant peu de lecteurs d'Elektor en auront entendu parler. Le Nagra (**fig. 1**) était destiné aux applications professionnelles portatives, que ce soit pour la radiodiffusion ou pour l'enregistrement audio synchrone en cinéma ou télévision.

Son père spirituel, Stéphane Kudelski, polonais de naissance, l'a conçu en fonction des normes techniques les plus élevées et mesurables à l'époque, en employant des transistors étasuniens, des potentiomètres anglais, des commu-

tateurs suisses, des condensateurs hollandais et des connecteurs allemands. Le nom Nagra provient du mot polonais pour enregistrement. Sa qualité et sa fiabilité ont acquis au Nagra un statut de légende. On a pu en rencontrer des froids polaires de l'Alaska jusqu'aux chaleurs torrides du Sahara. L'époque des magnétophones portatifs à bande avait commencé dans les années 50 avec des appareils entraînés par des mécanismes à ressort. Le Nagra III date du début des années 60, suivi bientôt par le Nagra IV, devenu le modèle le plus répandu.

Le Nagra tire son énergie de 12 piles au format D (R20) qui occupent ensemble un tiers du volume de la machine (**fig. 2**). La tension non régulée des piles était utilisée par le bobinage rapide et l'amplificateur de puissance pour la lecture. Ce dernier délivrait un son clair, d'une intensité et d'une qualité remarquables, avec une puissance de 1,5 W. Si vous étiez économe du bobinage rapide et de l'écoute sur HP, l'autonomie d'un jeu de piles était suffisante. Par la suite, on a trouvé des accumulateurs rechargeables dans ce format, ce qui faisait une dizaine de kilos à coltiner.

Une référence de tension interne de -10 V gouvernait toutes les parties du circuit qui traitaient le signal modulé. Si vous vous demandez le pourquoi de ce -10 V (nous y reviendrons), sachez que tout le circuit, contrairement à l'habitude, avait le

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

2



3





pôle positif à la masse. Peut-être était-ce lié aux transistors PNP au germanium utilisés à l'origine. Pour assurer la rétro-compatibilité, il n'était plus possible de changer. Cela a causé bien des maux de tête dans l'industrie naissante des accessoires. Un simple contact avec un autre appareil pouvait créer des courts-circuits parce que la plupart avaient le pôle négatif à la masse. C'est ce qui se passait quand sur un câble de connexion le blindage était raccordé aux deux bouts...

L'électromécanique n'offre pas de grandes surprises. Le moteur de cabestan entièrement blindé tournait à trois vitesses régulées électriquement, 9,5, 19 et 38 cm/s (3,75, 7,5 et 15 pouces/s). Un mécanisme savant d'entraînement et de friction actionnait les bobines débitrice et réceptrice avec le seul et unique moteur déjà évoqué (fig. 3). À pleine tension, il pouvait rembobiner assez vite et avancer à double vitesse. Des huiles et graisses spéciales, appliquées sur demande ou chez les revendeurs, permettaient le fonctionnement à des températures ambiantes de -20 à +50 °C (!) (-4 à +122 °F).

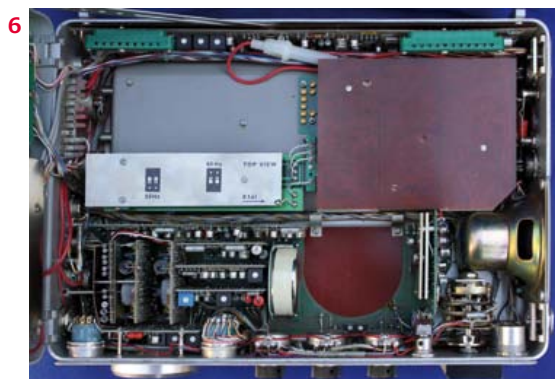
L'appareil représenté, un Nagra 4.2 des années 70, d'aspect identique à celui des premières machines, représente le summum de perfec-

tionnement technique de la série. On trouve sur la face avant, outre les organes de commande qui se passent d'explication, du côté droit, deux indicateurs rotatifs, dits en étoile (fig. 4). Ils permettaient un contrôle général des paramètres de fonctionnement tout en ne consommant qu'une fraction du courant d'un voyant.

Toute l'électronique était faite de composants discrets. En ce temps, les bons circuits intégrés audio étaient encore à l'étude, ils avaient mauvaise réputation quant à la fiabilité, la distorsion et le bruit. C'est seulement des années plus tard qu'un modèle à circuits intégrés a été construit. Les entrées et sorties passaient par des transformateurs symétriques, alors que les étages amplificateurs internes étaient asymétriques. Une des caractéristiques spéciales était la sortie symétrique flottante de 4,4 V d'amplitude, un ordre de grandeur imposé en ce temps-là pour les lignes de transmission par les autorités des télécommunications (fig. 5).

L'appareil était constitué de modules — déjà ! — disposés sur une carte mère. Tout était enfiché, avec des connecteurs verrouillés à contacts dorés (fig. 6).

Au passage, si le porte-fusible d'auto-radio au





milieu en haut arrête l'œil exercé de l'ingénieur électronicien, c'est parce qu'il n'est pas d'origine. Il est là parce que M. Kudelski, malgré son génie, avait oublié une chose : le fusible...

L'équipe avait consacré une attention particulière à l'instrument indicateur. Ils avaient à prendre en compte les différentes normes et conventions relatives aux indications. Le Modulomètre respectait parfaitement toutes les prescriptions, en combinant brillamment les accélérations de l'aiguille et l'inertie de l'indication (fig. 7).

Chacune des trois vitesses d'enregistrement était réglable individuellement pour la réponse en fréquence, égalisée pour une correction de la distorsion standard et à faible bruit. Même la symétrie HF et les distorsions par harmoniques 3 et 4 pouvaient être optimisées. Suivant les appareils, nous avons obtenu des bandes passantes de 20 Hz à 18 kHz (pour certains) à -2 dB. Avec un rapport signal/bruit non pondéré de -60 dB, ce résultat pouvait s'aligner avec celui de n'importe quelle machine de studio.

L'enregistreur comportait deux entrées microphone et des entrées supplémentaires configurables par l'utilisateur (fig. 8). Les entrées micro prenaient la forme de transformateurs d'entrée ou bien d'amplificateurs avec symétrisation électronique. Sur les premiers modèles, il fallait enficher le préamplificateur adapté au type de micro (dynamique haute ou basse impédance, condensateur etc.) (fig. 9). Par la suite, le préamplificateur est devenu commutable.

Les entrées étaient adaptables à un point extrême ; on n'avait besoin que rarement de la pré-atténuation commutable interne. Par la combinaison d'un limiteur enfichable et d'un limiteur de crête avec une courbe coudée, il était à peine possible de saturer l'enregistrement. Vous bénéficiiez aussi de la « saturation magnétique » de la bande, capable de « manger » des pointes supplémentaires.

Un problème s'est posé avec la tension d'alimentation fantôme de 48 V, déjà au rang des antiquités mais universelle. À l'époque, il était encore

techniquement impossible de produire cette tension d'alimentation de 48 V à partir des quelques 18 V disponibles. Naturellement, Nagra connaissait les convertisseurs continu-continu, mais les formes d'onde carrée et triangulaires utilisées bavaient trop loin dans la bande du signal utile. (De plus, le convertisseur était situé, à cause de la modularité, au point le plus sensible de la chaîne : l'entrée micro.)

C'est ici que le fabricant *Sennheiser* a apporté son aide, en mettant sur le marché les microphones HF à condensateur à basse tension, révolutionnaires à l'époque. Comparée aux alimentations fantômes 12 V, l'alimentation 10 V *T-Power* (le T signifie *Tonaderspeisung* [1], la Pologne est voisine de l'Allemagne) sur les conducteurs audio apparut comme la lumière du monde (probablement la raison de la référence de tension interne de 10 V). Sans qu'on sache pourquoi, la version 12 V fantôme a disparu sans un soupir, alors que le T-Power s'est maintenu longtemps. Malheureusement, ça n'a pas été sans quelques aléas. Dans ce système, comme l'annonce son nom germanique, la tension d'alimentation est appliquée symétriquement sur les deux conducteurs audio, ce qui effectivement n'interfère pas avec le signal utile. Cela dit, les amplificateurs régulés en tension n'apprécient pas du tout les tensions continues superposées à l'entrée.

Les entrées en courant spéciales du Nagra étaient largement indifférentes à la tension continue, mais il fallait deux condensateurs pour la bloquer. Et voilà le hic : un condensateur combiné à la résistance d'entrée de l'amplificateur constitue un filtre ! En termes de bande passante, cela nous condamne à des solutions de compromis. Si vous raccordez par erreur un microphone dynamique à une entrée T-Power, toutes les graves disparaissent, parce que la tension continue magnétise la bobine mobile... Nous avons tout de même trouvé le moyen de nous accommoder de ces petits problèmes.

Revenons à l'appareil enregistreur. On voit sur le dessus une tête magnétique supplémentaire entre les têtes d'enregistrement et de lecture. C'était l'élément essentiel de toute tâche de synchronisation : la tête de tonalité pilote (fig. 10). Voici quelques détails à son sujet : pour éliminer toute variation de synchronisation en studio, p. ex. par une dérive de vitesse, le matériau de la bande son séparée (*sep-mag* pour *separate magnetic*) était du même format et portait les mêmes perforations que la bande image [2]

(fig. 11). Les deux machines, lecture image et repiquage son, sont entraînées par des moteurs dits synchrones, dont la vitesse de rotation est déterminée non par la tension mais par la fréquence. Par conséquent, la vitesse de rotation de ces moteurs est identique et ils sont synchrones, asservis à la fréquence du secteur alternatif.

Les caméras film (la plupart alimentées par des piles) contenaient un petit générateur qui délivrait pour 24 images par seconde (plus tard 25) une fréquence exacte de 50 Hz. Si la vitesse de la caméra changeait (en fonction de paramètres mécaniques divers et nombreux), la fréquence variait proportionnellement. Cette fréquence était enregistrée au milieu de la bande son par la tête de tonalité pilote évoquée plus haut.

Pour éviter tout ronflement audible, l'entrefer de la tête pilote était tourné de 90°, ce qui éliminait la tonalité à pratiquement 100%.

Lors du transfert d'un enregistrement sonore sur la machine de repiquage son entraînée par le secteur alternatif, cette fréquence est comparée à la fréquence du secteur et utilisée pour réajuster électroniquement soit la vitesse du (ici le Nagra), soit celle de l'enregistreur. À cet effet, le Nagra permettait une correction de plus ou moins 4%, au moyen du *Synchronizer*. Pour des écarts supérieurs, on pouvait recourir à des accessoires ad hoc, ce qui n'arrivait toutefois que dans des cas extrêmes.

Une autre fonction, liée au *Synchronizer*, dite *Playback Procedure*, était utilisée pour les morceaux vocaux par exemple. Avant le démarrage, on enregistrait la fréquence du secteur comme tonalité pilote. Le Nagra recevait la tonalité pilote de la caméra, la comparait à la tonalité issue de la bande et laissait au *Synchronizer* le soin de compenser la différence. Ce système s'est montré efficace pendant des décennies.

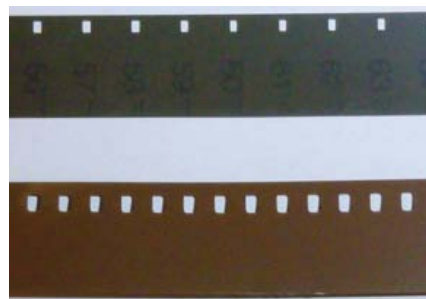
L'interface sous la forme de divers connecteurs constituait un système vraiment « ouvert » et la gamme d'accessoires était infinie, depuis les bobines de 3 000 pieds de bande, en passant par les tables de mixage et les appareils de pilotage à quartz, jusqu'aux applications sans fil. Le prix de vente était indiscutablement élevé pour l'époque, une version complètement équipée (sans valise de transport ni alimentation secteur) coûtait quelque 7 ou 8000 € d'aujourd'hui.

On ne peut manquer de signaler trois gadgets : le compteur de longueur de bande incorporé dans le galet à alignement automatique, le galet stroboscope et un nettoyeur de bande sous la forme

d'un coin métallique (figures 12 et 13).

Une machine de ce calibre avait naturellement des écouteurs appropriés. À l'époque, seule la firme *Beyer Dynamic* pouvait fournir un modèle suffisamment linéaire, dit *Normal-Telefon* (écouteurs de référence) de $2 \times 25 \Omega$. Sous une forme modifiée, ces écouteurs sont restés en usage plus de cinquante ans !

Ce magnétophone était monophonique (voir l'encadré). Naturellement, une version stéréo a suivi, et dans les premiers temps de l'ère de la vidéo, une version avec horodatage (*time code*). La technologie numérique en était encore à ses débuts à la fin des années quatre-vingts, mais une version numérique est apparue dès 1992, avec enregistrement sur quatre pistes PCM sur bande standard avec une profondeur incroyable de 24 bits ! Naturellement, le développement continue ; aujourd'hui par exemple il existe un Nagra à mémoire flash avec tous les paramètres nécessaires à un enregistrement numérique moderne du son.



11



12



13

(120570 - version française : Jean-Paul Brodier)

Liens

- [1] www.ips.org.uk/faq/index.php?title=Tonader_Power
- [2] http://fr.wikipedia.org/wiki/Cinéma#Techniques_du_son

Pourquoi préfère-t-on l'enregistrement monophonique pour les films ?

Si vous enregistrez de la parole avec un micro stéréo (ou avec deux micros séparés), la source sonore occupera une position bien définie dans l'espace acoustique virtuel. Maintenant, si vous modifiez l'angle de capture ou les distances à l'intérieur de la scène, comme c'est courant dans les films, vous créez une nouvelle position dans l'espace, différente de celle définie par l'enregistrement stéréophonique. Or l'ouïe humaine capte de tels changements spatiaux avec précision. Toute altération de ce genre est susceptible de déranger le spectateur. Aucun effet spécial, ni fondu ni panoramique, ne suffirait à masquer ce changement perçu par l'oreille. C'est pourquoi la parole enregistrée en mono ne trouve sa position définitive dans l'espace qu'au moment du montage son. Le produit final est parachévé par le mixage multipiste avec des « effets » sonores multi-canaux.

Défi énergie verte RL78

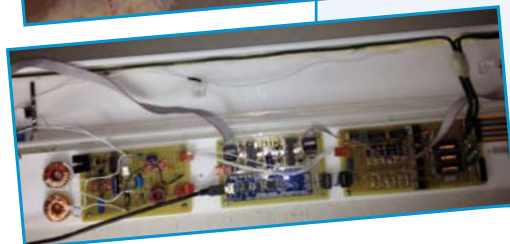
les gagnants

L'ambition du défi énergie verte RL78 était d'amener les ingénieurs à repenser leurs méthodes de conception afin de les mettre au diapason d'une époque où faible consommation, efficacité énergétique et ressources renouvelables font désormais partie du vocabulaire quotidien (Elektor, mai 2012, p.77). Renesas et ses partenaires (juillet & août 2012, p.8) avaient donc mis les participants de plus de 67 pays au défi de construire une application verte – récupération d'énergie, mesures ou autres systèmes à basse consommation – autour d'un microcontrôleur RL78 (juin 2012, p.78).

Merci à tous les participants, et félicitations aux gagnants !

Documentations et résumés :

<http://www.circuitcellar.com/contests/renesasRL78challenge/>



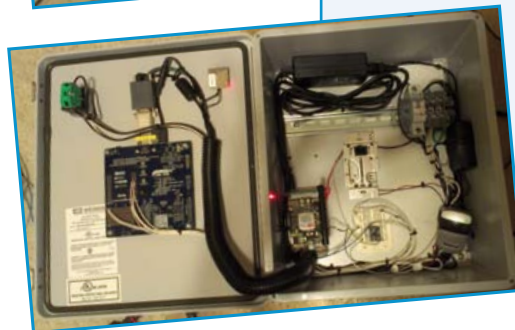
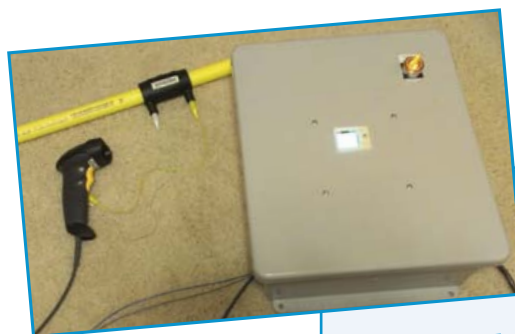
Premier prix

Robot de nettoyage électrostatique

Scott Potter (États-Unis)

Les miroirs des centrales solaires thermiques, ou héliostats, suivent la course du soleil et doivent rester propres pour optimiser la production de vapeur entraînant les alternateurs. Alimenté uniquement par cellules photovoltaïques, le Robot de nettoyage électrostatique à base de RL78 est un moyen de nettoyage à la fois fiable et novateur. Le robot traverse le miroir et utilise un champ électrique CA de haute tension pour éliminer poussières et débris.

1



Deuxième prix

Soudage en nuage par électrofusion

Michael Hamilton (États-Unis)

Ce poste de soudage en nuage par électrofusion consomme environ 400 fois moins d'énergie que ceux du commerce, et peut souder des raccords en polyéthylène de 5 à 50 mm de Ø. Piloté par un RL78, le poste lit un code à barres placé sur le raccord pour fixer les paramètres de fusion et de traçabilité. Outre les données du code, le système enregistre la localisation GPS si une carte SD est présente, et transmet les données de chaque fusion à une base de données en nuage pour assurer suivi et contrôle qualité.

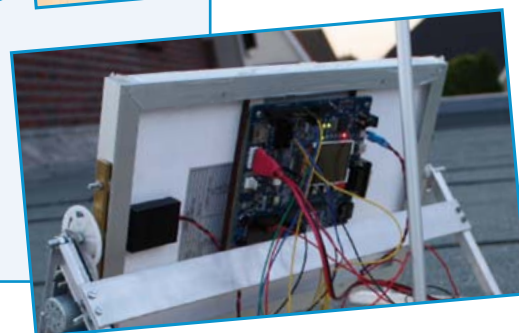
2

3

Troisième prix The Sun Chaser : une station de référence GPS

Sjoerd Brandsma (Pays-Bas)

Le « chasseur de soleil » est un système de récupération d'énergie solaire qui calcule automatiquement l'orientation face au soleil d'un panneau solaire monté sur disque rotatif. L'orientation est calculée d'après la localisation GPS enregistrée. La position exacte du panneau peut être déterminée à l'aide d'une boussole externe, de l'accéléromètre interne, d'un moteur CC et d'un moteur pas à pas. Le système utilise la carte d'évaluation Renesas RDKRL78G13 et son noyau temps réel Micrium μ C/OS-III.



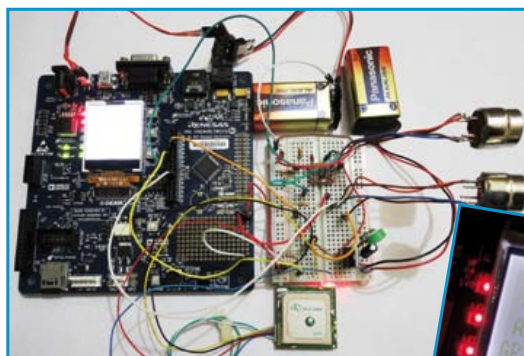
Mention honorable Chauffe-eau par concentration solaire

Pierre Berquin (France)

Ce chauffe-eau solaire piloté par la carte d'évaluation RL78 concentre en permanence les rayons du soleil sur une conduite d'eau. Le déflecteur, pourvu d'un contrepoids pour faciliter son inclinaison, ajuste automatiquement l'angle de réflexion pour récupérer un maximum d'énergie solaire avec un minimum de consommation électrique.

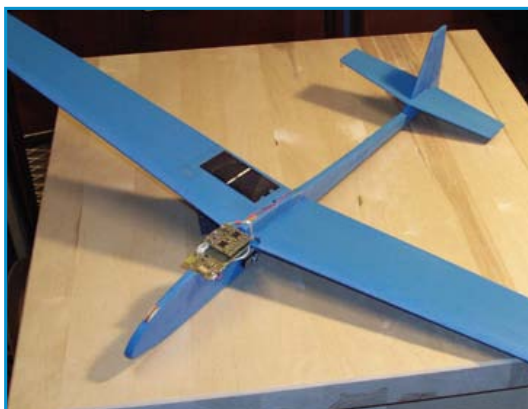


Mention honorable Traceur de qualité de l'air



Raul Alvarez Torrico (Bolivie)

Des doutes quant à la pureté de l'air le long de votre promenade ? Portatif, le Traceur de qualité d'air dresse des « cartes de pollution » par relevé des niveaux de CO_2 et de CO afin de déterminer les trajets les plus sains. Construit autour d'un RDKRL78G13, le Traceur reçoit les géodonnées de son module GPS, lit les concentrations de CO_2/CO le long d'un certain trajet, et stocke les données en carte SD. Un PC permet d'analyser ces données, de les représenter, et de les charger automatiquement dans une base de données MySQL en ligne qui présente les données sur une carte Google.



Mention honorable **High-Altitude Low-Cost** **Experimental Glider (HALO)**

Jens Altenburg (Allemagne)

Le projet de planeur HALO comprend trois éléments. Un ballon-sonde sert de transporteur. Un planeur (la charge du ballon) assure le retour. Une base au sol (hors-concours) sert aux communications et à l'affichage des données télémétriques. Le simulateur de vol REFLEX a été utilisé pour les essais. Le planeur possède un micro-récepteur GPS, des capteurs, et un microcontrôleur à basse consommation. Il peut décoller, atteindre une altitude préprogrammée, et revenir à un certain point de coordonnées.



Mention honorable **« Capteur météo »** **solaire radio-piloté**

Grzegorz Kaczmarek (Pologne)

Le « Capteur météo » facilite la mesure des paramètres météo. Conçu autour d'un



RL78, il mesure de façon cyclique température, humidité, pression et tension d'alimentation, et transmet les résultats à l'aide d'émetteurs-récepteurs radio numériques. Les récepteurs écoutent les données entrantes sur le même canal radio. Ce capteur simplifie le recueil des données météo et la construction de réseaux locaux de mesure, et est optimisé pour une faible consommation d'énergie et une longue utilisation de la batterie.

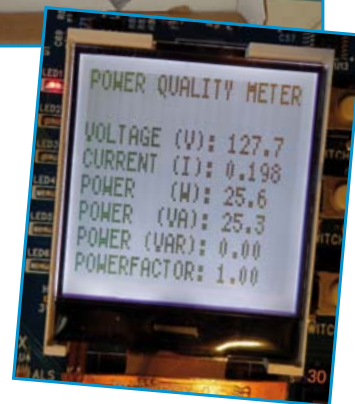
Mention honorable **Énergimètre portatif**

André Barbosa (Brésil)



Surveiller la consommation électrique de son foyer est une vertu en vogue. L'Énergimètre portatif recourt à un RL78 pour lire facteur de puissance, distorsion

harmonique totale, fréquence du réseau, tension et consommation électrique, et stocke les données pour analyse.



Hexadoku casse-tête pour électroniciens

Alors, comment ça va ? Pas trop de crises de foie après les agapes de fin d'année ... ? Champagne, foie gras, chocolat et toutes ces douceurs auxquelles il est si difficile de résister. Et la tête ? Vous ne contesterez pas l'utilité d'un petit décrassage mental, ça ne fait pas de mal. Aérez-vous les méninges. Détendez-vous ! Remplissez la grille selon les règles, envoyez-nous votre solution, et vous gagnerez peut-être l'un des quatre chèques-cadeaux Elektor mis en jeu.

Après ça, vous serez d'aplomb pour reprendre vos cogitations électroniques.

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

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

Participez et gagnez !

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

Où envoyer ?

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

La solution de la grille du numéro de novembre 413 est **BD18A**

Le gagnant du **chèque-cadeau Elektor** d'une valeur de **100 €** est Ron Ware de Wiley dans le Surrey, en Grande-Bretagne.

Les **3 chèques-cadeaux Elektor** d'une valeur de **50 €** chacun vont à : Panagiotis S. Krokidis de Plagiari, en Grèce,

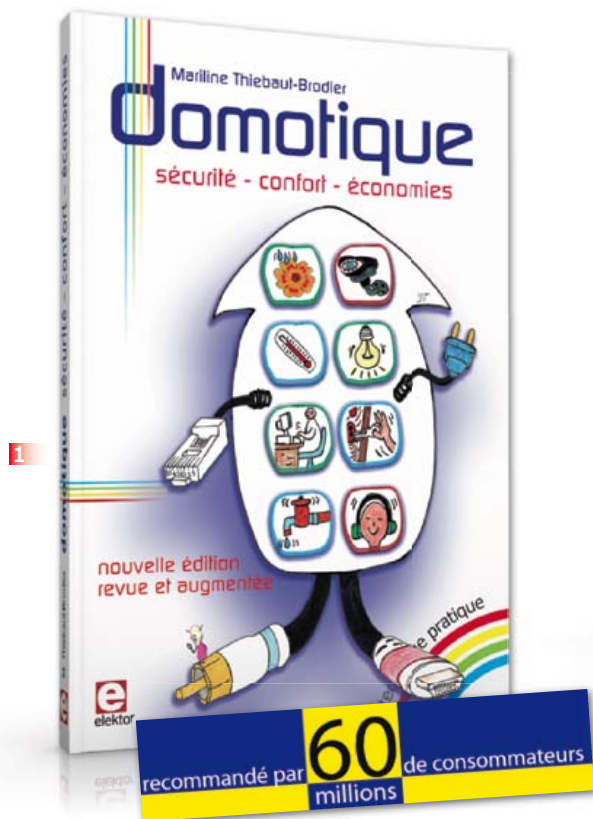
Didier Pelegri de Charnay-lès-Mâcon en France et P. Scheepers de Geldrop, aux Pays-Bas.

Bravo à tous et félicitations aux gagnants !

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

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



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

1 Domotique

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

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

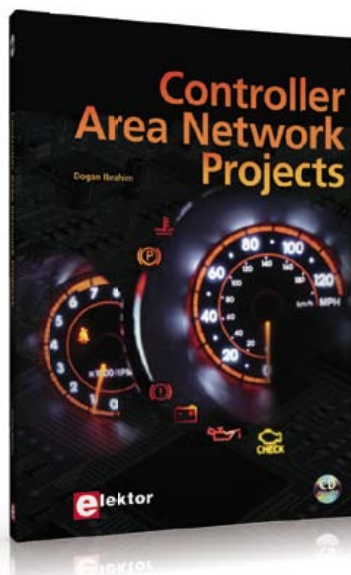
256 pages • ISBN 978-2-86661-182-8 • 33,00 €

livre avec CD-ROM

2 Controller Area Network Projects

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

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



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

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

35 projets d'initiation en C avec la carte mbed NXP LPC 1768

3 Microcontrôleurs RISC 32 bits à architecture ARM

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

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

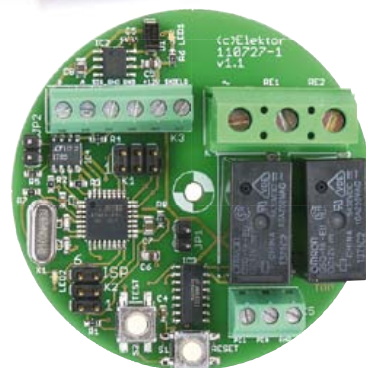
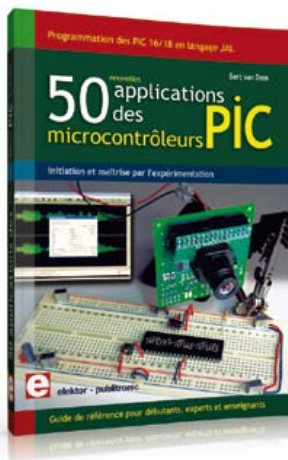
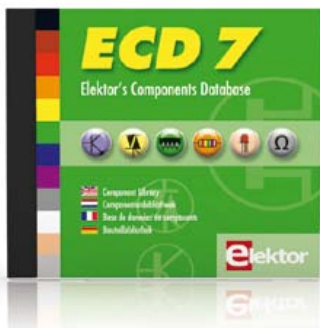
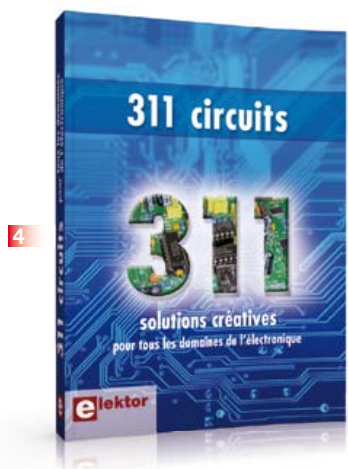


Solutions créatives pour tous les domaines de l'électronique

4 311 circuits

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

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



Base de composants interactive d'Elektor

5 ECD 7

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

ISBN 978-90-5381-298-3 • 29,50 €

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

6 Embarquez Linux !

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

Carte Linux Elektor (montée et testée)

Réf. : 120026-91 • 64,95 €

23 projets ludiques et instructifs à construire soi-même

7 Intelligence artificielle

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

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

Initiation et maîtrise par l'expérimentation

8 50 applications des microcontrôleurs PIC

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

feront découvrir une activité enrichissante.

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

Commande de charge sous 230V par RS485

9 Relais : un module encastrable dans une prise murale

En attendant que nos maisons soient assez intelligentes pour s'intéresser à nous, c'est nous qui nous intéressons à elles, avec cette platine compacte pour commuter deux charges domestiques, des volets roulants p. ex., avec le bus Elektor. Sur cette carte encastrable, deux entrées permettent de commander directement le microcontrôleur depuis l'interrupteur mural. Un module de plus pour la domotique contrôlée par PC, tablette et téléphone tactile !

Module prêt à l'emploi

Réf 110727-91 • 44,95 €

Réf 110727-92 (lot de 3 modules) • 119,95 €

Informations complémentaires et gamme complète sur :

www.elektor.fr/e-choppe

Elektor/Publitronic SARL

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

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

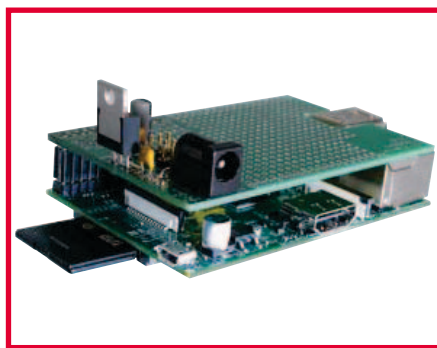
@ : ventes@elektor.fr

•bientôt dans Elektor



LCR-mètre hautes performances

Un LCR-mètre n'est pas un appareil que l'on trouve ordinairement dans le laboratoire d'un amateur d'électronique. Pourtant la recrudescence des montages à composants montés en surface ou l'utilisation de plus en plus fréquente de selfs dans les alimentations à découpage, rend pratiquement obligatoire l'utilisation d'un tel appareil de mesure. Celui que nous vous présenterons est de conception française et fonctionne soit comme instrument autonome, soit avec un ordinateur comme périphérique de visualisation et de commande.



Raspberry Pi : carte de prototypage

Raspberry Pi est un système de développement abordable au format carte de crédit, auquel l'utilisateur peut connecter directement une flopée de périphériques tels qu'un clavier ou un moniteur. C'est la carte idéale pour d'innombrables variations sur le thème « je voudrais le faire moi-même ». C'est pourquoi nous avons imaginé cette carte de prototypage, afin de stimuler votre verve créatrice. Cela commence par une alimentation régulée et un circuit à pastilles. Le format de la carte a été étudié pour en faciliter le couplage avec le module *Raspberry Pi*.



Thermo...

Encore un thermomètre ? Noooooon ne partez pas ! Vous n'avez encore jamais rien vu de tel. C'est un *thermolivre*. Planquer un appareil de mesure dans un livre, c'est une idée de présentation originale, mais pas assez pour justifier une publication. Ce circuit a d'autres fonctions intéressantes. Il ne se contente pas de mesurer la température, mais aussi l'humidité relative. Et vous n'êtes même pas obligé de choisir entre les deux fonctions, il les affiche à tour de rôle. Et enfin si vous frappez dans les mains, il l'entend et change de fonction.

Sous réserve de modifications. Le numéro de mars paraîtra le 26 février

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

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



The screenshot shows the elektorlabs website interface. At the top is the logo 'elektorlabs' with the tagline 'Sharing Electronics Projects'. Below the logo is a navigation bar with links: Home, News, Proposals, In Progress, Finished, and a search bar. The main content area features a 'mains gate help wanted' banner with a 3D model of a gate. Below this are three columns of project listings: 'Proposals' (with sub-tabs 'Active' and 'Popular'), 'In Progress' (with sub-tabs 'Active' and 'Popular'), and 'Finished' (with sub-tabs 'Active' and 'Popular'). Each column shows project thumbnails and titles like '[120728] HDMI - SPDIF Adapter', 'The PSoCmorph', and 'Elektor LED earring'. On the right side, there's a 'About Elektor.LABS' section, a 'Create a Project' section with a video player, and a 'Challenges' section with a 'Win a PSoC 3 Development Kit!' challenge. At the bottom right, there's a 'Join a Project' section for the '[120728] HDMI - SPDIF Adapter' project.

NOUVEAUTÉ

MÉMOIRE PROFONDE

OSCILLOSCOPES SIGNAUX MIXTES PORTABILITÉ & PERFORMANCES



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

TOUS LES MODÈLES ONT DÉCLENCHEMENTS NUMÉRIQUES AVANCÉS, DÉCODAGE SÉRIEL (I²C, SPI, RS232, CAN, LIN, FLEXRAY), TESTS DE LIMITE DE MASQUE, MÉMOIRE SEGMENTÉE, FILTRAGE NUMÉRIQUE, MISES À JOUR DU LOGICIEL GRATUITES ET 5 ANS DE GARANTIE

www.USBmso.com/PS209



NIDays

WORLDWIDE GRAPHICAL SYSTEM DESIGN
CONFERENCES

Le rendez-vous annuel des ingénieurs,
scientifiques et enseignants

**Au CNIT Paris La Défense,
le 12 février 2013**

Programme et inscription
gratuite sur www.nidays.fr



Il est temps, plus que jamais, d'innover pour relever les défis du monde moderne !

Vous préférez construire l'avenir plutôt que le subir ? Alors ne manquez surtout pas l'édition 2013 de NIDays et son lot d'innovations en matière de systèmes de mesure, de contrôle, d'automatisation et embarqués :

- 70 stands de démonstrations
- 2 conférences plénières, avec André Borschberg (Solar Impulse) en invité d'honneur
- 50 conférences techniques et métiers (dont 20 présentations d'utilisateurs)
- 14 sessions de TP pour s'initier aux matériels et aux logiciels
- 2 compétitions de robotique pour l'enseignement



Programme et inscription gratuite sur www.nidays.fr ▶ 01 57 66 24 24