



e-lock

la puce de sécurité d'Elektor

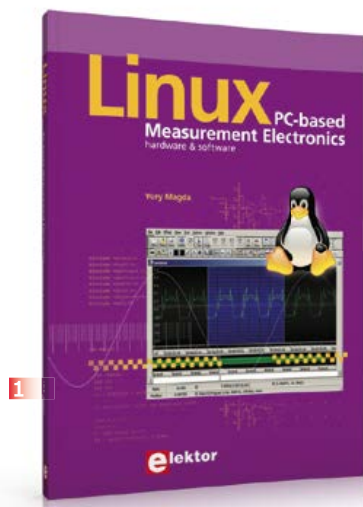


e-lock est inviolable...

prouvez le contraire & gagnez 25.000 \$
www.elektor.com/e-lock

● **e-lock** | alimentation de labo pilotée par Platino | **l'atelier du microcontrôleur (1)**
calcul de transfos de courant | **cerveau de servos** | Raspberry Pi comme passerelle vers l'internet
passerelles radio 433 MHz | générateur HF à synthèse numérique directe (DDS)
multiprise asservie | commande d'électrovanne ● **drôle de condensateur ?**
sondes : chassez le crocodile ● **récepteur Atwater Kent model 30 (1926)**





1

1 **Linux – PC-based Measurement Electronics** matériel et logiciel

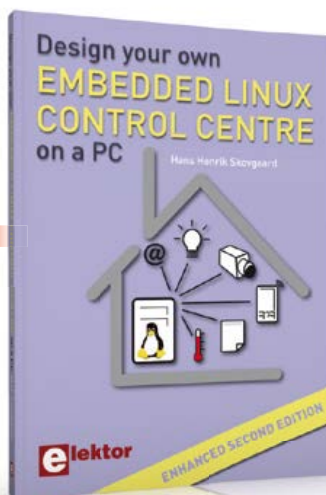
Apprenez rapidement à construire des applications sous Linux, pour collecter, traiter et afficher des données sur un PC à partir de capteurs analogiques et numériques. Ne vous contentez pas de commander des circuits raccordés à un ordinateur. Apprenez aussi à transmettre les données via un réseau ou à communiquer sans fil avec des systèmes embarqués ! Aucun appareillage spécial ni logiciel coûteux n'est requis. Les composants électroniques et outils logiciels utilisés sont gratuits ou à faible coût.

Langue : anglais • 263 pages • ISBN 978-1-907920-03-5 • 34,50 €

2 **Design your own Embedded Linux Control Centre on a PC**

2e édition entièrement revue et augmentée de 180 pages

Ce livre (en anglais) ne parle ni de X10, ZigBee, Z-wave, ni d'aucun autre de ces systèmes de domotique plus sophistiqués les uns que les autres, car c'est un livre pour les adeptes du fais-le-toi-même, ceux qui goûtent aux joies du recyclage de matériel existant. Au lieu de prendre la poussière et d'encombrer les placards, un



2



3

« vieux » PC peut encore servir pour la commande et la régulation de processus domestiques, sans engendrer aucun frais inutile, tout en procurant plaisir et satisfaction aux usagers.

Langue : anglais • 412 pages • ISBN 978-1-907920-02-8 • 39,50 €

3 **Module d'extension pour les cartes à µC Linux et autres**

La platine d'extension pour la carte Linux d'Elektor peut aussi se connecter à d'autres cartes à µC. Par exemple la carte à huit relais. La carte d'extension conviendra aux débutants pour qui Linux est (encore) trop compliqué ; elle conviendra aussi aux utilisateurs chevronnés qui préfèrent sculpter leurs programmes « dans le métal brut » (sans l'aide d'un système d'exploitation).

Réf 120596-91 • 42,95 €

4 **GNUBLIN LAN 32 Mo (avec RTC et relais)**

GNUBLIN est une carte GNU / Linux pour applications embarquées, idéale pour faire ses premiers pas dans le monde des microcontrôleurs fonctionnant sous Linux. La version GnuBLin est



4

équipée du processeur ARM9 LPC3131 de chez NXP et fonctionne avec un système Linux préinstallé qui s'amorce depuis une carte SD. Les interfaces essentielles au microprocesseur, comme I2C, SPI, la ligne PWM, les E/S numériques ou encore les entrées analogiques, sont dirigées vers l'extérieur.

Réf 130214-91 • 69,95 €

Autres produits GNUBLIN :

Bridge module • réf 130212-71

GnuPI Module pour Raspberry Pi • réf 130212-72

Module adaptateur BeagleBone-Black • réf 130212-74

Afficheur LCD (4x20 caractères) • réf 130212-92

Pilote de moteur pas à pas • réf 130212-93

Module IO Expander • réf 130212-94

Capteur de température • réf 130212-95

**informations complémentaires et
gamme Linux complète sur :**

www.elektor.fr/linux



Au service du génie

National Instruments met à la disposition des étudiants le matériel et le logiciel dont ils ont besoin pour développer leur expérience, aller au-delà de la théorie, et réaliser l'importance du rôle de l'ingénieur dans la société.

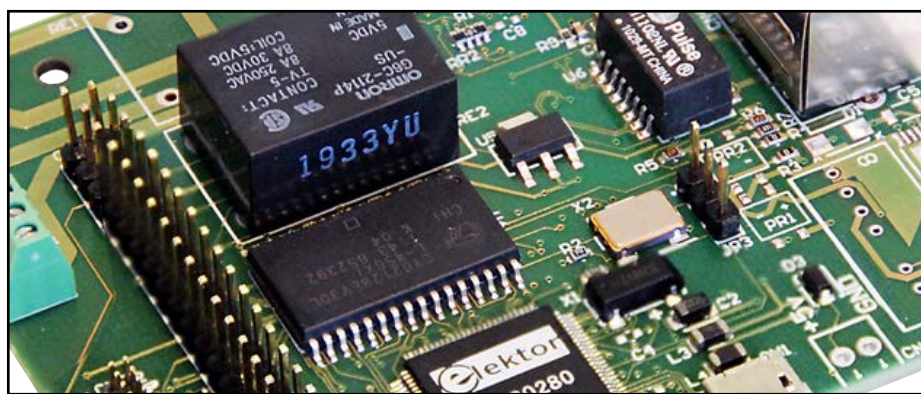
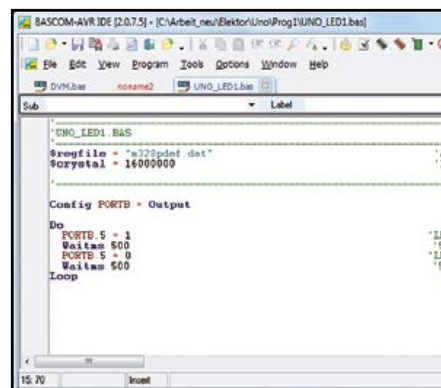
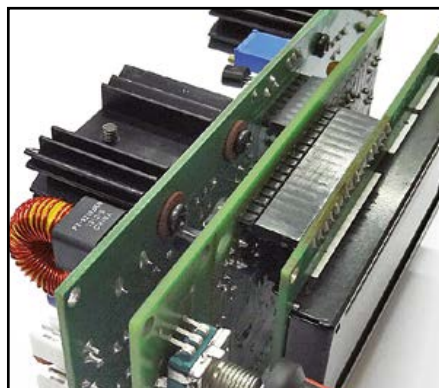
>> Découvrez comment NI supporte la prochaine génération d'innovateurs, en visitant ni.com/academic/f

01 57 66 24 24

NATIONAL INSTRUMENTS France • 2 rue Hennape – 92735 Nanterre Cedex, France • Tél. : 01 57 66 24 24 • Fax : (0)1 57 66 24 14 • Société de droit américain – capital social 1 000 dollars • US • 11500 N Mopac Expwy, Austin-Texas USA – 10056236 – 344 497 649 • RCS Nanterre – SIRET B 344 497 649 00048 – APE 516J - N.J.I. FR 57344497649

©2013 National Instruments. Tous droits réservés. National Instruments, NI, et ni.com sont des marques de National Instruments. Les autres noms de produits et de sociétés mentionnés sont les marques ou les noms de leurs propriétaires respectifs. Pour plus d'informations concernant les marques de National Instruments, veuillez vous référer à la partie Terms of Use sur le site ni.com/legal. 09680





● labs

18 Quel drôle de con... densateur ?

Un drôle de citoyen, avec deux cartes d'identité différentes.

37 gentil petit canard

Il est facile de collecter beaucoup d'informations, les rendre accessibles est une autre paire de manches.

● DesignSpark

16 afficheurs Nixie

drôle de composant n°4

Nixie, rime avec nostalgie.

30 alimentation de table régulée par Platino

modèle d'application embarquée

Pourquoi la fonction d'un

microcontrôleur se limiterait-elle à l'affichage de la tension et du courant ?

74 9^e jour – contours personnalisés (suite)

Finissons le contour de carte commencé la dernière fois pour un boîtier précis.



● projets

8 e-lock, la première puce de sécurité d'Elektor

L'e-lock est inviolable, prouvez-le contraire

Encore un article qui commence par « 2014, l'année de l'Internet des Objets... » ! Mais quid de la s-é-c-u-r-i-t-é ?

20 électrovanne mise au régime sous osmose inverse

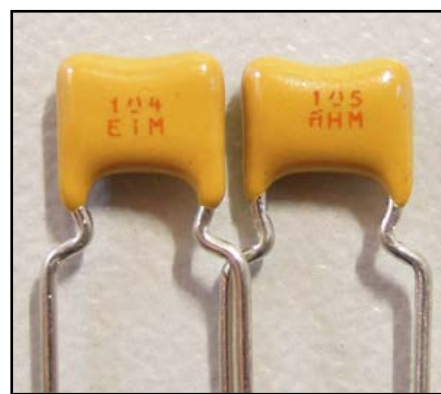
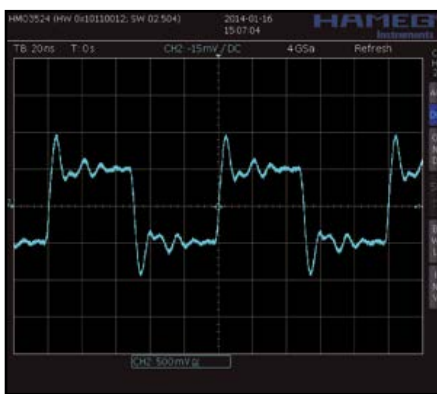
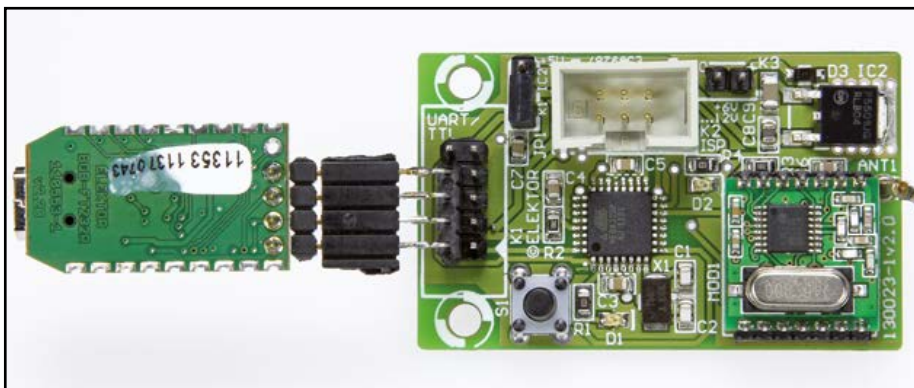
Actionner un robinet par un automate, c'est bien. Faire en plus des économies d'énergie substantielles, c'est mieux.

23 multiprise asservie

avec 1 seul composant

24 cerveau de servos

idéal pour le modéliste ferroviaire



38 journalisation de la fréquence du secteur

la écarts d'EDF sur clé USB

Les écarts de fréquence de la tension du secteur (50 Hz) indiquent la qualité instantanée du réseau.

44 sondes : chassez le crocodile

La mesure de signaux numériques rapides est souvent faussée par les sondes utilisées. Améliorez-les !

46 générateur HF à synthèse numérique directe (DDS)

Une autre conception de l'électronique

Avec des outils simples, il est possible de construire un impeccable générateur à synthèse numérique directe jusqu'à quelques centaines de mégahertz.

50 passerelles radio

De l'UART aux 433 MHz et retour

Pour la mesure et la régulation sans fil, il existe des modules radio, qui communiquent par une interface SPI.

58 ATmega sur l'internet (2)

Le Raspberry Pi comme passerelle vers le réseau

Un R. Pi suffit pour piloter via l'internet et le réseau local un ATmega32 relié par une ligne série.

62 calcul des transfos de courant

pour des mesures sans contact

Pour mesurer un courant sur le secteur sans contact galvanique, on utilise un transformateur de courant.

66 l'atelier du microcontrôleur (1)

Arduino et Bascom

Pourquoi les microcontrôleurs ont-ils envahi l'électronique ? Parce qu'on n'arrête pas le pro... cesseur.

● magazine

75 bientôt dans Elektor

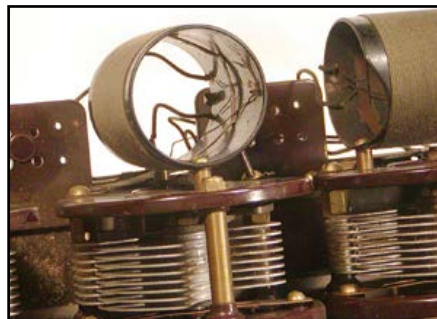
Avant-première des prochains numéros

77 hexadoku

vosre séance mensuelle d'hexadécimale ! Plongez !

78 Rétronique : récepteur Atwater Kent (1926)

On n'arrête pas le regret



37^{ème} année, n° 430
avril 2014

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

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

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

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

DROITS D'AUTEUR :

© 2014 Elektor International Media B.V.

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

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

Imprimé aux Pays-Bas
par Senefelder Misset - Doetinchem

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

Cher Kévin, je t'aime

nous ne nous connaissons pas, mais je me permets de te tutoyer car, contrairement à Denis Meyer, je **suis** un vieux sceptique : – *mind you* – ma fille était en terminale l'année de Tchernobyl.

Oui, la recherche de la dernière version est insensée quand on a le même travail à faire. Oui, la mise à jour automatique à l'insu de l'utilisateur est dangereuse à tous points de vue. Un de tes confrères informaticiens de haut niveau, dans une conférence à destination de la haute administration civile et militaire de notre République, sur les considérations de sécurité liées à

l'utilisation de Skype, l'assimilait à un virus, tout comme Windows. L'un et l'autre répondent à presque tous les critères de la définition du virus : pas souhaité, s'installe et se met à jour tout seul, on ne sait pas ce qu'il fait, ni comment s'en débarrasser...

Un RAID 5 logiciel sous Windoze XP devient inaccessible du jour au lendemain. Explication (trouvée péniblement) : Microsoft s'est invité en loucedé pour faire le tour des disques sous prétexte de mise à jour et a remis dans leur état d'origine les fichiers système qui avaient été modifiés pour permettre le fonctionnement du RAID logiciel. Chez moi, coproclaque !

Dans *La reine des pommes* de Chester Himes, un pauvre nègre se fait arnaquer par un magicien qui devait transformer ses dix billets de dix dollars en billets de cent. Tout au long du livre, il recherche activement le magicien, non pas pour lui réclamer ses 100 dollars mais pour lui en confier 100 autres, en espérant que ça va marcher. Ainsi des clients de Windows 3, 7, 8, sans compter les versions sans numéro...

Le titre américain du roman est *Fivecorner Square*, qu'on pourrait traduire par *deux et deux cinq* pour situer le niveau intellectuel du héros.

La nouveauté, c'est qu'il n'y aura plus de numéro de licence à saisir pour l'installation, seulement un numéro de carte bleue.

Sécurité, dis-tu ? Alors vivent Linux, SDCC, GCC, AVR-GCC, Raspberry Pi, Electric, Origano, Kicad...

Jean-Paul Brodier

Note du rédacteur en chef : ce cri du cœur de Jean-Paul Brodier répond à celui de Kévin publié ici même le mois dernier. Pour apprécier l'un, il est recommandé de goûter à l'autre. D.M.



Notre équipe

Rédacteur en chef :	Denis Meyer (redaction@elektor.fr)
Rédaction internationale :	Harry Baggen, Jan Buiting, Jaime Gonzalez Arintero, Jens Nickel
Laboratoire :	Thijs Beckers, Ton Giesberts, Wisse Hettinga, Luc Lemmens, Mart Schroijen, Clemens Valens, Jan Visser, Patrick Wielders
Coordination :	Hedwig Hennekens
Ont coopéré à ce numéro :	Jean-Paul Brodier, Robert Grignard, Hervé Moreau, Kévin Petit, NN
Service de la clientèle :	Jolanda van Kruchten
Graphiste :	Giel Dols
Elektor en ligne :	Daniëlle Mertens



France

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



United Kingdom

Carlo van Nistelrooy
+44 20 7692 8344
c.vannistelrooy@elektor.com



USA

Carlo van Nistelrooy
+1 860-289-0800
c.vannistelrooy@elektor.com



Germany

Ferdinand te Walvaart
+49 241 88 909-17
f.tewalvaart@elektor.de



Netherlands

Ferdinand te Walvaart
+31 46 43 89 444
f.tewalvaart@elektor.nl



Spain

Jaime González-Arintero
+34 6 16 99 74 86
j.glez.arintero@elektor.es



Italy

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



Sweden

Carlo van Nistelrooy
+31 46 43 89 418
c.vannistelrooy@elektor.com



Brazil

João Martins
+31 46 4389444
j.martins@elektor.com



Portugal

João Martins
+31 46 4389444
j.martins@elektor.com



India

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



Russia

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



Turkey

Zeynep Köksal
+90 532 277 48 26
zkoksal@beti.com.tr



South Africa

Johan Dijk
+31 6 1589 4245
j.dijk@elektor.com



China

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

Notre réseau



VOICE & COIL



audioxpress



vous connecte à



Nos annonceurs



CadSoft

www.cadsoftusa.com 21



Reichelt

www.reichelt.fr 84



Eurocircuits

www.elektorpcbservice.com 17



WIZnet

www.circuitcellar.com/wiznet2014 . . . 15



National Instruments

www.ni.com/academic/f 3



Pico

www.picotech.com/PS253 83

Pour placer votre annonce dans le prochain numéro d'Elektor

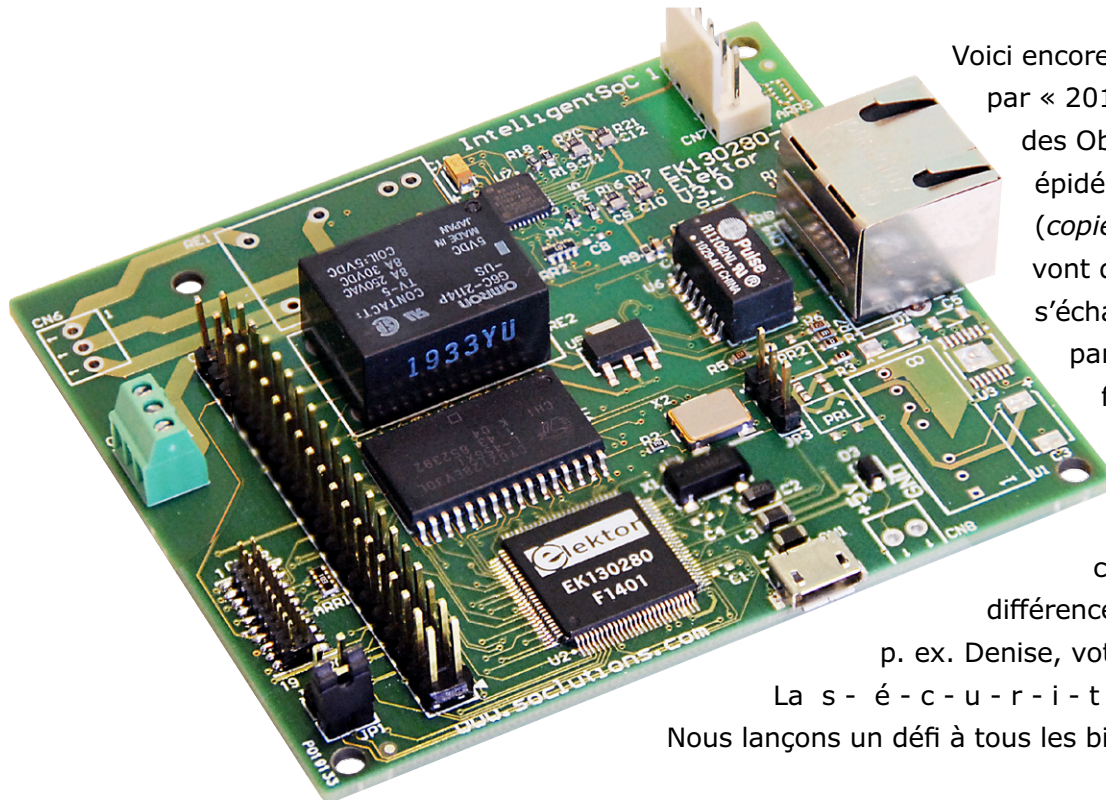
veuillez contacter Mme Ilham Mohammadi par téléphone au (+31) 6.41.42.25.25
ou par courrier électronique : i.mohammadi@elektor.fr

Vos correspondants

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

e-lock, la première puce de sécurité Elektor

L'e-lock est inviolable, prouvez-le contraire*



Voici encore un article qui commence par « 2014, l'année de l'Internet des Objets... » ! Selon une épidémie de prédictions (*copiées-collées*), les objets vont discuter entre eux, s'échanger des informations par l'internet. Il sera plus facile de communiquer avec votre cafetière qu'avec vos collègues. Mais comment la cafetière fera-t-elle la différence entre vous et...

p. ex. Denise, votre malicieuse voisine ?

La s - é - c - u - r - i - t - é , évidemment.

Nous lançons un défi à tous les bidouilleurs.

Eduardo Corral
(Espagne)

Développé par
Intelligent SoC
(info@soclutions.com)

La cafetière est peut-être le dernier de vos soucis, mais quand il s'agit d'objets de valeur mieux vaut prendre ses précautions. Elektor y a réfléchi et a développé en collaboration avec *Intelligent SoC* une puce hautement spécialisée pour protéger vos objets connectés : j'ai nommée l'e-lock, la première puce Elektor ! La puce e-lock permet de relier votre système de commande au réseau des réseaux ainsi que de le surveiller et commander depuis n'importe quel endroit du globe à l'aide de votre ordinateur, tablette ou téléphone tactile sans avoir à vous préoccuper de la sécurité des connexions et avec l'assurance que vous serez protégé contre les intrusions.

Le niveau de sécurité dont bénéficient nos ordinateurs et appareils mobiles ne devrait-il pas bénéficier

aussi aux millions d'appareils embarqués reliés à l'internet ? Leur vulnérabilité représenterait un sérieux obstacle à la saine croissance de l'Internet des Objets (IdO). Qui pourrait sereinement relier un appareil embarqué à l'internet en sachant qu'il peut être la proie de toutes sortes d'attaques lancées par des hackers en goguette ?

e-lock est la réponse

Assez disserté, revenons à la technologie. Nous avons pris contact avec *Intelligent SoC*, spécialiste-de-la-sécurité-des-appareils-embarqués [1] et décidé de développer conjointement e-lock, une puce — la première Elektor — spécialement conçue pour l'IdO et pour répondre aux deux problématiques qu'il implique : 1) permettre de

* bidouillez l'E-Lock → cassez le code → gagnez le prix de 61A8_{hex} dollars (\$25K). Détails sur www.elektor.com/e-lock.

gérer des appareils à distance via l'internet 2) le faire de manière sécurisée.

La puce e-lock est capable d'établir une connexion sécurisée à travers l'internet (TCP/IP) en utilisant le protocole de chiffrement TLS (*Transport Layer Security*). Elle possède 7 GPIO (E/S à utilisation générale), un CAN à 16 bits à quatre voies, un CNA à 12 bits et un bus I²C qui permet de discuter avec d'autres systèmes (**fig. 1**). La puce e-lock (un SoC) incorpore également une horloge temps-réel (RTC) capable de se synchroniser via l'internet à l'aide du protocole SNTP (*Simple Network Time Protocol*).

Carte d'évaluation

Dans son boîtier LQFP à 100 broches, la puce e-lock n'est pas facile à assembler à la main. C'est pourquoi nous avons conçu la carte d'évaluation (et d'application) qui vous permettra de tester la puce e-lock tout en servant de base à vos propres projets sécurisés pour l'IdO. Vous la trouverez sous la référence **130280-91** dans la boutique en ligne d'Elektor. Vous pourrez télécharger une description technique détaillée sur [2].

Pour vous aider à démarrer, la carte est livrée avec une application de démo qui permet de commander deux relais et surveiller la température de la pièce dans laquelle elle se trouve, le tout à distance et de manière sécurisée bien entendu. La carte possède également un connecteur d'extension destiné à accueillir de futures ajouts ou à ouvrir la porte au développement de vos propres applications. Le diagramme fonctionnel de la carte (**fig. 2**) se divise en plusieurs sections :

e-lock — le cœur de la carte. La puce gère les communications et le fonctionnement des appareils qui lui sont reliés. e-lock utilise deux protocoles de communication différents :

- Protocole de configuration Ethernet brut : utilisé pour configurer les principaux paramètres réseau et de sécurité tels que l'adresse IP, l'adresse de la passerelle, les serveurs DNS ou SNTP ainsi que les certificats, clés et autorités de certification. Cela se fait en utilisant l'adresse MAC unique de l'e-lock.
- Protocole d'application client/serveur TCP/IP sécurisé. Une fois l'e-lock configurée, elle se comporte comme un serveur TCP/IP sécurisé qui respecte le protocole d'application expliqué en détails dans la feuille de caractéristiques [2].

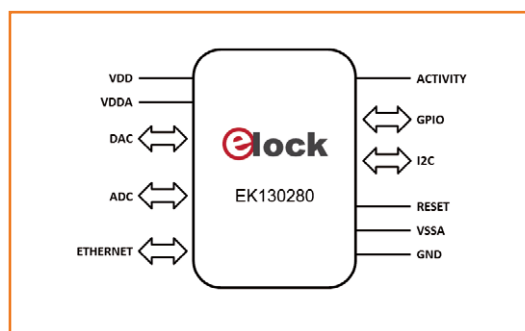


Figure 1.
Diagramme fonctionnel de la puce e-lock.

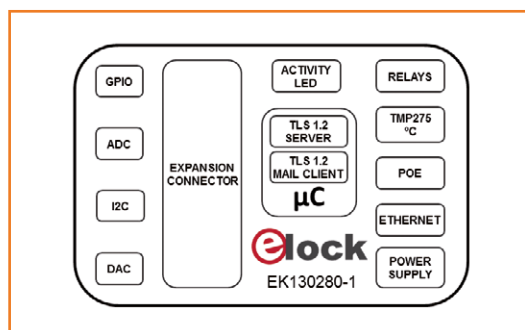


Figure 2.
Diagramme fonctionnel de la carte d'évaluation e-lock.

Principales caractéristiques

- Communication TCP/IP sécurisée bidirectionnelle avec TLS 1.2
- Chien de garde logiciel
- Adresse MAC unique
- Configuration du réseau via des trames Ethernet
- Tension d'alimentation : 5,0 V
- Gamme de température : -40 °C à 85 °C
- Conforme RoHS
- Contrôleur Ethernet avec interface RMII vers le PHY
- Power on Ethernet (POE) (en option)
- Horloge système à 50 MHz
- 1 capteur de température I2C
- 2 relais (un sur la carte)
- 7 lignes d'entrées/sorties à usage général
- 4 convertisseurs analogique/numérique, 16 bits
- 1 convertisseur numérique/analogique, 12 bits
- 1 bus I²C (Inter-Integrated Circuit)
- Capteur de température externe (avec câble de connexion)
- Connecteur d'extension
- Auto-configurable
- Livré pré-assemblée et testée via la boutique Elektor
- Un composant clé pour gagner le prix de \$25K (www.elektor.com/e-lock)

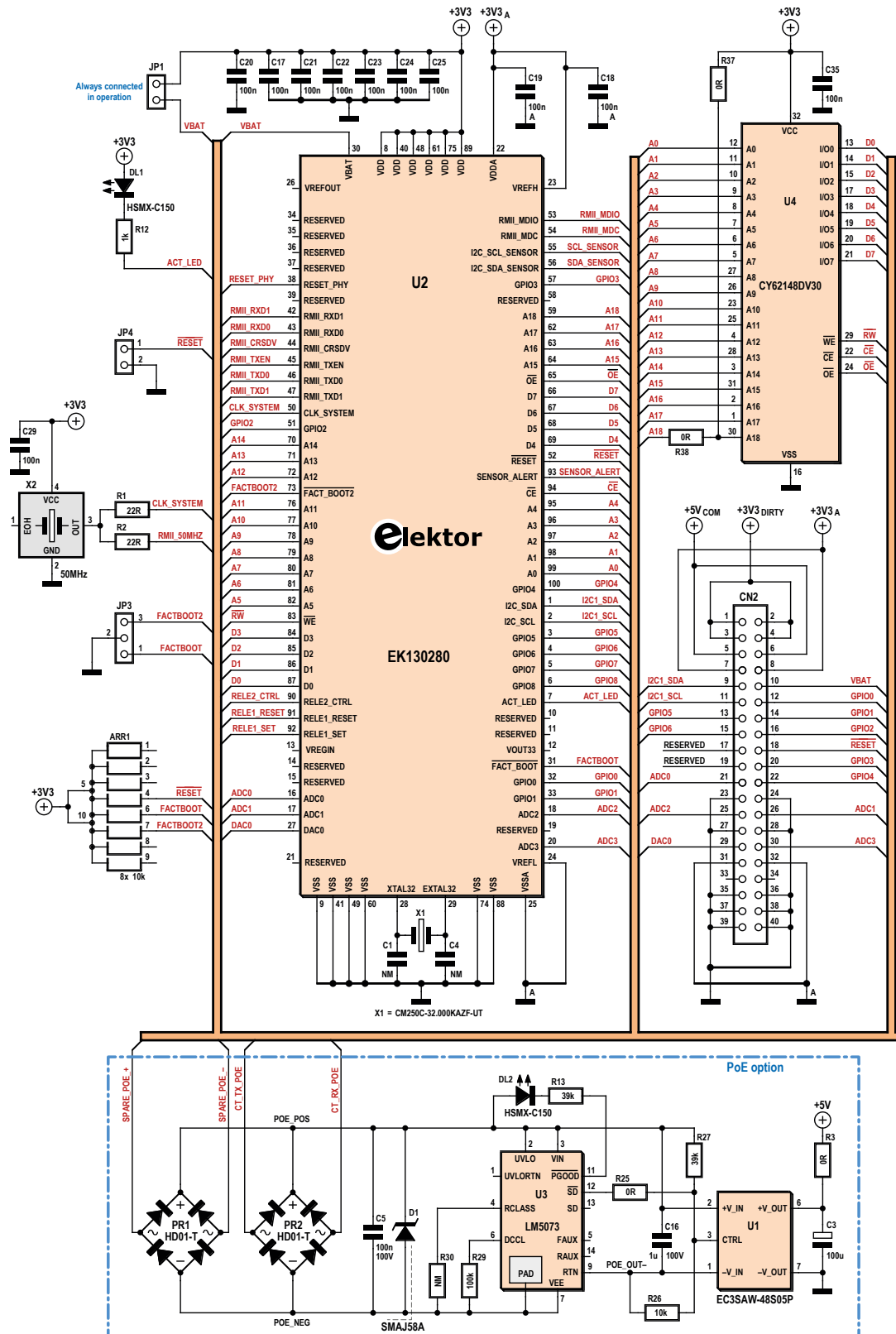
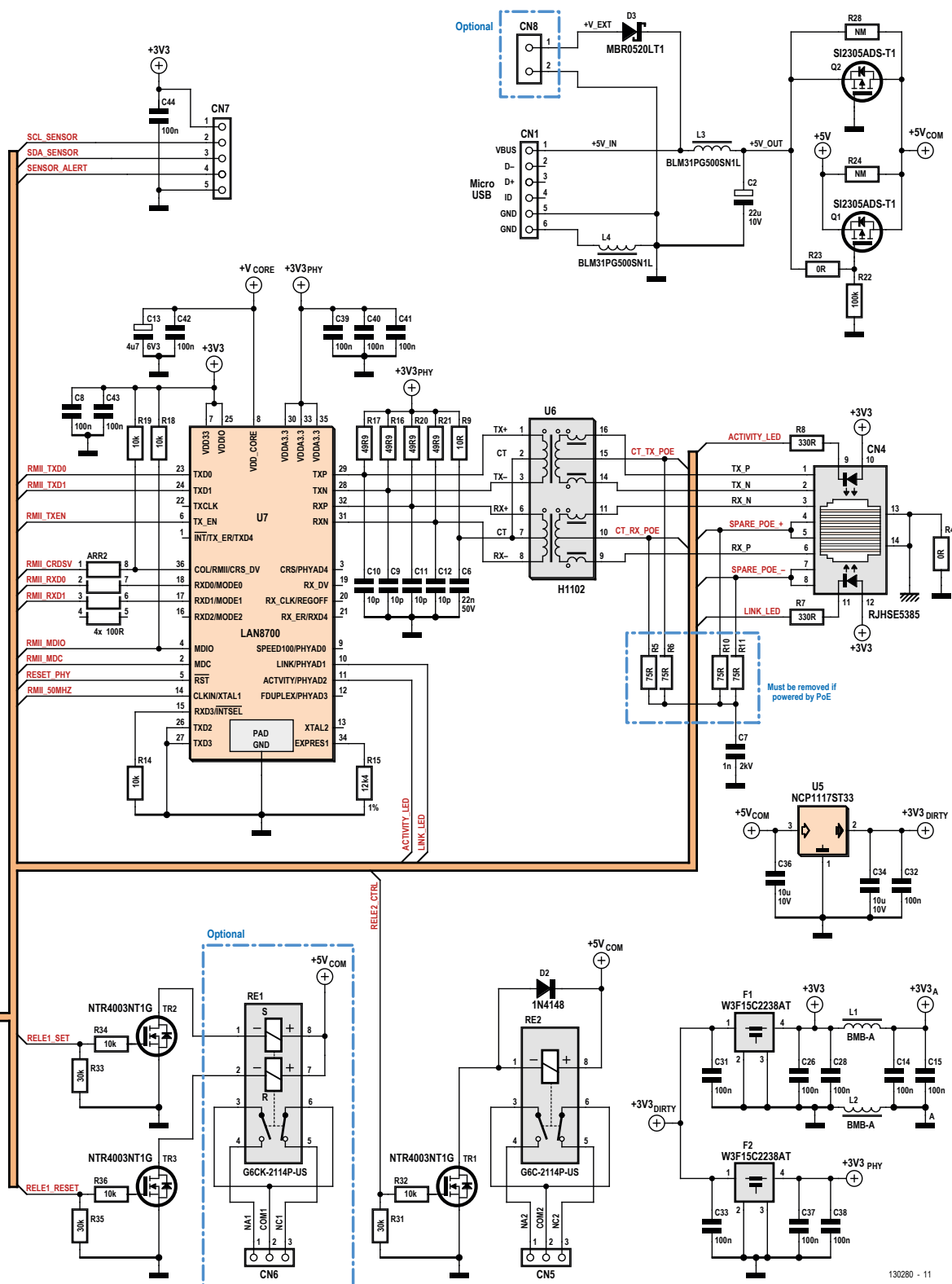


Figure 3.
Schéma complet de la
carte d'évaluation e-lock.

sécurisé, E-Lock est la solution.



130280 - 11

Les commandes de base prises en charge par l'e-lock pour la gestion des périphériques intégrés à la carte sont :

- Configurer les capteurs
- Configurer les relais
- Lire la configuration des capteurs
- Lire la valeur des capteurs
- Lire l'état des relais
- Changer l'état des relais

Les commandes permettant d'agir sur les périphériques reliés aux connecteurs d'extension sont :

- Configurer : initialiser les GPIO, voies du CAN ou du CNA ou encore les périphériques I²C
- Lire la configuration : pour connaître la configuration actuelle
- Lire : l'état des GPIO, une valeur depuis le CAN ou le CNA ou encore sur le bus I²C
- Écrire : l'état des GPIO, du CNA ou sur le bus I²C

Les commandes système supplémentaires permettent de :

- Configurer le système
- Lire la version
- Changer le certificat
- Changer la clé privée
- Changer le certificat de l'autorité de certification
- Envoyer une notification par email
- Télécharger un nouveau micrologiciel

Alimentation — la carte nécessite du 5 V pour fonctionner. Cette tension est fournie par un adaptateur secteur tout comme avec les smartphones via le connecteur micro USB CN1 ou en reliant une alimentation stabilisée directement à CN8 ; respectez la polarité indiquée par la sérigraphie dans ce dernier cas.

Ethernet — la carte se relie à un réseau Ethernet 10/100 Base-T avec un connecteur RJ45 standard doté des habituelles LED indiquant l'établissement du lien et l'activité. Entre le connecteur et la puce e-lock se trouvent les habituels transformateurs et puce PHY (*Physical Layer Transceiver*).

PoE — La carte est compatible avec la norme *Power over Ethernet* (PoE), bien que la version

actuellement disponible dans la boutique Elektor ne comprenne pas les convertisseurs C-C 48 vers 5 V.

Circuit de sortie — La carte possède deux circuits de sortie permettant de commander deux relais : RE1, un modèle double bobine à verrouillage et RE2, un modèle classique. Tous deux sont montés en interrupteur SPDT (*single-pole, double-throw*) dont les contacts sont disponibles sur les borniers CN6 et CN5 (commun au centre) respectivement.

Sur la version disponible dans la boutique Elektor une seule des sections est montée : RE2 et son connecteur CN5.

Capteur de température — Le capteur de température I²C a été déporté de la carte principale afin d'éviter que la chaleur dissipée par les composants qu'y s'y trouvent n'affecte les mesures. Comme le TMP275 est un composant CMS, il est livré monté sur un petit C.I. qui sera relié au connecteur CN7 de la carte principale avec un câble d'environ 20 cm.

Connecteur d'extension — Le but que nous nous sommes fixé dès le début de la conception est que cette carte soit plus qu'une démonstration et qu'elle puisse servir à nos lecteurs, à vous, de plateforme de développement pour vos projets IdO. C'est pourquoi les broches inutilisées de l'e-lock ont été routées vers le connecteur d'extension CN2, mettant à votre disposition les alimentations +3,3 V et +5 V ainsi qu'un bus I²C supplémentaire, quatre voies CAN, un CNA, sept GPIO et le signal de remise à zéro.

Vous trouverez le schéma complet, assez touffu, en **figure 3**. Le schéma est reproduit ici non pas pour que en faire un poster à en faire baver vos invités mais pour les lecteurs intéressés par la sécurité sur l'internet et le prix de \$25K à gagner sur www.elektor.com/e-lock.

Installation et configuration

Avant d'alimenter la carte d'évaluation e-lock, téléchargez le logiciel de démo pour PC, disponible sur la page du projet [2] et décompressez-le dans un dossier de votre disque dur.

Vous trouverez dans le dossier *ISLElektor* tous les fichiers nécessaires pour configurer et tester votre carte. Le dossier *doc* contient le fichier *network-configuration.pdf* dans lequel est décrit

le processus de configuration de l'e-lock et *ISLElektor_130280-applicationnote.pdf* qui décrit l'application de démonstration. Il est chaudement recommandé de lire ces documents avant de commencer à utiliser la carte.

Avant de pouvoir tester la carte, il faudra la configurer pour votre réseau : adresse IP du routeur, du DNS et de la carte.

Après ces réglages, reliez l'alimentation et un câble Ethernet à la carte puis lancez l'application de configuration *ISLRaw.exe* ; vous devriez voir la fenêtre de la **figure 4**.

L'application est capable de configurer plusieurs cartes reliées au même LAN. La première chose à faire est de repérer les adresses MAC en cliquant sur *Scan MAC*. L'application diffusera alors un message et le tableau situé au bas de la fenêtre se remplira avec les informations renvoyées par les cartes en attente de configuration.

Une fois toutes les informations reçues, sélectionnez la carte que vous souhaitez configurer depuis la liste ; le champ *Remote MAC* se remplira automatiquement. Enfin, saisissez l'adresse IP désirée puis cliquez sur *Set Remote IP* afin de configurer l'adresse IP de la nouvelle carte. Si tout s'est bien passé, la liste sera mise à jour avec la nouvelle IP.

La nouvelle fenêtre qui apparaîtra vous permettra alors de sélectionner les paramètres de sécurité, c'est à dire : certificat, clé, certificat de l'autorité de certification, serveurs SNTP et DNS et adresse de la passerelle. Une fois tous ces paramètres correctement renseignés, cliquez sur *Send* pour les envoyer à l'e-lock. Vous pourrez créer vos propres certificats en suivant les instructions décrites dans le document de configuration mais pour tester la carte vous trouverez des certificats d'exemple et des clés dans le dossier *certs* : choisissez *ISLserver-cert.pem* pour le champ *Cert*, *ISLserver-key.pem* pour le champ *Key* et *ISLca-cert.pem* pour le champ *CA*.

Logiciel de démo

Une fois l'installation terminée, vous pourrez tester le fonctionnement de la carte e-lock en lançant l'application de démo (**fig. 5**) *ISLElektor.exe*. N'oubliez pas, avant de la lancer, de brancher le capteur de température sur la carte.

ISLElektor est l'application utilisée pour interagir avec la carte d'évaluation e-lock, en échangeant des informations via une connexion TCP/IP sécurisée. Le protocole de communication est décrit de manière exhaustive dans la feuille de

caractéristiques de la carte [2].

Une fois *ISLElektor* lancée, tous les boutons sont désactivés à part *Connect*. Tout d'abord il vous faudra saisir l'adresse IP que vous avez attribuée à la carte lors de l'étape précédente. Cliquez ensuite sur *Connect* pour établir une connexion TCP/IP avec la carte. Une fois la connexion prête, une nouvelle fenêtre apparaît (**fig. 6**), et vous devrez sélectionner les fichiers de clé et de certificat du serveur. Comme pour la carte, vous trouverez les fichiers d'exemple dans le dossier

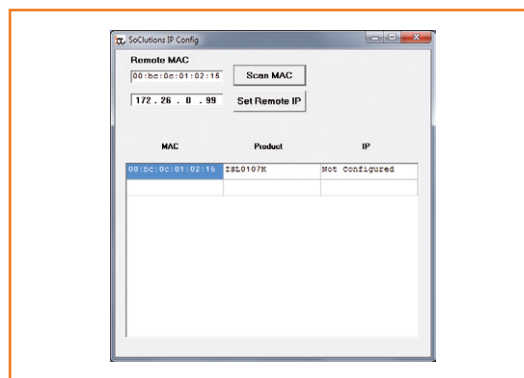


Figure 4.
Configuration de la carte d'évaluation e-lock.

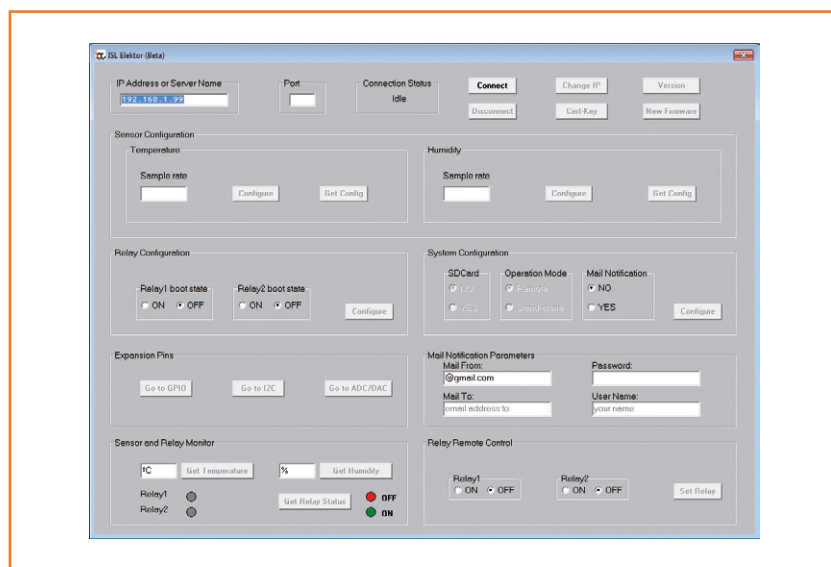


Figure 5.
Fenêtre principale de l'application de démo de la carte d'évaluation e-lock.



Figure 6.
Réglage des certificats client pour la carte d'évaluation e-lock.

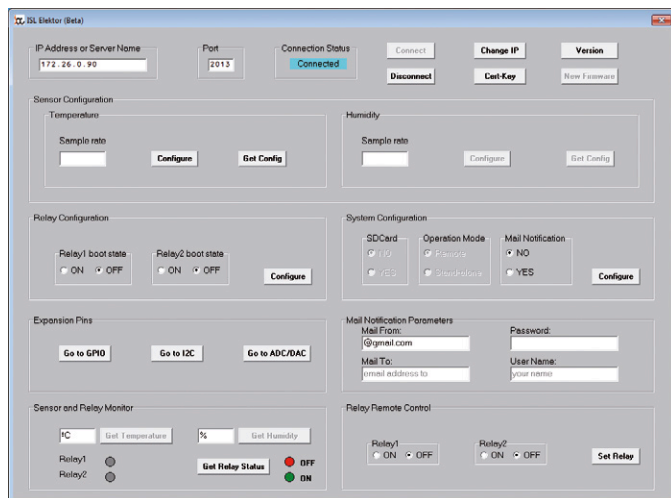


Figure 7.
Connexion réussie à la carte d'évaluation e-lock.

Le piège de la sécurité

Être en sécurité et se sentir en sécurité sont deux choses différentes. Vous, ainsi que des millions d'autres personnes utilisez l'internet pour accéder et fournir de l'information, rencontrer votre famille et vos amis ou encore gérer vos comptes et faire des achats. Pour vous tranquilliser, vous cherchez le https dans un coin de votre écran, et quand vous le trouvez, vous vous sentez en sécurité. Mais l'êtes-vous ? Très probablement non. Il existe aussi d'autres moyens d'accès à vos choses de valeurs. Beaucoup de banques ont pris d'innombrables précautions pour sécuriser leurs distributeurs automatiques et pourtant certains escrocs ont trouvé la parade : ils emportent le distributeur tout entier – adieu la sécurité.

Le décollage rapide de l'IdO ouvrira de nouvelles brèches pour ceux désireux de s'introduire dans les maisons et les systèmes. Ce que nous vous présentons ici c'est une solution de sécurisation pour vos objets connectés, par exemple la porte de votre garage, mais cela ne vous dispense pas de vérifier que la fenêtre est bien fermée !

À propos d'Intelligent Soc

IntelligentSoC a été fondée à Madrid, en Espagne en 2011 sous la forme d'une scission du département R&D de Datatech SDA, une entreprise d'ingénierie spécialisée dans la conception et le développement de systèmes de communication et de traitement de données, avec un fort accent sur la sécurité. Le but principal de l'entreprise est la conception, le développement et la fabrication d'IP, puces et modules mettant en œuvre les dernières technologies d'authentification basée sur la cryptographie elliptique (ECC), les algorithmes de Diffie-Hellman et diverses technologies de chiffrement AES 128/256.

Les produits d'IntelligentSoC sont actuellement installés là où la sécurité n'est pas une option, comme par exemple dans les systèmes aéronautiques ou militaires, et se préparent à prendre d'assaut l'Internet des Objets.

cert (*ISLclient-cert.pem* et *ISLclient-key.pem*).

Si la connexion réussit, une étiquette *Connected* apparaît dans la boîte *Connection Status* dotée d'un fond bleu, et toutes les fonctions incorporées à la carte seront alors disponibles (**fig. 7**). Dans le cas contraire, c'est *Error* qui apparaît. Dans ce dernier cas, vérifiez l'adresse IP, le câble Ethernet et les clés et certificats puis réessayez. Si vous êtes sûr que tous les paramètres sont corrects mais que les erreurs persistent, utilisez le cavalier JP3 (placez-le puis retirez-le) pour restaurer les paramètres d'usine ; vous pourrez alors recommencer le processus de configuration. Cliquez sur *Disconnect* pour mettre fin à la session ou modifier l'adresse IP ; il faudra alors cliquer sur *Change IP* pour redémarrer la carte avec la nouvelle adresse. Dans ce dernier cas, il est obligatoire de vous reconnecter en utilisant la nouvelle adresse IP.

Vous pouvez maintenant tester le bon fonctionnement de tous les éléments de la carte, changer leur configuration par défaut et jouer avec le connecteur d'extension. Vous trouverez toutes les informations nécessaires sur les possibilités du logiciel de démonstration dans le document *ISLElektor_130280-applicationnote.pdf* du dossier *doc*.

Le dossier *source* contient le code source de cette application que nous vous encourageons à modifier ou à utiliser comme base pour vos propres applications. Nous avons utilisé Borland C++ Builder6.

Aller plus loin...

Nous n'avons dans cet article que gratté la surface et la place ne nous permet pas de vous en dire plus sur ce qu'e-lock a à offrir. Je vous conseille de télécharger tous les documents disponibles sur la page du projet [2]. Vous y trouverez les fichiers de CAO du C.I., la feuille de caractéristiques de la carte e-lock et tous les logiciels. La lecture de ces documents ne vous permettra non seulement de comprendre tout ce que cette première puce Elektor a à vous offrir mais vous mettra sur la voie de la récompense de \$25.000 [3].

(130280 – version française -- Kévin PETIT)

Liens

- [1] www.solutions.com
- [2] www.elektor.fr/130280
- [3] www.elektor.com/e-lock

WIZnet

CONNECT THE MAGIC 2014 DESIGN CHALLENGE

DÉBUT : 3 MARS / FIN : 3 AOÛT

votre mission?

la révolution de l'IdO* : concevoir un système novateur à accès Internet avec le module **Ethernet WIZ550io** ou la puce **W5500** de Wiznet !

* IdO = internet des objets B

15 000 \$ de prix en espèces sont en jeu, ainsi que la possibilité de donner une portée internationale à votre projet grâce à WIZnet, Circuit Cellar et Elektor !

WIZnet offre 2000 modules WIZ550io à toute personne éligible au concours. Les demandes seront traitées dans l'ordre de leur arrivée et jusqu'à épuisement des stocks. Le concours démarre le 3 mars 2014 à 12 h (HNE) et finit le 3 août 2014 à 12 h (HNE). Procurez-vous un module WIZ550io ou une puce Ethernet intelligente W5500, participez, enrichissez le monde et enrichissez-vous !

www.circuitcellar.com/wiznet2014

Le module WIZ550io de WIZnet est une solution Ethernet complète qui comprend la puce Ethernet intelligente W5500 (Ethernet 10/100 MAC et PHY, matériel TCP/IP, tampon de RAM de 32 Ko) et une interface réseau (transformateur et prise RJ45) avec adresse IP par défaut et adresse MAC.

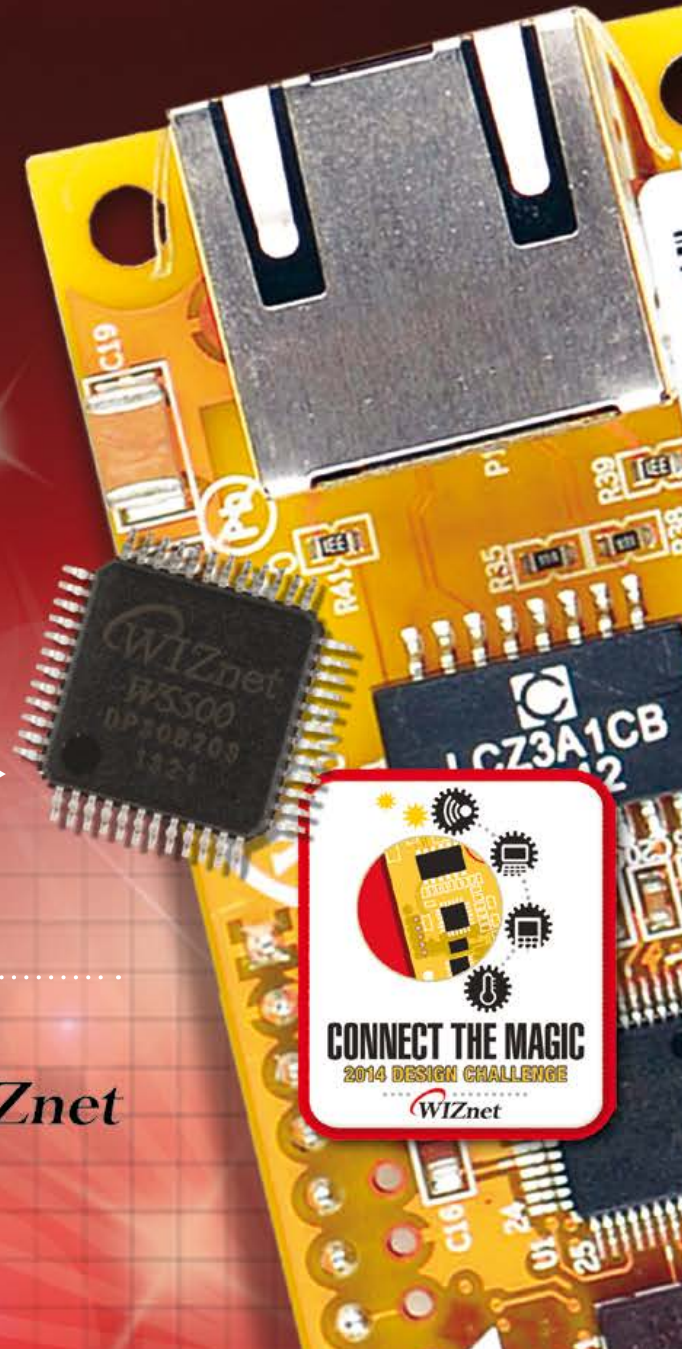
Assistance technique : wizwiki.net/forum

Achat : shopwiznet.com ou shop.wiznet.eu

POWERED BY



Pour rester informé,
abonnez-vous à CC.Post !



afficheurs Nixie

drôle de composant n°4

Neil Gruending
(Canada)

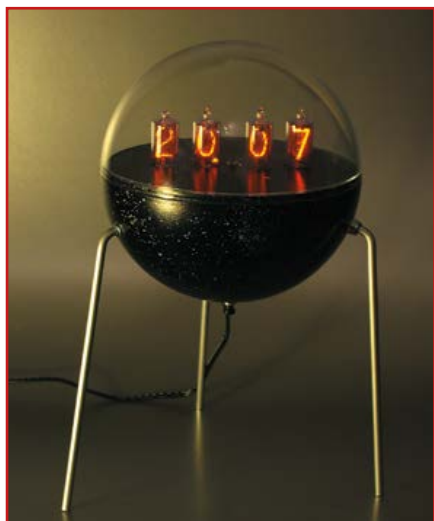


Figure 1.
L'horloge Sputnik Elektor
publiée en, euh,
1957 + 50 = 2007.

Figure 2.
Afficheur Nixie IN-12A
illuminé.



Nixie, la marque déposée des *tubes d'affichage au néon et à cathode froide*, rime avec nostalgie. D'ailleurs ils datent de l'époque des tubes.

Longtemps il n'y avait aucun autre moyen (électronique) d'afficher des caractères (à peu près) alphanumériques. Chassés un temps par les afficheurs à LED à sept segments, ils ont fait un retour en beauté. Ils sont beaux, oui, mais vénéneux, car ils fonctionnent sous environ 170 V_{CC}, une tension potentiellement létale. Gare aux règles de sécurité [1] !

Pour commander les sept segments de LED d'un bel afficheur moderne, commode et compact, il faut un décodeur extérieur (matériel ou logiciel) pour convertir un caractère en sa représentation

avec sept segments. Aujourd'hui c'est facile, mais pas dans les années 50 ou 60. Or les afficheurs Nixie n'ont pas ce problème puisque chaque chiffre possède sa propre broche de commande. Un Nixie numérique, par exemple, en possède neuf, une pour chacun des chiffres de 0 à 9.

Un afficheur Nixie est constitué d'un empilement d'éléments. Celui du dessus est l'anode, à laquelle on relie le positif. Au dessous se trouvent autant de plaques que de caractères, reliées à leur ligne de commande individuelle. C'est cet agencement particulier qui donne aux caractères des Nixie leur forme nostalgique et leur agréable lueur. Les Nixie fonctionnent selon le principe de la lampe au néon. On applique entre l'anode et une cathode une tension continue supérieure à la tension d'amorçage du néon et le caractère sélectionné s'illumine. Le truc pour ne pas les casser c'est d'utiliser une alimen-

tation limitée en courant ou d'introduire une résistance en série dans le circuit : un tube Nixie se comporte comme une zener lorsqu'il commence à conduire. La tension requise pour démarrer la conduction est assez élevée (habituellement supérieure à 150 V) ce qui rend délicat l'interfaçage avec les circuits logiques modernes. Vous en trouverez toutefois de nombreux exemples sur l'internet et dans Elektor comme *l'horloge Sputnik* (**fig. 1**) [2] et divers thermomètres [3],[4].

L'intensité du courant d'anode est critique pour les Nixie. Trop forte ou trop faible, elle peut réduire dramatiquement leur durée de vie. Elle agit sur la luminosité et la netteté des chiffres. La longévité d'un Nixie piloté correctement atteint 200.000 h, soit plus de 20 ans.

La **figure 2** montre un afficheur Nixie IN-12A russe illuminé, courant dans les horloges et encore utilisé dans certaines centrales nucléaires. Il est alimenté sous 170 V_{DC} avec une résistance de 40 kΩ sur l'anode ; sa broche correspondant au chiffre sept est à la masse. Vous pouvez voir la grille d'anode au dessus et, par transparence, les chiffres éteints devant le sept.

Pour le plus grand bonheur de la communauté grandissante d'amateurs d'horloges rétro, on trouve encore facilement en ligne des Nixie neufs (*New from Old Stock* ou NOS). C'est épatant.

(130463 – version française : Kévin PETIT)

Liens

- [1] www.repairfaq.org/REPAIR/F_safety.html
- [2] Sputnik donne l'heure, Elektor janvier 2007
www.elektor-magazine.fr/050018
- [3] Thermomètre à tubes Nixie, Elektor janvier 2011
www.elektor-magazine.fr/090784
- [4] Thermomètre hygromètre à tubes Nixie, Elektor juin 2012,
www.elektor-magazine.fr/110321

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

qu'est-ce que c'est que ce con... densateur ?

Thijs Beckers
(elektor.labs)



Dans l'esprit de nos lecteurs, le nom d'Elektor est associé à une pratique soutenue de l'électronique. Pour que celle-ci leur procure les plus grandes satisfactions et des réalisations réussies, nous avons élaboré un service de fourniture de circuits imprimés, de microcontrôleurs programmés, de kits et de modules prêts à l'emploi. Il s'agit d'augmenter par ce biais vos chances de succès lors de l'assemblage de vos propres circuits, chez vous, dans votre atelier ou dans le cadre de votre travail.

Mon rêve serait de pouvoir voler à votre secours chaque fois que c'est nécessaire. Faute de pouvoir réaliser cette utopie, je m'attache à vérifier soigneusement le contenu des semi-kits que nous proposons. Et notamment leur conformité avec la nomenclature publiée.

C'est ainsi qu'en vérifiant récemment, au hasard, un semi-kit de la *carte DSP audio universelle à ADAU1701* [1], je suis tombé sur les très étranges condensateurs de 100 nF (0,1 pF) ci-dessus. Jusqu'ici je pensais n'être pas un crétin, mais avec ces condensateurs le doute s'est insinué. Comment un condo de 100 nF peut-il être marqué « 105 » d'un côté et « 104 » de l'autre ? Un message subliminal... un indice de plus de la

présence parmi nous des extra-terrestres ? Un poisson d'avril ? Une simple erreur ?

Je l'ignore, en tout cas les 100 nF de cette série de kits que j'ai vérifiés portaient tous la même double inscription contradictoire : 104 d'un côté et 105 de l'autre.

En vain j'ai cherché une explication possible sur l'internet et sondé mes collègues qui en ont pourtant vu des vertes et des pas mûres. Selon le brillant LCR-mètre de Jean-Jacques Aubry publié en mars, avril et mai 2013 [2]), ce sont bien des 100 nF. Les kits de la carte audio DSP universelle qui en sont équipés marcheront, mais le mystère demeure.

Je serais curieux de savoir si vous, lecteurs, avez déjà rencontré une telle bizarrerie ou si vous avez une explication.

Une seule adresse : redaction@elektor.fr. Et en attendant, faites comme moi, désormais, quand vous lirez la valeur marquée sur vos condensateurs : vérifiez ce qui est écrit de l'autre côté !

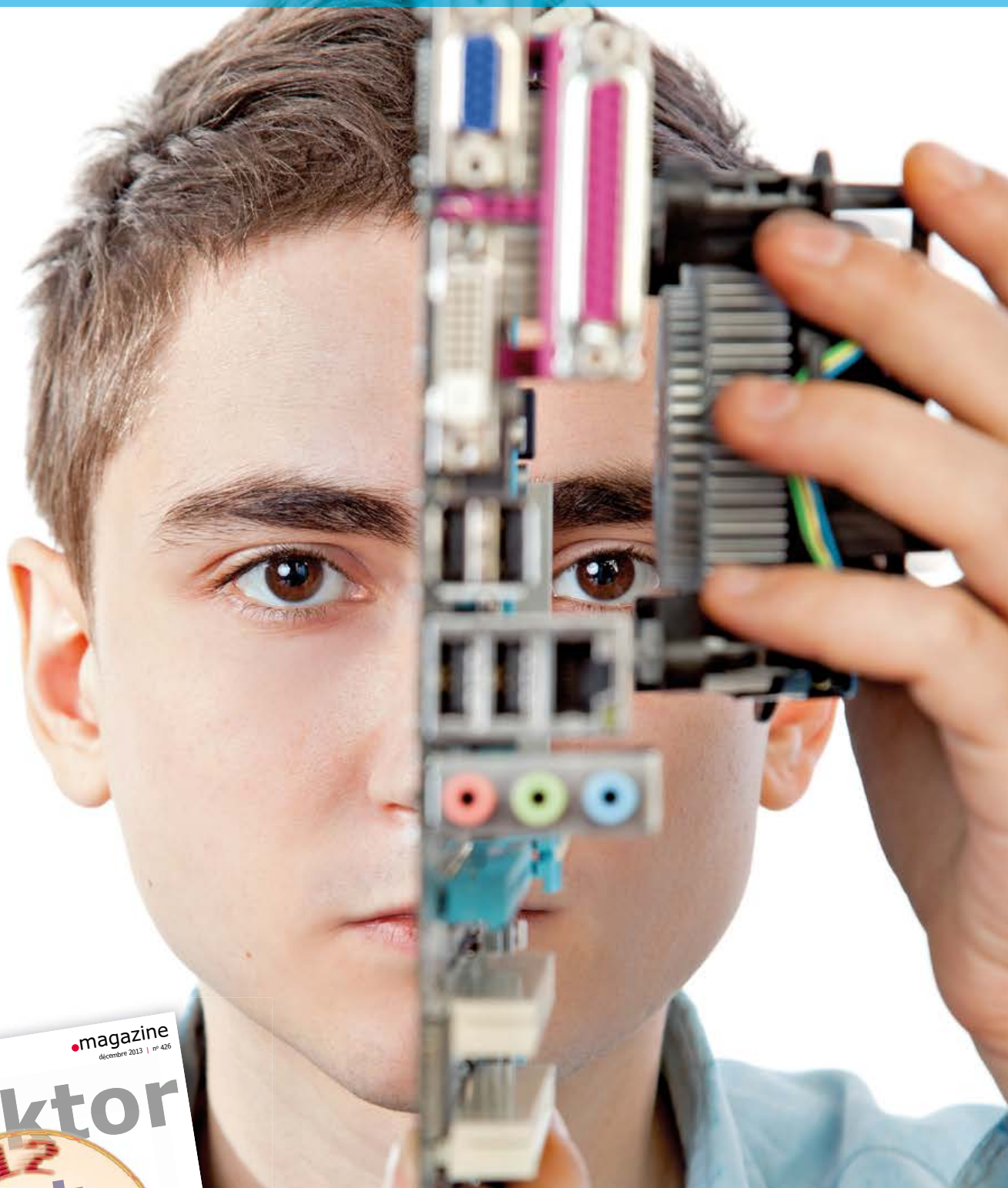
(130460)

Liens

[1] www.elektor-magazine.fr/130232

[2] www.elektor-magazine.fr/130093

le XXI^e siècle vient seulement de commencer



explorez l'électronique,
lisez **elektor**

www.elektor.fr/abo

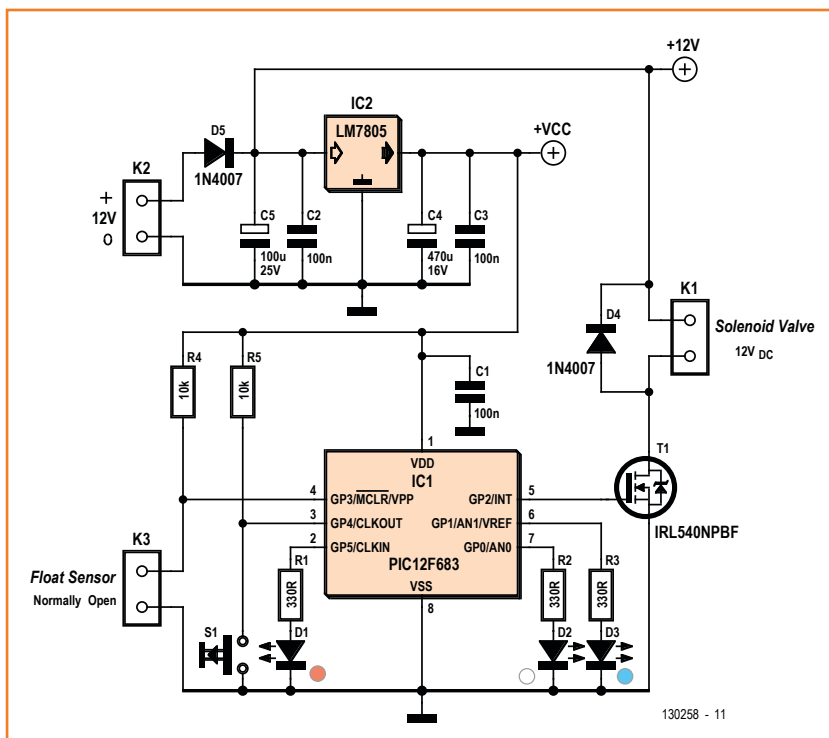
électrovanne mise au régime



sous osmose inverse

Maarten Vandekeybus (Belgique)

Il est toujours plus confortable de faire actionner un robinet par un automate que de devoir surveiller la manœuvre soi-même, surtout quand elle dure des heures. Le système permet aussi d'éviter des débordements intempestifs. En outre, le schéma proposé ici permet des économies d'énergie substantielles.



À la maison, j'ai un filtre d'eau par osmose inverse pour purifier l'eau courante et alimenter en eau potable d'excellente qualité un réservoir de 25 litres. Pour le remplir, je devais régulièrement ouvrir un robinet, mais comme l'opération complète dure quatre heures, il m'arrivait d'oublier d'aller le fermer à temps, ce qui provoquait une grosse perte d'eau.

Pour mettre fin à ces désagréments, j'ai donc construit un petit circuit pour automatiser le processus. Il ouvre une électrovanne jusqu'à ce que l'interrupteur d'un flotteur lui signale que le réservoir est plein.

J'ai commandé sur eBay une telle vanne de 12 V_{CC} pour 12 €. Elle fonctionne très bien, mais consomme beaucoup : 18 W pendant toute l'opé-

Figure 1.

Il ne faut guère plus qu'un PIC12F683 pour constituer l'électronique de l'électrovanne mise au régime.

ration. Je me suis souvenu d'un article d'Elektor qui expliquait comment réduire la consommation d'un relais électromagnétique, en se basant sur le fait qu'une fraction de sa puissance nominale suffisait à le maintenir activé après enclenchement. Après quelques essais, j'ai opté pour un circuit à microcontrôleur qui commande la vanne par l'intermédiaire d'un transistor MOSFET de puissance. Il s'occupe d'ouvrir la vanne à pleine tension, qu'il maintient pendant un moment avant de réduire la tension juste pour que la vanne reste activée, mais à plus basse consommation. Il y a en outre une temporisation d'une minute lors de l'arrêt pour éviter des commutations répétées dans un sens et dans l'autre à cause d'oscillations du flotteur. J'ai mis trois LED de couleur pour manifester les différents états du circuit.

Le schéma

Le schéma du circuit, revu et approuvé par le laboratoire Elektor, est à la **figure 1**. Le μC

utilisé est un PIC12F683 et c'est sa sortie GP2 qui pilote le MOSFET de puissance T1, lequel commande la vanne. La baisse de puissance pour l'électrovanne après l'activation s'opère par modulation de la largeur d'impulsion (MLI) sur une fréquence voisine de 488 Hz. Le rapport cyclique est alors de 30 / 256 (impulsion / pause). Mes essais ont montré que la puissance nécessaire pour *maintenir la vanne ouverte* tombait à 0,4 W après l'activation. Fameuse différence par rapport aux 18 W qui précèdent !

Les trois LED révèlent l'état actuel du système. La blanche, D2, indique qu'il est en mode automatique.

La bleue, D3, signale que la vanne est ouverte, tandis que D1, la rouge, montre qu'on est en attente : l'interrupteur du flotteur a basculé, mais la vanne n'a pas encore réagi. Les LED sont également pilotées en MLI par souci d'économie d'énergie.

Publicité

CadSoft presents


PCBSim

EAGLE V6

the new simulation tool for signal integrity tests*:

- Signal integrity testing before the layout to avoid redesign
- Integration in EAGLE and intuitive operation
- Accurate simulation results
- Support directly from manufacturer

More information:



*developed by Felicitas Customized Engineering

25 years CadSoft



www.cadsoftusa.com

Liste des composants

Résistances :

R1, R2, R3 = 330 Ω
R4, R5 = 10 k Ω

Condensateurs :

C1, C2, C3 = 100 nF
C4 = 470 μ F/16 V radial
C5 = 100 μ F/25 V radial

Semi-conducteurs :

D1 = LED rouge 3 mm
D2 = LED blanche 3 mm
D3 = LED bleue 3 mm
D4, D5 = 1N4007
T1 = IRL540
IC1 = PIC12F683 (programmé, 130258-41)
IC2 = 7805

Divers :

K1, K2, K3 = bornier à 2 vis, au pas de 5 mm
S1 = bouton-poussoir
circuit imprimé 130258-1 [1]

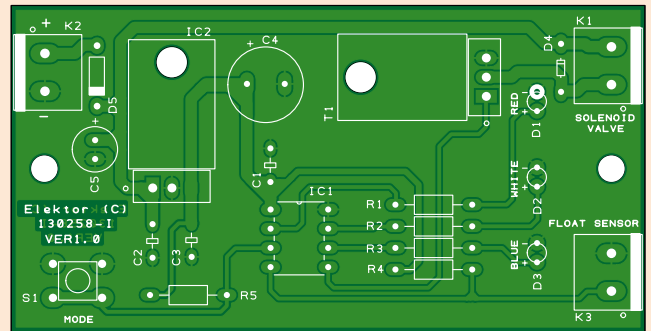


Figure 2.
Les composants sont à l'aise sur ce circuit imprimé, même avec peu d'expérience du soudage, on en vient à bout sans peine.

Le bouton-poussoir S1 commute entre trois modes de commande : arrêt, manuel, automatique.

- À l'arrêt, le circuit est inactif et toutes les LED sont éteintes.
- En manuel, la vanne est ouverte en permanence. Les LED blanche et bleue sont allumées.
- En mode automatique, la vanne s'ouvre et se ferme en fonction de la position de l'interrupteur du flotteur. Celui-ci est toujours ouvert au repos et se ferme quand l'eau monte au-dessus du niveau fixé par le flotteur. Pour éviter des soubresauts dus aux vagues à la surface de l'eau, la vanne réagit avec un retard d'une minute, indiqué par la LED rouge.

Lors de la disparition de la tension d'alimentation, manuelle ou en cas de panne de courant, l'état en vigueur du circuit est mémorisé dans

l'EEPROM interne, ce qui permet, dès le retour de la tension d'alimentation, de rétablir la situation d'avant l'interruption.

Pour l'alimentation, le plus simple est d'utiliser un adaptateur secteur capable de délivrer environ 1,5 A sous une tension continue de 12 V. Le stabilisateur IC2 s'en sert pour alimenter le PIC sous 5 V. La diode D5 protège d'une inversion de polarité.

Un circuit imprimé simple

Comme le circuit imprimé de la **figure 2** n'est pas surchargé et ne contient que des composants standard, sa construction ne donne pas de souci. Pensez à programmer le PIC avant de l'installer, vous pouvez télécharger les codes source et hexadécimal sur le site [1] d'Elektor, mais un exemplaire préprogrammé (130258-41) est disponible sur commande.

Il est recommandé d'enfermer tout le circuit dans un boîtier résistant aux éclaboussures, comme on le voit en tête de l'article. Vous en trouverez dans les magasins de bricolage.

(130258 – version française : Robert Grignard)

Lien

[1] www.elektor-magazine.fr/130258

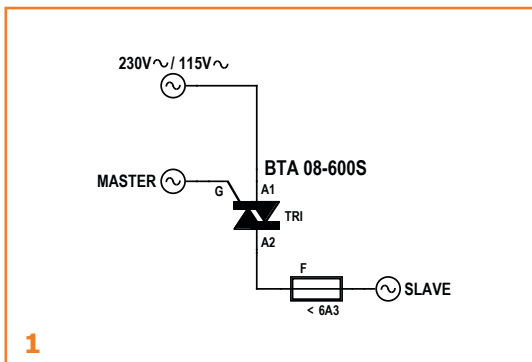
Figure 3.
L'électrovanne et l'interrupteur à flotteur que j'ai utilisés.



multiprise asservie par un seul triac

Les montages à un seul composant (analogique) sont rares. Celui-ci permet de se confectionner une rallonge asservie à l'aide d'un seul triac (balaise) !

La tension alternative de 230 V du réseau électrique règne sur l'anode A1, le maître (MASTER) règne sur la gâchette G (*gate* en anglais) tandis que l'esclave (SLAVE) est soumis à l'anode A2



du triac (**fig. 1**). Dès que l'appareil *maître* est mis sous tension, il circule de A1 vers G un courant qui amorce le triac : aussitôt, l'esclave relié à A2 est mis sous tension et le restera tant que le maître sera actif. Quand celui-ci est éteint, le courant à travers la gâchette disparaît et l'esclave s'éteint aussi. C'est tout !

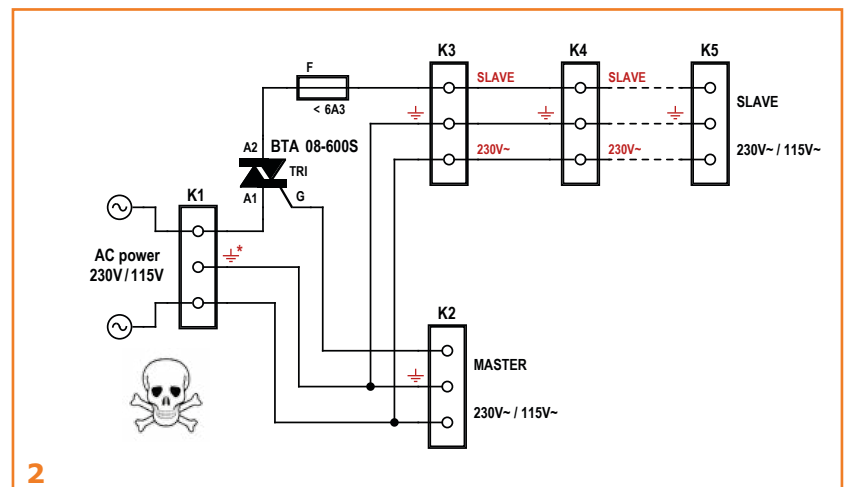
Il est possible d'invertir la gâchette et A1, de sorte que les 230 V~ règnent sur G et qu'A1 devienne l'entrée maître. L'effet est le même. La **figure 2** montre comment connecter le triac à une multiprise électrique dont trois des prises deviennent des esclaves, la quatrième sera le maître. On peut souder le triac sur un morceau de plaque perforée munie d'un bornier à vis. Il faut commencer par séparer dans la multiprise les connexions de la prise maître de celles des prises esclaves, pour insérer le circuit à triac.

Il est recommandé de protéger les sorties esclaves avec un fusible T (retardé). Comme il règne sur toutes les connexions de ce circuit un potentiel mortel et qu'il n'y a aucune séparation galvanique, il faut impérativement le monter de telle façon qu'aucun contact accidentel ne soit possible. La ligne de mise à la terre* doit rester connectée sans interruption à toutes les prises. C'est la consommation totale du maître et des esclaves qui dicte le type de triac à utiliser. Le modèle BTA08-6008 (600 V/8 A) présente l'avantage d'un courant et d'une tension d'amorçage très faibles. Étudiez bien la feuille de caractéristiques de celui que vous utiliserez. On ne badine pas avec le secteur.

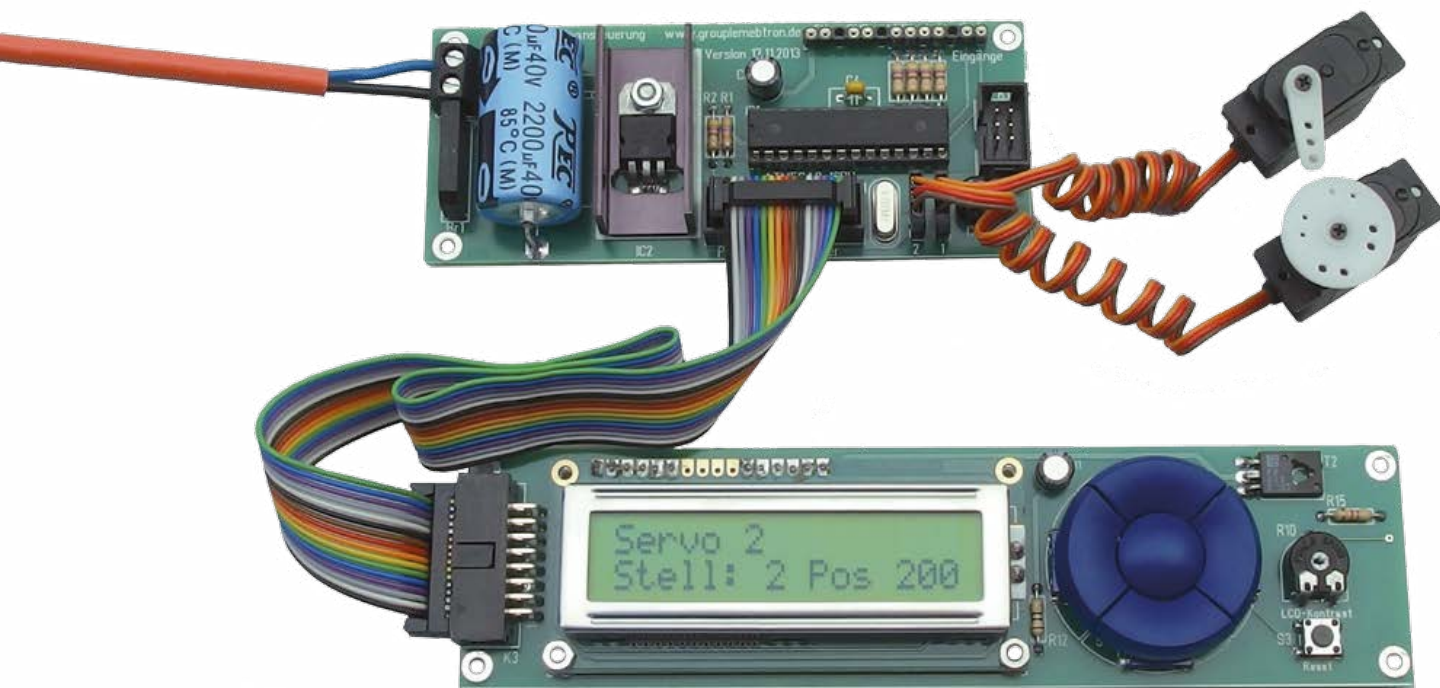
130419

Note du labo d'Elektor

À notre connaissance, cette astuce est inédite. On doit se demander toutefois si un triac est conçu pour une telle application, sachant que l'étendue de la surface de sa gâchette sur la puce est réduite et ne supporte pas de courant de forte intensité. Malheureusement, les fabricants de triacs n'indiquent que rarement l'intensité maximale du courant de gâchette. Sur un modèle, nous avons trouvé par exemple 500 mA. Il faut donc vous assurer que la consommation de votre maître ne dépasse pas cette limite.



cerveau de servos pour train miniature avec console libre



Peter Grundmann
(Allemagne)

Ce système permet de commander des servomoteurs pour un circuit de train miniature non numérique. Ils actionneront les barrières de passage à niveau, les aiguillages ou les signaux. Le petit circuit imprimé met en action deux servos et quatre LED. La programmation s'effectue par deux contacts sur la carte ou à l'aide d'une console de pilotage et de programmation séparée.

Pour mettre en action n'importe quelle partie d'un chemin de fer électrique miniature, en particulier pour les aiguillages et les signaux de sémaphore, on utilise généralement des servomoteurs conçus pour le modélisme aérien, automobile ou marin. On en trouve actuellement à peu de frais, disponibles en différentes tailles et classes de puissance. Ces servos réclament une commande précise, par des impulsions calibrées entre 1 et 2 ms de largeur. Raison pour laquelle il y a aussi des testeurs de servomoteurs. Ceux-ci

permettent de régler les paramètres souhaités au moyen d'un potentiomètre, jusqu'à ce que le servo réponde exactement. Employer ce genre de matériel sur un circuit de train électrique est tout sauf idéal. Aux modélistes adeptes du numérique, le marché offre un vaste choix de produits appropriés, mais rien pour les adeptes de l'analogique qui se demandent à quel servo se vouer. Elektor va ici au-devant de leurs besoins. Le modéliste ferroviaire attend de ses servos qu'ils assurent un mouvement de va-et-vient

(surtout pas trop rapide !) entre les positions possibles d'un dispositif mécanique (une barrière de passage à niveau, p. ex.).

Le circuit

Le schéma (**fig. 1**) donne la composition du cerveau pour deux servos. Centré sur un ATmega8, programmé en conséquence, il commence dès la mise sous tension par l'interrogation des entrées et l'envoi des impulsions aux servos. D'autre part, un petit circuit imprimé (**fig. 2**), muni d'un écran LCD et d'un clavier, joue le rôle de console pour effectuer les réglages nécessaires ; une fois ces réglages faits, on déconnecte la console. Le *cerveau de servomoteurs*, on l'installe en bonne place directement sur le circuit ferroviaire.

Quand la ligne d'entrée attribuée à l'un des servos

est mise à la masse, il s'enclenche et prend une des deux positions extrêmes possibles. Quand l'entrée est et reste ouverte, il (re)prend l'autre position extrême. Deux autres sorties sont attribuées à chacun des servos, on peut y brancher des LED, p. ex. pour les feux clignotants rouges sur le passage à niveau. Les LED s'allument et s'éteignent en alternance mutuelle avec en plus un effet de gradation.

Le système connaît d'autres modes de fonctionnement. On peut ainsi programmer le cerveau pour que les deux servos prennent entre les deux positions extrêmes jusqu'à six positions intermédiaires. La commande s'opère alors par boutons-poussoirs depuis la console, pour orienter p. ex. un segment de voie sur pont tournant.

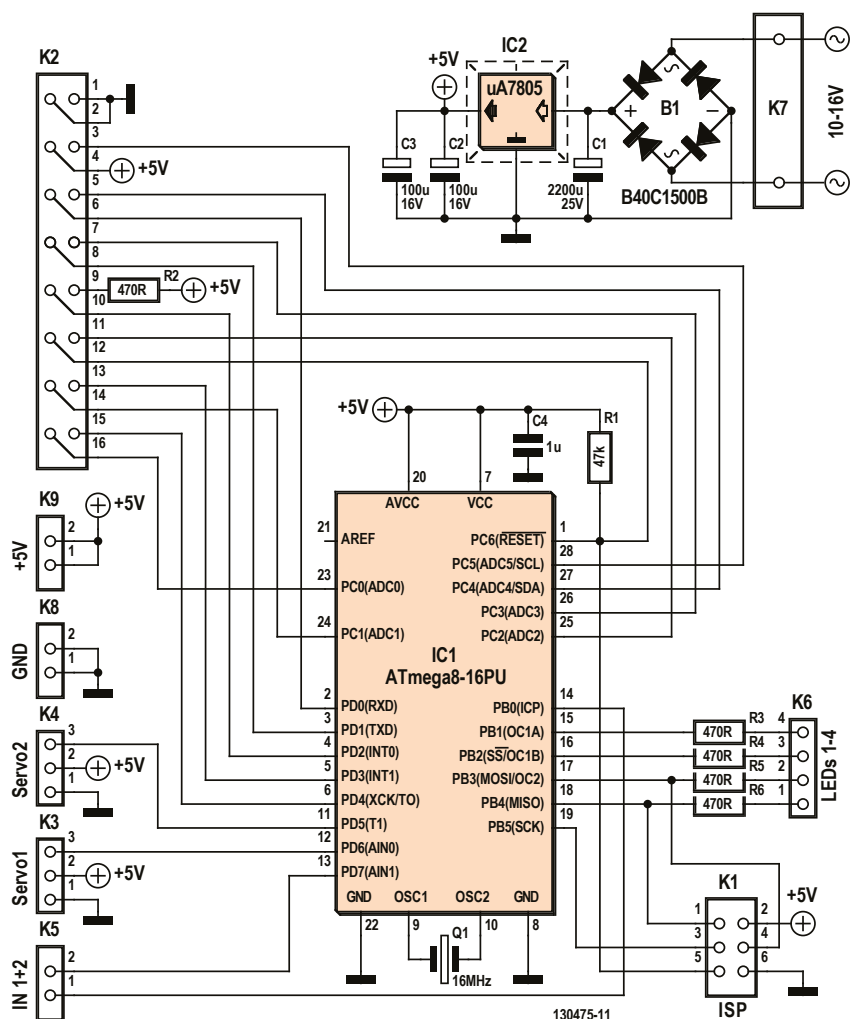


Figure 1.
Le schéma de l'unité à microcontrôleur pour deux servomoteurs et quatre LED.

Autre application, la commande d'un ascenseur ou d'une grue sur huit niveaux.

L'ensemble du projet est donc constitué du circuit à μC avec son alimentation et d'une carte avec écran et clavier comme console de commande et de programmation. Il vous faudra sans doute plusieurs exemplaires du circuit de pilotage des servos, mais vous n'aurez généralement l'usage que d'une seule console.

Le circuit sera alimenté par un transformateur qui délivre environ 1 A sous une tension entre 12 et 16 V. L'alimentation est simple : derrière le redresseur et le condensateur réservoir, un 7805 fixe à 5 V la tension de service et limite le courant aux alentours de 1,5 A. Le module fournit l'énergie aux servos, au μC et à l'afficheur, éclairage compris.

Le câblage du μC se limite à un quartz, avec ses habituels condensateurs, pour stabiliser la fréquence d'horloge et assurer ainsi un positionnement exact des servos. On trouve aussi une embase femelle d'une rangée pour le rac-

cordement de la tension d'alimentation de 5 V, les sorties pour les LED et les deux entrées de commande. Il y a aussi une embase femelle à 6 contacts pour la programmation *in situ* (ISP). Enfin, une embase d'une rangée de 16 picots permet la connexion par câble en nappe de la console de commande.

La liaison à 16 conducteurs permet de gérer l'écran de 2 lignes de 16 caractères ainsi que les boutons-poussoirs pour les réglages. Sur la carte, on trouve encore un potentiomètre pour le réglage du contraste de l'écran et un transistor qui commute l'éclairage d'arrière-plan. Cette possibilité n'est pas exploitée ici. La console peut en effet convenir à d'autres projets. La disposition circulaire des boutons n'est pas bon marché, mais elle garantit un maniement commode.

Réglage de la console et des positions des servos

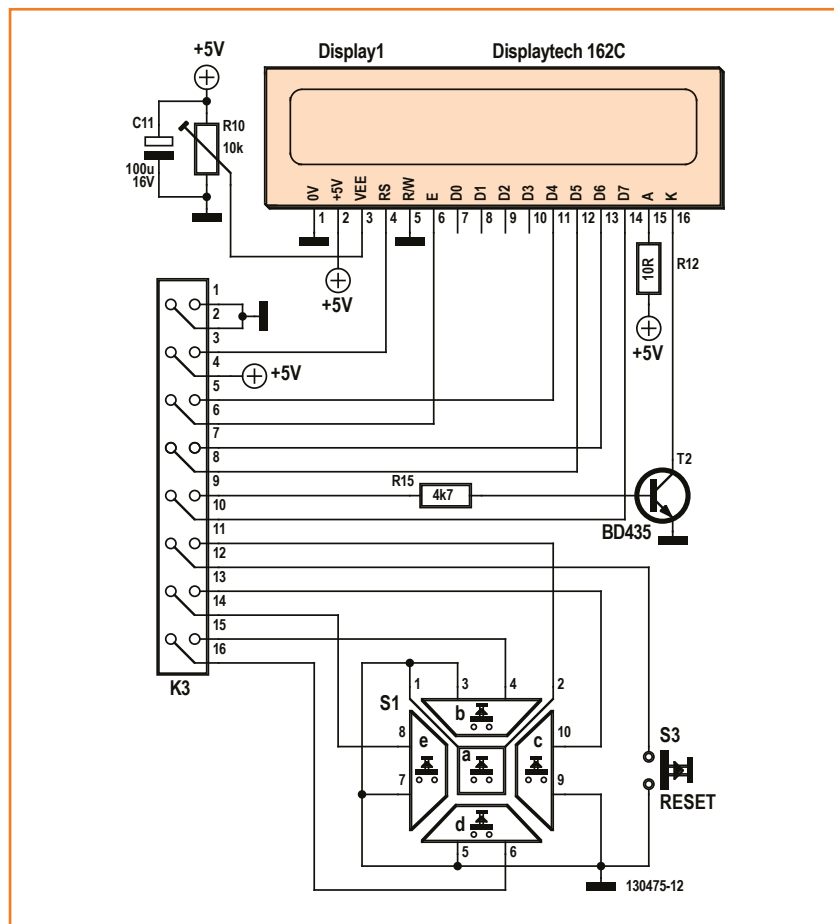
Le μC est programmé de telle sorte qu'à la mise sous tension, les deux servos se placent en fonction de l'état des entrées dans l'une des positions extrêmes. Ensuite, avec la console, vous pouvez instaurer des paramètres individuels. Lors de la mise en service, quand le message *Enter* => *PROGRAM...* apparaît, appuyez sur le bouton central qui sert de touche Entrée. Vous arrivez ainsi dans la liste des pas de programmation (fig. 6).

Raccordement et choix des servos

C'est sur les embases à 3 picots, à côté du quartz, que les servos se branchent par la prise à 3 contacts dont la plupart sont dotés, avec des câbles souvent tricolores : brun, rouge et orange, certains noir, rouge et blanc. Le fil noir ou brun, c'est le pôle négatif, du côté du bord du circuit imprimé. Le fil rouge, pôle positif, vient au milieu. Le fil orange ou blanc conduit les impulsions au moteur. Une éventuelle inversion du connecteur n'aura en principe aucune conséquence grave. Rien ne fonctionnera, il suffit de tourner le connecteur. Si vous avez un de ces connecteurs de servo munis d'un ergot latéral, celui-ci sera du côté du μC .

Le choix des servos est très ouvert du point de vue électrique, à condition qu'ils soient prévus pour une tension de service de 5 V. Malheureusement, tous ne conviennent pas à n'importe quelle application. Certains des modèles analogiques se mettent à trembler de manière incontrôlable à la mise sous tension, ce qui posera des problèmes

Figure 2.
Le schéma de la console de pilotage et de programmation.



Liste des composants

Commande de servos

Résistances :

R1 = 47 k Ω
R2, R3, R4, R5, R6 = 470 Ω

Condensateurs :

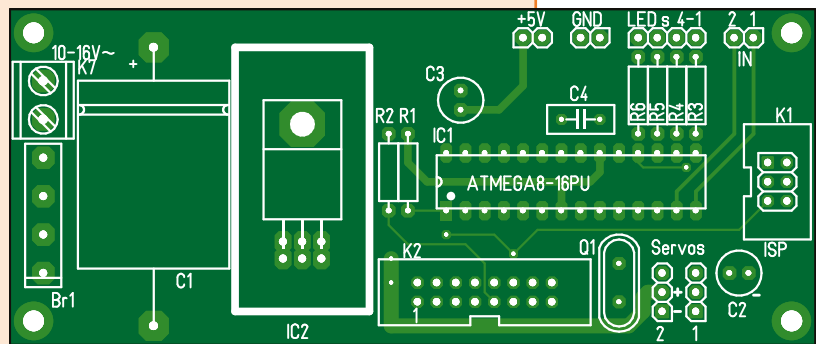
C1 = 2 200 μ F/25 V horiz.
C2, C3 = 100 μ F/16 V vert. au pas de 2,54 mm
C4 = 1 μ F au pas de 5 mm

Semi-conducteurs :

Br1 = B40C1500B au pas de 5 mm,
brochage : ~ + ~ -
IC1 = ATmega8-16PU, DIL28S (programmé :
130475-41)
IC2 = μ A7805 TO220 horiz.

Divers :

K1 = embase protégée à 6 picots au pas de 2,54 mm
K2 = embase protégée à 16 picots au pas de
2,54 mm, verticale
K3, K4 = embase à 3 picots au pas de 2,54 mm



K5, K6, K8, K9 = barrette femelle à 20 contacts
en ligne au pas de 2,54 mm
barrette à 20 picots en ligne, correspondante de K5
etc. au pas de 2,54 mm
K7 = bornier à 2 vis, au pas de 5 mm
Q1 = quartz 16 MHz HC49/U
support de CI 28 contacts, modèle bas, pour IC1
radiateur pour IC2
boulon avec rondelle M3×10 mm pour IC2
circuit imprimé 130475-1 [1]

Figure 3.
Le circuit imprimé pour
l'unité de commande des
servos.

avec une mécanique sensible. D'expérience, je peux dire que ces tremblements ne se produisent pas sur les servos numériques de prix.

Raccordement d'un clignotant par la console

Quand la console pointe dans le menu sur *input terminals*, une entrée libre de commande se manifeste : Pos: 2. Si l'on met l'entrée à la masse, c'est Pos: 1 qui apparaît.

Dans la position du menu *keys and display*, on change de servo avec les boutons *up* et *down*. Avec les boutons *left* et *right* on sélectionne une position. *Enter* confirme que le servo doit aller dans la position choisie.

Dans les deux modes, le choix de Position 1 fait démarrer le feu clignotant. Pour les deux LED commandées par un servo, on relie leurs anodes au +5 V et leurs broches négatives se répartissent sur les deux sorties de K6.

Remarque importante

Les servos habituels développent une certaine force de rappel, inutilisée le plus souvent en modélisme ferroviaire. Une mauvaise fixation du servo au montage ou un ajustement mal réglé peut entraîner des dégâts. La solution peut venir d'un ancrage élastique. Il faut aussi considérer qu'un servo essaie toujours d'atteindre la position demandée. En cas de blocage mécanique, il

consomme un fort courant qui risque de surcharger l'alimentation et aller, à la longue, jusqu'à endommager le servo. On s'en aperçoit au bourdonnement qu'il produit.

Construction de l'unité de servos

Tous les composants sont disponibles auprès des distributeurs concernés. Il faut programmer dans le contrôleur le logiciel à télécharger sur le site [1] d'Elektor. On s'épargne cette démarche en commandant à la même source un exemplaire préprogrammé (130475-41).

Dans le circuit imprimé, il y a deux rangées de trous pour permettre l'utilisation de tout type de régulateur de tension. Lors du montage de votre régulateur et de son radiateur, il convient d'y prêter attention, pour éviter que le radiateur ne vienne court-circuiter certaines des pastilles. Intercalez une rondelle isolante entre radiateur et circuit imprimé.

Pour le raccordement des LED et des entrées, utilisez des embases femelles d'une rangée ainsi que leurs pendants mâles. Scindez les barrettes pour obtenir le nombre requis de contacts. Il est ainsi possible d'en souder 13 sur le circuit imprimé et d'utiliser le reste pour les enficher par-dessus. L'alignement est plus facile à réaliser et les soudures plus propres.

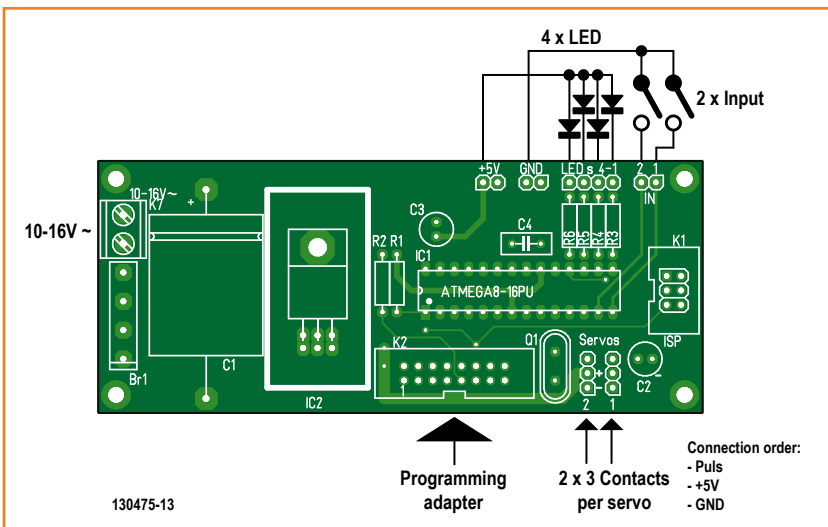


Figure 4.
Le branchement de la console, des entrées et des sorties sur l'unité à microcontrôleur.

R3 à R6 sont les résistances en série avec les LED. Elles adaptent le courant aux besoins des LED. Les exemplaires de 470 Ω conduisent à un courant proche de 7,5 mA dans des LED rouges.

Il y a différentes manières de fixer l'écran LCD sur le circuit imprimé. Si vous voulez l'ensemble le plus plat possible, vous pouvez le relier par de

petits fils au circuit imprimé. Il faut alors veiller à ce que la saillie qui entoure la fenêtre sur la face inférieure de l'écran ne provoque pas de court-circuit entre les pistes du circuit imprimé. Une liaison par connecteur facilite le service et le dépannage. Le clavier et l'écran ne sont alors plus au même niveau. À vous de choisir le moindre mal.

Si vous confectionnez le câble de liaison à 16 conducteurs entre *le cerveau de servos* et la console, n'oubliez pas qu'il y a des détrompeurs sur les parties mâle et femelle qui évitent de les brancher à l'envers. C'est donc au moment de serrer la fiche sur le câble qu'il faut faire coïncider la broche 1 avec le 1^{er} contact.

La mise en place des composants sur les deux circuits imprimés (**fig. 3 et 5**) est à la portée d'un électronicien peu expérimenté. À l'issue de ce travail, il y a lieu d'exercer un contrôle général de la distribution de la tension d'alimentation de 5 V. Ce n'est qu'après qu'on insère le contrôleur, dans le bon sens, évidemment ! Le potentiomètre ajustable pour le contraste de l'écran, on le tourne à fond vers la gauche, puis on peut relier entre elles les deux parties du système et y raccorder

Liste des composants

Console pilotage et programmation

Résistances :

R10 = 10 kΩ aj. RM5/10 mm
R12 = 10 Ω
R15 = 4,7 kΩ

Condensateur :

C11 = 100 μ F/16 V vert. au pas de 2,54 mm

Semi-conducteur :

T2 = BD433 ou BD435

Divers :

S1a = poussoir central pour *Enter*

bouton pour S1a

S1b à e = poussoirs pour circonférence

4 boutons pour S1b à e

S3 = bouton-poussoir

K3 = embase coudée à 16 picots

2 connecteurs pour câble plat à 16 conducteurs

câble plat à 16 conducteurs

écran LCD 16x2, p.ex. Displaytech162C avec ou sans
éclairage d'arrière-plan

barrette femelle à 20 contacts en ligne au pas de 2,54 mm

embase à 20 picots en ligne au pas de 2,54 mm pour LCD

2 boulons M2,6 avec entretoise

circuit imprimé 130475-2, cf. [1]

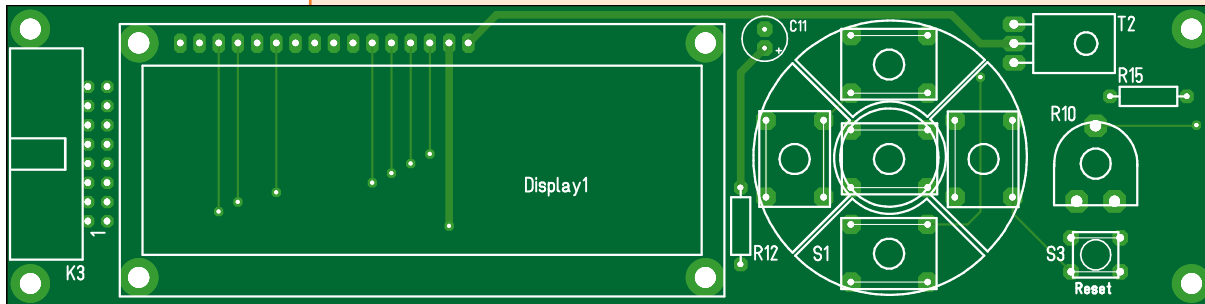


Figure 5.
Le circuit imprimé de la
console de pilotage et de
programmation.

l'alimentation. Si l'écran affiche quelque chose, c'est que tout a bien marché ! On cherche alors le meilleur contraste avec le petit potentiomètre. Et c'est parti !

Au programme !

Le code source est rédigé en BASCOM et remplit presque toute la mémoire disponible. Le programme a été développé pour aller jusqu'à huit servos et avec éventuellement un autre contrôleur. Il n'a donc pas été optimisé pour réduire l'utilisation de la mémoire. La majeure partie du programme sert à l'utilisateur pour effectuer les réglages, tandis que les routines nécessaires prennent moins d'espace. Le prix à payer pour le confort d'utilisation !

On commence par déclarer toutes sortes de variables et configurer les ports. En maints endroits, ce sont des noms d'alias qui sont utilisés pour faciliter l'adaptation à d'autres contrôleurs. Les noms des constantes facilitent aussi la compréhension du code source. Parfois, l'ordre des procédures peut sembler étrange, mais cela minimise autant que possible les vibrations incontrôlées des servos.

En service normal, tout se passe dans une boucle sans fin au cours de laquelle, dans chaque mode, on interroge les entrées ou les boutons, le cas échéant les nouvelles positions à atteindre. Suit alors la routine qui, selon la vitesse de positionnement demandée, calcule le moment où la position sera atteinte. C'est là que la position actuelle du servo remplacera la valeur désirée. Il faut pour cela parcourir jusqu'à 2 000 pas, bien qu'à l'écran n'apparaissent que des valeurs entre 0 et 200. La réalisation du clignotement des LED ne fait pas appel aux fonctions MLI disponibles. J'ai constaté que l'usage des interruptions (outre celles pour les impulsions des servos) provoquait de trop petites variations dans la largeur d'impulsion des servos et du coup, des déplacements trop petits, mais bien audibles.

Vient alors le tour de la programmation des servos, dans laquelle on utilise principalement l'instruction d'antirebond pour les boutons-poussoirs. Finalement, on trouve les routines d'interruption et les autres plus générales.

Quand on programme un contrôleur, il faut écrire dans la mémoire flash, mais aussi dans l'EEPROM.

L'oubli de cette dernière priverait de toutes les fonctions ! Quelques adaptations sont évidemment possibles. Se passer de multilinguisme, par exemple, permet de conserver quelques octets libres.

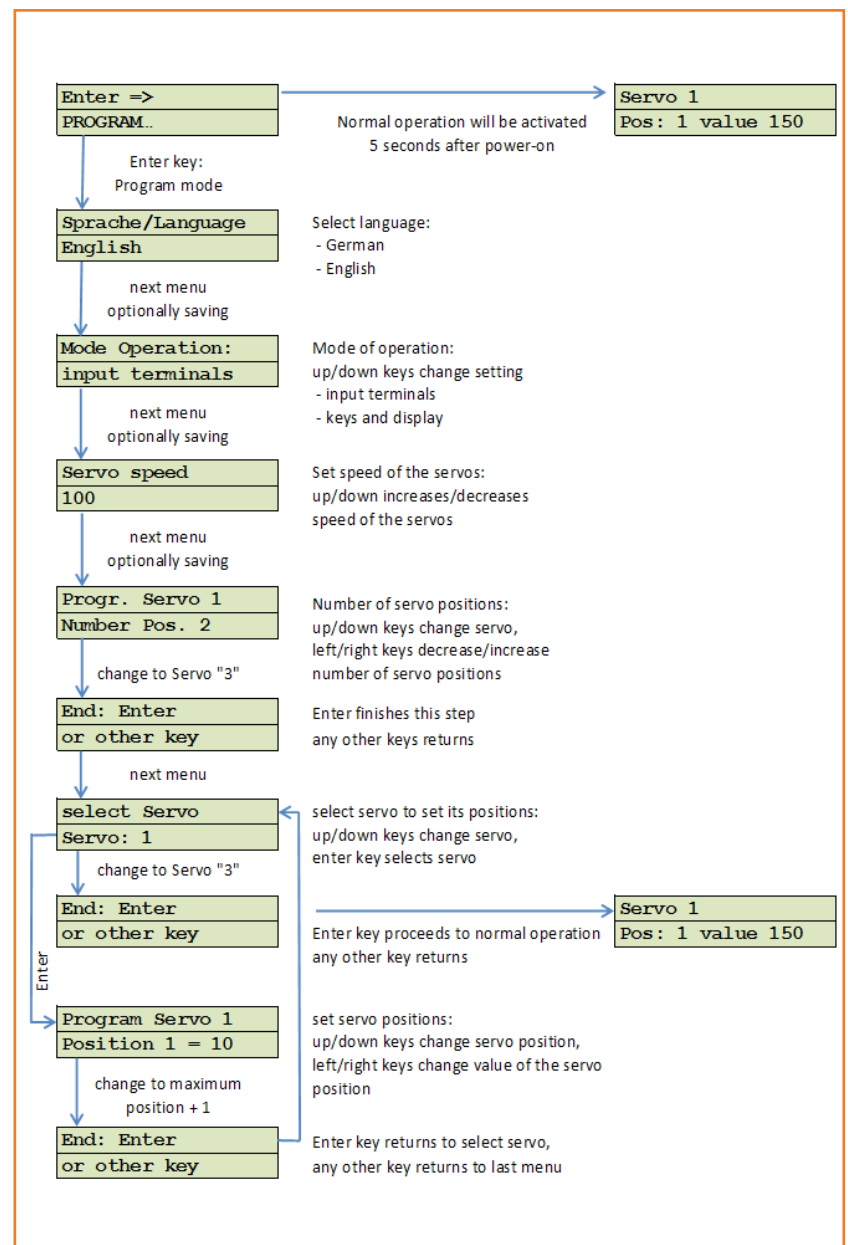
Amusez-vous bien !

(130475 – version française : Robert Grignard)

Lien

[1] www.elektor.fr/130475

Figure 6.
Diagramme de la logique de programmation du système.





alimentation de table régulée par Platino

modèle d'application embarquée

**Sunil Malekar /
Krishnachandran**
(elektor.labs, Inde)

De nombreuses alimentations de table sont affublées d'un microcontrôleur dont la fonction se limite en général au contrôle des lectures V et A. Celui de notre alimentation fait plus que de la figuration sur étiquette : fonctions de contrôle, régulation interne, ventilateur, tout est géré par la carte polyvalente Platino et son ATmega.

La petite alimentation décrite ici a été inspirée par *l'alimentation de laboratoire pour développement embarqué* publiée en avril 2012 [1], dont le microcontrôleur se contentait de piloter l'afficheur.

Ici, en revanche, le microcontrôleur contrôle la tension de sortie, l'intensité de sortie maximale, et bien plus. Et il ne manque pas de puissance puisqu'il s'agit de l'ATmega de la carte Platino d'Elektor [2]. Un bel exemple d'application d'une carte embarquée.

Présentation

Incorporer un microcontrôleur dans une alimentation de table, c'est chercher à réduire le nombre de potentiomètres et de boutons à l'aide d'une poignée de composants. Un tel parcours est semé d'embûches, mais grâce à la carte Platino, à du code Bascom et à un potentiomètre numérique, nous y sommes parvenus, réduisant au passage et le nombre de composants et l'encombrement de la carte.

Notre alimentation devrait convenir à la plupart des travaux « légers » puisqu'elle délivre par pas de 0,1 V des tensions comprises entre 0 V et 15 V, avec une limite de courant réglable entre 0 et 1000 mA. Elle fournit aussi des tensions fixes de 3,3 V et 5 V sur un connecteur USB, avec une limite de courant réglable entre 0 et 500 mA. L'afficheur a deux modes : paramétrage (*Setup*) et normal (*Normal*).

Description du circuit

C'est la forte interaction entre les composants matériels et logiciels de notre alimentation qui garantit la stabilité et la précision des valeurs de sortie et des paramètres sélectionnés par l'utilisateur. Le cœur du projet est la carte Platino d'Elektor accompagnée de son ATmega328p.

Notons, avant de survoler les différentes parties du circuit (**fig. 1**), que la tension d'entrée est fournie par un adaptateur 18 V CC d'ordinateur portable (ou tout autre adaptateur équivalent).

Tensions principale et auxiliaire

La tension d'entrée continue de 18 V est appliquée au bornier K1. La diode D1 protège le circuit contre les inversions de polarité. Les +18 V continus vont aux régulateurs IC3 et IC4.

Le premier (IC3, un LM7805) délivre les 5 V attendus par Platino et la circuiterie fonctionnant sous 5 V. Le second (IC4, un LM7812) délivre 12 V CC régulés au circuit de commande du ventilateur.

Tensions négatives auxiliaires

La branche qui régule la tension positive a besoin de -1,27 V pour abaisser sa sortie à 0 V. Pour cela le circuit qui entoure IC5 et IC6 convertit le +5 V en -1,27 V ($-V_E$) en passant par un niveau intermédiaire de -5 V. La sortie du régulateur de tension négative LM337L est réglée sur -1,27 V à l'aide de P2.



Mesure de la température

La température est mesurée par un capteur LM335 (IC7) dont la tension de sortie est appliquée sur l'entrée PC4 de Platino. L'ATmega lit et traite cette tension, puis affiche la température. La valeur mesurée sert à couper automatiquement l'alimentation lorsqu'une certaine température est dépassée. Vous pouvez régler le seuil de coupure (*Threshold Temperature*) dans le mode *Setup*.

Contrôle du ventilateur

Le petit ventilateur de 12 V CC placé à l'arrière du coffret refroidit l'unité et évite les avaries en cas de surchauffe. Il est alimenté par un MOSFET BS170 (T1) via la broche PD1 de la Platino. Si le cavalier n'est pas implanté sur K8, la broche PC6 de Platino est au niveau haut et le ventilateur par conséquent en mode manuel. Le mode *Setup* permet de mettre le ventilateur sur mode *Auto* ou *Manual*. En mode *Auto*, le ventilateur est commandé par Platino, c.-à-d. activé lorsque la température atteint $T = \text{Threshold Temperature} - 20^\circ\text{C}$. Le ventilateur reste désactivé en deçà de cette valeur. En mode *Manual*, le programme l'active/désactive selon la présence ou l'absence de cavalier sur K8.

Générateur de tension de référence

Le LM336 (IC11) produit la tension de référence *AREF* pour Platino. Le μC effectue les calculs de tension d'après cette valeur. Le LM336 en boî-

Caractéristiques

- Sortie 1 : 3,3 V ou 5 V / 500 mA max., sur connecteur USB-A
- Sortie 2 : 0 à 15 V / 1 A max., sur prises banane
- Protection des deux sorties contre les courts-circuits
- Puissance : 17,5 W
- Entrée CC : 18 V / 2 A (alim. d'ordinateur portable)
- Affichage : LCD 20 x 4 caractères
- Réglages par bouton unique
- Coupure automatique en cas de température élevée
- Modes automatique ou manuel pour le ventilateur
- Capteur de température interne
- Seuil de déclenchement du ventilateur paramétrable
- Mode normal pour les réglages V/A d'usage quotidien

tier TO-92 fournit une référence fixe précise et ne nécessite que peu de composants externes. L'ajustable P1 sert à mettre *AREF* sur exactement 5,00 V (servez-vous d'un bon voltmètre) une bonne fois pour toutes.

Régulateurs de tension positive

L'alimentation a deux sorties, donc deux régulateurs de tension positive, séparés mais identiques par souci de simplicité. Le LM2576 (IC1, IC2) est un régulateur/convertisseur abaisseur réglable, utilisé ici comme régulateur de tension positive. IC1 est configuré pour délivrer jusqu'à 1 A sous une tension de 0 V à 15 V. IC2 régule les 3,3 V (500 mA max.) et 5 V (500 mA max.) qui

peuvent être sélectionnées sur K7. Les voies IC1 et IC2 sont contrôlées de façon quasi numérique par Platino.

La tension de sortie de 0 à 15 V est disponible sur le connecteur K6. La tension auxiliaire négative $-V_E$ est reliée à la broche GND (3) du LM2576 de façon à ce que la tension de sortie descende jusqu'à 0 V, et non pas à jusqu'à seulement +1,23 V comme l'indique la fiche technique. Les +18 V CC appliqués sur IC1 et IC2 arrivent par D1. Les circuits qui filtrent leurs sorties sont identiques, seuls les courants nominaux des bobines (L1/L2) et des condensateurs réservoirs (C11/C21) diffèrent.

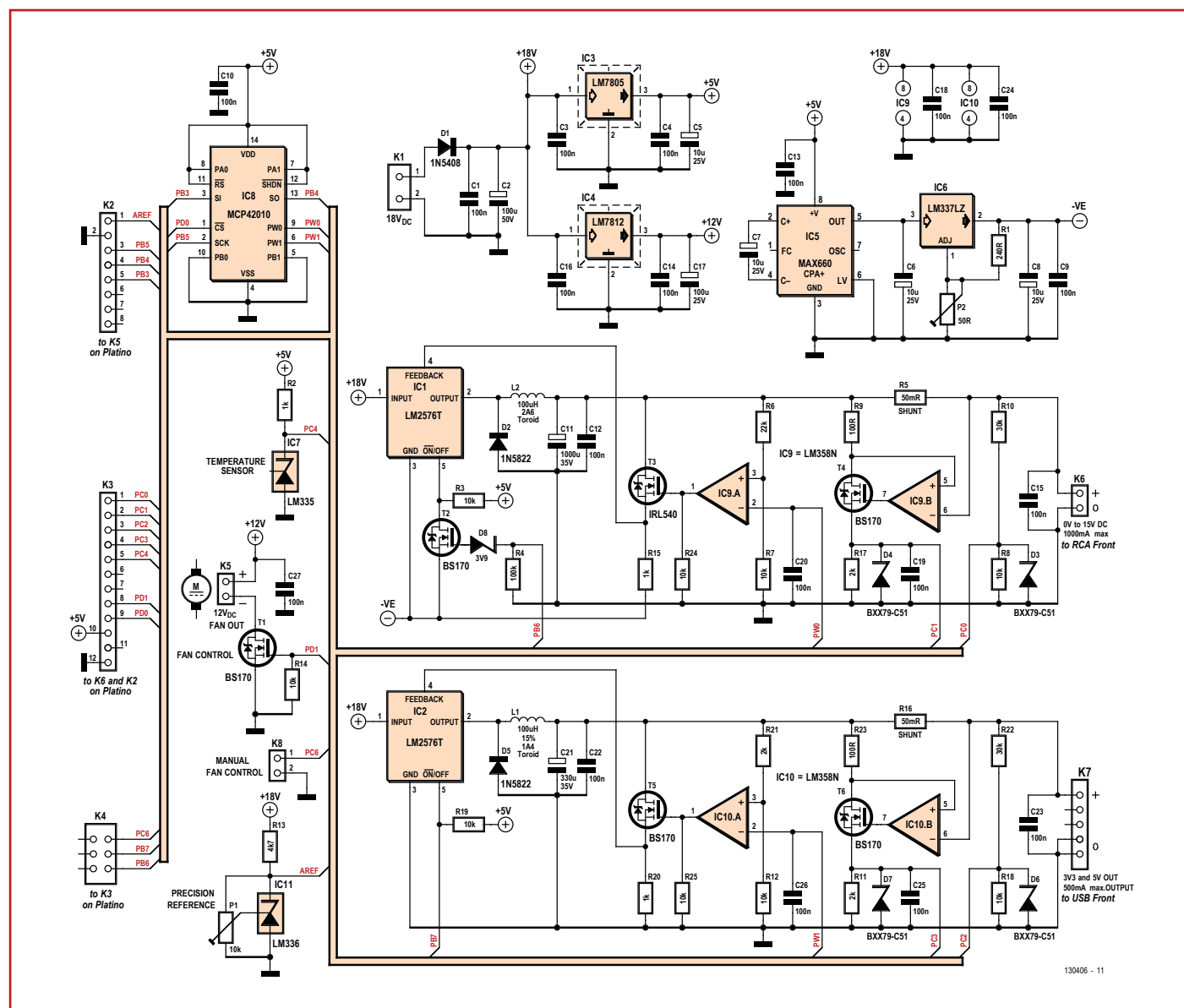
Figure 1.
Schéma de l'alimentation.
La tension d'entrée est
fournie par une alimentation
classique de portable de
18 V et 2 A.

Coupeure automatique

La coupeure automatique protège à la fois l'alimentation et l'utilisateur contre les courts-circuits, les surintensités et les températures excessives. La broche PB6 de Platino est utilisée pour commuter IC1 via le MOSFET T2. La source de T2 est reliée à $-V_E$, d'où la zener D8 pour éviter les tensions négatives sur PB6. La broche PB7 sert de même à désactiver IC2 en cas de court-circuit, de surintensité ou de surchauffe détectés sur sa sortie.

Réaction et correction d'erreurs

Une branche « réaction et correction d'erreurs » contrôle les tensions et les courants de sortie des



130406 - 11

deux régulateurs. Composée pour l'essentiel d'un ampli-op et d'un MOSFET, cette partie déclenche et contrôle la réaction d'IC1 et d'IC2 grâce à un signal fourni par Platino.

Dans le cas d'IC1, PW0 porte le signal « potentiomètre numérique » qui contrôle la tension de la réaction. Le signal de mesure du courant est transmis au μC via la broche PC1 ; celui qui mesure la tension est transmis via PC0.

Côté IC2, c'est PW1 qui porte le signal « pot. numérique ». Ceux des mesures de courant et de tension passent par PC3 et PC2.

Voyons comment opère cette branche de réaction et de correction d'erreurs. Les AOP IC9a et IC10a ajustent leurs sorties en fonction de la différence de potentiel entre leurs entrées inverseuses et non-inverseuses. Ces sorties déterminent l'impédance vue par la broche de réaction (4) du régulateur. Supposons qu'une tension comprise entre 0 et 5 V soit appliquée sur l'entrée inverseuse d'un ampli-op. Sa tension de sortie sera alors bien inférieure au seuil de commutation du MOSFET, dont la liaison drain-source représentera alors une impédance élevée. Pour que les 1,23 V soient atteints sur la broche de réaction, le régulateur met sa sortie au niveau haut ; comme cette sortie est réinjectée sur l'entrée non-inverseuse, la sortie de l'AOP augmente et l'impédance diminue. L'ensemble des circuits s'efforce ainsi de maintenir égales les tensions appliquées sur les entrées inverseuses et non-inverseuses de l'AOP.

La branche qui mesure le courant a pour cœur IC9b et IC10b. Le courant qui traverse la charge est converti en tension par les résistances shunt R5 (pour IC1) et R16 (pour IC2). La chute de tension aux bornes de R5 et R16 va aux entrées inverseuses des AOP. Les tensions font monter les sorties des AOP, l'impédance des MOSFET diminue, ce qui entraîne une chute de tension sur les entrées non-inverseuses. Idéalement, la boucle de réaction équilibre les tensions des entrées inverseuses et non-inverseuses. Une chute de tension identique à celle qui règne aux bornes des résistances en série se produit aux bornes des résistances R9 (IC1) et R23 (IC2) de 100 Ω . Le courant qui traverse la 100 Ω est donc proportionnel au courant de charge, qui traverse également les résistances R17 (IC1) et R11 (IC2) de 2 k Ω . Le μC se sert de cette « tension proportionnelle au courant » pour détermi-

ner la demande en courant de l'alimentation. Des zeners de 5,1 V (D3, D4, D6 et D7) protègent les broches concernées du μC . La plage de sortie de 0 à 15 V est convertie en intervalle de 0 à 5 V, la plage admissible pour le μC .

Potentiomètre numérique

Le MCP42010 (IC8) est un potentiomètre numérique de 50 k Ω , à deux canaux et doté d'une interface SPI. Il fournit la tension de réaction, sous le contrôle du μC . La ligne PW0 est reliée à IC1 via IC9a, la ligne PW1 à IC2 via IC10a. Les signaux SPI PB3, PB4, PB5 et PD0 en provenance du μC commandent le « curseur » du MCP42010 de façon à ce qu'il fournisse les niveaux de tension nécessaires à la boucle de régulation.

Sortie

La sortie du régulateur IC1 est reliée à la charge via les connecteurs banane du coffret reliés à K6. La tension de sortie d'IC2 va au connecteur USB du panneau avant via K7.

Programme

Écrit en BASCOM AVR pour ATMEGA328P, le code gère les éléments décrits ci-dessous. Vous pouvez le télécharger depuis [3]. La carte Platino permet de développer plus avant le projet.

Affichage

La partie du code qui gère l'affichage initialise le LCD de 20 x 4 caractères avant d'afficher les messages *Platino Instrument series 1.0* puis *Platino Adjustable Bench Power Supply*, le nom du projet. Apparaît ensuite le menu et ses options *Setup Mode* et *Normal Mode*. Servez-vous du codeur rotatif et de son bouton intégré pour sélectionner l'option souhaitée. Appuyez sur le bouton pour voir les options des modes *Normal* ou *Setup*. Pour revenir au menu principal, appuyez longuement sur le bouton.

Mode Setup

Sélectionnez et paramétrez ici la tension, le courant et le seuil d'activation du ventilateur. Appuyez longuement sur le bouton pour revenir au menu principal.

Mode Normal

Les sorties sont activées pour alimenter la ou les charges. Le codeur rotatif permet de régler la sortie entre 0 V et 15 V. Ici aussi, appui long



Liste des composants

Résistances

Tolérance et puissance par défaut :

5 % et 0,25 W

R1 = 240 Ω

R2, R15, R20 = 1 k Ω

R3, R7, R12, R14, R19, R24, R25 = 10 k Ω

R4 = 100 k Ω

R5, R16 = shunt de précision, 50 m Ω , 1 %, 2 W

R6 = 22 k Ω

R8, R18 = 10 k Ω , 1 %

R9, R23 = 100 Ω , 1 %

R10, R22 = 30 k Ω , 1 %

R11, R17, R21 = 2 k Ω , 1 %

R13 = 4,7 k Ω

P1 = aj. 10 k Ω

P2 = aj. 50 Ω

Condensateurs

C1, C3, C4, C9, C10, C12 à C16, C18, C19,

C20, C22 à C27 = 100 nF

C2 = 100 μ F/50 V, radial

C5 à C8 = 10 μ F/25 V radial

C11 = 1000 μ F/35 V radial

C17 = 100 μ F/25 V radial

C21 = 330 μ F/35 V radial

Inductances

L1 = 100 μ H/1,4 A (2062790)

L2 = 100 μ H/2,6 A (2215967)

Semi-conducteurs

D1 = 1N5408

D2 = 1N5822

D3, D4, D6, D7 = zener 5,1 V/0,4 W

D5 = 1N5819

D8 = zener 3,9 V/0,4 W

T1, T2, T4 à T6 = BS170

T3 = IRL540NPBF

IC1, IC2 = LM2576T-ADJ

(Texas Instruments) (9488146)

IC3 = LM7805

IC4 = LM7812

IC5 = MAX660CPA+

IC6 = LM337LZ

IC7 = LM335

IC8 = MCP42010-E/P (Microchip) (1332110)

IC9, IC10 = LM358N

IC11 = LM336BZ-5.0

Divers

K1 = bornier pour CI à 2 voies
K5, K6, K8 = barrette à 2 broches

K2 = barrette à 8 broches

K3 = barrette à 12 broches

K4 = barrette à 3x2 broches

K7 = barrette à 5 broches

ventilateur miniature, 12 V,
40 x 40 x 15 mm

2 prises banane, 1 rouge,
1 noire, montage sur panneau
(1176431) (1176430)

prise USB-A, montage sur

panneau (1667928)

coffret

Bopla 75 x 133 x 110 mm,
26160000 (1217479)

interrupteur, on/off,

inverseur, montage sur

panneau, p. ex. Schurter

1301.9205 (1162728)

circuit imprimé,

e-shoppe Elektor 130406

carte Platino,

e-shoppe Elektor 110645

(références du catalogue
Farnell entre parenthèses)

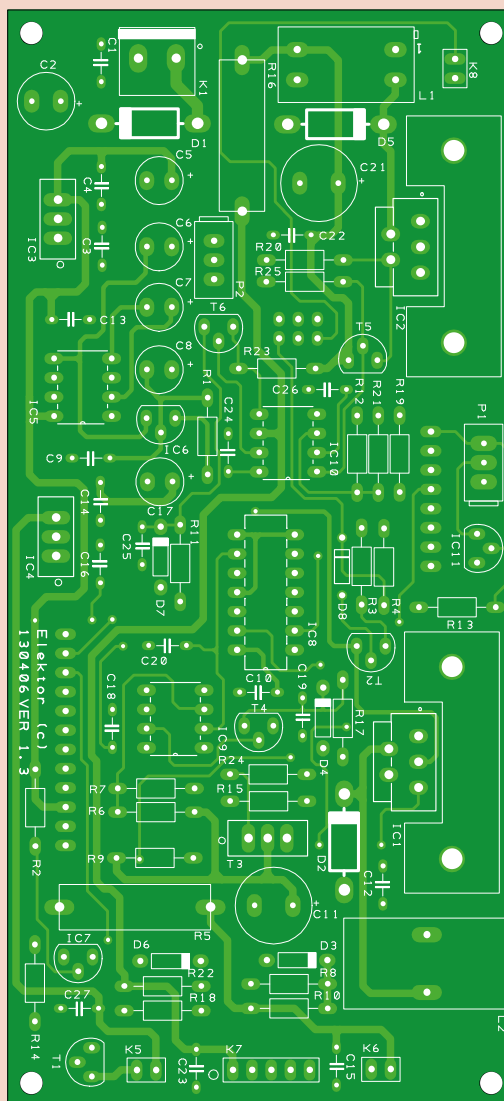
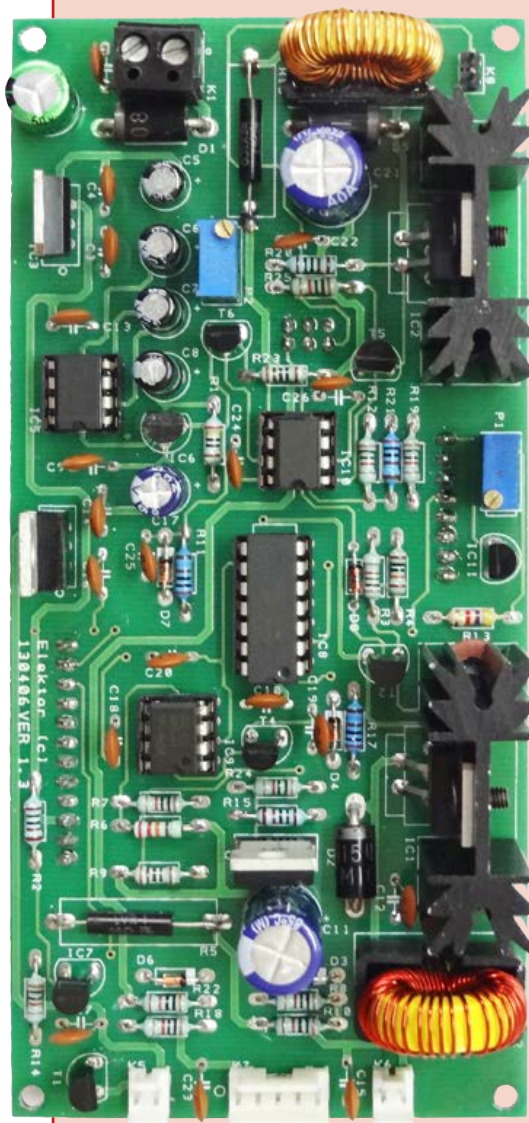


Figure 2.
Dessin de la carte
d'alimentation.

sur le bouton pour revenir au menu principal. L'ensemble des sorties n'est disponible que dans ce mode.

Mesure

Le microcontrôleur reçoit de la branche matérielle « réaction et correction d'erreurs » les données concernant la tension de sortie et la réaction d'intensité. Il mesure également la tension et le courant relatifs au signal de référence présent sur la broche *AREF*. La température est calculée puis affichée par le programme. Les signaux de mesure sont réduits à l'intervalle 0-5 V par le matériel, puis envoyés au CAN du µC.

Contrôle

Les signaux dont se sert le potentiomètre numérique IC8 pour contrôler les tensions et les courants de sortie sont produits par le µC d'après la valeur d'entrée du CAN. Le µC produit aussi les signaux de coupure automatique pour IC1 et IC2 d'après les valeurs de courant, température et seuil paramétrées dans le mode *Setup*. Le signal de commande du ventilateur produit par le µC dépend de même du paramétrage du mode *Setup*. Une partie du code lit la tension de sortie et incrémente ou décrémente le potentiomètre numérique de façon à atteindre le niveau de tension demandé. Le µC surveille aussi le courant et limite la tension de sortie si le courant de charge dépasse la limite de courant. Si un court-circuit est détecté (surintensité), le contrôleur coupe le régulateur concerné et remet à zéro le MCP42010. Il coupe de même le régulateur si un excès de température est détecté. L'utilisateur peut programmer les limites de courant et de température.

Les fonctions des broches utilisées par le matériel et le programme sont résumées dans le **tableau 1**.

Construction

Commencez par la Platino [2] : assemblez le LCD, le bouton/codeur rotatif, l'ATMEGA328P et le cortège de composants, puis positionnez les cavaliers (**tableau 2**).

Passez ensuite à la carte d'alimentation (**fig. 2**). Elle épousera parfaitement le dos de Platino grâce aux barrettes mâles. C'est un coffret Bopla qui loge la carte Platino, le LCD, le bouton/codeur

Tableau 1. Utilisation des broches du contrôleur	
Broche	Fonction
PB0 à PB2	codeur et son bouton
PB3, PB4, PB5, PD0	µcontrôleur <-> ports SPI du pot. numérique entrée des données = PB4 sortie des données = PB3 sélection de puce = PD0 horloge = PB5
PB7, PB6	activation/désactivation des régulateurs
PC0 à PC3	tensions de sortie et mesure du courant sur IC1/IC2
PC4	mesure de température
PC5	rétro-éclairage du LCD
PC6	commande du ventilateur



Figure 3. L'intérieur du coffret. Les trois cartes (LCD, Platino et carte d'alimentation) sont montées verticalement derrière le panneau avant.

rotatif, la carte d'alimentation, le petit ventilateur de 12 V, le jack d'alimentation, l'interrupteur on/off, ainsi que les connecteurs banane et USB (uniquement pour alimentation !)

Les photographies de l'article (**fig. 3** et **4**) montrent la construction de cette petite alimentation de table. Les cartes sont montées verticalement derrière le panneau avant, dans cet ordre : LCD, Platino, carte d'alimentation.

Sur le panneau avant, découpez trois ouvertures rectangulaires pour le LCD, l'alimentation USB et

Tableau 2. Placement des cavaliers sur Platino	
JP3	PC5
JP4	PB0
JP5	PB1
JP6	PB2
JP10	PB6
JP9	PB7
JP8	RESET
JP11	PB7
JP12	PB6

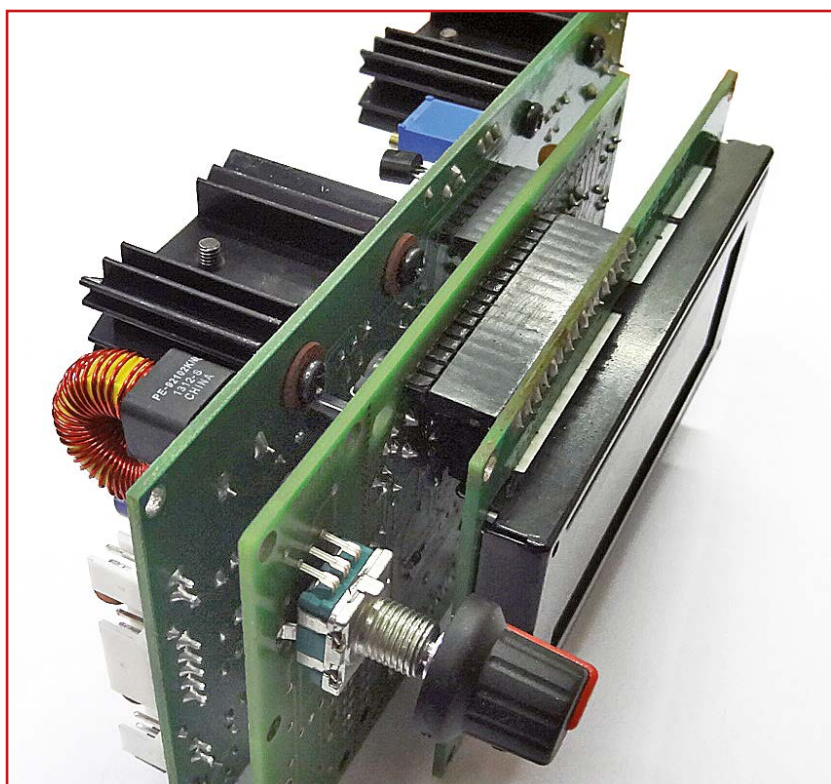


Figure 4. Gros plan sur les trois cartes. Il est plus pratique de tester l'alimentation telle quelle, c.-à-d. avant sa mise en boîtier.

de la prise d'alimentation 12 trous de 2 mm en forme de croix de 40 x 40 mm.

Il n'y a de câblage interne que pour le ventilateur, le bouton on/off, l'alimentation USB, l'entrée de +18 V et les prises banane.

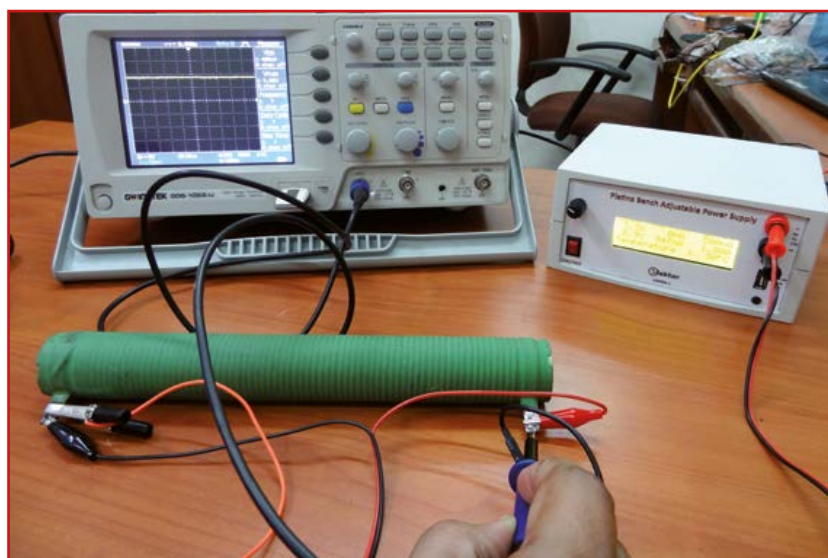
Essai

Testez votre alimentation sous pleine charge avec la rigueur et les précautions de sécurité qui s'imposent ! Chargez la sortie 0-15 V / 1 A en y reliant une résistance de puissance de 15 Ω / 30 W (**fig. 5**). Chargez la sortie 5 V / 500 mA à l'aide d'une résistance de 10 Ω / 10 W. Testez aussi une charge dynamique de 1 kHz pour 15 V (1 A), 10 V (1 A) et 5 V (500 mA). Vérifiez les fonctions de rétablissement après court-circuit et de coupure automatique en court-circuitant les sorties avec la masse.

le bouton marche/arrêt, et trois trous pour l'axe du codeur et les deux prises banane. Collez les supports, ceux des câbles en plastique p. ex., sur le fond du coffret afin de garantir l'immobilité des trois cartes. Sur le panneau arrière, découpez des trous pour le ventilateur (\varnothing d'env. 37 mm) et le connecteur d'alimentation (\varnothing d'env. 10 mm). Pour une meilleure ventilation, percez au-dessus

Pour vérifier la détection de surintensité et sa réponse, paramétrez la valeur limite, baissez légèrement la tension de sortie, puis regardez si le courant est maintenu à l'intensité définie.

(130406 – version française : Hervé Moreau)



Liens

- [1] Alimentation de laboratoire pour développement embarqué, Elektor avril 2012, www.elektor-magazine.fr/110645
- [2] Platine polyvalente pour montages à micro-contrôleur(s) AVR, Elektor octobre 2011, www.elektor-magazine.fr/100892
- [3] Programme : www.elektor-magazine.fr/130406

Figure 5. La résistance de 15 Ω et 30 W pour tester la sortie maximale (1 A sous 15 V). Ne la laissez pas trop longtemps ainsi durant le test (et ne la touchez pas non plus).

gentil petit canard

Ce n'est pas un tigre qu'elektor.labs a mis dans son moteur de recherche, mais un canard et pas n'importe lequel : *DuckDuckGo* est un moteur réputé pour respecter l'anonymat de l'utilisateur sur l'internet. Nos premières impressions sont positives : des réponses rapides, précises et sans mouchardage. Fini l'espionnage façon *Big Brother*. Vous devriez essayer.

Clemens Valens
(elektor.labs)

Il est facile de collecter des informations, les rendre accessibles est une autre paire de manches. Depuis son lancement, le site elektor.labs a recueilli beaucoup d'informations utiles sur de nombreux projets. De bons outils sont nécessaires pour exploiter toutes ces données. Et comme laboratoire, Elektor.Labs est aussi



Site Search laquelle fournissait des résultats spectaculairement précis ; même les partisans du moteur initial se disaient impressionnés. L'inconvénient des liens commerciaux ajoutés aux résultats par *Google* nous a paru compensé par la précision des résultats. Nous ne tardâmes pas à nous apercevoir que *Google Site Search* était si fort

DuckDuckGo

le cadre idéal pour expérimenter avec les moteurs de recherche et trouver la solution optimale.

Au début, nous utilisions par défaut le moteur de recherche intégré dans Drupal, le système de gestion de contenu du site, parfait tant que vous savez exactement ce que vous cherchez. Saisissez l'expression recherchée et il rapporte dans ses filets tous les articles contenant cette expression exacte. À la moindre divergence, il ne retrouve rien, et les caractères génériques, il ne les connaît pas.

Pourquoi ? Il existe pour les moteurs de recherche (au moins) deux écoles de pensée : l'une privilégie un nombre réduit de résultats précis, l'autre un grand nombre de résultats moins précis. Notre site a adhéré jusqu'ici à la première école ; à l'usage, nous avons préféré la seconde, mieux adaptée aux besoins de visiteurs qui ne savent pas encore précisément ce qu'ils cherchent.

C'est ainsi que le moteur de recherche par défaut de *Drupal* fut remplacé par la fonction *Google*

qu'il fournissait des résultats sur des pages inconnues, sous la forme p. ex. de listes de nouveaux messages à l'échelle du site. D'abord cela paraissait profitable, mais rapidement ces annonces intempestives ont submergé les résultats utiles. Les désactiver n'a servi à rien, car entretemps *Google* les avait mémorisées. Il ne restait qu'à tirer la prise de *Google Site Search*.

C'est ainsi qu'entre en scène le canard *DuckDuckGo*, moteur de recherche assez récent qui affirme préserver l'anonymat tout en limitant le *spam* au minimum. La fonction de recherche est moins parfaitement intégrée au site que ne l'était celle de *Google*. Lorsque vous cliquez sur le bouton de recherche d'Elektor.Labs, vous passez par une page de résultats sur le site *DuckDuckGo*. C'est un petit détour, certes, mais à la vitesse de l'éclair. Attention, n'effacez pas dans la boîte de recherche l'indication site: www.elektor-labs.com.

(130461)

www.elektor-labs.com/node/3795

journalisation de la fréquence du secteur

enregistrez sur une clé USB la trace des écarts d'EDF

Wolfgang Borst
(Allemagne)

De minuscules écarts de fréquence dans la tension du secteur révèlent des bouleversements dans l'équilibre entre l'offre et la demande d'énergie électrique. La *loupe de réseau* du numéro de janvier 2012 montrait comment les mesurer, je propose maintenant de les enregistrer automatiquement.

Nul besoin d'être électronicien pour savoir que la fréquence du réseau électrique est de 50 Hz en Europe, mais qu'aux États-Unis, par exemple,

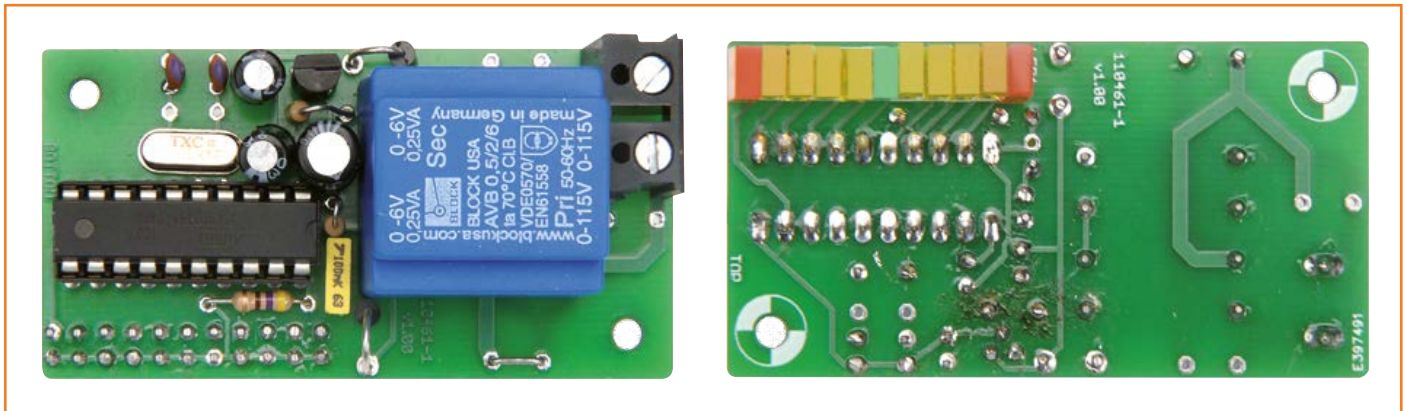
elle est de 60 Hz. C'est la mise en réseau de différents moyens de production électrique qui oblige au respect strict de la fréquence, la marge d'erreur ne devrait pas dépasser $\pm 0,2$ Hz. Dans des cas extrêmes, comme celui survenu le soir du 4 novembre 2006 dans la Ruhr, en Allemagne, une pénurie manifeste d'approvisionnement en électricité a fait sombrer la fréquence pendant plusieurs minutes à 49 Hz. Les écarts de fréquence constituent bel et bien une mesure de l'adéquation du bilan énergétique et indiquent la qualité instantanée du réseau.

Caractéristiques techniques

- Surveillance de la fréquence du secteur
- Enregistrement à longue durée
- Stockage sur clé USB
- Résolution de fréquence de 2,5 mHz
- Précision de mesure absolue : 25 mHz
- Affichage de la fréquence courante par LED
- Affichage graphique à l'écran de la courbe de fréquence
- Adaptable à 230 et 115 V ainsi que 50 et 60 Hz
- Code source intégralement disponible

Mesure et écriture pour le suivi

Pour la mesure, nous disposons déjà du projet de *loupe pour le réseau EDF*, paru dans *Elektor* en janvier 2012 [1]. Il mesure la fréquence courante du secteur et l'affiche sur 11 LED dans une plage de $\pm 0,2$ Hz. Aux extrémités, les LED rouges



signalent que le réseau nous fait un caprice. Le circuit imprimé est disponible et la **figure 1** présente le montage compact réalisé de la loupe de fréquence. Le transformateur y est embarqué, il s'adapte aux deux tensions existantes au monde, 115 et 230 V. De quoi résoudre la question de la mesure.

L'idée est bonne, mais à quoi bon disposer de la seule valeur du moment présent ? Je souhaite observer la chronologie de cette fréquence sur une certaine période. Pour cela, il faut enregistrer quelque part les mesures récurrentes effectuées par le fréquencemètre. Comme il restait sur *la loupe de réseau EDF* (ou de tout autre fournisseur de courant, d'ailleurs !) une broche inutilisée du microcontrôleur et un surplus de mémoire de programme, j'ai doté *la loupe de fréquence* d'une interface série pour en extraire les mesures. Reste à mémoriser ces données.

Pour enregistrer et stocker les données sérielles reçues, rien de plus simple que d'utiliser le PC que l'on a toujours sous la main. Il suffit alors d'un programme de terminal à régler sur les paramètres de l'interface annexe de la loupe pour introduire les données. Il m'a semblé plus agréable de visualiser les mesures sous forme graphique (**fig. 2**), et j'ai rédigé un programme en ce sens. Se passer de PC serait plus élégant. Or, il existe chez Elektor une solution toute faite pour la journalisation de donnéesérielles : la *Clé USB sur microcontrôleur* [2] de janvier 2012 aussi (**fig. 3**). Un PIC24 se charge d'enregistrer sur une banale clé USB les données séquentielles qui lui parviennent.

Il suffit donc de combiner les deux modules, de câbler une petite interface, avec une modifica-

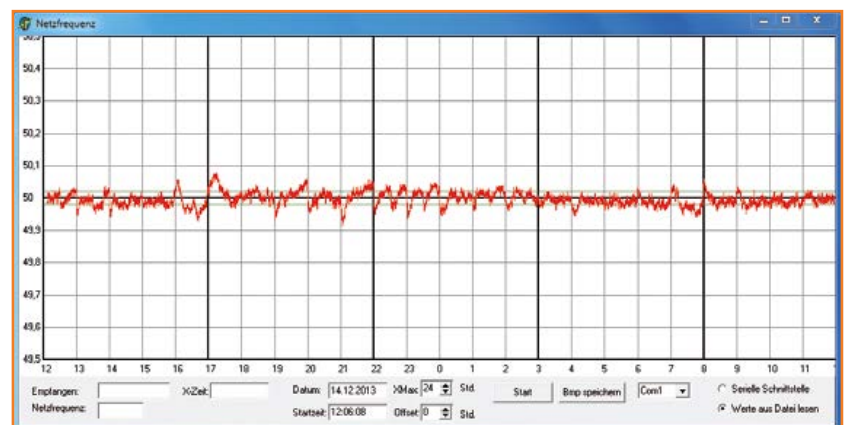
tion du micrologiciel de la loupe, pour conserver précieusement l'historique de l'état du réseau électrique. Pour le restituer, j'ai aussi rédigé un programme pour le PC qui fournit une représentation graphique à long terme de la situation.

Modification du matériel

Cette fameuse broche providentiellement libre du microcontrôleur, c'est P3.1, une sortie de l'UART matériel. Pour relier *la loupe EDF* à l'enregistreur de données ou au PC, il n'y a qu'à faire aboutir cette broche vers l'extérieur : un bout de câble à souder sur la broche 3 de IC1, plus une liaison de masse, de quoi câbler l'extension nécessaire. La **figure 4** montre le schéma d'origine de la loupe et le peu de composants nécessaires aux deux interfaces sérielles additionnelles. Pour la liaison à l'enregistreur USB, K3 est une embase à 2 × 5 picots au pas de 2,54 mm, de préférence munie d'une protection latérale. Elle se soude simplement sur un morceau de plaque perforée. Il faut savoir que l'alimentation secteur de la loupe n'est pas assez puissante pour y bran-

Figure 1.
Le prototype de la loupe de fréquence pour réseau de distribution électrique de janvier 2012. Un module compact avec affichage à LED.

Figure 2.
Le programme *Netzfrequenz_P* vous permet de visualiser graphiquement les données enregistrées de la fréquence du réseau.



cher encore l'enregistreur de données sérielles ainsi que la clé USB. C'est pourquoi les broches 1 (+5 V) et 2 (masse) de K3 ne sont pas reliées aux conducteurs correspondants du circuit imprimé de la loupe. On utilise un autre bloc secteur de 5 V, capable de délivrer 0,5 A, à brancher sur une prise adéquate reliée à ces broches. Sur K3, on connecte alors un câble plat à 10 conducteurs terminé sur une prise sortie correspondante. L'enregistreur obtient du même coup ses données et sa propre alimentation ainsi que celle de la clé. Si vous désirez garder la possibilité de brancher aussi la loupe sur l'interface sérielle d'un PC, ajoutez, sur la plaque perforée K2, un connecteur Sub-D à 9 contacts. Un petit inverseur bon marché permettra de fournir les données « dans le bon sens » au PC pour un décodage conve-

réaliser d'autres modifications, il vous faudra le compilateur C de la firme Wickenhäuser [4] dont la version gratuite de démonstration suffit à cette tâche.

Le micrologiciel offre différentes fonctions. D'abord, il mesure la durée de 50 (ou 60, le cas échéant) périodes de la fréquence du secteur et en communique une valeur de mesure. On obtient de la sorte une donnée toutes les 50 périodes, donc à peu près toutes les secondes.

Si vous connaissez l'heure de début de l'enregistrement, vous n'avez pas besoin d'horodatage pour savoir à quel moment correspond la fréquence du réseau enregistrée. Comme les gestionnaires maintiennent avec grande précision la **moyenne** de la fréquence du courant, même après plusieurs jours d'enregistrement, avec le calcul du temps d'observation, vous connaîtrez toujours exactement l'heure d'échantillonnage. C'est le principe de l'horloge synchrone.

On reçoit donc chaque seconde par l'interface sérielle une valeur de mesure, soit du PC, soit de l'enregistreur. Mon logiciel utilisateur se sert des données pour fournir les fréquences et en tracer le graphique. Le logiciel stipule comme paramètres de l'interface 4 800 bauds, pas de parité et un bit d'arrêt. Si l'enregistreur USB est bien configuré, il trouvera au moment de l'insertion de la clé USB un texte qui contient l'entrée COM_BAUDRATE: 4800. On le réalise aisément avec l'éditeur de texte de Windows ou en utilisant dans le téléchargement [3] l'exemplaire de données « config.txt ». Pour le reste, vous pouvez utiliser, tel quel, l'enregistreur de données sérielles.

Programme pour PC

Mon programme pour PC peut recevoir en direct les données de la loupe de fréquence du réseau par l'un de ses ports COM, mais il peut aussi lire et présenter les données contenues dans un fichier provenant par exemple de l'enregistreur USB. C'est pourquoi le programme dispose de deux modes de fonctionnement (cf. figure 2) à sélectionner en bas, à droite de l'écran. Avec *XMax*, on peut régler la durée en heures et avec *Offset* l'origine de l'axe des X par rapport au temps initial. Avec *Bmp speichern*, vous pouvez photographier le graphique affiché de l'évolution de la fréquence sous forme de fichier image.



Figure 3.
Le prototype de l'enregistreur de données sérielles de janvier 2012. Le microcontrôleur enregistre sur une simple clé USB les données en provenance de l'interface sérielle.

nable. Comme inverseur et tampon de sortie vers l'interface sérielle du PC, avec un débit de 4 800 bauds, un petit MOSFET (T1) avec résistance de polarisation haute (R4) fera l'affaire. S'il faut par exemple y brancher un long câble, mieux vaut alors éviter les soucis en embauchant un vrai pilote d'interface tel que le circuit intégré MAX232.

Modification du logiciel

La minime modification du matériel effectuée, il s'agit de s'attaquer au micrologiciel du contrôleur de la loupe de fréquence pour lui faire débiter les données par l'interface sérielle. J'ai bien sûr conservé le logiciel originel de Dieter Laues [1] qui commande à merveille les LED pour afficher la fréquence actuelle du secteur. Il va de soi que le code source et le fichier hexadécimal pour le microcontrôleur sont à votre disposition gratuitement sur le site d'Elektor [3]. Si vous voulez

Une installation en bonne et due forme du logiciel n'est pas nécessaire. Il suffit de mettre l'application *Netzfrequenz_P.exe* et le fichier *Netzfrequenz.ini* dans le même dossier. Tous les réglages du programme sont présents dans le fichier *.ini* de sorte que lors d'un lancement ultérieur, on retrouvera par exemple les mêmes dimensions de fenêtre qu'à la fermeture.

Si vous voulez intervenir dans le programme et mouliner à la hâte le résultat au compilateur, il faut savoir deux choses.

D'abord, qu'il s'agit d'un programme Delphi, et qu'ensuite pour l'exploiter, vous aurez besoin, entre autres, d'avoir installé au préalable *Com- mRec.dpk* pour l'interface série ; il se trouve

dans l'archive *ASYN32.ZIP* dans le dossier du même nom du téléchargement [3].

Mode Werte aus Datei lesen

C'est le mode de **lecture des valeurs** dans un fichier, lesquelles ont été écrites par l'enregistreur USB dans un fichier qui porte toujours le nom de *LOGGING.TXT* sur la clé. Il est donc recommandé d'en modifier le nom pour lui donner du sens, par exemple *fedf_20131214_120608*, dans lequel on reconnaît le format de date *aaaammjj* et l'heure de début d'enregistrement *hhmmss*. On ouvre le fichier choisi d'un clic sur *Start*. Vous pouvez saisir vous-même la date et l'heure dans les champs appropriés au bas de la fenêtre du programme. Vous pouvez également éditer le fichier avant l'importation pour y inscrire la date

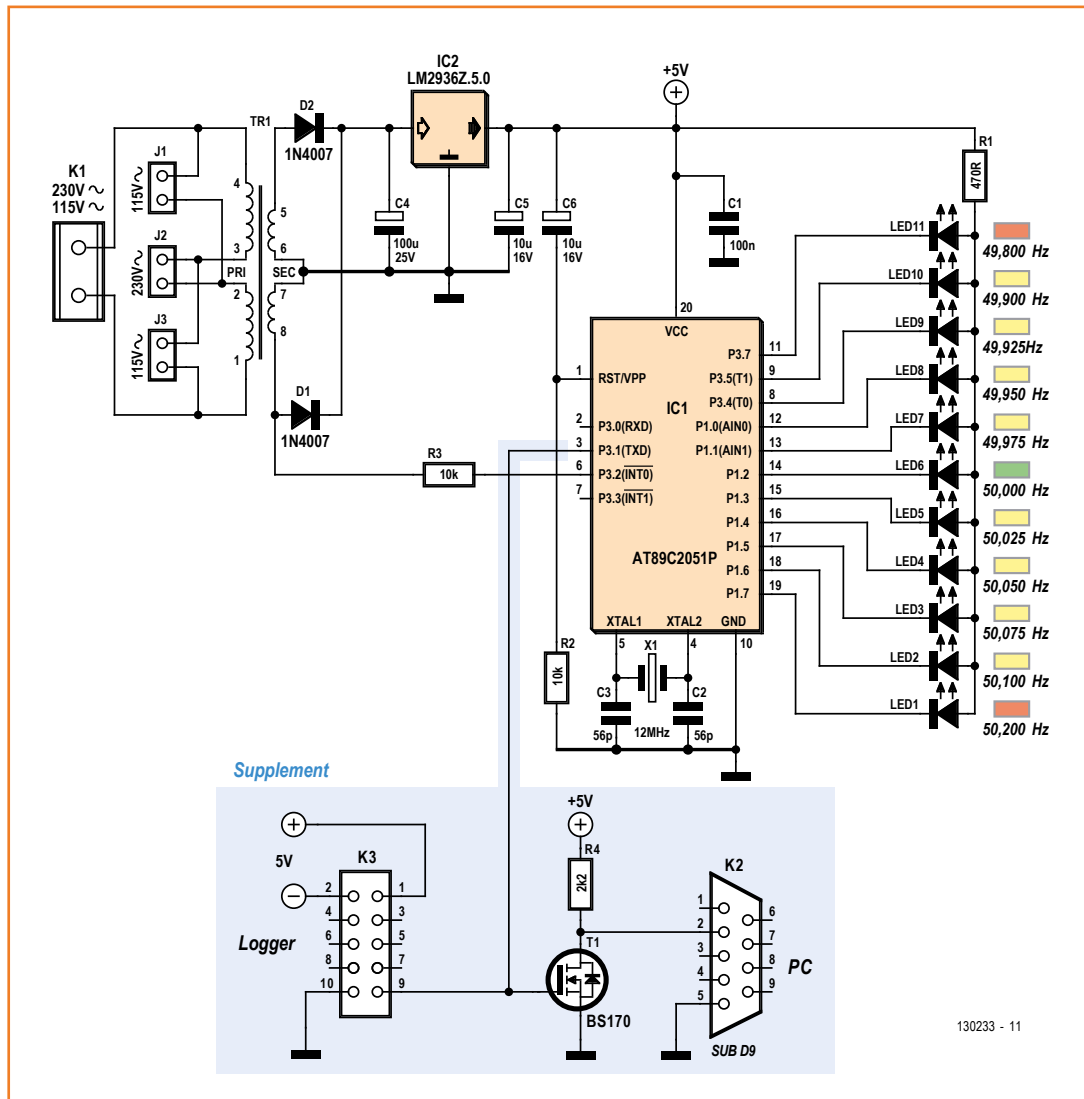


Figure 4.
L'extension à adjoindre au circuit de la loupe de réseau se limite à deux interfaces sérieles et très peu de composants.

130233 - 11

Fichier journal

La première donnée du fichier journal est souvent fragmentaire. On la supprime simplement, la première seconde d'enregistrement n'a pas une importance capitale. Comme le fichier journal est rédigé en pur ASCII, n'importe quel éditeur de texte convient.

Si vous ne désirez pas à chaque appel réécrire la date et l'heure, vous pouvez les inscrire à la place de la donnée altérée en tête du fichier. Les valeurs de fréquence sont exprimées dans le fichier en périodes au format d'entiers avec une résolution de 1 μ s. Donc, 50 Hz correspondent à une durée de 20 000 μ s. Un fichier tout droit sorti de la clé USB commence par exemple comme ceci :

9998
20001
20000
19998
...

remplacer la première ligne par la date et l'heure donne par exemple :

14.12.2013 12:06:08
20001
20000
19998
...

Le téléchargement du logiciel [3] en contient un exemple.

et l'heure en tête des données, voyez l'**encadré** Fichier journal.

Mode *Sérielle Schnittstelle*

Le mode **interface sérielle** permet de lire les données issues d'un port sélectionné (port COM de 1 à 9) et en même temps de les enregistrer sur le PC dans un fichier tel que fedf_20131214_120608, de quoi se souvenir plus tard de son contenu. L'enregistrement démarre d'un clic sur *Start*.

Conclusion

Tout compte fait, j'ai bien résisté à la tentation de réinventer la roue ! Il existait deux projets éprouvés d'Elektor, avec circuit imprimé et logiciel, il me restait à les combiner harmonieusement. Une addition qui vaut plus que la somme de ses termes, puisque le matériel d'enregistrement de la fréquence du secteur travaille en parfaite autonomie et laisse toute liberté d'évaluer plus

tard, sur une plus longue période, la qualité globale du courant fourni par le secteur.

Le logiciel est disponible en version 50 Hz et 60 Hz. Comme il est à code source ouvert, chacun aura le loisir d'y apporter de légères modifications. Je peux suggérer par exemple l'intégration d'une horloge électronique pour gérer automatiquement un horodatage. Si vous êtes intéressé, vous pouvez vous inspirer du circuit *Garde-temps* [5] paru dans le magazine Elektor de juillet/août 2011

(130233 – version française : Robert Grignard)

Liens

- [1] Le réseau EDF à la loupe, Elektor 1/2012 : www.elektor-magazine.fr/110461
- [2] Clé USB sur microcontrôleur, enregistreur de données sérielles, Elektor 1/2012 : www.elektor-magazine.fr/110409
- [3] Logiciel et code source : www.elektor-magazine.fr/130233 :
- [4] C-Compiler-Demo : www.wickenhaeuser.de
- [5] Garde-temps Elektor 7/8 2011 p. 49 : www.elektor-magazine.fr/110285

imprimante 3D elektor

Assez joué, passons aux choses sérieuses !

Elektor s'associe à *German RepRap*, le spécialiste de l'impression 3D, pour vous proposer ce modèle unique d'imprimante 3D sous la forme d'un kit soigneusement élaboré pour vous.

La nouvelle imprimante 3D (aux couleurs d'Elektor) donne d'excellents résultats et s'assemble beaucoup plus rapidement que les modèles courants à construire soi-même.

La présence de connecteurs pour les moteurs pas à pas et les cartes électroniques, ainsi que le châssis en acier inoxydable, robuste et stable, vous faciliteront le montage. Il ne reste quasiment plus rien à souder.

L'imprimante Elektor 3D est fournie avec un éventail d'accessoires en bonus (d'une valeur d'environ 380 €).

Caractéristiques techniques :

- dimensions (L x l x h) : 500 x 460 x 460 mm ;
- volume d'impression (X x Y x Z) : 230 x 230 x 125 mm ;
- vitesse : 100 mm/s pour les matériaux de 3 mm, jusqu'à 180 mm/s pour les matériaux de 1,75 mm, jusqu'à 350 mm/s à vide ;
- tension : 230/115 V CA ;
- matériaux : ABS / PLA / PS / PVA / Laywood (filament composé de bois) / Laybrick.

NOUVEAU!



Contenu de la livraison :

- matières plastiques (polyamide, particulièrement résistant et résilient) ;
- châssis en acier inoxydable (laque bleue) avec pieds réglables, vis, tiges filetées, arbres, douilles à billes linéaires, palier lisse, palier à billes ;
- courroies d'entraînement T 2.5 et poulies fraisées ;
- 5x moteurs pas à pas NEMA17 avec un couple moteur de 0,52 Nm ;
- carte RAMPS v1.4, y compris câble et alimentation ;
- tête de chauffe (hot end) modulable de 3 mm avec buse de 0,5 mm (buse de 0,4 / 0,3 mm en option).

Accessoires fournis en bonus :

- plateau d'impression chauffé 12 V ;
- 1x fil de plastique PLA de 750 g (rouge, Ø 3 mm) ;
- 1x fil de plastique PLA de 750 g (jaune-vert, Ø 3 mm) ;
- 1x fil de plastique PLA de 750 g (bleu, Ø 3 mm) ;
- logiciel sur CD, y compris le code d'impression pour un objet ;
- châssis bleu en acier inoxydable sur pieds amortisseurs.

Prix : 1.199,00 €
(TVA incluse)

Frais de port pour France Métropolitaine et Belgique : 16 €

Infos et commande : www.elektor.fr/imprimante3d

sondes : chassez le crocodile

Alfred Rosenkränzer
(Allemagne)

La mesure de signaux numériques rapides est souvent faussée par la sonde utilisée. Une liaison à la masse très courte a toute son importance. Avec une sonde ordinaire, à pince crocodile pour la masse, vous constaterez des dépassements de l'amplitude, étrangers au signal. Les sondes chères peuvent être munies d'un ressort de masse adaptable qui raccourcit la liaison. Un tel ressort, vous pouvez le fabriquer vous-même !

Mesurer à l'aide d'un oscilloscope, c'est la routine pour un électronicien. Les signaux issus d'un générateur sont normalement acheminés par

câble blindé, la plupart du temps muni de fiches BNC. En basse fréquence, avec les signaux audio par exemple, le câble n'est pas nécessairement bouclé : il n'est pas terminé par une impédance spécifique en correspondance avec la caractéristique du coaxial. En haute fréquence, avec de la vidéo ou des signaux numériques rapides, on boucle généralement le câble, la meilleure manière consiste à utiliser la résistance de terminaison commutable à l'entrée de l'oscilloscope, s'il y en a une.

Mais très souvent, on se sert de sondes de mesure à haute impédance, c'est le seul moyen de suivre un signal au long de son trajet dans un circuit. Les sondes communes présentent une impédance d'entrée de $10\text{ M}\Omega$ en parallèle avec une capacité d'une dizaine de picofarads. Avec la résistance d'entrée de $1\text{ M}\Omega$ de l'appareil, cela forme un diviseur de tension de rapport 10 : 1. Le signal est prélevé par la pointe de la sonde. La nécessaire liaison à la masse passe par un fil d'une dizaine de centimètres terminé par une pince crocodile. Sans oublier d'accorder au préalable la sonde en la branchant sur la sortie de test !

Quand on mesure des impulsions numériques rapides, on s'aperçoit vite que la forme d'onde observée peut varier en fonction de la conformation du fil de masse. C'est surtout visible comme dépassement du signal après un flanc. Il faut se demander si cette déformation fait vraiment partie du signal à mesurer (**fig. 1**).

Sur les sondes de mesure professionnelles comme chez *Agilent* ou *Tektronix*, on améliore la forme du signal présent en ôtant le fil de masse pour le remplacer par un ressort à adapter (**fig. 2 et 3**).



Figure 1.
Un signal rectangulaire à 10 MHz avec câble et terminaison de $50\ \Omega$ dans l'oscilloscope (A) : le même mesuré avec une sonde et fil de masse sur la terminaison (B).



Figure 2.
Sur cette sonde de mesure de la firme *Agilent*, le fil de masse a été remplacé par un ressort.

Bien sûr, il ne suffit plus d'appuyer la pointe sur la borne de test, il s'agit de trouver dans le voisinage un point de masse adéquat pour établir le contact. Je n'avais jamais vu ce genre de système sur des instruments de prix modéré. Par chance, toutes les sondes sont équipées d'un anneau de masse, juste derrière la pointe, pour donner contact dans la sortie de test. On peut l'entourer de quelques tours de fil dénudé et le tordre pour en faire quelque chose de similaire (**fig. 4**). À ce prix-là, on peut même envisager de souder ce raccord de masse dans le circuit ! Et si on l'a bousillé, on s'en refait un autre.

Largeur de bande

N'empêche, ces manipulations ont souvent pour effet de rétrécir la bande passante de la mesure. Pour s'en rendre compte, on peut mesurer un signal rectangulaire rapide, d'abord par un câble à BNC avec terminaison directe, puis avec la sonde. S'il apparaît plus grand avec la sonde, c'est que l'oscilloscope a encore de la réserve. Dans le secteur professionnel, il existe aussi une solution pour cela : des sondes de mesure d'une impédance d'entrée de 500 ou 1 000 Ω . De prime abord, cela semble bien peu en comparaison des sondes de 10 M Ω , mais en haute fréquence, c'est la capacité d'entrée qui domine. En outre, les signaux les plus rapides se situent aux alentours de 1 V_{pp} et les puces peuvent facilement fournir le courant nécessaire.

Le prix d'une de ces sondes est cependant de l'ordre de quelques centaines d'euros, inabordable pour la plupart des électroniciens. Mais on peut assez facilement en réaliser une soi-même avec des caractéristiques comparables.

Prenez un câble de mesure de 50 Ω avec fiche BNC d'un côté. Si nécessaire, coupez en deux un câble de 2 m et faites-en deux sondes. Dénudez l'extrémité coupée et soudez sur l'âme une résistance que vous isolerez et stabiliserez avec un morceau de gaine rétractable. Tordez la tresse et soudez-y un morceau de fil pour la connexion sur la platine. Un autre morceau de gaine rétractable par-dessus et l'affaire est faite (**figure 5**). Comme valeur de résistance, 450 ou 950 Ω conviendront bien : avec la résistance de bouclage de 50 Ω de l'oscilloscope, on obtient un diviseur par 10 ou par 20. On trouve de telles résistances dans le commerce de l'électronique. Si vous acceptez

que le rapport ne soit pas exact, vous pouvez prendre 470 ou 1 000 Ω . Si quelque chose se casse, coupez la fin du câble et construisez une nouvelle sonde.

La **figure 6** montre une sonde à basse impédance de HP. Dans la gaine en plastique blanc, il y a une résistance que l'on peut échanger. La connexion à l'oscilloscope se fait par fiche SMA et un câble SMA monté, son prix est à couper le souffle. On peut aussi construire une imitation avec une prise SMA, une résistance, du câble et de la gaine rétractable (à condition de disposer d'un câble SMA et un adaptateur pour BNC).

(130112 - version française : Robert Grignard)

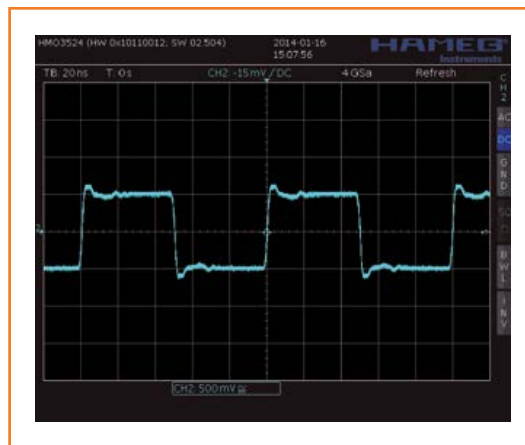


Figure 3.
Le signal rectangulaire à 10 MHz mesuré avec la sonde équipée d'un ressort pour un contact direct avec la masse.



Figure 4.
Une sonde de marque Hameg avec un ressort de masse fait maison pour remplacer le fil à pince crocodile.



Figure 5.
La sonde de mesure à basse impédance que j'ai fabriquée.



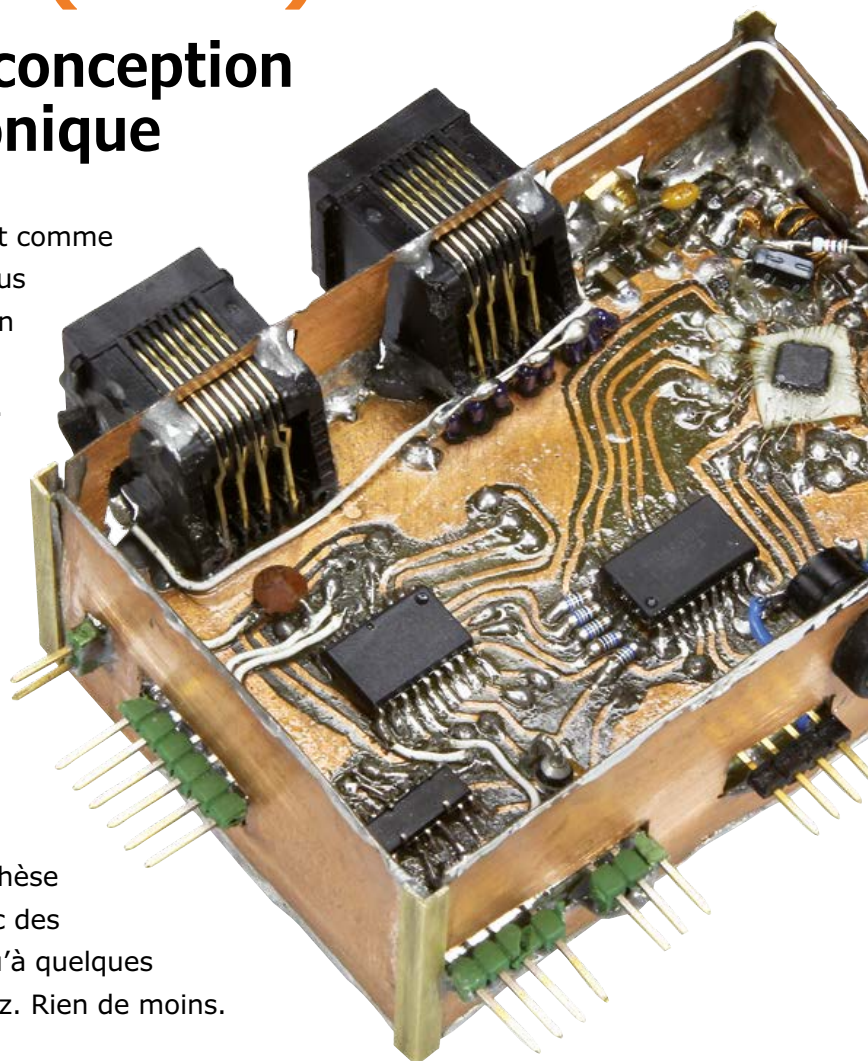
Figure 6.
Une ancienne sonde à basse impédance de HP.

générateur HF à synthèse numérique directe (DDS)

une autre conception de l'électronique

Wim Knoeff
PA3EHN (Pays-Bas)

Ceci n'est pas un projet comme ceux auxquels nous vous avons habitués, avec un circuit imprimé élégant et testé complètement. Dans cet article, nous voulons montrer qu'avec des outils simples et ordinaires, rien de plus, il est possible de construire un circuit à haute fréquence au fonctionnement impeccable. Il s'agit ici d'un générateur à synthèse numérique directe avec des signaux de sortie jusqu'à quelques centaines de mégahertz. Rien de moins.



Elektor publie régulièrement des projets de lecteurs. Ce sont souvent des circuits dont la conception ou l'exécution particulière nous ont plu, mais qu'il ne nous paraît pas sensé de faire remettre en forme par la grosse machine du labo d'Elektor. Le générateur décrit ici entre dans cette catégorie, mais pour de tout autres raisons. La première fois que ce générateur DDS nous est tombé sous les yeux, nous n'avons pas cru qu'il pouvait fonctionner effectivement. L'auteur

a attaqué la question sous un angle résolument personnel, sans utiliser aucun des programmes, usuels aujourd'hui, de dessin de schéma ni de circuit imprimé. Tout est fait à la main, le PC n'est intervenu que pour le travail de programmation. Les concepteurs d'aujourd'hui peuvent trouver ces méthodes dépassées, mais nous les qualifierons plutôt de « non conventionnelles » ou « novatrices ». Cet article décrit la conception et la construction d'un générateur à haute fré-

quence à synthèse numérique directe, avec une plage de sortie de 100 MHz et différentes caractéristiques annexes, comme modulation d'amplitude, fréquence divisée, balayage sur 10 ou 50 kHz et sélection automatique des filtres de bandes. Le projet dans cette forme n'est pas un circuit que tout un chacun peut reproduire de chic, mais une inspiration pour une *autre façon* d'aborder l'électronique !

Schéma

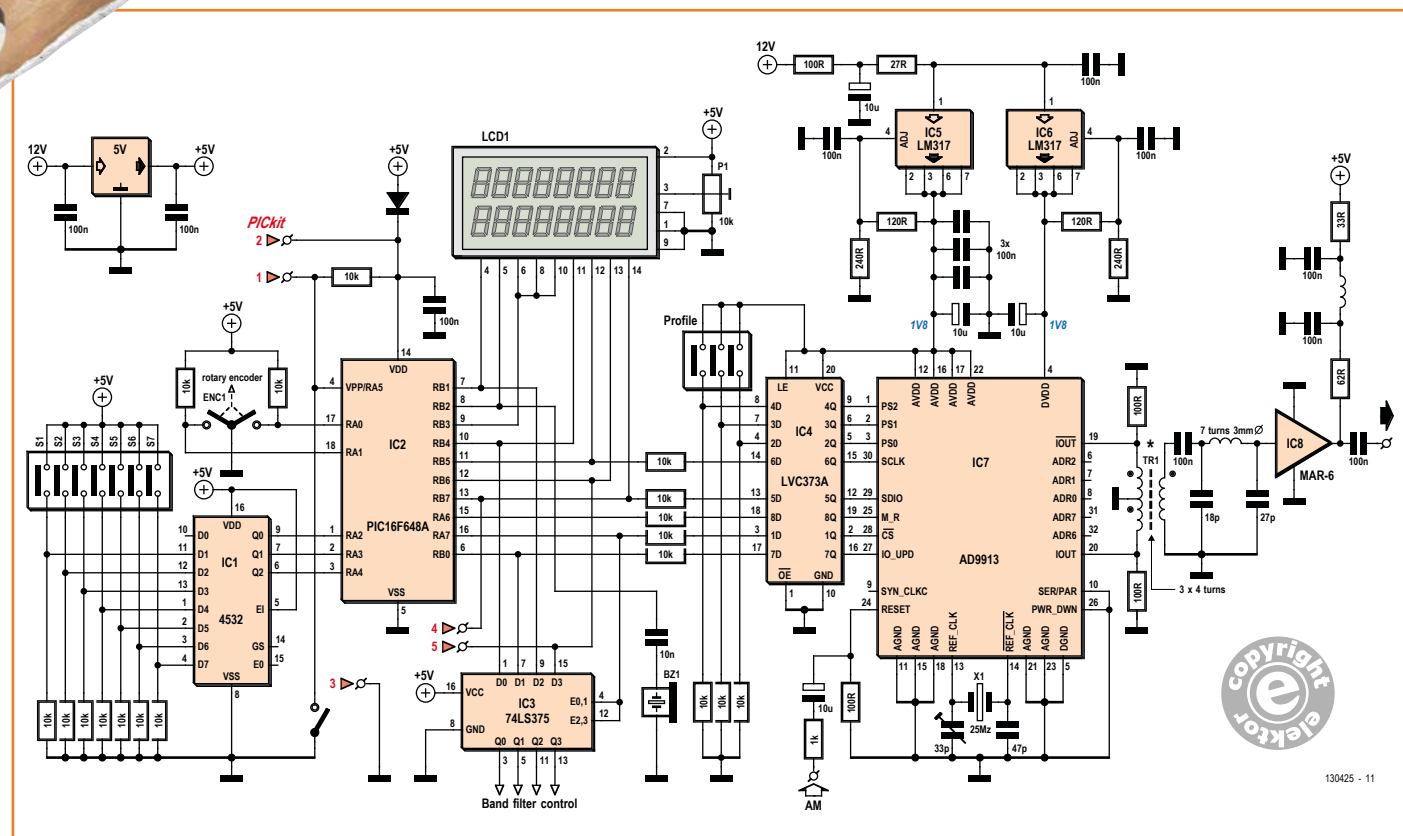
Le circuit intégré AD9913 [1] est destiné à la synthèse numérique directe (*Digital Direct Synthesis*) avec une faible consommation pour l'utilisation dans des appareils portatifs. Son convertisseur numérique-analogique à 10 bits produit 250 millions d'échantillons par seconde. Grâce à ce CN/A rapide et à une haute vitesse de traitement interne, l'AD9913 peut délivrer à sa sortie analogique des sinusoïdes à 100 MHz. L'AD9913 offre aussi un saut en fréquence (*frequency hop-*

ping) et un accord fin rapides, de même qu'une régulation précise du décalage de phase.

Le signal de l'oscillateur de l'AD9913 – la fréquence d'horloge de référence – est produit par un quartz de 25 MHz connecté aux broches 13 et 14. Le circuit oscillateur lui-même est situé dans la puce DDS ; le réglage fin de la fréquence se fait avec un condensateur ajustable (**fig. 1**). La fréquence du signal de référence est multipliée par une PLL interne (*Phase Locked Loop*, boucle à verrouillage de phase) et élevée jusqu'à la fréquence d'horloge utilisée par la puce DDS. Les touches de commande S1 à S7 assurent les réglages souhaités, par un codeur de priorité à 8 entrées (IC1) associé à un codeur rotatif connecté aux broches 17 et 18 du microcontrôleur IC2 (un PIC 16F648A) ; ce dernier délivre les séquences de bits nécessaires pour régler la puce DDS, et aussi les données pour l'afficheur LCD à deux lignes. Les données pour la puce DDS transitent par IC4 (74LVC373A, un octuple verrou avec entrées compatibles 5 V) vers l'AD9913. Les signaux d'IC2 sous 5 V se trouvent adaptés aux entrées du CI DDS qui fonctionnent sous 1,8 V. L'un des signaux d'IC2 (RA7 du PIC, CS pour le

Figure 1.

Le schéma de base du générateur DDS, redessiné en style Elektor. Le circuit AM accessoire n'est pas représenté ici. L'œuvre d'art originale, dans le style *Si tous les gars du monde*, est à télécharger [4].



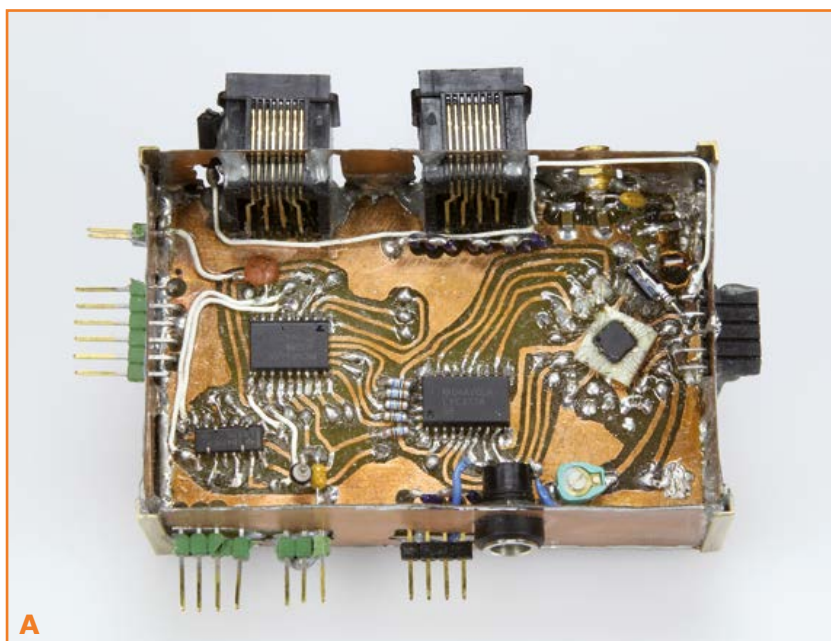
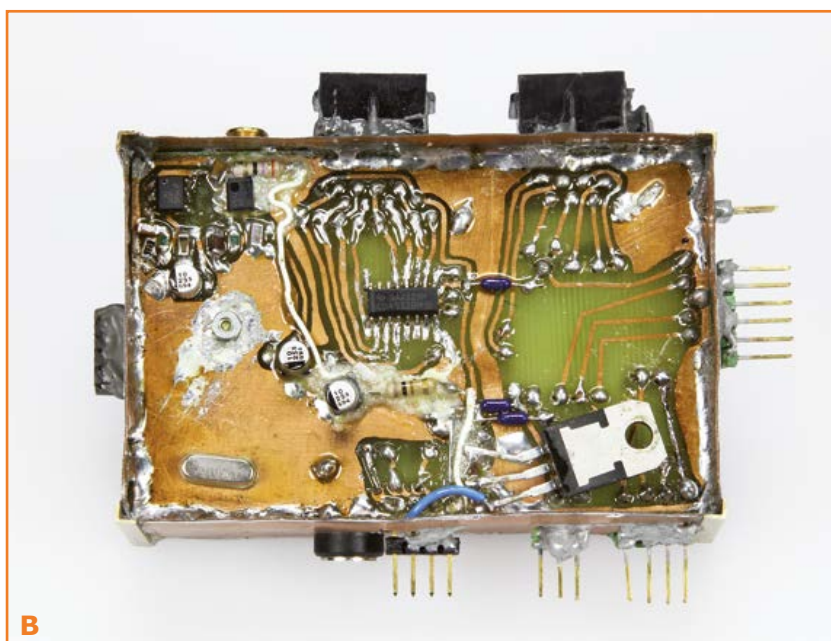


Figure 2.
Faces supérieure (A)
et inférieure (B) de la
réalisation de l'auteur.

DDS) commande aussi la lecture des données du filtre de bande dans IC3 (une quadruple bascule D 74LS375).

L'alimentation se compose d'une partie 5 V pour le microcontrôleur et ses périphériques, et d'une partie 1,8 V pour le DDS. Cette partie 1,8 V se partage en plus en une ligne analogique et une ligne numérique, avec deux régulateurs séparés. Les deux sorties complémentaires du DDS (bro-



ches 19 et 20) sont combinées dans un petit transformateur torique bobiné trois fils en main (entre autres pour éliminer les harmoniques pairs, voir la note d'application AN-912 d'Analog Devices [2]) et parviennent à la sortie à travers un filtre en pi et le tampon de sortie IC8 (MAR6). C'est là qu'est raccordé le connecteur de sortie.

Construction

L'auteur a dessiné à la main pour ce montage un circuit imprimé double face (fig. 2 et 3) et il témoigne par là qu'on peut construire plutôt simplement des circuits complexes (même avec des CMS) et pourtant fiables. L'article *Les puces LFCSP soudées à la main*, dans le numéro de janvier-février 2014 [3], explique comment souder la puce DDS sur le circuit imprimé à l'aide de petits morceaux de fils.

Pour les amateurs, les dessins de ce projet sont disponibles en téléchargement depuis la page *ad hoc* du site Elektor [4], nous ne donnons ici que le résultat final. L'aspect est bien différent d'un projet Elektor « normal », mais il fonctionne à la perfection, y compris en haute fréquence.

Nous décrivons brièvement ici la suite des opérations pour illustrer la méthode. On commence par les liaisons entre les deux faces de la platine, viennent ensuite les puces d'alimentation et tout ce qui les entoure (le régulateur 5 V est monté à l'envers), les connecteurs, les connexions latérales, le HEF4538 sur la face inférieure, le 74LS375 avec le 16F648A sur la face supérieure, le MAR6 avec tous les composants associés autour du DDS, puis quelques liaisons par fils. Tous les circuits intégrés se montent en surface, les autres composants ne sont pas critiques.

Pour le raccordement de l'écran LCD et des sept interrupteurs (NO), l'auteur a choisi deux nappes à huit conducteurs avec des connecteurs RJ45 pour simplifier la déconnexion et l'échange. La ligne d'alimentation +5 V de l'écran est un petit fil séparé avec une fiche miniature à côté du connecteur RJ45 du LCD. Le dossier à télécharger qui accompagne cet article [4] contient, en plus des deux faces de la platine, le schéma de câblage complet.

Ensuite le fichier HEX du même dossier [4] peut être chargé dans le 16F648A pour contrôle (voir le paragraphe *Logiciel*). Après cette vérification, le 74LVC373 est soudé et les signaux de pilotage peuvent être envoyés vers le DDS et vérifiés sur les îlots de connexion.

Montage de l'AD9913

Enfin le circuit intégré DDS, préparé selon [3] avec ses fils neufs et sa vis de masse, vient prendre place sur la platine. Tout d'abord les plages de masse des faces supérieure et inférieure de la platine sont reliées par un ruban de cuivre soudé dessus et dessous, enfilé à travers le trou de 3 mm du fil de masse. Ensuite la vis de masse de la puce DDS est poussée dans le même trou que le ruban de cuivre, avec les petits fils placés autant que possible au-dessus de leurs îlots de connexion respectifs. Ensuite un écrou vient prendre place sur la vis de masse en dessous de la platine. Conseil : ne plus jamais toucher à cet écrou ! Enduire de graisse thermique l'écrou, le ruban de cuivre et le circuit imprimé.

Enfin les fils de connexion du DDS sont raccourcis et soudés en place ; utilisez pour cela une loupe lumineuse et un bon fer à souder mis à la terre. Les broches 6, 7, 8, 9, 31 et 32 restent sans connexion.

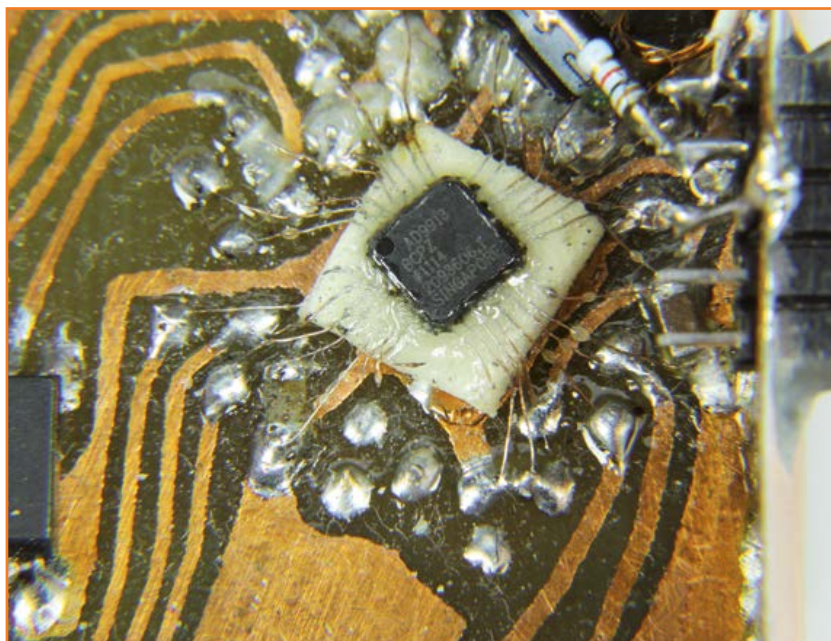
Après l'application de la tension d'alimentation de 12 V, il y a naturellement un moment d'émotion avant de savoir si tout est bien installé et connecté ; chez l'auteur, les trois prototypes ont fonctionné aussitôt ! Ces prototypes sont recouverts de vernis électrofuge incolore, ce qui assure la fixation des CMS sur la platine et leur donne une protection durable contre la corrosion.

Logiciel

De nombreux auteurs ont travaillé pendant des années sur le code source (.ASM) de ce générateur. L'intention est de garder ce code aussi transparent que possible, de façon à en faire un tremplin pour d'autres développements. Le fichier s'ouvre dans Notepad. On peut alors par exemple programmer une fréquence intermédiaire à soustraire (de préférence fixer la fréquence du DDS au-dessus de celle du LCD) ou modifier la fréquence de sortie maximale.

Chaque code source (modifié) doit être converti en un fichier HEX avec MPASM [5]. Ensuite, ce fichier HEX est chargé dans le PIC 16F648A avec un programmeur (p.ex. un PicKit2) relié au connecteur latéral « PICKit ». Pendant l'opération, l'alimentation et le câble du LCD doivent être déconnectés.

Après la mise sous tension apparaissent à l'écran le multiplicateur de la PLL et la fréquence de sortie maximale du logiciel chargé, pendant que le



buzzer émet un petit son de bienvenue ; la fréquence de départ (28,5 MHz) apparaît ensuite sur la ligne 1 du LCD, après action sur le codeur elle apparaît aussi sur la ligne 2.

Différentes versions de logiciel se trouvent dans le fichier à télécharger [4]. Elles sont nommées (code source et exécutable HEX) par le facteur de la PLL et la fréquence de sortie maximale (p. ex. « 10x_120mc »).

Ce projet montre qu'il existe, à côté des procédés standard, des façons originales de réaliser un projet électronique. Le dossier à télécharger associé à cet article contient en plus une courte description du mode d'emploi, du fonctionnement en fréquence divisée et des possibilités de modulation en amplitude.

(130425 – version française : Jean-Paul Brodier)

Figure 3.

Le montage de la puce DDS sur la platine. Les travaux préparatoires à effectuer sur le circuit intégré ont été décrits dans le numéro de janvier-février [3].

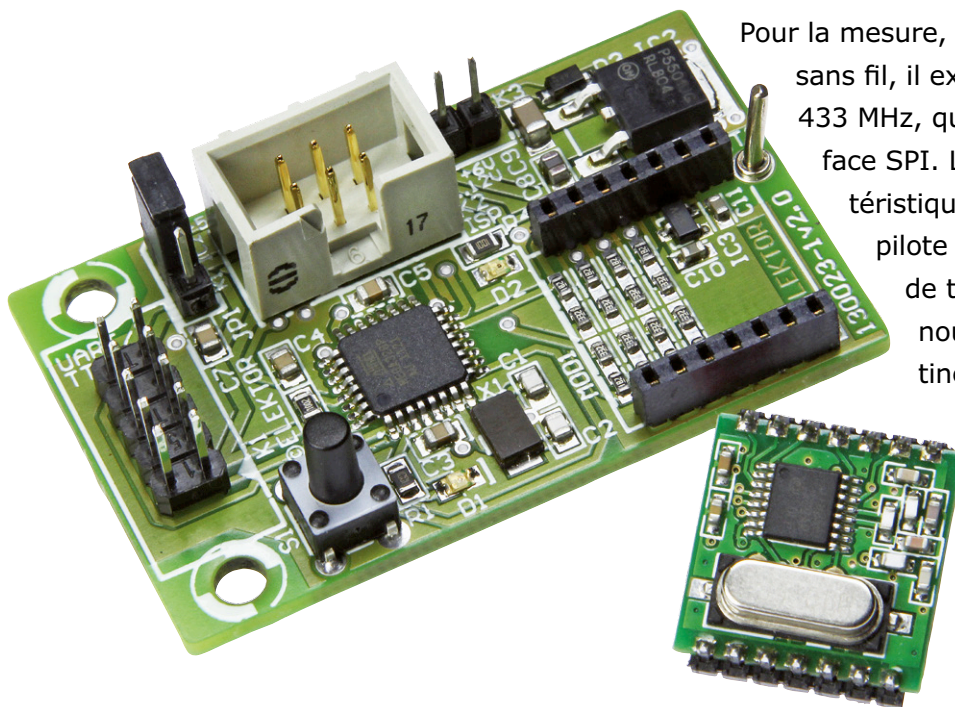
Liens ouèbe

- [1] www.analog.com/static/imported-files/data_sheets/AD9913.pdf
- [2] www.analog.com/static/imported-files/application_notes/AN_912.pdf
- [3] www.elektor-magazine.fr/120246
- [4] www.elektor-magazine.fr/130425
- [5] www.microchip.com/mplab

passerelles radio

De l'UART aux 433 MHz et retour

Jens Nickel (elektor)



Pour la mesure, la commande et la régulation sans fil, il existe de petits modules radio sur 433 MHz, qui communiquent par une interface SPI. L'étude de la feuille de caractéristiques et la programmation d'un pilote ne sont toutefois pas l'affaire de tout un chacun. C'est pourquoi nous avons conçu de petites platines qui portent, en plus du module 433 MHz, un ATmega328 et une interface UART. Un logiciel enfoui fait de ces platines des passerelles d'utilisation simple, pour envoyer le train de signaux de l'UART dans l'éther et retour.

Le fabricant chinois HopeRF produit de petits modules radio (alimentés sous 3,3) qui mettent en œuvre le CI Si4421 de Silicon Labs [1], avec un quartz et le câblage annexe (**figure 1**). Ce circuit intégré permet à un concepteur de micrologiciel expérimenté de réaliser assez simplement des applications de transfert de données sans fil. Le transcepteur – qu'on appelait naguère émetteur-récepteur – obéit à une interface SPI. Des suites d'octets déterminées servent de commandes pour commuter le CI entre émission et réception, fixer le débit de données, et d'autres choses encore. C'est par la même interface SPI que transitent les données à émettre et les données reçues.

Un concepteur qui possède une certaine expérience et dispose d'assez de temps peut adjoindre à son application un pilote SPI et asseoir là-dessus une petite bibliothèque Si4421 qui contiendra les octets de commande de la puce. Dans

le cas le plus simple, cette bibliothèque contient une fonction, que l'application peut appeler, pour l'émission par radio d'une suite d'octets. Une autre fonction pourrait par exemple inscrire dans un tampon circulaire les caractères reçus par radio, pour qu'ils puissent être traités par le programme d'application.

Pelle à caractères

On peut trouver sur l'internet aussi bien des pilotes SPI pour des microcontrôleurs spécifiques que des bibliothèques Si4421, mais il faudrait encore passer du temps à les adapter à un micrologiciel et à une application définis. Pour ce projet, nous allons emprunter une autre voie, celle que FTDI, par exemple, a déjà ouverte avec un grand succès dans le domaine de l'USB. Les CI convertisseurs USB/TTL de ce fabricant écossais sont appréciés parce qu'ils peuvent se connecter simplement à l'interface UART d'un microcontrô-

leur. Le programmeur de l'application ne s'occupe que de l'émission et de la réception de caractères par l'UART, sans se soucier de la programmation de l'USB ou d'un pilote particulier.

Nous avons conçu une petite carte où s'enfiche un module radio HopeRF. Sur la carte se trouvent, en plus de l'alimentation, un ATmega328 et une barrette à deux fois cinq broches par où ressortent les signaux RX/TX de l'UART (fig. 2). Le brochage de cette prise a été présenté dans le dernier numéro Elektor [2] ; vous trouverez dans l'encadré des détails sur l'ECC, *Embedded Communication Connector*. Le micrologiciel enfoui du µC à 8 bits se charge du pilotage du module radio. La carte est fournie avec un exemple de micrologiciel qui en fait une passerelle entre UART et émetteur radio. Une chaîne de caractères, 62 au maximum, terminée par un retour chariot (<CR>), transmise par l'UART à la passerelle, sera envoyée par radio. Dans l'autre sens, les chaînes de caractères reçues par radio sont transmises à l'UART. Le débit des deux canaux est de 9600 bauds.

Les cartes passerelles sont basées sur un projet de Günter Gerold, qui avait développé son module pour des applications de télémétrie sur « Wheelie ». Le laboratoire d'Elektor a créé une platine disponible aussi bien nue que câblée [3]. Le microcontrôleur de la version câblée est programmé avec le micrologiciel de la passerelle, mais il va de soi qu'on peut y « flasher » son propre logiciel.

Schéma

Le schéma (fig. 3) n'est pas excessivement compliqué. Le composant central est le microcontrôleur ATmega328 qui peut recevoir un nouveau programme par ISP via le connecteur K2. Le câblage standard (quartz, alimentation) est également conforme à la feuille de caractéristiques.

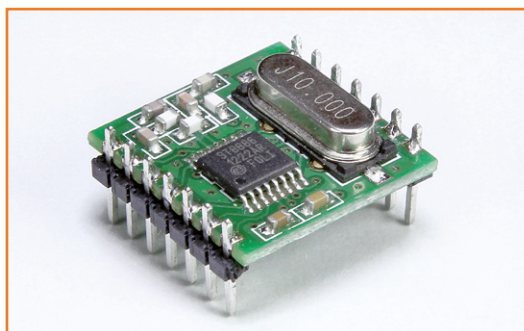


Figure 1.
Les modules radio à 433 MHz du fabricant chinois HopeRF sont pilotés par SPI (ici avec les barrettes soudées).

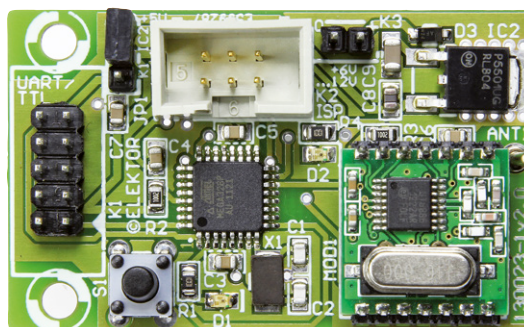


Figure 2.
La platine de la passerelle avec l'ATmega328, une touche, une LED et le connecteur à 2x5 broches pour les signaux de l'UART.

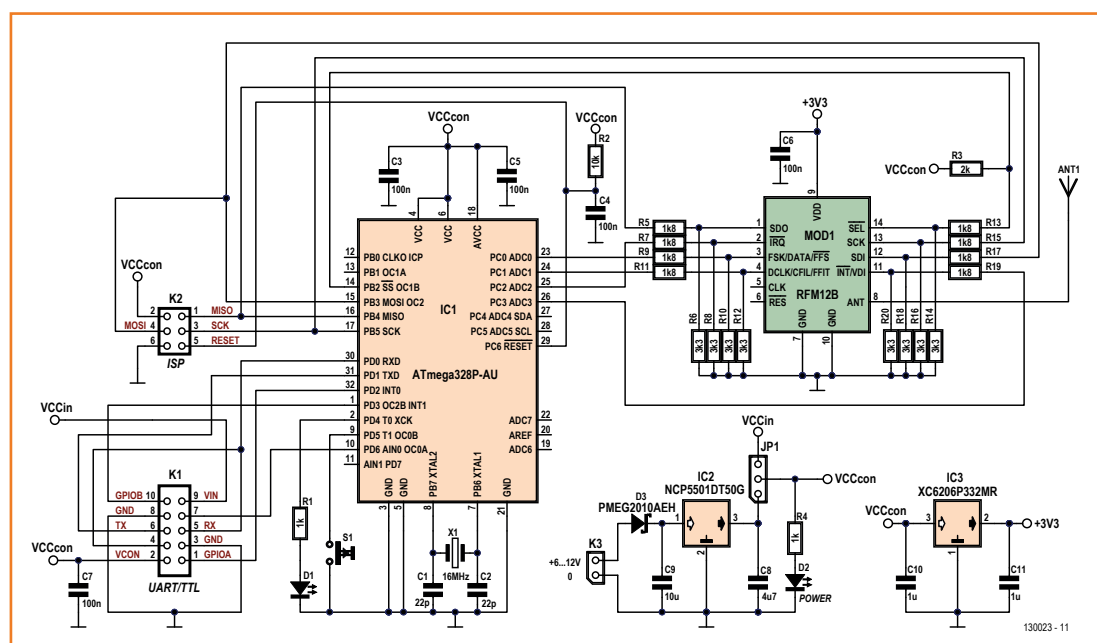


Figure 3.
Le microcontrôleur est relié au module radio par un total de huit lignes de signal. Cinq d'entre elles sont utilisées pour le chargement du programme.

Liste des composants

Résistances

R1, R4 = 1 k Ω (0805)
 R2, R3 = 10 k Ω (0805)
 R5, R7, R9, R11, R13, R15, R17, R19 = 1k8 (0603)
 R6, R8, R10, R12, R14, R16, R18, R20 = 3k3 (1%, 0603)

Condensateurs

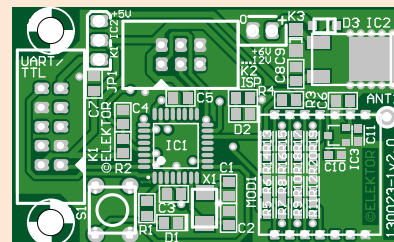
C1, C2 = 22 pF (0805)
 C3 à C7 = 100 nF (0805)
 C8 = 4,7 μ F
 C9 = 10 μ F/25 V (1206)
 C10, C11 = 1 μ (0603)

Semi-conducteurs

D1 = LED jaune (0805)
 D2 = LED verte (0805)
 D3 = diode Schottky PMEG2010AEH
 IC1 = ATmega328P-AU
 IC2 = LDO NCP5501DT50G
 IC3 = XC6206P332MR (SOT-23-3)

Divers

K1 = barrette HE10 2x5 (pas 2,54 mm)
 K2 = barrette HE10 2x3 (pas 2,54 mm)
 K3 = barrette HE10 1x2 (pas 2,54 mm)
 JP1 = barrette HE10 1x3 (pas 2,54 mm) avec cavalier
 S1 = touche
 MOD1 = module radio 433-MHz
 RFM12B-433-S1



X1 = quartz 16 MHz, 50 ppm, 18 pF
 barrette mâle 1x14 broches (pas 2 mm)
 barrette femelle 1x14 broches (pas 2 mm)
 circuit imprimé 130023-1 [3]

Les broches de ports PD4 et PD5 sont connectées à une LED et une touche. Une telle interface utilisateur minimale peut se révéler très utile pour le développement d'un programme, mais aussi pour des tests et la marche normale de la carte. Le module radio RFM12B est raccordé au microcontrôleur par un total de huit lignes. Quatre d'entre elles constituent l'interface SPI, elles sont

reliées à l'interface SPI matérielle de l'ATmega. Le microcontrôleur utilise cette même interface pour la communication avec un programmeur. La broche Chip Select \overline{SEL} , retenue au niveau haut par la résistance de polarisation R3, empêche le module radio de réagir aux octets SPI pendant la programmation. Quand des commandes ou des données sont effectivement destinées au

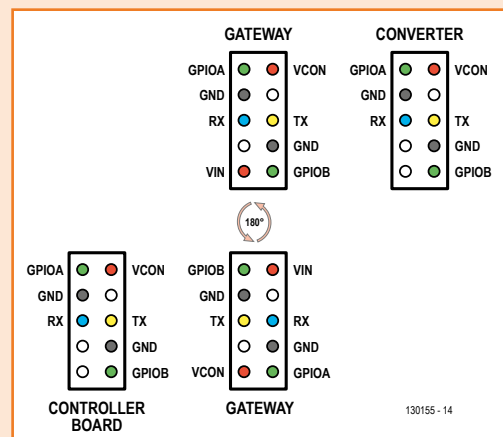
Spécifications du connecteur ECC pour les signaux de l'UART

Le connecteur ECC à 2x5 broches, utilisé dans ce projet, ne se charge que des signaux de l'UART. Deux signaux numériques sont ajoutés aux lignes RX et TX (GPIO, en vert sur le schéma). Nous avons décrit dans la dernière livraison d'Elektor un petit convertisseur UART/RS485 qui se connectera à une carte à μ C par le connecteur ECC. Nous avons en plus dans les cartons un convertisseur UART/RS232, et aussi des modules pour USB, WLAN et Bluetooth.

La broche en haut à droite permet l'alimentation du convertisseur par la carte à microcontrôleur. Une passerelle comme celle de cet article possède, contrairement à un adaptateur « bête », son propre μ C avec un logiciel. L'ECC permet de connecter une telle passerelle aussi bien comme un convertisseur que comme une carte à μ C. Dans le premier cas, la passerelle peut alimenter le convertisseur par la broche VCON. De même la passerelle peut être alimentée via la broche VIN par une carte à μ C. Si on relie les cartes par un câble en nappe, il suffit de tourner de 180° le connecteur du côté passerelle pour relier la broche VCON de la carte μ C à la broche VIN de la passerelle. En pratique, cela croise en même temps les signaux RX/TX, de façon à permettre le dialogue entre le μ C de la carte principale et celui de la passerelle.

C'est là l'application standard ; notre passerelle ajoute la fonction radio à une carte à μ C. C'est pourquoi la broche 1 du plan d'implantation de l'ECC est tournée vers l'intérieur ; c'est aussi la direction dans laquelle doit être orienté le connecteur du câble en nappe. Sur toutes les cartes à μ C, la broche 1 de l'ECC (et le repère du connecteur) est tournée vers l'extérieur. Attention : du fait de la permutation (de RX/TX) il n'est pas permis de déduire la fonction des lignes de leur brochage : la broche 1 correspond ici à GPIOA mais là à GPIOB.

Les broches dessinées en blanc ne sont pas affectées, elles peuvent servir pour des fonctions spéciales. Dans ce projet, nous avons relié une des broches à RX et une autre à une broche d'E/S (GPIO) de l'ATmega328.



module radio, la broche $\overline{\text{SEL}}$ (PB2) doit être tirée au niveau bas par le logiciel.

Le logiciel préchargé utilise en plus la broche FFIT du module radio. Les broches FFS, INT et IRQ sont reliées également à des lignes du microcontrôleur, pour que chacun puisse piloter le module radio avec encore plus de souplesse, au moyen de son propre logiciel.

Le connecteur K1 est celui de l'ECC cité plus haut (voir encadré). Les broches du milieu donnent accès aux lignes RX/TX du microcontrôleur. Trois autres broches sont reliées aux broches de ports PD2, PD3 et PD6 du microcontrôleur. En cas de besoin, ces lignes peuvent être utilisées par le logiciel pour le contrôle de flux de la communication ou pour des fonctions spéciales. La broche en bas à gauche véhicule la tension d'alimentation +5 V du μC . Elle peut alimenter un utilisateur peu gourmand, par exemple un convertisseur de signaux TTL/UART en RS232 ou RS485. Nous avons proposé le mois dernier [2] un adaptateur RS485 à connecter à l'ECC.

En sens inverse, la broche en haut à droite permet d'alimenter la passerelle par la platine d'application. C'est possible aussi par K3 et le régulateur de tension IC2. La source de tension est sélectionnée par le cavalier JP. La LED verte témoigne de la présence de la tension.

À partir de la tension d'alimentation de 5 V de la passerelle, un régulateur distille les 3,3 V requis pour alimenter le module radio ; les signaux de 5 V sont ramenés en même temps à 3,3 V.

Le circuit imprimé n'est pas bien grand, grâce aux CMS utilisés. Le tracé est téléchargeable sur la page de l'article. Il est possible de commander aussi un module assemblé prêt à l'emploi, avec un module radio RFM12B-433. Si vous souhaitez utiliser un autre module, attention aux 3,3 V de l'alimentation, car il circule encore des modules à alimenter sous 5 V dont la production a par ailleurs été arrêtée.

Il suffira de connecter à la borne ANT1 un morceau de fil de 17 cm (un quart de la longueur d'onde).

Applications

Le connecteur ECC laisse deux broches non définies, pour des applications particulières (voir le dessin dans l'encadré, broches blanches). Nous avons doublé sur une des broches le signal RX

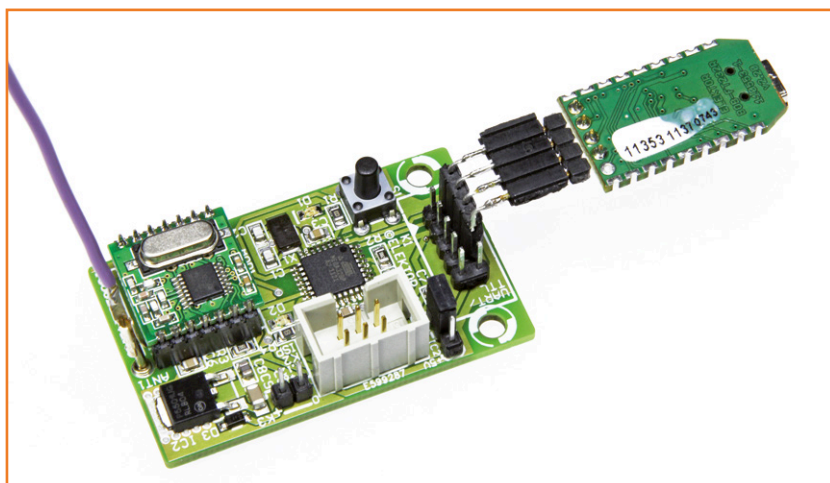
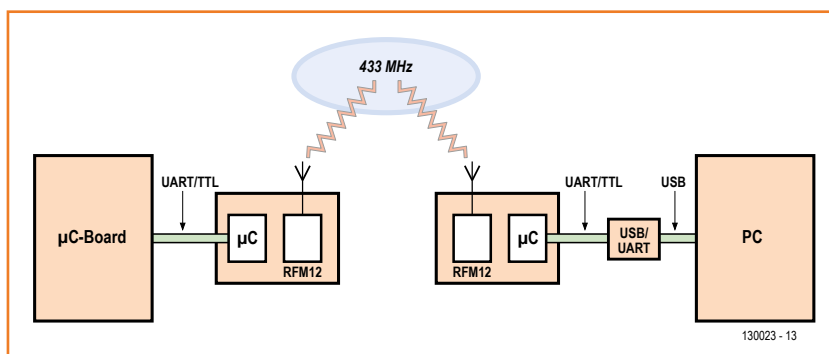


Figure 4.
Le convertisseur USB/UART BOB se connecte directement à la passerelle.

de l'UART, de telle façon que RX, TX et la masse se trouvent à la suite sur une rangée verticale. Cela nous permet, pour des tests ou en marche normale de la passerelle, d'enficher directement un convertisseur USB/UART BOB [3] (**fig. 4**). Si vous disposez d'un câble UART/USB 5 V de FTDI (également disponible chez Elektor [3]), vous pouvez le connecter ici, mais après avoir accordé le brochage du câble à celui du connecteur [4].

Le convertisseur USB se connecte à un PC. Un deuxième module radio est relié à une carte à microcontrôleur dont les lignes RX, TX et masse sont ressorties (relier TX du micro à RX de l'ECC et inversement). Il est possible maintenant d'envoyer du PC vers le microcontrôleur des chaînes de 62 caractères au maximum (**fig. 5**). Du côté du PC, il suffit d'un programme de terminal simple, tel que HTerm, disponible gratuitement sur l'internet [5]. On configure HTerm de telle façon qu'il ajoute automatiquement un <CR> (ASCII 13) à la chaîne de caractères lors de la pression sur la touche retour chariot. Pour la com-

Figure 5.
Synoptique de mise en œuvre de la passerelle. Les chaînes de caractères peuvent être transmises sans fil du PC à une carte à μC et retour.



munication radio en sens inverse, le microcontrôleur doit aussi envoyer par l'UART une chaîne de caractères terminée par un <CR>.

La passerelle programmée connaît deux modes : *Send* et *Listen* (émission et écoute). Lors de l'application de la tension d'alimentation, la passerelle est en mode *Send*, la LED jaune est éteinte. Dans ce mode, la passerelle reçoit des chaînes de caractères de l'UART et les retransmet aussitôt dans l'éther, mais elle ne reçoit rien par radio. Si on appuie sur la touche, la passerelle bascule en mode *Listen*, ce que signale la LED allumée. La carte est maintenant à l'écoute des caractères transmis par l'éther. Quand un <CR> arrive, la chaîne des caractères reçus jusque-là est délivrée par l'UART.

Commandes spéciales

On peut basculer la passerelle entre deux modes, en envoyant par l'UART la commande « @@c<CR> » (c = change). La commande « @@s<CR> » (s = *swap*, échange) bascule simultanément les deux passerelles d'un mode à l'autre. Toutefois, la passerelle reliée par radio ne bascule que si elle se trouve à l'écoute de la chaîne transmise par radio, c'est-à-dire si elle est en mode *Listen*. Une passerelle reliée par l'UART acquitte les deux commandes spéciales par les caractères « MDL » (= mode *Listen*) ou MDS (= mode *Send*). On obtient ainsi la confirmation du mode dans lequel se trouve la passerelle maintenant. On peut tester en même temps si la liaison par UART est correcte (fig. 6).

Dans le cas où on n'a besoin que d'une communication unidirectionnelle, il suffit de basculer une des passerelles en mode *Listen* pour que la transmission puisse commencer dans une direction. On peut à tout moment inverser la direction, en envoyant par l'UART à la passerelle qui est en mode *Send* la commande « @@s<CR> ». La passerelle répercute la commande par radio à sa correspondante.

Cela signifie en clair : « je cesse d'émettre, je passe en mode réception et j'attends ce que tu as à me communiquer par radio ».

Ce comportement est loin de couvrir tous les domaines d'application. Si par exemple une carte de capteur doit envoyer à une régulation sur PC le résultat d'une mesure périodique, il faut naturellement basculer la passerelle du côté PC en mode réception. Dans ce cas, le PC ne peut plus forcer

un changement de direction et envoyer une commande à la carte capteur, comme pour changer l'intervalle des mesures. C'est pourquoi le logiciel prévoit que les deux correspondants peuvent se trouver simultanément en mode réception. On peut toutefois envoyer par l'UART à chacune des cartes en mode *Listen* des caractères qui doivent être émis par radio. Pour des raisons logicielles, il faut ajouter un deuxième <CR> pour que la chaîne soit émise (appuyer une deuxième fois sur la touche <Enter> dans le programme de terminal). Dans une telle configuration bidirectionnelle, chaque correspondant peut émettre et recevoir. Naturellement, c'est au logiciel de l'application de chaque correspondant (carte à µC et PC) qu'il revient de s'assurer qu'il n'y aura pas de collision.

Logiciel

Il va de soi que le code source du logiciel embarqué peut être téléchargé du site Elektor, comme projet pour Atmel Studio 6 [3]. Il s'agit d'un logiciel modulaire qui repose sur la bibliothèque EFL (*Embedded Firmware Library*) [6]. Tout en bas de la pile logicielle est inclus le fichier *Controllerfile* de définition du µC ATmega328, qui met à la disposition des couches supérieures par exemple la fonction

```
uint8 SPIMaster_TransceiveByte(int8
    Handle, uint8 Databyte)
```

avec laquelle un caractère est envoyé par l'interface SPI. Dans ce cas, la fonction utilise un pilote de bas niveau du module radio, inclus dans le fichier *Boardfile* (*BoardEFL.c*). Le pilote de bas niveau comporte les fonctions :

```
void WirelessModule_Command(uint8
    WirelessBlockIndex, uint16 Command)
void WirelessModule_SendData(uint8
    WirelessBlockIndex, uint8* DataBuffer,
    uint8 DataLength)
void WirelessModule_ReceiveData(uint8
    WirelessBlockIndex)
```

Le paramètre *WirelessBlockIndex* est dans ce cas toujours à 0, mais il est possible en principe de piloter plusieurs modules radio. La fonction *WirelessModule_Command* permet d'envoyer au module une commande de deux octets, parmi celles que spécifie la feuille de caractéristiques

[2]. Pour cela, la fonction `SPIMaster_TransceiveByte` citée plus haut est appelée deux fois de suite, pendant que la broche `SEL` est maintenue au niveau bas.

La fonction `WirelessModule_SendData` envoie les caractères. Elle est précédée de l'envoi de l'octet `0xB8` au module radio. À partir de là, tous les octets qui parviennent au module par le port SPI sont émis. Pour éviter les parasites sur le transfert, les concepteurs du module radio ont prévu un signal de synchronisation de deux octets (`0x2DD4`) qui doit être envoyé avant les données utiles. Ensuite tous les octets sont transmis au module radio et émis par lui. À la fin de la chaîne de données à transmettre, la fonction accroche l'octet (défini dans `BoardEFL.h`) `CHAR_WIRELESS_ENDOFTTRANSMISSION = 0x04` (*End Of Transmission* en ASCII).

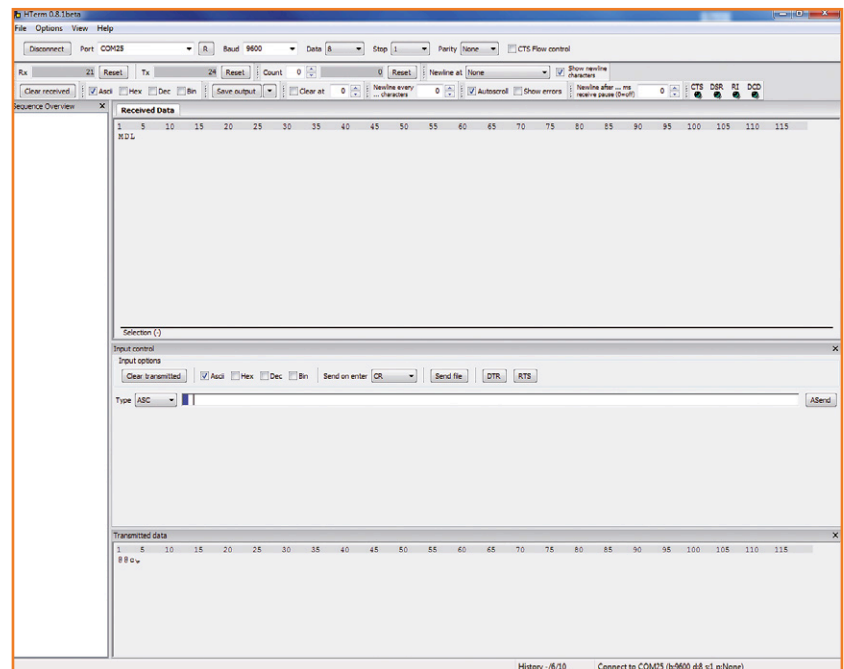
La fonction `WirelessModule_ReceiveData` est appelée à chaque fois en mode *Listen*. La partie la plus importante est une boucle de réception permanente. Au début de cette boucle se trouve une boucle plus petite qui teste constamment la broche `FFIT`. Elle passe au niveau haut quand le module a reçu un octet et l'a inscrit dans son propre tampon FIFO (*First In, First Out*). La broche `SEL` passe alors au niveau bas et la commande `0xB000` est envoyée au module, pour extraire le caractère reçu ; ensuite la broche `FFIT` est scrutée à nouveau. Le module radio de son côté ne commence à écrire les caractères dans son tampon FIFO que quand il a reçu le signal de synchronisation.

Les caractères reçus sont inscrits dans un tampon circulaire assigné au module radio ; comme tous les tampons circulaires de la bibliothèque EFL, il a une taille par défaut de 64 octets. Quand arrive le caractère `0x04`, la boucle de réception est abandonnée ; le signal EOT lui-même n'est pas conservé dans le tampon. Ensuite, le module radio sera averti par l'effacement d'un bit qu'il peut reprendre l'écriture dans le FIFO si le signal de synchronisation se présente à nouveau (au début de la prochaine chaîne de caractères).

Les boucles de réception sont abandonnées aussi si la touche est enfoncée ou si un caractère arrive par l'UART, pour permettre la bidirectionnalité décrite plus haut.

Configuration des modules radio

Nous avons vu maintenant toutes les fonctions nécessaires au pilotage du module. Naturellement, toutes les fonctions s'exécutent indépen-



damment du matériel, comme il se doit dans la bibliothèque EFL. Elles peuvent donc être utilisées sur d'autres cartes, avec un câblage différent. Le câblage de la passerelle, y compris ceux de la touche et de la LED, est défini dans la fonction `Board_init` du fichier `Boardfile` [6]. C'est le seul endroit où il faut adapter les définitions à un autre matériel.

Avant de pouvoir émettre ou recevoir des caractères, il faut configurer le module. Quels sont pour cela les caractères de commande à envoyer au module par la fonction `WirelessModule_Command` ? On n'ira guère plus loin sans passer par une étude approfondie de la feuille de caractéristiques. C'est la raison d'être de la bibliothèque `WirelessInterfaceEFL`. La fonction `WirelessInterface_LibrarySetup()` est appelée au démarrage de l'application, elle effectue la configuration de base du module, par exemple celle du débit à 9600 bauds. Chaque instruction du code est commentée. Le commentaire reprend le numéro de la section correspondante de la feuille de caractéristiques, ce qui permet de suivre ce qui se passe.

Dans la fonction `WirelessInterface_Send`, le module est d'abord configuré en émission. Pour cela le registre TX et l'émetteur sont activés. On ne le fait que juste avant d'émettre des données, de façon à économiser l'énergie.

Figure 6.
Un programme de terminal comme Hterm apporte une aide précieuse pour les tests.

Dans la fonction `WirelessInterface_Listen`, le mode FIFO du module radio est activé, ce qui entraîne entre autres choses que la broche FFIT utilisée plus haut signale l'arrivée d'un nouveau caractère. Par ailleurs, le récepteur est activé.

Passerelle 1:1 (*one to one*)

Il manque encore à notre logiciel un mécanisme qui lira périodiquement dans le tampon circulaire, en mode *Listen*, les caractères reçus par radio et les enverra par l'UART. Inversement, les caractères reçus par l'UART à la suite d'interruptions de l'UART seront écrits dans un (autre) tampon circulaire, lequel devra être lu périodiquement, lui aussi. C'est l'affaire de la bibliothèque *OneToOneGatewayEFL*, déjà décrite dans un autre article [7]. Les routines de la bibliothèque cherchent dans chaque tampon à son tour si un caractère <CR> a été reçu. Si oui, les caractères reçus jusque-là sont émis en une attelée vers l'autre canal de communication. La bibliothèque *OneToOneGatewayEFL* a été complétée pour ce projet par le traitement des commandes spéciales, annoncées par « @@ » et longues de trois caractères. On indique à la bibliothèque par la fonction `OneToOneGateway_SetSpecialCommandFunction()` quelle fonction

doit être utilisée en réponse (*Callback*) quand une commande spéciale est reçue par l'un des canaux. Dans notre cas, c'est la fonction `SpecialCommandGateway()`, qui fait partie du programme principal. Par ailleurs, c'est le troisième caractère, « c » ou « s », qui est passé comme paramètre. Suivant ce caractère, le mode de la passerelle est commuté en écoute ou émission. La commande spéciale peut toujours être envoyée par radio à l'autre module ; ce comportement a été fixé par le deuxième paramètre (TRUE) dans la fonction `OneToOneGateway_SetSpecialCommandFunction()`.

Test

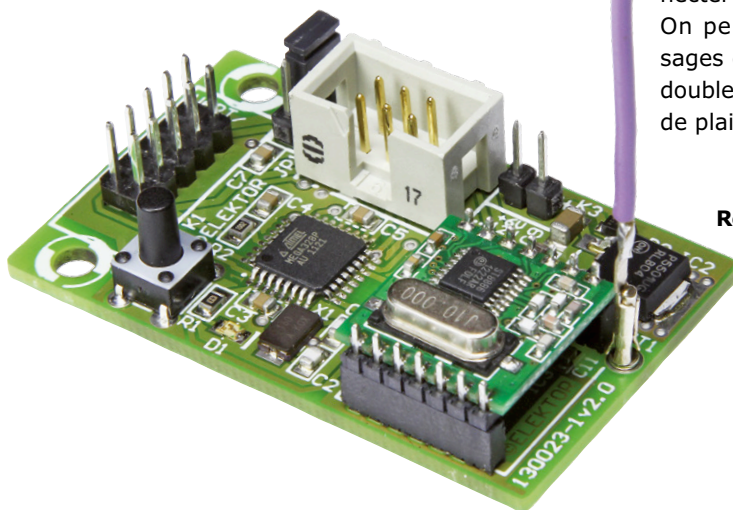
Pour le test de la passerelle, on peut équiper chaque unité avec un BOB ou un câble FTDI et connecter chaque convertisseur USB/UART à un port USB du même ordinateur. Avec deux programmes de terminal ouverts, les caractères doivent être émis et reçus.

Il faut d'abord basculer les deux passerelles indépendamment l'une de l'autre en mode *Listen*, par la commande « @@c<CR> ». Chaque module radio doit répondre « MDL » (= *Mode Listen*) par l'UART, ce qu'on voit dans le terminal. Ainsi est vérifiée au moins la communication par UART (si cela ne fonctionne pas, il suffit souvent de déconnecter et reconnecter le câble USB).

On peut maintenant envoyer de petits messages d'un terminal à l'autre (ne pas oublier le double <CR>). Nous vous souhaitons beaucoup de plaisir.

130023 – version française : Jean-Paul Brodier

Remarque : le progiciel proposé est purement expérimental. Nous n'en garantissons pas la stabilité.



Liens

- | | |
|--|--|
| [1] www.silabs.com/Support%20Documents/TechnicalDocs/Si4421.pdf | [6] www.elektor-magazine.fr/120668 |
| [2] www.elektor-magazine.fr/130155 | [7] www.elektor-magazine.fr/130154 |
| [3] www.elektor-magazine.fr/130023 | [8] www.elektor-labs.com/ecc |
| [4] www.elektor-magazine.fr/080213 | [9] www.elektor-magazine.fr/120126 |
| [5] www.der-hammer.info/terminal/ | |

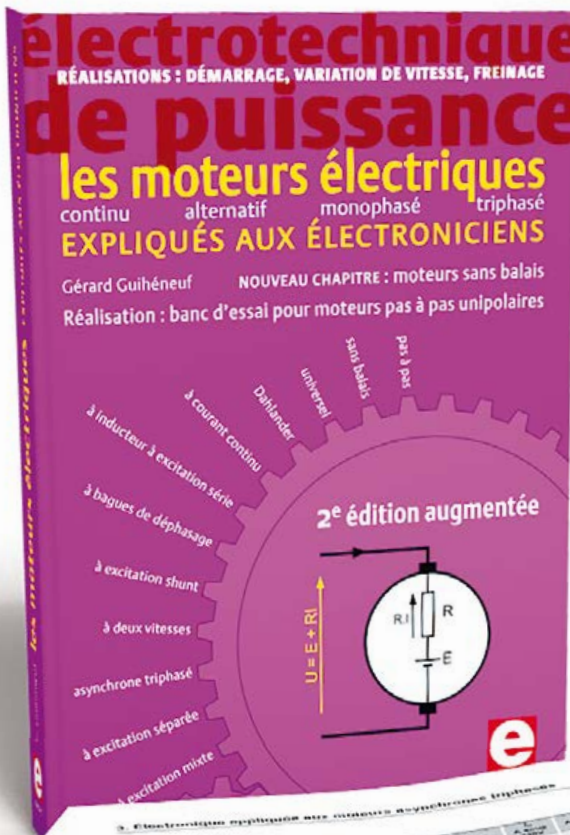
Nouvelle édition augmentée

100 pages supplémentaires !

électronique de puissance

les moteurs électriques

expliqués aux électroniciens



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

Le premier détaille les principes de variation de la vitesse des moteurs à courant continu. Les moteurs à alimentation alternative monophasée du 2e chapitre font appel à une électronique de puissance. Dans le 3e chapitre sont décrits les moteurs asynchrones triphasés, le moteur électrique le plus utilisé dans l'industrie.

Cette nouvelle édition s'enrichit d'un chapitre consacré aux **moteurs sans balais (brushless)**, du **moteur synchrone triphasé de plusieurs centaines de kW** au **moteur à courant continu de quelques centaines de watts**, en passant par les **moteurs pas à pas** ou encore le **surprenant moteur linéaire**. Le lecteur découvrira leurs modes de commande : codeurs incrémentaux ou absolus associés à un onduleur commandé en courant ou en tension ou bien capteurs à effet Hall pour l'autopilotage, commande en pas entiers, demi-pas, micro-pas... L'auteur propose également de réaliser un banc d'essai pour moteurs pas à pas unipolaires.

3.10.1. Caractéristiques principales des moteurs asynchrones triphasés

Type	200 W	400 W	600 W	1000 W	1500 W	2000 W	3000 W	4000 W	5000 W	7500 W	10000 W
U _N (V)	230	230	230	230	230	230	230	230	230	230	230
I _N (A)	1,0	1,8	2,7	4,0	6,3	8,0	12,5	16,0	20,0	31,5	45,0
U _{sc} (V)	230	230	230	230	230	230	230	230	230	230	230
I _{sc} (A)	6,0	10,8	16,2	24,0	37,8	48,0	72,0	96,0	120,0	189,0	270,0
U _{cc} (V)	230	230	230	230	230	230	230	230	230	230	230
I _{cc} (A)	0,8	1,4	2,1	3,2	5,0	6,3	9,6	12,5	15,8	25,1	35,4
U _{cc} (V)	230	230	230	230	230	230	230	230	230	230	230
I _{cc} (A)	0,8	1,4	2,1	3,2	5,0	6,3	9,6	12,5	15,8	25,1	35,4

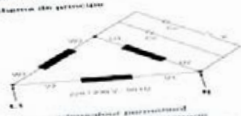
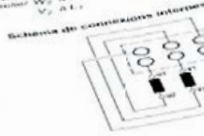
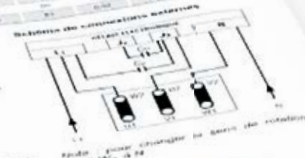


Figure 3-10 - Raccordement d'un moteur asynchrone triphasé à un réseau triphasé. Les caractéristiques du moteur sont indiquées sur la plaque signalétique. Les caractéristiques du réseau sont indiquées sur la plaque signalétique du tableau de bord.



nouveau chapitre + 100 pages

ISBN 978-2-86661-194-1

432 pages | 46,50 €

ATmega sur l'internet (2)

Le Raspberry Pi comme passerelle vers le réseau

Dieter Holzhäuser
(Allemagne)

Un ordinateur auxiliaire suffit pour piloter *via* l'internet et le réseau local un microcontrôleur ATmega32 relié par une ligne série. L'ordinateur auxiliaire sera ici du genre minimal, un Raspberry Pi. Nous montrons ici comment relier au réseau local et à l'internet le matériel décrit dans le premier article de la série.

La première partie [4] se terminait avec le clignotement d'une LED sur le RPi, commandé par un terminal et une liaison série. Le pilotage du programme de démonstration sur l'Atmega par le clavier et l'écran du RPi ressemblait déjà à une sorte de télécommande. Une vraie commande à distance de votre électronique maison, depuis n'importe quel endroit du monde, passe par le réseau domestique et l'internet. Voyons comment cela fonctionne.

Secure Shell / SSH

La commande à distance par terminal décrite dans le premier article peut être étendue à l'internet par SSH (*Secure Shell*). Dans le cas le plus simple, le protocole SSH fait apparaître sur un ordinateur les lignes de commandes d'un autre ordinateur relié par un réseau. L'échange de données par SSH est crypté et (relativement) sûr. La dénomination SSH est à la fois celle d'un protocole réseau et des programmes qui le mettent en œuvre. Le programme SSH qui tourne en arrière-plan sur le RPi est le serveur SSH. Il assure le service SSH. Le serveur est sollicité par un client SSH. L'ordinateur côté client peut se trouver dans le même réseau local (LAN, *Local Area Network*) que le RPi ou être relié par l'internet.

Si vous n'avez pas Linux sur l'ordinateur distant, démarrez (*boot*) simplement sur un *Live-CD* comme ceux que proposent régulièrement les magazines d'informatique. Les distributions usuelles permettent, sans installation particulière, d'ouvrir une console et de lancer un client SSH (voir plus loin).

Le Linux du RPi comporte un serveur SSH. Il peut être activé par *raspi-config*. (voir partie 1 [4]). L'internet propose toutes les informations nécessaires sur SSH [2].

SSH par LAN

Un RPi ne remplace pas un vrai PC de bureau. Pour un bon nombre d'applications, au contraire, il est logique de le piloter par un LAN sans écran ni clavier. Un LAN est un réseau local privé, habituellement avec un accès à l'internet pris en charge en totalité par un routeur dit *quelque-chose-box*. Le réseau local, comme l'internet, utilise les protocoles TCP-IP. Les nœuds d'un LAN ont normalement des adresses IP privées. Le routeur sépare le réseau local (LAN) du réseau global (WAN). Il « cache » le réseau local et le protège ainsi des accès depuis l'internet. Seul le routeur apparaît comme nœud avec une adresse IP publique. Dans cet article, c'est une *FRITZ!Box* qui apparaît au premier plan, mais tout autre modem-routeur (*quelque-chose-box*) peut être utilisé pour effectuer les mêmes tâches.

L'interface utilisateur de la *FRITZ!Box* permet de régler les détails relatifs au RPi. C'est là que vous trouverez entre autres son adresse IP et que vous adapterez la configuration pour que votre RPi ait toujours la même adresse IP locale. Si les données sont fixées comme sur la figure 1, l'appel du client SSH se fait sur une machine Linux connectée au réseau par la ligne de commande :

```
ssh pi@192,168,178,28
```

ou simplement :

```
ssh pi@raspberrypi
```

Après la saisie du mot de passe de l'utilisateur *pi* apparaît la ligne de commande du RPi. On démarre de là le terminal *picocom*. Si le programme de démonstration *suidemo1.c* tourne

sur l'ATmega, la ligne représentée dans la partie 1 s'affiche, avec les trois étoiles clignotantes. Après la sortie de picocom, on quitte ssh par exit.

SSH par l'internet

La liaison SSH par l'internet est établie suivant le même principe que sur le réseau local. Toutefois la configuration est notablement plus complexe parce que le client SSH sur l'internet doit se connecter au serveur SSH sur le réseau local. Le fonctionnement est inversé par rapport au mode usuel, dans lequel un navigateur sur le LAN demande en tant que client des pages web qui lui sont envoyées via l'internet par un serveur web. Le routeur internet est déjà préparé à recevoir des données, puisqu'elles sont la réponse à une requête émise par le LAN.

Du côté négatif, des données inconnues, arrivant par l'internet, comme celles d'un client SSH n'importe où dans le monde, représentent une faille de sécurité. En effet, elles doivent passer par des « ports » qui, dans un réseau local, sont normalement fermés. Les ports compris dans une adresse internet complète sont désignés par des numéros attribués à des services des réseaux. Par exemple, le port standard du service SSH est numéroté 22. Un routeur doit être expressément autorisé à transmettre au port 22 du RPi des données SSH « non requises ». Ce port n'est « capté » que par le serveur SSH du RPi.

La figure 2 montre l'ouverture du port sur la FRITZ!Box pour le serveur SSH du RPi. Il faut penser à récuser l'autorisation une fois qu'elle n'est plus nécessaire.

La page de configuration générale du routeur permet de trouver, en plus du nombre de ports ouverts, l'adresse IP publique de la FRITZ!Box. En appelant cette adresse (et non comme ci-dessus), un client SSH accède par l'internet au serveur SSH du RPi. Cet appel ne s'adresse pas seulement au RPi mais aussi au réseau local auquel il appartient, parce que la FRITZ!Box le propulse sur l'internet avec une adresse où on peut le trouver. Le hic de l'affaire est que l'adresse IP de la box change, du côté fournisseur, (presque) à chaque accès à l'internet. Même une connexion permanente n'empêche rien, parce que normalement le fournisseur d'accès doit la couper une fois par jour, ce qui donnera forcément lieu à l'attribution d'une nouvelle adresse.

Detailed description: This is a screenshot of the 'Details für raspberrypi' page in the FRITZ!Box web interface. It shows the configuration for a specific network device. The fields include: Name (raspberrypi), a 'Zurücksetzen' button, IPv4-Adresse (192.168.178.28), a checkbox 'Diesem Netzwerkgerät immer die gleiche IPv4' which is checked, and Anschluss über (LAN 1).

Figure 1.
Configuration de l'adresse IP de la FRITZ!Box.

Detailed description: This is a screenshot of the 'Portfreigabe' (Port Forwarding) section in the FRITZ!Box web interface. It shows the configuration for opening port 22 for the 'ssh-Server' application. The settings include: Bezeichnung (ssh-Server), Protokoll (TCP), von Port (22) bis Port (22), an Computer (raspberrypi), an IP-Adresse (192.168.178.28), and an Port (22).

Figure 2.
Ouverture du port SSH du RPi.

Detailed description: This is a screenshot of the 'Freigaben' (Port Forwarding) section in the FRITZ!Box web interface, specifically the 'Dynamic DNS' tab. It shows the configuration for using Dynamic DNS. The settings include: Dynamic DNS-Anbieter (Benutzerdefiniert), Update-URL (http://dynup.de/dyn.php?username=fritzbox&g...), Domainname (fritzbox50.p7.de), Benutzername (fritzbox), and Kennwort (****).

Figure 3.
Données DDNS de la FRITZ!Box.

Malheureusement, il n'y a pas de référence pour les adresses IP de routeurs dans les DNS (*Domain Name Service*) publics. Le seul recours est le DDNS, *Dynamic DNS*. De cette façon un nom de domaine fixe est attribué à la FRITZ!Box. Ce nom de domaine appelle d'abord un serveur DDNS qui appelle ensuite la FRITZ!Box par son adresse IP actuelle. C'est possible parce que la FRITZ!Box communique (*update*) au serveur DDNS chaque changement de son adresse IP.

Cela se passe ainsi en pratique : les fournisseurs de DDNS (par exemple « dyndns:free » [3]) proposent leurs services gratuitement pour un certain temps. Ensuite il faut s'enregistrer. Les données d'accès DDNS arrivent par courriel. Le report dans la box est relativement simple (figure 3).

Saisir d'abord le nom de domaine par lequel la box est accessible. Ensuite une partie définie du nom le complète à gauche. Par ailleurs la *FRITZ!Box* réclame un numéro de compte DDNS et un mot de passe. Pour finir, le DDNS est ouvert pour la *FRITZ!Box* et les informations d'accès au serveur DDNS sont saisies.

Dès le transfert des informations, la box essaie de s'enregistrer auprès du serveur DDNS, ce qui est possible aussi par programme. L'URL de `dyndns.free` est :

```
http://dynup.de/dyn.php?username=xxx&password=xxx&hostname=xxx.xxx.de
```

Les `xxx` remplacent les noms réels.

Le résultat est visible sur la page `online-monitor` ou sur la vue générale. Maintenant la liaison est établie entre l'internet et le serveur SSH du RPi sous le nom attribué :

```
ssh pi@xxx.xxx.de
```

Ici aussi les `xxx` masquent les noms.

Il existe aussi des applications SSH pour *smartphone*. Le programme de l'ATmega32, le `shell` et `picocom` attendent le dialogue avec un terminal. Il faut que les touches `Ctrl`, `Esc` etc. soient disponibles. Or presque aucun des claviers de *smartphone* ne comporte ces touches d'origine. Il faut donc les ajouter ou les simuler. L'appli Android `vSSH`, par exemple, a déjà un clavier étendu.

Pour conclure

Naturellement, ménager un accès par l'internet au programme de l'ATmega32 ne vaut la peine que si le petit programme de démonstration est remplacé par une application plus puissante et utile. Le programme `mmdemo1.c` (téléchargement gratuit [4]) constitue une interface utilisateur série avec quatre cas de communication dans un système multitâches rudimentaire qui tient

dans quelque 300 lignes de code. On peut, en respectant le schéma du programme, le transformer en une application plus étendue. Le point remarquable est la possibilité pour un terminal de communiquer avec plusieurs microcontrôleurs sur un bus en ET câblé à basse impédance qui autorise de grandes distances entre les nœuds. La description détaillée du programme se trouve en annexe du texte source.

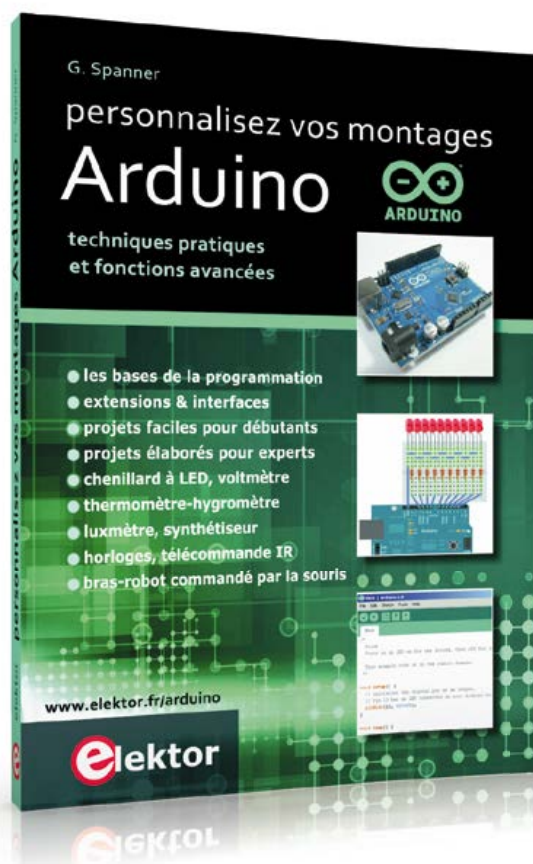
Encore un conseil important : si vous voulez utiliser une solution similaire, télécommander par l'internet des appareils réels et pas seulement quelques LED, vous pouvez renforcer la sécurité en n'autorisant l'accès au matériel qu'après la saisie d'un mot de passe (le RPi et SSH en réclament déjà un). Ne négligez pas la sécurité, car des farceurs ou des individus moins bien intentionnés pourraient profiter de l'anonymat et vous causer des désagréments.

130481 – version française : Jean-Paul Brodier

Liens

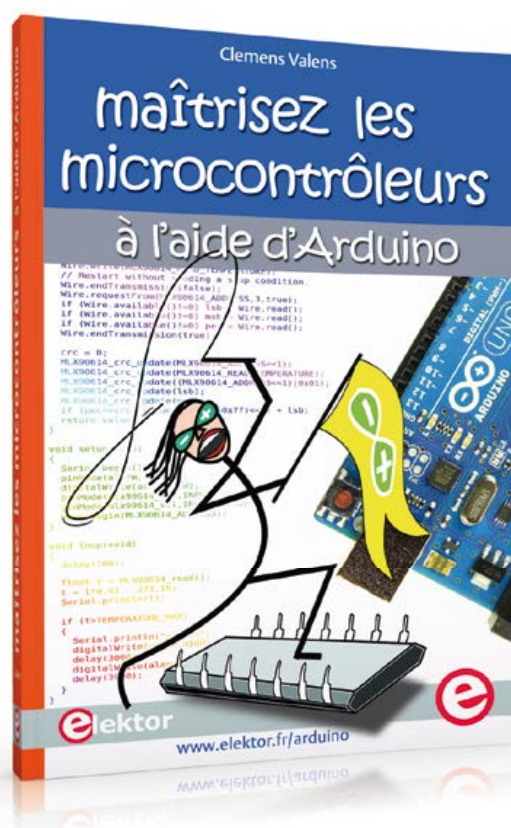
- [1] Site de l'auteur : www.system-maker.de
- [2] Informations sur le SSH : <http://doc.ubuntu-fr.org/ssh>
- [3] Fournisseur DDNS : www.dyndn.fr
- [4] 1^{ère} partie : www.elektor-magazine.fr/130213
2^e partie : www.elektor-magazine.fr/130481

Changez de loisirs, devenez dresseur de puces !



Ce livre vous emmène à pas guidés vers la maîtrise des techniques essentielles d'Arduino. La route a été soigneusement tracée : projets regroupés par thème, bases théoriques mais approche toujours concrète.

Des montages Arduino étudiés spécialement pour se débarrasser de ses amis et de sa famille et pour se retrouver enfin seul et libre de passer tout son temps à apprendre la programmation des microcontrôleurs !



Également disponibles:



www.elektor.fr/arduino

calcul des transfos de courant pour des mesures sans contact ?

Martin Oßmann
(Allemagne)

Pour mesurer un courant, vous utilisez d'habitude un *shunt* qui transforme le débit en une tension à lire sur un voltmètre. Sur le secteur, ce procédé est risqué. Il est possible de faire ce genre de mesures sans contact, à l'aide d'un *transformateur de courant*, sauf en continu. En dépit de cette restriction, le procédé mérite d'être approfondi, les applications ne manquent pas : p. ex. les mesures de formes d'ondes à l'oscilloscope sur les alimentations à découpage.

Un transformateur de courant, on le construit souvent sur un tore de ferrite. La mesure du courant s'opère à travers le noyau (**fig. 1**). On y place un enroulement secondaire bouclé par un *shunt* sur lequel on mesure la tension, à l'oscilloscope dans l'exemple de la **figure 2**.

Notre source d'impulsions de tension [1] du récent numéro double fournit ici des impulsions de 10 V (courbe en bleu). Avec une résistance de 20 Ω , on obtient des impulsions de courant de 500 mA qui produisent sur le shunt de mesure des signaux de 50 mV (courbe rouge). Avec le transformateur de courant, on retrouve des impulsions de 5 mV. La fréquence est de 250 kHz. Le résultat est parlant : le transformateur de courant restitue bien la forme rectangulaire du signal d'origine. Mais que se passe-t-il quand la fréquence diminue jusqu'à 50 Hz (**fig. 3**) ? Le transformateur de courant ne rend plus la forme rectangulaire du signal (courbe verte).

À quoi cela tient-il, c'est ce que nous allons voir. Une modélisation va nous aider à trouver quelles caractéristiques doit présenter un transformateur de courant.

Circuit équivalent du transformateur

Pour construire une pince de courant, il faut un modèle bien représentatif de ce qui se passe dans le transformateur de courant. Commençons par là.

Un transformateur se compose d'un bobinage primaire (indice 1) et d'un enroulement secondaire (indice 2). Leurs nombres de spires sont égaux à n_1 et n_2 . On peut mesurer leurs inductances dites à vide L_{1o} et L_{2o} (**fig. 4**). Si les enroulements primaire et secondaire sont réalisés de manière homogène sur le noyau, on peut les calculer au moyen de la valeur A_L . Voyons ce qu'il en sort :

$$L_{1o} = n_1^2 \times A_L \quad \text{et} \quad L_{2o} = n_2^2 \cdot A_L$$

Le rapport du nombre de spires est alors $N = n_2 / n_1$. Pour un noyau en *Vitroperm* avec $A_L = 80 \mu\text{H} / n^2$ on obtient par exemple :

$$n_1 = 1 \quad L_{1o} = 80 \mu\text{H} \quad n_2 = 25$$

$$L_{2o} = 50 \text{ mH} \quad N = n_2 / n_1 = 25$$

Autre grandeur intéressante pour un transformateur, le facteur de couplage k . On peut obtenir sa valeur en mesurant l'inductance de court-circuit. Boucler le secondaire et mesurer : dans notre exemple $L_{1k} = 0,2 \mu\text{H}$. Puis court-circuiter le primaire et mesurer L_{2k} . Cette valeur n'est pas sans



relation avec les trois autres, on peut écrire :

$$L_{2k} \times L_{1o} - L_{1k} \times L_{2o} = 0$$

Grâce à cette équation, on peut par exemple calculer L_{2k} , ce qui donne ici :

$$L_{2k} = \frac{L_{1k} \cdot L_{2o}}{L_{1o}} = 125 \mu H$$

Le facteur de couplage, on l'obtient par :

$$k = \sqrt{1 - \frac{L_{1k}}{L_{1o}}} = 0,9987$$

Dans notre cas, c'est pratiquement 1, ce qui veut dire qu'un transformateur de courant à noyau torique présente un couplage proche de l'idéal.

Le transformateur est complètement déterminé par les trois valeurs L_{1o} , L_{2o} et L_{1k} . Pour décrire le comportement de circuits à transformateur, on se sert d'un circuit équivalent. La forme générale en est représentée à gauche de la **figure 5**. Le circuit équivalent se compose d'un transformateur idéal dont le rapport de transformation est de 1 : M. Il y a encore L_m , l'inductance principale ou de magnétisation, L_{1s} l'inductance de fuite primaire et L_{2s} l'inductance de fuite secondaire. Le fait que le modèle compte quatre grandeurs ne suffit pas à déterminer totalement le transformateur, on a aussi le choix de certaines valeurs, même si ces éléments n'ont pas de répercussion physique directe.

Passons au modèle de droite de la figure 5. Il contient l'inductance de fuite primaire L_s qui dérive directement de la mesure en court-circuit $L_s = L_{1k}$. L'inductance principale L_m provient de la mesure à vide ou de la formule d'enroulement $L_m = L_{2o} = n_2^2 \times A_L$. Comme rapport de transformation M, on a : $M = N/k = 25,0313... \approx N$. Comme le facteur de couplage est voisin de 1, M vaut environ autant que le rapport des nombres de spires. Ainsi avons-nous tous les éléments pour déterminer le circuit équivalent de la **figure 6**, nous pouvons les envoyer à un simulateur tel que SPICE.

À présent, faisons circuler le courant I à mesurer dans le primaire. Côté secondaire, nous avons raccordé une petite résistance R, mesurons la

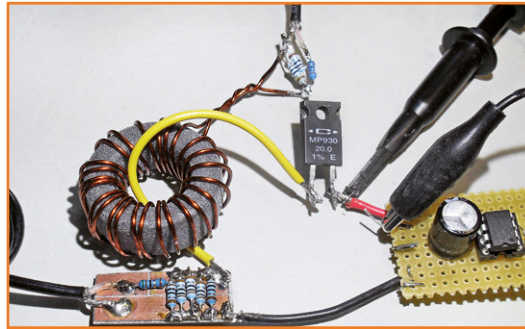


Figure 1.
Un tore de ferrite utilisé comme transformateur de courant.

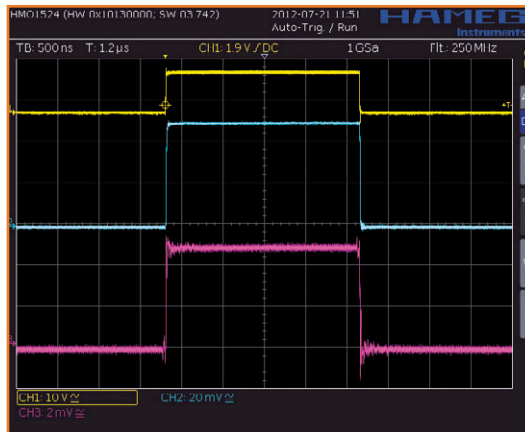


Figure 2.
Mesure du courant d'un signal rectangulaire à 250 kHz.

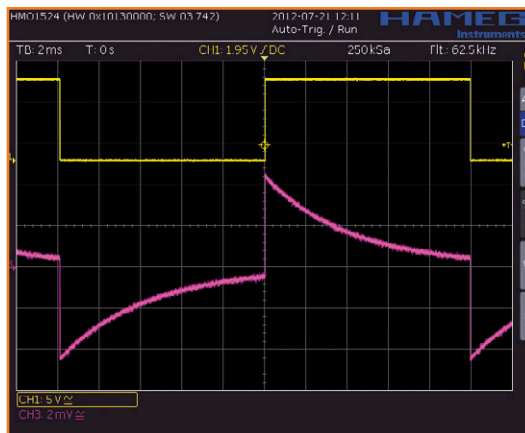


Figure 3.
Une mesure d'un signal rectangulaire à 50 Hz à travers un tore de ferrite.

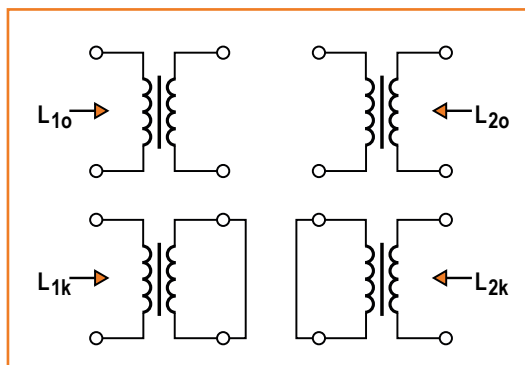


Figure 4.
Les mesures sur un transformateur à vide et en court-circuit.

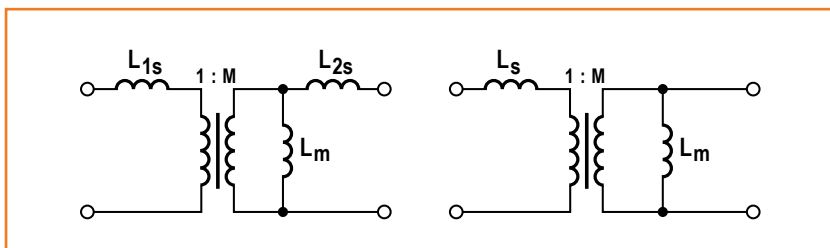


Figure 5.
Le circuit équivalent d'un transformateur.

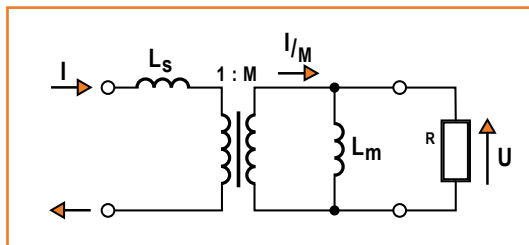


Figure 6.
Le circuit équivalent du transfo de courant.

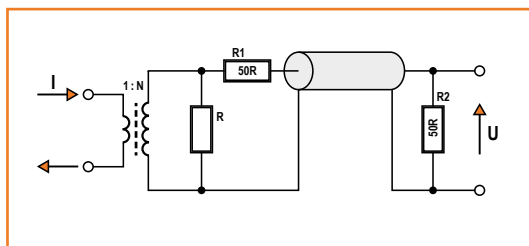


Figure 7.
Dispositif de mesure par transfo de courant.

tension à ses bornes. Comme l'inductance de fuite est en série, elle n'influence pas le courant qui traverse le transformateur. Dans le secondaire, il passe un courant transformé I/M qui, à cause de la situation en parallèle de L_m avec R , provoque la tension de mesure U . Pour les transformateurs de courant, on a la plupart du temps $n_1 = 1$, alors considérons que $N = M = n_2$.

Si la fréquence du courant I est tellement grande que l'impédance de la bobine est beaucoup plus grande que R , on peut négliger L_m et il reste $U = R \cdot I / N$. L'impédance de transfert devient ainsi $R_{TR} = U / I = R / N$. Une résistance en série avec une self est un filtre passe-haut, la forme d'onde de la **figure 3** le confirme. Nous pouvons donc dire que la fréquence de coupure se situe à :

$$f_g = \frac{R}{2 \cdot \pi \cdot L_m}$$

dont on déduit :

$$R_{TR} = \frac{R}{N} \quad f_g = \frac{R}{2 \cdot \pi \cdot A_L \cdot N^2}$$

Calcul des composants

Avec ces informations, on peut alors déterminer les valeurs à adopter pour le transformateur de courant. Prendre R plus grand fera augmenter la tension du signal, mais en même temps, la limite inférieure de fréquence remonte, on ne pourra plus mesurer de basses fréquences et du même coup, on augmente la chute de tension dans le circuit primaire. Le choix de la valeur de R résultera donc d'un compromis. Comme les oscilloscopes ne donnent souvent que de pauvres images dans le domaine des millivolts, nous choisirons pour une valeur suffisante pour les dépasser avec des valeurs moyennes de courant.

Parlons du nombre de spires. Si N est grand, l'impédance de transfert et donc la tension fournie s'amenuisent, mais L_m s'accroît, ce qui abaisse la limite inférieure de fréquence. Ici aussi, le compromis est de rigueur. On constate aussi qu'une plus grande valeur de A_L est favorable en permettant une plus basse fréquence de coupure sans altérer l'impédance de transfert.

Le tore de ferrite

Comme exemple, calculons les caractéristiques pour un tore de ferrite avec une valeur A_L d'environ $3 \mu\text{H}/\text{n}^2$. Avec $R = 0,5 \Omega$ et $N = 25$ on a :

$$R_{TR} = \frac{R}{2N} = \frac{0,5\Omega}{50} = 0,01\Omega$$

$$f_g = \frac{R}{2 \cdot \pi \cdot A_L \cdot N^2} = 42 \text{ Hz}$$

Le facteur 2 au dénominateur de l'impédance de transfert résulte des deux résistances de 50Ω , R_1 et R_2 . Elles adaptent le signal au câble coaxial de 50Ω , tout en formant un diviseur de tension par deux (**fig. 7**).

Avec un courant de 1 A , on n'aura que 10 mV sur l'oscilloscope. La fréquence de coupure basse avoisine 50 Hz . Avec ce genre de tore, nous n'atteindrons jamais une bonne précision sur des signaux à 50 Hz .

Vitroperm, le matériau fascinant

La firme *Vacuumschmelze* propose le *Vitroperm* (**fig. 8**), un matériau dont la perméabilité est très élevée aux basses fréquences. Le noyau torique T60006 L2025-W380 offre, sous le 1 kHz , une perméabilité A_L d'environ $80 \mu\text{H}/\text{n}^2$. Avec

les mêmes paramètres, il ramène à 2 Hz la fréquence de coupure. Un tel noyau permet de réaliser une pince de courant largement utilisable jusqu'à moins de 50 Hz.

Quelques mots à propos de la résistance interne de la pince de courant. Le *shunt* R du secondaire est transformé côté primaire par le facteur $1/N^2$. En utilisant $R = 0,5 \Omega$ et $N = 25$, la résistance effective du côté primaire ne vaut que $0,0008 \Omega$, ce qui est négligeable dans la plupart des cas. Reste en série l'inductance de fuite $L_{lk} = 0,2 \mu H$, qui à 1 MHz représente quand même une impédance de $1,2 \Omega$. La possibilité d'utiliser le transfo de courant à de plus hautes fréquences est conditionnée principalement par les capacités parasites. Mais notre exemplaire simple peut mesurer sans difficulté dans la gamme des mégahertz.

(130410 – version française : Robert Grignard)



Figure 8.
Des tores en Vitroperm.

Liens

[1] www.elektor-magazine.fr/130005

[2] www.elektor-magazine.fr/130410

Publicité



Android apprendre à programmer des applis

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.

ISBN 978-2-86661-187-3

280 pages

33,50 €

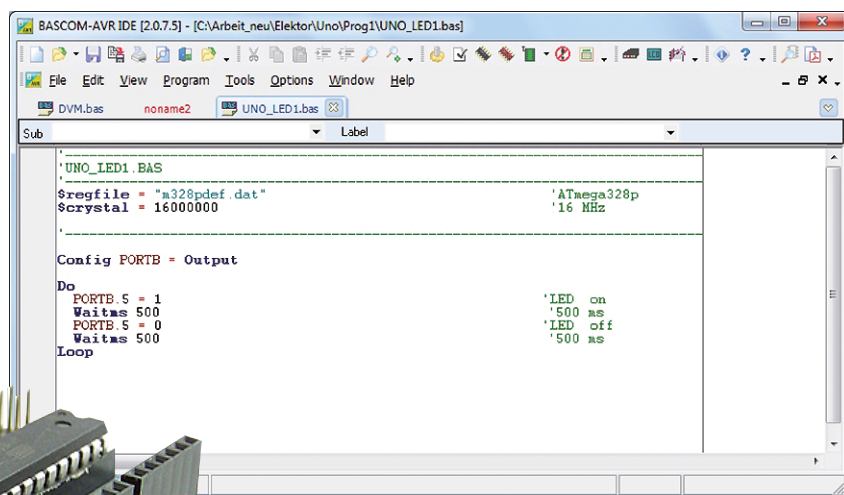
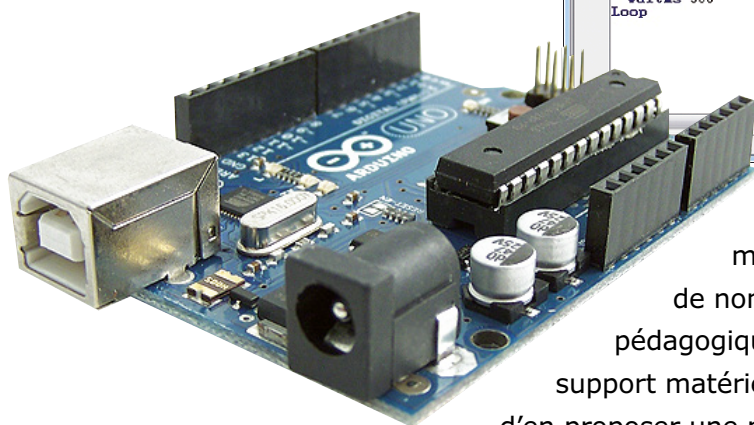


www.elektor.fr/android

l'atelier du microcontrôleur (1)

Arduino et Bascom

Burkhard Kainka
(Allemagne)



Une introduction simple au monde des microcontrôleurs, voilà ce que nous réclamons de nombreux lecteurs. Comme notre dernière série pédagogique remonte à quelques années déjà, et que son support matériel n'est plus disponible, le moment est venu d'en proposer une nouvelle. Celle-ci s'adresse au lecteur qui a quelques connaissances en électronique analogique et souhaite utiliser un microcontrôleur dans ses montages.

Pourquoi les microcontrôleurs* ont-ils envahi l'électronique ? Parce qu'on n'arrête pas le processus. Parce qu'ils sont puissants et qu'il en fleurit chaque jour de nouveaux, toujours plus rapides et dotés de mémoires de plus en plus grandes. Le novice qui entend cette réponse rétorquera : « Puissants ? Tu parles, Charles, la première application de la majorité des microcontrôleurs est une LED qui clignote ! Je fais ça avec trois composants conventionnels ». Il est vrai qu'on pourrait tout aussi bien faire clignoter une LED avec un temporisateur NE555, deux résistances et un condensateur. Et la comparaison avec le NE555 est d'autant plus pertinente qu'on trouve dans ce circuit intégré ce qu'on trouve dans un μ C. Alors comparons.

Anciens contre modernes ?

Voyons d'abord la sortie du 555. Le schéma de principe de sa fiche technique (**fig. 1**) nous indique qu'il s'agit d'une sortie *push-pull* (c'est-à-dire à transistors complémentaires) capable de fournir ou d'absorber un courant, donc de commuter des états haut et bas. Un μ C typique possède lui aussi de telles sorties, appelées ports. Le terme *port* désigne également un ensemble de sorties, ou encore de broches configurables chacune en entrée ou en sortie. La circuiterie du clignotant serait donc ici aussi composée d'une LED et de sa résistance-talon reliée à la masse (GND) ou à la tension d'alimentation (V_{cc}). Le NE555 possède également une sortie à collecteur ouvert,

* dans cette série, nous écrivons indifféremment microcontrôleur, μ C ou contrôleur.

mais celle-là il ne peut la forcer qu'au niveau bas (par absorption ou drainage de courant). Nombreux sont les microcontrôleurs capables de reproduire cette fonction. La seule différence tient au nombre de sorties : plusieurs pour un μC , une seule pour un NE555 (ses sorties *push-pull* et à collecteur ouvert ne sont pas indépendantes l'une de l'autre).

Voyons maintenant les entrées du 555. Il en possède deux, chacune reliée à un comparateur. Ces deux bornes commandent une bascule interne (*flipflop* sur le schéma). Une application typique est le multivibrateur astable, un circuit bistable dont beaucoup se servent pour construire rapidement un circuit clignotant. Brandissez quelques formules, choisissez les bons composants, et le circuit fera ce qu'il doit faire. Dimensionner un circuit à cœur de 555 et programmer un μC sont vraiment deux activités de même nature. Autre similitude, le μC et le NE555 possèdent chacun une entrée de remise à zéro (*reset*) qui, comme son nom l'indique, permet de ramener le circuit à son état initial. Très souvent cette entrée *reset* est au repos au niveau haut, et il faut la mettre au niveau bas (ou à la masse) pour obtenir l'initialisation (*reset*).

La **figure 2** est un générateur de signaux rectangulaires très simple, utilisé ici pour faire clignoter une LED. La fiche technique du NE555 contient d'autres exemples de circuits de base, dont une bascule monostable et un modulateur de largeur d'impulsion, autant de tâches typiques pour un μC . Au vu des innombrables applications du NE555, on se dit que les microcontrôleurs sont... superflus. Demandez à Google tout ce qu'il est possible de réaliser avec un 555 et autres CI semblables : cela va de la cellule photo-électrique à la commande de servomoteur, en passant par le traitement des signaux analogiques d'un capteur. Le potentiel de ce circuit intégré légendaire surpasse l'inventivité des électroniciens !

Développement (plus) rapide

On réalise mieux à quel point NE555 et μC se ressemblent lorsqu'on les compare côte à côte (**tableau 1**). Si nous devons trancher en toute objectivité, nous dirions que le μC possède *un peu de tout un peu plus*, et par conséquent convient mieux à la réalisation de tâches complexes. Là où il faudrait par exemple dix NE555 pour obtenir un certain résultat, il est probable qu'un seul μC

suffirait. Ce n'est qu'à partir d'un certain niveau de complexité que le μC s'avère la solution la plus légère et la moins coûteuse. Pour certaines petites tâches, il peut en effet être plus judicieux et plus rentable de se tourner vers l'électronique conventionnelle.

Le lecteur déjà familier des μC aura en tête un autre avantage : une fois le circuit à μC conçu, plus besoin de brancher son fer à souder, tout ce qui suit c'est la programmation. Modifier les fonctions d'un circuit puis tester les changements apportés est facile et rapide. En fait, le μC est un ordinateur universel et bon marché. La courbe d'apprentissage est raide, mais le retour sur investissement est garanti si l'on mise sur le temps que l'on gagne...ra.

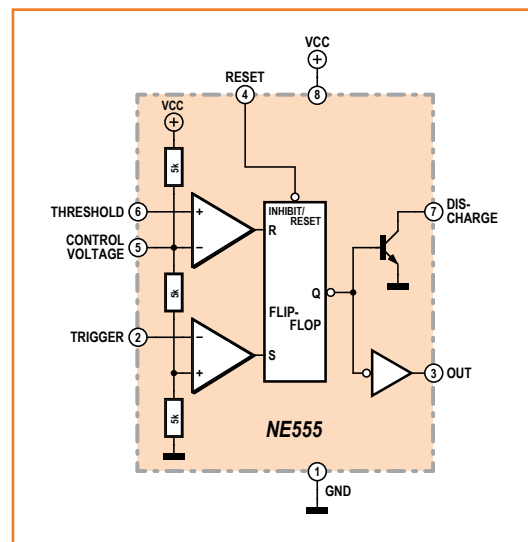


Figure 1.
Schéma de principe du NE555.

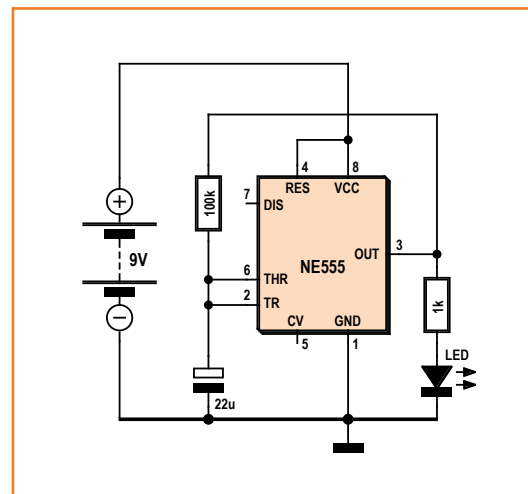


Figure 2.
Un clignotant à LED avec le NE555.

Tableau 1. Tableau comparatif du NEE55 et de l'ATmega

NEE55	Microcontrôleur ATmega
Sortie de commutation	Ports de sortie
Entrées	Ports d'entrée
Commande temporelle par circuit RC	Temporisateur interne, commandé par quartz
Entrée <i>reset</i>	Entrée <i>reset</i>
Entrées des comparateurs	Comparateur, entrées analogiques
Fonction MLI (<i>PWM</i>)	Sorties MLI (<i>PWM</i>)
Traitement du signal analogique	Convertisseur analogique/numérique
Bascule	Cellules de mémoire

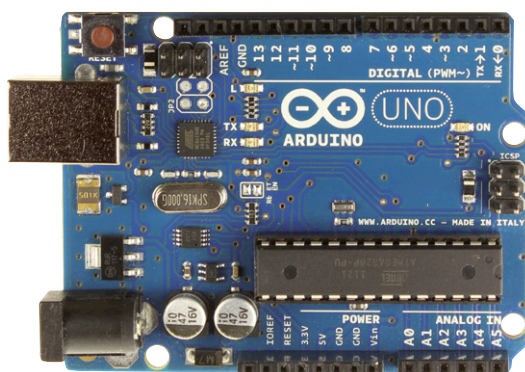


Figure 3.
La carte Arduino Uno.

Tableau 2. Caractéristiques d'Arduino.

Microcontrôleur	ATmega328
Tension de fonctionnement	5 V
Interface USB	alimentation 5 V et programmation
Alimentation externe	7 à 12 V
Broches d'E/S numériques	14 (dont 6 sorties MLI)
Canaux MLI	6
Entrées analogiques	6
Courant par broche d'E/S	40 mA (max.)
Sortie 3,3 V	50 mA (max.)
Mémoire flash	32 Ko (dont 0,5 Ko pour le chargeur d'amorçage)
SRAM	2 Ko (ATmega328)
EEPROM	1 Ko (ATmega328)
Horloge	16 MHz
Prix (membres Elektor)	24,75 € [1]

La fiche technique d'un μC peut décourager l'esprit le mieux disposé : 300 pages de prose technique, parfois plus. Heureusement pour nous, la route qui mène au monde des μC n'est pas uniquement pavée de fiches techniques. De nombreux langages de programmation permettent d'obtenir des résultats sans qu'il soit nécessaire de tout comprendre dès le départ. Il faut accepter les inévitables zones d'ombre, et persévérer. Le reste vient tout seul.

Arduino et Bascom

Les microcontrôleurs forment une famille nombreuse. Quel modèle privilégier dans un cours d'initiation ? Quelle carte et quel langage de programmation conseiller à un débutant ? Choisir n'a pas été facile, mais après de longues discussions nous nous sommes mis d'accord : la carte Arduino Uno pour le matériel, et Bascom pour la programmation !

Arduino est la plus populaire des plateformes pour amateurs. Le code s'écrit dans un langage de programmation simple et peut être transféré directement du PC à la mémoire du contrôleur à l'aide d'un câble USB. Il existe une grande variété de cartes et de *shields* Arduino (les cartes d'extension), qui plus est à des prix remarquablement bas. Et matériel et logiciel sont à source ouverte, donc documentés de façon exhaustive. La carte Uno [1] que nous avons privilégiée est elle aussi bon marché. Elle est équipée d'un ATmega328, un μC AVR du fabricant Atmel. Cette puce est rapide, se programme facilement, et est en outre dotée d'une mémoire suffisante pour accueillir les programmes relativement longs que vous pourriez être amenés à exécuter plus tard (**tableau 2**).

La plateforme Arduino dispose de son propre environnement de développement intégré (EDI), c'est-à-dire d'un logiciel qui permet d'écrire le micrologiciel d'un μC depuis un PC. Le code est d'abord écrit dans un éditeur de texte interne, puis compilé en octets exécutables qui seront finalement écrits dans la mémoire du μC .

Le code doit être écrit dans un langage que le compilateur peut traduire en code machine, c.-à-d. en instructions compréhensibles par le μC . Le langage propre à l'EDI Arduino est une version simplifiée du C, un langage bien connu des programmeurs professionnels. Ce C à la sauce Arduino dispose d'instructions spécialisées qui

facilitent l'écriture des programmes, et en général ceux qui se sont familiarisés avec ce langage l'apprécient et se débrouillent bien avec. Alors pourquoi avoir choisi le « BASIC » ? Pour plusieurs raisons. D'abord parce qu'il se compile avec le compilateur Basic pour $\mu\text{C AVR}$ et que ce compilateur est répandu et populaire. Ensuite parce que l'apprentissage est simple : le Bascom a lui aussi des instructions spécialisées, p. ex. pour la transmission de caractères depuis un contrôleur vers un PC, ou encore pour l'affichage de caractères alphanumériques. La raison qui aura fait pencher la balance en faveur de Bascom est toutefois celle-ci : Bascom est compris par tous les $\mu\text{C AVR}$. C'est bien une carte Uno (**fig. 3**) qui nous servira de plate-forme d'apprentissage, mais en fin de compte vous pourrez utiliser n'importe quel $\mu\text{C AVR}$, pas forcément un ATmega, sur n'importe quelle carte du marché ou faite maison. Vous pourrez ainsi utiliser, comme le font beaucoup de projets, un petit ATtiny à seulement 8 pattes.

L'interface USB est un des atouts de la carte Arduino. Elle permet non seulement d'acheminer les programmes jusqu'au μC (voir ci-dessous), mais aussi d'envoyer et de recevoir des caractères entre le PC et l'ATmega328. Ainsi peut-on commander le contrôleur depuis un PC, ou encore envoyer et afficher sur un PC des valeurs mesurées par le contrôleur. La carte peut en outre être alimentée par USB. En d'autres termes, il suffit d'un câble USB pour démarrer.

Premier programme

Le circuit de la **figure 4** est un clignotant à LED commandé par un ATmega328P. Seul le contrôleur est représenté, l'environnement matériel de la carte Arduino est absent du schéma. Rien à assembler ici, la carte est déjà équipée d'une LED. Nous utiliserons ce type de schéma simplifié pour tous nos exemples d'applications. Vous n'aurez besoin que d'un μC et d'une plaque d'essai pour reproduire les expériences. Vous pouvez très bien suivre cette série sans carte Arduino, mais pour un débutant l'utilisation d'une carte Uno sera plus pratique et plus simple.

Voyons le code Bascom de la LED clignotante (**listage 1**). Les deux lignes qui commencent par le caractère \$ sont des directives de compilation, autrement dit des instructions qui précisent au compilateur son environnement de travail. La

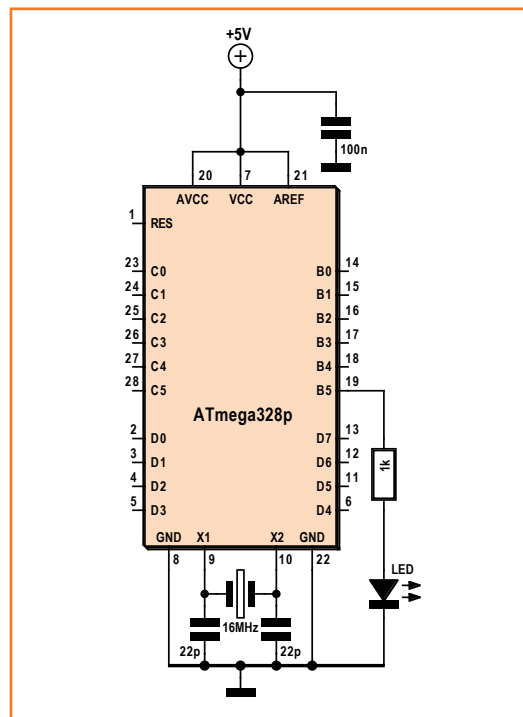


Figure 4.
Un clignotant à LED avec μC .

directive `$regfile` indique le type de contrôleur pour lequel le code doit être compilé. Nous utilisons ici l'ATmega328P, d'où la ligne :

```
$regfile = "m328pdef.dat"
```

La directive `$crystal` permet de définir la fréquence d'horloge du processeur. Plus cette fré-

Listage 1. Le clignotant à LED

```
'-----
'Uno_LED1.BAS

'-----
$regfile = "m328pdef.dat" 'ATmega328p
$crystal = 16000000        '16 MHz

'-----

Config Portb = Output

Do
  Portb.5 = 1      'LED on
  Waitms 500       '500 ms
  Portb.5 = 0      'LED off
  Waitms 500       '500 ms
Loop
```


quence est élevée, plus le μC exécute rapidement les instructions. Cette valeur est cruciale lorsque des opérations exigent une synchronisation précise. C'est le cas par exemple de la transmission de caractères vers le PC à un certain débit. L'horloge de l'ATmega est cadencée par un quartz de 16 MHz, ce que nous signalons au compilateur avec la ligne :

```
$crystal = 16000000
```

Les broches du contrôleur peuvent être utilisées en tant qu'entrées ou sorties. Nous devons donc également configurer leur sens. À la mise sous

proche de celle de la tension d'alimentation. Il circule donc un courant à travers la LED. L'instruction `Portb.5 = 0` remet la broche au niveau bas, c.-à-d. abaisse de nouveau la tension à un niveau proche de la masse. Le déroulement du programme est temporisé par les instructions `Waitms 500`. Le nom `Waitms` parle de lui-même : attendre 500 ms avant d'exécuter l'instruction suivante. Les 4 lignes sont placées dans une boucle `DO FOR` qui exécute indéfiniment les instructions qu'elle contient. Seule solution pour arrêter le programme, mettre la carte hors tension ou l'initialiser avec le bouton reset !

Compilateur

Comment compiler le code et comment l'envoyer dans le μC ? Le compilateur Basic Bascom-AVR a été conçu par Mark Alberts pour les familles de microcontrôleurs 8051 et AVR. Il existe une version complète (payante) et une version d'essai (gratuite). Les deux peuvent être téléchargées depuis le site de MSC-Electronics [2]. La version *Demo* suffit pour débiter, même si la taille du code compilable est bridée à 4 Ko (4096 octets). La mémoire de la Uno est de 32 Ko, mais il y a largement de quoi faire avec 4 Ko, une capacité par ailleurs amplement suffisante pour les exemples de cette série.

Installer l'EDI Bascom ne présente aucune difficulté. Le menu *File/New* ouvre une fenêtre d'édition vierge dans laquelle on peut taper son propre code. On peut bien sûr aussi y charger un programme Bascom déjà écrit (**fig. 5**). Un programme Bascom (d'extension `.bas`) est un simple fichier texte, et peut donc être ouvert et lu avec un éditeur de texte comme Notepad. On parle de « code source » car les caractères de ce code représentent la source à partir de laquelle le compilateur les traduit en code machine. Le fichier compilé qui porte l'extension `.hex` est parfois appelé « code hexa » car les octets qui y représentent le code source contiennent des valeurs codées en hexadécimal (chaque octet contient deux chiffres hexadécimaux). Nous reviendrons régulièrement sur les représentations binaire et hexadécimale des nombres traités par les microcontrôleurs, ainsi que sur leurs différentes notations.

Vous pouvez recopier le **listage 1** ou le télécharger depuis la page du site Elektor associée à cet article. Le fichier `UNO_LED1.bas` se trouve dans

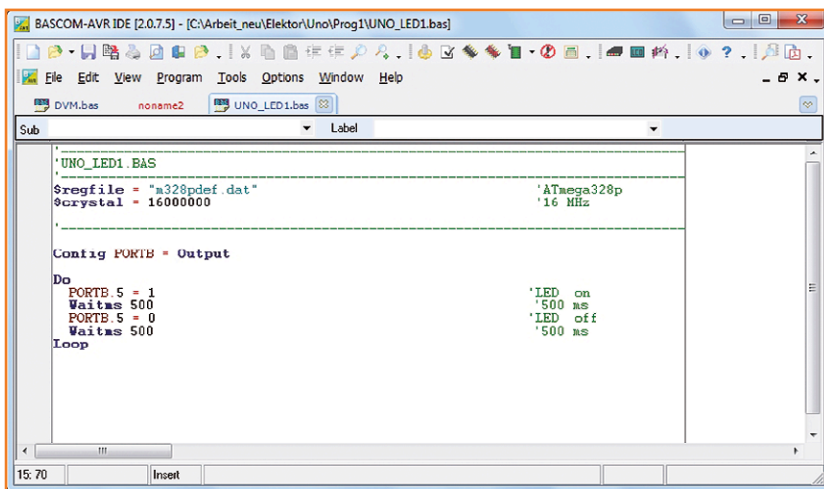


Figure 5.
Notre premier programme
dans l'éditeur de Bascom.

tension, ou après une initialisation, les broches de tous les ports sont configurées en entrées. Ici nous définissons le port correspondant en tant que sortie, car c'est ce dont nous avons besoin :

```
Config Portb = Output
```

Cette instruction définit les six broches du port B comme des sorties. Notre programme n'exploite cependant que la broche PB5, les autres n'ont aucun rôle. Pourquoi la PB5 ? Tout simplement parce qu'une LED jaune y est reliée.

```

Portb.5 = 1
Waitms 500
Portb.5 = 0
Waitms 500
  
```

L'instruction `Portb.5` met la ligne connectée à la LED au niveau haut, c.-à-d. sous une tension

l'archive [3]. Rien de plus simple que de compiler le programme, il suffit de cliquer sur *Program/Compile*, ou d'appuyer sur F7, ou encore de cliquer sur l'icône en forme de CI noir. Les erreurs éventuelles sont signalées dans la fenêtre inférieure. Si la compilation a réussi, une fenêtre affiche brièvement le pourcentage de mémoire utilisé. Elle est ici inférieure à 1 % et arrondie à 0 %.

Le compilateur a produit le fichier hexadécimal *UNO_LED1.hex* et le fichier binaire *UNO_LED1.bin*. Leurs formats sont différents mais leur contenu identique : le code exécutable. C'est ce code que nous devons maintenant « graver » dans la mémoire flash du contrôleur. Il existe plusieurs façons d'y parvenir, nous ne présenterons ici que la plus simple. Nous en verrons d'autres au cours des prochains épisodes.

Chargeur de démarrage

Pour être exécuté, le programme doit être présent dans la mémoire flash du µC. À l'instar de celles d'une EEPROM, les données d'une mémoire flash peuvent être conservées pendant plusieurs dizaines d'années sous forme de charges électriques. Et une mémoire flash peut toujours être reprogrammée (« flashée », c'est-à-dire programmée très rapidement), donc rien ne s'oppose à ce qu'un programme qui y est enregistré soit modifié ultérieurement. Les puces peuvent être programmées à l'aide d'un programmeur. Ce dispositif spécialisé se relie à des broches spécifiques du contrôleur, et écrit dans la mémoire flash les octets qu'il reçoit de l'EDI via l'interface USB.

Un programmeur n'est cependant pas indispensable. Si la carte à µC est équipée d'une interface USB, on peut l'utiliser pour transmettre le programme compilé du PC au contrôleur. Le contrôleur doit toutefois également posséder un petit programme capable de recevoir les octets compilés et de les écrire dans la mémoire flash. Ce programme est appelé chargeur d'amorçage, *bootloader* en anglais. Il était à l'origine appelé *bootstrap*, terme qui désigne un petit programme d'amorçage capable d'en lancer un plus gros. Le mot *bootstrap* est une référence à l'expression imagée *pull oneself up by one's bootstraps* (« débrouille-toi tout seul ») dont le sens littéral est : « se tirer par les bottes pour se soulever ». Dans la réalité, s'extraire p. ex. de la boue en se tirant par les bottes est bien sûr une idée de des-

sin animé. Et il en va de même pour un µC vide, incapable de se programmer lui-même. Mais avec un chargeur, c'est possible. Et l'Arduino en a un. Mais l'EDI Bascom peut-elle commander le processus de programmation via le chargeur d'amorçage ? Mark Alberts a heureusement pressenti qu'un jour quelqu'un souhaiterait programmer une Arduino en Bascom et a donc tout prévu. Tout ce que nous avons à faire est d'entrer les bons paramètres dans l'EDI. Nous décrirons le processus pour la version *Demo*, car elle présente une petite différence par rapport à la version complète.

Il faut d'abord télécharger et installer l'EDI Arduino [4]. Le pilote USB dont a besoin la carte

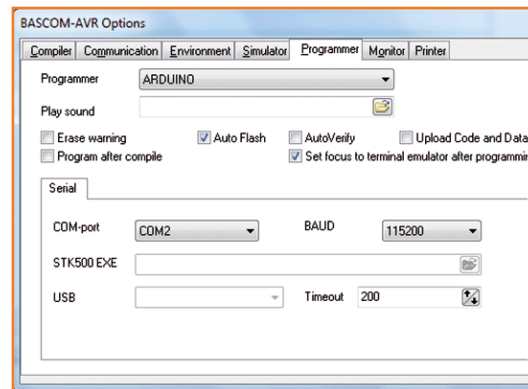


Figure 6.
Les paramètres pour lancer le chargeur d'amorçage Arduino.

Uno est fourni avec le programme. Il est chargé automatiquement lorsqu'on relie la carte au PC à l'aide d'un câble USB. La carte doit alors apparaître dans le *Gestionnaire de périphériques* de Windows, qui devrait aussi indiquer le port affecté à la carte (p. ex. COM2 ou COM3). Vous pouvez modifier ce numéro de port, mais en général ce n'est pas nécessaire. Quoi qu'il en soit, notez ou reprenez ce numéro, vous en aurez besoin. L'EDI Arduino contient de nombreux programmes d'exemples (menu *Fichier/Exemples*), soit dit en passant un bon moyen de découvrir le potentiel de la carte. Mais revenons à Bascom.

Le menu *Options/Programmer* de l'EDI permet de sélectionner divers programmeurs et chargeurs d'amorçage. Ici (**fig. 6**) nous choisissons ARDUINO (pas ARDUINO STK500/2). Nous devons également paramétrer le port COM (COM2 pour ma carte), ainsi que le débit (115200 bits/s, ou bauds) dans le menu déroulant BAUD. La vitesse de transmission doit être la même des

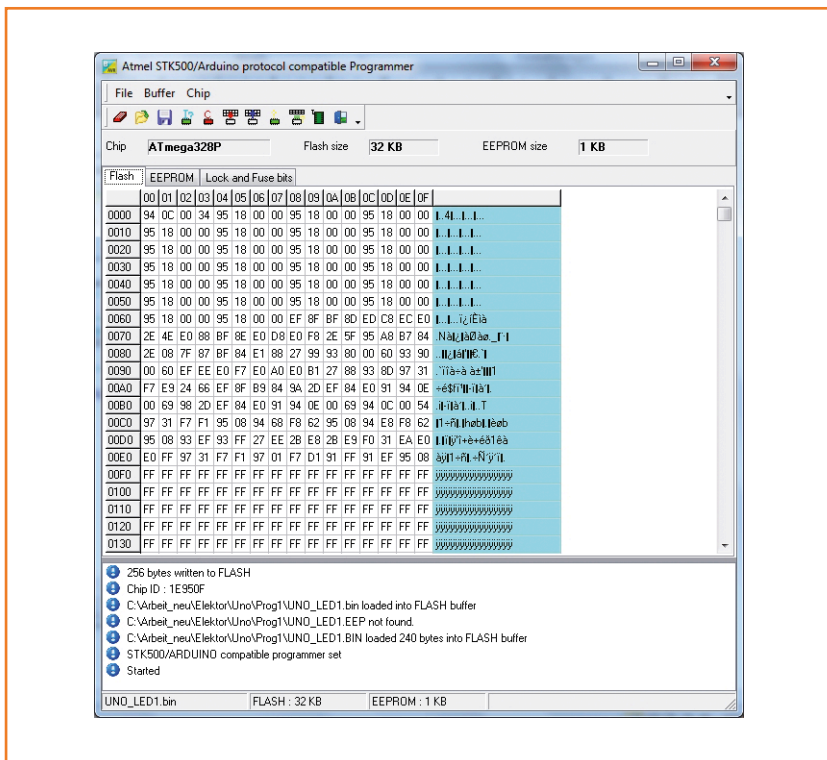


Figure 7.
Le programme compilé au
format hexadécimal.

deux côtés ! Nous reparlerons du débit dans un prochain article, lorsque nous verrons comment envoyer au PC du texte et des valeurs de mesure. Soyez attentif si vous vous en tenez à l'aide de Bascom et aux exemples fournis : les choses ont évolué depuis, les développeurs de la jeune carte Uno ont adopté le débit maximum de 115200 bauds. Bien sûr pour avoir un débit de données élevé, et donc une programmation ultrarapide de la mémoire flash. Tout le monde est pressé de nos jours.

Peu de risques de faire quelque chose d'incorrect avec l'option *Auto Flash* (fig. 6), donc nous la cochons ; elle nous épargnera plus tard l'effort d'un clic. Activons enfin *Set focus to terminal emulator*, une option qui s'avèrera très utile lorsque nous enverrons du texte et des données au PC. Si la fenêtre *Options* n'a pas de bouton Ok, appuyez sur la touche Echap pour la fermer. C'est la petite différence entre les versions *Demo* et complète dont nous avons parlé plus haut : certaines versions de démonstration n'ont pas de bouton Ok, mais dans un cas comme dans l'autre le bouton Echap permet de fermer la fenêtre. Voilà, nous sommes prêts à programmer le contrôleur. Compilons d'abord le programme avec

p. ex. la touche F7. Le programmeur se lance ensuite avec F4 ou avec l'icône verte en forme de CI appelée *Program chip*. Une nouvelle fenêtre s'ouvre (**fig. 7**) et affiche au format hexadécimal le fichier compilé (UNO_LED1.bin).

Ça marche !

Plus rien à faire, le processus est maintenant automatique. Cela dit rien ne vous empêche de lancer une programmation manuelle (mode *Manual Program*). Faites toutefois très attention, la fonction *erase* du menu *Chip* peut effacer tout le contenu de la puce, y compris le chargeur d'amorçage. Ne faites confiance qu'à la fonction *Write Buffer into Chip* du menu *Chip*. Les fonctions qui contiennent le mot *erase* sont souvent synonymes de grand malheur !

La fenêtre inférieure du programmeur affiche plusieurs messages, dont le dernier et le plus excitant est : *Started*. Signe d'un trafic de données intense entre Arduino et PC, les LED TX et RX de la UNO clignotent durant la programmation. La fenêtre disparaît à la fin du processus, le code nouvellement chargé est exécuté, et la LED jaune clignote. Le programme est petit mais le succès n'en est pas moins grand ! Et faites l'essai vous-même si vous en doutez encore, il est facile de modifier le programme. Vous verrez aussitôt le résultat. Changez p. ex. le délai de 500 ms en 100 ms, compilez votre code une nouvelle fois, et lancez la programmation de la carte. La LED clignote maintenant plus rapidement ? Bravo, vous êtes lancé, ça marche vite et bien ! Et nous aussi nous continuerons d'avancer à bon pas dans le prochain numéro !

(120574 – version française : Hervé Moreau)

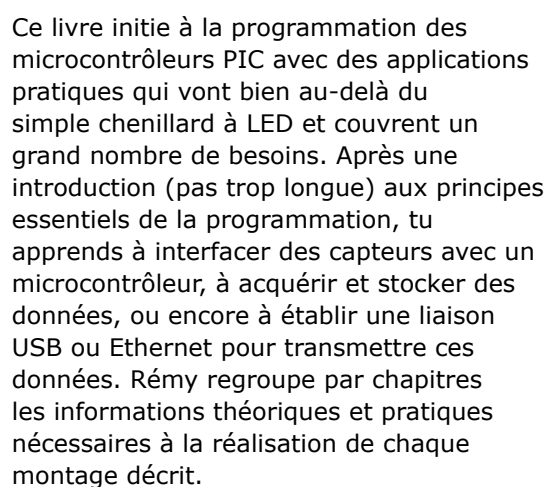
Liens

- [1] www.elektor.fr/arduino
- [2] www.mcselec.com
- [3] www.elektor-magazine.fr/120574
- [4] <http://arduino.cc/en/Main/Software>

les microcontrôleurs PIC pour les débutants

qui veulent programmer **sans patauger**

NOUVEAU



Après la lecture, gagné par la bonne humeur communicative de l'auteur et fort de sa longue expérience (qu'il partage volontiers), tu n'auras qu'une seule envie : aller plus loin, créer toi-même des montages encore plus ambitieux (transposer le code dans d'autres langages, le porter sur d'autres plates-formes de développement, t'attaquer aux PIC32)...

**Le premier pas coûte,
après ça va tout seul.**



elektor

www.elektor.fr/debutPIC



9^e jour – contours personnalisés (suite)

Neil Gruending
(Canada)

Finissons le contour de carte commencé la dernière fois.

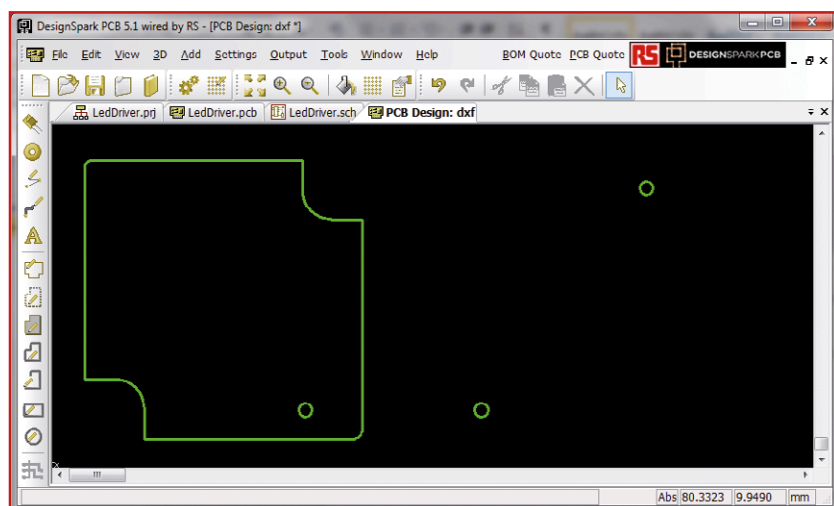
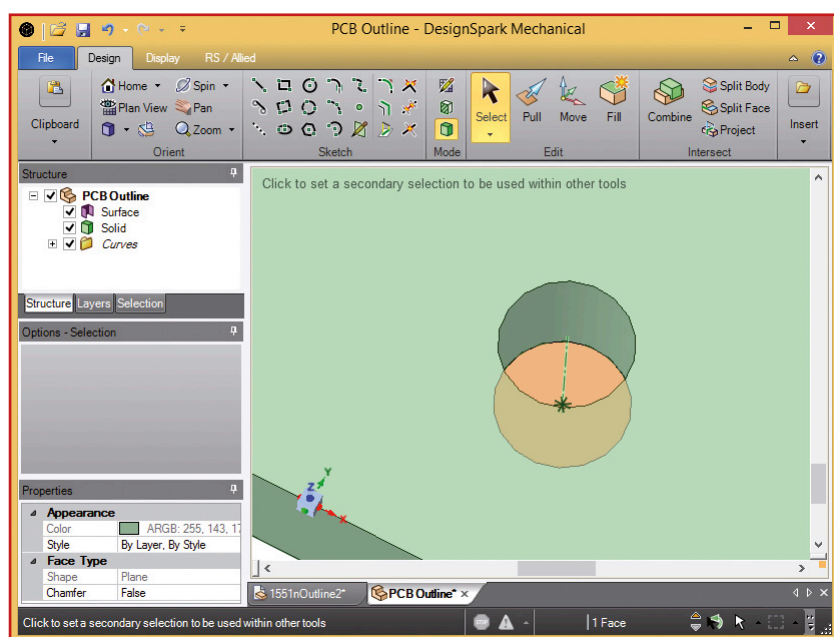


Figure 1. Import *DesignSpark PCB* de la dernière fois. Les trous sont mal placés.

Figure 2. Il faut effacer le plan recouvrant les trous de montage.



La dernière fois nous avons commencé un contour de carte pour un boîtier Hammond 1551N. Nous avons tout d'abord importé le modèle STEP dans *DesignSpark Mechanical* puis avons modélisé le circuit imprimé pour qu'il rentre dans le boîtier. Aujourd'hui nous allons terminer l'importation de la carte dans *DesignSpark PCB* afin qu'elle soit prête à accueillir ses composants.

Correction de l'importation

La **figure 1** montre où nous nous étions arrêtés. Le contour du C.I. est correct, mais pas les trous de montage, mal placés. D'autres logiciels que j'ai essayés sont capables d'importer le fichier DXF créé la fois dernière. L'export en DXF depuis ces logiciels crée un fichier importable par DS PCB. Il y a donc dans notre DXF d'origine quelque chose que DS ne digère pas.

Après quelques essais j'ai compris que j'avais fait quelques erreurs en créant le modèle avec *DS Mechanical*. Le premier problème concerne les trous de montage qui n'étaient pas vraiment des trous dans le modèle de la carte : ils étaient recouverts par une fine couche. Pour le voir, il m'a fallu faire pivoter le modèle : le fond des trous est apparu ombré au lieu d'être blanc. Le deuxième problème est que le contour de la carte n'est pas considéré comme clos par DS PCB, lequel rechigne après l'import. En déplaçant les coins, j'ai remarqué que deux d'entre eux n'étaient pas reliés correctement à cause de lignes qui ne voulaient pas se redimensionner correctement.

Commençons par corriger les trous de montage dans le modèle *DS Mechanical* du C.I. Utilisez l'outil *pull* pour tirer le contour du PCB de 1,6 mm au-dessus du plan *sketch*, cela transforme notre dessin en un solide. Sélectionnez ensuite les cercles du plan *sketch* d'origine et effacez-les (**fig. 2**).

Sélectionnez soigneusement le cercle recouvrant le trou de montage comme un solide et pas juste comme un cercle que vous ne pourriez pas effacer correctement. Si vous voyez le fond blanc à

travers le trou, il est bien vide et vous avez réussi. Une fois les trous vides, cliquez avec le bouton droit sur *Curves* dans la structure du contour puis choisissez *delete* pour effacer les lignes dont nous n'aurons plus besoin.

Avant d'exporter le nouveau DXF il vous faudra sélectionner la vue de dessus du C.I. (*Design→Orient→Top*). Si vous ne le faites pas, *DS Mechanical* exportera une vue en perspective en 3D et cela ne marchera pas. La **figure 3** montre l'aspect de la carte corrigée avec le nouveau DXF dans DS PCB. Il affiche toujours le même avertissement lors de l'importation, mais cela est dû aux trous de montages. Comme nous allons les supprimer plus tard, ce n'est pas un problème.

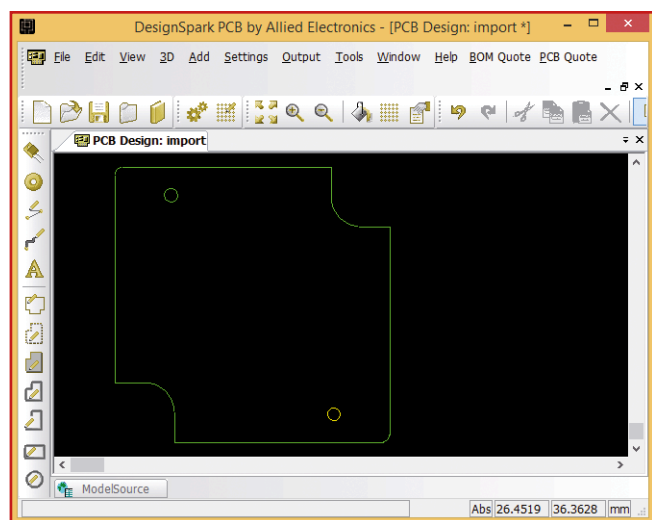
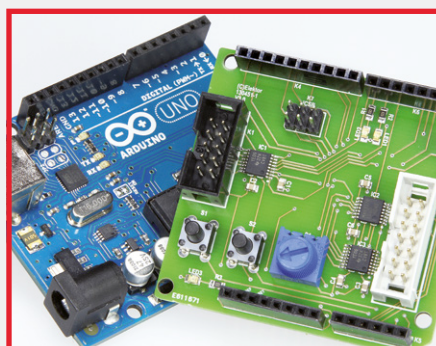


Figure 3. Import de la carte corrigée.

Ajout des trous de montage

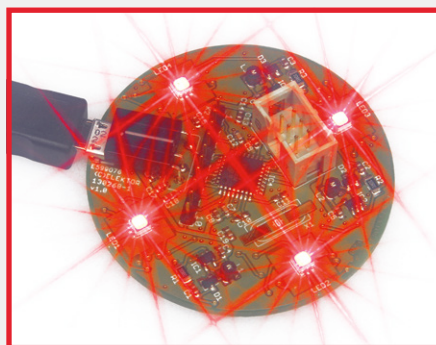
Habituellement, je place mes trous de montage de sorte qu'ils soient ni métallisés ni entourés de pastilles de cuivre qui pourraient entraîner un arrachement du cuivre lors du serrage des vis de montage. Ces types de trous existent habituellement dans les bibliothèques de composants à cause des zones d'exclusion requises sur les couches extérieures. DS PCB peut le faire, mais c'est difficile : on ne peut pas ajouter de zone d'exclusion à un composant de bibliothèque DS PCB. Il faudrait donc les ajouter à la main pour chacun des trous. Aujourd'hui nous nous contenterons d'un trou métallisé, bien plus facile à utiliser dans DS. Vous pouvez vous passer de métallisation, mais les pastilles sont alors plus fragiles et le risque d'arrachage du cuivre plus grand. Commencez par créer un nouveau symbole PCB pour notre trou de montage. Nous allons utiliser des vis ANSI #2. Ajoutez donc une pastille de diamètre 5,5 mm avec un trou de 2,5 mm ; cela correspond à la taille recommandée [1] pour une vis à tête cylindrique. Ces dimensions conviennent également à une vis métrique M2. Une fois l'édition de la pastille terminée, ajoutez-la à votre bibliothèque. Vous devrez

•bientôt dans Elektor



**Arduino :
shield pour
l'expérimentation**

Cette interface d'expérimentation, simple mais polyvalente, servira p. ex. dans la série d'articles « l'atelier du microcontrôleur » qui vient de démarrer. Elle est compatible avec la plupart des cartes Arduino : 2 poussoirs, 2 LED, 1 potentiomètre, des adaptateurs de niveau et un connecteur universel ECC.



**mini lampe RVB
à commande IR**

L'éclairage à LED polychrome est à la mode. On peut acheter ces lampes toutes faites mais l'électronicien, lui ne résiste pas à l'envie d'en bricoler une lui-même. Quatre LED en série alimentées sous 5 V au moyen d'un convertisseur de tension, et... la lumière fut ! Une banale télécommande IR au standard RC5 permet d'en varier la luminosité.



**la 3^e dimension
pour les dalles tactiles**

Le principe est de type capacitif projeté : un champ électrique projeté par deux antennes devant l'interface tactile existante est perturbé par les mouvements en l'air de la main, à plusieurs centimètres de la surface. Ces perturbations affectent un oscillateur qu'il suffit de suivre pour analyser les mouvements. Un logiciel de reconnaissance de gestes fait le reste.

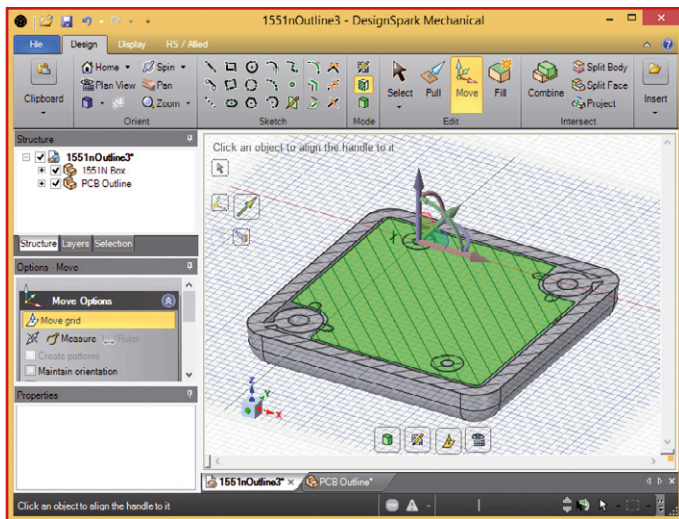


Figure 4. Création d'une vue en coupe avec les zones d'exclusion.

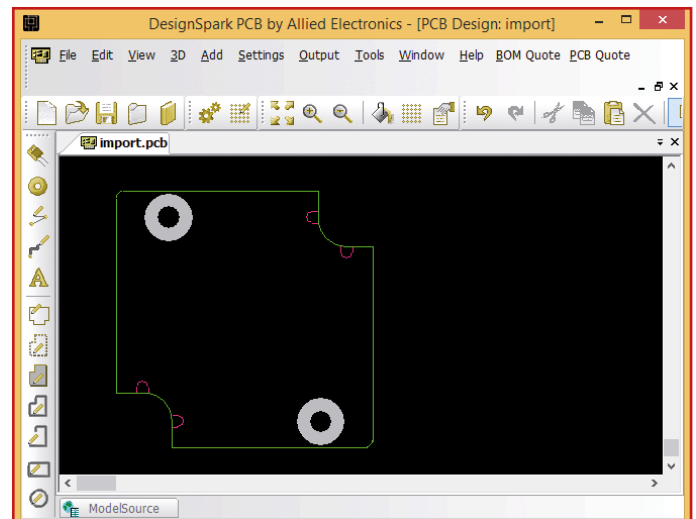


Figure 5. Le C.I. avec les trous de montage et les zones d'exclusion.

ensuite créer un composant pour C.I. pour le trou de montage. Cela se fait en ajoutant un nouveau composant à la bibliothèque, comme d'habitude, sauf qu'ici vous décochez la case *Schematic Symbol* de la fenêtre *New Component*. Vous pourriez également l'ajouter au fichier de technologie du C.I., mais si vous utilisez plusieurs de ces fichiers il faudrait faire répéter l'opération pour chacun d'entre eux. Avec une bibliothèque, il n'y a qu'une copie du trou.

Vous pouvez maintenant ajouter deux trous de montage au projet. Je les ai placés manuellement sur le C.I. puis ajustés vers leur emplacement en saisissant des coordonnées identiques à celles des trous importés. Pour déterminer les coordonnées des trous importés, cliquez sur l'un d'entre eux avec le bouton droit, sélectionnez *Properties* et notez les coordonnées du centre. Vous pourrez saisir les coordonnées du composant des trous en cliquant avec le bouton droit puis en choisissant *Type Coordinate*. Une fois les nouveaux composants correctement placés, vous pouvez effacer les trous importés.

Ajout des zones d'exclusion

La dernière chose que nous ferons aujourd'hui sera l'ajout des zones d'exclusion autour des entretoises de montage. Dans DS PCB, cela se fait en les dessinant sur la couche de documentation et en y faisant attention vous-même : DS

ne sait pas utiliser le DRC pour ça.

Il nous faudra tout d'abord utiliser *DS Mechanical* pour créer une vue en coupe dans laquelle la carte croiserait les entretoises de montage du boîtier (**fig. 4**). Il faudra ensuite créer un nouveau composant contenant le contour de la carte et les trous et entretoises de montage. Pour aujourd'hui, il n'y a aucun avantage à faire une forme en 3D : contentez-vous d'exporter en DXF la forme en 2D. Cela engendrera des erreurs lors de l'importation des trous de montage, mais rien de bien grave. Assurez-vous que la couche inférieure de documentation est bien sélectionnée dans le champ *To Design Layer* lors de l'importation du DXF dans DS PCB.

Après quelques retouches, le résultat final dans DS PCB ressemblera à la **figure 5**. Il y a maintenant de la place pour les têtes de vis autour des trous de montage et les zones d'exclusion sont visibles sur la couche de documentation inférieure, représentée en rose.

Conclusion

Aujourd'hui nous avons corrigé nos problèmes d'importation et fini d'ajouter les informations mécaniques nécessaires à notre C.I. La prochaine fois, pour nous assurer que nous n'avons fait aucune erreur, nous le réimporterons dans *DesignSpark Mechanical*.

(130463 - version française -- Kévin PETIT)

Lien : [1] <http://blogs.mentor.com/tom-hausherr/blog/tag/pcb-mounting-holes>

hexadoku

vosre séance mensuelle d'hexadécimalie !

Lâchez tout ! Sautez dans vos bottes, prenez votre bouée, votre masque, votre tuba, votre brosse à dents (on ne sait jamais), une couverture chaude (pour après), un crayon, surtout n'oubliez pas la gomme et l'indispensable taille-crayon, empoignez la gueuse, respirez un grand coup et plongez, c'est l'heure de votre séance d'immersion en hexadécimalie profonde. Vous en sortirez ragaillardi(e). Remplissez la grille selon les règles, envoyez-nous votre solution, et vous gagnerez peut-être l'un des cadeaux offerts par Elektor.

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

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

Participez et gagnez !

Nous tirerons au sort **cinq bonnes réponses** internationales reçues dans les délais ; leur auteur recevront chacun un chèque-cadeau d'une valeur de **50 €** à valoir sur des livres d'Elektor. À vos crayons !

Où envoyer ?

Envoyez votre réponse (les chiffres sur fond grisé) avec vos coordonnées par courrier électronique exclusivement avant le **1^{er} mai 2014** : **hexadoku@elektor.fr**

Les gagnants

La solution de la grille du numéro de janvier-février (427-428) est : **1F734**

Les gagnants sont **Arun Annaji** (Inde), **Per Troelsen** (Danemark), **Arwin Vosselman** (Pays-Bas), **Brian Unitt** (Royaume-Uni) et **Udo Altmann** (Allemagne).
Bravo à tous et félicitations aux gagnants !

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

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

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.

récepteur Atwater Kent model 30 (1926)

Gerard Fonte
(États-Unis)



Au milieu des années soixante, j'étais adolescent et commençais à m'intéresser à l'électronique. Je venais de lire quelque chose au sujet de la restauration de vieux récepteurs de radio quand je vis sur le trottoir un meuble récepteur de radio Silverstone monstrueux qui attendait les éboueurs. Hélas, je ne pouvais pas demander au chauffeur du bus de s'arrêter. De toute façon, qu'aurais-je fait dans le bus de ce machin de 1,30 m de haut sur 60 cm de large ! Il pesait au moins 20 kg. Élève plein de ressources, j'ai appelé mes parents depuis l'école et les ai convaincus d'aller le récupérer illico. Avec l'aide d'un ami, j'ai pu remplacer le condensateur de filtrage en court-circuit qui faisait rougir les plaques du tube redresseur (pour ceux qui ne sont pas familiers des tubes, seul le filament doit rougir). La radio avait une sonorité et un volume fantastiques. Mieux encore, il avait plusieurs gammes d'ondes courtes – très important pour un ingénieur en herbe à l'époque. J'ai ensuite continué à collectionner de vieilles radios et cette sorte de choses.

Un âne dans un sac

Puis vint ma première vente aux enchères ; un grand meuble radio en triste état, le bois

vermoulu, l'arrière fermé. Impossible de voir l'électronique. Mise à prix : 200 \$. À tout hasard, je proposai deux dollars. Personne n'enchérit. Je l'emportai donc pour deux dollars. Une seule enchère, gagnée ! Pas mal pour une première fois. Le meuble était presque aussi haut que moi et en le traînant vers la voiture, j'entendais quelque chose brinquebaler à l'intérieur. Quand j'ouvris le dos, je découvris un impeccable récepteur *Atwater Kent Model 30*. Quelle affaire ! Et dire que je n'avais que deux dollars en poche.

La base de l'AKA-30

L'*Atwater Kent Model 30* a été construit vers 1926. C'est un poste à détection directe (*Tuned Radio Frequency*) à six tubes. C'est-à-dire qu'il fonctionne comme un poste à galène avec amplificateur. On voit sur le schéma redessiné, **fig. 1**, trois étages amplificateurs à haute fréquence accordés, suivis d'un détecteur par la grille, suivi de deux amplificateurs audio (BF). Soit six étages avec un tube pour chacun.

L'accord des trois amplificateurs HF est effectué par un simple circuit L-C parallèle (inductance-condensateur) exactement comme un récepteur à galène. Les trois condensateurs variables à air servaient à l'accord. Le montage en cascade de ces trois amplificateurs accordés augmentait le gain (évidemment) et la sélectivité (l'aptitude à séparer des signaux proches

en fréquence). Chaque étage accordé rétrécit considérablement la bande passante. Avec trois étages, la sélectivité est plutôt bonne. À l'époque, les émetteurs étaient moins nombreux et n'interféraient pas. Cela signifiait aussi que chaque condensateur devait être réglé correctement pour capter une station. Le *Model 30* était le premier *Atwater Kent* à disposer d'un cadran unique pour l'accord simultané des trois circuits. C'était réalisé par le couplage des trois condensateurs au moyen de poulies et d'une bande métallique. Il y avait une commande de gain HF qui agissait sur la tension de filament des trois amplificateurs HF. La détection par la grille (en anglais *grid leak detector*, « à fuite de grille », voir l'encadré) était un moyen simple et économique de démoduler un signal HF. Le procédé consiste à appliquer une faible polarisation à la grille, par une forte résistance entre elle et la masse. La résistance du *Model 30* était d'environ 2 MΩ. Son effet est de déplacer le point de fonctionnement (la grille est normalement à un potentiel négatif par rapport à la cathode) du tube afin de n'amplifier que la moitié négative du signal. De cette façon, la détection s'accompagne d'une amplification. Bien que cela fonctionnât, de nombreux facteurs affectaient les performances. Parmi eux la force du signal, la contre-réaction du tube et les variations de la résistance de grille (ces composants dériveraient avec la température et l'humidité). Après l'étage à « fuite de grille » venaient deux

EST^D 2004

Rétroronique 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 téléphorier à redaction@elektor.fr

Fuites ! Ploc, ploc !

La résistance de fuite de grille, bien réelle, a mené directement au développement d'une « lèchefrite pour fuites de grille » (fictive) censée récupérer tous ces électrons, pour les empêcher de se répandre sur le plancher. Comme la clé à molette pour gaucher et la lime à épaissir, cet instrument a fait la joie de tout le monde, sauf de l'apprenti nouveau venu, lancé dans une recherche vaine à travers tout l'atelier. Des rumeurs prétendaient que ces ustensiles existaient et que des margoulins en avaient vendu à des clients naïfs. On trouve bien aujourd'hui des cordons secteur « audiophile » à 1 699,99 dollars.

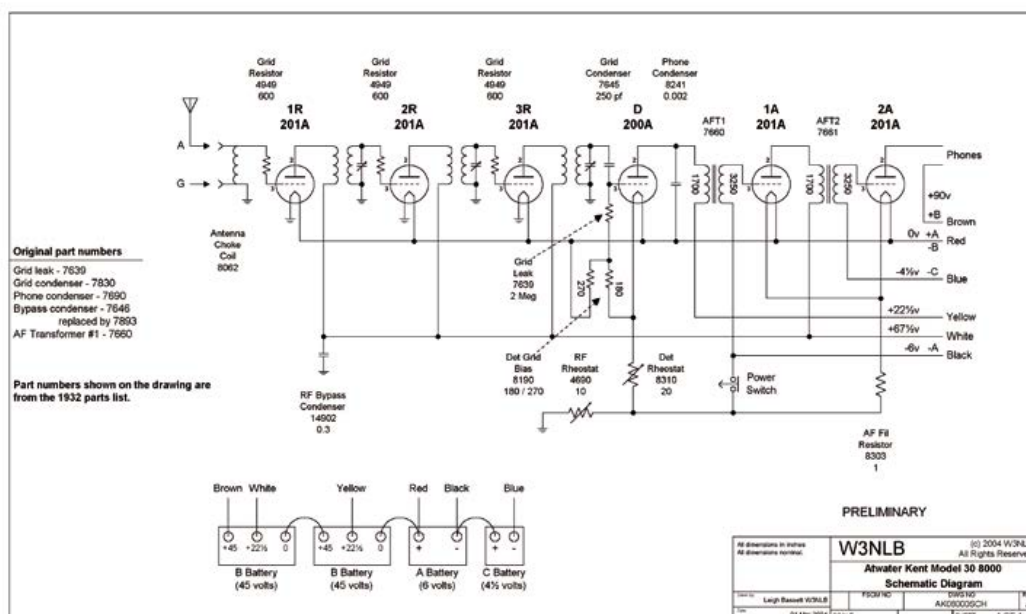
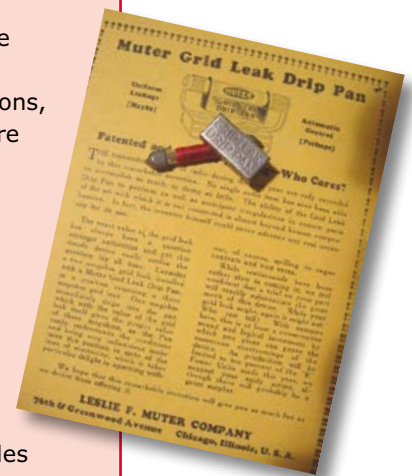


Figure 1.
Le schéma de l'AK-30,
redessiné, avec l'aimable
autorisation de Leigh,
W3NLB et Atwaterkent.info.



Figure 2.
Vue de l'AK-30, châssis sorti
du coffret.

étages d'amplification audio. Le résultat final était un signal capable d'attaquer confortablement des écouteurs ou un haut-parleur faiblement. Un haut-parleur de l'époque n'était pas grand-chose de plus qu'un écouteur monté sur un cornet acoustique. La commande de volume faisait varier la tension de filament du détecteur au lieu de la tension d'entrée des étages amplificateurs BF (ce qui est la norme aujourd'hui).

Mon AK-30

Le coffret était quasi parfait et tous les tubes étaient en place (**fig. 2**). Toutefois il y avait quelques problèmes. Les poulies métalliques qui couplaient les condensateurs variables tombaient en miettes (**fig. 3**, merci Allen [2]). Je n'avais



Tubes

Pour comprendre les radios de l'époque, il faut connaître un peu les tubes. D'abord, ils étaient chers : jusqu'à 9 \$ la pièce au début des années 20, soit 120 \$ d'aujourd'hui. En 1930, le prix était tombé à 1,5 \$ (20 \$ actuels). Un poste des débuts pouvait coûter le prix d'une Ford T. Un poste du milieu des années 20, comme le *Model 30*, était un investissement appréciable.

Il y eut des variations dans la fabrication des tubes, mais finalement la plupart des fabricants se rallièrent au culot à quatre broches longues. Tous les tubes avec

le même culot étaient en général interchangeables.

Par exemple, les UX-201, UX-201A, UX-301, UX-301A, etc. étaient grosso modo similaires. Je dis grosso modo parce que le courant de chauffage des derniers tubes était notablement inférieur. C'était très important parce que ces radios fonctionnaient sur piles. Les filaments 5 V du début consommaient

carrément 1 A. Les versions postérieures sont tombées à 0,25 A. Avec six tubes, l'alimentation des filaments venait d'un accumulateur de voiture de 6 V.



jamais rien vu de pareil auparavant (ni après d'ailleurs). Je suppose que c'est dû à la fatigue du métal. Il s'agit de métal fritté et non moulé ou usiné. (Le métal fritté est au départ une poudre qui est pressée et chauffée directement dans sa forme finale.) Les écarts de 40 °C en été à -20 °C en hiver sont une cause probable.

Deuxième problème : l'un des tubes avait le filament coupé. Aujourd'hui il est facile d'en trouver sur eBay ou quelque site de collectionneurs. Quand j'ai eu ce récepteur, l'internet n'existait pas. Essayer de repérer un tube qui n'était plus fabriqué depuis des décennies n'était pas une tâche quelconque.

Le dernier problème n'en était pas vraiment un. Les deux tubes BF n'étaient pas du modèle spé-

Les filaments étaient câblés en parallèle, donc il n'y avait pas d'inconvénient à monter des tubes avec des courants de chauffage différents.

Les tubes ne devaient pas contenir d'oxygène. Sinon, le filament se serait oxydé (consommé !) et coupé. Il y avait deux méthodes pour éviter la présence d'oxygène. La première consistait à faire le vide. Malheureusement, le vide à cette époque n'était pas très poussé. Aussi ajoutait-on dans l'enveloppe un « getter » pour capturer le reliquat d'oxygène. Le getter était un composé chimique qui réagissait avec tout l'oxygène libre et le fixait. Les getters donnaient une coloration à l'intérieur de l'enveloppe du tube. La coloration était différente suivant la nature du getter, généralement noir-argenté, mais certains étaient bleu vif et d'autres irisés. La deuxième méthode consistait à remplir l'enveloppe d'un gaz inerte, en général de l'argon. Cela éliminait le getter et l'enveloppe était parfaitement claire. Naturellement, le problème de l'argon est qu'il amorce à haute tension et qu'il rayonne d'un beau bleu (très semblable au néon classique). Pour une raison ou une autre, les tubes détecteurs (UX-200A) étaient à l'argon et les tubes HF (UX-201A) avaient des getters. Peut-être parce que les détecteurs travaillaient sous une tension plus basse et risquaient moins l'amorçage.

cié. C'étaient apparemment des types de remplacement. Tous les tubes de l'époque étaient de simples triodes et beaucoup étaient interchangeables. Ils pouvaient amplifier d'un facteur 8 environ. (Le gain d'un transistor 2N2222 est de 50 au minimum.) Au lieu de deux tubes UX-201A, il y avait un UX-200A et un SX-112A (fig. 4). Le SX-112A était un tube plus récent conçu pour la sortie audio. Il pouvait délivrer rien moins que 200 mW de puissance.

Piles

Le *Model 30* a besoin de cinq tensions continues : 90 ; 67,5 ; 25,5 et - 4,5 V. Elles sont fournies par des piles. Les 90 V alimentent le casque ; les 67,5 V vont aux plaques des amplificateurs HF et



du premier amplificateur BF. Les 6 V alimentent les filaments. Le - 4,5 V polarise l'amplificateur BF de sortie (le débit est négligeable et la pile peut durer des années). Évidemment, les piles devaient être remplacées régulièrement, ce qui fait que le coût d'utilisation des premiers postes de radio n'était pas négligeable. C'est seulement quelques années plus tard que les postes alimentés en alternatif devinrent abordables, grâce au développement de tubes redresseurs capables de répondre aux exigences de puissance.

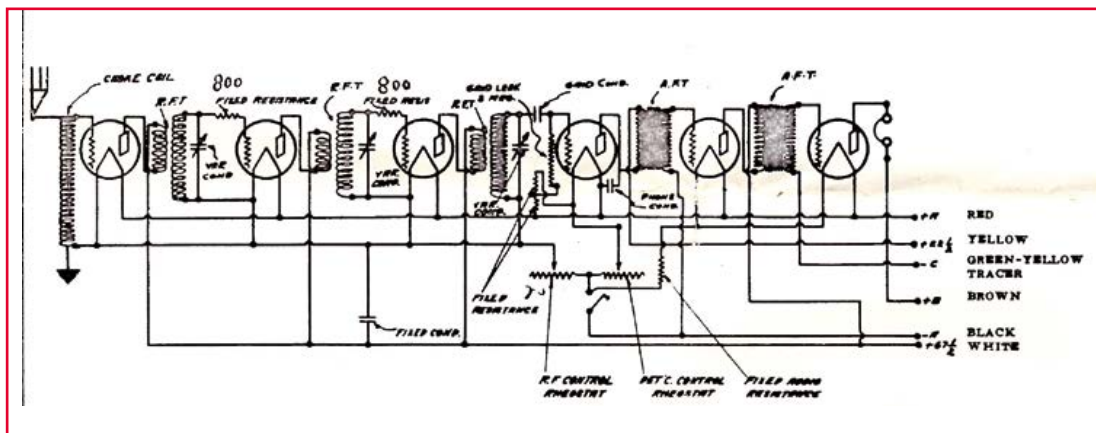
Figure 3.
Poulies désagrégées.
Photo reproduite avec
l'autorisation d'Allen Wooten,
WD4EUI.

Détails de la construction

Les bobines HF offrent quelques détails de construction intéressants. L'arrière du coffret est évidé à trois endroits sur environ 2,5 cm pour laisser de la place aux bobines. La figure 5 montre le châssis sorti du coffret. Cela ressemble à une erreur de conception. Je pense que le cof-

Figure 4. Jeu de tubes de
mon AK-30.





fret aurait dû être conçu dès le départ pour loger les bobines.

Le deuxième détail est que les bobines sont montées à angle droit l'une par rapport à l'autre (figure 6). Le but est de limiter le couplage induc-

tif entre elles. Sans cela, il y aurait probablement des fuites qui provoqueraient réaction et oscillation. Les oscillations étaient un souci sérieux dans les premiers récepteurs.

Enfin, les bobines sont en fait des transformateurs. Un autre enroulement se trouve à l'intérieur du mandrin. Je ne sais pas si c'était la méthode standard à l'époque de l'Atwater Kent. C'est en tous cas une technique inhabituelle.

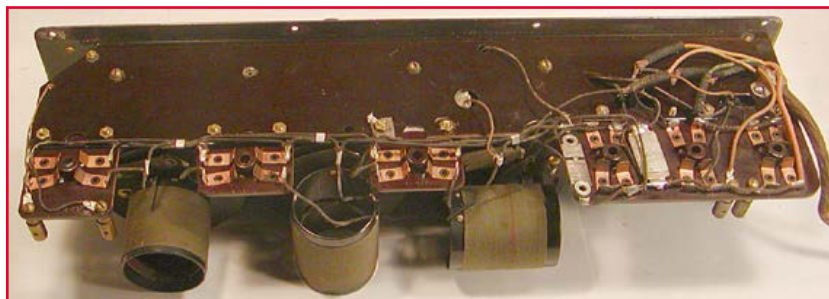


Figure 5. On voit clairement ici les bobines d'accord qui dépassent du châssis.

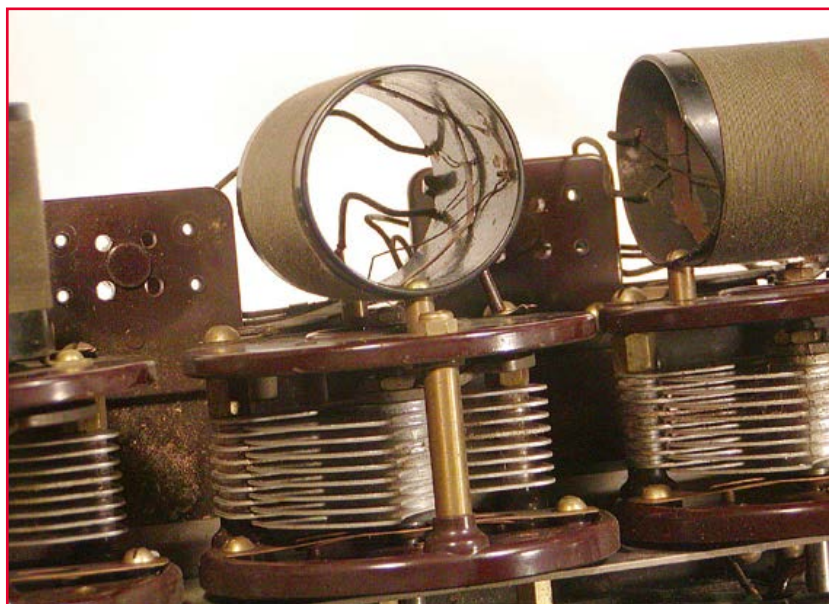


Figure 6. Détail du montage des bobines perpendiculaires l'une à l'autre.

Tourner la page

Les postes de radio à amplification directe avaient pratiquement disparu en 1930. Ils étaient remplacés par des circuits à réaction qui apportaient un gain élevé et une meilleure sélectivité avec moins de tubes, ou par les superhétérodynes comparables au récepteur d'aujourd'hui. L'alimentation par le secteur est devenue la norme et les prix des postes ont plongé.

Malheureusement, ma collection de radios géantes n'a pas survécu à mon départ pour la fac et un petit appartement. Des trois pièces particulières que j'avais (le *Silverstone*, l'*AK-30* et l'*Ediphone*), je n'ai pu garder que l'*AK-30* et l'*Ediphone*. J'ai donné le *Silverstone* colossal à ma sœur aînée, mais elle n'a pas apprécié le cadeau : au bout de quelques mois, un beau jour, il avait disparu. Tant pis. Aujourd'hui, un poste en état physique et électrique quasi parfait va chercher dans les 1000 dollars.

130426 – version française : Jean-Paul Brodier

Liens

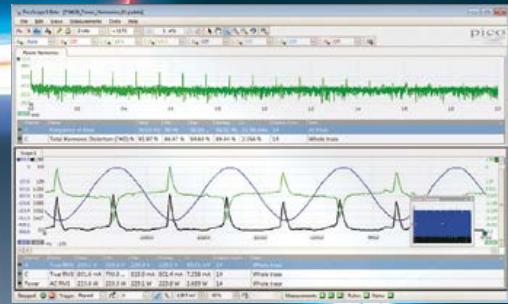
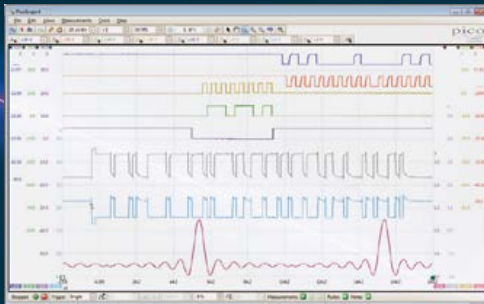
[1] www.atwaterkent.info

[2] www.wd4eui.com

7 fonctions de PicoScope:

1. Oscilloscope
2. Analyseur de spectre
3. Générateur de fonctions
4. AWG
5. Analyseur logique
6. Analyseur de Protocole Série
7. Test de forme d'onde automatique

OSCILLOSCOPE PC 8 CANAUX



Pour seulement €1688

- Haute Résolution
- Alimenté par USB
- Mémoire Profonde



COMPREND DES MESURES AUTOMATIQUES, ANALYSEUR DE SPECTRE, SDK COMPLET, DÉCLENCHEMENTS AVANCÉS, PERSISTANCE DE COULEURS, DÉCODAGE SÉRIE (CAN, LIN, RS232, I²C, I²S, FLEXRAY, SPI), MASQUES, MATHS, LE TOUT EN STANDARD. MISES À JOUR GRATUITES.

12 bit • 20 MHz • 80 MS/s • Mémoire 256Mé • AWG 14 bit

www.picotech.com/PS253

Votre partenaire en ligne compétent pour

Eléments de construction
Alimentation électrique
Technique de mesure

Technique atelier & de brasure
Technique domotique & sécurité
Technique réseau

Technique PC
Technique Sat/TV
Communication

Pas de main libre ? Pas de problème !

Grâce au capteur HF intégré, cette lampe s'allume par pur mouvement !

Lampe LED de détection des mouvements

Les micro-ondes qui se déclenchent fonctionnent également au travers du verre, du plastique ou des murs peu épais.



- ✓ Longue durée de vie
- ✓ Faible consommation d'énergie
- ✓ Faible besoin d'entretien

- Remplace l'ampoule 40 W classique
- Culot : E27
- Angle de détection 360°
- Rayon max. de détection 8 m
- 7 W, 460 lumens, 3000 K, blanc chaud



18,95
LED E27 7W HF



NOTRE BESTSELLER

Projecteur SMD-LED



- Remplace 35 W lampe halogène
- Puissance : 4 W
- Durée de vie : 20.000 heures
- 290 lm, blanc chaud

DELOCK 46337 **5,25**



- Remplace 20 W lampe halogène
- Puissance : 2,5 W
- Durée de vie : 30 000 heures
- 180 lm, blanc chaud

DELOCK 46345 **4,45**



Projecteur intégré LED



- Remplace l'ampoule de 20 W
- Puissance : 2 W
- 170 lm, blanc chaud

GB 30588 **3,60**



Lampe LED



- Remplace l'ampoule classique 20 W
- Puissance : 3 W
- Durée de vie : 25.000 heures

HEIT 16966 **5,10**



Ampoule LED

pour pratiquement toutes les applications, vous trouvez en ligne:



<http://rch.it/led>

Projecteur LED

en aluminium résistant aux intempéries, pour l'intérieur et l'extérieur

- Verre de protection trempé
- Durée de vie : 30.000 h.
- Angle de rayonnement : 120°
- Réglable par étrier métallique



10 W, blanc froid, 920 lm HEIT 37083

10 W, blanc chaud, 870 lm HEIT 37009



14,95

14,95

HEITRONIC®

Lampe globe SMD-LED

Efficacité énergétique classe A

- Remplace l'ampoule 40 W classique
- Culot : E27
- Durée de vie 30.000 h.
- Couleur de lumière : blanc chaud
- 6,7 W, 470 lumens, 3200 K



5,95
GB 30285



FAIRE DES ÉCONOMIES
D'ÉNERGIE
À PETIT PRIX!

++ BAS PRIX ++

Commander maintenant! www.reichelt.fr

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

Modes de paiement internationaux:



Prix du jour ! Prix à la date du : 10. 02. 2014

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