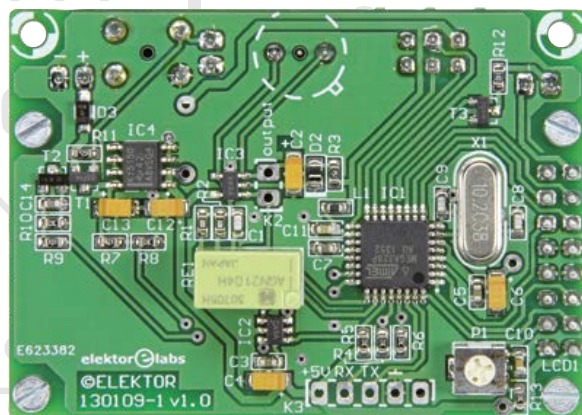
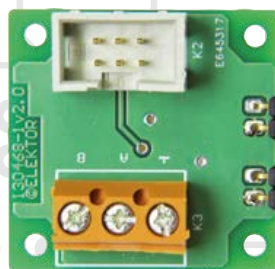
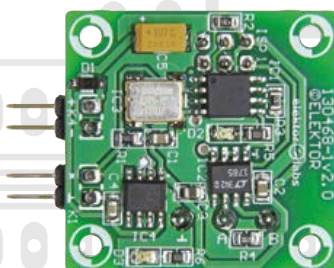
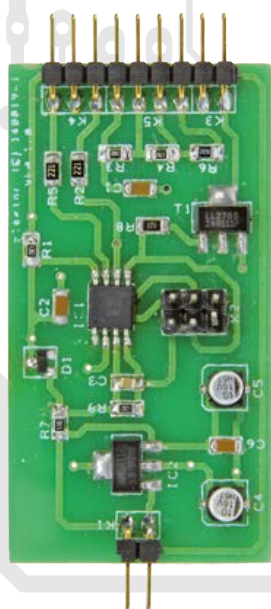
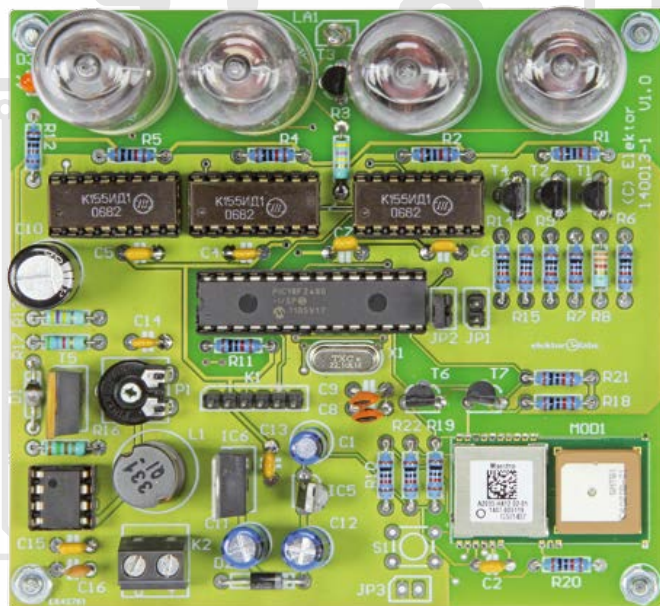


elektor



luxmètre de 1 lux à 100 000 lux | module capteur de température à interface RS-485
transformateurs à circuits gravés | **gradateur PWM pour baladeuse à LED**
antenne ferrite à gyrateur | rétronique BK Precision BK560 | programmer avec Mono
l'atelier du microcontrôleur (6) | **horloge Nixie à récepteur GPS**
banc d'essai : **débogueur/programmeur Atmel-Ice**

L 19624 - 436 - F: 7,80 € - RD





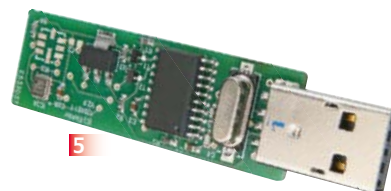
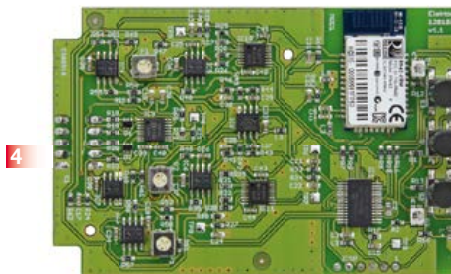
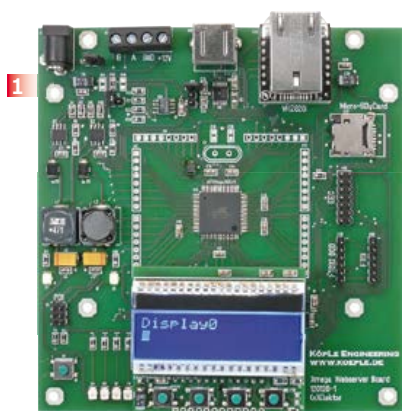
PENDANT QUATRE JOURS, L'AVENIR EST ICI, MAINTENANT.

Le CES International 2015. L'élite dans l'innovation, le business et la technologie, et le seul spectacle sur Terre qui promet le futur. Et ceci d'une manière dont jamais vous ne l'auriez imaginé. Inscrivez-vous maintenant sur CESweb.org.

2015 International
CES
THE GLOBAL STAGE FOR INNOVATION

PRODUCED BY
 CEA

JANVIER 6 - 9, 2015
LAS VEGAS, NEVADA
CESWEB.ORG
#CES2015



1 Xmega sur carte polyvalente

La voici enfin, la carte à microcontrôleur qui mesure, pilote et régule à qui mieux mieux ! Branchons-y un module TCP/IP, il en sortira un serveur internet ou d'autres applications en réseau. Sa mémoire de masse réside dans une micro SD. Pour s'en servir, il y a quatre boutons, autant de LED et un afficheur amovible. Et surtout plein d'interfaces sur la même carte.

Module contrôleur assemblée, testée prête à l'emploi réf 120126-91

Module d'affichage réf 120126-92

2 Andropod

Commandez vos montages

avec un smartphone ou une tablette Android

Écran tactile à haute définition, méga puissance de calcul, connexion réseau et fonctions de téléphonie sans fil : les téléphones tactiles et les tablettes Android seraient des centrales de commande

presque idéales pour nos projets d'électronique s'il était plus facile de s'y connecter. Voici Andropod, votre interface sérieuse TTL et RS485. Décollage vertical garanti ! L'offre de matériel Android (par différents fabricants) et de logiciel est énorme et le code source du système d'exploitation est libre. Le puissant kit de fonctions logicielles ou *framework* d'Android offre accès à presque toutes les fonctions matérielles et permet de programmer des applications élégantes et conviviales.

Réf 110405-91

3 Pilote de moteur pas à pas Pour ElektorBus (module)

Voyager avec l'ElektorBus ouvre de nouveaux horizons. Grâce à sa modularité logicielle autant que matérielle, ce bus accélère le développement d'applications. Exemple pratique : la mise au pas des moteurs !

Réf 120509-91

4 Elektor Cardioscope

Un électrocardioscope à construire soi-même, utilisant une tablette ou un téléphone Android comme terminal sans fil de visualisation des électrocardiogrammes. La réalisation consiste à combiner habilement une petite interface à PIC pour commander un étage d'entrée analogique et beaucoup de logiciel.

Réf 120107-91

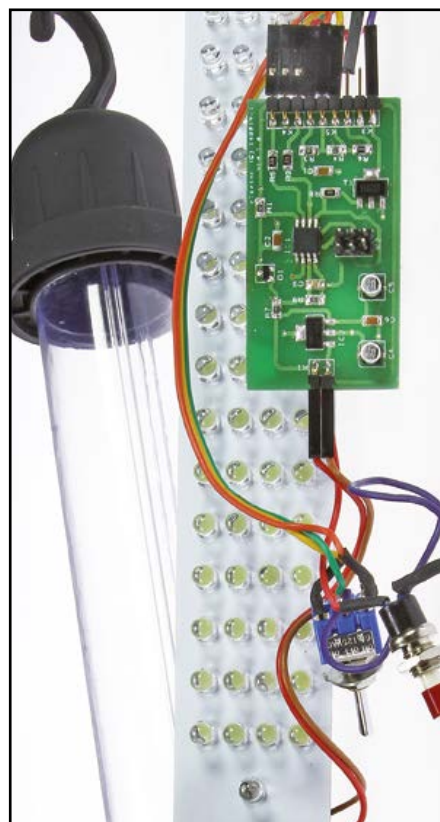
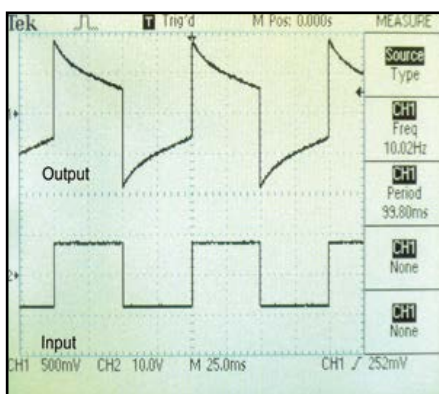
5 Barostick

Clés USB, clés du succès ? Elles sont partout, elles sont notre album d'images, de vidéos, de musiques préférées, d'articles, de fichiers et même de températures. Et la pression atmosphérique, y aviez-vous pensé ? C'est fait : sur un baromètre sans mercure, avec un capteur Bosch, hectopascals et degrés Celsius rejoignent Windows pour se faire tirer le portrait.

Réf 120481-91

Informations et gamme complète sur :

www.elektorpcbservice.com



● communauté

6 de nous à vous

● e-labs

16 Horloges en temps réel

compactes, complètes et précises

La famille d'horloges en temps réel (HTR) du fabricant Micro Crystal réunit des modules polyvalents à faible consommation, tout petits mais très précis. Le quartz horloger caractéristique (32,768 kHz) est intégré.

57 magnétrons

drôle de composant n°9

Le magnétron est la source de micro-ondes utilisée pour les radars (notamment militaires). On s'est aperçu par hasard que ces ondes pouvaient chauffer de la nourriture.

65 Make it in Munich

Si vous allez à Munich pour le salon electronica 2014, faites une pause au stand d'Elektor

76 DesignSpark

14^e jour : l'autorouteur

Pas de travail manuel, laissez faire le routage automatique de DesignSpark.

● industrie

8 Forum LED Europe 2014 contrôleur DC/DC LTC3882 Protoélectronique.com

Prototypage en ligne chez Protoélectronique.com.
Les LED et leurs applications, au ForumLED Europe avec 120 exposants

10 le nouveau débogueur/programmeur Atmel-Ice

Banc d'essai de ce programmeur pour les ARM et les AVR (Xmega y compris).

● projets

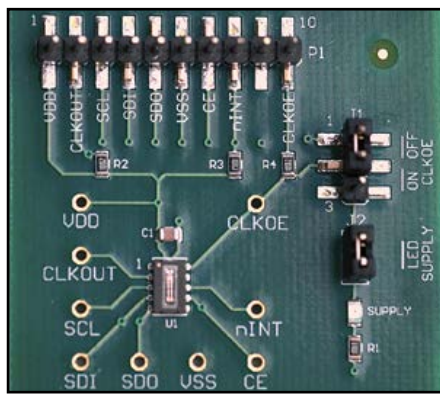
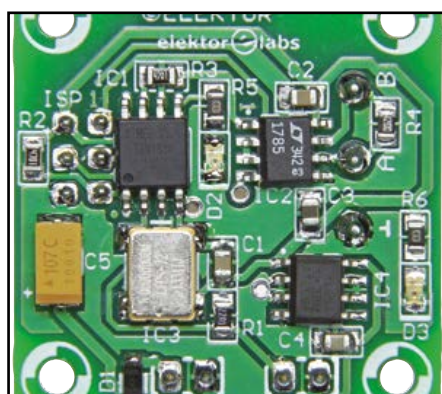
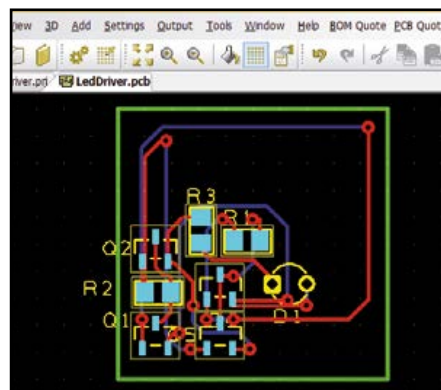
18 antenne ferrite à gyrateur

le condensateur variable remplacé par un potentiomètre

Un gyrateur inverse les propriétés d'une inductance et celles d'un condensateur. C'est un convertisseur d'impédance.

24 transformateurs à circuits gravés principe et simulation

Les pistes cuivrées des circuits imprimés font d'excellents transformateurs et bobines.



30 module capteur de température à interface RS-485

RS-485 permet un transfert fiable des valeurs mesurées sur de longues distances.

38 luxmètre de 1 lux à 100 000 lux

Le rendement des lampes halogènes et des lampes à LED permet de faire des économies en énergie. Mesurons leur luminosité !

44 gradateur PWM pour baladeuse à LED

Ce montage a été conçu pour une lampe à LED utilisée lors de randonnées pédestres.

48 horloge Nixie à récepteur GPS Le mariage (au 3^e top et au 7^e ciel) du tube et du satellite

L'heure exacte est fournie par un module GPS à antenne intégrée.

54 mise à jour du logiciel du régulateur photovoltaïque

nouvelle fonction de mesure de température de la batterie

La modification du régulateur photovoltaïque décrite dans cet article vise à gérer l'utilisation de la batterie aux limites de la plage admissible définie par son fabricant, et même au-delà.

58 transfo de courant (encore) meilleur

Cet article fait suite à deux articles de Martin Oßmann, parus au printemps, sur le calcul et l'utilisation de transformateurs de courant qu'il propose de simplifier et d'améliorer.

60 programmer avec Mono pour Windows, Linux, Mac OS et Android

Fatalité ou nécessité, l'électronicien troque de plus en plus souvent son fer à souder pour le clavier et écrit maintenant des programmes.

66 l'atelier du microcontrôleur (6)

L'interface SPI

Transmettre des informations en série, autrement dit une unité d'information après l'autre, est la façon habituelle de communiquer.

● magazine

78 BK Precision BK560

testeur de CI programmable la grande époque des TTL & CMOS

Comme Hewlett Packard, Apple et nombre de grandes entreprises, B & K a commencé dans un garage, aux États-Unis, en 1948.

81 hexadoku

Gymnastique cérébrale !

82 futurologie

bientôt dans Elektor

37^{ème} année, n° 436
octobre 2014

ISSN 0181-7450
Dépôt légal : septembre 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.

Simulé ou inversé ?

Longtemps, la *simulation* n'a pas eu bonne presse, à peine moins que la *dissimulation*, encore plus mal vue. C'est un sujet que l'on n'abordait que pour réprocher ceux qui s'y adonnent (et se laissent prendre : vous ne devenez simulateur que le jour où votre simulation ne convainc plus). La *simulation* apparaît dans les jargons militaire et technique au milieu du XX^e siècle pour « désigner la représentation du comportement de systèmes physiques par des calculateurs d'abord analogiques, puis numériques » (définition du Robert). Ensuite, le mot *simulation*, dépourvu de connotation péjorative, a été employé en sciences humaines, en linguistique, en économie (simulation de gestion). Pour leur formation, les pilotes passent plus de temps dans des *simulateurs* de vol que dans de vrais avions. En électronique, tout dépendra de votre interlocuteur. S'il a été formé à l'analogique, et s'il a connu les débuts hésitants de la simulation et ses années de tâtonnements, il n'en parlera vraisemblablement que depuis le camp des réticents. Si celui avec qui vous évoquez la simulation est né après le sens moderne du mot, sa mémoire est vierge de l'empreinte olfactive de la bakélite chaude. Ce qu'il voit sur un écran n'est pas pour lui une image de la réalité, mais la réalité elle-même. Le pauvre... Oui, la simulation a progressé, comme en témoigne l'usage convaincant qu'en fait Robert Lacoste dans son article sur la réflectométrie temporelle du mois dernier (Elektor n° 435, page 74). Ce qui m'a conduit à évoquer le sujet ici est une remarque d'un autre Robert, à qui nous devons la version française de l'article sur le **gyrateur**, p.18 de ce numéro-ci. Dans le texte original, l'auteur allemand écrit que le condensateur *simule* l'inductance, et inversement. « Non, rectifie son traducteur Robert Grignard, le gyrateur ne simule pas une inductance à partir d'un condensateur. Il n'y a ni simulation ni faux-semblant, on mesure réellement une inductance ! Le gyrateur *inverse* le condensateur en inductance. C'est un peu comme pour le sucre inversé, quand une enzyme a transformé du saccharose dextrogyre en fructose lévogyre, mais c'est toujours du vrai sucre. On dit aussi en optique qu'un miroir inverse l'image. Comparez les équations des impédances du condensateur et de la bobine : elles se distinguent par un signe moins. Multipliez leurs formules par -1 , vous passez de l'un à l'autre. C'est exactement ce que fait le gyrateur. Multiplier par -1 un signal sinusoïdal revient à le déphaser de 180° , le gyrateur fait donc « tourner » l'impédance. »

Où l'on voit qu'une traduction infidèle mais précise est préférable à un original approximatif.

Bonne lecture !

Denis Meyer

(avec la complicité des Robert, vrais et simulés)



Notre équipe

Rédacteur en chef :

Rédaction internationale :

Laboratoire :

Coordination :

Ont coopéré à ce numéro :

Service de la clientèle :

Graphiste :

Elektor en ligne :

Denis Meyer (redaction@elektor.fr)

Harry Baggen, Jan Buiting, Jaime González-Arintero, Jens Nickel

Thijs Beckers, Ton Giesberts, Wisse Hettinga, Luc Lemmens, Mart Schroijen, Clemens Valens, Jan Visser, Patrick Wielders

Hedwig Hennekens

Rémi Descistes, Robert Grignard, Hervé Moreau,

Kévin Petit, Guy Raedersdorf, Birgit Capenberghs, NN

Cindy Tyssen & Vanessa Noville

Giel Dols

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 29



CES International 2015

www.cesweb.org 2



Eurocircuits

www.elektorpcbservice.com 84



Messe München

www.electronica.de 33



National Instruments

www.ni.com/labview-platform/f 37



Pico

www.picotech.com/PS465 83



Reichelt

www.reichelt.fr 15



Schaeffer AG

www.schaeffer-ag.de 82

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

prototypage en ligne

La société Altrics est spécialisée dans la sous-traitance (cartes électroniques, câblage filaire et intégration). Outre la production de moyennes et grandes séries, elle propose depuis quelques mois un nouveau service pour faciliter la tâche aux bureaux d'étude. Il s'agit d'un portail sur internet qui permet de réaliser des prototypes de cartes électroniques câblées : www.protoelectronique.com.

Le site permet d'obtenir un devis très rapidement. En outre il indique le coût de revient des cartes et donne une vision globale de la disponibilité des composants. Bien entendu Altrics s'occupe de l'approvisionnement en

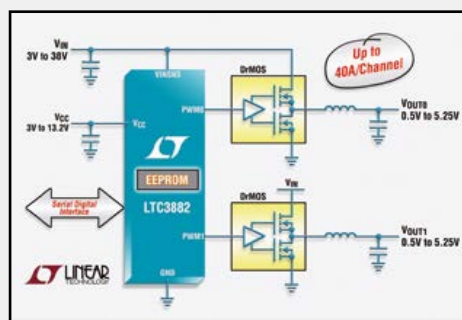
composants. Ensuite, il ne reste plus qu'à commander la carte avec le délai de son choix (standard, rapide ou urgent) et dans la quantité nécessaire (d'un exemplaire à quelques dizaines). Un autre avantage de ce système est que le service est accessible 7 jours sur 7, 24 heures sur 24.

La société Altrics, spécialisée dans l'électronique de puissance, l'énergie et l'éclairage à LED, est répartie sur plusieurs sites :

- plate-forme d'achat industrielle, technique et logistique en France (Alsace)
- moyennes et grandes séries en Tunisie (moyens de production CMS de dernière génération et main d'œuvre qualifiée)
- petites et moyennes séries au Portugal.

Le site tunisien est maintenant équipé d'une machine à laser pour marquer unitairement tous les PCB, ce qui permet d'atteindre une traçabilité totale. Le site français est doté de nouveaux postes pour la réalisation des prototypes de cartes électroniques câblées. Les collaborateurs d'Altrics accompagnent les clients : industrialisation, optimisation des processus de production dès la conception des cartes électroniques, maîtrise de la chaîne logistique et intégration de sous-ensembles électroniques dans leur boîtier final.

www.protoelectronique.com



contrôleur DC/DC à sortie double avec interface numérique et résistance de mesure de courant de 0,25 mΩ

Le LTC3882 est un convertisseur DC/DC, abaisseur, synchrone, multiphase, à sortie double, avec interface numérique.

Il utilise un contrôle de tension par modulation du front montant, pour obtenir une très bonne réponse aux transitoires, et fonctionne avec des inductances de très faible résistance série (0,25 mΩ), ce qui permet des rendements plus élevés et jusqu'à 40 A par phase en sortie. Les composants de puissance compatibles comprennent les DrMOS, les blocs de puissance ou les MOSFET à canal N avec pilotes associés à FET discrets.

L'interface numérique permet aux concepteurs et opérateurs isolés

de commander et superviser le fonctionnement en puissance du système et la consommation. Le LTC3882 et sa commandant numérique simplifient le système, l'optimisation et l'analyse des données pendant la mise au point du prototype, puis lors de sa mise en œuvre. Les applications vont des systèmes de transport à fibres optiques, aux commutateurs et routeurs de transfert de données et de télécommunications, à l'équipement de test industriel, à la robotique, aux systèmes informatiques et RAID, où le coût du matériel électrique, de refroidissement et de maintenance est crucial.

Le LTC3882 offre aussi la possibilité de configurer et de gérer par télémétrie la puissance et les paramètres de gestion de la puissance via PMBus – un protocole, en libre accès, au standard de l'interface numérique série I2C. L'interface série 2 fils du LTC3882 permet la programmation des marges de tension, des réglages et des rampes de montée et de descente avec des retards. Les tensions d'entrée et de

sortie, la puissance de sortie, la température et les valeurs de crête sont lisibles. Le composant comprend deux boucles analogiques de contrôle, rapides, un circuit de précision pour signaux mixés et une EEPROM.

Le logiciel gratuit LTpowerPlay™ permet d'évaluer le LTC3882, de même que le convertisseur USB-PMBus et les kits de démonstration. La précision de la tension de sortie DC sur la gamme de températures est de $\pm 0,50\%$, la relecture du courant précise à $\pm 1\%$, le format du CAN delta sigma est de 16 bits. Il fonctionne à partir d'une alimentation de 3 V à 13,2 V, et est conçu pour une conversion de la tension d'entrée de 3 V à 38 V, en tensions de 0,5 V à 5,25 V. Le partage du courant entre les deux canaux est précis, à la fois à l'état stable ou en fonctionnement transitoire. Jusqu'à quatre LTC3882 peuvent être mis en parallèle pour un fonctionnement à 2, 3, 4, 6 ou 8 phases. Au démarrage, les tensions de sortie, la fréquence de commutation et la désignation de l'angle de la phase du canal peuvent être fixées par des résistances externes ou chargées à partir de l'EEPROM. 40°C à 125°C.

Caractéristiques : LTC3882

- Double contrôleur abaisseur, synchrone, mode tension
- Boucles analogiques rapides de contrôle de la tension
- Interface numérique pour la gestion déportée d'alimentation

- Gamme de tensions d'entrée : 3 V à 38 V
- Gamme de la tension de sortie : 0,5 V à 5,25 V
- Relecture du courant : précision de $\pm 1\%$
- Erreur maximum de $\pm 0,5\%$ de la tension continue en sortie, sur la gamme de températures
- Boîtier QFN-40, 6mm x 6mm, avec acquisition de données et l'EEPROM intégrées

Lecture des données :

- Tensions d'entrée et de sortie, intensités de sortie, puissance de sortie et température
- Pannes et alertes
- Rapport d'enregistrement d'erreur d'identification

Ecriture des données :

- Tension de sortie, séquences de tension et marges de tension
- Données numériques des rampes de démarrage et d'arrêt
- Fréquence de commutation et phases
- Configuration du contrôle PWM
- Niveaux de sous-tension et de surtension en entrée / sortie
- Limite d'intensité en sortie
- Dépassement de température, alertes et seuils d'erreurs

ForumLED Europe 2014

Si vous voulez tout savoir sur les LED et leurs applications, vous avez rendez-vous au ForumLED Europe pour visiter l'exposition, voire pour assister aux conférences. L'objectif de cette manifestation est de réunir des experts et des professionnels du monde entier ainsi que de comprendre les futures tendances des marchés des LED, de découvrir les nouveaux secteurs d'applications des LED et de bénéficier des dernières informations en matière de technologie et de marketing.

Les 120 exposants seront répartis sur deux espaces complémentaires :

- *ForumLED Technologies*, l'espace dédié aux technologies et services de l'industrie de la LED
- *ForumLED Lighting Products*, l'espace dédié aux systèmes et matériels d'éclairage à LED.

C'est l'occasion de découvrir des applications novatrices, de nouvelles solutions techniques et de lier des contacts avec de futurs partenaires.

Les deux journées de conférences seront animées par les meilleurs spécialistes mondiaux. Cette année, huit thématiques seront abordées au cours de sessions animées par des experts issus de laboratoires, d'entreprises de pointe et d'instituts publics spécialisés :

- Marchés des LED
- Normalisation & métrologie

- Alimentations & circuits de commande
- Technologies de fabrication émergentes
- LED blanches : nouvelles recherches, nouveaux résultats
- Les luminophores
- Smart & the city
- Éclairage & bâtiment intelligent

Pour sa sixième édition, l'association exposition & conférences du ForumLED Europe aura encore lieu à Paris, Grande Halle de la Villette, les 14 et 15 octobre 2014.

www.forumled.com



un pour tous !

le nouveau débogueur/programmeur Atmel-Ice



Wolfram Pioch
(Allemagne)

Lors du dernier salon *Embedded World*, Atmel a présenté un nouveau programmeur/débogueur bon marché. Nous avons testé pour vous ce petit boîtier capable de prendre en charge aussi bien les familles ARM qu'AVR, dont les microcontrôleurs Xmega.

Figure 1.
Le kit *Basic* est livré avec ce seul câble équipé de deux connecteurs à 10 et 6 contacts (le premier au pas de 1,27 mm).

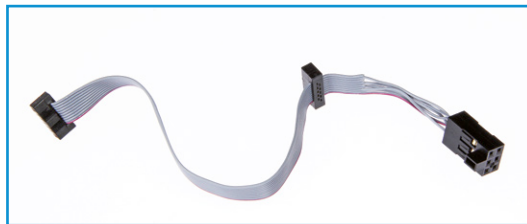
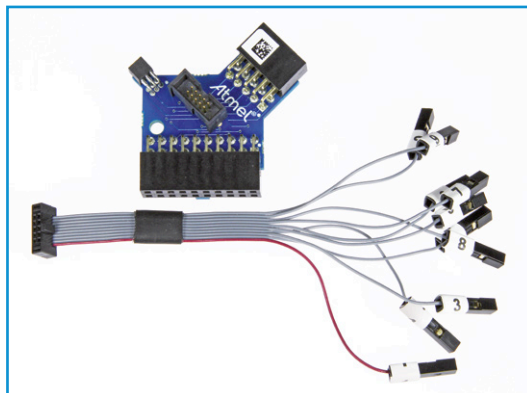


Figure 2.
Le kit *Full* comprend en plus un câble « squid » à prises individuelles, ainsi qu'un adaptateur en forme d'étoile.



Atmel-ICE (ICE pour *In Circuit Emulator*) est le dernier des programmeurs/débogueurs conçus par Atmel. Sa parenté avec le débogueur JTAGICE3 est indéniable : mêmes dimensions (85 x 55 x 16 mm), même connecteur à 10 broches au pas de 1,27 mm. Un câble adaptateur permet de relier le boîtier aux connecteurs de programmation habituels dont sont pourvues les différentes cartes AVR de prototypage et de développement.

L'Atmel-ICE peut cependant déboguer plus de contrôleurs que les seuls ATmega, Xmega et ATTiny. Le fabricant Atmel avait du reste déjà montré qu'il était possible de prendre en charge la famille ARM avec l'environnement de développement AVR Studio (ce qui explique que cet EDI s'appelle désormais Atmel Studio).

Atmel-ICE se veut lui aussi une solution universelle du type « un pour tous ». Il comporte en effet un second connecteur à 10 broches (identifié par SAM) pour la programmation et le débogage

des contrôleurs ARM, p. ex. pour le SAM3X8E (ARM Cortex M3) de la carte Arduino Due.

Lorsqu'on compare ses caractéristiques avec celles des autres débogueurs Atmel (cf. le tableau), on voit que l'adoption du nouvel Atmel-ICE implique certains compromis en termes de vitesse. Ainsi la cadence maximale pour la programmation des contrôleurs ARM est-elle nettement inférieure à celle autorisée par le débogueur SAM-ICE.

Adaptateur

Atmel propose trois versions de l'Atmel-ICE : sous forme de carte sans boîtier, sous forme de boîtier accompagné d'un seul câble, et sous forme de kit adaptateur (kit *Adapter*). C'est ce dernier kit qu'Inteltek, le distributeur allemand d'Atmel [1], nous a gracieusement fourni. Il est affiché à 120 € TTC chez le revendeur en ligne Reichelt. Le kit *Basic* est livré avec un unique câble en nappe (**fig. 1**) doté de deux connecteurs à 10 contacts au pas de 1,27 mm. On peut p. ex. connecter une de ces embases au connecteur de débogage de la carte *Arduino Due*. Le câble possède également une embase à 6 contacts au pas habituel de 2,54 mm, ce qui permet de la brancher sur les connecteurs à 6 broches de nombreuses cartes ATmega (SPI) et Xmega (PDI). Le kit *Adapter* contient un câble supplémentaire équipé de 10 connecteurs (câble *squid*) ainsi qu'une carte adaptatrice en forme d'étoile (**fig. 2**). L'embase à 20 contacts au pas de 2,54 mm de cette carte permet de réaliser une interface SAM-JTAG ou SWD (comme avec le SAM-ICE).

La carte offre en outre la classique embase JTAG au pas de 2,54 mm pour AVR, ainsi qu'un connecteur à 6 contacts pour les signaux ISP, PDI, *debugWire* et *aWire*, toutefois au pas de 1,27 mm. Sa présence réjouira tous ceux qui conçoivent des cartes à microcontrôleurs compacts, puisqu'il permet d'utiliser sur un encombrement réduit une embase à 6 points et au pas de 1,27 mm comme connecteur de programmation/débogage. Le lien [2] fait un tour d'horizon de toutes les possibilités offertes par ce nouvel outil.

Comme d'habitude, PC et débogueur/programmeur se relie avec un câble USB. Lors de nos tests, la prise micro-USB s'est malheureusement abîmée assez rapidement, ce qui nous a obligés à effectuer une réparation maison (suffisamment intéressante pour être décrite dans un autre article). Ce problème mécanique est hélas

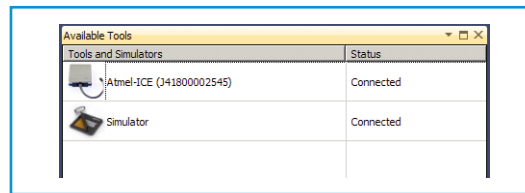


Figure 3.
Sélection des outils.

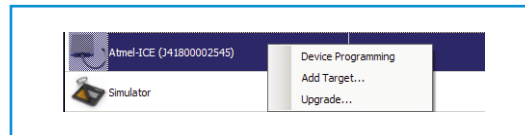


Figure 4.
Vérification de la version du micrologiciel avec *Upgrade*.

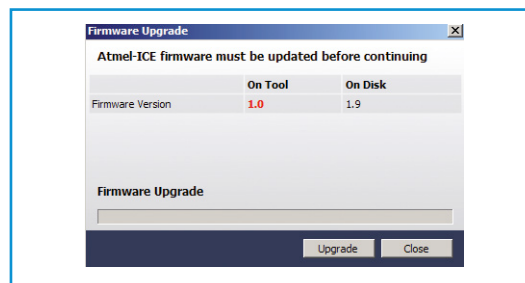


Figure 5.
Version obsolète du micrologiciel.

récurrent avec les prises micro-USB simplement soudées. Atmel fabriquera peut-être un jour une version équipée d'une prise USB-B « normale ».

Programme

Le nouvel Atmel-ICE n'est pris en charge qu'à partir de la version 6.2 d'Atmel Studio, et cette plateforme de développement doit être installée sur le PC avant la première utilisation d'Atmel-ICE [3]. Le programmeur/débogueur est reconnu automatiquement lorsqu'on enfiche la prise USB dans l'ordinateur, et le pilote USB correspondant est installé. La première chose à faire est de vérifier qu'Atmel-ICE a bien été détecté par Atmel Studio.

Le menu *View* → *Available Atmel Tools* ouvre la fenêtre de la **figure 3**. Travailler avec Atmel-ICE suppose d'utiliser la dernière version du micrologiciel : un clic droit ouvre un menu contextuel dans lequel nous pouvons sélectionner *Upgrade...* (**fig. 4**) ; si le micrologiciel du programmeur n'est pas à jour, l'avertissement de la **figure 5** apparaît.

Avant d'entreprendre le débogage d'un projet, nous devons nous assurer qu'Atmel-ICE dialogue effectivement avec le processeur. Nous avons testé la fonction de programmeur de l'Atmel-ICE avec trois processeurs différents.

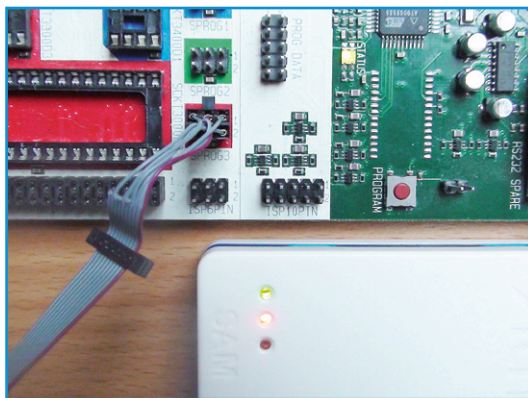


Figure 6.
Programmation de
l'ATmega32 de la carte
STK500.

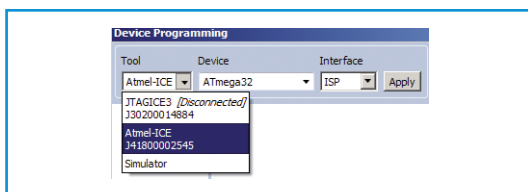


Figure 7.
D'abord choisir l'ATmel-ICE...

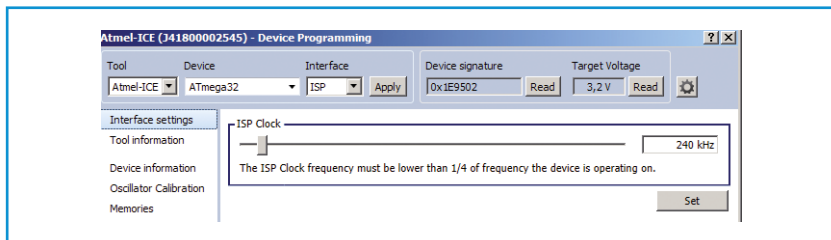


Figure 8.
... puis le processeur et
l'interface.

Le premier était l'ATmega32 de la carte de développement STK500 (fig. 6). Nous avons utilisé le câble du kit *Basic* pour relier l'ATmel-ICE à l'embase du STK500 correspondant au processeur utilisé. Le câble en nappe doit être connecté de telle sorte que le fil rouge marqué « 1 » du câble à 10 contacts soit du même côté que la désignation « 1 » de l'embase STK. L'ergot du connec-

teur à 6 contacts doit être orienté vers le haut. Le programmeur se lance depuis le menu *Available Atmel Tools* → *Atmel ICE* → *Device Programming*. On peut aussi le lancer en cliquant sur l'icône de la barre d'outils représentant une puce zébrée par un éclair.

Le menu déroulant *Tools* de la fenêtre *Device Programming* permet de sélectionner l'Atmel-ICE (fig. 7). Le menu déroulant *Device* attend le processeur cible (ATmega32), et l'interface utilisée se paramètre depuis *Interface* (ISP ici). Cliquer sur le bouton *Apply* valide ces choix (fig. 8). Pour finir, nous réglons sur environ 240 kHz le paramètre *ISP Clock* du menu *Interface Settings* ; cette fréquence ne doit pas excéder le quart de la fréquence d'horloge du processeur. N'oublions pas ici qu'un processeur neuf est cadencé par défaut à 1 MHz avec l'oscillateur RC interne. Une fréquence d'horloge de 240 kHz convient cependant toujours. Après avoir configuré une fréquence d'horloge plus élevée à l'aide des fusibles, il reste bien sûr possible d'augmenter à l'unisson la fréquence ISP.

Cliquer sur le bouton *Read* placé sous *Device signature* (fig. 9) établit une première communication avec le processeur. Si aucun message d'erreur n'apparaît, c'est que le bon processeur a été sélectionné.

Xmega

Nous avons ensuite testé le Xmega192A3 d'une carte STK600. L'essai nécessite cette fois-ci l'utilisation de l'interface PDI. Le connecteur s'enfiche sur l'embase SPI/PDI avec l'ergot orienté côté processeur (fig. 10). La configuration du programmeur est identique à celle de l'exemple

Caractéristiques des principaux débogueurs Atmel

	AVR ONE!	JTAGICE3	Atmel-ICE	SAM-ICE
interface	SPI,PDI,TPI debugWire JTAG	SPI,PDI,TPI debugWire aWire JTAG	SPI,PDI,TPI debugWire aWire JTAG/SWD	JTAG/SWD
horloge SPI	32 kHz à 32 MHz	8 kHz à 1,8 MHz	8 kHz à 5 MHz	
horloge JTAG	32 kHz à 32 MHz	32 kHz à 15 MHz	32 kHz à 7,5 MHz	
horloge PDI	32 kHz à 32 MHz	32 kHz à 10 MHz	32 kHz à 7,5 MHz	
débit dW	non mentionné	4 kbit/s à 0,5 Mbit/s	4 kbit/s à 0,5 Mbit/s	
débit aWire	non mentionné	7,5 kbit/s à 7,5 Mbit/s	7,5 kbit/s à 7 Mbit/s	
horloge SWD			32 kHz à 2 MHz	0 à 8 MHz

SPI, sinon que nous devons sélectionner PDI pour l'interface. Et ici nous pouvons pousser le curseur *PDI Clock* jusqu'à la fréquence maximale de 7,5 MHz.

Il est également possible de programmer la carte Xmega par JTAG en reliant l'adaptateur de l'Atmel-ICE à la prise JTAG du STK600 avec le côté composants orienté vers le processeur (**fig. 11**). Nous sélectionnons JTAG pour l'interface, et là aussi nous pouvons paramétrer la fréquence d'horloge sur 7,5 MHz.

Nous avons procédé au dernier test avec la carte *Arduino Due* (ATSAM3X8E). Le câble du kit *Basic* se connecte à l'interface de débogage à 10 contacts (**fig. 12**) de façon à ce que le fil identifié par « 1 » pointe vers le centre de la carte. Le connecteur de l'Atmel-ICE à utiliser est bien sûr le SAM. Utiliser par mégarde le connecteur AVR n'entraîne aucun dommage ; la LED verte du débogueur ne s'allume que si le bon connecteur est utilisé. Relier l'Atmel-ICE et une carte Arduino équipée d'un *shield* ne pose par ailleurs aucun problème.

Dans le menu *Interface Settings*, c'est le ATSAM3XE de l'*Arduino Due* qu'il nous faut sélectionner comme processeur. Attention ! Veillez à bien choisir SWD (*Serial Wire Debug*) comme interface, car JTAG ne fonctionne pas. La fréquence maximale n'est (hélas) que de 2 MHz.

Débogage

Ces préalables posés, voyons comment procéder au débogage d'un projet complet, qu'il soit AVR ou ARM. Nous testerons ici la nouvelle interface SAM. Comme premier projet, nous vous recommandons de choisir un exemple Atmel Studio approprié et de le personnaliser en fonction de vos besoins.

Nous allons illustrer le processus à suivre avec l'exemple d'une carte *Arduino Due*. Nous n'utiliserons pas un croquis (*sketch*) Arduino, mais un « vrai » programme C. Nous montrerons cependant dans un prochain article comment écrire et déboguer des croquis Arduino avec Atmel Studio.

La première étape consiste à lancer *File* → *New* → *Example project*. Dans la fenêtre qui s'ouvre, choisissez *SAM*, *32 bits* dans le menu déroulant *Device Family*, et *Applications* dans le menu *Category*. Cliquez ensuite sur *Getting Started*

Application on SAM - Arduino Due/X (**fig. 13**). Le nom du projet ainsi que l'emplacement mémoire peuvent être modifiés dans le bas de la fenêtre. Laissez tel quel le nom par défaut, et saisissez le chemin d'accès *D:\ATMEL ICE* dans la zone de texte *Location*. Acceptez les termes de la licence

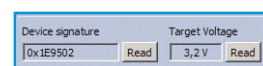
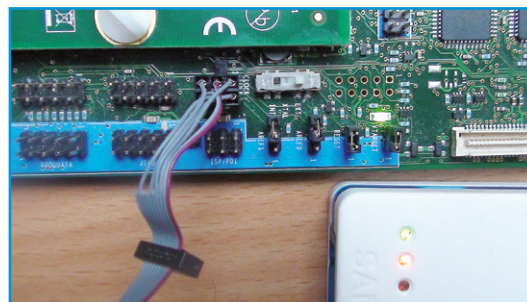


Figure 9.
Lire la signature du dispositif est de bon augure pour la suite.

Figure 10.
Programmation du Xmega via PDI.

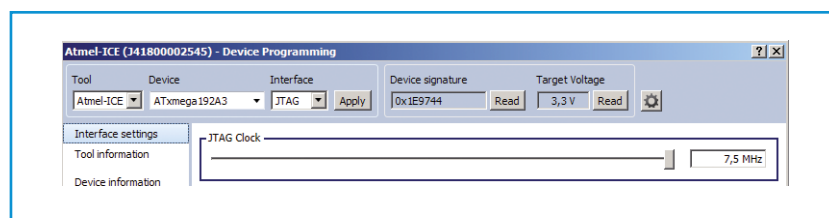


Figure 11.
Xmega peut aussi être programmé via JTAG.

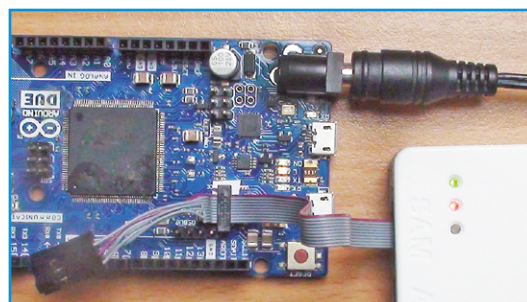


Figure 12.
Arduino Due reliée à Atmel-ICE.

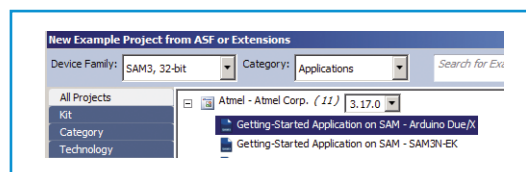


Figure 13.
Sélection d'une application d'exemple

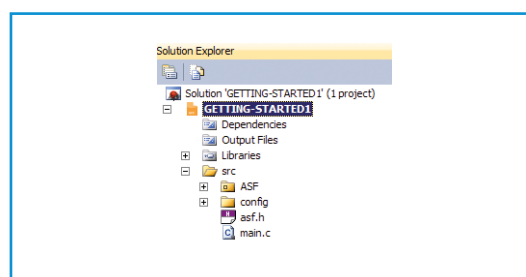


Figure 14.
Fichiers de l'application dans l'explorateur.

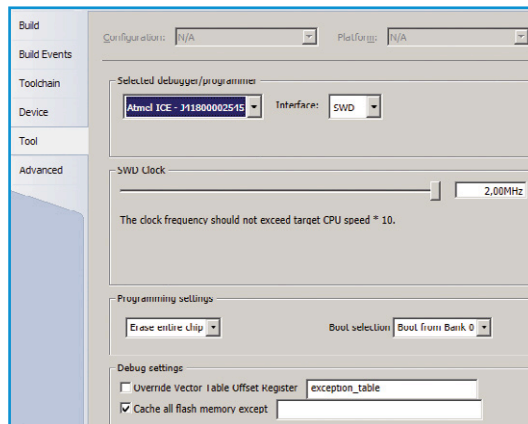


Figure 15.
Le débogueur et l'interface se paramètrent depuis l'onglet *Tool* du menu des propriétés de l'application.

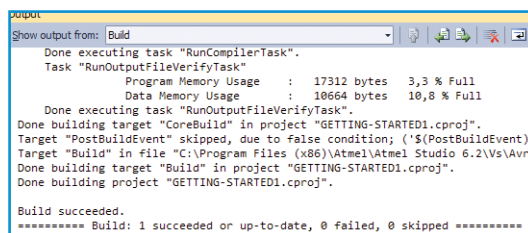


Figure 16.
La fenêtre de sortie.

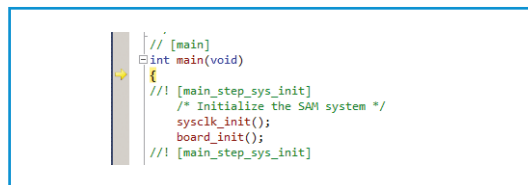


Figure 17.
Alt-F5 arrête le programme sur la première ligne.

après avoir validé vos choix avec *OK*. Atmel Studio confirme alors la création du projet. L'arborescence du projet s'affiche à droite de la fenêtre, dans le *Solution Explorer* (fig. 14). Cliquer sur l'entrée *main.c* ouvre le fichier correspondant dans l'éditeur.

Un clic droit sur le nom du projet *GETTING-STARTED1*, ou un clic sur *Project* → *GETTING-STARTED1 Properties* dans le menu principal (Alt-F7), ouvre la page des propriétés du projet.

Le processeur utilisé peut être modifié dans l'onglet *Device*. Le débogueur (Atmel ICE) et l'interface (SWD) se paramètrent dans l'onglet *Tool*. L'option *Cache all flash memory except* doit être décochée uniquement si la mémoire flash est modifiée au lancement du programme (p. ex. via un chargeur d'amorçage) et si l'on souhaite observer ces modifications. Autrement la fenêtre *Memory* montre seulement l'état de la mémoire flash au moment où le programme démarre.

Tous les paramètres restants, y compris ceux des autres fenêtres, ne nécessitent aucune modification. Les paramètres s'enregistrent via *File* → *Save Selected Items* (Ctrl-S), ou en passant par *File* → *Save All* (Ctrl-Shift-S).

Le programme se compile via *Build* → *Solution* (ou avec F7). Comme nous n'avons encore rien modifié, il ne devrait y avoir aucune erreur, ce que confirme la fenêtre de sortie (fig. 15). Nous y sommes enfin : la combinaison Alt-F5 charge dans le processeur le fichier ELF, visible dans l'explorateur sous *output*, et le programme s'arrête sur la première ligne de la fonction *main* (fig. 16). Avec F5 au contraire, le programme s'exécute aussitôt chargé dans le processeur : une LED clignote jusqu'à l'arrêt du programme.

Toutes les fonctions de débogage, p. ex. sauter au point d'arrêt suivant, se lancent depuis le menu *Debug* du menu principal.

Pour conclure, une astuce à retenir : si rien ne fonctionne, un remède souvent efficace, surtout lorsqu'un autre programme a été précédemment chargé dans le processeur, consiste à effectuer un *Erase Chip* complet (onglet *Memories* dans la fenêtre du programmeur).

Bilan

Si Atmel-ICE combine les fonctions d'un débogueur SAM et d'un débogueur AVR pour un prix modéré, il faut toutefois accepter un certain compromis sur la vitesse si on la compare à celle qu'offrent les autres débogueurs Atmel. Avoir une vitesse faible n'a que peu d'importance pour les projets de petite à moyenne taille, mais dans le cas de fichiers volumineux et de développement de programme soutenu, le temps de programmation plus court d'un AVR ou d'un SAM-ICE reste appréciable.

(140275 – version française : Hervé Moreau)

Liens

- [1] www.ineltek.de
- [2] www.atmel.com/webdoc/atmelice/index.html
- [3] www.atmel.com/tools/atmelstudio.aspx

Votre partenaire en ligne compétent pour

Eléments de construction
Alimentation électrique
Technique de mesure

Technique atelier & de brasage
Technique domotique & sécurité
Technique réseau

Technique PC
Technique Sat/TV
Communication

Les langues de notre
boutique:



FLUKE®

S'ABONNER MAINTENANT !

Newsletter en anglais

Recevez chaque semaine les toutes
dernières informations sur



- ✓ Nouveautés
- ✓ Les meilleures offres
- ✓ Réductions des prix



http://rch.it/Flu

Technique de mesure pro

pour l'industrie, l'artisanat et les professionnels du métier

Thermomètre IR visuel

Mesure ultra précise de la température combinée
avec les avantages d'une caméra à infrarouge



- ✓ Amplitude de mesure :
-10 °C à 250 °C
- ✓ Précision : ± 2 °C ou ± 2 %
- ✓ Champ visuel : 28° x 28°

- Caméra numérique intégrée
- 5 modes de fondu avec carte thermique
- Intelligent & entièrement automatique
- Etablissement de rapports professionnels avec le logiciel SmartView®
- Fonction détection automatique des points de mesure chauds/froids
- Fonctions alarme & accéléré
- Inclus : accessoires complets

FLUKE VT04

749,00

Multimètre numérique TRMS

D'une performance impressionnante, extrêmement convivial, ultra précis, sûr et très fiable – le Fluke 175 est une nouvelle référence !

- Sélection manuelle et automatique de la plage
- Mesure de fréquence, de capacité, de résistance, vérification du passage, test des diodes
- Enregistrement min./max./moyenne
- Mode smoothing pour des valeurs mesurées plus stables
- EN 61010-1, CAT III 1000 V, CAT IV 600 V
- Inclus : batterie, instructions de mesure et notice d'utilisation



Meilleures ventes ! FLUKE®

- ✓ Mesure à valeurs efficaces vraies de la tension et du courant
- ✓ Précision de base : 0,1 %
- ✓ Résolution 6.000 digits

FLUKE 175

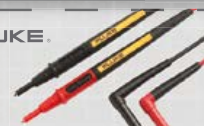
199,00

WearGuard™ indique les détériorations des câbles de mesure



Kit de câbles de mesure TwistGuard™ FLUKE

- Longueur réglable des pointes de mesures : 4–19 mm
- Homologué pour : CAT II 1000 V, CAT III 1000 V, CAT IV 600 V, max. 10 A
- Degré d'encrassement 2



FLUKE TL175 26,85

Typique FLUKE®

✓ robuste ✓ fiable ✓ précis

Pince ampèremétrique ultra efficace

Grâce à leurs mesures à valeurs efficaces vraies et leur ergonomie optimisée, les pinces ampèremétriques de la série 320 sont les appareils de mesure les plus importants pour l'installation et la recherche d'erreurs dans les applications industrielles et la technique des bâtiments.

- Mesure de la tension et du courant AC/C jusqu'à 400 A / 600 V
- Mesure à valeurs efficaces vraies permettant des mesures précises
- Mesure de résistance jusqu'à 40 kohms avec reconnaissance du passage
- Mesure de la température, de la capacité et de la fréquence
- Design mince et ergonomique
- Diamètre de câble max. : 30 mm
- Inclus : câbles de mesure, pochette de transport rembourrée, notice d'utilisation

EN 61010-1
CAT III
600 V

EN 61010-1
CAT IV
300 V

FLUKE 325

235,00



FLUKE® Toute dernière technologie de mesure et de sécurité



Contrôleur de tension et de passage

- Affichage LCD-/LED, éclairé
- Lampe de poche intégrée
- Tension continue/alternative : 6–690 V
- Test de passage signal & vibrations
- Indicateur du champ magnétique rotatif, mesure de fréquence, mesure de résistance
- Fonction pour contrôler les RCD
- Contrôle de phase unipolaire
- Spécifications IP : IP64

EN 61010-1
CAT III
600 V

EN 61010-1
CAT IV
600 V

FLUKE T150 VDE

144,00



Commander maintenant! www.reichelt.fr

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

Prix du jour ! Prix à la date du : 05. 08. 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)

Modes de paiement internationaux:



Micro Crystal : horloges en temps réel compactes, complètes et précises

Viacheslav Gromov
(Allemagne)

Nos circuits ne sont pas encore tous en mesure de se mettre à l'heure tout seuls par l'internet ou par un signal radio. Et même quand ils le sont, il faut prévoir une solution de substitution hors connexion ou hors réception. On dispose pour cela de la fonction de *Real Time Clock*, intégrée dans des circuits comme le DS1302. Ce composant classique a besoin d'un quartz et n'offre pas d'interface normalisée, alors que les RTC modernes sont des modules prêts à l'emploi.

La famille d'horloges en temps réel (HTR) du fabricant *Micro Crystal* réunit des modules polyvalents à faible consommation, tout petits mais très précis. Le quartz horloger caractéristique (32,768 kHz) est intégré. Or, qui dit horloge, dit Suisse. Vous ne

serez donc pas surpris d'apprendre que, une fois n'est pas coutume, *Micro Crystal* est une société helvétique du groupe *Swatch*. Outre les quartz et le modules HTR, elle propose des cartes d'évaluation si bien présentées qu'elles donnent vraiment envie de s'y mettre et d'expérimenter (**fig. 1**).

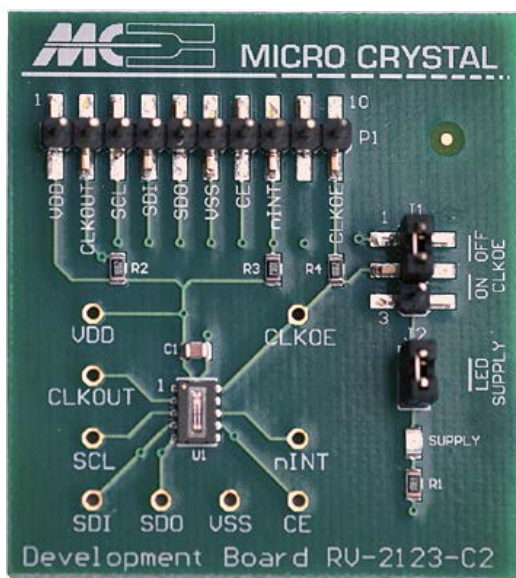


Figure 1.
Carte d'évaluation
(Photo : Richard Endres).

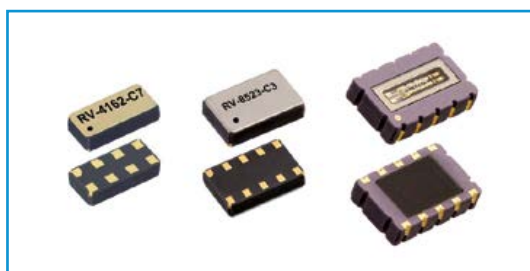


Figure 2.
Trois tailles de boîtier, de
gauche à droite :
C7, C3 et C2.

Anatomie des coucous

Micro Crystal propose des modules avec interface I²C ou SPI à haut débit. Trois modèles de boîtier (**fig 2**) : C2 pour le plus grand : 5,0 x 3,2 x 1,2 mm, avec hublot. Le format moyen (C3) fait 3,7 x 2,5 x 0,9 mm. Quant au format C7 (3,2 x 1,5 x 0,8 mm), personne ne fait de RTC plus petite que ça. Certains modules existent en différents formats de boîtier.

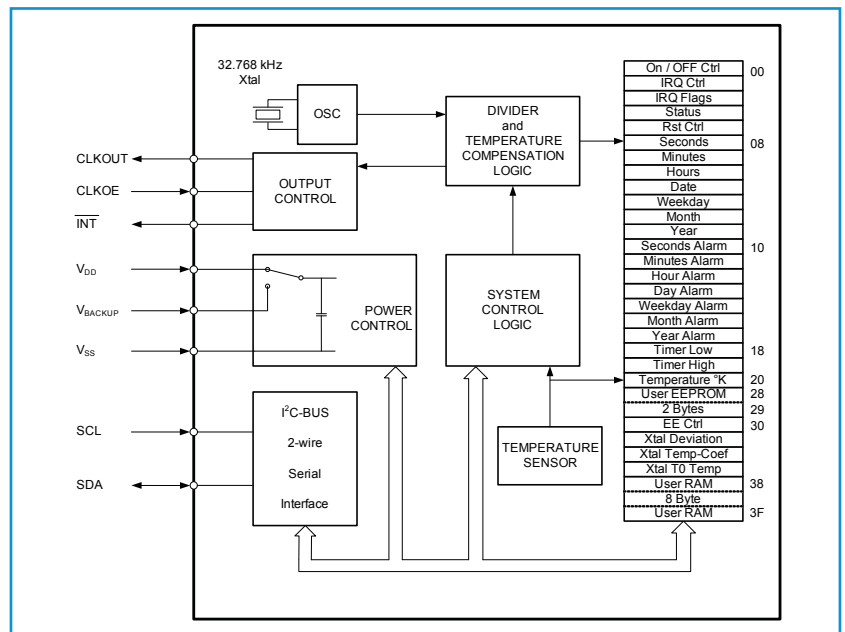
Le principe est le même, mais les caractéristiques varient selon les modèles, notamment le nombre de registres et la configuration des périphériques, mais aussi la consommation et la précision (**tableau 1**). Dans la colonne *tension*, le minimum correspond à la tension en dessous de laquelle le module perd l'heure.

Je retiens le modèle RV-3029, avec son interface I²C, pour étudier l'anatomie des modules HTR. Le capteur thermique (**fig. 3**) que je viens d'évoquer ne sert pas tant pour la température ambiante que pour celle du quartz. Le fabricant sauvegarde dans des registres spécifiques des paramètres de compensation thermique pour le quartz. Ces valeurs peuvent être modifiées par l'utilisateur, s'il sait ce qu'il fait...

L'oscillateur à quartz est suivi par un diviseur correcteur, qui incrémente directement le registre des secondes. Le contenu de tous les registres est au

format BCD. Dans le registre des heures, un bit détermine le mode 12 ou 24 h. Le bloc *Output Control* fournit sur la sortie CLKOUT un signal d'horloge qui dépend de CLKOE. Sa fréquence varie de 1 à 32 768 Hz, selon les bits FD1 et FD0 dans le registre de commande de l'EEPROM. En cas d'interruption, la broche INT est au niveau bas. Les interruptions proviennent du temporisateur (*Timer*), d'un réveil, de la fonction *Power Control* ou du système de récupération automatique. Ce dernier cas résulte d'un éventuel plantage de l'automate interne. Il y a deux registres temporisateurs, l'un pour l'octet de poids fort, l'autre pour l'octet de poids faible, qui forment un décompte à 16 bits, dont l'horloge est fixée par les registres de commande. Ceux-ci permettent aussi d'indiquer si le temporisateur doit être rechargé automatiquement en fin de décompte. La plupart des interruptions doivent être autorisées dans les registres de commande. Des indicateurs signalent si tel ou tel événement a déjà eu lieu ; il faut les remettre à zéro après chaque événement, même si les interruptions sont désactivées. D'autres indicateurs signalent l'événement qui a causé l'interruption ; ceux-ci doivent aussi être initialisés après une interruption, pour que la broche /INT revienne au niveau haut.

Le bloc *Power Control* est un astucieux dispositif de lecture de niveaux de tension et, en fonction du résultat, de commutation d'une résistance interne entre V_{DD} et V_{BACKUP} . Ceci a pour effet de recharger un supercondensateur connecté à V_{BACKUP} . Quand la tension d'alimentation passe sous le seuil de 2,1 V (V_{LOW1}), CLKOUT, le capteur thermique et la fonction d'écriture dans les registres sont inhibés. L'interface I²C reste active, mais à vitesse réduite.



Exploitation

Lors de la mise sous tension et après remise à zéro, la plupart des fonctions (réveil et temporisation) sont inactives. Le fabricant préconise, après chaque initialisation de mise sous tension, une remise à zéro du système (*Reset*) à l'aide du bit SysR du registre de commande. Ensuite, on configure les autorisations d'interruption à l'aide du registre de commande, de même que les périphériques. Le condensateur de 10 nF est indispensable sur la ligne d'alimentation à proximité immédiate du module. Le manuel est riche [1] en informations sur le fonctionnement et l'utilisation.

(130580 – version française Rémi Descistes)

Lien

[1] www.microcrystal.com/index.php/products/real-time-clocks

Figure 3.

La fiche technique du module RV-3029.

Tableau 1. Les modules HTR					
boîtier	précision (25 °C)	interface	tension	courant	caractéristiques
RV4162-C7	±20 ppm	I ² C	1,0 à 4,4 V	350 nA	boîtier miniaturisé
RV8523-C3	±20 ppm	I ² C	1,2 à 5,5 V	130 nA	faible consommation
RV8564-C2/C3	±20 ppm	I ² C	1,2 à 5,5 V	250 nA	standard
RV3029-C2/C3	±3 ppm	I ² C	1,3 à 5,5 V	800 nA	précision
RV1805-C3	±20 ppm	I ² C	1,5 à 3,6 V	50 nA	très faible consommation
RV8803-C7	±10 ppm	I ² C	1,5 à 5,5 V	250 nA	boîtier miniaturisé
RV2123-C2/C3	±20 ppm	SPI	1,1 à 5,5 V	130 nA	faible consommation
RV049-C2/C3	±3 ppm	SPI	1,3 à 5,5 V	800 nA	précision

antenne ferrite à gyrateur

le condensateur variable remplacé par un potentiomètre

Martin Oßmann
(Allemagne)

Un gyrateur permet d'invertir, de renverser symétriquement les propriétés d'une inductance et celles d'un condensateur. C'est un convertisseur d'impédance, nous allons voir comment ça marche, puis nous réaliserons avec lui une antenne ferrite à syntonisation sans condensateur variable.

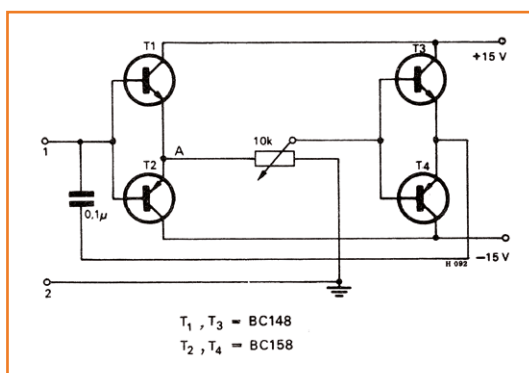


Figure 1.
Une capacité variable jusqu'à 100 µF, publié dans l'édition allemande d'Elektor en novembre 1970.

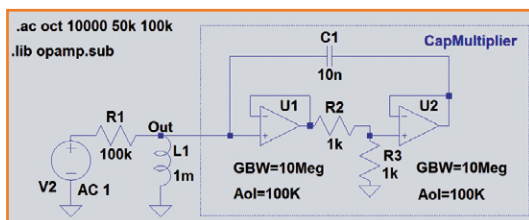
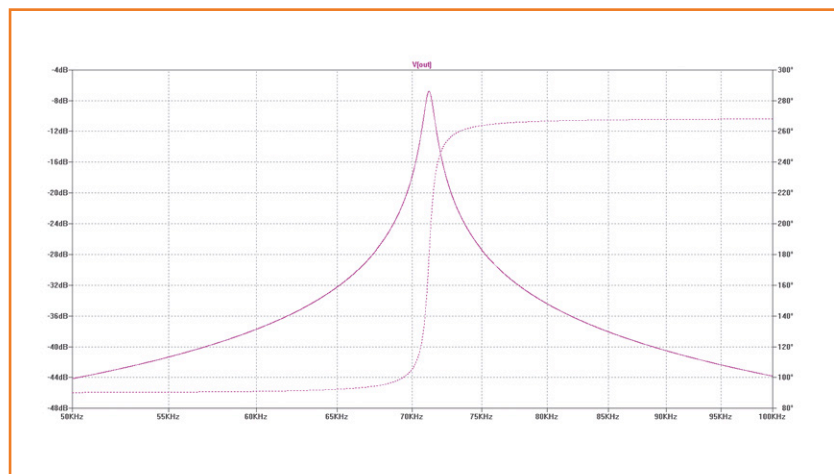


Figure 2.
Simulation d'un circuit oscillant à condensateur variable.



J'étais encore adolescent quand j'ai trouvé dans un magazine Elektor de 1970 le circuit pour faire le condensateur variable de la **figure 1**.

Très vite, il m'est venu l'idée de réaliser un système d'accord du circuit oscillant d'un récepteur à ondes moyennes par un tel montage avec un potentiomètre pour remplacer le condensateur variable. J'ai bien essayé, à l'époque, mais ça n'a pas marché. Pourtant, j'ai conservé l'espoir d'y arriver. Il existe bien des diodes à capacité variable, mais aucune ne peut atteindre plusieurs nanofarads.

Avec les faibles moyens d'un écolier bricoleur, je n'arrivais pas à comprendre pourquoi le remplacement du condensateur variable avait tourné court. Aujourd'hui, muni d'appareils de mesure et d'une solide compréhension des circuits analogiques, je sais pourquoi ça a foiré, mais en outre, avec l'appoint des amplificateurs opérationnels, je peux réaliser mon rêve de syntonisation par potentiomètre et vous faire profiter de mes réflexions et de mes expériences.

Substitut de condensateur

Pour comprendre le circuit, je l'ai d'abord simulé sur LTspice, la **figure 2** en donne le schéma. En ajoutant une résistance et une bobine à un filtre RLC actif, on imite un circuit oscillant.

On voit à la **figure 3** la courbe de fréquence quand on simule son comportement en petits signaux avec un balayage alternatif.

Jusqu'ici, tout répond aux attentes. On voit une belle courbe de résonance. Le circuit simule bien

Figure 3.
Réponse en fréquence du filtre RLC de la figure 2.

un condensateur de $0,5 \times 10 \text{ nF}$, selon le diviseur de tension R_2/R_3 . Avec $L = 1 \text{ mH}$, le calcul donne une fréquence de résonance de 71 kHz environ, que la simulation reproduit bien.

Construisez le circuit, vous serez amèrement déçu. Au lieu de fonctionner comme un filtre sélectif, il entre allègrement en oscillation. Dans une simulation des transitoires, Spice aussi la restitue.

À la **figure 4**, on voit la croissance exponentielle comme résultat de la simulation. On peut s'étonner puisque les deux amplis ensemble ne réalisent qu'un gain $A_v = 0,5$. Mais attention ! Le condensateur C_1 et la self L_1 forment dans la rétroaction un circuit oscillant série, la tension de sortie de l'ampli U_2 s'accroît du facteur de surtension Q , ce qui remplit la condition d'oscillation. Dans la simulation en petits signaux, Spice ne reconnaît pas ce genre de structures. Cette observation doit nous faire comprendre qu'une telle simulation ne peut parfois nous livrer qu'une vérité tronquée quand on lui soumet une construction susceptible d'osciller.

Gyrateur à amplificateurs opérationnels

Pour faire office de condensateurs et de bobines, on peut aussi se servir de gyrateurs qui inversent leurs fonctions. On va donc les utiliser comme convertisseurs d'impédance. D'une manière plus générale, on les emploie dans des circuits dits quadripôles actifs. Le gyrateur relève d'une longue tradition chez Elektor, on en trouve déjà dans les tout premiers numéros d'Elektor [2] des éditions allemande et hollandaise. La **figure 5** présente le circuit traditionnel d'un gyrateur à amplis op. Son impédance Z invertie

$$z = \frac{z_1 z_3 z_5}{z_2 z_4}$$

Si par exemple vous prenez un condensateur pour Z_2 et des résistances pour les autres impédances, vous allez obtenir une inductance. En effet :

$$z = j\omega \frac{R_1 R_3 R_5 C_g}{R_4} = j\omega L_g$$

On en tire :

$$L_g = K C_g \quad \text{où} \quad K = \frac{R_1 R_3 R_5}{R_4}$$

La configuration de la **figure 6** est aussi un gyrateur pour lequel :

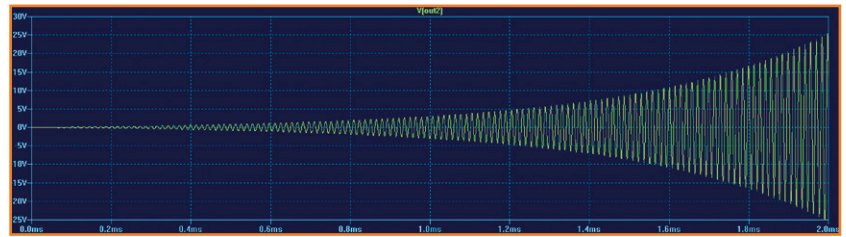


Figure 4.
La simulation en transitoires indique l'instabilité.

$$z = \frac{z_1 z_3 z_5}{z_2 z_4}$$

Avec un autre condensateur et une résistance, on peut transformer le circuit de la figure 5 en filtre passe-bande RLC, comme le montre la **figure 7**. À gauche, par le symbole du gyrateur, à droite, le schéma équivalent. La constante K

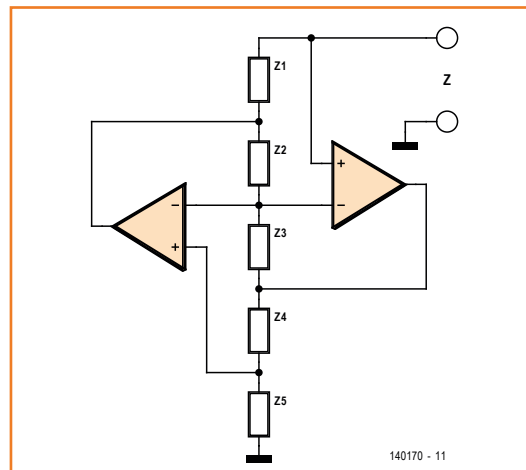


Figure 5.
Gyrateur à amplificateurs opérationnels.

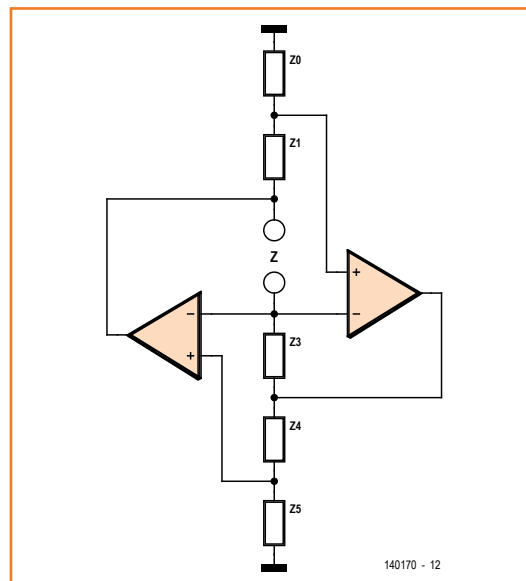


Figure 6.
Variante du circuit de gyrateur.

Figure 7.
Structure d'un filtre RLC
avec gyrateur.

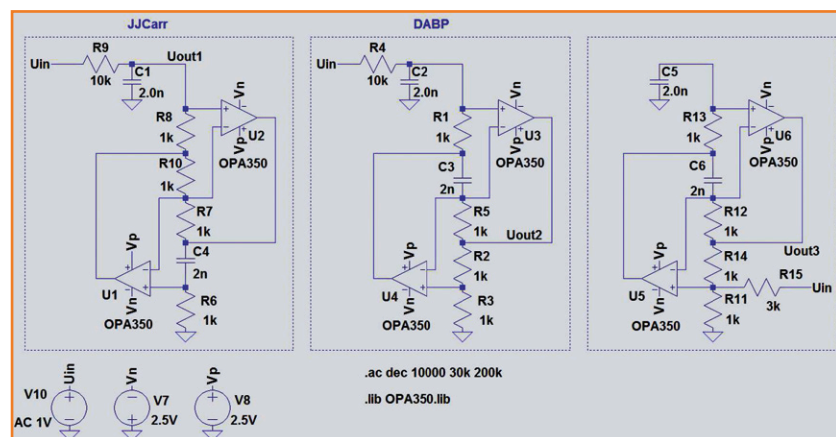
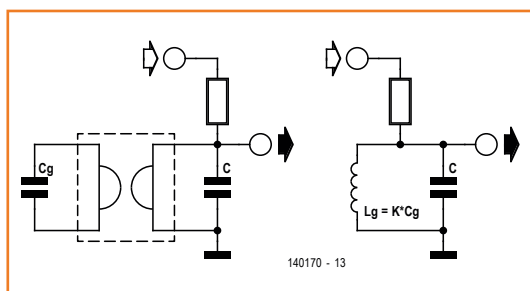
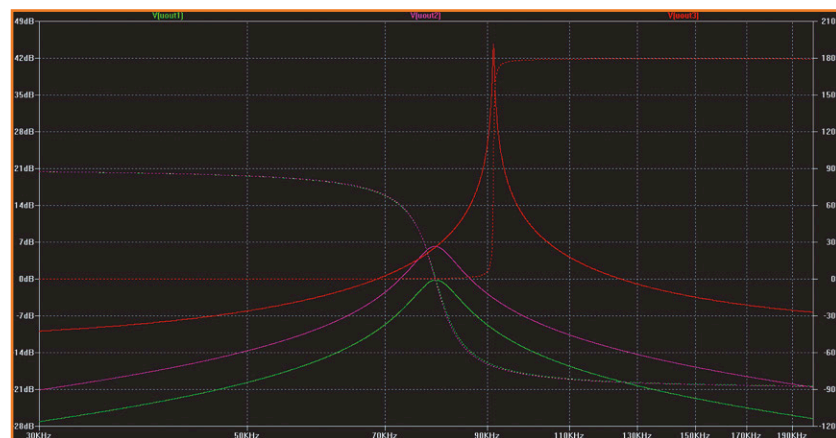


Figure 8.
Trois modèles de filtres RLC
à gyrateur.

Figure 9.
Réponses en fréquence
des circuits de la figure 8 :
vert = Uout1, mauve =
Uout2, rouge = Uout3.



À la **figure 8**, une simulation sur LTspice vous en montre trois variantes. Celle de gauche correspond au filtre de [3]. Le gyrateur équivaut à une inductance mise dans un circuit oscillant LC auquel on applique une tension U_{in} par la résistance R9. La tension de sortie U_{out} est prise à haute impédance sur le circuit oscillant parallèle, normalement par l'intermédiaire d'un convertisseur d'impédance. Le gain maximum à la résonance est de 1 (0 dB) et le facteur de mérite Q est ici déterminé principalement par la résistance R9.

Au milieu, on a affaire à un filtre passe-bande à deux amplificateurs (*DABP*) décrit dans [4]. Sous cette présentation, il n'est pas facile d'y reconnaître un filtre à gyrateur. La différence avec l'exemple de gauche réside dans le couplage par la sortie de l'ampli op, à basse impédance, cette fois, ce qui permet une certaine amplification.

Dans le circuit de droite, la tension d'entrée U_{in} n'est pas appliquée par une résistance au circuit oscillant, mais directement au gyrateur. Cette idée a été utilisée dans [5], par exemple, pour le couplage de filtres à gyrateur, grâce à quoi le facteur de mérite n'est plus abaissé par la résistance de couplage. On atteint ainsi un bien meilleur facteur Q et un plus grand gain. On peut alors obtenir, à une fréquence moyenne de 90 kHz, une bande passante (à -3 dB) de 300 Hz, ce qui correspond quand même à un Q de 300.

La **figure 9** illustre, par leurs courbes de fréquence, les performances des trois filtres. Le fait que le circuit de droite a une fréquence de résonance plus haute tient à R15, le comportement en petits signaux du gyrateur en parallèle tient à R11. Changer R11 peut tout égaliser.

Essai pratique

Prenons le circuit de droite de la figure 8 pour un essai suivant le schéma de la **figure 10**. Nous mettrons en œuvre un double ampli op AD8042 caractérisé par un produit gain \times bande passante $GBW = 160$ MHz.

Pour mes expériences sur les gyrateurs, j'ai préparé la plaquette perforée de la **figure 11**. Elle me permet d'échanger facilement les impédances pour jouer avec les diverses configurations possibles.

On voit à la **figure 12** la courbe de fréquence mesurée sur le filtre. Pour une fréquence moyenne de 455 kHz, le circuit présente, à -3 dB, une lar-

geur de bande de 3 kHz. Le facteur de surtension avoisine $Q = 150$, on ne doit donc pas amplifier derrière le filtre. Si le circuit avait tendance à osciller à cause de capacités parasites, il suffirait d'y ajouter une légère atténuation par une résistance en parallèle sur C1. C'est bien une indication qu'avec les amplificateurs opérationnels modernes, on peut construire des filtres à gyrateur à grande surtension dans la gamme des ondes moyennes. Reste encore à atteindre mon but initial : une antenne ferrite à syntoniser par une résistance !

Antenne ferrite accordée par potentiomètre

Dans ma quête du remplaçant du condensateur variable, il m'est venu une idée : si l'on peut réaliser avec un gyrateur des inductances et des résistances variables, il devrait être possible de fabriquer une antenne ferrite accordable. La **figure 13** schématise ma première intuition. On y trouve une inductance fixe L_g associée au condensateur du circuit oscillant C_g pour former, avec l'antenne ferrite L_a , un circuit oscillant. Des soucis de stabilité ont empêché cette idée de se transposer dans la réalité.

Une autre solution existe : un gyrateur avec encore une inductance variable L_g , laquelle se branche en parallèle ou en série sur l'antenne ferrite pour faire varier la fréquence de résonance du circuit oscillant au moyen de la constante K du gyrateur. C'est ce que la **figure 14** représente, à gauche le circuit du gyrateur, à droite son circuit équivalent.

La première réalisation pratique, je l'ai calculée pour la réception de DCF, le schéma est à la **figure 15**. Le gyrateur est celui de la figure 6, dans laquelle Z_0 est remplacée par le condensateur C1. L'inductance invertie de la sorte forme alors avec L_a et C2 un circuit oscillant. Dans l'oscillogramme de la **figure 16**, on reconnaît bien le signal de DCF77 par les marques typiques des secondes. Avec R3, il est commode d'accorder la fréquence de résonance et l'on atteint une bande passante de 350 Hz. Ceci démontre la faisabilité du projet.

La **figure 17** présente la variante dans laquelle l'inductance produite se trouve en parallèle sur L_a et C1. La fréquence de résonance est réglable entre 140 kHz et 200 kHz. Le signal reçu de TDF

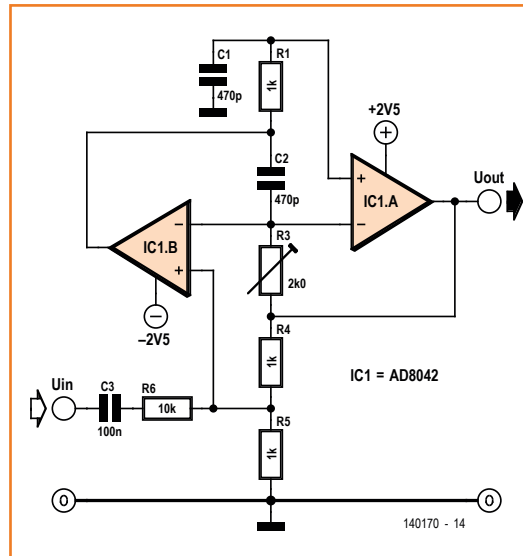


Figure 10.
Schéma pratique d'un filtre RLC à 455 kHz.

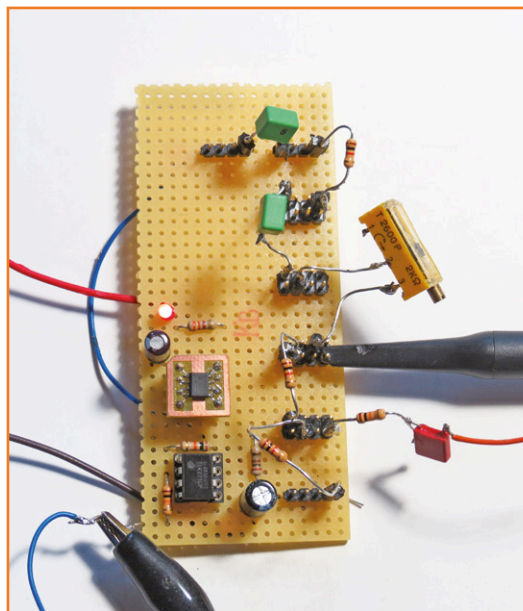


Figure 11.
Plaque d'expérimentation pour circuits à gyrateur.

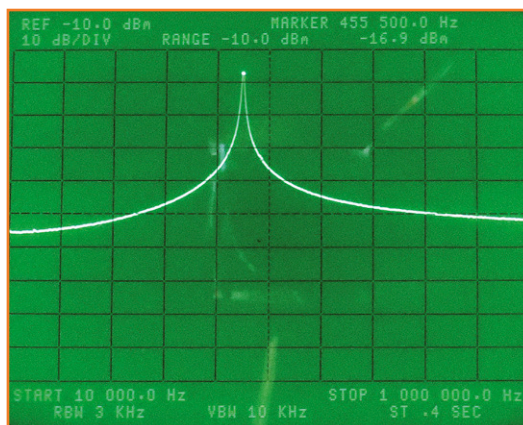


Figure 12.
Réponse en fréquence du filtre à 455 kHz.

Figure 13.
Les schémas d'un condensateur variable.

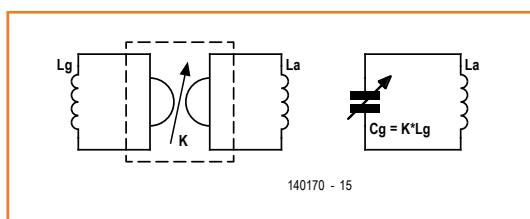


Figure 14.
Principe d'inductances variables.

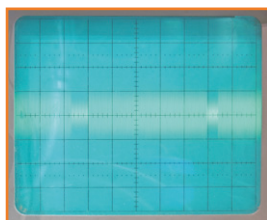
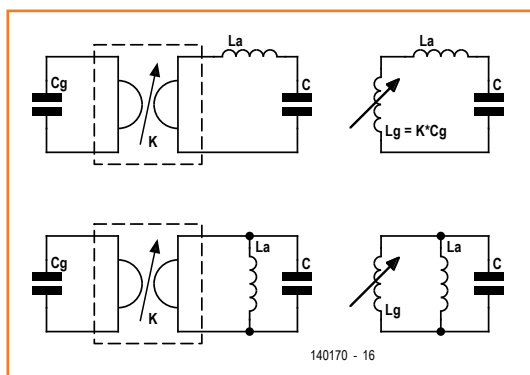


Figure 16.
Les marques des secondes dans le signal de DCF.

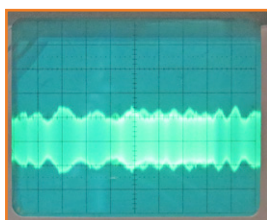


Figure 18.
Le signal de TDF capté sur 162 kHz.

sur 162 kHz est visible à la **figure 18**, mais le même circuit permet aussi de capter confortablement la BBC sur 198 kHz. On atteint alors une largeur de bande de 5 kHz. Ces deux derniers montages trouvent également à s'employer, comme émetteurs, dans les expériences du cours SDR [6]. Le facteur de mérite obtenu dépend des caractéristiques des amplificateurs opérationnels. Une simulation préalable est alors d'un grand secours, surtout pour examiner les questions de stabilité.

Ceci couronne enfin mes efforts pour réaliser ce vieux rêve de syntoniser une antenne ferrite au moyen d'un potentiomètre, sans condensateur variable.

(140170 - version française : Robert Grignard)

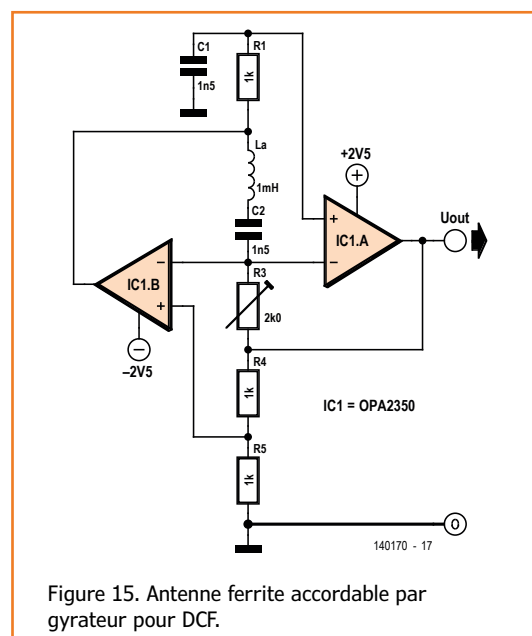


Figure 15. Antenne ferrite accordable par gyrateur pour DCF.

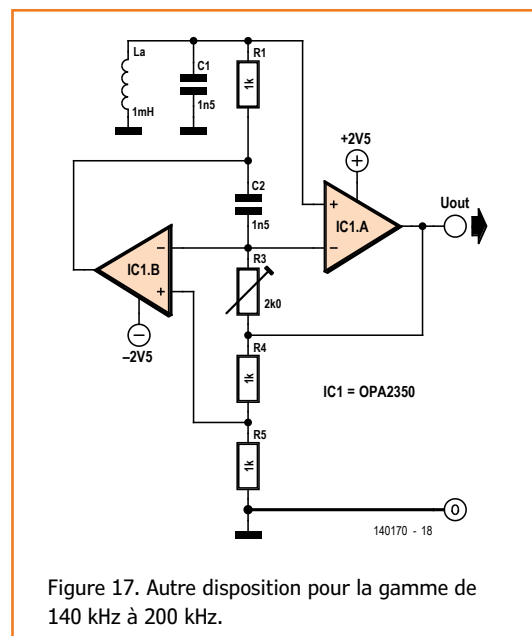


Figure 17. Autre disposition pour la gamme de 140 kHz à 200 kHz.

Liens et littérature

- [1] Elektor nov. 1970 : variable Kapazität bis 100 uF, p. 40
- [2] Elektor mai 1970 : Der Gyrator, eine drahtlose Induktivität, p. 27
- [3] Joseph Carr, Réception des hautes fréquences Tomes 1 et 2, Publitronic - Elektor, 1999 : www.eyrolles.com/Sciences/Livre/reception-des-hautes-frequences-9782866610852
- [4] Linear Circuit Design Handbook, Analog Devices, Elsevier, p. 640
- [5] Lynch, T.H., The right gyrator trims the fat off active Filters, Electronics, juillet 1977, p. 115
- [6] B. Kainka, Construire une radio logicielle (SDR), Publitronic - Elektor, 2008 : www.elektor.fr/construire-une-radio-logicielle-french

2^e édition revue et augmentée du livre l'électronique pour les débutants

NOUVEAU : troisième kit d'initiation disponible

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

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

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

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

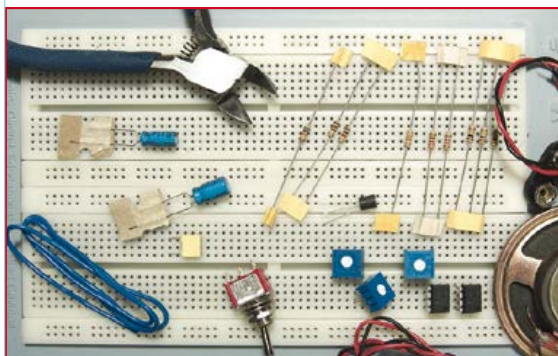
Kit n°1 : orgue | 119016-73 | 24,50 €



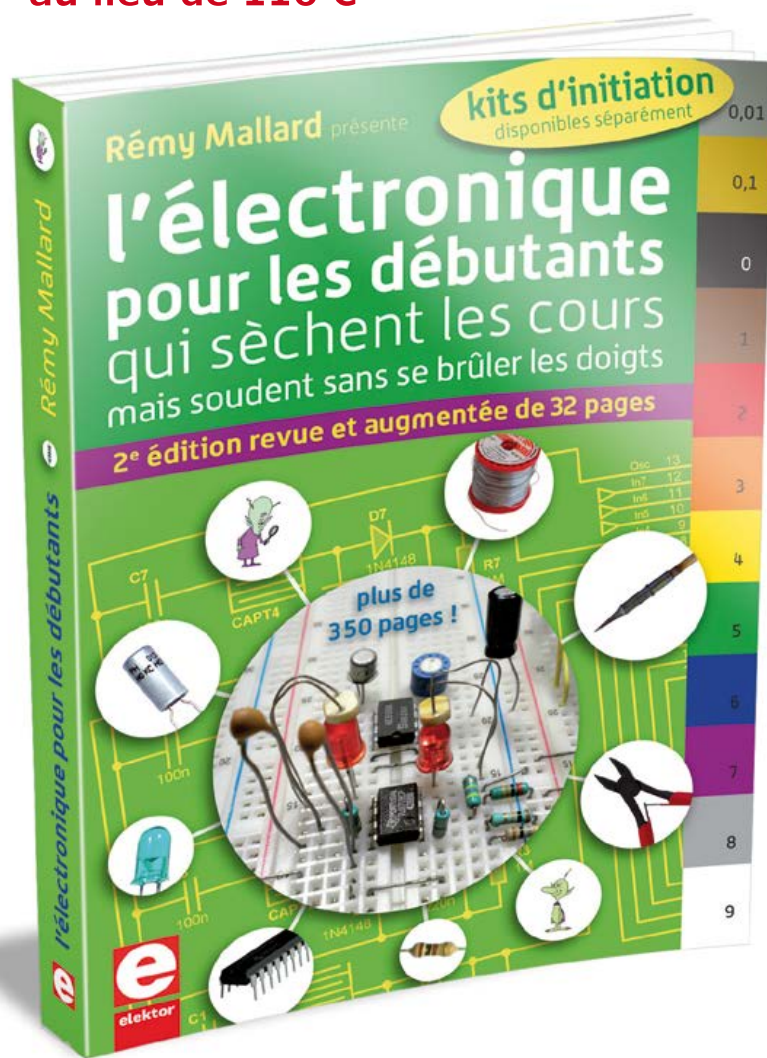
Kit n°2 : chenillard & thermomètre
réf. : 119016-72 | **24,50 €**



Kit n°1 : sirène | réf. 119016-71 | 24,50 €



offre spéciale :
livre + trois kits = 99 €
au lieu de 116 €

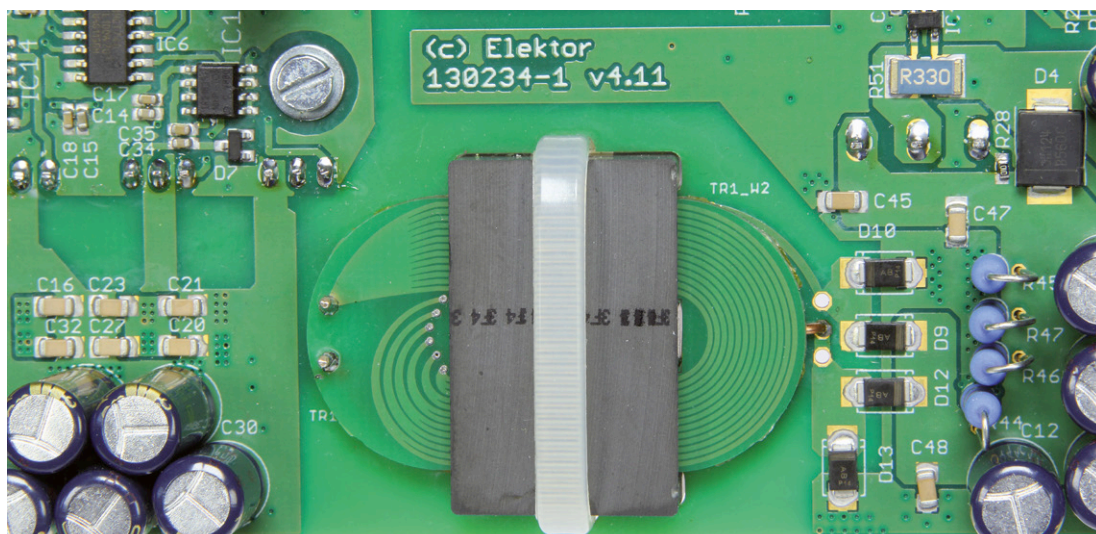


transformateurs à circuits gravés

principe et simulation

Martin Christoph
(Allemagne)

Bobines et transformateurs, l'électronique ne peut pas s'en passer. Dans le projet *Alim de labo PRO* du numéro de septembre 2014, nous avons vu à quel point ils sont indispensables. Particularité, les bobinages du transformateur y sont faits des pistes cuivrées de circuits imprimés. Ici, ce sont les spécificités de ces composants en environnement magnétique et leur simulation qui nous intéressent.



La diversité est de règle pour les transformateurs comme pour les inductances. On a souvent besoin d'un tel composant fait sur mesure pour une application donnée. Pour les transformateurs en particulier, le choix est étroit parce qu'il faut tenir compte de la puissance, mais aussi du rapport de transformation et même de la fréquence de fonctionnement. Et quand d'autres considérations entrent en jeu, comme les pertes magnétiques, il ne reste guère de choix. Sauf celui de faire le transfo soi-même !

Alors, plutôt que de chercher les tôles appropriées et le fil adéquat, il y a une issue possible dans **l'intégration des bobinages à des circuits imprimés**, gravés pour cela, dans lesquels frayer un passage au noyau. La liberté de forme et d'organisation des enroulements est limitée par la structure et par le procédé de fabrication

du circuit imprimé, dont la forme plane ne facilite pas la construction. Voici quelques conseils pour guider vos expérimentations.

Les avantages

Le projet d'*alim de labo PRO* [1] et son transfo planar spécial en ont inspiré plus d'un. Lors des premières approches, nous avons bobiné à la main des transformateurs conventionnels. On s'est vite aperçu que le transformateur serait un obstacle à la reproduction du projet. Il faut trouver le bon fil et le bobiner exactement comme sur le prototype pour conserver le même rapport. Le premier avantage de la construction au moyen de bobinages intégrés au circuit imprimé, aussi la raison principale pour le recommander, c'est que la mise en place et **les caractéristiques sont bien mieux reproductibles**.

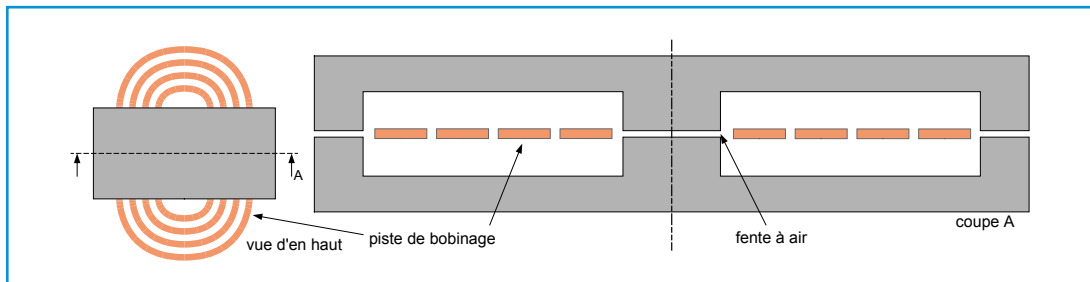


Figure 1.
Représentation schématique
d'une bobine gravée dans un
circuit imprimé.

Autre avantage : un moindre coût, tant pour une construction industrielle que pour la fabrication maison. Pour un petit laboratoire d'électronique, au lieu de stocker tout l'arsenal de câbles et de fils, il est préférable d'assembler des transfo à partir de circuits imprimés, gravés au format adéquat pour la fonction voulue.

Autre aspect non négligeable dans ce type de conception, la qualité des isolants mis en œuvre dans les circuits imprimés multicouches : les enroulements situés dans les couches internes et les vias enfouis (*buried vias*), c'est-à-dire les pontages entre couches internes, tout est ainsi protégé. Les surfaces non utilisées des couches externes servent de blindage pour éviter les lignes de fuite et sont exemptes de traversées. Comme les enroulements sont enfouis dans le circuit imprimé, ils sont bien isolés du noyau et des autres bobines. C'est ainsi que l'on fabrique les produits comme les modules à IGBT (transistor hybride) de 1,7 kV.

On peut concevoir des transfo aux bobinages imprimés à très faibles pertes magnétiques parce que les enroulements primaire et secondaire peuvent y être placés directement l'un en face de l'autre, impossible d'assurer un meilleur couplage. Pour les commutations à haute fréquence, de faibles pertes magnétiques sont un atout majeur.

La **figure 1** montre une coupe transversale dans une spire. L'ensemble se compose de deux demi noyaux en E et un enroulement réalisé en pistes de cuivre. Le noyau a une fente à air habituelle pour les bobines. Vu la symétrie, il suffit d'en observer une moitié.

La **figure 2** montre en haut le flux de perte magnétique indésirable d'une moitié, qui ne se ferme pas seulement par le noyau, mais aussi par la fenêtre qui contient l'enroulement. Ainsi s'installe inévitablement, issu du noyau et principale-

ment au voisinage de l'entrefer, un flux de perte qui pénètre l'ensemble des spires adjacentes dans le circuit imprimé. La partie inférieure de la figure 2 montre comment ce flux induit dans les pistes un courant qui circule sur une face en sens inverse de l'autre. Ce courant se superpose au courant nominal et occasionne des pertes dans la résistance du conducteur. Le courant supplémentaire augmente avec le flux de perte et la fréquence de fonctionnement, puisque la tension induite dans le conducteur dépend de ces grandeurs.

Il ne faudrait pas confondre ce processus avec l'effet de peau qui survient seulement en raison de l'augmentation de densité de courant en périphérie du conducteur lorsque la fréquence augmente. Les pistes larges, mais minces du circuit imprimé offrent moins de prise à l'effet de peau ; c'est tout indiqué pour les hautes fréquences. Ces pertes s'élèvent en fonction de l'intensité du flux qui traverse la surface des pistes. En fait, ce qu'il y a de mieux, ce sont des pistes étroites et hautes (**fig. 3**). Mais cela exigerait une grosse épaisseur de cuivre sur le circuit imprimé, ce

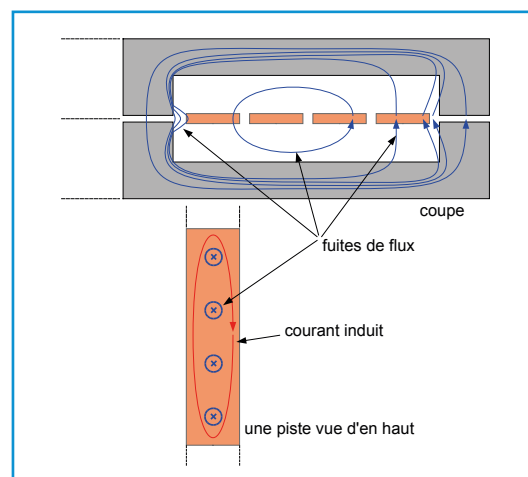
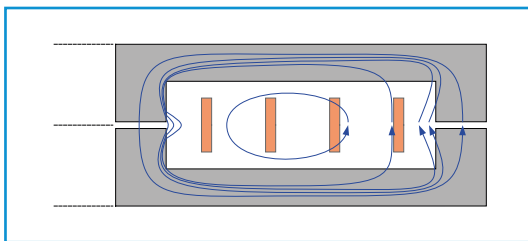


Figure 2.
Flux de perte de la structure
et courant induit résultant.

Figure 3.
Agencement pour le minimum de flux de perte par les surfaces des pistes.



serait cher et difficilement réalisable. Une solution avec des composants magnétiques conventionnels consiste à utiliser simplement des feuilles pour le bobinage comme à la figure 3.

Simulation FEM

Pour une enquête plus approfondie sur les effets de l'induction, il existe une simulation par la méthode des éléments finis ou FEM (*Finite Element Method*). Pour compliquée qu'elle paraisse, de petits arrangements permettent d'afficher facilement les résultats souhaités. Déjà, comme la construction est symétrique, il ne faut en simuler que la moitié. Et si les conducteurs se trouvent au milieu, un quart suffira.

Admettons que la partie intéressante se situe sous le matériau du noyau, nous obtenons un arrangement plan d'une profondeur déterminée. Nous pouvons alors simuler la coupe à travers le noyau avec la profondeur du noyau comme paramètre. Il existe pour ce faire le programme FEMM, la branche magnétique de FEM, que l'on peut télécharger gratuitement [2] avec la documentation. Les explications étape par étape simplifient la mise en route de la simulation. Contrairement aux méthodes habituelles, l'exactitude des paramètres n'est pas tellement importante.

Le programme une fois chargé et installé, on le lance et par le menu *File* → *New* pour rédiger en dialogue un nouveau « *Magnetics Problem* ». Par le menu *Problem* on arrive au dialogue « *Problem Definition* ». Comme *Problem Type* on choisit « *Planar* », « *Millimeter* » pour les unités et pour la fréquence p. ex. 100 kHz. Pour un noyau E32/6/20 la profondeur est de 20 mm.

Pour la géométrie, il y a cinq boutons dans la barre d'outils, par lesquels vous pouvez spécifier le genre d'objet : *Node*, *Segment*, *Arc Segment*, *Label* ou *Group* – voyez aussi le menu *Operation*. L'objet choisi, vous pouvez par un clic ou avec la touche Tab (avec *Node*, *Label* et *Group*) créer un nouvel exemplaire, le sélectionner d'un clic droit et accéder aux propriétés de l'objet avec la

barre d'espacement. Le maniement est un peu particulier, mais on s'y fait vite.

D'abord, le programme trace une limite autour du problème. Si vous considérez qu'il n'y a pas de flux magnétique qui s'échappe du noyau, vous aurez de mauvais résultats pour l'extérieur de l'entrefer. Alors, entourez d'air le dispositif. Un demi-cercle de dimension appropriée évite de définir une zone infiniment grande. Choisissez donc *Operate on nodes*, appuyez sur Tab et saisissez les coordonnées $x = 0$ et $y = -20$. Sélectionnez ensuite *Operate on segments* et cliquez, pour les relier par une ligne perpendiculaire, d'abord sur le point inférieur, puis sur celui d'en haut. Choisissez alors *Operate on arc segments*, cliquez de nouveau sur les deux points, dans la boîte de dialogue qui s'ouvre prenez pour *Arc Angle* 180 et le demi-cercle est tracé. La suite de la construction de la (demi) géométrie se fait de la même manière.

Vous pouvez contrôler totalement le programme au moyen d'un script personnel en Matlab ou Octave. Le manuel vous donne toutes les explications nécessaires.

On dessine le bobinage comme à la figure 1. Pour des expériences, les mesures exactes ne sont pas importantes. Pour information, une piste normale fait 35 μm de haut. Il est facile de construire la géométrie de la moitié du noyau avec les données du feuillet de caractéristiques [3] pour un noyau E32/6/20. Comme entrefer entre les moitiés du noyau, on peut prendre 0,2 mm.

Le choix du matériau s'opère par *Properties* → *Materials Library*. On peut le faire glisser à la souris du champ de gauche vers la droite dans « *Model Library* ». Pour commencer, on prend *Air*, *Copper* et *Soft magnetic ferrite*.

Avec *Properties* → *Circuits* on instaure une boucle de courant. Avec *Add Property* du dialogue, on définit une spire du nom de « *Winding1* » dans la configuration *Series* pour un courant de 0.1 A. Il faut à présent donner aux domaines circonscrits des *block labels* et les sélectionner d'un clic droit. Avec la barre d'espacement, on obtient les dialogues de propriétés correspondantes dans lesquelles les matériaux ont été désignés. Pour le bobinage, on a déjà indiqué dans *in circuit* le « *Winding1* » et pour *number of turns* on met un 1.

Reste maintenant à préciser les conditions aux limites pour le demi-cercle formé précédemment. En raison de la symétrie, il convient de donner $A = 0$ pour la ligne à angle droit qui représente le milieu du dispositif. On va dans *Properties* sur *Boundary* et on clique sur *Add Property*. Comme *Name* on saisit p. ex. $A=0$ et pour *BC Type* on choisit « *Prescribed A* ». Les *Prescribed A parameters* restent tous à « 0 ». Après fermeture du dialogue, on sélectionne d'un clic droit dans *Operate on segments* la ligne à angle droit du demi-cercle. Puis avec la barre d'espacement, on atteint la boîte de dialogue pour choisir la propriété indiquée $A=0$.

La limite en demi-cercle donne à penser qu'il y a là une quantité infinie d'air. Comme avec la précédente, on va y ajouter une limitation supplémentaire. On choisit comme *BC Type* « *Mixed* » et comme coefficients :

$$c_1 = 0; \quad c_0 = \frac{1}{\mu_0 r_0} = 39788735,77$$

$$\text{où } r_0 = 0,020 \text{ m}$$

Vous trouverez le calcul dans le manuel FEMM page 155.

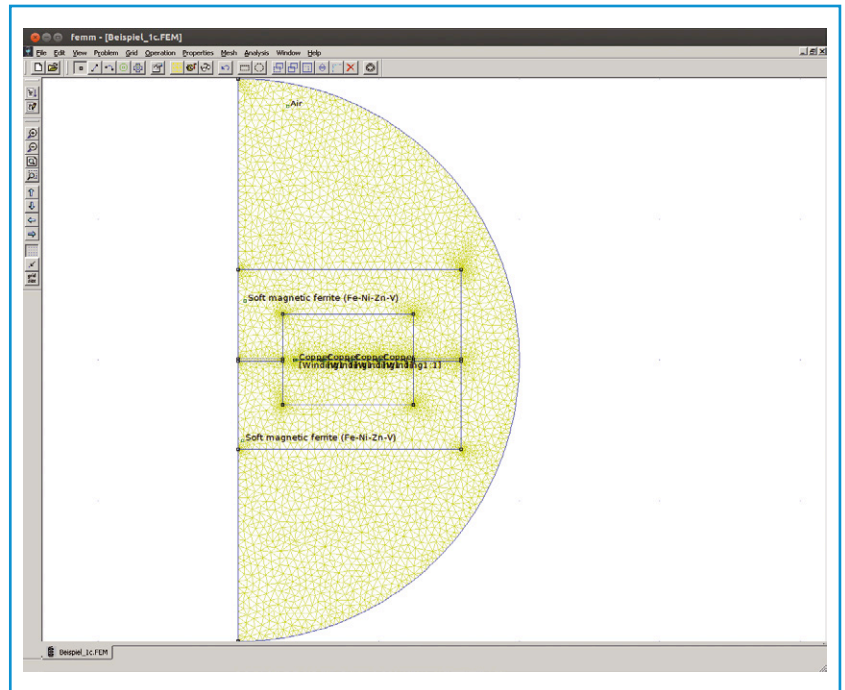
Cette seconde limitation est appliquée au demi-cercle par un clic droit sur *Operate on arc segments* et la barre d'espacement.

Nous avons ainsi tout spécifié, cliquons dans la barre d'outils sur *Run mesh generator*. À l'issue du processus, un clic sur *Run analysis*. Si ça n'a pas marché, c'est sans doute parce que quelque part la géométrie n'a pas été fermée ou qu'un matériau n'a pas été attribué à un domaine. Pour tous les domaines fermés dans le demi-cercle, un matériau doit avoir été attribué, même celui rempli d'air. Finalement, vous devriez voir une construction semblable à la **figure 4**.

Les fichiers FEMM pour les trois parties sont disponibles gratuitement [4].

Résultats

Avec *View results*, vous avez à présent le loisir d'analyser tous les résultats possibles. Sous *View* → *Density plot*, vous pouvez connaître p. ex. avec $|Flux Density|$ (T) la densité de flux dans le noyau. Vous pouvez ainsi vérifier si le matériau du noyau est déjà entré en saturation ici où là. Si vous choisissez $|Current Density|$ (MA/m²), vous aurez la répartition du courant dans



le conducteur. On voit ici qu'elle n'est pas uniforme, mais qu'elle est influencée par les champs de fuite repérés. C'est particulièrement visible à proximité de l'entrefer. Une augmentation de fréquence amplifie encore l'effet. La **figure 5** présente le résultat d'une simulation dans un conducteur pour $f = 100$ kHz.

Figure 4.
Schématisation en FEMM.

Operation Contours permet de tracer une ligne à travers le dispositif pour représenter une grandeur dans un graphe X-Y. Vous pouvez ainsi tracer une ligne qui coupe les pistes cuivrées et y voir la répartition du courant.

Figure 5.
Simulation : les lignes suivent le flux magnétique, la couleur indique la densité de courant.

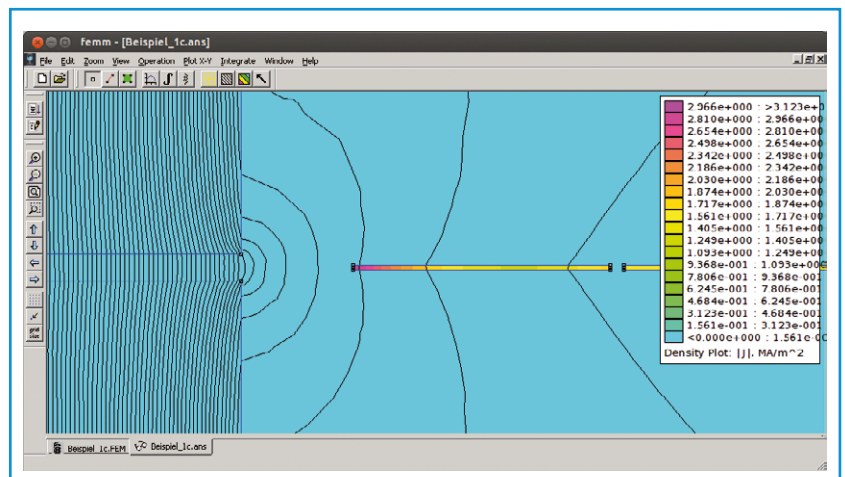


Figure 6.
Trois configurations
différentes des pistes,
mais
surfaces identiques.

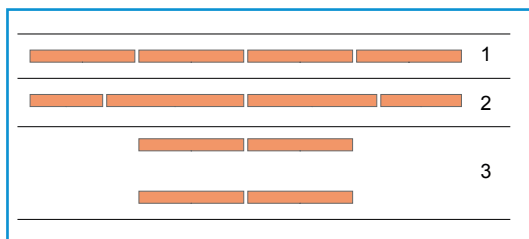


Tableau 1.
Comparaison des pertes de puissance

Modèle	Pertes de puissance	par rapport à 1
1	129,634 mW	±0 %
2	126,359 mW	-2,5 %
3	114,508 mW	-11,7 %

Très intéressant aussi : *View Circuits Probs* qui vous donne directement les caractéristiques du dispositif comme le coefficient d'induction et la perte de puissance. Avec les pertes, vous pou-

vez évaluer quelle modification est préférable. Le **tableau 1** donne les résultats de trois dispositions des conducteurs de la **figure 6** pour une même quantité totale de cuivre.

On voit ainsi que c'est l'arrangement uniforme des bobinages sur le circuit imprimé qui est le moins avantageux. Avec des largeurs différentes des pistes, les pertes diminuent déjà de 2,5 %. Il est aussi préférable d'utiliser les deux faces d'un même circuit imprimé en laissant le bord libre. Mais avec un circuit imprimé à double face on n'arrive plus à passer de l'intérieur du bobinage vers l'extérieur avec une piste. Malgré tout, des pertes d'au moins 10 % inférieures justifient pleinement le pont de câblage supplémentaire.

Conclusion

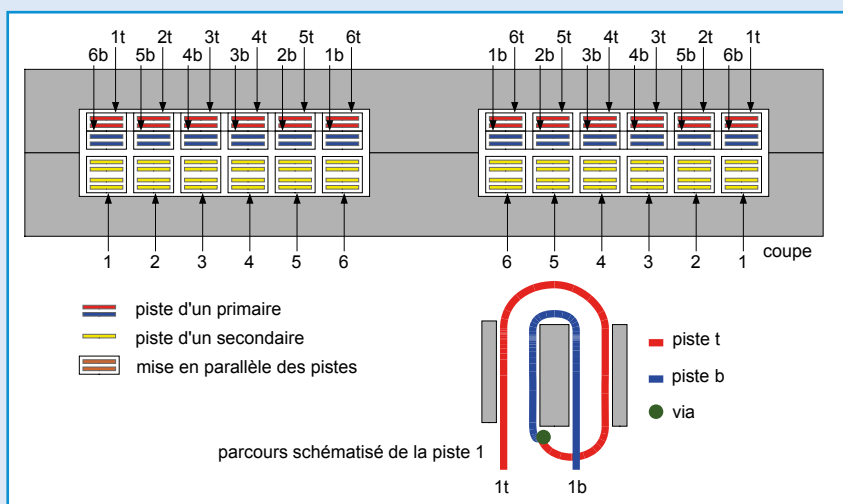
Cet article n'apporte qu'un bref aperçu de l'optimisation des composants magnétiques. Il a cependant montré que même de légères modifications du tracé conduisent à une réduction des pertes. Néanmoins, l'intégration sur circuit

Agencement du transfo à circuits imprimés de l'alim Pro

La structure du transformateur dans l'alimentation de laboratoire du numéro de septembre 2014 résulte de diverses considérations et conclusions, mais n'a pas été simulée dans sa totalité. La figure montre la coupe transversale à travers le bobinage du transfo. On voit bien que les pistes conductrices y sont systématiquement branchées en parallèle surtout pour supporter le courant voulu. Le bobinage secondaire (jaune) compte six spires et ne s'étend pas sur toutes les couches, seules quatre d'entre elles sont reliées en parallèle. On a ainsi quatre chemins parallèles sans vias, connectés l'un à l'autre au début et en fin de piste. En outre, des pistes étroites réduisent l'influence des champs de pertes. Comme les quatre chemins sont directement superposés et donc présentent la même surface, ils sont aussi traversés par des flux équivalents, ce qui induit dans chaque conducteur la même tension et garantit un partage de courant équitable entre les lignes du circuit.

L'arrangement du primaire est différent. En raison de l'échange des lignes de circuit branchées en parallèle aux extrémités, toutes ces pistes ont à peu près la même surface. Chaque ligne est donc traversée par le même flux magnétique et reçoit par induction la même tension, donc le même courant. Vu que la perte dans la résistance varie comme le carré du courant, une distribution uniforme du courant entraîne un minimum de perte. Si l'on observe bien le circuit imprimé du transformateur, on

s'aperçoit que les lignes conductrices échangées individuellement sur un tour se divisent de nouveau et se reconnectent. C'est la conséquence du processus de conception, parce qu'en fin de compte, il ne restait plus assez de vias possibles. Électriquement, ce n'est pas ce qu'on pouvait rêver de mieux, mais la réalité ne rejoint pas toujours l'idéal...



imprimé ne s'adapte pas forcément bien à n'importe quelle application. Mais la prendre en considération dès la conception d'ensemble permet d'aboutir à des solutions intéressantes. Avec des inductances, les applications dans lesquelles la composante continue est prépondérante, comme le convertisseur à découpage ou le filtre pour tension continue, l'avantage est considérable, du fait que les courants de Foucault y sont peu présents. Quand la composante alternative est grande, toute votre attention est requise lors de la conception des bobinages. Avec de puissants champs de fuite, laisser beaucoup de cuivre à proximité de l'entrefer n'est pas vraiment judicieux. Sur les transformateurs, il n'y a normalement pas d'entrefer, donc moins d'effet négatif. C'est en raison de ses très faibles fuites d'induction que le transfo sur circuit imprimé est particulièrement intéressant dans les applications à haute fréquence. Cette solution compacte se distingue aussi par la facilité de construction.

(130487 - version française : Robert Grignard)

Liens

- [1] www.elektor-magazine.fr/130234
- [2] www.femm.info
- [3] www.farnell.com/datasheets/1632190.pdf
- [4] www.elektor-magazine.fr/130487

L'auteur

Martin Christoph est ingénieur en électrotechnique et a travaillé à l'ISEA (Institut d'électronique de puissance et d'entraînements électriques) de l'université technique d'Aix-la-Chapelle (Allemagne) dans le domaine de l'électronique de puissance.

Publicité

The new EAGLE has landed!

Version 7

now available

For more information, visit www.cadsoftusa.com



... what are *your* plans
between 11th & 14th November 2014?

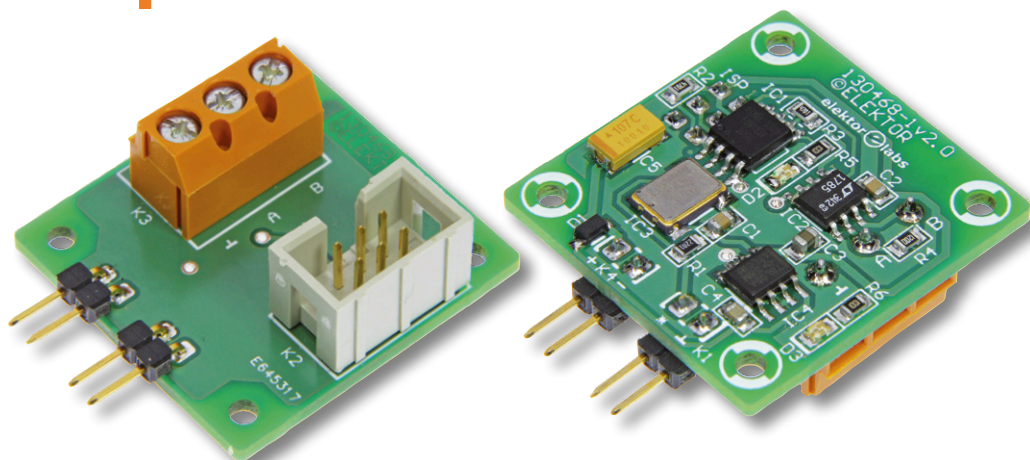


module capteur de température à interface RS-485

Technique :

André Goldberg,
Mauk van der Laan
et **Ton Giesberts**

Rédaction : **Jens Nickel**



Nombre de tâches d'automatisation requièrent des capteurs de température (°T). L'interface RS-485 permet un transfert fiable des valeurs sur de longues distances. Notre module capteur de °T utilise un µC ATtiny et un pilote RS-485 ; on peut y connecter plusieurs capteurs de température en parallèle. Elektor propose un programme standard qui, pour la transmission des valeurs mesurées, fait appel au protocole ElektorBus. Cerise sur le gâteau, nous vous proposons un logiciel pour l'affichage des données.

Le succès, en 2011, de la série d'articles consacrés à l'ElektorBus est encore dans toutes les mémoires : des centaines de courriels de lecteur décrivant leurs développements personnels, avec de précieux retours d'expérience. André Goldberg est indubitablement l'un des fans les plus assidus de l'ElektorBus ; il a réalisé un module de suivi et de commande de la température ambiante. L'un des attraits majeurs de l'ElektorBus sont ses nœuds qui, reliés à des capteurs de température, pourraient relayer des mesures vers un ordinateur.

Le charme du protocole ElektorBus est sa simplicité : chaque message compte 16 octets. Une carte de quatre capteurs peut transmettre par exemple à un automate de commande quatre entiers (de -1023 à 1023), c'est-à-dire les valeurs mesurées par ses quatre capteurs. Un message envoyé dans l'autre direction pourrait par exemple demander au capteur de transmettre, dorénavant, ses mesures en degrés en Celsius ou en Fahrenheit ; une autre possibilité est de confi-

gurer un intervalle de mesure. Pour tout cela et bien davantage, le protocole ElektorBus connaît des commandes adéquates [1].

Pour le transfert des octets, on utilise l'interface RS-485 qui, en mode semi-duplex, requiert deux lignes + une ligne de masse [2]. Par principe, RS-485 est relativement insensible aux parasites. Avec le taux de transmission modéré de l'ElektorBus, 9600 bauds, les lignes peuvent dépasser 30 m.

Mini-nœuds de bus

Elektor a décrit plusieurs projets (avec platine) pour les nœuds de l'ElektorBus [3][4], trop grandes cependant pour certaines applications. C'est pourquoi André Goldberg a imaginé une carte compacte de 18 sur 26 mm seulement, qu'il a proposée à notre laboratoire. Son microcontrôleur ATtiny45, avec six broches d'entrées-sorties polyvalentes (*General Purpose I/O*), est en boîtier CMS tout comme le pilote RS-485. La connexion des lignes de bus se fait par îlots de soudure.

Pour obtenir un circuit compact, les broches de programmation ISP et deux des broches GPIO du μ C sont aussi des îlots de soudure. Il n'y a pas d'oscillateur externe. Le DS18S20 choisi est un capteur de température *1-Wire*. Il n'occupe qu'une des broches GPIO, car la valeur de température mesurée est transmise du capteur au μ C par protocole asynchrone ; il n'y a donc qu'une seule ligne de données et pas de ligne d'horloge [5]. Cette ligne de données permet en outre l'alimentation des capteurs. Notez qu'il est même possible de connecter plusieurs capteurs *1-Wire* (= à 1 fil) en parallèle sur la même broche de port. Le fabricant attribue à chaque capteur un ID (**ID**entifiant) indélébile codé sur 64 bits. Grâce à cet ID, le μ C peut lire la température d'un capteur spécifique. S'agissant, dans le cas du DS18S20, de valeurs de 9 bits, la résolution est de 0,5 °C.

Le cahier des charges

Les caractéristiques du matériel imposent au logiciel les exigences suivantes :

- L'ATtiny45 ne possède pas d'UART matériel pour piloter le circuit de commande RS-485. Il faut donc réaliser un UART (émission et réception) logiciel, avec deux des broches GPIO du microcontrôleur.
- Il faut une bibliothèque pour la lecture du capteur *1-Wire*. La lecture et l'écriture des

données requièrent une autre broche GPIO. La présence possible de plusieurs capteurs sur une même ligne, impose au microcontrôleur de connaître les ID des différents capteurs pour pouvoir en cartographier les résultats de mesure.

- L'ATtiny ne possède pas suffisamment de mémoire de programme pour pouvoir, en supplément, utiliser la bibliothèque du protocole ElektorBus déjà décrite dans Elektor [6] [7]. Il faudra donc implémenter manuellement les parties du protocole nécessaires.

Pour l'UART logiciel et le bus *1-Wire*, André Goldberg utilise deux bibliothèques à code source ouvert. Il a commencé par lire les adresses de chacun des capteurs °T (à l'aide d'un micrologiciel différent) pour ensuite les coder de façon fixe dans le logiciel. Pour pouvoir transmettre quatre valeurs de température en degrés Celsius, le programme commence par multiplier les valeurs de 10, pour ensuite, conformément au protocole ElektorBus, les coder sous la forme de quatre nombres entiers. André a ensuite compilé *manuellement* dans le logiciel les octets restants du message ElektorBus.

L'électronique

Notre collègue du labo, Ton Giesberts, a *elektorisé* le circuit imprimé d'André. Les connexions

Liste des composants

Résistances (0805) :

R1 = 2,2 k
R2,R3 = 4,7 k
R4 = 120 Ω
R5,R6 = 1 k

Condensateurs :

C1,C2,C4 = 100 n, 25 V, 10 %, X7R, CMS 0805
C3 = 10 μ , 25 V, 10 %, X5R, CMS 0805
C5 = 100 μ , 16 V, 20 %, tantale, CM, boîtier F

Semi-conducteurs :

D1 = PMEG2010AEH
D2 = LED jaune, CMS 0805
D3 = LED verte, CMS 0805
IC1 = ATtiny85-20SU, CMS SO-8S2
IC2 = LT1785, CMS SO-8
IC3 = oscillateur à quartz 8 MHz, 5 x 7 mm, CMS (LF SPX0019079)
IC4 = 78L05

Divers :

K1,K4 = embase 1x2 contacts, pas de 2,54 mm, en équerre
K2 = embase 2x3 contacts ou connecteur 2x3, pas de 2,54 mm
K3 = bornier à vis 3x1, pas de 5,08 mm
circuit imprimé 130468-1 v2.0

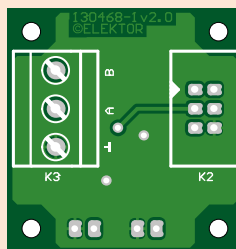
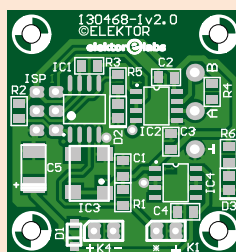


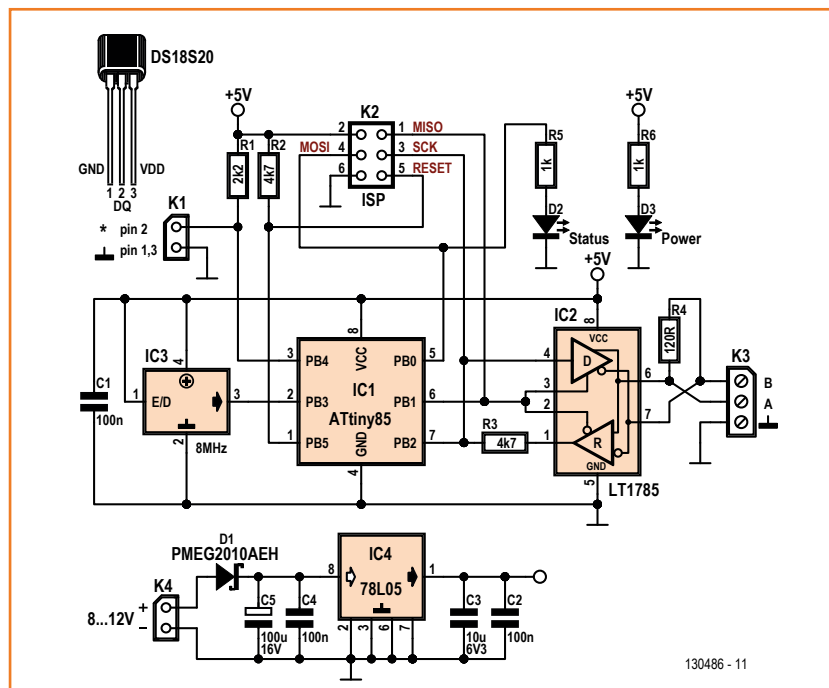
Figure 1.
Le circuit imprimé simple face ne mesure que 32 x 29 mm. La mise en place des connecteurs se fait par le dessous.

se font toutes sur des embases ou des borniers à vis, implantés en fonction des besoins. Il a rajouté un oscillateur à quartz, garant d'une meilleure fiabilité de la transmission, précieuse en cas de fortes variations de température. Malgré ces adjonctions, le circuit double face de Ton reste compact (**fig. 1**) : 32 x 31 mm [8]. Les borniers à vis de l'ElektorBus, les embases 2x3 de programmation, les deux embases 1x2 à 90° pour les capteurs *1-Wire* et l'alimentation sont en dessous. Les composants CMS peuvent être soudés à la main.

Nous devons à Mauk van der Laan une amélioration décisive du circuit. Au lieu d'accaparer deux des six broches GPIO pour l'émission et la réception de l'UART logiciel, il n'en utilise qu'une, alternativement pour l'émission et la réception puisqu'en communication *semi-duplex* du bus RS-485, un nœud de bus n'« écoute » et ne « parle » jamais simultanément. Avantage : la broche GPIO libérée permettrait de connecter une LED de visualisation des états.

Au cœur du schéma de la **figure 2**, l'ATtiny85 [9], doté de plus de mémoire Flash que l'ATtiny45. Toutes ses six broches de port sont utilisées, certaines de façon multiple. PB4 est le point de connexion des capteurs de température *1-Wire* (par le biais de l'embase K1). PB3 se voit connecter l'oscillateur à quartz intégré CMS. L'interface SPI PB0...PB2 et la broche de réinitialisation PB5

Figure 2.
Au cœur du schéma, un ATtiny85. L'UART logiciel ne requiert qu'une broche (PB2), car il n'y a pas de réception et d'émission simultanément.



constituent, ensemble, l'interface ISP, qui permet la programmation du minuscule microcontrôleur AVR. En fonctionnement, la commande de la LED d'état se fait par PB0.

Le niveau sur PB1 sert à faire basculer l'émetteur-récepteur RS-485 entre *émission* (niveau haut) et *réception* (niveau bas). PB2 est la broche évoquée plus haut par laquelle se font, respectivement, l'émission et la réception des signaux de l'UART.

Le bornier K3 reçoit les lignes de l'interface RS-485. L'implantation de la résistance de terminaison de 120 Ω est facultative.

Configuration

Nous proposons pour l'ATtiny85 un micrologiciel standard (téléchargement gratuit et code source ouvert [8]). Mauk van der Laan y a intégré une bibliothèque d'UART logiciel, une bibliothèque *1-Wire* (bibliothèque Arduino modifiée) et une petite interface ElektorBus. Le pilotage de l'ensemble est assuré par un pseudo-système d'exploitation capable d'un multitâche rudimentaire (cf. encadré). Le code C++ annoté mérite votre attention de lecteur, étudiez-le, vous y trouverez des choses intéressantes.

On peut connecter à K1 jusqu'à quatre capteurs $^{\circ}\text{T}$ du type DS18S20, en parallèle ; conformément à la fiche de caractéristiques, il faut forcer à la masse chacune de leurs broches VDD, sachant que l'alimentation des capteurs se fait par la ligne de données. Le logiciel utilise quatre emplacements de mémoire (*Slots* 1...4) dans l'EEPROM de l'ATtiny pour les identifiants (ID) des capteurs de température. À la mise sous tension, le microcontrôleur détecte les ID de tous les capteurs présents sur le bus *1-Wire* ; il les compare ensuite aux identifiants enregistrés. S'il ne trouve pas d'ID connu, le logiciel libère l'emplacement de mémoire correspondant. Si, au contraire, un ID est inconnu, il l'enregistre dans le premier emplacement de mémoire libre.

On voit donc clairement maintenant le processus à suivre lors de la configuration du système. On commence par connecter l'un des capteurs *1-Wire* et on démarre la carte par application de l'alimentation. L'ID est alors enregistré dans le 1er emplacement de mémoire. L'approche la meilleure consiste à doter le capteur d'un autocollant (« 0 » ou « 1 » selon que l'on programme en Basic ou en C ;-)). Ensuite, couper l'alimentation, connecter le second capteur à K1 et remettre sous tension. Après lecture de l'ID du second capteur et son

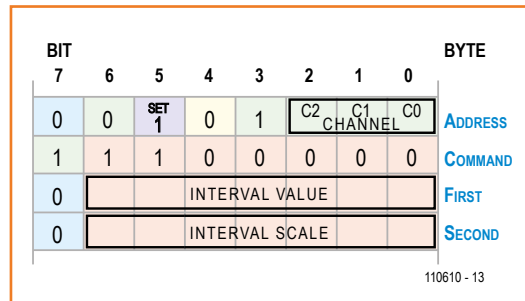
enregistrement, on utilise le même processus pour les capteurs numéro trois et quatre.

La LED d'état visualise très élégamment la progression de la configuration. Au démarrage du logiciel, elle clignote rapidement, pour ensuite s'allumer quatre fois brièvement ou un peu plus longuement. « Brièvement » est synonyme d'un emplacement de mémoire libre, « longuement » indique une adresse ID déjà détectée.

Si l'on a connecté un capteur °T pour lequel on ne souvient plus à quel emplacement de mémoire il correspond, on pourra tout simplement le déconnecter, remettre le circuit sous tension et faire effacer l'emplacement de mémoire correspondant. Ensuite, on reprend le processus de configuration décrit.

Interface de l'ElektorBus

Les quatre capteurs °T représentent quatre sous-nœuds (*subnode*). Notre micrologiciel standard envoie, avec espacement, les quatre valeurs de

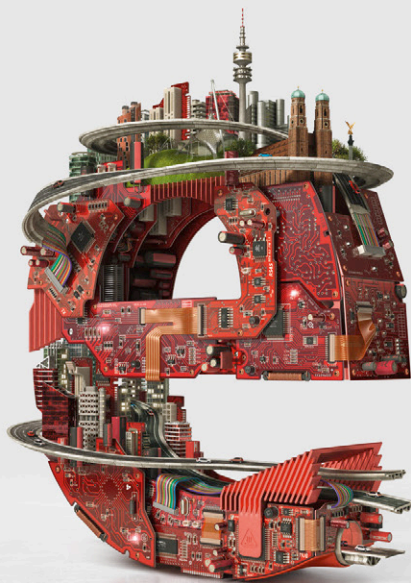


température mesurées dans un message ElektorBus. À cet effet, on utilise les quatre *Channels* (canaux) du message ; nous pouvons transmettre, sur chaque canal, une valeur comprise entre -1023 et +1023 [1]. Chaque canal occupe, dans la zone de données d'utilisateur du message, deux octets (octets 6...13, comptés à partir de 0). L'adresse de l'expéditeur (octets 4 et 5) est, dans notre nœud, fixée à « 5 » une fois pour toutes. Les octets 2 et 3 contiennent l'adresse du desti-

Figure 3.

Structure du message de paramétrage de l'intervalle de mesure sur le module capteur de °T (ne sont visualisés ici que les octets 6...9) *Interval Scale* est à 9, constituant la clé des dixièmes de seconde. Il serait possible, par le biais des bits de canal (CHANNEL), de paramétrer un intervalle spécifique pour chacun des capteurs de °T, le micrologiciel standard n'utilisant cependant pas cette option.

Publicité



Welcome to Planet e.

Tout l'univers de l'électronique en un seul lieu !

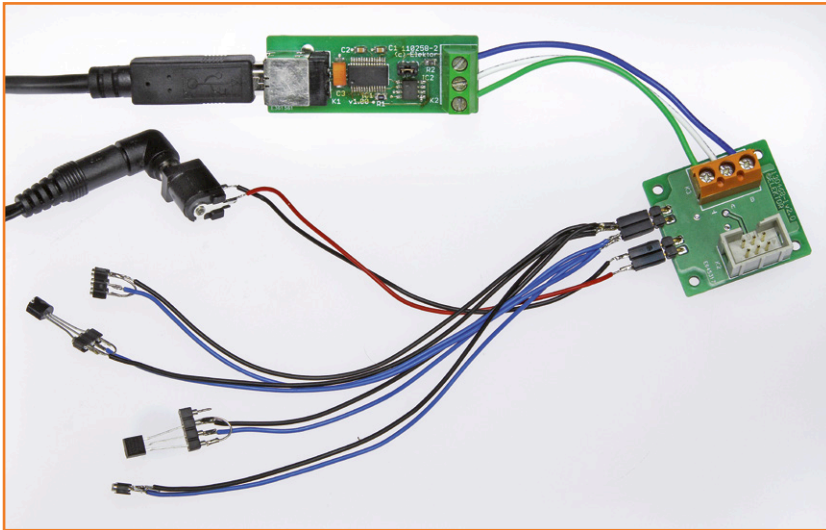
Billets et enregistrement
www.electronica.de/en/tickets

26e Salon Mondial des Composants,
 Systèmes et Applications en électronique
 Messe München
 11-14 novembre 2014
www.electronica.de

50 years
 electronica

 **electronica** 2014
 inside tomorrow

Figure 4.
Le module capteur de °T envoie ses valeurs de mesure à un PC par RS-485.



nataire, qui, dans notre micro logiciel standard, est toujours fixée à « 10 ». Les octets 14 et 15 (Somme de vérification/CRC) restent inutilisés. Le micrologiciel standard est paramétré pour une communication de 1:1, mais pas pour travailler avec plusieurs cartes de capteurs sur un bus. Dans ce qu'on appelle le *DirectMode*, on évite

les collisions en définissant tout simplement un agenda d'émission strict. Le nœud de capteur envoie les valeurs de température à intervalle spécifié au *Master (Maître)*, de préférence un logiciel de commande sur le PC). Si le Maître doit envoyer une commande au nœud de capteur, il attend l'arrivée d'un message comportant des valeurs de mesure, et envoie son propre message immédiatement après. Avec la vitesse de bus normale de 9 600 bauds, les intervalles de mesure peuvent durer jusqu'à 100 ms ; le temps de conversion des capteurs °T, bien plus long, de 750 ms (max.), amène à opter pour des intervalles de plusieurs secondes.

Le micrologiciel standard créé par Mauk envoie les valeurs de température des capteurs °T identifiés à un intervalle de 1 s par défaut. Les valeurs de la température en degrés Celsius sont alors multipliées par 10. Simultanément, le nœud de capteur est à l'écoute d'une commande de modification de l'intervalle : options disponibles : 5 s et 500 ms (**fig. 3**). Il est en outre possible, depuis le système de commande, de basculer l'unité de température vers les degrés Fahrenheit (mais aussi de revenir aux degrés Celsius). Lorsque le

Multitâche sur l'ATtiny

Le micrologiciel standard a été programmé par Mauk van der Laan, développeur et concepteur indépendant. Le logiciel de cet auteur est modulaire ; un système d'exploitation propriétaire, basé sur les automates (finis), assure la synchronisation correcte des *Tasks* (= tâches) individuelles (cf ci-après).

Le fichier *Onewire.h* contient la bibliothèque *1-Wire*. *OneWireTask.cpp/h* est l'automate responsable de la détection des ID et de la lecture des valeurs de température. L'auteur a dérivé le pilote d'UART logiciel *ElektorBusSw1w.cpp/h* pour la communication RS-485 en semi-duplex d'une bibliothèque RS-232 à code source ouvert. À noter que l'interface ElektorBus *ElektorBus.cpp/h* est également capable de travailler avec un UART matériel. Dans *config.h* on a défini que *ElektorBus.cpp* s'adresse, dans le cas du micrologiciel standard, à l'interface de l'UART logiciel :

```
#define ELEKTORBUS_DRIVER_INCLUDE «ElektorBusSw1w.h»
```

ElektorBusTask.cpp/h est l'automate chargé du pilotage de l'interface de bus.

Le système d'exploitation de Mauk van der Laan permet la mise en œuvre d'un traitement multitâche dit coopératif. Contrairement au multitâche préemptif, il n'y a jamais ici

d'interruption de tâche par interruption venant de l'extérieur, mais le contrôle est rendu délibérément à un ordonnanceur (*scheduler*). On n'a pas besoin, par conséquent, de mécanismes de verrouillage pour éviter une interruption à un mauvais endroit. Contrairement aux vrais systèmes d'exploitation enfouis, tels que FreeRTOS, les tâches ne disposent pas ici de leur propre pile. Au contraire, chaque tâche ne se voit assigner qu'une seule variable d'état, d'où une empreinte mémoire réduite. Ceci explique que le système d'exploitation puisse aussi s'exécuter sur un ATtiny avec 8 Ko de Flash et 2 Ko de RAM.

Chacune des tâches est implémentée à l'aide d'automates finis (*state machine*). L'ensemble du code est intégré dans une expression *SwitchCase* ; il se produit, en fonction de la valeur d'une variable d'état, un aiguillage vers la section de code correspondante. À la fin, la tâche appelle la méthode *nextState()* et rend le contrôle à une instance centrale (*Runtime*). Sinon, la tâche peut aussi appeler aussi *nextDelay()*, avec l'indication d'un délai en millisecondes. Par cette requête, la tâche demande à ne plus être appelée pendant un certain temps (*Delay* = retard).

L'ordonnanceur du *Runtime* gère une liste des tâches en cours d'exécution, qui sont toutes dérivées de la classe *Task*.

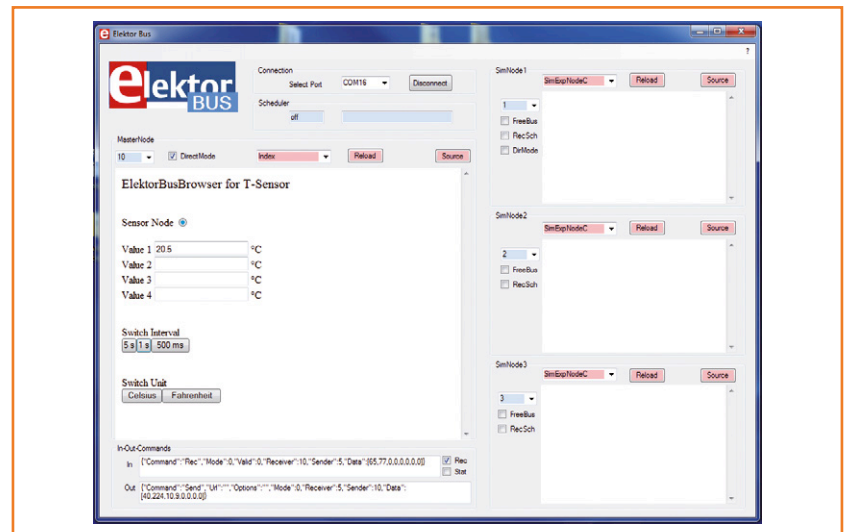
nœud de capteur reçoit un tel message, il envoie dès le premier intervalle suivant, dans un message propre, les nouveaux états réels de l'unité et de l'échelle (pas de message de valeurs mesurées dans ce cas-là). Ensuite, on poursuit avec les valeurs de mesure. Comme les valeurs exprimées en degrés Fahrenheit pourraient dépasser 102, le capteur configure automatiquement, à la réception d'un message demandant un passage aux degrés Fahrenheit, la mise à l'échelle en la faisant passer de -1 (dixième de degré) à 0 (degré entier).

Logiciel PC

Pour un premier test, un programme de terminal fait l'affaire ; il permet de voir si la carte envoie des valeurs de mesure une fois par seconde. Nous avons aussi conçu un logiciel de démonstration personnalisé pour le PC. L'interface est écrite en HTML/Javascript. Le fichier à télécharger [8] comporte le logiciel ElektorBusBrowser.exe et le dossier UIBus que l'on traînera vers le bureau. Via RS-485, nous connectons le module de capteur au convertisseur RS-485/USB d'Elektor [3] puis ceux-ci, par USB à nouveau, au PC (fig.

4). Une fois le programme ElektorBusBrowser lancé, on paramètre, en haut, le numéro du port COM assigné au convertisseur RS-485/USB avant d'appuyer sur le bouton *Connect*. Si le nœud de capteur est fonctionnel et configuré on devrait maintenant voir s'afficher déjà des valeurs de mesure. Il ne sera cependant possible de renvoyer

Figure 5.
Comme d'habitude, l'interface utilisateur PC repose sur l'HTML. On peut ici modifier l'intervalle de mesure et l'unité des valeurs de température.



L'ordonnanceur appelle successivement la méthode `execute()` de chacune des tâches (si tant est que la tâche n'avait pas demandé de retard, ce qui ne serait pas déjà arrivé à expiration). Chaque tâche s'exécute uniquement pendant une courte période (quelques centaines de cycles d'horloge), de sorte que l'on a vraiment l'impression d'un multitâche réel.

L'exemple est celui du code d'une tâche faisant clignoter une LED :

```
class BlinkTask : public UserTask {
```

```
    enum States {Idle, BlinkingOn, BlinkingOff};
```

```
public:
```

```
    void start() {
        nextState(BlinkingOn);
    }
```

```
    void stop() {
        nextState(Idler);
    }
```

```
private:
```

```
    void turnOn();
```

```
void turnOff();
virtual void execute();
virtual byte getTaskId() { return BLINKTASK_ID; }
};
```

```
void BlinkTask::execute() {
```

```
    switch(state) {
        case BlinkingOn: // Allumer la LED et
                        // attendre 500 ms
```

```
        turnOn();
        nextDelay(BlinkingOff, 500);
        return;
```

```
        case BlinkingOff: // Éteindre la LED et
                        // attendre 500 ms
```

```
        turnOff();
        nextDelay(BlinkingOn, 500);
        return;
```

```
        default: // État invalide, laisser tomber
                panic(1);
```

```
    }
```

```
}
```

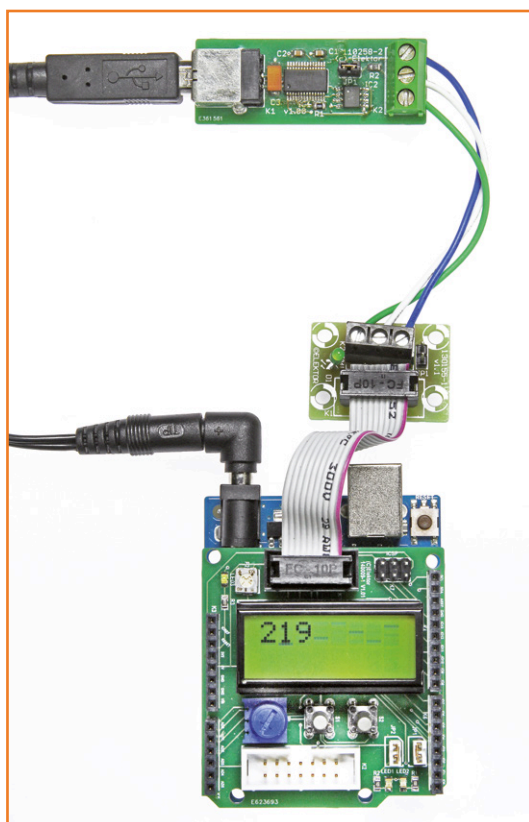


Figure 6.
Simulation faisant appel à un Arduino Uno et à la carte *shield* (extension) d'Elektor. Sur l'ElektorBus, ces noeuds se comportent exactement comme le ferait le module capteur de °T.

des commandes qu'à condition d'avoir coché la case *DirectMode*.

L'interface utilisateur toute simple s'explique d'elle-même (**figure 5**). Le label « °C » et « °F » à côté des valeurs de température ne change qu'après la réception, par le nœud de capteur, d'un message l'informant du changement d'unité. L'application de la nouvelle échelle est automatique. Pour voir s'afficher le code HTML/JavaScript, il suffit d'appuyer sur le bouton *Source*.

Si vous possédez la passerelle AndropOD, vous pouvez également connecter le module de capteur à un téléphone ou une tablette tactiles Android [10]. L'interface HTML/JavaScript fonctionne également telle quelle sous le programme

ElektorBusBrowserForAndropod, que l'on pourra télécharger gratuitement sur Google Play.

Simulation

Lors du développement de la commande du PC, le micrologiciel standard pour le module n'était pas encore disponible ; nous avons simulé un nœud de capteur °T à l'aide d'une carte *Arduino Uno* dans laquelle avait été implanté une extension Elektor [11] et à laquelle avait été connecté un module RS-485 [12] (**fig. 6**). Un potentiomètre permet de modifier les « valeurs de température » qui s'affichent en dixièmes de degré Celsius. Par le biais du Canal 0, l'*Arduino Uno* envoie la valeur simulée au PC, à l'intervalle spécifié, cela va de soi. Il réagit aussi aux commandes de pilotage du PC, exactement comme le fait notre module capteur de °T doté du micrologiciel standard. Nous proposons en outre le code source de l'ATmega328 [8] ; il a été conçu sur la base de l'EFL [7] et utilise la bibliothèque ElektorBus. On trouvera, dans le dossier *Hardware* du projet Atmel-Studio, des fichiers de code pour l'*Arduino Uno*, l'extension Elektor et le module RS-485-ECC. Ceux-ci rendent les couches logicielles de hiérarchie supérieure indépendantes du matériel et sont, à leur tour, indépendants l'un par rapport à l'autre. Nous vous proposerons, dans les prochains numéros, d'autres logiciels pour la nouvelle extension.

Rien ne vous oblige à utiliser le micrologiciel standard et l'ElektorBus : créez vos propres protocoles propriétaires avec votre propre logiciel. K1 accepte d'autres capteurs, des interrupteurs, voire des actionneurs ; la broche de port PB4 peut fournir des signaux de sortie numériques et mesurer des tensions. Si vous développez un projet personnel qui utilise ce module, n'hésitez pas à nous le présenter sur www.elektor-labs.com ou par un message à la rédaction !

(130468 – version française : Guy Raedersdorf)

Liens

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

[2] www.elektor-magazine.fr/110225

[3] www.elektor-magazine.fr/110258

[4] www.elektor-magazine.fr/110727

[5] <http://datasheets.maximintegrated.com/en/ds/DS18S20.pdf>

[6] www.elektor-magazine.fr/120582

[7] www.elektor-magazine.fr/120668

[8] www.elektor-magazine.fr/130468

[9] www.atmel.com/images/atmel-2586-avr-8-bit-microcontroller-attiny25-attiny45-attiny85_datasheet.pdf

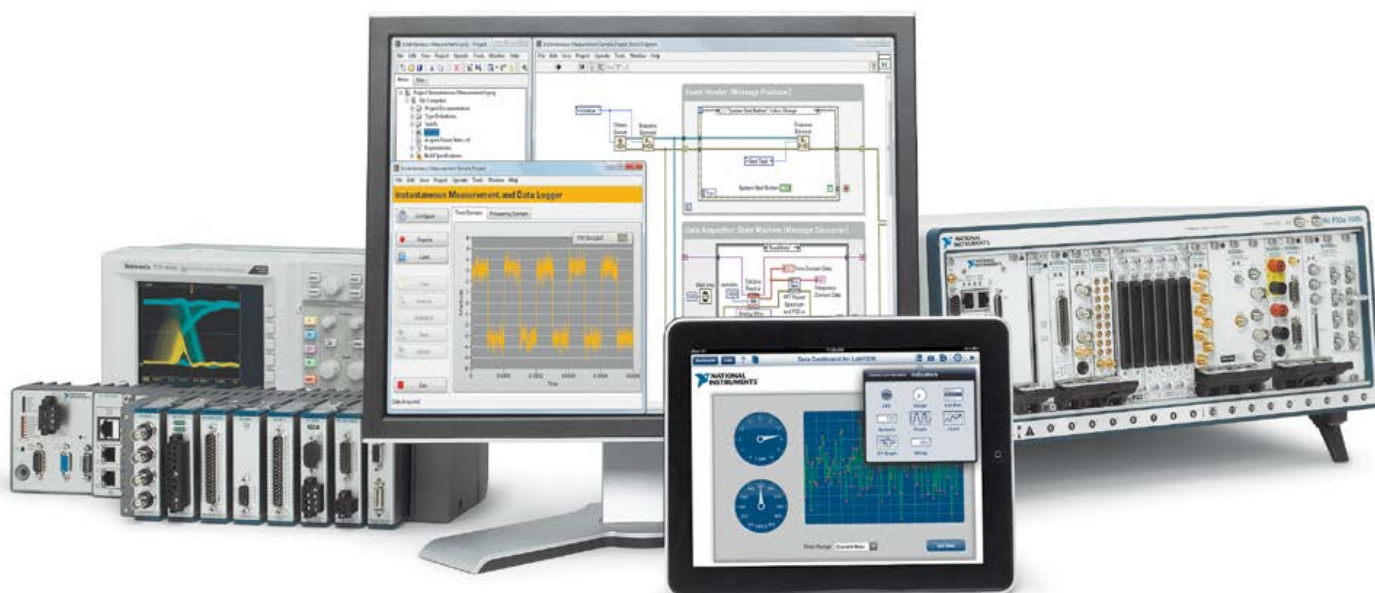
[10] www.elektor-magazine.fr/110405

[11] www.elektor-magazine.fr/140009

[12] www.elektor-magazine.fr/130155

Une infinité d'applications une seule plate-forme

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



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

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



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

01 57 66 24 24

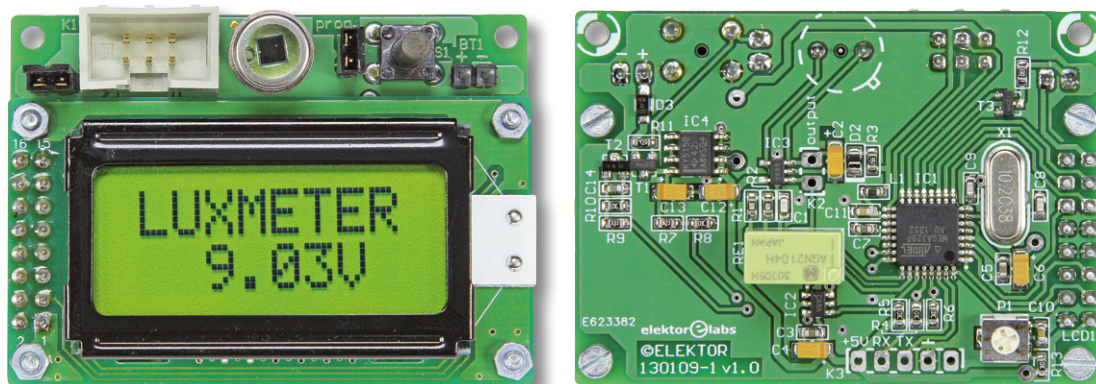
NATIONAL INSTRUMENTS France • 2 rue Hennape - 92735 Nanterre Cedex, France • Tél. : (0)1 57 66 24 24 Fax : (0)1 57 66 24 14 • Société de droit américain - capital social 1.000,00 dollars • US - 11500 N Mopac Expwy, Austin-Texas USA - 10056236 - 344 497 649 • RCS Nanterre - SIRET B 344 497 649 00022 - APE 516J - N.I.I. FR 57344497649

©2014 National Instruments. Tous droits réservés. National Instruments, NI, LabVIEW, et ni.com sont des marques de National Instruments. Les autres noms de produits et de sociétés mentionnés sont les marques ou les noms de leurs propriétaires respectifs. Pour plus d'informations concernant les marques de National Instruments, veuillez vous référer à la partie Terms of Use sur le site ni.com/legal. 18343



luxmètre

de 1 lux à 100 000 lux



Karl-Anton Dichtel
(Allemagne)

Depuis que l'Union européenne a banni presque toutes les lampes à incandescence traditionnelles, il ne reste que les halogènes, les économes en énergie et les LED, dont le rendement est meilleur. Pour savoir de combien, faut-il s'en remettre aux fabricants ? C'est beau la confiance, mais mesurer soi-même, c'est plus beau encore ! Notre luxmètre est là pour ça.

Ce n'est pas à un électronicien qu'il faut apprendre les avantages de l'éclairage à LED sur tout ce qui a existé jusqu'à présent. Et quand on voit les promotions renversantes sur des spots de quelques watts, venus de très loin, qui remplacent les halogènes bien plus gourmands, comment résister ? Des expériences décevantes peuvent faire hésiter sur la qualité, la durabilité tellement vantée et surtout sur les chiffres avancés par les étiquettes. Mes yeux vieillissent-ils si vite ou est-ce cette lumière qui ne me convient pas ?

Pour ne plus devoir gober tout cru les assertions de la publicité, le bon réflexe, c'est d'effectuer des mesures pour connaître la luminosité réelle, de manière objective et fiable. D'ailleurs, un luxmètre ne sert pas seulement à tester des lampes, il vous aide à déterminer si l'éclairage est approprié là où vous travaillez, lisez votre magazine favori ou soudez des CMS. Et pour les lieux de travail, l'Union européenne, entre autres, a publié des directives contraignantes à ce sujet.

Un luxmètre peut aussi servir à d'autres évaluations, notamment avec les lampes aux halogènes

que l'on continue d'utiliser. Même si les vapeurs contenues dans ces ampoules ramènent au filament les atomes de tungstène qu'il a émis, certains d'entre eux atterrissent malgré tout sur l'enveloppe de l'ampoule et lui font perdre sa transparence, donc du pouvoir éclairant. On ne s'en aperçoit pas tout de suite, mais progressivement, la lampe émet moins de lumière mais dégage plus de chaleur. Et que dire des lampes fluocompactes, véritables vaches à lait pour l'industrie, à considérer comme un pis-aller, en attendant mieux et pour suivre les directives européennes. Sans parler des soucis de recyclage qu'elles occasionnent. Tout comme les TL, elles s'opacifient, noircissent et donc s'assombrissent. Comment savoir à quel moment les remplacer ? Ce n'est pas notre œil qui nous renseignera, il faut un expert indépendant, un instrument de mesure *ad hoc*.

Mesurer la lumière

Tout appareil de mesure a recours à un capteur pour traduire une grandeur physique en un courant qui, autant que possible, lui est proportionnel. L'électronique prend le relais. Il existe maintenant des capteurs qui intègrent aussi l'élec-

tronique et fournissent directement une valeur numérique. Cependant, le capteur classique pour la lumière, c'est la photodiode. On en trouve pour la transmission de signaux IR, mais nous avons besoin d'un modèle précis, centré sur le spectre visible et la sensibilité de l'œil humain. Si j'ai choisi la BPW21R de la **figure 1**, c'est parce qu'elle remplit exactement la fonction, présentée sous un boîtier métallique stable avec fenêtre en verre.

Cette photodiode scellée hermétiquement présente une grande surface active de 7,5 mm² et est donc particulièrement sensible. En outre, elle est très linéaire, comme le montrent ses caractéristiques [1]. La figure 2 illustre l'incroyable linéarité de la proportionnalité entre le courant de court-circuit et l'intensité de l'éclairement. La présentation sous forme logarithmique sur les deux axes est nécessaire pour couvrir l'énorme intervalle de luminosité mesurable : plus de sept ordres de grandeur ! Alors qu'il s'agit d'un composant classique, ce capteur de lumière est vraiment idéal pour construire soi-même un luxmètre. Une photodiode fournit un courant proportionnel à l'intensité de l'éclairement exprimée en lux, alors que pour les lampes, on donne généralement le flux lumineux en lumens. Comme 1 lx = 1 lm / m², on doit en principe pouvoir les convertir l'un dans l'autre. Sauf que l'un dépend de la surface irradiée par la source lumineuse et l'autre de la distance et de l'angle de rayonnement. Avec internet, heureusement, il n'est plus nécessaire de chercher un papier et un crayon, sur divers sites, par exemple [2], vous saisissez les valeurs et un calculateur vous donne *illico* le résultat de la conversion, de lux en lumen et réciproquement.

Comment mesurer...

Donc, la photodiode fournit un courant dit de court-circuit. La relation avec l'intensité d'éclairement sera très linéaire si la diode est pour ainsi dire court-circuitée, qu'elle débite le courant sur un circuit d'une très basse résistance d'entrée. Imaginez la tension produite aux bornes d'une résistance de 1 Ω en parallèle sur le capteur éclairé par 1 lx : c'est environ 9 nV. Insignifiant pour l'appliquer au CA/N d'un microcontrôleur. Avec une résolution de 10 bits, un contrôleur AVR typique peut, sous une tension de référence de 1 V, atteindre une résolution d'à peu près 1 mV, avec cinq ordres de grandeur d'écart, on est loin du compte. Intercaler une amplification de tension demanderait un gain de 100 000 000 pour pro-

Caractéristiques

- Luxmètre à six gammes :
1 lx, 10 lx, 100 lx, 1 klx, 10 klx et 100 klx
- Plus petite valeur mesurable : 0,01 lx sur la 1^{re} gamme
- Sélection de gammes automatique
- Extinction automatique après 1 minute
- Sortie des résultats de mesure par interface série
- Composants courants uniquement
- Faible consommation
- Étalonnage par interface série et programme de terminal.

duire 1 V pour 1 lx, autant dire que c'est illusoire. Le circuit de la **figure 3** indique la marche à suivre. Le capteur se trouve entre les deux entrées de l'amplificateur opérationnel IC3 dont l'entrée non inverseuse est polarisée par R3/D2 à une tension auxiliaire d'environ 0,65 V. Quand D1 est éclairée, elle produit un courant qui ferait baisser la tension sur l'entrée inverseuse de IC3... s'il n'y avait pas la rétroaction introduite par R1 ou R2, selon le cas ! En conséquence, la tension de sortie de l'ampli op devient positive au point qu'il circule dans la résistance de rétroaction un courant qui compense complètement celui de D1 et ramène la tension entre les entrées de IC3 à 0 V. Du coup, D1 est pratiquement en court-circuit et travaille en parfaite linéarité. C'est ce qu'on appelle un convertisseur courant-tension. Pour assurer la précision de la mesure, IC3 n'est pas un ampli op standard, c'est un modèle MCP6061T de Microchip caractérisé par des courants d'entrée minuscules dans la gamme des picoampères et une tension de décalage d'entrée d'à peine 150 μV pour n'introduire aucune altération de la mesure. Le signal converti par IC3 est disponible sur K2 pour toute vérification souhaitée.

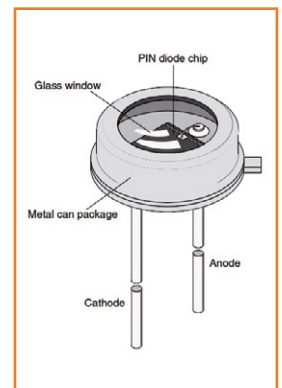


Figure 1.
Dessin de la photodiode BPW21R dans son boîtier.

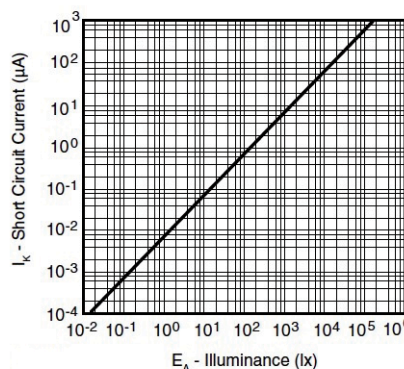


Figure 2.
Relation entre le courant de court-circuit I_k et l'intensité d'éclairement E_A de la BPW21R.

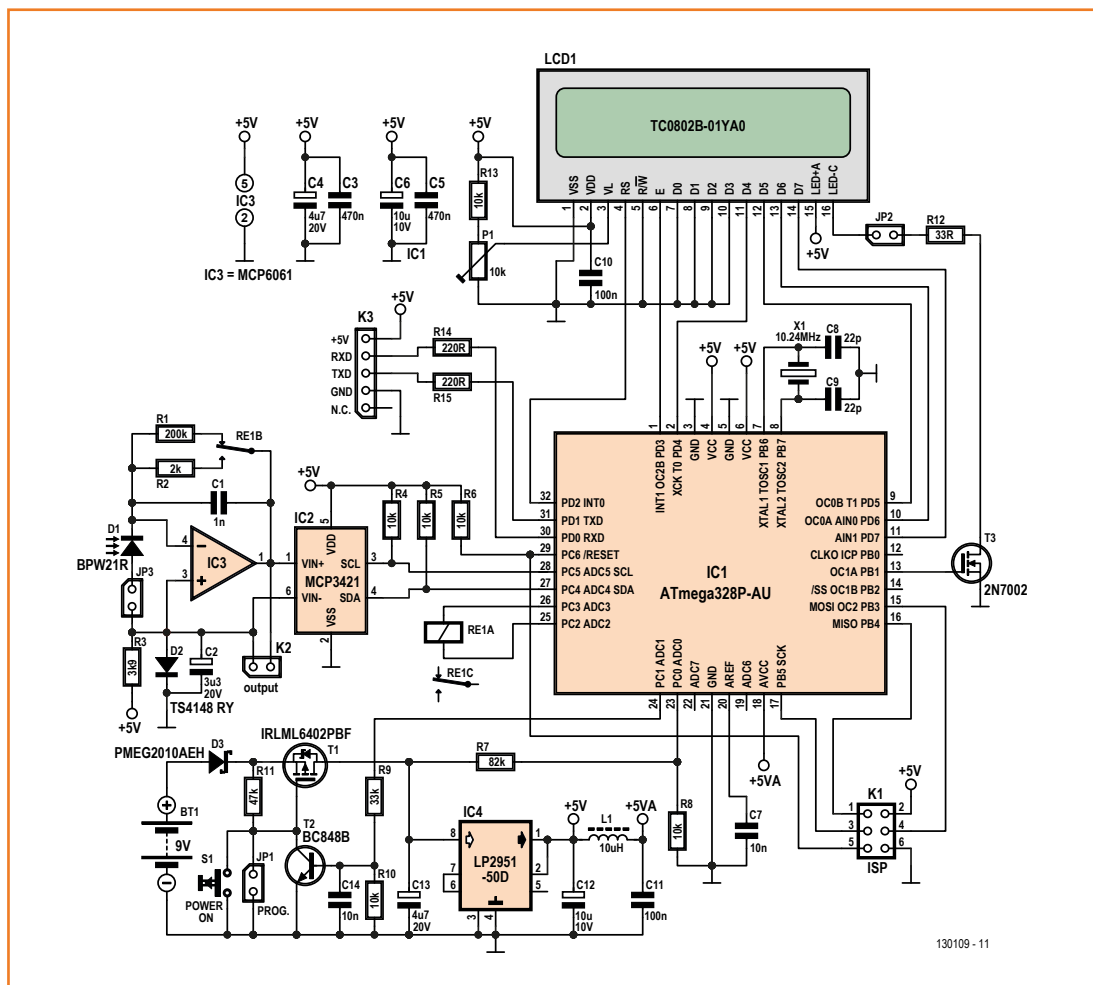


Figure 3.
Le circuit rassemble une photodiode, un convertisseur courant-tension, un CA/N de précision, un microcontrôleur et un écran LCD.

... et avec quoi

Pour 1 lx et avec RE1B en position R1, il y a en sortie de IC3 quelque 1,8 mV, puisque $U = 9 \text{ nA} \times 200 \text{ k}\Omega$. Encore trop peu pour le CA/N d'un microcontrôleur normal. On pourrait remettre un amplificateur – lui aussi adapté aux mesures de précision – mais ne lésinons pas sur les moyens : au lieu d'utiliser le CA/N de l'ATmega328, j'ai ajouté un autre convertisseur, IC2 qui dispose d'un préamplificateur de qualité et d'une résolution de 18 bits. Elle semble exagérée, aussi n'en utiliserons-nous finalement « que » 15 bits, mais nous verrons que cette configuration est fondée.

Le point essentiel de ce projet réside dans l'intervalle de mesure, parce que l'œil humain jouit d'une très grande latitude d'exposition à la lumière, qui s'étend de 1 lx pour admirer la lune jusqu'à 100 klx en plein soleil. Cela représente cinq ordres de grandeur et il faut que l'électro-

nique puisse suivre cette extrême ampleur de dynamique. Voici comment satisfaire à cette exigence. L'inversion entre R1 et R2 change le niveau d'un facteur 100. L'étage suivant doit encore s'en tirer avec une dynamique de 1 à 1 000. Comme l'amplification interne de IC2 peut encore être commutée entre le simple et l'octuple, on obtient dans la dynamique résultante des mesures précises sur toute l'étendue du domaine de mesure. Les détails des commutations entre les différents gains et les gammes de mesure sont repris dans le code du micrologiciel. La précision de l'électronique se situe aux environs de 0,5 %.

IC2 est piloté par le microcontrôleur IC1 sur bus I²C, tant pour la configuration d'IC2 que la lecture des données. IC1 alimente aussi le relais par deux sorties de manière à fournir à sa bobine du courant sous les deux polarités. Une connexion directe est possible du fait que RE1 ne demande

pas plus de 24 mA. C'est un relais bistable qui ne nécessite qu'une brève impulsion dans un sens pour basculer et rester dans cette position sans alimentation. On le ramène dans l'état initial d'une courte impulsion de courant de polarité inverse de la première, ce qui économise encore la batterie. Un contact de relais est ici préférable à un MOSFET.

IC1 a encore trois autres tâches. L'une de ses broches envoie à l'écran LCD des quartets (4 bits) et une autre ligne de port commute T3 lors d'une mesure pour l'éclairage d'arrière-plan. Deux autres lignes s'occupent du transfert de données sérielles à niveau TTL. L'embase à picots T3 est configurée pour accepter la passerelle USB/série BOB d'Elektor [3]. On accède ainsi au système par un port USB via son interface sérielle.

Une fonction auxiliaire de IC1 est d'économiser l'énergie : lors de l'allumage de l'instrument par le bouton-poussoir S1, la broche 24 maintient l'alimentation par T1 et T2 pendant une minute, après quoi l'appareil se déconnecte tout seul.

Une autre économie d'énergie, bien utile sur un instrument de mesure portable, forcément alimenté sur batterie, a trait à l'éclairage de l'écran. C'est T3 qui l'allume, pour autant que le cavalier JP2 soit installé. L'éclairage s'allume à l'enclenchement de l'appareil et s'adapte à l'éclairement environnant s'il se situe sous les 64 lx. Rien que cette action économise quelque 20 mA, ce qui n'est pas négligeable comparé à la consommation normale de 14 mA pour la mesure. En outre, ce n'est pas un dispendieux régulateur ordinaire qui produit du 5 V stabilisé, mais bien IC4, un circuit spécial à basse puissance et faible déchet. Il fonctionne même sous une tension de batterie de 5,5 V, après quoi sa tension de sortie s'effondre. En dessous de 5 V, l'écran est coupé. À la sortie de IC4, L1 sert à filtrer l'alimentation de la partie analogique de IC1.

Construction

L'implantation des composants demande une main sûre, une pointe fine pour la soudure, une pincette, de la pâte à souder ou de la soudure fine et une loupe. Le capteur est un bon vieux composant traversant, mais les résistances et les condensateurs sont au gentil format CMS 0603. J'encourage donc très fort les débutants à chercher la tutelle d'une personne expérimentée. La **figure 4** montre la face supérieure du circuit

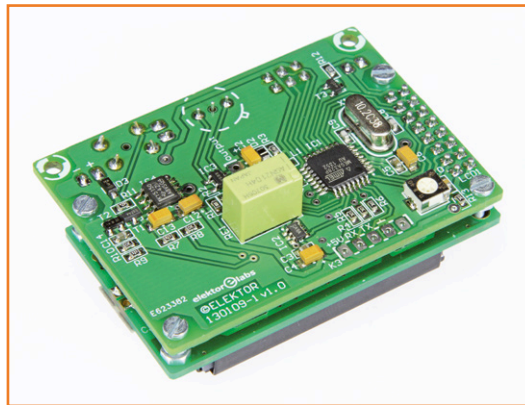


Figure 4.
La face supérieure du circuit imprimé du prototype. On y trouve presque tous les composants, dont les « beaux petits » CMS au format 0603.

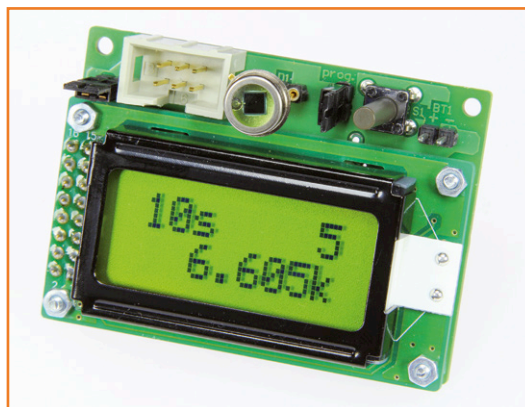


Figure 5.
Le circuit imprimé surmonté du module d'écran. En haut, la photodiode et le bouton-poussoir d'allumage.

imprimé du prototype, elle devrait suffire comme épouvantail ! La loupe vous servira à dépister les courts-circuits entre les broches des puces CMS, vous les éliminerez au moyen de tresse à des-souder et de flux, ça marche encore bien avec le pas de ces puces-ci.

L'écran se place sur l'autre face du circuit imprimé (**fig. 5**). Si vous voulez le garder amovible, installez-le sur des barrettes tulipes mâles et femelles à broches longues. Sinon, utilisez 16 fils de longueur appropriée. L'important, c'est que la distance soit suffisante pour éviter tout contact électrique accidentel entre l'envers de l'écran et le circuit imprimé.

Micrologiciel

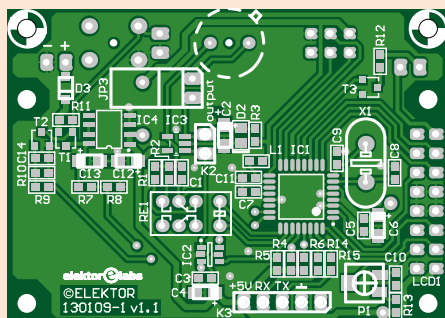
Le logiciel pour IC1 a été rédigé en C avec Studio AVR IDE V4.18, il est disponible gratuitement avec le fichier hexadécimal sur le page de ce projet [4]. De l'horloge à 10,24 MHz, on dérive d'abord des impulsions de 5 ms pour activer le relais et de celles-ci un cycle de 0,5 s pour gérer la séquence. Le code pour piloter l'écran

se trouve dans une contribution du site *mikro-controller.net* [5] abondamment commentée (en allemand). La partie du programme pour l'I²C est un recyclage d'un projet de station météo dans lequel subsiste un reste de code inutilisé jusqu'ici pour un codeur rotatif, entre autres. Pour plus de détails, voyez les commentaires dans le code.

Pour le flashage de IC1, on se sert de K1, l'interface standard ISP à 6 voies. Comme le μ C doit être alimenté aussi durant le chargement, il faut brancher une batterie et installer JP1. On peut aussi maintenir S1 enfoncé pendant le transfert. Le micrologiciel démarre dès la fin du flashage.

Comme tous les octets d'une EEPROM dans un nouveau μ C sont à \$FF et que pour un nouveau flashage il faut les remettre à cette même valeur, le μ C peut déterminer facilement si l'appareil de mesure démarre pour la première fois. S'il trouve la valeur FF_H dans deux octets de l'EEPROM, il écrit provisoirement un facteur de correction de 1 à ces deux adresses. Cela correspond à une conversion de 9 nA en 1 lx. On pourra après coup enregistrer un étalonnage plus précis par l'interface série.

Le pas suivant consiste à mesurer, sans photodiode (après avoir ôté JP3), le décalage d'entrée de l'ampli op et du CA/N, il s'affiche sur l'écran



et est stocké dans l'EEPROM pour effectuer la correction.

À partir de là, l'instrument de mesure affiche toutes les demi-secondes une nouvelle mesure et place toutes les secondes sur l'interface série une nouvelle valeur, suivie d'un point-virgule comme séparateur pour faciliter l'introduction dans une feuille de calcul sur un PC des données transmises. La minute écoulée, le micrologiciel éteint l'appareil de mesure.

Lors du prochain démarrage au moyen du poussoir S1, l'EEPROM ne sera plus remplie de FF_H et l'instrument ira directement à la mesure de l'éclairement. Pour cela, il faut que JP3 soit installé.

Divers

Appuyez sur S1 pour que le logiciel mesure et affiche la tension de batterie. Après une seconde, il passe à la mesure de l'éclairement. La figure 5 détaille l'écran : la ligne du haut indique le temps écoulé et la gamme de mesure. Le luxmètre commute automatiquement entre les six échelles de mesure et le 5 à droite montre qu'il a choisi la gamme de 10 klx, dans laquelle la mesure affichée en bas représente donc 6 605 lux.

Le débit binaire de la transmission par l'interface série est de 38 400 bauds, un bon compromis entre vitesse et sécurité.

Pour savoir ce que vaut ce luxmètre fait maison, nous l'avons comparé à un produit du commerce, le modèle Testo 540. Sans étalonnage, celui-ci n'a donné que de faibles écarts de l'ordre de 3 à 4 %. En lumière faible, on a noté 108 lx au lieu de 105 lx et en lumière forte, 65 klx au lieu de 63 lx. Au bureau, éclairé par tubes fluorescents, les différences étaient plus marquées, 1 115 lx au lieu de 1 600 lx, soit plus de 30 %. Cela peut provenir de la température de couleur, plus chaude, de l'éclairage artificiel ou tenir au fait que l'appareil Testo mesure à travers un dôme opalin, mais on ne sait pas quelle mesure est la meilleure.

En tout cas, on retiendra que ce luxmètre à construire soi-même est un instrument polyvalent qui n'a pas à craindre la comparaison avec les produits commerciaux.

(130109 – version française : Robert Grignard)

L'étalonnage

Pour effectuer l'étalonnage, on a le choix entre une source lumineuse de référence dont on connaît la distance avec précision ou un autre luxmètre.

En prenant le cas le plus défavorable, si le luxmètre affiche 820 lx alors qu'il n'y en a réellement que 410, donc la moitié, le facteur de correction sera de 0,5. Comme ce facteur est à introduire en chaîne ASCII série sur cinq chiffres, on le code « 05000 ». Autre exemple réaliste, le luxmètre affiche

seulement 400 lx au lieu de 410, le facteur vaut $410 / 400 = 1,025$. La chaîne ASCII correspondante sera de « 10250 ».

Quand le luxmètre est allumé et relié par l'interface série au PC sur lequel le programme de terminal égrène de nouvelles valeurs chaque seconde, que se passe-t-il ? Le micrologiciel attend en fait la saisie d'un chiffre. Si c'est un 1, il passe en mode d'introduction du facteur de correction. Pour cela, l'écran du PC affiche d'abord :

```
Mode 1: Calibration - Input 5 digits or ESC with #
actual calibration factor = 10000
Input Calibration:
```

Vous aviez calculé la chaîne à fournir pour le nouveau facteur de correction. Après l'introduction des cinq signes, le cycle de mesures normales reprend. La copie d'écran représente le dialogue en HTerm. Pour que le nouveau facteur soit pris en compte, il faut mettre à zéro le contrôleur, ce que l'on peut faire en réunissant brièvement les broches 5 et 6 de K1. Le facteur reste en vigueur jusqu'à ce qu'un nouveau soit enregistré. On peut relancer la procédure quand on veut.

Time(s):	Range:	Unit:
1:12.00	10	LUX
2:12.00	10	LUX
3:12.00	10	LUX
4:12.00	10	LUX
5:12.00	10	LUX
6:12.00	10	LUX
7:12.00	10	LUX
8:12.00	10	LUX
9:12.00	10	LUX
10:12.00	10	LUX
11:12.00	10	LUX
12:12.00	10	LUX
13:12.00	10	LUX
14:12.00	10	LUX
15:12.00	10	LUX
16:12.00	10	LUX
17:12.00	10	LUX
18:12.00	10	LUX
19:12.00	10	LUX
20:12.00	10	LUX
21:12.00	10	LUX
22:12.00	10	LUX
23:12.00	10	LUX
24:12.00	10	LUX
25:12.00	10	LUX
26:12.00	10	LUX
27:12.00	10	LUX
28:12.00	10	LUX
29:12.00	10	LUX
30:12.00	10	LUX
31:12.00	10	LUX
32:12.00	10	LUX
33:12.00	10	LUX
34:12.00	10	LUX
35:12.00	10	LUX
36:12.00	10	LUX

Liens

- [1] caractéristiques BPW21R : www.vishay.com/docs/81519/bpw21r.pdf
- [2] www.lumitronix.fr/Lumen-zu-Lux/
- [3] passerelle USB : www.elektor.fr/ft232r-usb-serial-bridge-bob-110553-91
- [4] www.elektor-magazine.fr/130109
- [5] www.mikrocontroller.net/articles/AVR-GCC-Tutorial/LCD-Ansteuerung

gradateur PWM pour baladeuse à LED

Pascal Rondane
(Tours)

Ce montage a été conçu pour une lampe à LED utilisée lors de randonnées pédestres ; en effet, quand on marche dans l'obscurité, une fois les pupilles dilatées, l'éclairage normal de ces torches est trop puissant et devient gênant. Ce n'est pas la seule application où le confort d'une lampe à LED gagne à voir son vif éclat réduit.

pour éviter
l'éblouissement

Comme son nom l'indique, la fonction principale de ce module est de réduire progressivement la luminosité selon les besoins du moment. Une fonction accessoire de clignotant à pleine puissance est bien utile, par exemple quand des piétons noctambules voient arriver un véhicule et souhaitent signaler leur présence au conducteur. Le circuit à microcontrôleur assure deux fonctions en parallèle : d'une part, la commande de luminosité par MLI et le choix du mode d'éclairage

(fixe avec intensité réduite, ou clignotant à pleine puissance) et, d'autre part, la surveillance de l'état de charge de la batterie et son affichage. La modulation de largeur d'impulsion utilisée ici pour obtenir une gradation de la luminosité convient pour les baladeuses à LED, mais plus généralement pour tout éclairage à LED alimenté par batterie. Le gradateur a été conçu pour une tension de batterie de 7,2 V, mais le régulateur accepte jusqu'à 14 ou 15 V.

Caractéristiques

Tension d'alimentation : 5 V à 15 V
Consommation : 3,9 mA (sans LED D2)
Courant de sortie : 1 A
Seuils de coupure (paramétrables)
- batterie chargée : 7 V
- batterie faible : 6 V
- batterie déchargée (coupure) 5,4 V
dimensions : 60 X 30 mm

Le µC sait tout faire

La commande de l'électronique du gradateur (**fig. 1**) est assurée par un microcontrôleur à 8 broches ATtiny45 programmé *in situ* (IC1). L'intensité de la luminosité des LED de la torche — représentées à droite sur le schéma, dans l'encadré en pointillés verts — est commandée par l'utilisateur au moyen d'un inverseur à contact fugitif à deux positions (S2). Comme, le nombre d'entrées/sorties d'un ATtiny45 est réduit, l'interrogation des deux organes de commande du

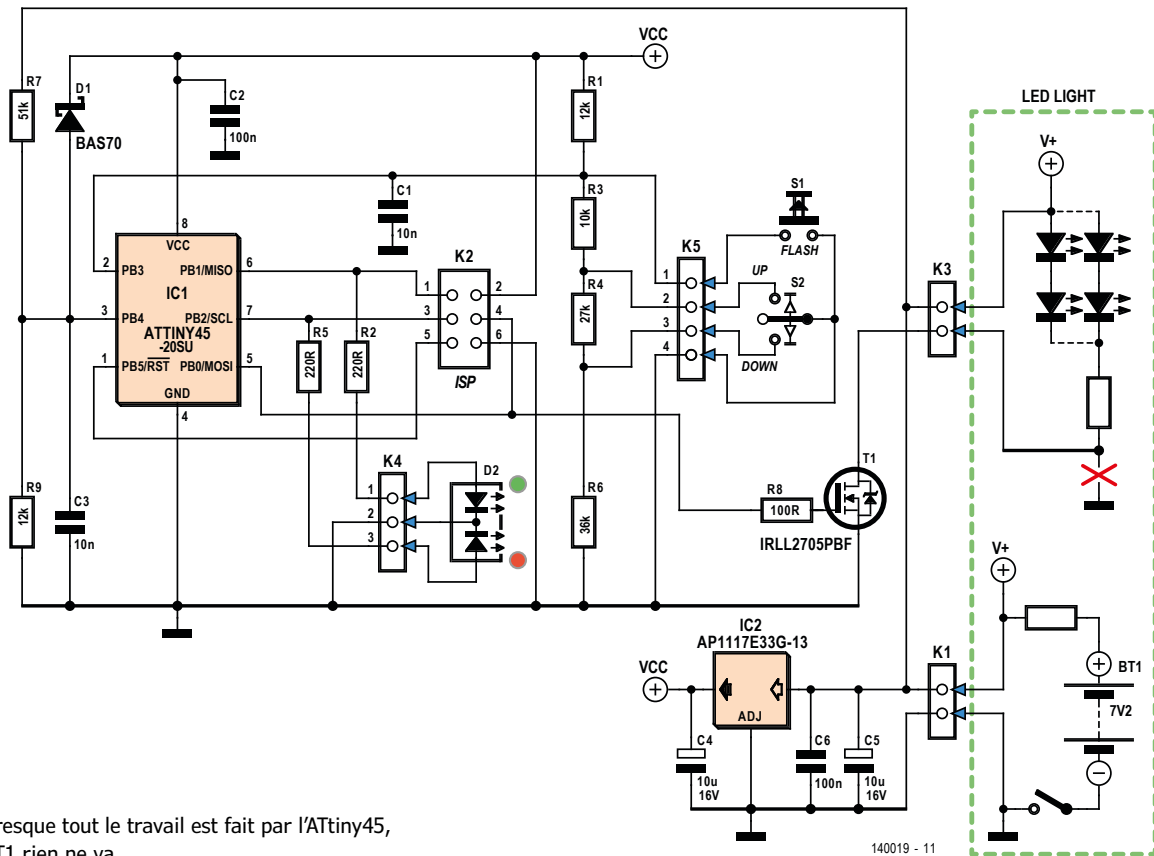


Figure 1. Presque tout le travail est fait par l'Attiny45, mais sans T1 rien ne va.

gradateur est analogique. L'entrée de conversion analogique-numérique de la broche 2 du μ C est en effet reliée à un diviseur de tension formé par R1, R3, R4 et R6, lui-même relié par K5 à l'inverseur S2 et au bouton poussoir S1 : ainsi le μ C détermine l'état de S1 et S2 en fonction de la tension qui règne sur la broche 1 de K5. Quand ils sont actionnés par l'utilisateur, S1 ou S2 court-circuitent les résistances de la moitié inférieure du diviseur de tension. En fonction de la tension mesurée, le μ C effectue le réglage de luminosité souhaité : augmentation ou diminution de la luminosité (S2) ou pleine puissance clignotante (S1). Les seuils de tension correspondants sont :

- 2,8 V : S1 et S2 ouverts
- 2,5 V : augmentation (UP)
- 1,5 V : diminution (DOWN)
- 0 V : flash

La modulation de largeur d'impulsion (PWM) est aussi une affaire de logiciel. La sortie de modula-

tion de largeur d'impulsion (broche 5) du micro-contrôleur commande l'alimentation de la série de LED de la baladeuse par l'intermédiaire du transistor de puissance MOS T1.

Au démarrage, pour y voir clair sans être ébloui, la lampe s'allume toujours à mi-puissance. Avec le diviseur de tension que forment R7 et R9 connectées entre la tension d'alimentation et l'entrée du convertisseur A/N du μ C (broche 3), ce dernier surveille en permanence le niveau de tension de la batterie de la lampe et affiche avec

Les connecteurs

- K1 : alimentation
- K2 : programmation in situ du μ C
- K3 : liaison de masse des LED de la torche (modifiée)
- K4 : indicateur d'état de charge de la batterie
- K5 : S1 et S2



Figure 2.

Le circuit imprimé n'est pas trop serré pour un soudage manuel des CMS et juste assez petit pour une baladeuse cylindrique.

Liste des composants

Résistances (SMD1206, 1%, ¼ W)

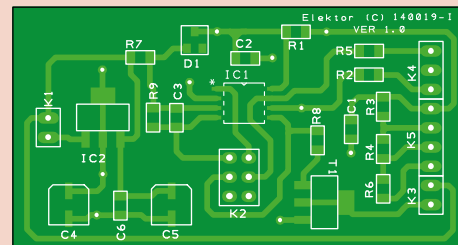
R1, R9 = 12 kΩ
 R2, R5 = 220 Ω
 R3 = 10 kΩ
 R4 = 27 kΩ
 R6 = 36 kΩ
 R7 = 51 kΩ
 R8 = 100 Ω

Condensateurs

C1 = 10 nF* (SMD1206)
 C2, C6 = 100 nF (SMD1206)
 C3 = 10 nF (SMD1206)
 C4, C5 = 10 µF, 63 V rad. (SMD1206)

Semi-conducteurs :

IC1 = ATtiny45-20SU (SOIC8) programmé (140019-41)
 IC2 = AP1117E33G-13 (SOT-223)
 D1 = BAS70* (SOT-23)
 T1 = IRL2705PBF (SOT-223)



Divers :

K1, K3 = embase mâle soudée à 2 voies
 K2 = embase mâle à 2 x 3 voies
 K4 = embase mâle soudée à 3 voies
 K5 = embase mâle soudée à 4 voies
 S1 = bouton poussoir
 S2 = inverseur on-off-on à contact fugitif
 circuit imprimé 140019-1

* cf texte

```
#####
'# Battery voltage test (Voltage hysteresis = 0V)
#####

Select Case Etat_led 'Voltage hysteresis = 0V

Case 1 : 'Operating voltage => OK

    Tension_batt = Tension_batt_basse - Tension_hysteresis '6V-0V = 6.0V
    If Adcval < Tension_batt Then 'ADC measurement and switching to low battery mode
        Etat_led = 2
    End If

Case 2 : 'Low battery

    Tension_batt = Tension_batt_dechargee - Tension_hysteresis '5,5V-0V = 5.5V
    If Adcval < Tension_batt Then 'ADC measurement and go to discharged battery
        ' mode - cut the load
        Etat_led = 3
    End If
    Tension_batt = Tension_batt_chargee + Tension_hysteresis
    If Adcval >= Tension_batt Then
        Etat_led = 1
    End If

Case 3 : 'If low battery/ cutoff LED
    Tension_batt = Tension_batt_chargee + Tension_hysteresis '7,2V + 0V = 7.2V
    If Adcval >= Tension_batt Then 'ADC measurement and return
        ' "OK mode "if battery is charged
        Etat_led = 1
    End If

End Select
```

Extrait du code source.
 Cette partie du programme
 surveille la tension
 d'alimentation.

D2 l'état de charge de la manière suivante : **vert** = charge suffisante, **orange** = charge faible, **rouge** = décharge.

Dans ce dernier cas, le microcontrôleur prend aussi l'initiative d'éteindre toutes les LED de la lampe ; considérant que la batterie est à plat, il lui épargne ainsi une décharge profonde. Il les rallume automatiquement dès que la tension d'alimentation remonte.

Pour une tension d'alimentation de 8 V, on relève 1,5 V sur la broche 3 d'IC1 et D2 est verte. Pour 6 V de tension d'alimentation, on y relève 1,16 V et D2 est orange. Pour 5,4 V de tension d'alimentation, on ne relève plus que 1,0 V, la LED est rouge, et la baladeuse est éteinte.

Lors de la conception d'un tel circuit, on s'efforce d'en réduire la consommation au strict minimum, car ce serait un comble de dissiper l'énergie alors qu'on essaye de réduire l'éblouissement ! C'est pourquoi l'indicateur tricolore clignote quel que soit d'état de la charge.

La diode D1, une Schottky BAS70 ou équivalente, protège l'entrée PB4.

Les rebonds de l'inverseur S2 sont gommés par C1 (10 nF). Si vous tombez sur un modèle parkinsonien, vous remarquerez des irrégularités dans la gradation. Avant de changer d'interrupteur, essayez d'augmenter la valeur de C1 (100 nF), cela pourrait suffire à éliminer cet inconvénient. Le soin de réguler la tension d'alimentation à 3,3 V est laissé à un ASM117.

Entre le microcontrôleur alimenté sous 3,3 V et les LED généralement alimentées sous 7,2 V, l'intermédiaire est le transistor MOSFET de puissance qui assure la commutation de puissance.

Logiciel

Le programme [1] a été écrit avec Bascom-AVR, ce qui permet de le modifier très facilement, même avec la version de démonstration. Après la déclaration des variables pour les seuils de coupure et de réenclenchement des seuils d'alarme de la batterie, on paramètre Timer0 pour la gestion de la MLI et Timer1 pour la gestion du mode de clignotement de la LED d'état de la batterie. Le programme passe l'essentiel de son temps à attendre une action sur ses entrées. Plus loin, on trouve les routines de gestion de la LED de signalisation en fonction de la tension de la batterie rechargeable, et la gestion des seuils du convertisseur AN pour la lecture de la position des interrupteurs.

En mode 'flash', c'est-à-dire quand S1 est appuyé, les LED sont pilotées au maximum de la largeur

d'impulsion ; une brève temporisation filtre les inévitables rebonds des contacts lorsque l'on appuie sur ce bouton, évitant ainsi les déclenchements importuns.

Les commentaires du programme vous guideront si vous souhaitez modifier le programme. Par exemple pour accélérer l'incrémentement et la décrémentation, il suffit de réduire la valeur de la variable Pas, ou pour définir une valeur de MLI par défaut autre que celle que j'ai prévue... On pourra aussi jouer sur la durée de la temporisation anti-rebonds de S1 en diminuant la valeur de Tempo_bp_on_value pour réduire le temps de réaction de ce poussoir. Une dernière idée de modification serait de choisir une autre valeur par défaut pour la MLI au démarrage (Pwm = 100).

Bricolage

Le dessin de circuit imprimé apparaît sur la **figure 2**. Le faible encombrement obtenu par la mise en oeuvre de composants montés en surface permet d'incorporer facilement la carte à une lampe baladeuse cylindrique. Nous avons cependant éprouvé des difficultés inattendues avec certains modèles de ces baladeuses, si solidement assemblés par collage qu'il a été impossible d'en déboîter le tube transparent sans l'endommager.

Une fois la lampe démontée, la seule modification électrique à y apporter est d'interrompre la liaison de masse commune à la cascade de LED (souvent 2 x 30). C'est le **X** rouge sur le schéma de la figure 1. Il suffit ensuite de souder un fil qui relie ce point commun de la série de LED (ou des deux séries, le cas échéant) à la borne de K3 elle-même reliée à T1. Deux autres fils relieront à K1 l'un la masse de la lampe et l'autre le plus de la batterie. Il ne reste plus qu'à percer deux trous dans le corps de la lampe pour y monter S1 et S2.

(140019)

Liens

[1] www.elektor-magazine.fr/140019

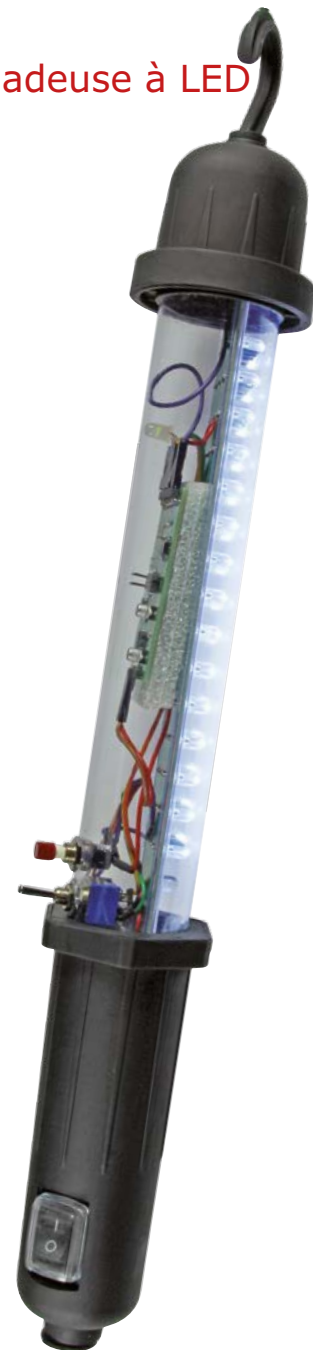
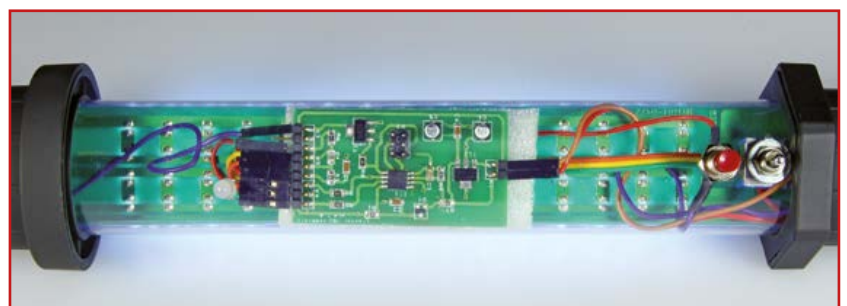


Figure 3. Avec un peu de chance, vous tomberez sur un modèle de lampe facile à démonter et à remonter après y avoir inséré votre gradateur.



horloge Nixie à récepteur GPS

Le mariage (au 3^e top et au 7^e ciel) du tube et du satellite

Willem Tak
(Pays-Bas)



La lumière chaude émise par les tubes Nixie, avec leur astucieux grillage ardent, ajoute une touche rassurante aux projets qui emploient ces composants d'un autre âge. Leur facture artisanale est combinée ici avec la technique moderne de la communication par satellite, pour donner l'heure exacte fournie par un module GPS à antenne intégrée. Un projet pas comme les autres !

Comme l'indique mon nom, j'ai une certaine prédisposition pour l'horlogerie. C'est même un tic chez moi. Je suis fasciné par les horloges à tubes Nixie, mais je déteste les boutons de mise à l'heure. La solution du récepteur horaire DCF77 ne me tentait pas non plus, on en a assez vu. C'est donc d'un module GPS que mon horloge

Nixie à 4 chiffres (heures et minutes) reçoit son signal horaire. Elle se passe ainsi de tout réglage, et peut être intégrée dans un coffret compact d'où n'émergeront que les tubes.

Le module GPS que j'avais retenu est fabriqué par *GlobalSat* (EM-406 ou EM-411). Le labora-

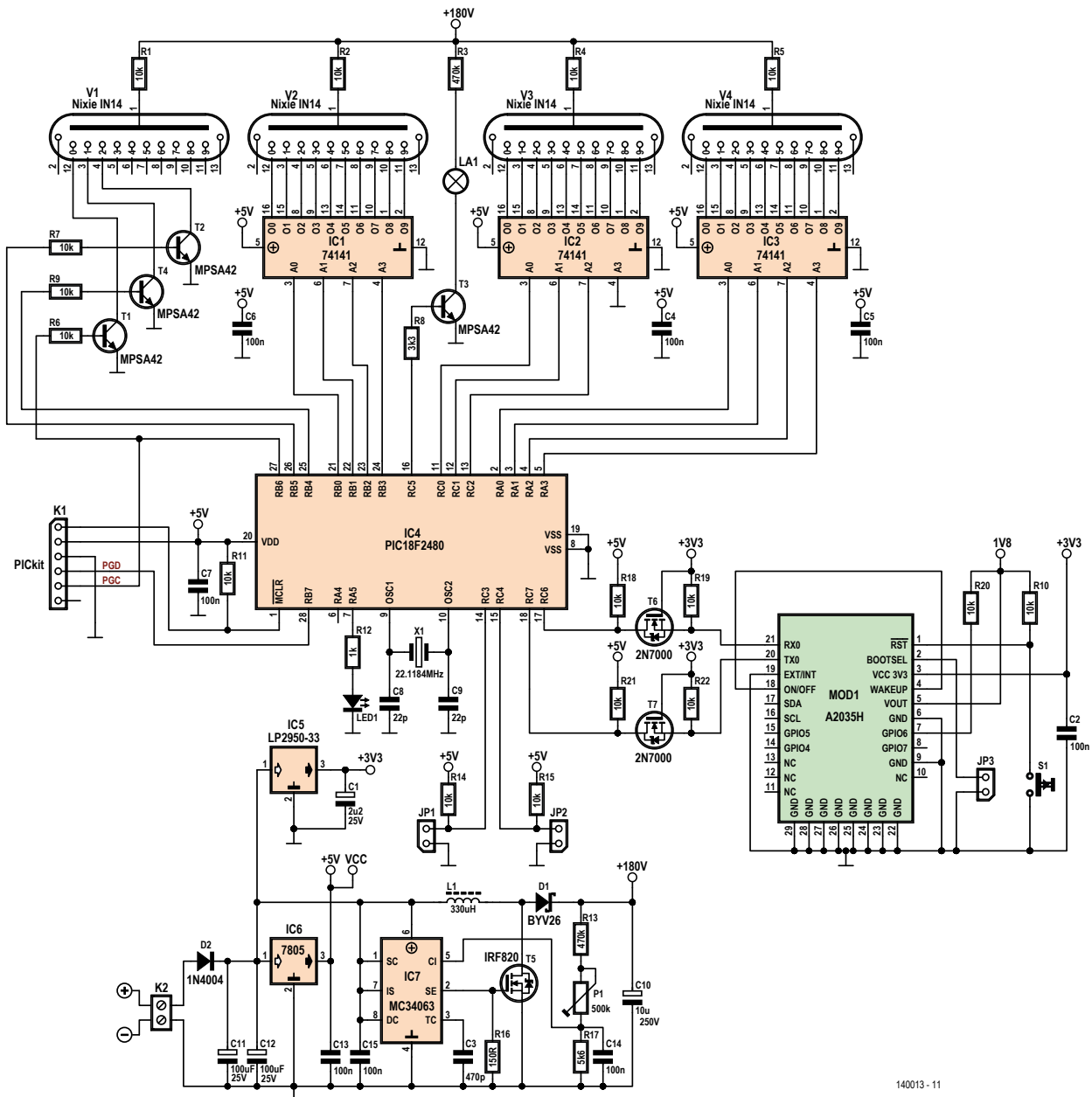
toire d'Elektor a opté, en raison de sa meilleure disponibilité, pour un module A2035H de *Maestro* (également utilisé sur l'extension pour la carte FPGA d'Elektor).

Les modules *Globa/Sat* sont utilisables sans modification du progiciel, mais non sans quelques modifications à faire à la main sur la carte. La sensibilité des modules des deux marques est suffisante pour une réception à l'intérieur.

Le matériel

Les vedettes de ce circuit restent les tubes Nixie (**fig. 1**), déjà utilisés dans divers projets (entre autres [1] et [2]). Nous ne nous étendrons donc pas ici. Retenez que sur ces tubes à cathode froide, chaque chiffre a sa propre cathode. Soumis à une tension suffisante (180 V), un chiffre s'auréole d'une aura lumineuse. Le potentiel requis est obtenu avec un convertisseur élévateur (*step-up*, IC7) MC34063, une diode très rapide

Figure 1.
Schéma de l'horloge Nixie à 4 chiffres et à récepteur GPS.



140013 - 11

(D1) et une bobine (L1) ; l'énergie des pics de tension est stockée dans un condensateur haute tension (C10).

Quand votre circuit sera prêt à fonctionner, il faudra d'ailleurs vérifier avec soin la présence d'une boucle de réaction vers la broche 5 du circuit intégré ; si elle est absente, la tension d'alimentation peut augmenter démesurément, avec risque d'explosion du condensateur électrochimique. À travers les résistances R1, R2, R4 et R5, la tension de 180 V arrive aux anodes des tubes Nixie. En raison de la variété des modèles dis-

ponible au néon, pilotée par transistor elle aussi. Si le montage fonctionne correctement, cette luciole, clignotera une fois par seconde ou restera allumée en permanence (selon la position de J2). La lecture du module GPS et la commande des pilotes de Nixie/transistors sont confiées à un PIC de type 18F2480 (IC4), cadencé à 22,1184 MHz afin de disposer du facteur de division adéquat pour la sortie à 4800 Bd de l'unité de GPS (MOD1). Le PIC utilise, en option, une LED de service (LED1) qui clignote brièvement après une réinitialisation, et signale ensuite chaque chaîne de GPS correcte. Si elle reste allumée, c'est qu'il y a un problème.



ponibles, il faudra, avant de se lancer dans la réalisation, déterminer le courant idéal pour les tubes à utiliser. Dans la plupart des cas, la valeur de résistance est de l'ordre de 10 à 15 k Ω . Ici, 12 k Ω pour le prototype d'Elektor, avec des IN-16, d'origine russe, abordables et raisonnablement disponibles.

Les circuits de commande des Nixie sont, sans surprise, des circuits 74141 conçus pour ça, mais malheureusement difficiles à trouver. Alternative : le K155ID1 de source russe. La commande des dizaines d'heures (V1, à gauche, vu de face) n'utilise pas de circuit intégré mais trois transistors (pour les chiffres « 0 », « 1 » et « 2 »). Il se trouve, entre les heures et les minutes, une

La programmation *in situ* du PIC est possible par K1, auquel on pourra raccorder un programmeur PICKit 2.

Le cavalier JP1 détermine l'affichage ou non du zéro non significatif (ouvert = pas de zéro, fermé = allumage du zéro). JP2 détermine le clignotement ou non du point décimal (ouvert = fixe, fermé = clignotement).

L'antenne de réception des signaux GPS est intégrée dans le module GPS (A2035H) qui fonctionne en autarcie. S1 et JP3 ne sont nécessaires qu'en cas de mise à jour du progiciel de l'A2035. Comme la tension d'alimentation du module GPS est de 3,3 V et celle du PIC de 5 V, j'ai inséré dans les lignes de données deux adaptateurs de niveau simples, constitués d'un FET (T6 et T7) et de résistances (R18/R19 et R21/R22).

Un 7805 (IC6) fournit la tension d'alimentation de 5 V du PIC et des circuits intégrés de commande (*driver*), un LP2950-33 celle de 3,3 V pour le module GPS ; l'ensemble du montage peut être alimenté par un adaptateur secteur (9 à 15 V_{CC}).

Le logiciel

Les codes hexa et source du μ C de cette horloge peuvent être téléchargés gratuitement. Elektor propose aussi le contrôleur programmé (140013-41) [3].

Sous la forme présentée ici, mon horloge n'a que quatre chiffres, mais la structure du programme lui permettrait d'afficher les secondes sur une version à 6 chiffres. J'en décrirai donc le logiciel complet.

L'acquisition des données GPS est classique : le module GPS fournit la chaîne GPS GPRMC. Lors de l'initialisation, il est défini que seule cette chaîne doit être envoyée une fois par seconde ; toutes

les autres chaînes GPS sont inhibées (*disable*). Au moyen d'une routine qui vérifie continuellement la présence d'un caractère dans le tampon RS232, le PIC guette l'apparition de données asynchrones provenant du module GPS. Si, après la mise en route, le flux de données RS232 ne démarre pas, le chien de garde interne le constate et lance une réinitialisation (*reset*) après 1 minute environ. Pendant ce temps, la LED de service LED1 reste allumée. Dès la réception d'une chaîne valide, la surveillance de la ligne de données RS-232 se poursuit, mais de façon différente.

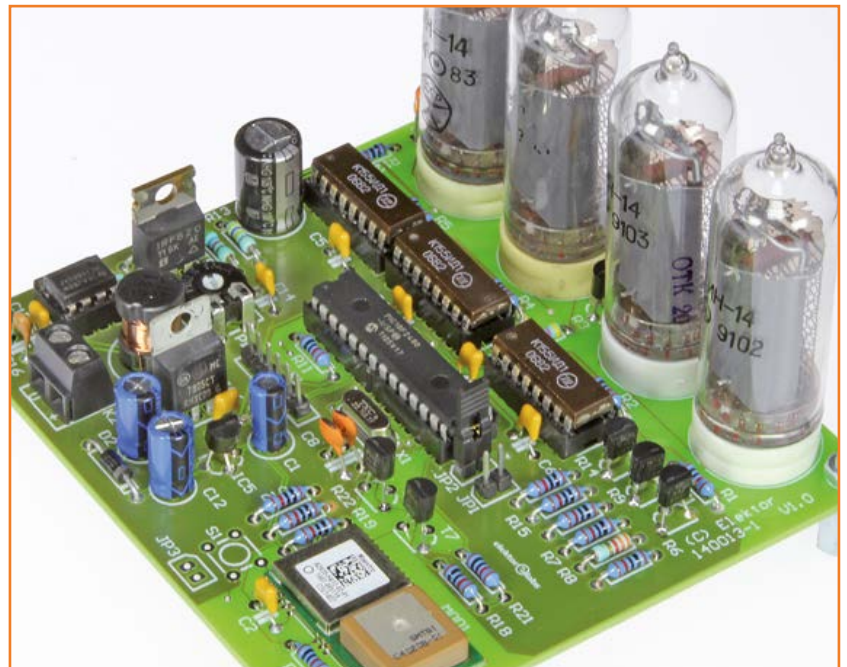
À l'arrivée des données, le programme vérifie s'il s'agit d'une chaîne GPRMC (ce qui devrait toujours être le cas), et si oui, attend la saisie de l'ensemble de la chaîne. Les caractères reçus sont enregistrés jusqu'à la réception d'un retour à la ligne (CR). Le programme calcule ensuite la somme de vérification de la chaîne de caractères reçue, et la compare à la somme de vérification transmise. Si elles sont identiques, les données sont considérées comme valides.

Puis, il calcule le décalage (1 h ou 2 h) à appliquer pour obtenir l'heure d'été/d'hiver correcte. En effet, comme GPS ne possède pas d'indicateur pour ça, le calcul détermine de quelle heure de l'année il s'agit, puis, en se servant de tableaux, si ladite heure est soumise au régime d'été ou d'hiver. Les tableaux sont valables jusque fin 2020. Pour finir, l'heure fournie par le module GPS sous forme ASCII est convertie en hexadécimal, en variables distinctes pour les heures, les minutes et les secondes. Notons qu'il y a déjà eu, auparavant, une conversion d'ASCII vers BCD pour assurer la compatibilité du logiciel avec ses versions antérieures.

En principe, nous devrions maintenant disposer d'une heure GPS valide, mais il reste à résoudre un problème qui peut se poser en cas d'affichage sur 6 chiffres, avec les secondes. La chaîne GPS est sensible aux conditions de réception ; elle passe mieux à un endroit qu'à un autre. Il est vrai que l'intelligence embarquée et la sensibilité du module sont excellentes, mais il arrive parfois une chaîne bancal. Avec l'affichage heures/minutes (comme ici), on ne s'en rendra que rarement compte, mais en cas d'affichage des secondes, une chaîne invalide peut perturber l'affichage, qui sautera une seconde, voire plus. Pour résoudre ce problème, j'ai choisi d'afficher non pas directement la chaîne GPS, mais de créer, avec un

temporisateur, une horloge locale en temps réel : une boucle d'interruption est appelée toutes les 50 ms exactement, grâce à un quartz. L'impulsion correspondante apparaît sur la broche 6 du PIC (RA4). Dans le logiciel, les variables PIC_HR et GPS_HR, représentent l'heure du PIC et l'heure dérivée de la chaîne GPS.

L'horloge interne démarre à 00:00 (.00) ; la mise à l'heure requiert donc forcément une chaîne GPS valide. Le logiciel attend d'en avoir reçu une, ce qui peut parfois durer. Une fois une telle chaîne (valide) reçue, le PIC est synchronisé et l'hor-



loge devient fonctionnelle, pilotée désormais par la boucle d'interruption. Ensuite, le programme démarre un double mécanisme de contrôle.

Pour la fiabilité, il vérifie constamment la stabilité de la réception GPS. Chaque chaîne de GPS est analysée et, si elle est valide, stockée dans un tampon qui retient jusqu'à 10 heures, codées par trois octets chacune, qui sont la représentation binaire du nombre de secondes écoulées depuis minuit (soit $24 \times 60 \times 60$ au maximum, malheureusement > 65536).

À la réception d'une heure, le contenu du tampon se décale d'une position, l'heure la plus ancienne est supprimée, l'heure la plus récente est mise

en position 10. Si le flux de données GPS est stable, la différence dans le tampon entre l'heure la plus ancienne et la plus récente sera toujours de 9 secondes exactement. Si c'est le cas, l'indicateur GPS_STABLE est activé. Le signal GPS est utilisable pour synchroniser l'heure du PIC.

Cette synchronisation est effectuée au moins une fois par heure. Lorsque l'heure du PIC est à xx.00.30, la routine d'interruption requiert une demande de synchronisation. En présence d'un signal GPS stable, l'heure du PIC est remise à l'heure GPS. L'instant de synchronisation de xx.00.30 a été choisi pour qu'une correction éven-

tuelle ne concerne que les secondes, et pas ni les heures ni les minutes.

Sur les différentes horloges que j'ai construites, l'écart ne dépasse jamais 1 s à condition que la boucle d'interruption dure précisément 50 ms. Pourtant, l'expérience montre que cette synchronisation horaire n'est pas toujours suffisante ; il arrive régulièrement que l'unité GPS fournisse en démarrant une heure erronée. En effet, une chaîne apparemment valide apparaît (souvent dépourvue d'ailleurs de coordonnées complètes), dont l'heure est en fait fausse. Or le module la conserve souvent relativement longtemps. C'est d'autant plus ennuyeux que l'heure fausse, trans-

Liste des composants

Résistances :

R1,R2,R4,R5,R6,R7,R9,R10,R11,R14,R15,R18,R19,R20,R21,R22 = 10 k Ω
R3,R13 = 470 k Ω
R8 = 3k3
R12 = 1 k Ω
R16 = 150 Ω
R17 = 5k6
P1 = aj. hor. 500 k Ω

Condensateurs :

C1 = 2 μ 2/50 V, pas de 2 mm
C2,C4,C5,C6,C7,C13,C14,C15 = 100 n
C3 = 470 p, Y5P, pas de 2,5 mm
C8,C9 = 22 p, COG/NP0, pas de 2,5 mm
C10 = 10 μ /250 V, radial, pas de 5 mm (Panasonic ECA2EHG100)
C11,C12 = 100 μ /50 V, radial, pas de 3,5 mm

Bobines :

L1 = 330 μ H/900 mA, radiale (diamètre 10 mm, hauteur 15 mm)

Semi-conducteurs :

D1 = BYV26 (diode ultra rapide, 600 V/1 A)
D2 = 1N4007
LED1 = LED rouge, 3 mm
T1,T2,T3,T4 = MPSA42 (300 V/ 500 mA)
T5 = IRF820 (N-MOSFET, 500 V/2,5 A)
T6,T7 = 2N7000 (N-MOSFET, 60 V/200 mA)
IC1,IC2,IC3 = K1551D1 (74141)
IC4 = PIC18F2480-I/SP (programmé - 140013-41)
IC5 = LP2950-33LPE3
IC6 = MC7805
IC7 = MC34063

Divers :

V1,V2,V3,V4 = tube Nixie IN-14
LA1 = luciole au néon avec fils de connexion
X1 = quartz 22,1184 MHz
MOD1 = A2035H, module GPS à antenne interne (Maestro Wireless Solutions)
JP1,JP2,JP3 = embase à 2 broches + cavalier (JP3 optionnel)
K1 = embase à 6 broches, au pas de 2,54 mm
K2 = bornier encartable à 2 contacts, au pas de 5,08 mm
S1 = bouton-poussoir pour montage sur CI, 6x6 mm (optionnel)
circuit imprimé 140013-1 [1]

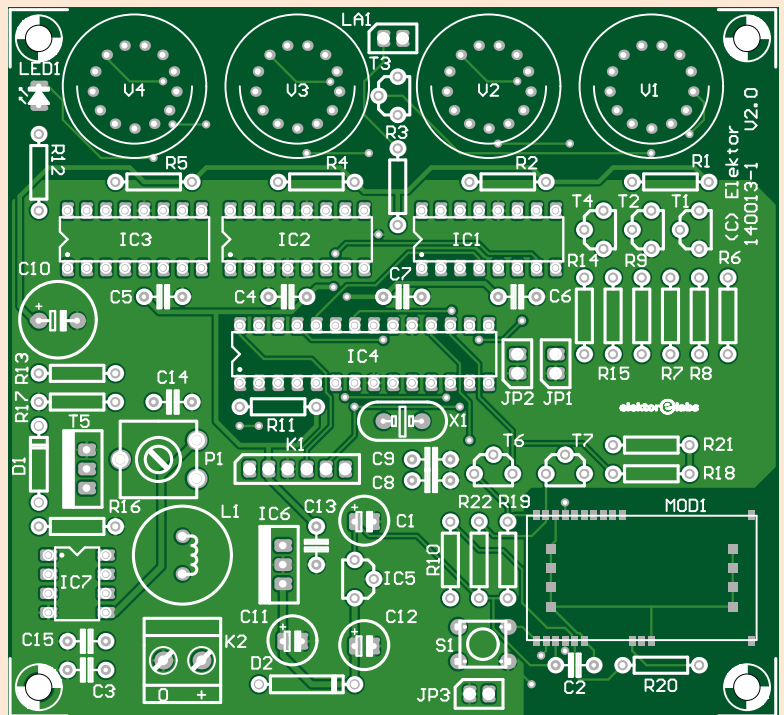


Figure 2.
Toute l'électronique, récepteur GPS inclus, est sur la même carte. Il ne manque que l'adaptateur secteur.

mise au PIC, ne serait corrigée — dans le pire des cas — qu’après près d’1 h.

C’est ici qu’intervient le second mécanisme de vérification (qui élimine aussi les boucles d’interruption imprécises). Pour l’heure du PIC également, on calcule, pour chaque seconde, une valeur de 3 octets représentative du nombre de secondes écoulées. Tant qu’il existe un signal GPS stable ou lorsqu’il s’en présente un, on soustrait l’une de l’autre la valeur des « secondes GPS » et celle des « secondes PIC ». Si le résultat dépasse 3 s (choix arbitraire), l’heure du PIC est synchronisée avec l’heure GPS.

Cette méthode peut sembler complexe (et en fait elle l’est) mais elle a fait ses preuves et fonctionne parfaitement à long terme. La luciole au néon prise entre les heures et les minutes, témoigne de la fiabilité du signal. Si elle clignote ou reste allumée en permanence (J2), tout va bien ; éteinte, elle indique que l’horloge fonctionne sur son seul signal interne parce qu’aucune chaîne de GPS valide n’a été reçue (depuis environ 30 s) et elle pourrait donc se mettre à dériver.

Réalisation

Le dessin du circuit imprimé (**fig. 2**) est téléchargeable [3]. Tous les composants prennent place sur la carte, seule l’alimentation (adaptateur secteur) est externe. Comme il s’agit de composants traversants, leur mise en place n’a rien de délicat, seul le module GPS demande une attention, une méthode et des outils particuliers (fer à souder à pointe fine !), car ses petits îlots de soudure devront être soudés aux plots correspondants du circuit imprimé. Le four à refusion est le seul moyen d’arriver à souder les petits plans de masse de la face inférieure du module, mais, heureusement, le montage fonctionne parfaitement même sans souder ces plans de masse, nous l’avons vérifié.

Lors du montage, commencez par les petits composants (bas) et passez ensuite aux plus grands (haut). Les régulateurs de tension se passent de radiateur. Manipulez délicatement les tubes Nixie, ils sont fragiles et leurs fils sont minces. Il existe, pour ces tubes, une sorte de support, qui leur assure une meilleure stabilité. Commencez par couper (pas trop court !) leurs fils de façon étagée, pour en faciliter l’insertion successive dans les orifices du circuit imprimé en commençant par le plus long. Une fois toutes les connexions enfilées, soudez un premier fil, vérifiez l’aplomb

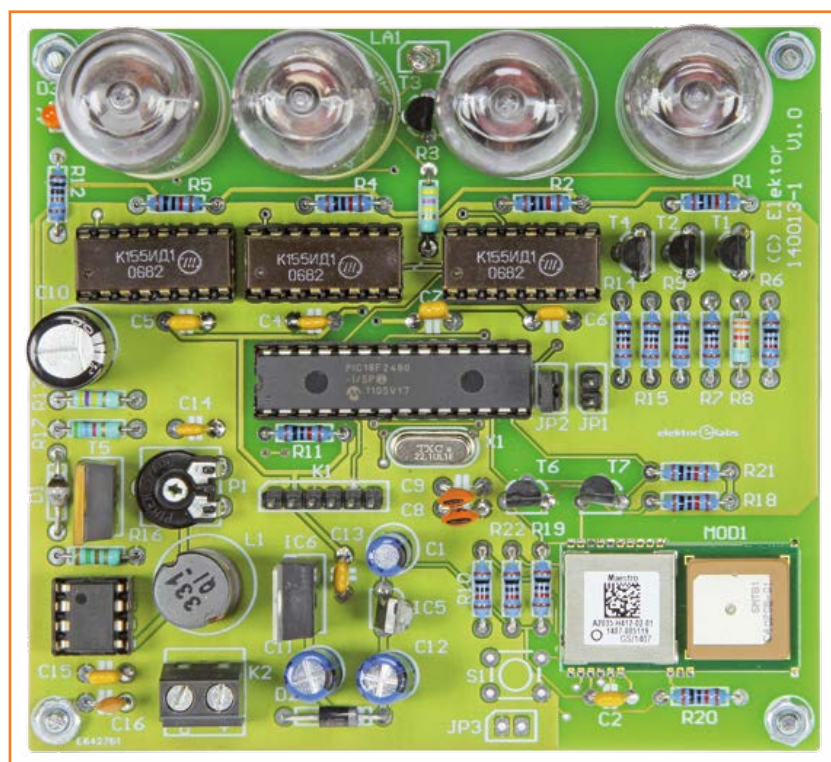
du tube (sans forcer !), soudez-en un deuxième, revérifiez la position du tube (sans forcer !), puis soudez les autres fils.

Connectez un adaptateur secteur (12 V/1 A p. ex.) et attendez que le module GPS ait une bonne réception.

Montez votre circuit dans un joli boîtier, en vous assurant surtout que rien ni personne ne puisse entrer involontairement en contact avec des pièces où règne le potentiel de 180 V !

La prochaine fois, pour les nostalgiques, je rajouterai le tic-tac.

(140013 – version française : Guy Raedersdorf)



Liens

- [1] Spoutnik donne l’heure : www.elektor-magazine.fr/050018
- [2] Thermomètre hygromètre à tubes Nixie : www.elektor-magazine.fr/110321
- [3] www.elektor-magazine.fr/140013

mise à jour du logiciel du régulateur photovoltaïque

nouvelle fonction de mesure de température de la batterie



La modification du régulateur photovoltaïque décrite dans cet article vise à gérer l'utilisation de la batterie aux limites de la plage admissible définie par son fabricant, et même au-delà.

Pascal Rondane
(Tours)

Dans le numéro double de janvier-février 2014, nous avons présenté un module de régulation de puissance photovoltaïque. Prévu à l'origine pour alimenter une petite station météo, celui-ci est utilisable dans bien des circonstances où l'autonomie est primordiale, et avec elle une gestion optimale de l'énergie récoltée par un panneau solaire et stockée dans une batterie. Il était question dans ce premier article [1] d'une fonction

de surveillance de la température de la batterie, restée dans l'oeuf... La voici enfin, aboutie et dûment testée par l'auteur. Remarquez que cette nouvelle version n'est utile que si on souhaite utiliser cette fonction améliorée, elle n'est pas à proprement parler une correction de la précédente version.

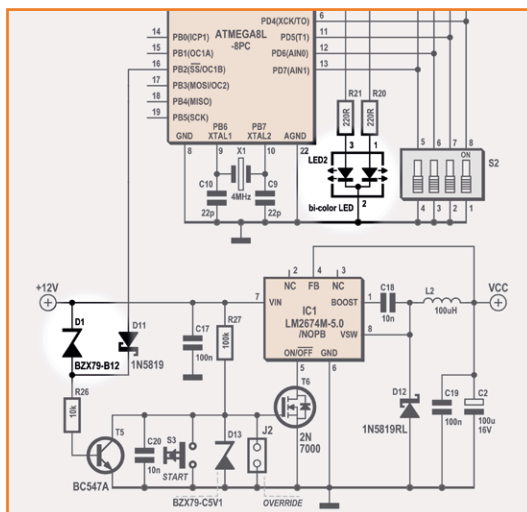
Côté matériel

Deux options sont possibles :

1. En cas de dépassement des seuils d'alerte de la température de la batterie, le microcontrôleur coupe l'électronique. Il faut ensuite un réenclenchement *manuel* du régulateur. On supprime pour cela la zener D1 sur le schéma existant (**fig. 1**), ce qui enlève aussi au module la possibilité de se réenclencher de façon entièrement automatique lorsque la batterie remonte au-dessus de 12 V après une décharge profonde.
2. En cas de dépassement de température de la batterie, le μC allume la LED d'état (*status*) D2 et coupe automatiquement la charge utilisateur et le panneau solaire. La diode D1 reste câblée, il n'y a pas de coupure du régulateur, celui-ci ne perd aucune de ses fonctions décrites dans le premier article.

Côté matériel, il n'y a donc pas grand-chose de

Figure 1.
Avec la nouvelle version du logiciel de surveillance de la température de la batterie, deux modes de fonctionnement sont possibles, dont un sans D1.



changé. Nous y reviendrons dans un instant. Voyons d’abord le nouveau logiciel.

Côté logiciel

L’ajout de la fonction de mesure de température de la batterie a donné naissance à une nouvelle version du logiciel (Photovoltaic regulator-Ver 2.5.2) [1] dont je précise d’emblée qu’elle ne peut être modifiée et compilée que sur la version complète de Bascom.

Pour mesurer la température, je fais appel à un tableau décrivant la fonction de transfert entre température ambiante et celle fournie par le fabriquant de la CTN, qui permet une acquisition par pas de 5°. Ensuite, je procède à une linéarisation entre ces deux points de mesure afin d’obtenir une précision d’un degré.

Le programme est commenté et les formules de calcul documentées permettent de comprendre facilement le fonctionnement.

Aux lecteurs intéressés par ces questions, je signale une autre méthode de linéarisation qui met en parallèle une résistance de valeur identique à celle de la CTN [2].

La routine de mesure de température est appelée régulièrement depuis la boucle principale du programme ; sur le port série, on peut récupérer la température et l’état du système (voir les **copies d’écran**).

Les seuils d’alarme de température sont fixés par trois variables facilement paramétrables dans la partie Temperature alarm thresholds setting du programme. Celui-ci considère la CTN comme

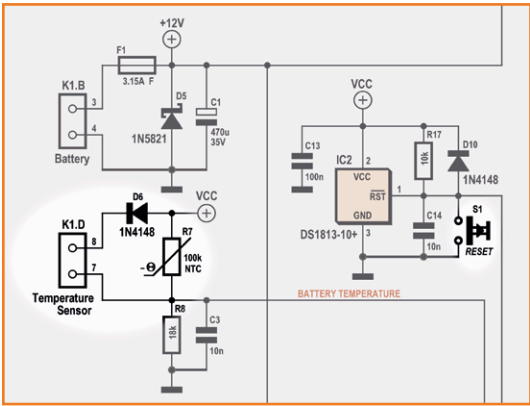


Figure 2. Si la CTN est reliée au bornier K1.D, il faut remplacer D6 par un pont de câblage.

```
Photovoltaic Regulator Ver 2.5.2, Elektor 2014
Gel Battery
Number of completed cycles: 0
Number of charge cycles: 0
Voltage = 12.82 V
Temperature = 60°
High Temperature alarm !
```

```
Photovoltaic Regulator Ver 2.5.2, Elektor 2014
Gel Battery
Number of completed cycles: 0
Number of charge cycles: 0
Voltage = 12.80 V
Temperature = -23°
Low Temperature alarm !
```

```
Photovoltaic Regulator Ver 2.5.2, Elektor 2014
Gel Battery
Number of completed cycles: 0
Number of charge cycles: 0
Voltage = 12.82 V
Temperature = -40°
No Temperature Sensor !
```

LED2				
couleur	clignotement (rapport cyclique)	panneau solaire	charge utile (S2=off)	batterie
orange	lent (33%)	déconnecté	connectée	chargé
vert	lent (2%)	connecté	connectée	charge en cours
orange	rapide (10%)	connecté	connectée	faible
rouge	lent (2%)	connecté	déconnectée	déchargée
éteinte	-	connecté	déconnectée	trop faible
état		message		
orange fixe	CTN absente		No Temperature Sensor !	
rouge fixe	température inférieure à la consigne		Low Temperature alarm !	
rouge fixe	température supérieure à la consigne		High Temperature alarm !	
Au démarrage, avant qu'apparaisse l'indication conforme à ce tableau, il y a deux brèves séquences rouge-orange-vert-off suivies de 2 sec. de vert. Cette LED s'éteint quand le niveau de la batterie est trop faible (10,8 V) ; l'alimentation 5 V est alors mise hors service et le µC s'arrête de fonctionner.				

Tableau 1. Les indications de la LED d’état bicolore D2.

absente lorsque « la température mesurée » est inférieure à -40° , ce qui n'est pas fréquent sous nos latitudes.

Le nouveau **tableau** montre comment la LED D2 rend compte de l'état du système. Les indications données dans le premier article restent valables, elles sont reprises et complétées ici pour la nouvelle version du logiciel.

Modifications

ALIMENTATION-MODULE CONVERTISSEUR

Pour que le μ C coupe l'électronique en cas de dépassement de la température de la batterie, supprimer la D1. Le redémarrage après coupure est alors manuel et se fait avec le bouton poussoir S1.

CAPTEUR DE TEMPÉRATURE (CTN)

Il y a deux manières de raccorder la CTN (**fig. 2**) : soit directement au circuit imprimé (R7), et aucune modification n'est nécessaire, soit au bornier K1.D, et il faut alors remplacer la diode D6 (1N4148) par un bout de fil ou une résistance de $0\ \Omega$. Cette suppression est nécessaire, sinon la diode introduit une chute de tension qui fausserait la mesure de température.

Si vous souhaitez déporter la CTN hors du coffret, la longueur de fil (une paire torsadée) ne devrait pas dépasser 40 cm. La CTN peut être isolée par de la gaine thermo-rétractable puis collée sur la batterie par une goutte de colle.

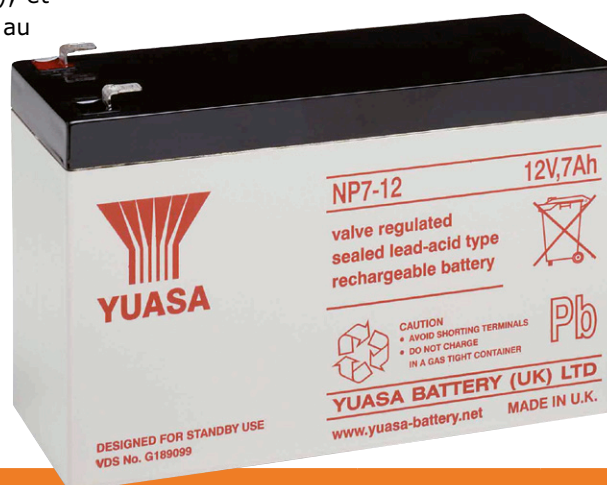
J'ai utilisé une CTN de Vishay : NTCLE100E3104JB0 (*Farnell* : 1187033) à laquelle j'ai adapté le logiciel. Si vous utilisez un autre modèle, il suffira de modifier les valeurs du tableau Temperaturetable dans le programme.

Pour compléter, voici à titre d'exemple, les caractéristiques d'une batterie Yuasa telles qu'elles sont données par le fabricant.

(140033)

Liens

- [1] Description détaillée du régulateur
www.elektor-magazine.fr/080305
- [2] Logiciel source
www.elektor-magazine.fr/140033
- [3] <http://ressources.univ-lemans.fr/AccesLibre/UM/Pedago/physique/02/electro/thermist.html>



Spécifications

Tension nominale	12	V
Capacité en 20h (C20) à 1,75V/élé. (20°C)	7	Ah
Capacité en 10h (C10) à 1,75V/élé. (20°C)	6,4	Ah

Dimensions

Longueur	151 (± 1)	mm
Largeur	65 (± 1)	mm
Hauteur		mm
Hauteur bornes incluses	97,5 (± 2)	mm
Poids (typique)	2,2	kg

Types de bornes

Cosse à languette (type Faston, montage rapide, démontable)	4,7	mm
---	-----	----

Plage de température de fonctionnement

Stockage	-20 à +60	°C
Charge	-15 à +50	
Décharge	-20 à +60	

Tableau 2. Caractéristiques de la batterie au plomb NP7-12 de Yuasa.

magnétrons

drôle de composant n°9

Après plusieurs articles consacrés à de *drôles de semiconducteurs*, il serait temps de varier les plaisirs : aujourd'hui on parle magnétrons. Vous les utilisez sans doute beaucoup, mais plutôt à la cuisine : ce sont les éléments centraux des fours à microondes, mais ils sont également utilisés par l'industrie des semiconducteurs, entre autres, pour chauffer, sécher ou encore faire durcir. Comment fonctionnent-ils ?

Le dessin de la **figure 1** montre la composition d'un magnétron à cavité. La cathode au centre du magnétron est chauffée, ce qui entraîne l'écoulement des électrons dans la cavité. Ils s'écouleraient directement vers l'anode, formant une diode, si les aimants supérieur et inférieur ne forçaient pas leur écoulement circulaire dans la cavité et leur accélération jusqu'à ce qu'ils puissent atteindre l'anode de l'antenne et s'échapper du magnétron. Chacune des extrémités du magnétron comprend également deux cavités résonnantes dont la fonction est de définir la fréquence en sortie de l'antenne. L'une des premières applications des magnétrons était comme source d'énergie radio pour les systèmes radar. Ils étaient l'un des moyens les plus efficaces de produire les courtes impulsions radio de haute puissance nécessaires en imagerie radar. L'un des défis avec les magnétrons est que la fréquence en sortie change avec la température et l'impédance de charge. De plus, la gamme de fréquences d'émission des magnétrons est très large et il n'y a pas de récepteurs radar très sélectifs ni très sensibles. Toutefois, ces inconvénients ne pesaient guère, tant la puissance de sortie et l'efficacité des magnétrons sont grandes. Les magnétrons commerciaux nécessitent une tension d'anode de 4 kV à 10 kV en fonction du modèle. Les modèles plus petits, entre 1 kW et 2 kW, utilisés dans les fours à microondes, sont refroidis par air. Pour les gros modèles industriels de 3 kW et plus, il faut un circuit de refroidissement à eau. Pour qui souhaite expérimenter, les fours à microondes mis au rebut sont une formidable source de magnétrons, mais n'oubliez jamais que les rayonnements microonde sont dangereux tout comme les tensions d'alimentation de plusieurs milliers de volts. J'ai récupéré un magnétron pour voir si je pouvais mesurer sa dérive en fréquence

au fil du temps (**fig. 2**), ce qui s'est révélé bien plus difficile que prévu : le magnétron changeait de fréquence plus rapidement que ce que l'analyseur de spectre était capable de mesurer en temps réel et j'ai donc utilisé un mode de détection de crête pour voir de combien la fréquence variait. Au début de ce test, la fréquence était d'environ 2,465 GHz puis elle a dérivé pendant les 30 secondes du test. La ligne à -20dB montre la déviation en fréquence ;

Neil Gruending
(Canada)

Figure 1.
Structure d'un magnétron.

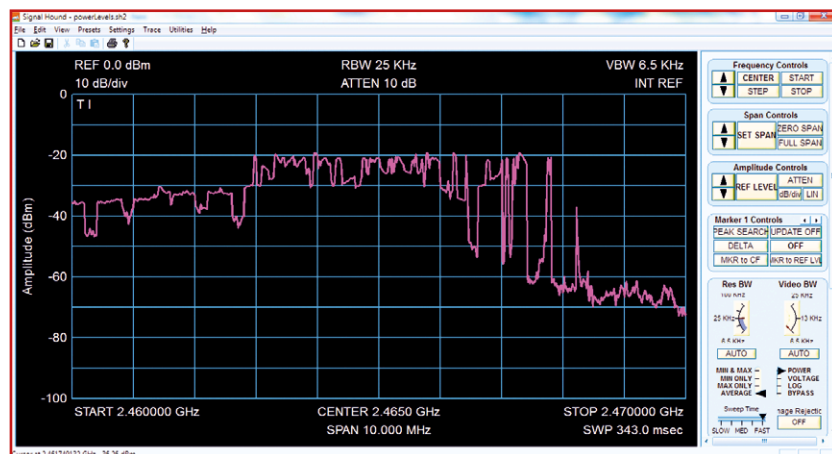
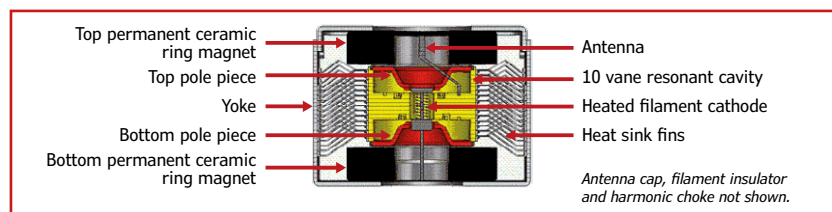


Figure 2.
Spectre en sortie d'un magnétron.

elle aurait probablement été plus plate si j'avais augmenté la durée du test. Un autre défi lorsque l'on travaille avec les magnétrons est de dissiper toute l'énergie RF. Pour ce test j'ai utilisé un guide d'onde clos avec une charge refroidie par eau. Pour finir, j'insiste : si vous voulez vous amuser avec un magnétron, ne négligez pas la sécurité, je tiens à vous retrouver ici le mois prochain.

(140241 - version française : Kévin Petit)

Liens

- [1] www.cst.com/Applications/Article/Magnetron-And-Microwave-Oven-Design-To-Solve-Wi-Fi-Interference-Issues

transfo de courant (encore) meilleur

J. T. van Es

(Pays-Bas)

Mon intervention fait suite aux deux articles de Martin Oßmann, parus en avril et mai derniers, sur le calcul et l'utilisation de transformateurs de courant. Martin cherche à abaisser la fréquence de coupure basse déterminée par L/R , entre autres par le matériau du noyau.

Rien de faux, mais ne pourrait-on pas faire mieux et plus facilement ?

Figure 1.
Le circuit de base.

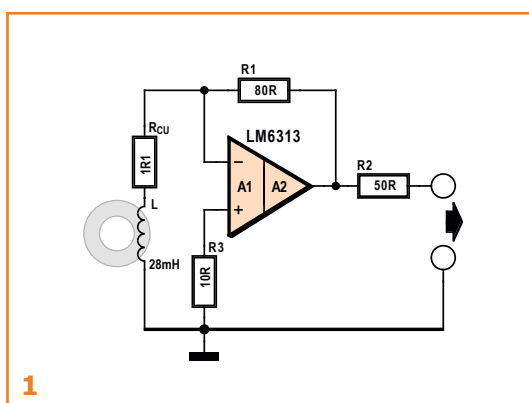
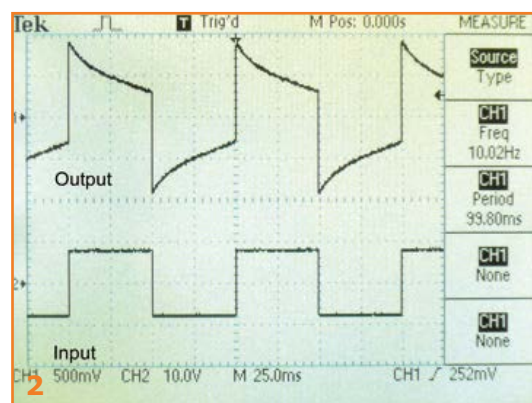


Figure 2.
Le signal de sortie à 10 Hz sans rétroaction positive.



La **figure 1** montre le trajet du signal dans un ampèremètre que j'avais fabriqué en 2005 dans le même but. Rangé depuis, je l'ai exhumé pour la circonstance. Le transfo de courant est une bobine de capteur d'un injecteur d'essence *Tubantor* de 1983. Elle contient un noyau de ferrite et 80 spires. La bande passante à -3 dB de l'ensemble va de 7 Hz à 5,5 MHz. Un signal rectangulaire à 10 Hz ne sort pas indemne de l'aven-

ture, la **figure 2** le prouve. R3 sert à corriger le décalage d'entrée de l'amplificateur.

Augmenter le couplage de l'ampli au moyen de R4 et R5 (**fig. 3**) rend négative sa résistance d'entrée (réglable par R5) et compense la R_{cu} . Quand la résistance du circuit diminue, le rapport L/R augmente. On en voit l'effet à la **figure 4**.

Figure 3.
Dosage de la réaction par R4 et R5.

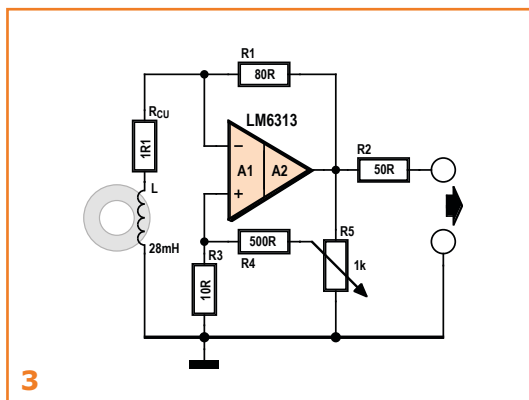
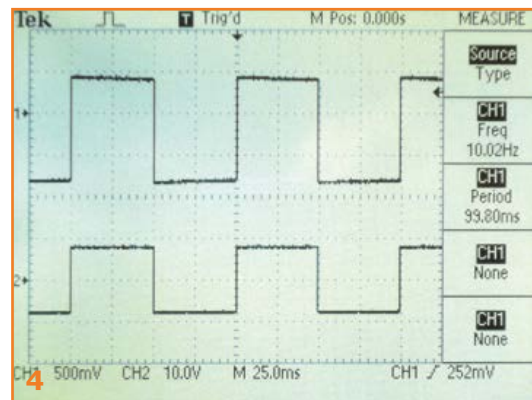


Figure 4.
Le signal de sortie à 10 Hz avec réaction.



En poussant davantage la rétroaction positive (ou réaction), on arrive à 1 Hz (**fig. 5**), mais la stabilité en continu devient précaire. Aussi, la **figure 6** indique-t-elle une solution par l'introduction d'un ampli op pour en assurer le réglage avec R7 et C1. Cet ampli doit avoir une entrée à FET qui permet l'emploi de grandes valeurs pour R7 et C1, lequel, pour limiter les pertes, sera d'un modèle à film.

Maintenant que le bobinage secondaire est presque en parfait court-circuit, il ne peut plus y avoir de tension à ses bornes. Côté primaire non plus, mise à part la résistance du cuivre. L'induction dans le noyau, qui est déjà faible dans un transfo de courant, atteint aussi un minimum absolu. La valeur de L peut alors être plus basse, tout bénéfice pour le comportement en HF. La mesure de signaux en continu n'a aucune chance de réussir, il faudrait pour cela un capteur à effet Hall. L'ennui de cette technique, c'est qu'elle nécessite une fente dans le noyau, au détriment du facteur μ (μ) et donc de la valeur de L.

Jadis, vers 1985, nous avons ainsi développé un moniteur de faisceau de particules avec un noyau torique *Vitrovac* dont le diamètre intérieur permet d'y glisser une bouteille de bière. La bande passante atteinte va de 23 mHz à 20 MHz !

(140266 - version française : Robert Grignard)

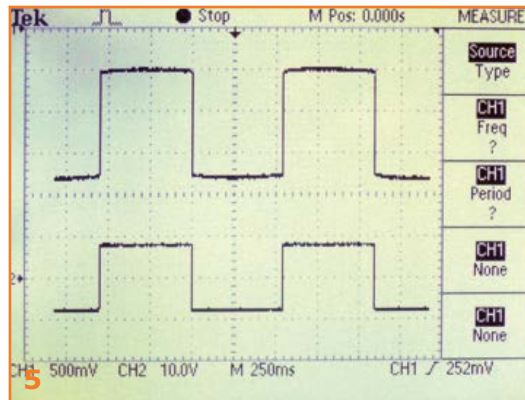


Figure 5.
Le signal de sortie à 1 Hz avec plus de réaction.

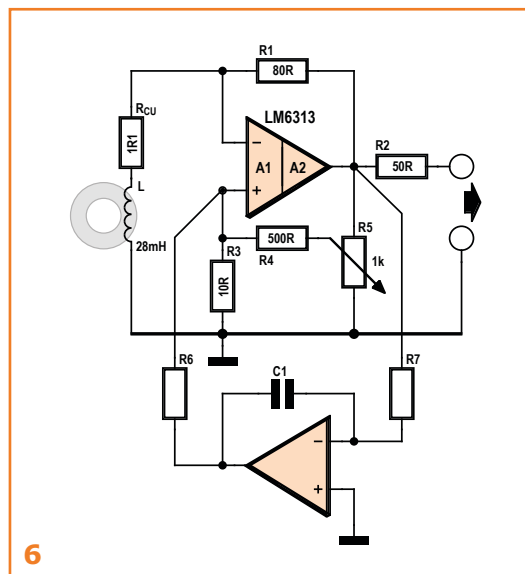
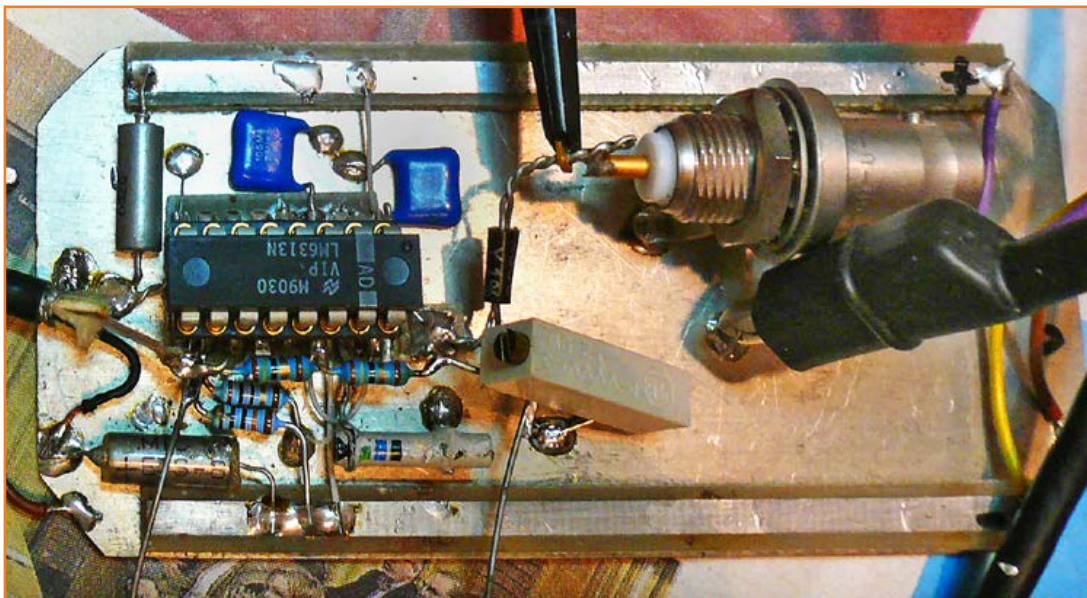


Figure 6.
Amélioration de la stabilité en continu par ampli op et R7/C1.



Mon prototype d'amplificateur.



programmer avec Mono

pour Windows, Linux, Mac OS et Android

Fatalité ou nécessité, l'électronicien troque de plus en plus souvent son fer à souder pour le clavier et écrit maintenant des programmes. La plateforme Mono présente de ce point de vue un avantage : elle permet de construire des applications pour Windows, Linux, Mac OS, et même Android et iPhone. En voici un tour d'horizon, agrémenté d'un exemple pratique.

Veikko Krypczyk
(Allemagne)

Le rêve de tout développeur est de n'avoir à écrire le code de son application qu'une seule fois, de n'avoir à créer qu'une seule interface graphique, puis de fournir une unique version de son produit fini aux principaux systèmes d'exploitation. Le rêve de l'électronicien qui prévoit pour son circuit une commande par PC n'est pas différent : que son programme fonctionne aussi bien sous Windows que sous Linux ou Mac OS.

Pourquoi Mono?

Soyons clair : il n'existe pas de panacée pour développer des programmes, et l'outil idéal multiplateforme n'est pas pour demain. Autrement dit, la généralisation se fera toujours aux dépens de la spécialisation ! L'important est de se servir des bons outils et de trouver un juste milieu entre les différents objectifs du développement. Mono est largement compatible avec la plateforme .NET de Microsoft, et de surcroît utilisable sur plus de systèmes d'exploitation. Un argument en faveur de Mono est la (croissante) popularité de C#, et plus généralement des technologies .NET. La plateforme .NET compte entre autres avantages d'offrir une bibliothèque de classes étoffée, un grand choix de langages (C#, Visual Basic, Delphi, etc.) et un environnement de développement efficace. Disposer de ces atouts sur d'autres systèmes d'exploitation est pour l'essentiel ce

qui a motivé le développement de Mono et justifie aujourd'hui son utilisation. Et pour qui souhaite réduire sa dépendance à Microsoft, de nombreux composants Mono sont à code source ouvert. Le projet Mono, qui existe maintenant depuis une dizaine d'années [1], comprend les sous-projets suivants [2] :

- Mono : l'élément de base ; d'un point de vue technique, son modèle est le *framework* .NET. L'utilisation de Mono pour Windows et Linux est gratuite.
- Xamarin.iOS (MonoTouch) : permet d'écrire en C# des applications pour iPhone et iPad. Son utilisation est soumise à l'achat d'une licence, mais il existe une version d'essai gratuite. Les versions payantes sont les éditions *Indie* (399 \$/an), *Business* (999 \$/an) et *Enterprise* (1899 \$/an).
- Xamarin.Android : permet de développer des applications pour la plateforme Android. Les éditions et le tarif des souscriptions sont identiques à ceux de Xamarin.iOS.

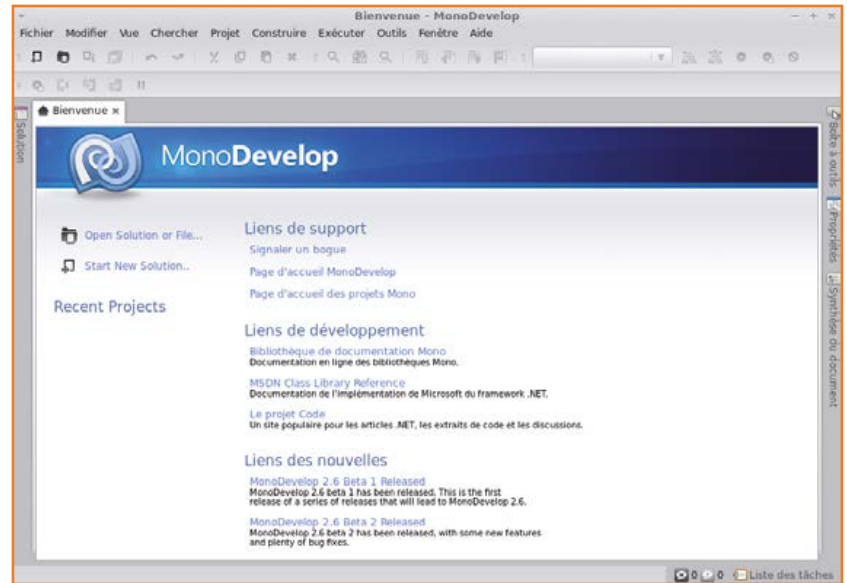
Démarrer bien chaussé

La clé du succès avec Mono exige quelques préalables : bien connaître les dessous des systèmes d'exploitation, p. ex. Windows et Linux, avoir un bagage minimal en C#, maîtriser les fondements de la création d'applications dotées

d'interfaces utilisateur graphiques, avoir déjà utilisé un environnement de développement intégré (EDI), et comprendre le fonctionnement du *framework* .NET. Développer des programmes indépendants des systèmes d'exploitation n'est donc pas à la portée des débutants ; si vous en faites partie, vous pouvez vous tourner vers l'environnement gratuit Visual C# 2010 de Microsoft (cf. Elektor de février et mars 2009 [3]). Après quelques projets réussis sous Visual, vous pourrez tenter le développement multiplateforme, et à ce moment-là Mono deviendra une option intéressante.

Environnement de développement

Mono lui-même n'est que l'équivalent de l'environnement d'exécution .NET. L'écriture des programmes se fait sous MonoDevelop, un environnement de développement intégré complet, comparable à Visual Studio d'un point de vue structure, conception et utilisation (voir **encadré**). Les informations les plus récentes



concernant cet EDI se trouvent sur le site de MonoDevelop [4].

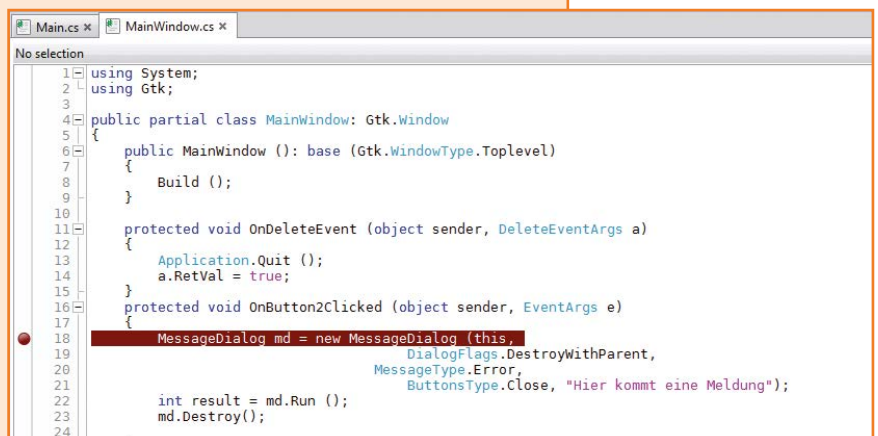
Figure 1.
La fenêtre de bienvenue de MonoDevelop (version 2.6) sous Linux.

MonoDevelop le multi-développé

Avec son interface utilisateur typique des environnements de programmation intégrés, MonoDevelop offre un large potentiel d'utilisation. Voici quelques menus incontournables :

- *Fichier -> Nouveau -> Solution* : crée un nouveau projet.
- *Projet -> Options de projet* : pour modifier les paramètres de construction du projet, par exemple les options du compilateur.
- *Construire et Exécuter* : pour construire ou reconstruire un projet, et pour exécuter l'application du projet avec ou sans débogage.
- *Exécuter -> Nouveau point d'arrêt* : lors de son exécution le programme est interrompu au point spécifié (voir capture d'écran) et vous pouvez inspecter l'état des variables ou encore les objets.
- *Contrôle de version* : permet d'extraire ou de publier un système de contrôle de version.
- *Outils -> Gestionnaire de compléments* : pour installer, désinstaller ou gérer diverses extensions.

Avant de vous lancer dans un projet « sérieux » avec MonoDevelop, prenez le temps de vous familiariser avec ses fonctions en construisant quelques éléments d'interface simples, votre méthode de travail en sera d'autant plus affûtée. Et comme toujours, pensez à sauvegarder régulièrement votre projet !



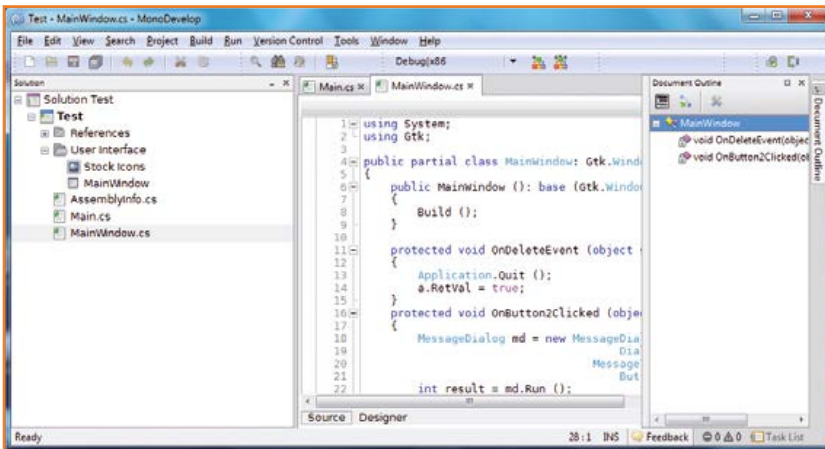


Figure 2. MonoDevelop 3.0 sous Windows.

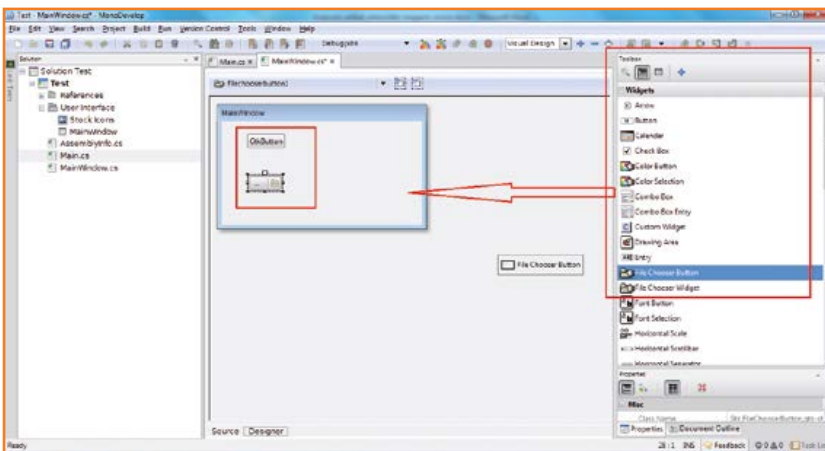


Figure 3. Utilisation de l'outil Concepteur de MonoDevelop.

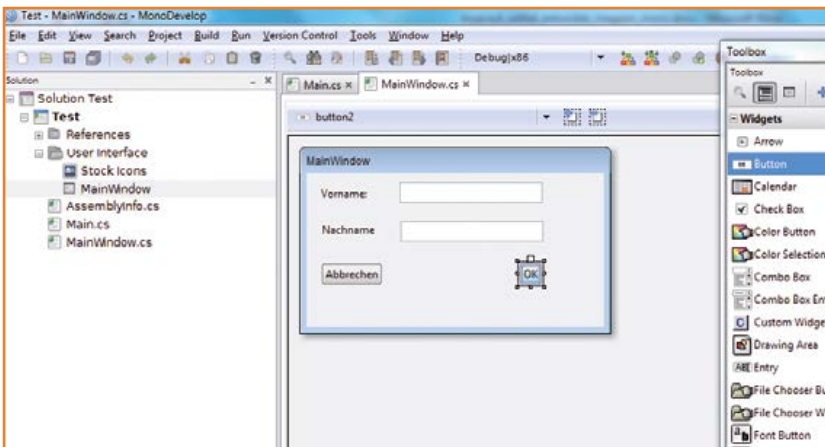


Figure 4. Ebauche de l'interface graphique du projet Test.

Installation sous Linux

Voyons d'abord comment installer Mono et MonoDevelop sous Linux. Je travaille avec la version 12.04 LTS d'Ubuntu, mais d'autres distributions sont bien sûr possibles. Ouvrez votre gestionnaire de logiciels, entrez MonoDevelop dans la boîte de recherche, sélectionnez-le, puis cliquez sur *Installation*. Le paquet et les éventuelles dépendances seront automatiquement téléchargés et installés. Notez que toutes les distributions ne disposent pas de la version la plus récente de MonoDevelop dans leurs dépôts. La mienne est la version 2.6. Le meilleur moyen d'avoir la dernière version est de récupérer le code source sur le site de MonoDevelop et de le compiler soi-même.

Découvrons ensemble MonoDevelop. Une fenêtre de bienvenue s'affiche au démarrage (**fig. 1**). Cliquez sur *Fichier/Nouveau/Solution* pour créer un nouveau projet. La fenêtre qui s'ouvre propose par défaut une solution C#. Sélectionnez *Projet Gtk# 2.0*, donnez un nom au projet, indiquez son emplacement, puis cliquez sur *Suivant*. La fenêtre qui s'ouvre vous demande (ou impose) la version Gtk# cible, dans mon cas il s'agit de la 2.8. Mono peut en principe gérer différentes interfaces utilisateur graphiques, dont GTK, Qyoto et Qt4Dotnet.

Laquelle de ces bibliothèques choisir ?

Les facteurs en jeu sont trop nombreux pour que la réponse soit unique. Une page du site Mono [5] suggère néanmoins quelques critères de choix en recensant les avantages et inconvénients des différentes bibliothèques graphiques. Nous utiliserons ici Gtk#. GTK+ (*GIMP Toolkit*) est un environnement universel pour la conception d'interfaces graphiques [6]. Il comprend un ensemble complet d'éléments graphiques (*widgets*), tels que ceux nécessaires à la création de boîtes de dialogue. GTK+ est écrit en C mais supporte d'autres langages que le C/C++, Python p. ex. La liaison entre Gtk+ et le *framework* Mono (et le langage C#) est réalisée à l'aide d'un *wrapper*. Le suffixe # derrière GTK indique clairement une utilisation de GTK par Mono (C#).

... et sous Windows

Revenons à Windows pour y installer Mono et MonoDevelop. La version la plus récente de

MonoDevelop est à ce jour la 4.0. Notez qu'avant de l'installer vous devez installer le paquet « GTK# for .NET ». Bonne surprise au démarrage, l'interface est moderne et légère (**fig. 2**). Répétez la procédure suivie sous Linux pour créer une nouvelle solution.

Premier test

L'outil *Concepteur* (**fig. 3**) permet d'assembler rapidement les éléments d'une interface graphique et d'en réaliser une première ébauche. Utilisons-le pour placer sur un formulaire deux éléments *Labels*, deux zones d'entrée (*Entry*) et deux boutons (**fig. 4**). Avant cela nous devons toutefois insérer un conteneur. Choisissez un conteneur de type *Fixed*, car il permet d'agencer précisément les éléments du formulaire.

La liaison entre les éléments de l'interface graphique et la logique du programme s'effectue à l'aide de « signaux », un processus de liaison qui correspond à la liaison d'événements utilisée par l'interface *Windows Forms* de .NET.

Après avoir p. ex. sélectionné le bouton OK, nous ouvrons l'onglet *Signaux de Propriétés*, et double-cliquons sur le signal *Clicked*. Un gestionnaire de signal sera ajouté à *MainWindows.cs* sous la forme d'un corps de fonction à compléter. Cette fonction sera appelée lorsque le bouton OK sera cliqué ou, en termes de MonoDevelop, lorsque le signal *Clicked* sera déclenché. Ladite fonction (**listage 1**) ouvre une boîte de dialogue qui affiche un message. Particularité de GTK, la destruction de cet objet *MessageDialog* doit se faire via le code (cf. [7] pour la syntaxe de la classe correspondante). Après compilation, l'application a la sortie attendue (**fig. 5**). La structure des répertoires des projets est pour l'essentiel la même que celle créée par Visual Studio. MonoDevelop crée entre autres fichiers un exécutable d'extension .exe.

Pour tester ce mini-projet sous Linux, il suffit d'y transférer le dossier du projet que nous venons de compléter. Ne reste plus alors qu'à effectuer un clic droit sur l'exécutable et à sélectionner *Ouvrir avec Mono Runtime* pour voir s'afficher le formulaire (**fig. 6**).

Portage d'applications

Développer une application prévue dès le départ pour être utilisée sur différents systèmes

Listage 1 : Code source de l'application de test

```
protected void OnButton2Clicked(object sender, EventArgs e)
{
    MessageDialog md = new MessageDialog (this,
    DialogFlags.DestroyWithParent,
    MessageType.Error,
    ButtonsType.Close,
    "Voici un message");
    int result = md.Run();
    md.Destroy();
}
```

d'exploitation ne pose pas de problème : l'objectif est pris en compte tout au long de la programmation. Il est cependant fréquent que l'application en question existe déjà pour .NET (donc pour Windows).

Se pose alors la question de l'effort à déployer pour son portage. Dans ce cas il faudra en effet remplacer les classes de la plateforme .NET par leur équivalent Mono. Le travail à fournir et les difficultés auxquelles s'attendre peuvent être



Figure 5.
Le formulaire sous
Windows 7.

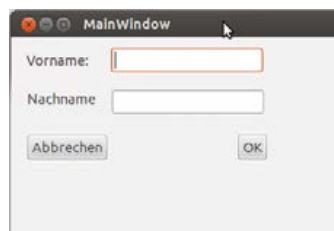


Figure 6.
Le formulaire sous Linux.

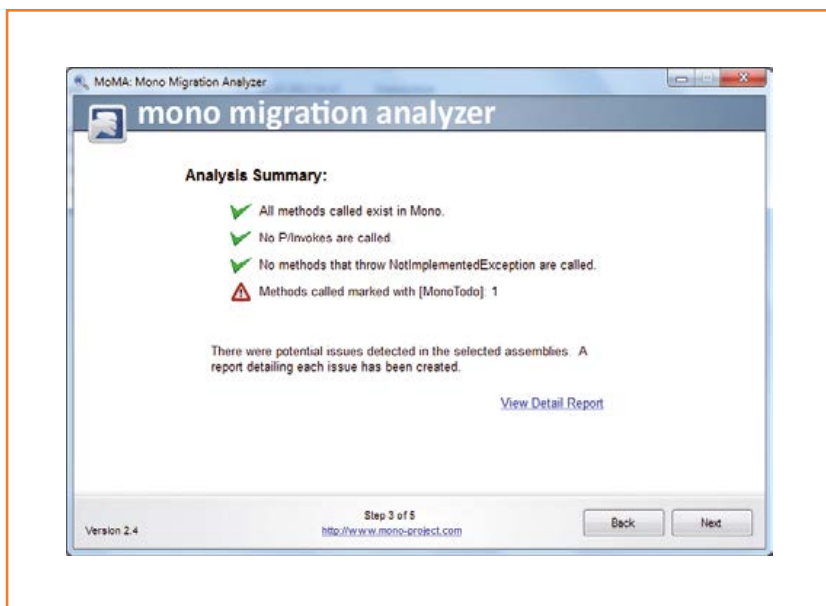


Figure 7.
L'outil *Mono Migration Analyzer* (MoMA) pour le portage d'applications sur la plateforme Mono.

estimés à l'aide de l'outil *Mono Migration Analyzer* (MoMA). Le processus d'analyse se lance après avoir sélectionné un fichier .exe ou .dll (fig. 7).

MoMA ne fait que fournir un rapport, au développeur de remplacer le cas échéant les parties concernées du programme (p. ex. des appels directs de l'API) par les méthodes Mono correspondantes. Le processus de portage pourra alors être finalisé.

Mono pour d'autres systèmes

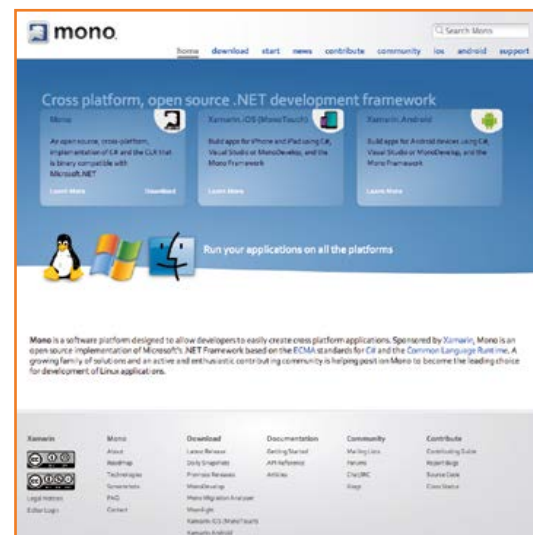
La cadre d'applications Mono offre beaucoup plus que ce que notre tour d'horizon a montré. Mono peut également être utilisé pour construire des applications pour Android et iOS. Un conseil toutefois : prenez le temps de vous familiariser avec tous les ressorts de Mono avant d'investir dans l'achat d'une licence Xamarin (cf. ci-dessus). Pour cela rien de tel que de réaliser plusieurs projets avec la version *open source* pour PC

de Mono. Du point de vue de l'électronicien, MonoDevelop est en effet une option intéressante, par exemple pour réaliser des commandes de matériels périphériques.

(120490 – version française : Hervé Moreau)

Liens

- [1] [http://fr.wikipedia.org/wiki/Mono_\(logiciel\)](http://fr.wikipedia.org/wiki/Mono_(logiciel))
- [2] www.mono-project.com/Main_Page
- [3] www.elektor.fr/080668
- [4] <http://monodevelop.com/>
- [5] www.mono-project.com/Gui_Toolkits
- [6] www.gtk.org/overview.php
- [7] <http://buttle.shangorilla.com/1.1/handlers/monodoc.ashx?link=T%3AGtk.MessageDialog>

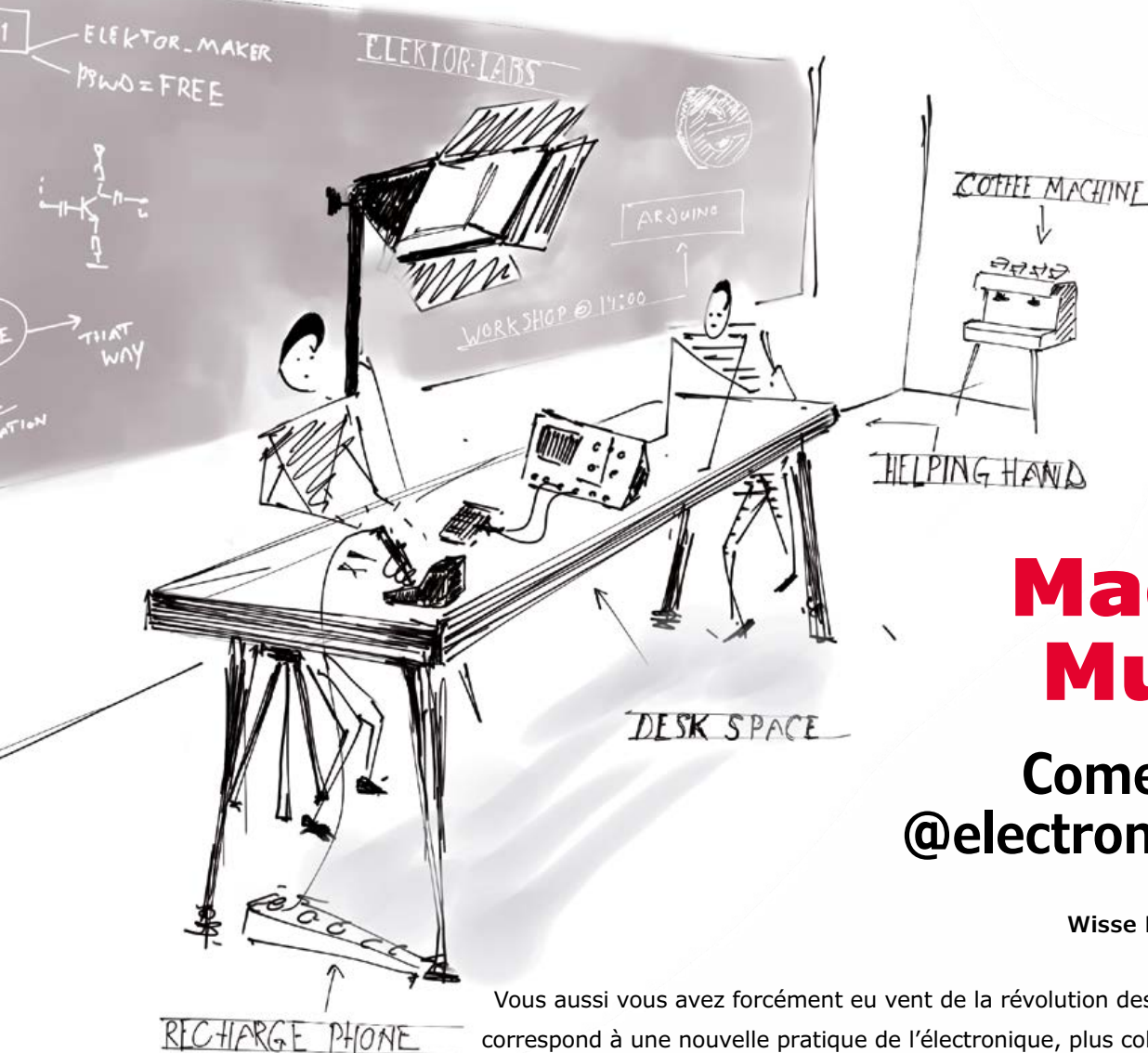


L'auteur

Veikko Krypczyk a étudié l'économie d'entreprise avec une spécialité en informatique de gestion. Sa thèse de doctorat portait sur les algorithmes d'optimisation des itinéraires. Programmeur passionné, il est auteur d'articles spécialisés, consultant en développement logiciel, et aime manier le fer à souder durant son temps libre pour concrétiser divers projets.

Pour toute question ou suggestion, écrivez-lui à info@it-fachartikel.de. La page de son site personnel (en allemand) est à l'adresse <http://it-fachartikel.de>.

ELEKTOR-MAKER-SPACE



Made in Munich

Come MAKE it @electronica 2014

Wisse Hettinga (Elektor Labs)

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

Une grande partie de l'équipe d'Elektor vous accueillera pour donner forme à ce projet **AVEC VOUS**.

Peu importe votre degré de maîtrise des langues, allemand, anglais, néerlandais, espagnol ou C++, on se débrouillera. Profitez des ressources mises à votre disposition : tables, outils, appareils de mesure, imprimante 3D et WiFi à l'oeil.

Pour le café, nous acceptons les dons. Et si vous avez oublié le chargeur de votre téléphone, raison de plus de passer au stand d'Elektor, vous en trouverez un chez nous.

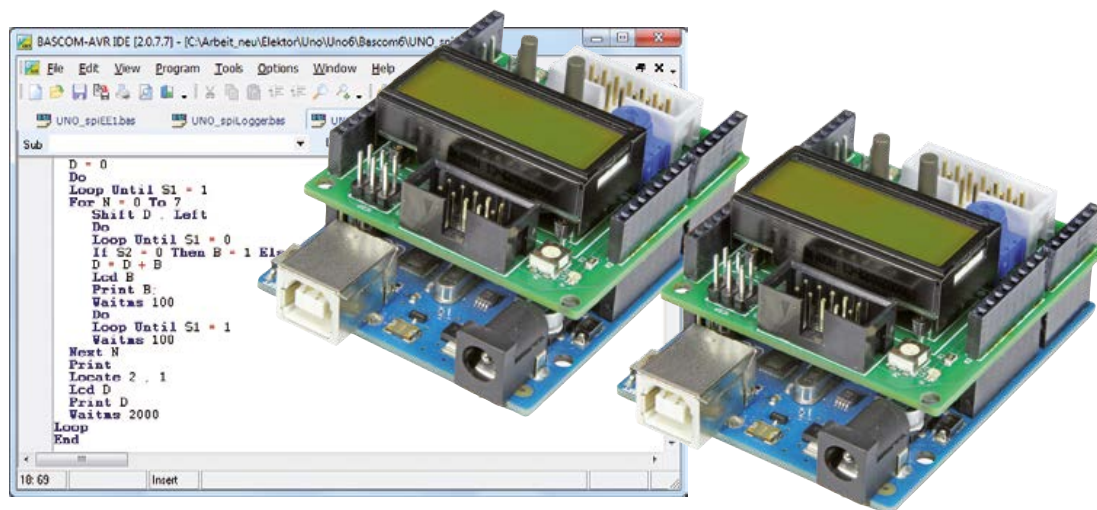
Elektor Maker Space | hall A6 | stand 380

Suivez les activités d'Elektor !

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

l'atelier du microcontrôleur (6)

l'interface SPI



Burkhard Kainka
(Allemagne)

Transmettre des informations en série, autrement dit une unité d'information après l'autre, est la façon habituelle de communiquer. C'est ce que nous faisons lorsque nous parlons, lisons, écrivons, ou encore téléphonons. Si dans ce cas un seul fil suffit, le fait d'ajouter une ligne d'horloge rendra la communication encore plus fiable !

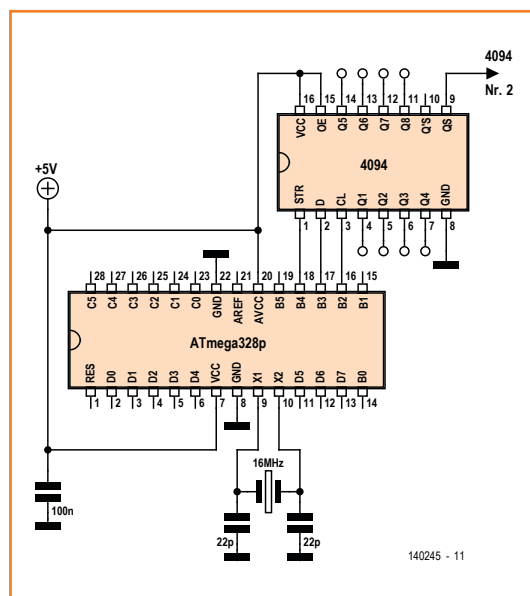


Figure 1.
Connexions entre le registre 4094 et l'ATmega.

Sur un bus SPI (*Serial Peripheral Interface*), les données sont transmises bit à bit sur un fil allant p. ex. d'un microcontrôleur à un afficheur, une EEPROM ou encore une carte SD. Dans la plupart des cas nous souhaitons également *lire* des données, c'est pourquoi un deuxième fil doit permettre aux bits d'aller dans l'autre sens.

L'intérêt d'ajouter un troisième fil, à savoir celui d'une ligne d'horloge, est de permettre à un signal dit d'horloge d'indiquer sans ambiguïté au destinataire à quel instant le bit suivant est présent sur le fil.

Les deux interlocuteurs de la ligne n'auront ainsi besoin ni de se mettre d'accord sur un protocole de synchronisation, ni d'utiliser un temporisateur (*timer*) pour superviser ce protocole. Avec ces trois fils, nous obtenons une transmission (relativement) robuste.

Extension de ports avec registres à décalage

Notre premier exemple n'utilisera pas une véritable interface SPI mais quelque chose de semblable. Inventés bien avant les microcontrôleurs, les registres à décalage se prêtent bien à l'explication du principe d'une transmission série. Ils permettent en outre de pallier le faible nombre de ports des cartes à microcontrôleurs, en particulier les Arduino. On peut ainsi construire facilement une extension de ports à l'aide du 4094, un registre à décalage à 8 bits. Il faut trois fils pour obtenir huit nouvelles sorties. Ou seize si on relie un second registre à décalage. Ou même quatre-vingts si on câble en série dix composants. La méthode s'avère particulièrement peu onéreuse lorsqu'on a besoin de nombreuses sorties.

La **figure 1** montre le câblage du 4094 et de l'Arduino. Le registre à décalage à 8 bits dispose d'une entrée de données *D* et d'une entrée d'horloge *CL*. Il faut d'abord placer les octets sur l'entrée *D* les uns après les autres en commençant par le bit de poids fort, puis appliquer une impulsion d'horloge positive sur *CL* pour chaque bit. On fait ensuite glisser l'information d'une bascule à l'autre du registre. Le 4094 possède aussi une entrée de transfert *STR* (*strobe*). Une impulsion *STR* transfère vers les bascules *D* des sorties toutes les données présentes dans le registre au moment de l'impulsion. Nous pourrions relier l'entrée de transfert à V_{CC} , mais dans ce cas tous les résultats intermédiaires d'un décalage seraient visibles. En déclenchant l'impulsion de transfert à la fin, seul le résultat final est visible.

Le code associé (**listage 1**) est simple. Pour afficher un octet *D*, nous copions d'abord son bit de poids fort dans la variable *B* de type *Bit* ($B = D.7$), puis nous le plaçons sur le port correspondant. Nous appliquons ensuite une impulsion d'horloge positive d'environ 1 ms sur *CL*. Une microseconde aurait suffi, mais ralentir l'affichage rend plus facile l'observation du processus à l'oscilloscope. L'instruction de décalage exécutée après l'impulsion d'horloge (*Shift D , Left*) décale tous les bits de l'octet *D* d'une position vers la gauche : celui qui était auparavant le bit 6 devient le prochain sur la sortie, et ainsi de suite, jusqu'à ce que les huit bits aient été décalés. Nous déclenchons alors l'impulsion de transfert, et les huit bits se retrouvent sur les sorties du 4094. Le code incrémente constamment l'octet de don-

nées à transmettre afin que les sorties du registre à décalage soient également modifiées durant l'exécution du programme. L'état en cours est affiché à la fois sur le LCD (si vous utilisez le shield Elektor) et sur le terminal. Il est donc

Listage 1. Affichage via un registre à décalage.

```
'-----
'UNO_shift.BAS  Shift Register 4094
'-----

$regfile = "m328pdef.dat"
$crystal = 16000000
$baud = 9600

Dim Dat As Byte
Dim D As Byte
Dim N As Byte
Dim B As Bit

Sr Alias Portb.4      '4094 pin 1
Da Alias Portb.3      '4094 pin 2
Cl Alias Portb.2      '4094 pin 3
Config Portb = Output

...

Dat = 0
Do
  Cls
  Lcd Dat
  Lcd "  "
  Print Dat
  D = Dat
  For N = 1 To 8
    B = D.7
    Da = B
    Waitms 1
    Cl = 1
    Waitms 1
    Cl = 0
    Waitms 1
    Shift D , Left
  Next N
  Sr = 1
  Waitms 1
  Sr = 0
  Waitms 1
  Dat = Dat + 1
  Waitms 500
Loop
End
```

facile de comparer le résultat avec les états de sortie du registre.

Si l'on a besoin de plus de huit sorties, on peut utiliser la sortie QS du 4094. Toutes les données introduites apparaissent en QS après 8 impulsions d'horloge. Nous pouvons relier à cette sortie un autre registre à décalage, et même relier de cette façon un nombre quelconque de 4094 les uns à la suite des autres, en câblant en parallèle les lignes *Clock* et *Strobe*. Le code doit bien sûr être modifié en conséquence. On décale d'abord vers la sortie les 16 (voire les 80) bits de données, puis on déclenche l'impulsion commune de transfert.

Transmission manuelle de données

Les données cheminent certes sur une seule ligne, mais une interface série cadencée (comme

ici avec le registre à décalage, ou comme avec un bus SPI) nécessite à tout le moins une ligne d'horloge supplémentaire. Établissons une comparaison avec la télégraphie morse pour mieux le comprendre. En télégraphie, il faut se mettre d'accord sur la vitesse de transmission, et la transmission d'une unité d'information ne doit pas être interrompue : si un télégraphiste commence à envoyer un X (– . . –) et s'arrête pour se gratter le nez, c'est au final un N (–.) et un A (–.) qui seront reçus. Il en va de même avec une interface série asynchrone : la valeur du débit doit être fixée d'avance. Une fois la transmission d'un octet initiée, tous les bits doivent être envoyés à la même cadence. Avec une interface SPI, c'est l'inverse : certains retards sont permis et le facteur temps n'est donc pas critique. Autrement dit,

Listage 2. Pianissimo entre le maître et l'esclave SPI .

```
'-----
'UNO_spi1.BAS  Shift in/out
'-----
$regfile = "m328pdef.dat"
$crystal = 16000000
$baud = 9600

Dim D As Byte
Dim B As Bit
Dim N As Byte

S1 Alias Pinc.0
Portc.0 = 1
S2 Alias Pinc.1
Portc.1 = 1
Led1 Alias Portc.2
Ddrc.2 = 1
Led2 Alias Portb.2
Ddrb.2 = 1

...

Do
  D = Rnd(255)
  Cls
  Lcd D
  Print D
  Locate 2 , 1
  For N = 0 To 7
    B = D.7
    Lcd B
    Print B;
    Led2 = B
    Waitms 300

    Led1 = 1
    Waitms 200
    Led1 = 0
    Waitms 500
    Shift D , Left

  Next N
  Led1 = 0
  Led2 = 0
  Waitms 2000
  Cls
  Print
  D = 0
  Do
    Loop Until S1 = 1
    For N = 0 To 7
      Shift D , Left
      Do
        Loop Until S1 = 0
        If S2 = 0 Then B = 1 Else B = 0
        D = D + B
        Lcd B
        Print B;
        Waitms 100
      Do
        Loop Until S1 = 1
        Waitms 100
    Next N
    Print
    Locate 2 , 1
    Lcd D
    Print D
    Waitms 2000
  Loop
End
```


avec la ligne d'horloge supplémentaire, la transmission ne dépend plus du tout du débit : un bit par minute ou un million de bits par seconde, peu importe, les données seront toujours transmises correctement. Une transmission SPI comporte toujours un maître et un esclave. Les données peuvent cheminer dans les deux directions, mais l'horloge est toujours contrôlée par le maître. Testons le procédé en mode manuel. L'utilisateur est le maître et peut transmettre un octet à l'aide des deux boutons (**fig. 2**). La méthode n'est pas très usuelle, certes, mais aide à bien comprendre ce qu'il se passe. Un des boutons de notre « manipulateur morse » contrôle l'envoi des données, l'autre l'horloge. Pour le reste, rien à vous apprendre, vous savez déjà comment assembler un octet. Voici le principe : nous envoyons d'abord le bit 7. S'il s'agit d'un 1 nous appuyons sur S2, sinon nous ne faisons rien. Nous appuyons ensuite brièvement une fois sur S1 sans lâcher S2. La station réceptrice (l'esclave, ici la carte Arduino) sait alors quand elle doit lire le bit de la ligne. Suivent ensuite les bits 6, 5, etc., jusqu'au bit 0.

Le programme du **listage 2** affiche le flux des données dans les deux sens. Le contrôleur est d'abord le maître et vous êtes l'esclave. Le programme envoie un octet de valeur aléatoire. La LED1 représente le signal d'horloge, la LED2 les données. En vous concentrant un peu, vous devriez pouvoir lire les octets transmis. Je reconnais que l'exercice n'est pas facile, donc la valeur de l'octet est dévoilée sur le LCD et envoyée au terminal. Les bits individuels apparaissent par ailleurs l'un après l'autre sur le LCD et dans le terminal (p. ex. 01010011 pour 83).

Invertissons les rôles : vous êtes maintenant le maître, avec pour tâche de renvoyer l'octet reçu. L'avantage de ne pas être contraint par un débit déterminé devient manifeste : vous avez en effet tout votre temps pour réfléchir à la valeur du prochain bit à transmettre. Supposons que vous deviez envoyer un 100. Le bit 7 a un poids de 128 et n'est donc d'aucune utilité. 0 par conséquent. Le bit 6 a un poids de 64 et vaut donc 1. Reste à trouver 36 pour faire 100. Le bit suivant permet d'envoyer 1×32 . Reste 4. Ce qui donne deux bits à 0 (0×16 et 0×8), un bit à 1 (1×4), et enfin deux 0. Vous avez envoyé le nombre binaire 01100100 en déclenchant pour chaque bit une impulsion d'horloge. Pas évident pour nous, mais le microcontrôleur, lui, lit tout

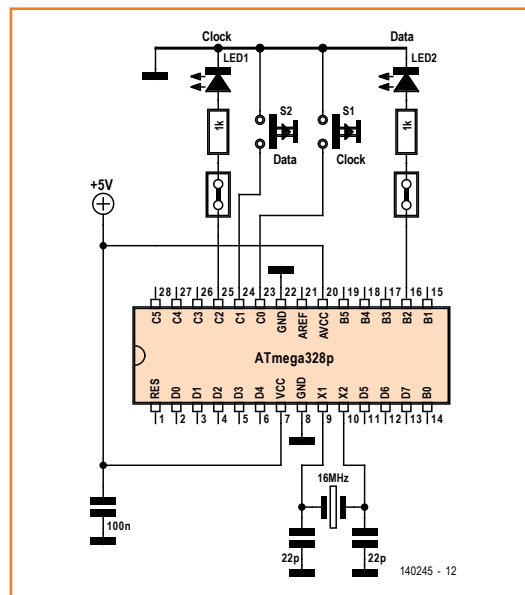


Figure 2.
Le vrai-faux manipulateur morse.

ça comme s'il lisait un album à colorier. Le tout forme une sorte de jeu de concentration dans lequel vous gagnez lorsque le contrôleur affiche le bon résultat sur le LCD.

Si vous regardez attentivement le listage 2, vous remarquerez que les bits lus sont inversés. La raison en est que l'activation d'une touche produit un bit 0. Le résultat est donc inversé au moment de la lecture afin que l'utilisateur n'ait pas à se tordre le cou ou les neurones. Notez également la création d'un nombre pseudo-aléatoire avec $D = \text{Rnd}(255)$. C'est en fait la même séquence de chiffres qui est créée à chaque redémarrage du programme. L'aide de Bascom livre une astuce pour y remédier.

De contrôleur à contrôleur

Voyons maintenant comment utiliser l'interface SPI pour envoyer des données d'un microcontrôleur à un autre. Les données en question seront les 10 bits du convertisseur AN. Autre avantage ici de la liaison SPI, la largeur de données est flexible. Qu'il s'agisse de 8, 10, 12 ou 16 bits, le procédé est identique. Pour relier nos deux microcontrôleurs, il serait à vrai dire plus simple d'utiliser une interface série *asynchrone* avec TXD et RXD, donc notre exemple sert surtout à illustrer le mécanisme de la transmission. L'avantage du bus SPI est plus manifeste lorsqu'il s'agit de commander un matériel externe.

Nous utiliserons ici aussi un troisième fil, à savoir la ligne de sélection de puce /CS (*chip select*).

Listage 3. Le maître SPI.

```

'-----
'UNO_spi2.BAS  SPI Master
'-----
$regfile = "m328pdef.dat"
$crystal = 16000000
$baud = 9600
Dim B As Bit
Dim Dout As Word
Dim N As Byte
Dim I As Byte

Sck Alias Portb.5
Ddrb.5 = 1
Mosi Alias Portb.3
Ddrb.3 = 1
Cs Alias Portb.2
Ddrb.2 = 1

Cs = 1
Mosi = 0
Sck = 0

Config Adc = Single , Prescaler = 32 ,
Reference = Avcc
Start Adc

Cls
Cursor Off

Waitms 200
Do
  Dout = Getadc(3)  'Pot
  Locate 1 , 1
  Lcd Dout
  Lcd "  "
  Cs = 0
  Waitms 20
  For N = 1 To 10
    Mosi = Dout.9
    Waitms 1
    Sck = 1
    Waitms 1
    Sck = 0
    Waitms 1
    Shift Dout , Left
  Next N
  Cs = 1
  Waitms 100
Loop
End

```

Listage 4. L'esclave SPI.

```

'-----
'UNO_spi3.BAS  SPI Slave
'-----
$regfile = "m328pdef.dat"
$crystal = 16000000
$baud = 9600

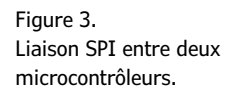
Dim Addr As Byte
Dim B As Bit
Dim Dout As Word
Dim Din As Word
Dim N As Byte
Dim I As Byte

S1 Alias Pinc.0
Portc.0 = 1
S2 Alias Pinc.1
Portc.1 = 1
Sck Alias Pinb.5
Portb.5 = 1
Mosi Alias Pinb.3
Portb.3 = 1
Cs Alias Pinb.2

Portb.2 = 1
...

Do
  Do
    Loop Until Cs = 0
    Din = 0
    For N = 1 To 10
      Shift Din , Left
      Do
        Loop Until Sck = 1
        Din = Din + Mosi
      Do
        Loop Until Sck = 0
    Next N
  Do
    Loop Until Cs = 1
    Locate 1 , 1
    Lcd Din
    Lcd "  "
    Print Din
  Loop
End

```



Du convertisseur AN au pilote d'afficheur en passant par les mémoires, on trouve une ribambelle de composants dotés de connexions SPI. Fré-

quemment utilisées et peu chères, les mémoires EEPROM sérielles en font partie. Avec ses 64 Ko, le CI 25LC512 (à ne pas confondre avec le 24C512 doté d'un bus SPI) tombe à point lorsque le maigre kilooctet de l'EEPROM de l'ATmega328 ne suffit pas. Les EEPROM I²C sont très utilisées dans les projets amateurs ; dans le domaine professionnel, ce sont plutôt les EEPROM SPI qui sont employées pour leur grande fiabilité. Le schéma de la **figure 4** montre comment relier l'EEPROM à l'ATmega. Puisque ce sont là aussi

les connexions SPI de l'interface SPI d'origine qui sont utilisées (connecteur ICSP à 2x3 broches), on peut obtenir un petit module mémoire pratique en soudant le CI et son support sur une petite embase à 6 contacts. Il faut juste câbler une ligne externe supplémentaire, à savoir relier la ligne CS à B2, car la ligne Reset du connecteur ICSP ne peut pas être utilisée.

Nous avons cette fois-ci deux lignes de données. La ligne *MOSI* (*Master Out Slave In*) sert à l'affichage et est reliée à l'entrée série *SI* de l'EEPROM. La ligne *MISO* (*Master In Slave Out*) sert à lire les données de *SO*. Le microcontrôleur est toujours le maître et fournit l'horloge en *SCK*. Cette fonction est confiée à un sous-programme (déclaré par *Sub*, voir l'encadré) qui transmet les données dans les deux sens et en une fois (**listage 5**). Les données doivent être placées dans la variable globale *Dout* avant l'appel du sous-programme ; après son appel, les données lues sont contenues dans la variable *Din*. Il existe bien certaines situations dans lesquelles il y a seulement écriture, mais dans ce cas des bits 0 sont normalement envoyés dans l'autre sens. Même si la fiche technique du 25LC512 n'est pas d'un abord très facile, sa lecture est indispensable pour savoir quand, comment et quelles données transmettre. La puce mémoire reçoit d'abord (après la mise au niveau bas de la ligne CS) de simples « octets de commande » (p. ex. « 3 ») qui correspondent aux actions à exécuter. Pour lire le contenu de la mémoire nous devons ainsi envoyer une commande « 3 », suivi de l'octet de poids fort et de l'octet de poids faible de l'adresse. Il est dès lors possible de lire autant d'octets de données que souhaité en incrémentant automatiquement la valeur de l'adresse (**listage 6**).

Le code affiche l'adresse en cours de lecture ainsi que les octets lus. Une EEPROM neuve ou entièrement effacée renvoie uniquement la valeur 255. Voyons comment y ajouter des données. Pour cela nous avons l'octet de commande « 2 ». Son utilisation nécessite toutefois un préalable : activer l'opération d'écriture avec la commande « 6 » (**listage 7**). On peut vérifier le succès de cette opération en consultant le registre d'état de l'EEPROM à l'aide de la commande « 5 ». Une action ne sera effective que si la ligne CS a été mise au niveau bas avant l'action demandée, puis remise au niveau haut à la fin. Si le registre d'état renvoie la valeur 2, la puce est activée pour le processus d'écriture. Moins simple qu'écrire avec un stylo, d'accord, mais cette façon de faire pro-

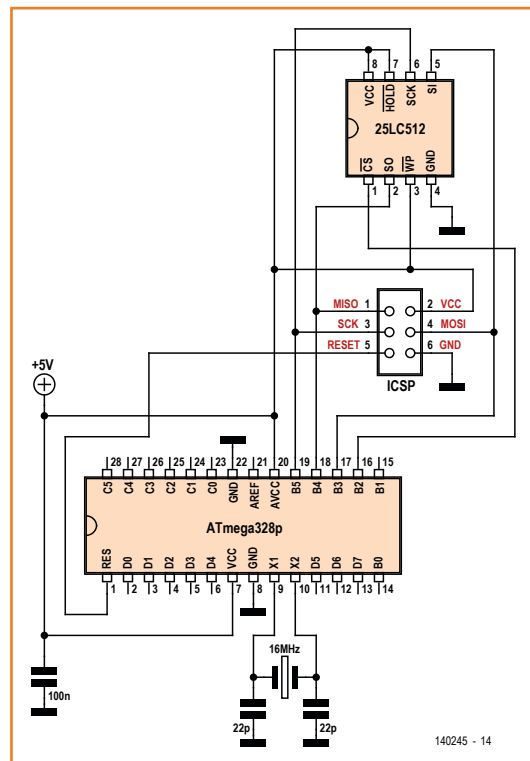


Figure 4.
Schéma de câblage d'une
EEPROM sérielle.

Listage 5. Lecture et écriture via MISO et MOSI.

```
Sub Spioutin
  Din = 0
  For N = 0 To 7
    Shift Din , Left
    Din = Din + Miso
    If Dout.7 = 1 Then Mosi = 1 Else Mosi = 0
    Waitus 3
    Sck = 1
    Waitus 2
    Sck = 0
    Waitus 2
    Shift Dout , Left
  Next N
End Sub
```


Listage 6.**Lecture de l'EEPROM (extrait).**

```

Cs = 0
Dout = 3          'read
Spioutin
Dout = 0          'A8...A15
Spioutin
Dout = 0          'A0...A7
Spioutin
I = 0
Do
  Locate 1 , 1
  Lcd I
  Print I;
  Print " ";
  I = I + 1
  Spioutin
  Locate 2 , 1
  Lcd Din
  Print Din
  Waitms 200
Loop

```

tège contre les erreurs.

La mémoire des EEPROM est divisée en pages, de 128 octets pour ce qui est de la 25LC512. Nous pouvons transférer un maximum de 128 octets jusqu'à la limite de la page suivante. Nous mettons ensuite CS au niveau haut, puis laissons suffisamment de temps à l'EEPROM pour qu'elle puisse procéder à la programmation. D'après la fiche technique, 5 ms suffisent. N'écrasez pas la limite de page, c'est comme si vous écrasiez les pieds du chaos en personne (j'ai testé). Les pages suivantes de la mémoire contiennent

Listage 7. Activation de l'accès en écriture (extrait).

```

Cs = 0
Dout = 6          'write enable
Spioutin
Cs = 1
Waitms 20
Cs = 0
Dout = 5          'read status
Spioutin
Spioutin
Cs = 1
Locate 1 , 1      'status
Lcd Din
Print Din

```

Sous-programme

Un sous-programme, ou procédure selon la doc Bascom, est appelé *Sub* en Bascom. On utilise une procédure chaque fois qu'une même portion de code doit être appelée depuis différents endroits d'un programme. Toute procédure doit être déclarée au début du programme afin que le compilateur la reconnaisse comme telle. La procédure peut alors être appelée comme n'importe quelle fonction Bascom.

```

Declare Sub Spioutin()
...
Dout = 6
Spioutin
...

Sub Spioutin
  Din = 0
...
  Shift Dout , Left
End Sub

```

Il faut toujours être attentif aux variables utilisées par une procédure. La procédure Spioutin n'utilise ici que des variables dites globales, c.-à-d. déclarées au début du programme principal, et par conséquent reconnues partout dans le programme. Une autre façon de faire est de passer directement les variables à la procédure :

```
Declare Sub Spioutin (Byteout as Byte)
```

Dans ce cas la variable Byteout n'est plus globale mais locale, c.-à-d. reconnue uniquement à l'intérieur de la procédure. On utilise alors Spioutin comme ceci :

```
Spioutin 6
```

On gagne une ligne de code par rapport à l'appel précédent.

On peut transmettre plusieurs variables à une procédure, comme avec la procédure Bascom Spiout :

```
Spiout Dout , 1
```

Lorsqu'on débute, on s'expose en général à moins d'erreurs en écrivant des procédures n'utilisant que des variables globales. Trouver la meilleure façon de passer des paramètres prend toute son importance lorsqu'on progresse.

Listage 8. Utilisation de la SPI logicielle (extrait).

```

Config Spi = Soft , Din = Pinb.4 , Dout = Portb.3 ,
Ss = None , Clock = Portb.5 , Mode = 0
Spiinit

Cs Alias Portb.2
Ddrb.2 = 1
Cs = 1

    Cs = 0
    Dout = 4          'write disable
    Spiout Dout , 1
    Cs = 1
    Waitms 1
    Cs = 0
    Dout = 3          'read
    Spiout Dout , 1
    Dout = 0          'A8...A15
    Spiout Dout , 1

Dout = 0          'A0...A7
Spiout Dout , 1
I = 0
Do
    Locate 1 , 1
    Lcd I
    Lcd " "
    Spiin Din , 1
    Locate 2 , 1
    Lcd Din
    Lcd " "
    Print I ;
    Print " ";
    Print Din
    Waitms 200
    I = I + 1
    If I >= 65535 Then Exit Do
Loop

```

d'autres données, disons... celles que vous souhaitez écrire. Le code de l'exemple écrit dans la première page 128 octets de l'adresse 0 à l'adresse 127, puis une suite croissante d'entiers allant de 0 à 127. Et ce sont précisément ces données que l'on retrouvera lors de la lecture. Le code du programme *UNO_spiEE1.bas* peut comme d'habitude être téléchargé depuis le site

d'Elektor [1]. Il exécute, dans l'ordre, les instructions suivantes :

- 6 : validation d'écriture (*write enable*).
- 5 : lecture du registre d'état, affichage durant 1 s.
- 2 : écriture de 128 octets à partir de l'adresse 0.
- 3 : lecture à partir de l'adresse 0 ; boucle sans fin

Listage 9. Enregistreur de données avec routine d'interruption (extrait).

```

Tim0_isr:
    '4000 µs
    Timer0 = 6
    Ticks = Ticks + 1
    If Ticks = 250 Then
        Ticks = 0
        U = Getadc(4)
        U = U / 4
        Addr = Seconds
        Addr = Addr And 127
        If Addr = 0 Then          'start of page
            Cs = 0
            Dout = 6              'write enable
            Spiout Dout , 1
            Cs = 1
            Waitus 100
            Cs = 0
            Dout = 2              'write
            Spiout Dout , 1
            Waitus 100
            Cs = 0

            Dout = 3
            Addr = High(seconds)
            Dout = Addr          'A8...A15
            Spiout Dout , 1
            Addr = Low(seconds)
            Dout = Addr          'A0...A7
            Spiout Dout , 1
        End If
        Dout = U
        Spiout Dout , 1
        Addr = Seconds
        Addr = Addr And 127
        If Addr = 127 Then      'End of page
            Cs = 1
        End If
        Print Seconds
        Seconds = Seconds + 1
        If Seconds = 0 Then Seconds = 65535
    End If
    Return

```

Un enregistreur de données

Utilisons la mémoire de 64 Ko pour construire un enregistreur de données. Dans notre programme, les valeurs de mesure de l'entrée analogique ADC4 sont collectées une fois par seconde et enregistrées. La mémoire est pleine au bout d'environ 18 secondes.

Inutile de tout programmer, Bascom dispose de fonctions adaptées à de nombreuses situations. Ici nous avons le choix entre créer une interface SPI logicielle sur certains ports, et créer une interface matérielle sur les broches du contrôleur prévues à cet effet. L'interface SPI matérielle est particulièrement rapide et est p. ex. souvent utilisée pour la commande des afficheurs graphiques. Sa configuration implique toutefois de nombreux paramètres, en outre assez complexes pour obliger l'utilisateur à décortiquer la fiche technique de l'ATmega328. Utiliser la fonction SPI logicielle est plus simple, et elle permet également un débit relativement élevé. Écrire soi-même une fonction SPI ne manquerait toutefois pas d'intérêt puisque cela permettrait de reprogrammer les chronogrammes des fiches techniques. La SPI logicielle reste cependant plus pratique et plus rapide, c'est donc elle que nous utiliserons. La configuration de l'interface détermine quelles sont les lignes à utiliser. Pour *Din*, *Dout* et *clock*, nous avons besoin des lignes *MISO*, *MOSI* et *SCK* (comme toujours accessibles depuis le connecteur ICSP). Ici nous ne devons pas laisser Bascom commander automatiquement la ligne *SS* (qui correspond à la ligne *CS*), car avec l'EEPROM on souhaite transférer le plus possible de données durant une unique phase *CS* active. C'est donc « à la main » que nous manipulerons cette ligne (port B2). Il existe quatre modes SPI, donc nous ne devons pas oublier le paramètre `Mode = 0`.

Le programme était trop long pour être reproduit entièrement ici. La portion de code du **listage 8** montre l'utilisation de la SPI logicielle pour la lecture du contenu de l'EEPROM. L'instruction `Spiout Dout , 1` envoie un octet (argument « 1 ») dont la valeur est contenue dans la variable *Dout*. L'instruction `Spiin Din , 1` lit un octet, *Din* contenant la valeur lue sur le bus SPI. Le rôle du programme est de lire toutes les données de l'EEPROM et d'afficher leurs valeurs sur le LCD et dans le terminal.

Vous pouvez là aussi télécharger ce programme (*UNO_spiLogger.bas*) depuis [1]. Appuyer sur S1 lance une série de mesures. Vous pouvez interrompre la mesure à n'importe quel moment avec S2 pour lire les données enregistrées.

La temporisation durant la capture des données est faite à l'aide d'une interruption par temporisateur (routine du **listage 9**). La tension sur ADC4 est lue une fois par seconde et enregistrée. Le code tient compte de la taille de 128 octets des blocs de données de l'EEPROM. Un accès en écriture est activé pour chaque début de bloc et l'adresse en cours est transmise. Suivent 128 octets. La ligne *CS* est mise au niveau haut à la fin du bloc de façon à ce que l'EEPROM puisse enregistrer le bloc entier. Comme la ligne *CS* est reliée au port PB2, la LED2 du shield est dans le même temps mise sous tension. Un flash de la LED indique donc l'enregistrement d'un bloc.

(140245 – version française : Hervé Moreau)

Liens

[1] www.elektor-magazine.fr/140245

Les caprices du programmeur Arduino sous Bascom

Vous êtes nombreux à utiliser le chargeur de démarrage Arduino d'origine depuis l'EDI Bascom et à avoir rencontré le problème suivant : si un programme recourt à un affichage série, le programmeur Arduino ne fonctionne plus comme attendu la fois suivante et se bloque. Des lecteurs ont trouvé une solution : utiliser l'EDI Arduino entre deux programmations, p.ex. en chargeant le programme d'exemple *Blink*.

Il y a plus simple : activer brièvement le bouton Reset de la Uno au moins trois fois, en espaçant chaque appui d'environ une seconde, après avoir démarré le programmeur dans Bascom. La programmation se déroule alors sans problème, même si le programme précédent utilisait une instruction *Print*. Les germanophones pourront consulter à ce sujet le fil *BASCOM-AVR-Kurs* du forum allemand (www.elektor-magazine.de/forum).

Ce problème n'apparaît pas si vous utilisez le chargeur *MCS Bootloader* adapté à la Uno. Rappelons que vous pouvez également utiliser un programmeur externe.



14^e jour : l'autorouteur

Neil Gruending
(Canada)

Aujourd'hui nous allons laisser le travail manuel de côté pour nous intéresser aux outils de routage automatique de DesignSpark.

Si vous n'avez jamais utilisé un routeur automatique de pistes ou « autorouteur » avant, voici votre chance. Nous allons commencer par présenter l'autorouteur avant de voir comment le configurer et l'utiliser.

Qu'est-ce qu'un autorouteur ?

Un autorouteur de C.I. peut router tout ou partie de votre carte à l'aide d'un jeu de règles prédéfinies. La plupart des autorouteurs, celui de DS y compris, utilisent une technologie à passes multiples à base de formes qui augmente leur chance de router complètement une carte. Les *autorouteurs à grille* utilisent une grille dont chaque case ne peut contenir qu'une piste ou un objet à la fois, ce qui ne facilite pas le routage de cartes complexes, avec beaucoup de pistes.

Cependant, un *autorouteur à formes* utilise les formes réelles du cuivre sur la carte pour choisir où router les pistes ; cela lui permet de serrer les pistes au plus près. L'autorouteur de DS est également un modèle à passes multiples ce qui veut dire qu'il essaiera de router une carte jusqu'à ce que toutes les erreurs de routage aient disparu ou que le nombre maximal de tentatives autorisées soit dépassé.

L'autorouteur de DesignSpark se configure via la fenêtre *Route All Nets* de la **figure 1** à laquelle on accède par le menu *Tools* → *Auto Route Nets* → *All Nets*. Les autorouteurs essaient toujours de calculer la route la plus facile ou la moins coûteuse pour un signal et l'ajout de règles permet de diriger le calcul de ces routes. Tout comme l'outil de placement automatique de composants, l'autorouteur peut être invoqué de plusieurs manières, mais ne se configure que via le menu *Route All Nets*.

L'une des règles clé définit ce que l'autorouteur doit faire des pistes existantes lorsqu'il démarre. Par défaut, il coupera ou effacera les pistes nécessaires pour router la carte, mais ce n'est que rarement ce que voudrez. Pour éviter ça vous pouvez utiliser les règles *Keep Preroutes* et *Keep Fixed Routes*. Une route fixe est un segment de piste marqué *Fixed for Router* dans ses propriétés ; DS positionne cette propriété automatiquement pour les pistes routées manuellement. La règle *Keep Fixed Routes* oblige l'autorouteur à ne modifier d'aucune façon les pistes appartenant à une route fixe. Une préroute est une piste que l'autorouteur ne touchera pas au début, mais qu'il pourra re-router lors de passes ultérieures. L'activation de la règle *Keep Preroutes* impose à l'autorouteur de conserver ces segments préroulés.

Miter Track est une autre règle importante. Par défaut, l'autorouteur utilise des angles à 90°, mais, si vous activez la règle *Miter Track* (NdT : « angler les pistes »), alors l'autorouteur essaiera, lors de la dernière étape après le routage, de briser le plus d'angles à 90° qu'il pourra.

Les règles *Max Effort* et *Passes* permettent de choisir le temps et l'effort maximaux que l'on autorisera à l'autorouteur. Plus *Max Effort* sera élevé et plus l'autorouteur s'efforcera de finir le routage lors d'une passe ; normalement vous n'augmenterez pas trop cette valeur : l'autorouteur peut habituellement faire mieux en abandonnant la passe et en essayant lors de la suivante. La règle *Passes* représente le nombre maximal d'essais que l'autorouteur est autorisé à faire pour router la carte. Lorsque ce paramètre est réglé à plus de un, l'autorouteur s'autorisera des

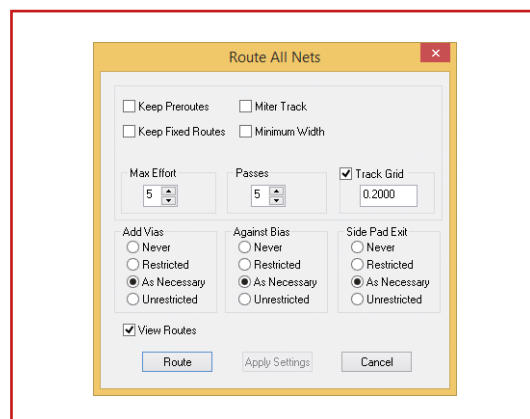


Figure 1. La fenêtre *Route All Nets*.

erreurs telles que les recouvrement de pistes qu'il essaiera de réparer lors des passes suivantes. La règle *Against Bias* fera en sorte que l'auto-routeur suive ou non les règles de préférence directionnelle pour le routage de la carte. Cette préférence se règle pour chaque couche depuis l'onglet *Layers* de la fenêtre *Design Technology* ; les valeurs autorisées sont X, Y ou None. Une préférence en X poussera l'auto-routeur à préférer le routage horizontal des pistes ; une préférence en Y, à l'inverse, favorisera le routage vertical. Le plus souvent, vous ne spécifierez pas de préférences : cela donne à l'auto-routeur le plus de flexibilité.

Je vous recommande d'activer la règle *Track Grid* et de sélectionner la grille de routage que vous désirez. L'auto-routeur n'a pas besoin de la grille, mais elle vous facilitera grandement le fig-nolage manuel. Je vous recommande également de sélectionner *As Necessary* pour les règles *Add Vias* et *Side Pad Exit*.

Utilisation de l'auto-routeur

Il est temps d'examiner les différentes manières d'utiliser l'auto-routeur de DesignSpark. Pour router une carte entière, utilisez *Tools* → *Auto Route Nets* → *All Nets*. La **figure 2** montre le résultat du routage sur notre pilote de LED des articles précédents. Il y a un peu de fig-nolage à faire, mais c'est normal pour une carte routée automatiquement. Comment l'auto-routeur a-t-il su quel type de pistes et de vias utiliser pour le routage ? Par défaut, l'auto-routeur décidera tout seul, mais vous pouvez passer outre en utilisant des classes de signaux (NdT : *net classes*) (**fig. 3**). DS utilise les classes de signaux pour créer des groupes de signaux qui partageront les mêmes paramètres de routage. Par défaut, DS créera un groupe, nommé *signal*, dans lequel il placera tous vos signaux, mais vous pouvez créer autant de groupes que vous le voulez. Les groupes de signaux sont créés et édités depuis l'onglet *Net Classes* et l'association des signaux aux groupes se fait depuis l'onglet *Nets*. J'ai choisi de régler le groupe *signal* pour utiliser le style de pistes *Signal* et le type de vias *SignalVia*.

DS possède également d'autres modes de routage automatique et peut, par exemple, router un signal ou un groupe de signaux. Mais, je pense que la fonction la plus utile de l'auto-routeur est sans doute sa capacité à déployer toutes les connexions d'alimentation d'un composant à

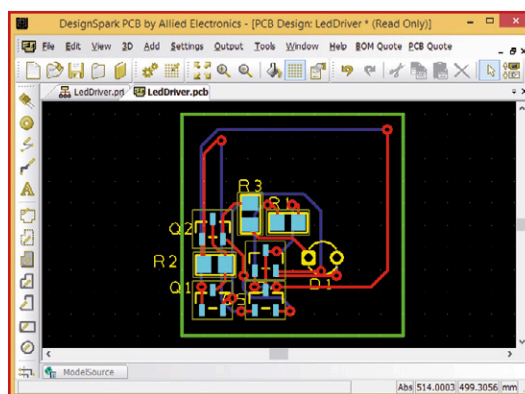


Figure 2.
Routage de notre carte de pilote de LED.

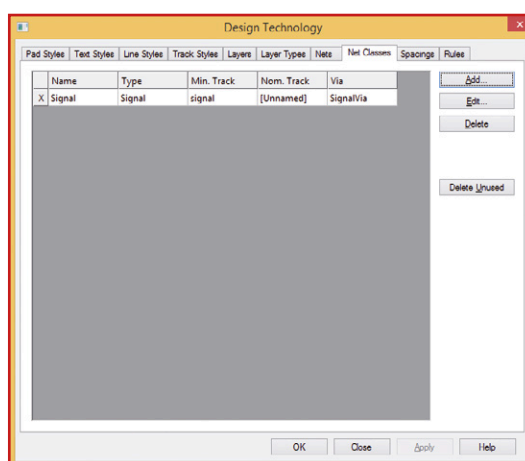


Figure 3.
La fenêtre *Net Class*.

l'aide de courtes pistes ou de vias si votre carte est dotée de plans d'alimentation. Tout ce que vous avez à faire, c'est demander à l'auto-routeur de ne router que les signaux des plans ; vous allez économiser pas mal de temps. Vous pouvez aussi utiliser l'auto-routeur de manière interactive en sélectionnant les signaux ou les composants sur la carte puis en cliquant dessus avec le bouton droit et enfin en choisissant *Auto Route*. L'auto-routeur ne s'occupera que des éléments sélectionnés.

Conclusion

Nous venons de faire un tour d'horizon des outils de routage automatique de DesignSpark qui vous feront sans aucun doute économiser du temps. Habituellement, je préfère router mes cartes complètement à la main, mais l'auto-routeur de DesignSpark est suffisamment flexible pour se charger de la majorité de ce fastidieux travail. La prochaine fois, nous parlerons de quelques autres fonctions pour C.I. de DesignSpark.

(140242 – version française : Kévin Petit)

BK Precision BK560

testeur de CI programmable

la grande époque des TTL & CMOS



Pascal Rondane
(Tours)
p.rondane@gmail.com

Comme *Hewlett Packard*, *Apple* et nombre de grandes entreprises, B & K a commencé dans un garage, aux États-Unis. En 1948, alors que les Américains moyens commençaient à acheter des téléviseurs en grand nombre, l'entrepreneur Carl Korn et son partenaire Philippe Ban, frustrés par le manque de matériel pour tester les composants électroniques de ces téléviseurs, décidèrent de créer, à Chicago, leurs propres appareils de mesures. C'est ainsi qu'est née la *Central Television Service Company*, une entreprise bientôt prospère, de vente de régénérateurs de tubes cathodiques et de testeurs de tubes.

En 1951, la société, rebaptisée *B & K Precision*, étend ses activités dans d'autres domaines : le test et la mesure électroniques. En 1961, *B&K*

Precision devient une succursale de *DynaScan*, sans cesser la fabrication d'appareils de test et de mesure de précision qu'elle poursuivra pendant soixante ans.

Mon BK560

C'est en 2000, lors de la fermeture de la société où je travaillais, que j'ai racheté l'appareil décrit dans cet article ainsi qu'un transistor-mètre BK502B ; nous l'avons utilisé pendant des années pour la maintenance de produits IBM (imprimantes, PC) et Motorola, et pour des cartes industrielles.

Tous ces produits utilisaient des microprocesseurs (Z80, 8052, 68000) mais aussi énormément de logique TTL. Dans ce contexte, cet appareil prenait tout son sens, car il permettait de tester un grand nombre de circuits intégrés (TTL, CMOS). Deux possibilités s'offraient à l'utilisateur : soit le test sur le support intégré à l'appareil soit *in situ* au moyen d'une sonde directement connectée sur les circuits intégrés soudés sur le circuit imprimé. Lorsque le BK560 était en marche, plus question d'écouter la radio sur la bande FM. On parlait peut-être déjà de compatibilité électromagné-

EST^D 2004

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

tique en ce temps-là, mais on était encore loin d'appliquer des normes CEM ! On retrouve un peu de l'ambiance si particulière des années 1980 avec la pub de la **figure 1**.

Fonctionnement

Le BK560 dans la version que je possède contient en mémoire une bibliothèque de 1500 composants, évolutive, ce qui permettait d'ajouter de nouveaux composants. Il teste les compteurs et les bascules, mais pas les monostables avec des composants RC externes ni des amplis opérationnels.

Le test *In circuit* ne peut pas non plus être réalisé sur des circuits dont la sortie est directement reliée à l'entrée (comme c'est souvent le cas des bascules), ni sur les circuits reliés à une horloge. Il est possible de contourner ce problème en déconnectant les horloges des entrées du composant et en utilisant la méthode *Learn* (décrite ci-dessous).

Deux modes de fonctionnement sont possibles :

Le test *hors circuit* : on saisit au clavier la référence du circuit intégré, que l'on insère sur le support de test et on lance la procédure ; l'afficheur indique si le circuit est bon ou, dans le cas contraire, le numéro des broches défectueuses.

Le test *In circuit* peut être décliné en deux modes :

test In CKT : on débranche le(s) circuit(s) d'horloge, on alimente la platine en test, puis, après sélection de la référence du circuit intégré, on branche la sonde sur celui-ci et on lance le test.

test Learn : on débranche le(s) circuit(s) d'horloge, on alimente la platine de référence, on renseigne le numéro du CI (référence sur le circuit imprimé ou sur le schéma) et on appuie sur la touche « TEST » ; l'appareil mémorise le résultat pour une comparaison ultérieure.

Cette approche permet, à partir d'une carte de référence, de créer une séquence de test automatique pour ce type de carte. Le constructeur souligne qu'avec cette méthode, l'appareil peut être utilisé par des personnes non qualifiées...

Dans ce mode, lors de la coupure de l'appareil, les informations stockées en RAM étaient perdues. L'option des mémoires EEPROM (2 K x8) faciles à insérer dans les supports prévus en face avant (**figure 2**) permettait de sauvegarder la configuration mémorisée.



Sous le capot

Je ne possède pas le schéma, mais on le trouve facilement (35 \$) [2]. À l'ouverture du BK560, on découvre un empilement de trois cartes double face (**fig. 3**) qui regroupe l'afficheur en matrice de 20 caractères fluorescents verts, un microcontrôleur 8031 et 4 EPROM 2764 plus un support libre pour une probable extension.

La plupart des connecteurs sont scellés par un point de colle. On notera deux détails intéressants : une reprise pour le moins curieuse avec du fil en nappe et un connecteur HE10 (**fig. 4**), la raison la plus probable étant une simplification du routage... Remarquez l'astucieux montage de charnières permettant de basculer les PCB...

Sur la gauche, on trouve un circuit imprimé avec un support ZIF à force d'insertion nulle pour le test « in situ », un connecteur HE10 pour la limande de test et les supports pour les EEPROM d'extension.

1



2

L'alimentation est confiée à un transformateur multi tensions (100 V à 240 V) et à des régulateurs linéaires, méthode courante à cette époque.

Je n'ai pas réussi à établir avec certitude si cet appareil a été fabriqué aux États-Unis, comme tout porte à le laisser croire.

Test 1 – 2 – 24

La documentation technique mentionne 24 lignes d'E/S bidirectionnelles à 3 états pour y raccorder les broches du circuit intégré à tester. Pour les circuits TTL, la tension d'alimentation V_{CC} étant de 5,0 V, le seuil de 2,0 V définit le niveau logique

haut et le seuil de 0,8 V le seuil du niveau logique bas. Pour les CMOS, la tension d'alimentation V_{DD} (c'est bien comme ça qu'on l'indique pour ce type de circuits) est comprise entre 5 V et 15 V : $0,7 \cdot V_{DD}$ définit le seuil haut et $0,3 \cdot V_{DD}$ le seuil bas.

Sur la face arrière, on remarque une prise DB25 pour une liaison RS232. Le BK560 était livré (en option ?) avec une disquette contenant le logiciel, un câble de liaison, le mode d'emploi ainsi qu'une EEPROM de mise à jour.

Le mode d'emploi préconise un IBM PC ou compatible de 256 Ko de mémoire, un lecteur de disquettes 5 pouces 1/4 et un système DOS 2.00 ou ultérieur. Pas de doute, nous sommes dans les années 80.

Le programme permet de créer des nouvelles procédures de test pour les CI. Celles-ci sont ensuite chargées dans le BK560 pour rester en bibliothèque de façon permanente en EEPROM. Le logiciel permet de définir le nombre de broches et le type de circuit intégré TTL ou CMOS ; si celui-ci est à sorties à collecteur ouvert, une résistance de charge de 10 k Ω est insérée.

Un test séquentiel permet de tester le circuit intégré par éléments séparés (7400 etc.), et une fonction de remise à zéro est disponible pour les compteurs, bascules, etc... Je n'ai pas souvenir d'avoir jamais utilisé cette fonction de l'appareil.

Conclusion

Cet appareil me sert encore quelquefois pour le dépannage (de plus en plus rare) de cartes remplies de CI logiques. Il a rendu de très bons services ; les rares problèmes rencontrés étaient liés à l'emploi de la sonde de test externe, à de mauvais contacts dus au vernis ou aux remontrées de soudures sur du circuit imprimé double face et à l'usure de la sonde elle-même.

Je possède tous les documents d'origine en anglais et en français, à l'exception des schémas. Si vous êtes intéressé, n'hésitez pas à me contacter.

(140051)

Liens

[1] www.bkprecision.com

[2] BK560 manuals: www.bkmanuals.com

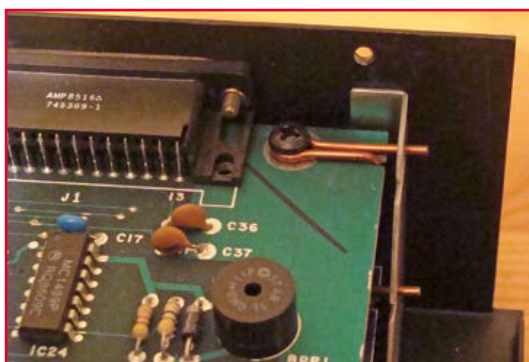
3



4



5



hexadoku gymnastique cérébrale

Assez de schémas, de circuits, de programmes, de soudage et de dessoudage, posez le fer à souder, rangez la pince coupante, ouvrez la fenêtre pour évacuer les vapeurs méphitiques. C'est l'heure de votre gymnastique neuronale. Remplissez la grille selon les règles, envoyez-nous votre solution. Vous gagnerez peut-être un cadeau d'une valeur de 50 €.

Après ça, vous serez d'attaque pour reprendre vos cogitations électroniques.

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

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

Participez et gagnez !

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

Où envoyer ?

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

Les gagnants

La solution de la grille du numéro de juillet/août (433-434) est : **CE234**

Les cinq bons Elektor d'une valeur de **50 €** vont à : **Ciril Zalokar** (Slovénie) ; **Jacek Butowski** (Pologne) ;

Pascal Schmitz (Allemagne) ; **Chris Smith** (Australie) ; **Brian Wood** (Royaume-Uni).

Bravo à tous les participants et félicitations aux gagnants !

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

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

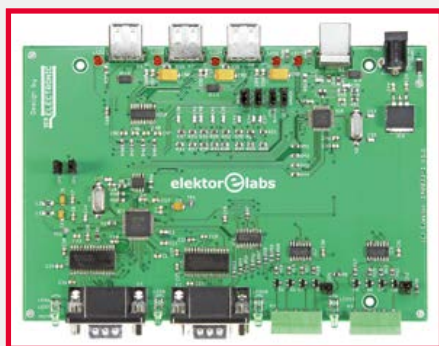
Tout recours est exclu de même que le sont, de ce jeu, les personnels d'Elektor International Media et leur famille. Un seul gagnant par foyer.

•bientôt dans Elektor



**alimentation de
labo universelle
40 V/2 A**

En électronique, les alimentations de labo sont de toute évidence un produit de grande consommation, comme les pommes de terre avec lesquelles elles ont en commun de ne se ressembler qu'en apparence. Ce nouveau projet d'elektor.labs est commandé par un μC , et combine (avec discernement !) techniques linéaires et découpage.



**convertisseur
USB \leftrightarrow RS232 -
RS422/485**

Vous êtes nombreux à attendre un tel circuit, car vous avez dans votre arsenal des appareils en parfait état de marche avec une interface RS-232 ou RS-422/485 mais rien de tel sur vos ordinateurs tous fanatiquement convertis à l'USB. Certains apprécieront aussi l'absence de microcontrôleur sur ce convertisseur qui fait aussi concentrateur USB.



**modules
de logiciel C**

Page 42 du numéro 433/434 de juillet-août, nous présentions sous le titre « mon premier shield » une extension d'expérimentation pour vous encourager à varier les plaisirs avec l'Arduino. Voici à présent les briques de logiciel en C qui vont avec. Chaque sous-ensemble matériel se voit attribuer un bloc de code. Après c'est comme de jouer au Lego.

Informations non contractuelles
Parution du numéro de novembre : 21 octobre

Publicité





**Designer
de faces
gratuit**

CONCEVEZ – NOUS PRODUISONS

Des plaques de qualité professionnelle

Dès une pièce et pour un prix modéré !
Téléchargez notre Designer de faces avant gratuitement sous www.schaeffer-ag.de, concevez votre plaque puis commandez-la directement.

www.schaeffer-ag.de

ECD7

NOUVELLE EDITION

Base de composants d'ELEKTOR

Cet ensemble consiste en une quadruple banque de données (circuits intégrés, transistors, diodes et optocoupleurs) complétée par neuf applications satellites, au nombre desquelles on trouvera notamment de quoi calculer la valeur de la résistance associée à une diode zener, à un régulateur, à un diviseur, ou un multivibrateur astable, mais aussi le code de couleur de la résistance et de l'inductance. Avec ce CD-ROM, vous disposez donc de données fiables sur



plus de 7.800 circuits intégrés ; plus de 35.600 transistors, FET, thyristors et triacs ; environ 25.000 diodes et plus de 1.800 optocoupleurs. Le clou, c'est que vous allez pouvoir rajouter dans la base de données ce qui y manque encore, car elle est interactive ! Ainsi chaque utilisateur pourra lui-même rajouter des composants, en modifier les caractéristiques déjà enregistrées ou les compléter.

ISBN 978-90-5381-298-3 • 29,50 €



Pour commander en ligne :
www.elektor.fr/ecd7

VENEZ VERIFIER PAR
VOUS MEME A
ELECTRONICA MUNICH,
11-14 NOVEMBRE 2014,
SALLE A1, STAND 115

pico[®]
Technology

OSCILLOSCOPES A SIGNAUX MIXTES

4 VOIES ANALOGIQUES + 16 LOGIQUES

DEBUG RAPIDE DE DESIGNS COMPLEXES MIXTES

- USB 3.0
- MEMOIRE ULTRA LARGE
- MEMOIRE SEGMENTEE
- DECLENCHEMENTS RAPIDES



INCLUS: MESURES AUTOMATIQUES, ANALYSE DE SPECTRE, PROGRAMMATION, DECLENCHEMENTS SOPHISTIQUES, MODE PERSISTANCE COULEUR, DECODAGE SERIE (CAN, LIN, RS232, I²C, I²S, FLEXRAY, SPI), MASQUES, MATHS, TOUT EST EN STANDARD, MISES A JOUR GRATUITES, GARANTIE 5 ANS



	3204D MSO	3205D MSO	3206D MSO	3404D MSO	3405D MSO	3406D MSO
Voies	2 ANALOGIQUES, 16 NUMERIQUES			2 ANALOGIQUES, 16 NUMERIQUES		
Bande Passante	60 MHz	100 MHz	200 MHz	60 MHz	100 MHz	200 MHz
Mémoire d'Acquisition	128 Mé	256 Mé	512 Mé	128 Mé	256 Mé	512 Mé
Vitesse Echant. Max.	1 GS/s					
Générateur de Signal	Générateur de Fonction + Générateur de Fonction Arbitraire					
Entrées Numériques	Fréq. Max. 100 MHz, 500Mé/s d'échantillonnage max.					

www.picotech.com/PS365

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