

Radio logicielle : nouvelle série d'articles

eLektor

www.elektor.fr

radio logicielle
générateur de signal
ultra précis avec ATtiny



Android + Bluetooth + Arduino

pour télécommander vos circuits à microcontrôleurs depuis un téléphone

Récepteur Petites Ondes

pour renouer avec les plaisirs élémentaires de l'électronique

Platino dans la peau d'Arduino

pour réussir l'intégration de votre propre matériel
dans le monde des Arduino

L 19624 -405- F: 7,20 €



Modules et platines Arduino™



Plate-formes microcontrôlées "open-source" programmables via un langage proche du "C". Fonctionnent de façon autonome ou en communiquant avec un logiciel sur ordinateur.

A partir de 4,78 €

Analyseurs logiques 4 à 32 voies



Raccordement sur PC via bus USB. Nombreuses possibilités de décodage de signaux au format I2C™, SPI™, UART, CAN 2.0B, USB 1.1, 1 Wire™, DMX512, IRDA, LIN, MODBUS...

A partir de 59 €

Oscilloscopes numériques



Modèles 2 voies 25 à 200 MHz - Ecran couleur 640 x 480 pixels - Sortie USB - Nombreuses fonctions intégrées - Avec ou sans analyseur logique 16 voies intégré.

A partir de 437 €

Logiciels de C.A.O



Ergonomiques et très intuitifs - Saisie de schéma - Logiciels de simulation - Conception de prototype - Conception de circuit imprimés - Réalisation de face avant, etc...

A partir de 24 €

Modules ZigBee™



Modèles pilotables via commandes AT ou entièrement programmables pour un fonctionnement autonome. Kits de développement et platines d'évaluation disponibles

A partir de 20 €

Cordon d'interface USB <> GPIB



Fiable, performant et économique. Permet le pilotage d'un équipement GPIB ou le rattachement de copies d'écran via un logiciel d'émulation de traceur open source.

A partir de 179 €

Modules CUBLOC et PICBASIC



Plate-formes microcontrôlées ultra performantes utilisables via une programmation en langage Basic évolué. Documentation entièrement en Français. Prise en main immédiate.

A partir de 19 €

Boîtiers d'interface LabJack



Interfaces professionnelles pour PC permettant de disposer d'entrées/sorties tout ou rien et d'entrées de conversion analogique/ numérique via un port USB ou Ethernet.

A partir de 109 €

Modules mbed et LPCXpresso



Plate-formes microcontrôlées sur cœur ARM™ 32 bits à architecture Cortex-M3. Nombreux périphériques intégrés: I2C™, SPI™, UART, CAN, ADC, PWM, DAC, USB, Ethernet...

A partir de 24 €

Afficheurs graphiques 4D Systems



Afficheurs graphiques couleurs TFT ou OLED avec ou sans dalle tactile, pilotables via des commandes ASCII ou programmables pour fonctionner en mode autonome.

A partir de 28 €

Plate-forme FOXG20



Système microcontrôlé avec Linux embarqué. Cœur ARM9™ Atmel™ AT91SAM9G20. Nombreux périphériques intégrés: I2C™, SPI™, UART, PWM, USB, Ethernet...

A partir de 167 €

Module CMUcam3



Plate-forme sur base LPC2106 avec caméra vidéo pour acquisition et traitement numérique permettant la reconnaissance de couleurs et le suivi d'objet en temps réel.

A partir de 150 €

Kits d'évaluation FPGA



Kits d'évaluation, cordons JTAG et platines de test pour développement sur FPGA Xilinx™ Spartan-3™, Spartan-6™, Virtex-II™, Virtex-5™. Tarif spécial éducation national.

A partir de 71 €

Modules de restitutions sonores



Modules et platines permettant l'enregistrement et la restitution de fichiers sonores mono ou stéréo. Pilotage via entrées logiques ou port série. Dispo avec ou sans ampli. audio.

A partir de 12 €

Programmateurs de composants



Modèles économiques et modèles professionnels capables de supporter plus de 57830 composants. Avec supports ZIF ou câble ISP. Nombreux adaptateurs optionnels.

A partir de 16 €

Modules Bluetooth™



Dispos sous la forme de clef USB ou série ou de modules OEM avec antenne intégrée ou sortie pour antenne externe. Pour transmission de données ou de signaux audios.

A partir de 26 €

Modules FEZ / GHI electronics



Plate-formes microcontrôlées sur base ARM™ programmables sous environnement .NET Micro Framework™. Gestion I2C™, CAN, USB, Ethernet, UART, One Wire™...

A partir de 37 €

Interfaces CAN



Petits modules d'interfaces CAN <> USB ou CAN <> Série vous permettant de piloter des dispositifs CAN depuis un PC via l'échange de données au format ASCII.

A partir de 96 €

Kits d'évaluation Mikroelektronika



Kits d'évaluations, programmeur/debugger et compilateurs pour microcontrôleurs PIC / dsPIC30/33 / PIC24 / PIC32MX / AVR / 8051 / PSoC / ARM et bases GPS / GSM.

A partir de 32 €

Modules et TAG RFID



Tags, platines de test et modules OEM pour développement d'applications RFID techno. 125 KHz ou 13,56 MHz Unique™, Mifare™, Hitag™, I-Code SLI™, Q5™, etc...

A partir de 2 €

Modules GSM / GPRS



Terminal en boîtier prêt à l'emploi et modules OEM pour transmission GSM / GPRS. Utilisation simple via commandes AT. Kit d'évaluation, platines de test et antennes GSM.

A partir de 44 €

Module de reconnaissance vocale



Modules OEM prêt à l'emploi et modules OEM microcontrôlés programmables permettant le développement d'applications de reconnaissance vocale mono ou multi-locuteurs

A partir de 47 €

Interfaces TCP/IP <> Série



Modèles en boîtier prêt à l'emploi ou sous la forme de modules OEM permettant de disposer d'une conversion transparente de type TCP/IP <> série ou WLAN <> série

A partir de 21 €

Télécommandes radio



Modèles 1 à 4 canaux à modulation AM ou FM. Portée 10 à 300 m. Codage par dip ou en mode anti-scanner. Récepteurs à sorties relais configurables en mode M/A ou impuls.

A partir de 49 €

Boussoles électroniques



Boussoles électroniques compensées ou non en inclinaison avec liaison USB, RS232, série (niveau TTL), I2C™ ou PWM. Nombreuses applications pour robotique ludique.

A partir de 38 €

Modules radiofréquences FM



Modules OEM émetteurs, récepteurs, transceivers, modems radio en bande 433 MHz et 868 MHz. Type large bande ou bande étroite. Mono fréquence ou multi-canaux.

A partir de 9,57 €

PC industriel au format rain-din



PC industriel au format rail-din pour environnements sévères. Base Atom™ Intel™ Z530P. Faible consommation. Faible dissipation thermique. Très nombreux périphériques intégrés.

A partir de 693 €

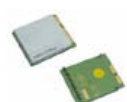
Modules GPS



GPS en boîtier à sortie RS232 ou USB. Modules OEM avec antenne intégrée ou entrée pour antenne externe. Kit d'évaluation, data-logger, antennes amplifiées.

A partir de 39 €

Modules de transmission vidéo



Modules radio multi-canaux OEM pour transmission vidéo et audio (stéréo) en bande 2,4 Ghz ou 5,8 Ghz. Antennes omnidirectionnelles, directives et antennes patch.

A partir de 15 €

Serveurs Web



Modules OEM et boîtiers prêts à l'emploi intégrant un serveur web paramétrable permettant de piloter des sorties, de lire l'état logique d'entrées et la valeur d'entrées A/N.

A partir de 53 €

Modems radio longue portée



Modems radio longue portée (1 à 5 Km) en bande 868 Mhz pour transmission de données (RS232 / RS485), transmission de signaux tout ou rien, signaux analogiques (4-20 mA).

A partir de 234 €

Capteurs divers



Sélection inégalée de capteurs: infrarouge, ultrason, magnétique, accéléromètre, gyroscope, pluie, niveau d'eau, humidité, barométrique, force, température, potentiométrique

A partir de 3,23 €

elektor sur papier

le plaisir de la lecture classique



elektor à l'écran

le plaisir de la lecture sur les nouveaux
supports : PC, portable ou tablette



elektor PLUS

le plaisir ultime de lire partout

**Profitez plus de votre magazine grâce
à la formule avantageuse elektor PLUS !**

Abonnez-vous ou changez de formule maintenant : www.elektor.fr/abo

Ha ha ha !

... c'est sympa de lâcher cinq minutes votre tablette tactile et/ou votre super téléphone pour lire Elektor sur papier, mais je doute que l'édito en soit la partie la plus intéressante. Encore que, on s'y amuse parfois... J'ai reçu de la Ville de Paris une invitation à participer à une campagne de sensibilisation sur le thème :

« Abus d'écrans chez les adolescents ? » à destination, je cite, « des professionnels de la prévention, des parents et des adolescents ». Je ne suis plus adolescent, mais l'abus d'écran, je connais. Sujet bateau. Écoutez ce pathétique appel : « Les écrans étant omniprésents dans notre quotidien... gna gna gna... phénomène de société... gna gna... réponses éducatives appropriées... gna gna gna ». Paradoxe campagne appuyée sur « une série de 5 petits films (5 mn) proposée dans un DVD » : pour parler d'abus d'écrans, on ne trouve donc rien de mieux que de le faire sur des écrans.

Or tout le monde sait – mais c'est tellement plus confortable quand on l'oublie – qu'un écran, comme son nom l'indique, est fait pour masquer. Même « intelligent », il cache au moins autant qu'il montre.

Conformément à sa vocation, Elektor s'obstine à révéler ce qu'ils cachent, en aplanissant le chemin pour aller voir derrière, et découvrir ce qu'on peut tirer soi-même de ces écrans et de leur tentante tactilité. Ce numéro est placé sous le signe du quadruple **A** dont on nous serine qu'il sanctionne des valeurs sûres : Android, AndroPOD, Arduino. Pour Android, Elektor n'est pas à la traîne, mais pour l'Arduino, longtemps considéré à tort dans la maison comme un jouet pour artistes, pas sérieux donc, il était temps qu'on y vienne. Alors on met les bouchées doubles. Et pour l'AndroPOD, l'aventure vient seulement de commencer.

Vous craignez que j'aie oublié le 4^e **A** ? Celui de l'AVR peut-être ?

Non, je pensais à une valeur plus sûre encore, et même inépuisable : l'**avenir**.

Denis Meyer

6 De vous à nous : Le réseau Elektor

Informations légales

8 De vous à nous : courrier des lecteurs

Retour étonnant sur les LED fantômes déjà évoqués en sept. et en nov. 2011.

10 Infos & actualité

Nouveaux produits

12 2,4 GHz : porteuse polymorphe

Wi-Fi, Bluetooth, four à micro-ondes... nous utilisons tous les jours sans y penser des appareils qui tutoient des fréquences de l'ordre de 2,4 GHz.

18 Android + Bluetooth + Arduino

Un modèle bon marché de téléphone Android peut faire une somptueuse télécommande pour un circuit à micro-contrôleur. Il suffit d'une platine Arduino et d'une application Bluetooth.

24 Retour aux sources (3)

Le comportement de votre transistor est inscrit dans les courbes caractéristiques, mais quoi de plus formateur que de mettre la main à la... patte ?

30 Indicateur de charge pour génératrice de secours

Aucun réseau électrique n'est à l'abri d'un aléa météorologique, aucune génératrice de secours n'est à l'abri d'une surcharge.

34 Radio logicielle avec AVR (1)

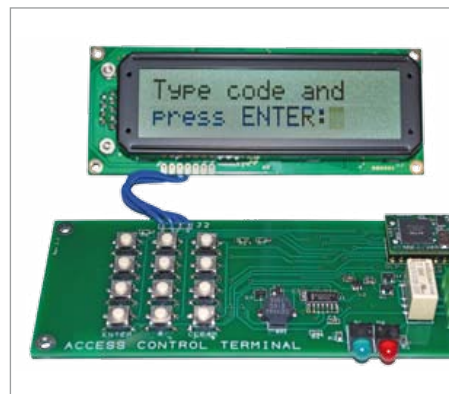
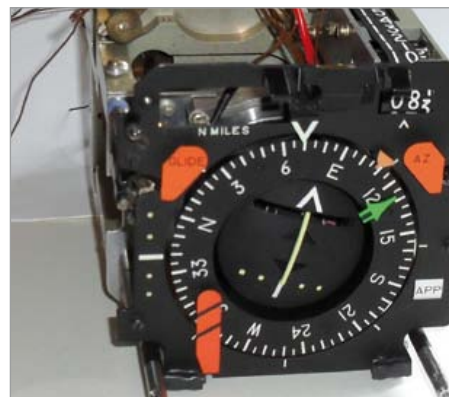
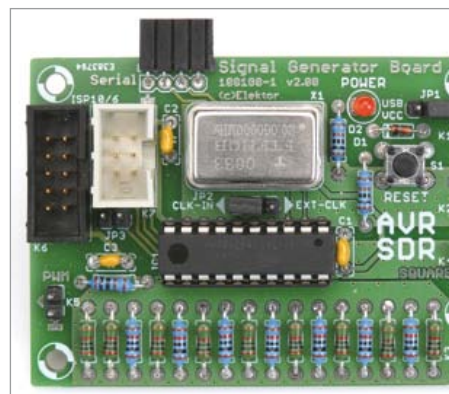
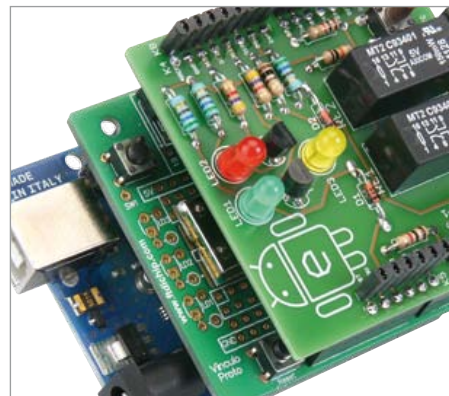
L'ATtiny en générateur de signal ultra précis pour SDR. Ceci est le premier article d'une série consacrée à la SDR.

42 Protégeons l'USB

L'interface USB est robuste, mais il y a des précautions à prendre, notamment en matière de limitation du courant sur la ligne d'alimentation +5 V.

43 LABOratoire

- Le cavalier à trois pattes de l'AndroPOD
- Alim. découpl. ch. filt. DC
- Histoire(s) de prises
- Polarité des LED CMS
- Un câble pour le bus



SOMMAIRE

35^e année
Mars 2012
N° 405



18 Android + Bluetooth + Arduino

Un téléphone pour télécommander vos circuits à microcontrôleurs

Même un modèle bon marché de téléphone Android peut faire la plus somptueuse interface ou télécommande pour un circuit à microcontrôleur. Découvrez ici comment, à l'aide d'une platine Arduino et d'une application Bluetooth, réaliser quelques élégants systèmes, sans fil, de commutation et de mesure. Puis nous programmerons des applications Android et nous découvrirons les indispensables logiciels gratuits pour PC.



34 Radio logicielle avec AVR (1)

Les contrôleurs AVR d'Atmel sont très demandés, notamment parce que leurs outils de développement sont gratuits. Ces processeurs conviennent aussi pour le traitement du signal numérique, nous allons le voir dans cette série d'articles. Même les débutants suivront sans peine, l'information de base y est expliquée simplement. Fidèle à la tradition elektorienne, l'expérimental sera au cœur de l'action. Pour le matériel à construire soi-même, Elektor pense à votre confort et vous en propose une platine. Quant au logiciel, comme toujours, le code source est disponible sur notre site.



62 Cap sur l'avionique : Recyclons un instrument de vol

Comme vous peut-être, l'auteur passe ses journées dans l'électronique, mais les merveilles mécaniques le fascinent aussi. Dans un entrepôt de surplus il est tombé sur un *Horizontal Situation Indicator*, fringant instrument d'avionique, et n'a pas résisté à ce chef-d'œuvre d'électromécanique : il l'a dépiauté, examiné sous toutes les coutures et analysé sa construction et ses fonctions. Embarquement immédiat, décollage vertical.



75 Attaque bit à bit du module USB FTDI

Tirer profit de propriétés peu connues des circuits intégrés USB de FTDI

Cet article décrit la conception électrique et les conditions logicielles d'une serrure électronique sans clé, composée d'un pavé numérique, d'un affichage à cristaux liquides, des contacts d'un relais pour déverrouiller la porte et d'une interface USB. Même si cet article plonge dans les entrailles du circuit intégré FT232RL de FTDI et son mode bit à bit, la technique est compréhensible sans connaissance approfondie de l'USB ni le recours à un microcontrôleur !

47 Astuces de conception

- Adaptateurs DIP pour programmeurs in situ
- Un thermomètre relatif

48 récepteur Petites Ondes à faible consommation

Renouez avec la réception et la démodulation des ondes radio. Après les Grandes Ondes, voici les Petites Ondes.

54 AndroPOD (2) Interface utilisateur : la souplesse HTML

Marier vos circuits électroniques avec un smartphone ? Ils seront heureux et ils auront beaucoup d'enfants.

62 Cap sur l'avionique : Recyclons un instrument de vol

Un Horizontal Situation Indicator, fringant instrument d'avionique, est un chef-d'œuvre d'électromécanique.

68 Platino dans la peau d'Arduino

Platino est compatible avec Arduino, mais Arduino n'est pas entièrement compatible avec (tous les microcontrôleurs qu'accepte) Platino.

74 Concours RS DesignSpark ChipKIT

Des projets inattendus et novateurs...

75 Attaque bit à bit du module USB FTDI

Conception électrique et logiciel d'une serrure électronique sans clé.

78 Consonant : préamplificateur de haute qualité (1978)

Ça commence mal, mais ça finit bien.

81 Hexadoku

Combien de grilles Hexadoku différentes de 16 x 16 chiffres est-il possible de composer ?

86 Avant-première

Pendant que vous lirez ce numéro, nous préparons celui d'avril. Si vous avez des idées pour les suivants, n'hésitez pas à les partager.







Notre équipe

Rédacteur en chef :	Denis Meyer (redaction@elektor.fr)	Directeur éditorial : Wisse Hettinga
Rédaction internationale :	Harry Baggen, Thijs Beckers, Eduardo Corral, Jens Nickel, Clemens Valens	
Laboratoire :	Christian Vossen (resp.), Thijs Beckers, Ton Giesberts, Luc Lemmens, Raymond Vermeulen, Jan Visser	
Graphistes :	Giel Dols, Jeanine Opreij, Mart Schroyen	Secrétaire de rédaction : Hedwig Hennekens
Directeur <i>Elektor online</i> :	Carlo van Nistelrooy	
Ont coopéré à ce numéro :	Antoine Authier, Jean-Paul Brodier, Robert Grignard, Hervé Moreau, Kévin Petit, NN	

Nos réseaux



Nos équipes internationales

 United Kingdom Wisse Hettinga +31 (0) 46 4389428 w.hettinga@elektor.com	 Spain Eduardo Corral +34 91101 93 95 e.corral@elektor.es	 India Sunil D. Malekar +91 9833168815 ts@elektor.in
 USA Hugo Vanhaecke +1 860-875-2199 h.vanhaecke@elektor.com	 Italy Maurizio del Corso +39 2.66504755 m.delcorso@inware.it	 Russia Nataliya Melnikova 8 10 7 (965) 395 33 36 nataliya-m-larionova@yandex.ru
 Germany Ferdinand te Walvaart +31 46 4389417 f.tewalvaart@elektor.de	 Sweden Wisse Hettinga +31 46 4389428 w.hettinga@elektor.com	 Turkey Zeynep Köksal +90 532 277 48 26 zkoks@beti.com.tr
 France Denis Meyer +31 46 4389435 d.meyer@elektor.fr	 Brazil João Martins +55 11 4195 0363 joao.martins@editorialbolina.com	 South Africa Johan Dijk +27 78 2330 694 / +31 6 109 31 926 j.dijk@elektor.com
 Netherlands Harry Baggen +31 46 4389429 h.baggen@elektor.nl	 Portugal João Martins +351 21413-1600 joao.martins@editorialbolina.com	 China Cees Baay +86 21 6445 2811 CeesBaay@gmail.com

35^{ème} année, n°405 - mars 2012

ISSN 0181-7450 Dépôt légal : février 2012
CPPAP 113 U 83713

ELEKTOR / PUBLITRONIC SARL
c/o Regus Roissy CDG - 1, rue de la Haye - BP 12910
FR - 95731 Roissy CDG Cedex - France
Tél. : (+33) 01.49.19.26.19 - Fax : (+33) 01.49.19.22.37
www.elektor.fr

Banque ABN AMRO : Paris

IBAN : FR76 1873 9000 0100 2007 9702 603
BIC : ABNAFRPP

DROITS D'AUTEUR : © 2012 Elektor International Media B.V.

Toute reproduction ou représentation intégrale ou partielle, par quelque procédé que ce soit, des pages publiées dans la présente publication, faite sans l'autorisation de l'éditeur est illicite et constitue une contrefaçon. Seules sont autorisées, d'une part, les reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective, et, d'autre part, les analyses et courtes citations justifiées par le caractère scientifique ou d'information de l'oeuvre

Nos adhérents

Nous
avons



adhérents
dans



pays



Adhérez vous aussi au réseau Elektor...

ElektorHebdo en ligne

notre lettre hebdomadaire d'information électronique gratuite

et à ...

Elektor, le mensuel d'électronique

Tarifs, conditions, offres, nouveautés :

www.elektor.fr/abo

Nos annonceurs



Beta Layout
www.beta-layout.com 71



DesignSpark chipKIT™ Challenge 73
www.chipkitchallenge.com 73



Eurocircuits
www.elektorpcbservice.com 61



Farnell
www.element14.com 87



Hameg
www.hameg.com 9



Jackaltac
www.jackaltac.com 11



Lextronic
www.lextronic.fr 2



Microchip
www.microchip.com 17



MikroElektronika
www.mikroe.com 33



National Instruments
www.ni.com/academic/f 88



Renesas Contest
www.circuitcellar.com/RenesasRL78Challenge 29



Schaeffer AG
www.schaeffer-ag.de 71

Pour placer votre annonce dans le prochain numéro d'Elektor

veuillez contacter Mme Ilham Mohammadi par téléphone au (+33) 01.49.19.26.19

ou par courrier électronique : i.mohammadi@elektor.fr

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

Certains circuits, dispositifs, composants, etc. décrits dans cette revue peuvent bénéficier de droits propres aux brevets; la Société éditrice n'accepte aucune responsabilité du fait de l'absence de mention à ce sujet. Conformément à l'art. 30 de la Loi sur les Brevets, les circuits et schémas publiés dans Elektor ne peuvent être réalisés que dans des buts privés ou scientifiques et non commerciaux. L'utilisation des

schémas n'implique aucune responsabilité de la part de la Société éditrice. La Société éditrice n'est pas tenue de renvoyer des articles qui lui parviennent sans demande de sa part et qu'elle n'accepte pas pour publication. Si la Société éditrice accepte pour publication un article qui lui est envoyé, elle est en droit de l'amender et/ou de le faire amender à ses frais; la Société éditrice est de même en droit de traduire et/ou de faire traduire un article et de l'utiliser pour ses autres éditions et activités, contre la rémunération en usage chez elle.

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

Imprimé aux Pays-Bas par Senefelder Misset - Doetinchem

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

LED fantômes

Les questions les plus triviales appellent des réponses étonnantes. En voici une reçue d'un lecteur en réponse à notre appel lancé dans le numéro de sept. 2011 d'Elektor.

Serge Plaisance
38790 St George d'Espéranche

Monsieur,
Veuillez trouver ci-après une note concernant le clignotement fantôme des LED.

Je vous prie d'excuser mon écriture, mais un handicap me gêne considérablement. L'électroluminescence des cristaux s'accompagne de phénomènes secondaires remarquables :

- 1/ l'énergie des photons se transforme partiellement ou complètement en énergie thermique ;
- 2/ la lumière provoque dans le solide des réactions photochimiques ;
- 3/ le cristal émet des photons de fréquence différente en corrélation avec l'interaction *spin* orbitale des électrons bande de valence et bande de conduction faisant apparaître une structure fine différenciée.

Les états stationnaires excités répartis dans tout le cristal sont déterminés par le vecteur d'onde \vec{k} (une quasi-impulsion $\hbar\vec{k}$ et une énergie $\hbar\Omega(k)$), ils portent le nom de phonons.

Les nombres v_k figurant dans les fonctions $|v_k\rangle$ indiquent le nombre de phonons dans un état donné. Le calcul de $|v_k\rangle$ d'un état avec v_k phonons peut être effectué par application successive de l'opérateur de Bose b_k^+ sur la fonction de l'état vide

$$|v_k\rangle (v_k!)^{-1/2} (b_k^+)^{v_k}$$

Les excitations de grande largeur d'onde

$$Ka = \frac{2\pi a}{\lambda} \ll 1$$

peuvent être considérées comme des ondes élastiques se propageant dans le milieu. La vitesse de ces ondes (vitesse du son) est déterminée par l'expression $v_{ac} = \sqrt{E/\rho}$ où E est le module de Young et ρ la densité.

D'autre part on appelle excitons l'interaction de l'électron et du trou



dans le cristal, elle donne naissance à des états particuliers liés et neutres. À la rupture de l'équilibre thermodynamique du cristal, la luminescence chaude se traduit par de nouveaux phénomènes : Si l'énergie du photon incident $\hbar\omega > Eg$ énergie du bord inférieur de la bande excitonique. Alors un processus indirect, avec participation d'un phonon compensant la variation d'impulsion donne

naissance à un exciton d'énergie cinétique

$$\hbar\omega - Eg - \hbar\Omega_g$$

La durée de vie de l'exciton par rapport à l'émission d'un phonon optique est

$$\tau_r \approx 10^{-12} s$$

et la durée de vie relative à l'émission de lumière est $\tau_r \approx 10^{-8} s$.

Ainsi l'émission d'un photon d'énergie $\hbar\omega - 2\hbar\Omega$ correspond à une probabilité égale à $\omega \approx \tau_r / \tau_i \approx 10^{-4}$

La perte d'énergie $\hbar\Omega$ est liée à l'émission d'un phonon compensant la variation d'impulsion de l'exciton.

Les excitons entrent en interaction avec les phonons optiques transversaux en perdant de l'énergie lors de la thermalisation par quantité discrète $\hbar\Omega$.

À chaque nième étape de ce processus en cascade on peut avoir une émission d'un photon avec probabilité de $\sim 10^{-4}$ et d'énergie

$$\hbar\omega - (n+1)\hbar\Omega_0$$

Le rayonnement du cristal présente à la fois les aspects de la luminescence et de la diffusion Raman.

Comme dans le spectre de diffusion Raman, la position des maxima des bandes d'émission change lorsque la fréquence de la lumière incidente varie. Les fréquences d'émission sont distantes d'un nombre entier de fréquences de la raie excitatrice. Les photons des états initial et final étant corrélés.

Par ailleurs l'émission s'amortit chaque fois que l'exciton perd un quantum d'énergie. Lorsque la température augmente, les raies s'élargissent et leur intensité diminue jusqu'à n'être plus perceptible.

La LED s'éteint.

Serge Plaisance

Vu sur le forum d'Elektor :

Fabrication de circuits imprimés

JeanPascal, l'auteur du message, a écrit :

Suite au vieillissement des tubes de mon insoleuse, et à la lecture de l'article de R. JOUCHET (Courrier des lecteurs dans le n° de janvier 2012), j'envisage de réaliser une insoleuse à LED UV. J'ai donc repris l'article du n° 344 (02/2007) et j'ai cherché des LED UV que j'ai trouvées chez www.planetleds.fr, mais la température de couleur est de 405 nm et l'angle de diffusion de 15 à 140°

Donc s'il y a des spécialistes :

- est-ce que température de couleur équivaut à longueur d'onde ?

- et si oui, est-ce que 405 nm convient pour l'insolation des CI ?

Concernant l'angle de diffusion, je vais refaire les calculs pour avoir l'espacement des LED, mais pour un angle de 140° cela signifie-t-il que la densité lumineuse est plus faible (donc augmentation du nombre de LED) ?



Merci d'avance pour vos lumières

Lampes à économie d'énergie

jpbru24, l'auteur du message, a écrit :

En faisant mes courses dans le grand magasin du coin, j'ai constaté qu'en quelques jours le bac de récupération des dites ampoules en avait reçu des centaines. Par curiosité, j'en ai récupéré une dizaine parmi les intactes (ce qui est le cas de la moitié seulement d'entre elles, bonjour les gaz toxiques pour les clients !) et me suis mis à les disséquer.

Deux seulement avaient le tube mort, trois avaient des résistances claquées, deux des transistors HS mais que je n'ai

pu remplacer, car réservés à la haute tension. Deux ampoules m'ont intrigué, car rien ne semblait mort, mais après dessoudage des composants j'ai trouvé pour l'une un condo marqué 2,2 nF alors que je mesure 11 nF, et l'autre marqué 3,3 nF mais sur lequel j'ai mesuré 10 nF. Après remplacement par des composants de valeur correcte, ces lampes fonctionnent à nouveau. La capacité qui a augmenté m'intrigue, car ces capas apparaissent intactes et sont placées aux bornes réservées pour les tubes au starter. En résumé, sur dix lampes, cinq fonctionnent à nouveau. Le manque de fiabilité de ces ampoules serait dû en grande partie à des composants électroniques sous-dimensionnés ou de bas de gamme.

Programmation des PIC 12C509

J'ai en ma possession des PIC 12C509 et je souhaiterais les programmer. Mais je ne trouve pas de fichier dans la bibliothèque de JAL.



Je n'en trouve pas non plus sur internet. Autodidacte, je ne sais encore modifier les fichiers présents dans la bibliothèque pour les adapter à mes besoins. Pouvez-vous m'aider à résoudre ce problème ? Je me suis procuré le programmeur préconisé dans le livre « 50 applications des microcontrôleurs PIC » de Bert van Dam. Malheureusement, son manuel utilisateur est en anglais et je n'ai pas trouvé ce manuel en français. Je ne sais pas si je peux connecter ce programmeur au PIC 12C509. Pouvez-vous me renseigner à ce sujet ?

Gilbert Hullik

La réponse de Bert van Dam, l'auteur du livre :

Il existe des centaines de microcontrôleurs PIC. Tous ne sont pas compatibles avec le langage de programmation JAL gratuit. C'est pourquoi j'ai composé une bibliothèque qui exploite pleinement les possibilités des microcontrôleurs PIC les plus courants, et facilite les migrations (liste complète dans le chap. 12 du livre). Puisque Microchip, fabricant du 12C509, déconseille l'utilisation de ce composant désormais obsolète, je recommande le 12F675, plus rapide et compatible avec JAL, doté d'une mémoire flash et de fonctions supplémentaires. Pour programmer le 12F675, il faut un programmeur flash, tel que le Wisp648 recommandé dans le livre.

Une précision d'Elektor : ce conseil bien intentionné ne répond que partiellement à la question. On ne saurait évidemment en vouloir à JAL de ne pas être compatible avec tous les PIC, il y en a tellement. Peut-être que parmi les lecteurs d'Elektor quelqu'un saura comment adapter JAL à votre micro. C'est la mission du forum d'Elektor, utilisez-le pour ça.

OBD2 sans fil

Elektor n° 400, oct. 2011

J'ai acheté et monté le kit Bluetooth OBD2. Tout fonctionne parfaitement avec le logiciel Moddiag. Pour information, j'ai également essayé sur un smartphone Android avec le programme *Torque* disponible en version gratuite sur l'*Android Market* et cela fonctionne parfaitement (il existe également une version payante avec des fonctions étendues). Meilleures salutations et merci pour vos articles et vos réalisations très intéressantes.

Philippe Monnard
Centre de formation UPSA-VD
1400 Yverdon-les-Bains

Merci pour cette information utile. On ne parle jamais assez des trains qui arrivent à l'heure.

Le bus arrive

Elektor n°396, juin 2011

J'ai cru comprendre en lisant les articles sur « le bus arrive » qu'il existe une liste de diffusion. Mais je ne l'ai jamais trouvée. Existe-t-elle uniquement en allemand ? (L'auteur, Jens Nickel est allemand, je pense) Le bus RS485 d'Elektor ne répond pas au standard TIA-EIA-RS485 en ce qui concerne la polarité des signaux A et B. Mais c'est tout à fait normal, il existe deux normes à ce sujet. La vraie, et celle de facto de *Texas Instruments* (suivie par d'autres fabricants d'ailleurs) Voir <http://en.wikipedia.org/wiki/RS-485>, chapitre « Signs of common mistakes » ou la référence 6 dans cette même page Wikipedia. Attention donc si vous connectez d'autres produits du commerce sur du RS485 comme j'en ai fait plusieurs fois l'expérience, notamment avec des produits de B&B Electronics. Il manque les pull-up et pull-down de bus (voir <http://www.bb-elec.com/bb-elec/literature/tech/485appnote.pdf> page 18), mais attention, le pull-up doit se trouver entre A et le +5V, et le pull-down entre B et la masse.

Raymond Ranwez

La liste de diffusion est internationale, la langue utilisée est l'anglais, plusieurs Français y participent, mais pour des raisons pratiques elle est à accès limité.

Nous attendons de Jens Nickel la réponse à vos questions.

Publicité

SCOPES and more

HAMEG®
Instruments

A Rohde & Schwarz Company

**Great Value in
TEST & MEASUREMENT**

Règles du jeu

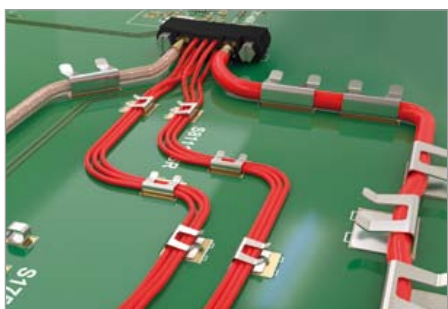
Les points de vue et opinions exprimées ne sont pas nécessairement ceux du rédacteur en chef ou de l'éditeur.

La rédaction se réserve le droit de traduire, de corriger ou d'écourter les messages. Si vous écrivez au COURRIER DES LECTEURS, précisez le numéro du magazine concerné.

Ecrivez-nous :
redaction@elektor.fr ou
Elektor (rubrique Courrier) BP 12910
FR - 95731 Roissy CDG Cedex

Fini le bricolage, attachez vos fils et câbles proprement !

Harwin étend sa famille EZ-BoardWare d'accessoires CMS pour cartes à circuits imprimés, suivant ainsi la tendance de l'industrie à se détourner des « bricolages » liés aux accessoires traditionnels de gestion de câbles, de CEM, de test et de batteries, qui sont souvent massifs et exigent en général des tâches d'assemblage secondaires.



La famille innovante d'accessoires à montage en surface EZ-BoardWare assouplit le processus de fabrication, réduit les coûts de fabrication, fait gagner de la place et facilite la maintenance sur site. Elle propose des solutions simples et élégantes aux pénibles défis posés par les fonctions accessoires. Remplaçant des solutions de compromis, souvent traversantes ou vissées, les nouveaux composants peuvent tous être montés en surface avec les autres composants, ce qui économise de l'espace, du temps et des coûts.

Les dernières nouveautés de la famille incluent les réceptacles EZ-SMT offrant pour les applications moins exigeantes une alternative plus économique aux réceptacles à contacts formés au tour ; les cavaliers à bas profil EZ-Link, dont la hauteur atteint maintenant juste 0,97 mm ; la gamme élargie de clips de gestion de câble EZ-Cable, avec des versions pouvant gérer deux fils ou un câble coaxial, et même fournir la masse pour améliorer la CEM.

(110703-C) www.harwin.com/ez-boardware

Contrôleur PXI Express avec processeur Intel Core i5

ADLINK Technology annonce son premier contrôleur PXI Express, le PXIe-3975, de



3U, doté du processeur Intel Core i5-520E destiné aux systèmes de test hybrides basés PXIe. Avec un débit système maximum de 2 Go/s ce contrôleur vise les systèmes de test PXIe moyenne gamme pour les applications de tests électroniques fonctionnels et universels.

Le contrôleur est doté d'une mémoire DDR3 1066 MHz (8 Go max.) et d'un disque dur SATA de 160 Go, 7200 RPM. Grâce à sa grande capacité d'interfaces E/S, il fournit une connectivité aisée à d'autres instruments. Il offre une évolutivité flexible avec un connecteur d'extension ExpressCard/34 situé sur le panneau avant. Les utilisateurs peuvent ainsi étendre les fonctions du système à l'aide de modules ExpressCard faciles à installer, tels que ExpressCard vers USB 3.0 ou un kit de bus d'extension de ExpressCard à PXI / PCI. Ce contrôleur propose également une grande flexibilité au niveau des interfaces, incluant deux ports Gigabit Ethernet, dont un pour la connexion LAN et l'autre pour le contrôle d'instruments LXI, quatre ports USB 2.0 pour connecter des périphériques et contrôler des instruments USB ainsi qu'un connecteur GPIB Micro D pour le branchement d'instruments GPIB permettant le contrôle des systèmes de test hybrides basés PXIe.

(110703-G) www.adlinktech.com/PXI/

Un inverseur avec une densité de puissance de 30 kW/l

Une équipe du consortium de R et D japonais Future Power Electronics Technology (FUPET), issu d'un partenariat industrie – universités incluant Fuji

Electric, Nissan Motor, Sanken Electric et Toshiba, développe des convertisseurs d'énergie capables de fonctionner à haute température en produisant une forte densité de puissance. A l'aide des JFET SiC de SemiSouth Laboratories, cette équipe a développé un inverseur triphasé de 500 cm³ fournissant 15 kW (30 kWh/l) pour un moteur triphasé avec 99 % de rendement de conversion. Comportant un système de refroidissement compact optimisé, ce module d'alimentation fonctionne jusqu'à 200 °C.

« Nous pensons avoir atteint la plus forte densité de puissance au monde pour un inverseur de petit volume », déclare Satoshi Tanimoto, directeur de recherche au centre R et D du FUPET. Selon les estimations, lorsque les panneaux photovoltaïques utiliseront des inverseurs de cette taille et de cette capacité, il sera possible de fournir assez d'électricité pour cinq ménages avec un seul inverseur.



Les JFET de SemiSouth sont compatibles avec les puces de commande de grille standard, et présentent un coefficient de température positif pour faciliter leur mise en parallèle, une fréquence de commutation extrêmement rapide sans « courant de queue » jusqu'à la température de fonctionnement maximum de 150 °C et une très faible $R_{DS(on)max}$. Ils sont disponibles en boîtier TO-247 et peuvent aussi être fournis sous forme de puce nue à intégrer dans des modules.

L'équipe du FUPET a obtenu ces résultats à 50 kHz, la fréquence minimum visée. L'inverseur contient aussi un module d'inductance de seulement 5 nH. Le FUPET a pour objectif de réaliser l'an prochain un inverseur de densité de puissance de 40 kW/l.

(110703-B) www.semisouth.com

Mieux compter les coulombs augmente la durée des batteries Li+



Les jauges traditionnelles pour batteries Li+ sont montées dans la batterie et nécessitent le support de nombreux composants discrets. Pour estimer la capacité de batterie restante, les jauges s'appuient sur des compteurs de coulombs. Ces dispositifs posent un problème à cause de petites erreurs de décalage de conversion analogique/numérique qui s'accumulent indéfiniment. Une résistance de mesure de grande taille et coûteuse est nécessaire pour corriger ces dérives. De plus, la batterie doit passer régulièrement par les états batterie pleine, vide et veille.

De récentes avancées ont permis d'implémenter la jauge du côté système, au lieu de l'insérer dans la batterie. Si cette approche réduit le coût de l'application, elle a peu d'effet sur la surface de carte, car il est toujours nécessaire d'utiliser une grande résistance.

Finalement, les algorithmes de correction des dérives actuellement disponibles ont des effets secondaires indésirables. Ils introduisent de brusques sauts de l'état de charge dû au fait que les algorithmes s'appuient sur la mesure de la tension de la batterie en état de veille, puis font une correction en se basant sur la relation entre la tension de circuit ouvert de la batterie et l'état de charge de la batterie.

La nouvelle technique ModelGauge m3 de Maxime utilisée par le circuit MAX17047

combine la précision à court terme et la linéarité d'un compteur de coulombs et la stabilité à long terme d'une jauge à mesure de tension. Cet algorithme effectue continuellement de petites corrections, et évite ainsi les brusques corrections opérées par les algorithmes traditionnels de comptage de coulombs.

Comme l'algorithme réduit la sensibilité aux erreurs de décalage du convertisseur analogique/numérique de mesure de courant, le circuit peut utiliser une plus petite résistance de mesure sans compromettre la précision de l'estimation de l'état de charge de la batterie.

Le MAX17047 est capable de donner l'alarme si la température de la batterie est anormale, contribuant à la sécurité et à la durée de vie du système. Enfin, il prévient aussi le système quand la batterie est insérée ou enlevée, et gère en même temps la séquence de démarrage / d'arrêt.

(110703-7) www.maxim-ic.com/ModelGauge

Calibrateurs compétitifs, compacts et précis

Pour générer, simuler et mesurer avec une grande précision tous les signaux bas niveaux rencontrés dans l'industrie, Chauvin Arnoux commercialise une nouvelle gamme composée de trois instruments : C.A 1621, C.A 1623 et C.A 1631. Il s'agit des instruments tout terrain compacts et robustes avec une prise en main aisée et un grand afficheur qui offre une bonne lisibilité en toute circonstance. Alimentés par piles, ils sont indépendants et offrent une autonomie appréciable.

Le champ d'application de ces instruments est large, ils conviennent aux besoins des ingénieurs, techniciens et installateurs de multiples secteurs d'activité comme le chauffage et climatisation, l'industrie de transformation (chimie, pétrochimie,

plastique, industrie agroalimentaire...), les organismes de contrôle, les laboratoires et l'enseignement ou encore dans les hôpitaux.

Ils assurent deux fonctions principales : la simulation et la mesure. Le C.A 1621 permet la calibration de sondes thermocouples (J, K, T, E, R, S, B, N). Sa plage de mesure et de simulation de température est de -250 °C à +1 800 °C et sa plage de mesure et de simulation millivolt va de -10 à 100 mV. Le C.A 1623 sert pour la calibration de sonde résistive Pt10, Pt50, Pt100, Pt200, Pt500, Pt1000, Pt1000 (JIS) avec une plage de mesure et de simulation de température de -200 °C à +800 °C et une plage de mesure et simulation de résistance de 0 à 3 200 Ω. Enfin, le C.A 1631 est adapté pour la calibration de signaux de processus en tension et courant. Il mesure et délivre une tension continue de 0 à 20 V (± 0,02 %) ou il mesure et délivre une boucle de courant continu de 0 à 24 mA (0,015 %).

(110713-1)

www.chauvin-arnoux.com



Publicité



PCBs Muuuuch Cheaper...

No-frills policy

16.94 EURO*

5 pcs, 100 mm x 100 mm
*per piece, incl. VAT (21%)
+ shipping costs e. g. Germany 10.71 EURO



www.jackaltac.com

2,4 GHz : porteuse polymorphe

Une fréquence très fréquentée

Frank Bigrat (France)

Wi-Fi, Bluetooth, four à micro-ondes... nous utilisons tous les jours sans y penser des appareils qui tutoient des fréquences de l'ordre de 2,4 GHz. Leurs applications sont nombreuses, mais pas forcément connues du grand public. Nous vous proposons un rapide tour d'horizon des technologies basées sur le 2,4 GHz puis plus précisément une explication de la technique de l'étalement de spectre qui permet d'en tirer de nombreux bénéfices.

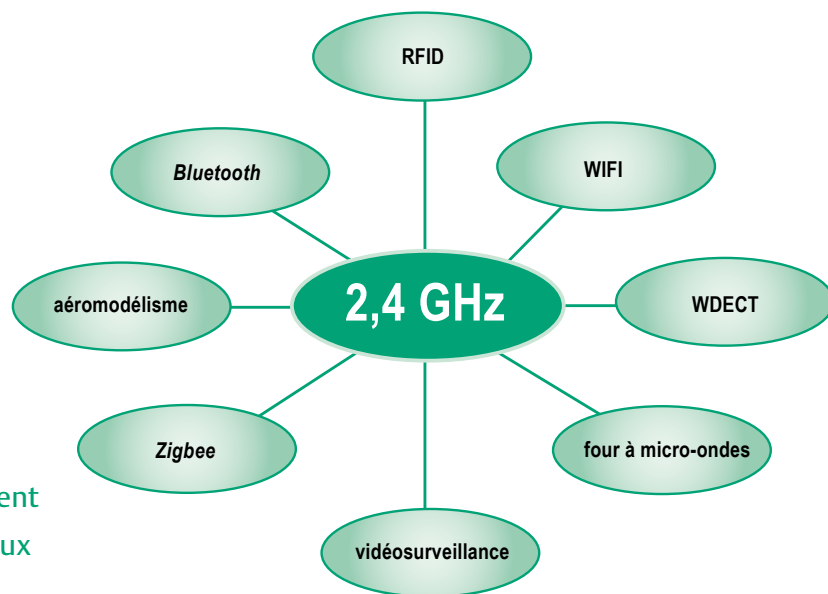


Figure 2 – Les différentes applications de la fréquence 2,4 GHz.

La bande ISM

La fréquence 2,4 GHz appartient à une bande réservée aux applications Industrielles, Scientifiques et Médicales, la bande ISM, qui se décompose comme l'indique sommairement le **tableau 1**. Un exemple d'appareil bien connu travaillant dans la bande ISM est le four à micro-onde. Moins connues, sont les machines médicales à diathermie qui chauffent les tissus humains par émission d'ondes électromagnétiques, permettant notamment la relaxation des muscles.

Seule la bande des 2,4 GHz peut être utilisée sans restriction dans le monde entier. C'est ce qui explique la profusion des appareils de communication qui fonctionnent sur cette bande de fréquences. Elle trouve des applications dans des domaines très divers, que je propose de survoler (**figure 2**) :

- **Wi-Fi (Wireless Fidelity)** : elle est, avec le *Bluetooth*, la technologie 2,4 GHz la plus répandue et la plus connue. Utilisé couramment aujourd'hui pour l'accès sans fil à l'internet depuis les ordinateurs portables et les tablettes, le Wi-Fi est apparu sur des ordinateurs Apple dès la fin des années 1990. Les réseaux locaux sans fil sont dits WLAN, pour *Wireless Local Area Network*.
- **RFID (Radio Frequency Identification)** : utilisées quotidiennement par exemple par les usagers des transports en commun des grandes villes, cartes *Navigo* dans la région parisienne, ces radioétiquettes permettent d'écrire et de récupérer des données sans contact physique. L'énergie est fournie par l'onde

radio émise par le lecteur lors de l'interrogation de l'étiquette.

- **Bluetooth** : C'est le fabricant de téléphones mobiles *ERICSSON* qui a conçu ce standard de communication sans fil pour éliminer les câbles entre appareils électroniques proches, essentiellement dans les applications domestiques. Les modules utilisés par la plupart des fabricants rayonnent une puissance de 2,5 mW, ce qui limite leur portée à 20 m. Cette faible puissance des appareils *Bluetooth* est compensée par une autonomie de plusieurs mois.
- **Zigbee** : Ces modules permettent de créer des réseaux sans fil personnels dits *Wireless Personal Area Networks (WPAN)* de faible coût et d'une puissance si faible que leur autonomie est de plusieurs années. Ainsi la domotique et les nombreux capteurs et télécommandes qu'elle met en œuvre utilisent de plus en plus ce protocole. On retrouve aussi le ZigBee dans les applications médicales, les détecteurs de fumée et d'intrusion et dans la télécommande de la dernière génération de la *freebox*.
- **WDECT (Wideband Digital Enhanced Cordless Telecommunication)** : Technologie de téléphones sans fil, évolution du DECT (*Digital Enhanced Cordless Telecommunication*) réservé à une utilisation domestique ou bureautique. WDECT utilise un protocole FHSS que nous expliquerons plus loin.
- **Four à micro-ondes** : Cet appareil devenu indispensable pour beaucoup d'entre nous a été imaginé dans les années 1950. La possibilité de réchauffer des aliments au moyen de micro-ondes (de 1 GHz à 300 GHz) fut observée par hasard par l'ingénieur

Tableau 1 – La bande ISM est réservée aux applications Industrielles, Scientifiques et Médicales.

Gamme de fréquences	Fréquence centrale	Disponibilité
6,765 – 6,795 MHz	6,780 MHz	Selon réglementations locales en vigueur
13,553 – 13,567 MHz	13,560 MHz	
26,957 – 27,283 MHz	27,120 MHz	Pas d'informations particulières communiquées
40,66 – 40,70 MHz	40,68 MHz	
433,05 – 434,79 MHz	433,92 MHz	Pas d'informations particulières communiquées
902 – 928 MHz	915 MHz	États-Unis, Groenland et certaines îles du Pacifique uniquement
2,400 – 2,500 GHz	2,450 GHz	
5,725 – 5,875 GHz	5,800 GHz	Pas d'informations particulières communiquées.
24 – 24,25 GHz	24,125 GHz	
61 – 61,5 GHz	61,25 GHz	Selon réglementations locales en vigueur
122 – 123 GHz	122,5 GHz	Selon réglementations locales en vigueur
244 – 246 GHz	245 GHz	Selon réglementations locales en vigueur

Percy Spencer alors qu'il travaillait à proximité d'un radar en activité. En effet, la fréquence de 2,45 GHz permet d'agiter les molécules d'eau et ainsi de réchauffer les aliments.

- **Aéromodélisme** : Quand il arrive sur un terrain, un pilote de modèle réduit d'avion se préoccupe toujours de la question de l'occupation éventuelle de « sa » fréquence par d'autres pilotes. Seul un partage équitable du temps de vol permet d'éviter le crash des avions ; un problème que l'auteur de ces lignes ne connaît que trop bien. Dans ce contexte, la technologie des 2,4 GHz est intéressante, car elle permet de s'affranchir de ces problèmes de fréquence. Les fabricants les plus connus, *Futaba*, *Graupner*, *Spektrum*, *Multiplex*... ont bien compris l'avantage des techniques de transmission numérique et proposent une large gamme de modèles d'émetteurs et de récepteurs en 2,4 GHz. Certains fabricants proposent même sur leur matériel, en plus

tion commerciale des produits, on préfère les noms plus évocateurs et plus attractifs. L'origine du 2,4 GHz remonte à 1947 et l'invention du four à micro-ondes (voir plus haut) : La *Federal Communications Commission*, organisme américain de certification et de validation de tous les périphériques qui émettent des ondes électromagnétiques, a décidé d'attribuer la bande des 2,4 GHz aux appareils de communication sans fil.

Techniques de transmission : DSSS et FHSS

À l'évocation des bénéfices de la technologie des 2,4 GHz pour l'aéromodélisme, on se pose aussitôt la question de savoir comment font les appareils qui émettent sur une même fréquence pour ne pas se perturber les uns les autres.

La réponse est (assez) simple : on fait appel à la technique de l'étalement de spectre (*Spread Spectrum*) dans laquelle le domaine de

Il est difficile de connaître la part exacte d'implication et donc l'influence des industriels des télécommunications dans les études de nocivité

de la liaison montante classique, une liaison descendante qui permet d'afficher en temps réel, sur l'émetteur du pilote, divers paramètres de vol, tels l'altitude de l'avion ou l'état de charge des batteries. Cette technique a été mise à profit dans l'article **Télémessure pour batteries LiPo par Zigbee**, dans le numéro de février 2012 d'Elektor [1].

- **Vidéosurveillance** : les boutiques d'accessoires électroniques proposent différents modèles de caméras de surveillance dans la bande des 2,4 GHz.

Comme on l'imagine, *Wi-Fi*, *Bluetooth*, *Zigbee* et autres sont des noms commerciaux et souvent des marques déposées. Ces protocoles de communication sont régis par des normes aux noms moins faciles à retenir : IEEE 802.11 pour le *Wi-Fi*, IEEE 802.15 pour le *Bluetooth* et le *Zigbee*. Elles définissent le protocole de transmission de données, le taux de transfert, la portée en intérieur et en extérieur, la puissance rayonnée, et bien davantage. Pour faciliter la pénétra-

tion commerciale des produits, on préfère les noms plus évocateurs et plus attractifs. L'origine du 2,4 GHz remonte à 1947 et l'invention du four à micro-ondes (voir plus haut) : La *Federal Communications Commission*, organisme américain de certification et de validation de tous les périphériques qui émettent des ondes électromagnétiques, a décidé d'attribuer la bande des 2,4 GHz aux appareils de communication sans fil.

On distingue deux types de technique d'étalement de spectre :

- le *Direct Sequence Spread Spectrum* (DSSS) ou étalement de spectre par séquence directe
- le *Frequency Hopping Spread Spectrum* (FHSS) ou étalement de spectre par saut de fréquence

À l'origine, le but de cette technique était de lutter contre l'interception des signaux par un ennemi. Il s'agissait de dissimuler l'information transmise en donnant au signal émis l'apparence du bruit. En théorie de l'information, le bruit est une information non pertinente.

Inventée pour permettre le guidage de torpilles sous-marines, cette technique est connue depuis la Deuxième Guerre mondiale, mais ne fut utilisée par les Américains qu'à partir des années soixante, plus

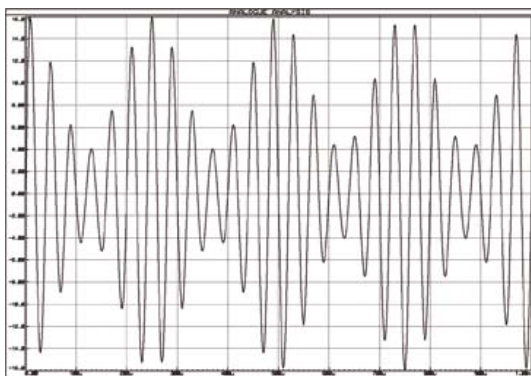


Figure 3 – Un signal modulé en amplitude (AM).

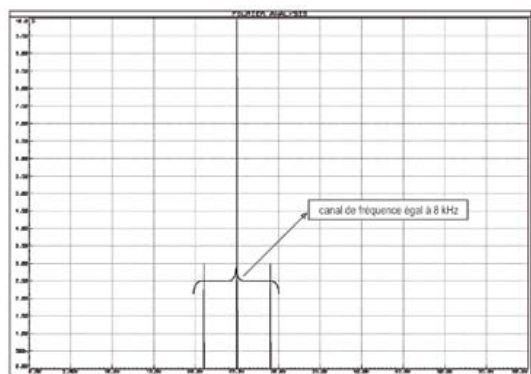


Figure 4 – Spectre d'un signal modulé AM.

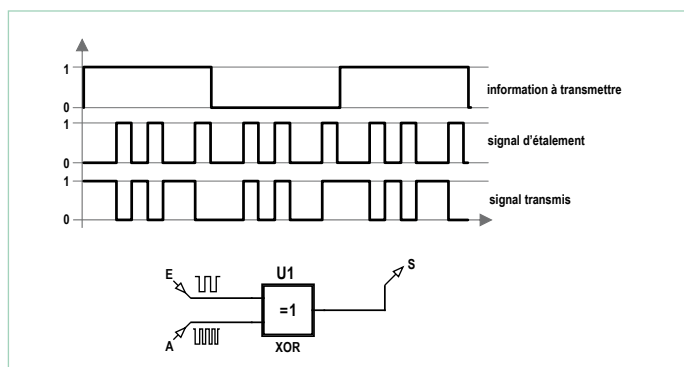


Figure 5 – Le principe du DSSS.

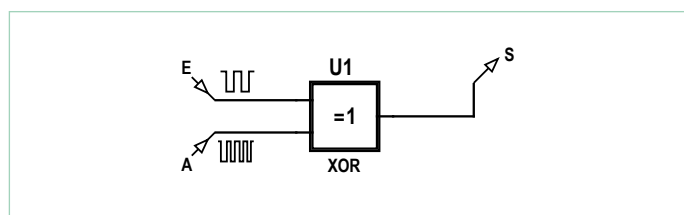


Figure 6 – Un générateur DSSS très simple

précisément lors de la crise des missiles cubains, en octobre 1962. Cette grave crise internationale, exacerbée par l'installation par l'URSS de missiles nucléaires sur l'île de Cuba, à quelques centaines de kilomètres des côtes de la Floride, a culminé par le blocus de l'île par les États-Unis et des démonstrations de force qui n'ont pas été loin de dégénérer, transformant alors en *guerre chaude* ce que l'on appelait la guerre froide.

Puisque nous évoquons ce qui n'est plus aujourd'hui qu'une anecdote, il est intéressant de noter que la technique fut imaginée et brevetée par un curieux personnage, pianiste et compositeur connu des amateurs avertis, mais aussi auteur, critique et endocrinologue autoproclamé, du nom de George Antheil, en étroite coopération avec sa compagne l'actrice hollywoodienne Hedy Lamarr, divorcée d'un industriel de l'armement autrichien.

De nos jours, la technique est surtout utilisée pour diminuer le risque d'interférences avec d'autres signaux.

Pour bien comprendre les notions de DSSS et FHSS, un petit rappel sur la modulation s'impose. Pour transmettre dans de bonnes conditions un signal sinusoïdal s , de fréquence f (pour notre exemple $f = 4$ kHz), ce signal va moduler, en amplitude ou en fréquence, un autre signal sinusoïdal P , de fréquence F (pour notre exemple $F = 25$ kHz), appelé *porteuse*.

Si on analyse le spectre résultant — lequel n'est qu'un graphique de l'amplitude de chaque composante sinusoïdale, appelée harmonique, d'un signal — d'une modulation d'amplitude classique (**figure 3**), le signal se retrouve décalé et centré autour de la porteuse P (**figure 4**). De plus, on voit aisément que le signal transmis occupe une bande de fréquences appelée canal, et dont la largeur est ici égale à 8 kHz.

La modulation utilisée pour transmettre des signaux numériques est la PSK dont nous reparlerons en détail dans une autre partie de cet article.

Dans la technique d'étalement de spectre par séquence directe, dite DSSS, chaque bit du signal numérique à transmettre est remplacé par une séquence de n bits (**figure 5**). Cette séquence est définie en mélangeant le signal numérique à transmettre avec une séquence binaire appelée *signal d'étalement* dont le schéma pseudoaléatoire est défini entre l'émetteur et récepteur. Avec cette technique, le spectre du signal transmis est n fois plus large.

Dans notre exemple, le signal d'étalement possède un débit huit fois supérieur à celui du signal à transmettre : son spectre s'étale donc dans un espace huit fois plus étendu. Le signal qui en résulte est transmis par radio en utilisant une modulation de type PSK (voir plus loin).

Utilisons un simulateur (par exemple Proteus, de la société *Labcenter Electronics*) pour étudier les effets du DSSS sur le spectre d'un signal. Le signal à transmettre E est un signal rectangulaire, de fréquence 1 kHz, et le signal d'étalement est une séquence binaire choisie arbitrairement : 0100101101.

La **figure 6** présente la structure très simple que nous utiliserons pour réaliser un DSSS et la **figure 7** présente le résultat de la simulation.

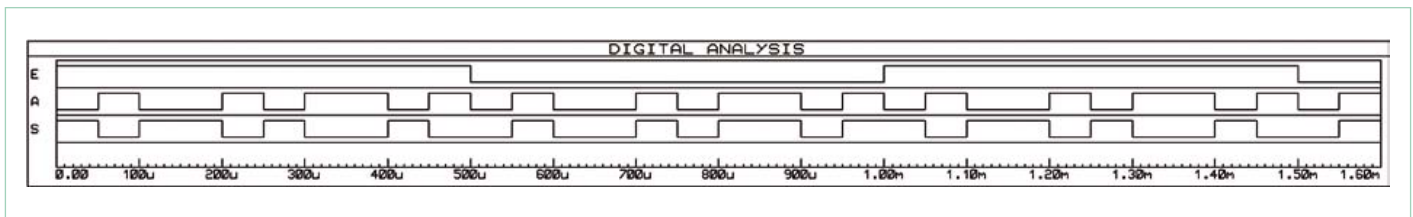


Figure 7 – Le résultat de la simulation.

Le spectre du signal original est donné **figure 8**. On voit aisément que les harmoniques constituant ce signal s'atténuent très vite à partir d'une fréquence supérieure à (50 kHz).

La **figure 9** présente le spectre du signal d'étalement. On voit bien que celui-ci s'étend sur une bande de fréquences plus étendue que 200 kHz.

Le spectre du signal transmis après étalement est donc celui donné par la **figure 10** : ce dernier s'étend lui aussi au-delà de 200 kHz. Le spectre est donc bien étalé et possède bien l'apparence d'un signal bruité.

Dans la technique de l'étalement de spectre par saut de fréquence, ou FHSS, la fréquence de la porteuse change périodiquement selon un schéma pseudoaléatoire et occupe un canal donné durant un temps donné (appelé *dwell time*, et égal à 300 ms conformément à la norme IEEE 802.11). L'émetteur et le récepteur définissent l'ordre dans lequel ils vont utiliser les fréquences. Ceci peut se résumer par le graphique de la **figure 11**.

Le FHSS exploite 79 canaux larges de 1 MHz. Avec cette technique, la probabilité de collision, c'est-à-dire que plusieurs émetteurs travailleraient à un instant donné sur le même canal, est très faible.

Modulation du signal

Associée à l'étalement de spectre, on module le signal avec une technique bien connue des électroniciens : la modulation de phase ou PSK pour *Phase Shift Keying*. Dans ce type de modulation, on modifie la phase à l'origine ϕ d'un signal sinusoïdal (donné par l'équation $U(t) = \hat{U} \cdot \sin(2 \cdot \pi \cdot f \cdot t + \phi)$), selon que l'on veut transmettre un 1 ou un 0. Faisons le choix arbitraire d'avoir $\phi = 0$ lorsque l'on transmet un 1 et $\phi = \pi$ lorsque l'on transmet un 0, le signal modulé aura l'allure donnée par la **figure 12**.

Les MMIC

Arrivés à ce point, nous devons nous poser une autre question : comment transmettre efficacement des signaux à une fréquence aussi élevée que 2,4 GHz quand on sait que la bande passante des composants courants n'est que de quelques centaines de MHz ? Cette prouesse n'est possible que depuis l'apparition des circuits intégrés monolithiques hyperfréquence, appelés MMIC (prononcez MIMIC) pour *Monolithic Micro-Wave Integrated Circuits*.

Comme leur nom ne l'indique pas, ce sont des circuits intégrés à base d'arséniure de gallium (GaAs), un semi-conducteur aux caractéristiques électriques bien meilleures que celles du silicium habituellement utilisé, qui leur permettent de fonctionner à des fréquences comprises entre 300 MHz et 300 GHz et de réaliser toutes les fonctions indispensables à une chaîne de transmission radio (mixage de fréquence, amplification de puissance, amplification faible bruit...).

Il est également possible de créer des MMIC à base de phosphore d'indium (InP). Ce semi-conducteur aux caractéristiques électriques encore meilleures que celles de l'arséniure de gallium permet de

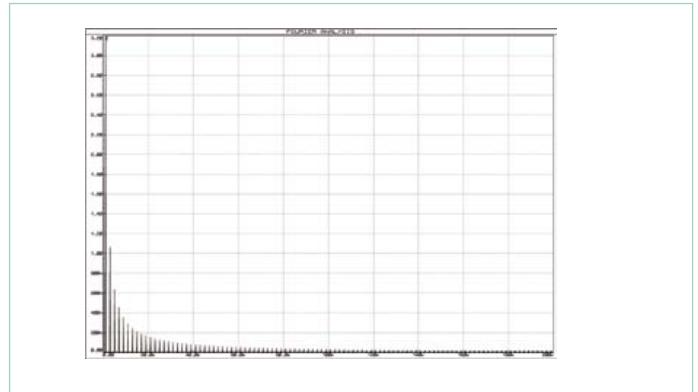


Figure 8 – Le spectre du signal à transmettre.

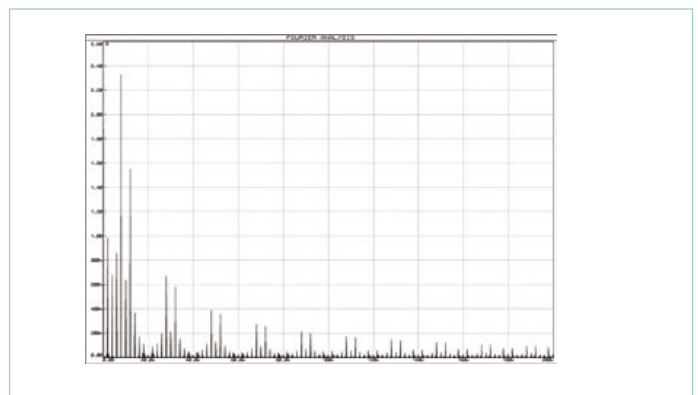


Figure 9 – Le spectre du signal d'étalement.

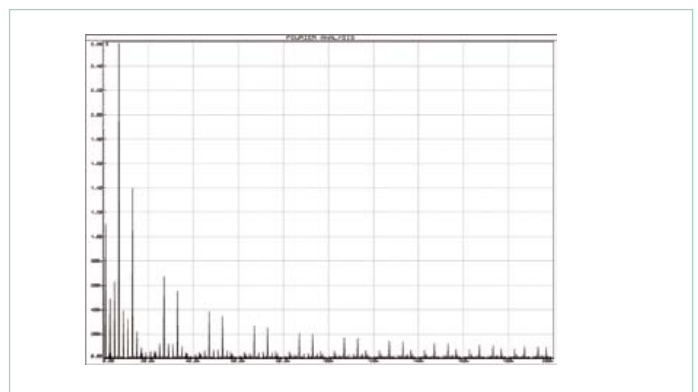


Figure 10 – Le spectre du signal transmis après étalement.

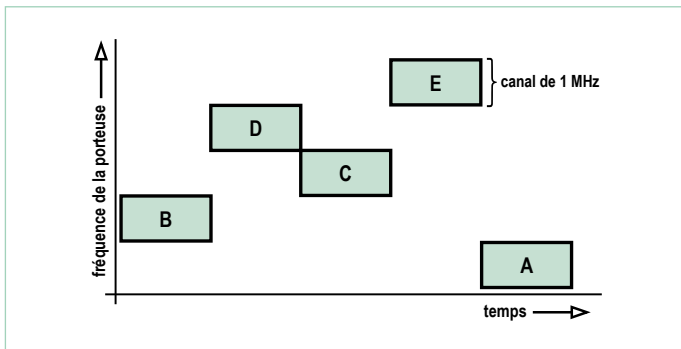


Figure 11 – Le principe du FHSS.

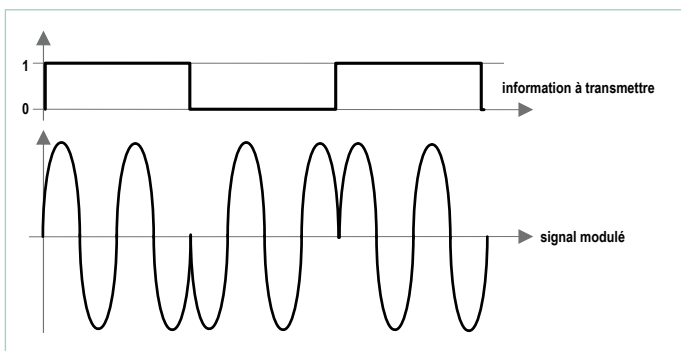


Figure 12 – La modulation de phase (PSK).

réaliser des composants aux bandes passantes encore plus étendues, mais aussi plus chers.

Depuis les années quarante, l'industrie électronique est engagée dans une course aux hautes fréquences. D'abord obtenues au moyen de tubes électroniques (magnétrons, tubes à ondes progressives...) les fréquences atteintes à cette époque ne dépassaient pas 2 GHz. Ces appareils sont encore utilisés de nos jours, notamment pour les applications de forte puissance, tels les fours à micro-ondes, les radars ou la transmission par satellite.

La révolution survenue dans les années soixante-dix avec l'apparition des circuits intégrés, a favorisé le développement d'une technologie qui couvre des domaines aussi divers que le militaire (guidage par radar, brouillage et contre-mesures...), les télécommunications (réception de satellite, téléphonie sans fil...), l'industrie (analyse de matériaux), le médical (traitement de certaines tumeurs...), le spatial (radar spatial comme le satellite SPOT...), ou encore l'automobile (radar anti-collision, capteur de vitesse pour le déclenchement des coussins d'air ou l'ABS...), les transports (GPS, radars routiers...).

Effets sur l'organisme des ondes radio

En dépit de l'utilisation massive des micro-ondes en télécommunication, les effets à long terme sur l'organisme des champs élec-

tromagnétiques de hautes fréquences sont encore mal connus. Plusieurs études récentes tendent à prouver leur nocivité (qui se traduirait par des troubles du sommeil, une fatigue excessive, des maux de tête, des vertiges, des nausées, des troubles auditifs, des troubles de la mémoire, des difficultés de concentration...). Dans un communiqué de presse très discuté [2], daté de nov. 2011, l'Association Santé Environnement de France affirmait l'existence d'un lien entre les effets cités précédemment apparaissant chez les habitants de plusieurs HLM et la présence d'un nombre élevé d'antennes relais installées sur cette résidence. D'autres études affirment le contraire : le *Scientific Committee on Emerging and Newly Identified Health Risks*, groupe d'experts de la Commission européenne, a réaffirmé en mars 2007 : « après avoir revu l'ensemble des études scientifiques in vivo, in vitro et épidémiologiques menées depuis le dernier avis européen de 2001, qu'aucun effet sur la santé n'a été démontré de manière formelle » (Source : http://orange-en-orance.orange.fr/Developpement_durable/la_recherche.html?p=4.2.5)

Enfin, les autorités nationales, comme l'Agence Française de Sécurité Sanitaire de l'Environnement et du Travail, émettent des avis de prudence tout en favorisant la poursuite des recherches pour dresser un tableau plus complet des risques sanitaires potentiels. Comment faire la part des choses, alors que nous savons que nous sommes de plus en plus massivement exposés à ce rayonnement ? Le débat est donc loin d'être clos.

On observe qu'il est difficile également de connaître la part exacte d'implication dans ces études des industriels des télécommunications, notamment par financement direct ou indirect, et donc leur influence.

Le seul principe qui devrait s'appliquer est le principe de précaution. Sans aller jusqu'à renier une technologie qui a apporté énormément à notre société, il paraît évident que l'abus d'ondes électromagnétiques est à proscrire. Où commence l'abus ?

Suffit-il d'éteindre nos téléphones, nos ordinateurs, nos oreillettes et nos serveurs Wi-Fi quand leur fonctionnement n'est pas indispensable ?

(110331)

[1] www.elektor.fr/110759/
Télémessure de batteries LiPo par ZigBee

[2] www.elektor.fr/effetsnocifs/

L'auteur

Né en 1969, l'année où l'homme a posé le pied sur la Lune, je me suis intéressé dès l'adolescence à l'électronique et l'informatique. Après un bac F2 (l'ancêtre des bacs STI, Sciences et Techniques Industrielles), un BTS électronique et une année de licence, j'enseigne cette discipline depuis 16 ans à des élèves de 1^{re} et terminale dans divers établissements de la région parisienne.

Nouveaux microcontrôleurs 8 bits avec logique configurable intégrée dans des boîtiers de 6 à 20 broches



Grâce aux nouveaux microcontrôleurs 8 bits PIC10F/LF32X et PIC12/16F/LF150X de Microchip, il est possible d'ajouter des fonctionnalités, réduire la taille et diminuer les coûts et la consommation énergétique de vos systèmes destinés à des produits peu coûteux ou jetables, intégrant des cellules logiques configurables (CLC), un générateur de formes d'ondes complémentaires (CWG) et un oscillateur à commande numérique (NCO).

Grâce aux cellules logiques configurables CLC, la configuration logicielle de logique combinatoire et séquentielle est possible, ce qui permet d'ajouter des fonctionnalités, de diminuer le nombre de composants externes et de réduire la taille du code. Par ailleurs, le générateur de formes d'ondes complémentaires CWG permet une meilleure efficacité en commutation pour de nombreux périphériques, tandis que l'oscillateur à commande numérique NCO autorise la commande de la fréquence linéaire et une meilleure résolution pour des applications du type générateur de signaux sonores et commande de ballasts d'éclairage.

Les microcontrôleurs 8 bits PIC10F/LF32X et PIC12/16F/LF150X associent une faible consommation énergétique à un oscillateur interne intégré 16 MHz, un CAN, un module indicateur de température, et jusqu'à quatre périphériques PWM, le tout dans des boîtiers compacts de 6 à 20 broches.

Pour tout savoir sur les microcontrôleurs PIC® à faible nombre de broche intégrant des périphériques nouvelle génération, rendez-vous sur : www.microchip.com/get/eunew8bit

OUTILS DE DÉVELOPPEMENT POUR UN DÉMARRAGE RAPIDE



Kit de développement Lab
PICDEM™ - DM163045



Plateforme d'évaluation « F1 »
PIC16F193X - DM164130-1



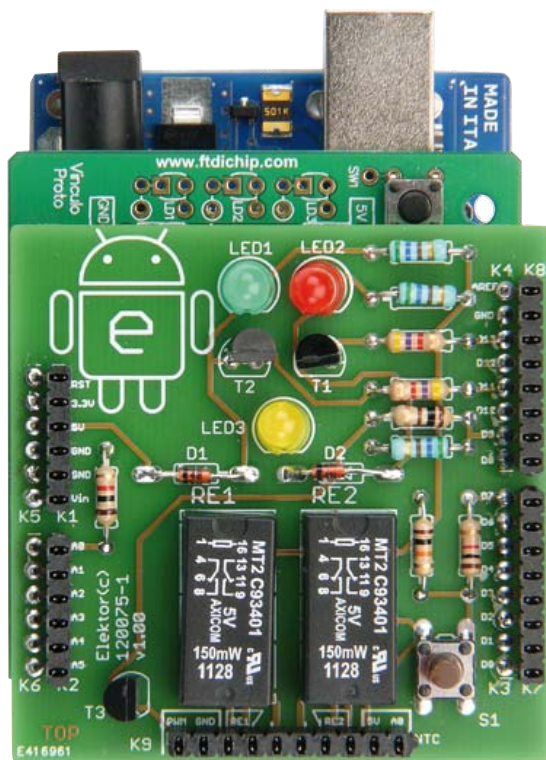
Carte de démonstration PICkit™ faible
nombre de broche DM164120-1

Outil de configuration CLC gratuit :
www.microchip.com/get/euclctool

Android + Bluetooth + Arduino

Un téléphone pour télécommander vos circuits à microcontrôleurs

Jos van Kempen



Même un modèle bon marché de téléphone *Android* peut faire la plus somptueuse interface ou télécommande pour un circuit à microcontrôleur. Découvrez ici comment, à l'aide d'une platine *Arduino* et d'une application *Bluetooth*, réaliser quelques élégants systèmes, sans fil, de commutation et de mesure.

Puis nous programmerons des applications *Android* et nous découvrirons les indispensables logiciels gratuits pour PC.

Rares sont les systèmes de commande à microcontrôleur dépourvus d'interface utilisateur : il y a forcément des boutons-poussoirs ou des encodeurs rotatifs, des LED et souvent un afficheur à cristaux liquides. Et pourquoi se priverait-on d'un écran tactile ? Ou d'une télécommande omnidirectionnelle ? Parce que ce serait du luxe, donc forcément cher ? Parce qu'on recule devant le travail pour réaliser de tels circuits ? On aurait tort. Une belle interface avec télécommande n'est ni onéreuse ni difficile à programmer. Vous disposez probablement déjà d'un ordiphone ou *smartphone*, avec *Bluetooth* : ce sera votre base. Certaines cartes à microcontrôleur sont équipées d'origine de la fonction *Bluetooth* ; pour d'autres, il existe des adaptateurs ou *shields*. Nous avons utilisé pour cet article une petite platine *Arduino* avec son adaptateur *Bluetooth* (environ 15 €)

et nous avons aussi développé un petit adaptateur spécialement pour ce projet.

Elektor a déjà publié des articles à propos du matériel à ajouter pour réaliser la fonction *Bluetooth* (en septembre 2004 et janvier 2010). Un autre article de juin 2011 a décrit la programmation d'une carte à microcontrôleur *Android*, et en décembre 2011, une commande *Android* par la sortie audio. Nous vous recommandons la (re)lecture de ces articles.

De nombreux lecteurs sont ferrés sur la programmation des microcontrôleurs, mais pas nécessairement sur celle d'un téléphone. Aussi aborderons nous brièvement la première, mais approfondirons la programmation sous *Android*, depuis le téléchargement du

logiciel jusqu'à la mise au point des différentes composantes d'une interface. Avec ces informations, vous serez en mesure ensuite de la personnaliser avec le microcontrôleur de votre choix. Le code source des applications décrites ici, tant pour le microcontrôleur que pour le téléphone, est disponible au téléchargement [1].

Matériel et logiciel du microcontrôleur

On installe sur la platine *Arduino* un adaptateur *Bluetooth*. Il communique par l'interface *UART*. On peut, en *BASCOM*, au moyen d'instructions telles que *input* et *print*, échanger avec la platine *Arduino* des commandes et des résultats de mesure.

Pour ce projet, nous avons donc conçu un adaptateur simple, avec quelques LED et des relais sur les sorties numériques, un FET sur la sortie MLI analogique, un interrupteur pour une entrée numérique et une thermistance CTN comme capteur analogique. La figure 1 vous en montre le schéma, la platine est sur la figure 2. Cette interface permet d'essayer toutes les possibilités de communication par *Bluetooth* entre un téléphone *Android* et une carte *Arduino*. Le connecteur K9 met à votre disposition les raccordements pour tous les relais, interrupteurs et capteurs.

Explorons la structure du programme en *BASCOM* (listage 1).

On vérifie dans une boucle si un caractère est arrivé. Si l'*Arduino* reçoit un **R**, la sortie D11 s'active, la LED1 s'allume et le relais RE1 travaille. Si c'est un **r** qui entre, la LED1 s'éteint et RE1 est mis au repos. Il en va de même pour la sortie D13 sur laquelle sont branchés la seconde LED et le relais, sauf que ceux-ci réagissent aux commandes **O** et **o**. Après réception avec *Input* d'un **P**, comme dans PWM, une valeur est attendue (suivie par un code de fin de ligne « r/n/ »), après quoi

cette valeur est utilisée pour la commande de la sortie MLI. Le FET T3 peut débiter 0,5 A sous 60 V. La LED3 fournit une indication du taux de modulation.

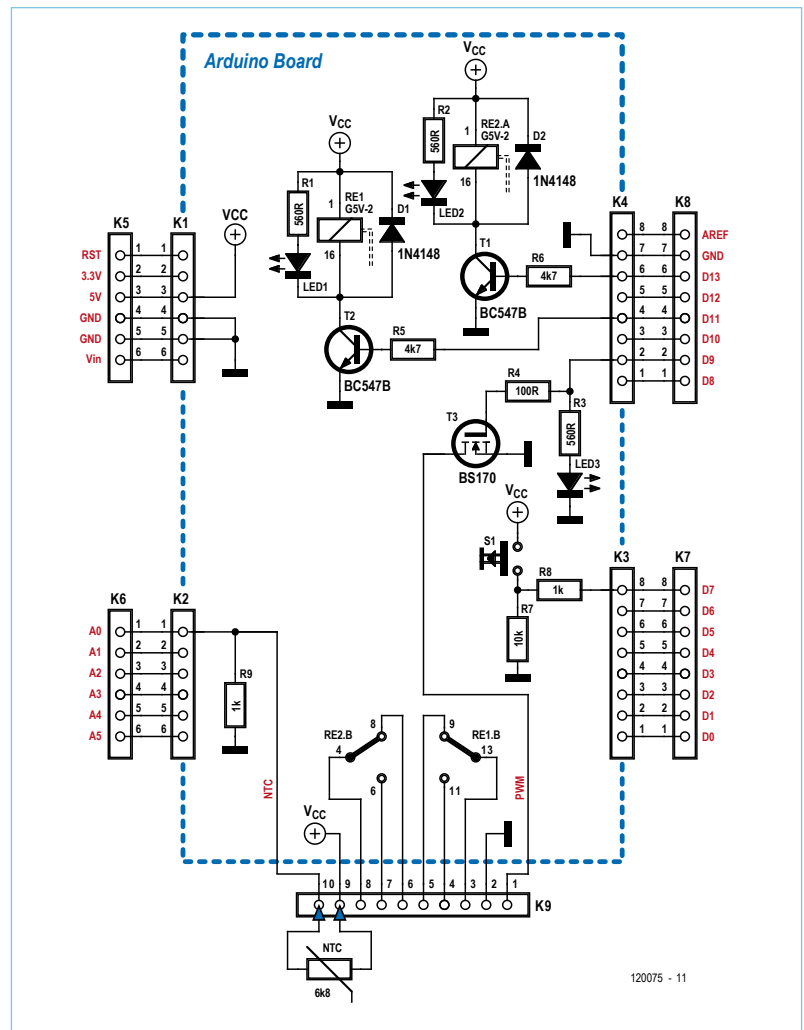


Figure 1. Le schéma de l'adaptateur avec ses LED et relais, une CTN, un bouton-poussoir, une sortie MLI avec FET et un témoin à LED.

Liste des composants

Résistances :

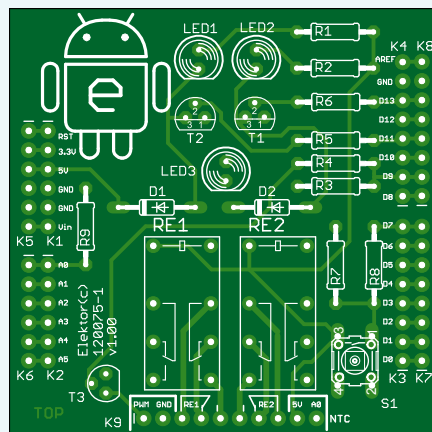
R1, R2, R3 = 560 Ω
R4 = 100 Ω
R5, R6 = 4,7 kΩ
R7 = 10 kΩ
R8, R9 = 1 kΩ

Semi-conducteurs :

D1, D2 = 1N4148
LED1, LED2, LED3 = LED rouge, 5 mm
T1, T2 = BC547
T3 = BS170

Divers :

K1, K2 = embase à 6 broches
K3, K4 = embase à 8 broches
K5, K6 = embase femelle à 6 contacts



K7, K8 = embase femelle à 8 contacts

K9 = embase femelle à 10 contacts

RE1, RE2 = relais miniature 5 V (p.ex. TE connectivity MT2-C93401 ou OMRON G5V-2-H1)

S1 = bouton-poussoir normalement ouvert (p.ex. B3F-1000)

Platine 120075-1 (voir [1])

Figure 2.

Le tracé des pistes est arrangé pour que les connecteurs s'adaptent au standard de la platine *Arduino*.

Listage 1. Le programme dans le microcontrôleur a été rédigé en BASCOM.

```

$baud = 9600 : Ucsr0a = &H00                                'Thanks to J.F. Theinert
'Software under CC-BY-NC-SA licence by Jos van Kempen.
Config Adc = Single , Prescaler = Auto , Reference = Avcc    'ADC (analog) input initialize
Start Adc                                                    'PWM (analog) output initialize
Config Timer1 = Pwm , Pwm = 8 , Compare A Pwm = Clear Down , Prescale = 8
Config Pinb.1 = Output                                       'PB1 =digpin9=pwm1a
Pwm1a = 0
Dim Pwm_str As String * 5                                    'text 0-255
Dim Pwm_b As Byte
Dim Value As Integer
D13 Alias Portb.5 : Config D13 = Output                      'Dig13 no resistance needed for LED
D11 Alias Portb.3 : Config D11 = Output                      'Dig11
D7 Alias Pind.7 : Config D7 = Input                          'Initialize DigInput Dig7
Declare Sub Set_pwm
Dim B As Byte
Do
  B = Ischarwaiting()
  Print B
  If B = 1 Then                                              'if incoming command
    B = Waitkey()
    Select Case B
      Case "R" : D11 = 1
      Case "r" : D11 = 0
      Case "O" : D13 = 1
      Case "o" : D13 = 0
      Case "P" : Call Set_pwm
    End Select
  End If
  Waitms 300
  Pwm_b = Pwm_b + 3
  Value = Getadc(0)                                          'A0
  Print "T" ; Value ; "t"
  Waitms 40
  If D7 = 1 Then Print "G" Else Print "g"
  Waitms 30
Loop
End
Sub Set_pwm
  Input Pwm_str Noecho
  Pwm_b = Val(pwm_str) : Waitms 30
  Pwm1a = Pwm_b
  Print "*" ; Pwm_b : Waitms 30
End Sub

```

Par ailleurs, au cours de chaque cycle, deux valeurs sont mesurées et imprimées. Il s'agit de celle du capteur NTC R9 sous la forme T;adc(0);t et celle de l'entrée D7 à laquelle est relié le poussoir S1 : **G** pour un 1 et **g** pour un 0. La communication se déroule à 9 600 bauds, après quoi une courte pause, ici entre 30 et 40 ms, est nécessaire.

Quels logiciels utiliser ?

Pour la programmation d'une application sur téléphone *Android*, on trouve des logiciels gratuits.

1. On programme en langage *Java*, avec le kit de développement *Java JDK* disponible sur le site d'*Oracle* [2].
2. On trouve le kit de développement de logiciel SDK pour *Android* sur un site [3] qui propose aussi de nombreuses informations sur la programmation, des fichiers d'aide et également des pilotes

USB pour votre téléphone. Ceux-ci vous permettront ultérieurement de télécharger des applications, à moins que vous ne les possédiez déjà sur votre PC si vous pratiquez l'échange par USB avec votre téléphone, de photos, de musique, etc.

3. Téléchargez du même site le *plugin* ADT pour *Eclipse* et retenez le nom du répertoire où vous l'aurez placé.
4. Comme environnement de développement (IDE), on utilise *Eclipse Classic 3.6.2* [4]. Le *plugin* le plus récent au moment d'écrire ces lignes est l'ADT12, incompatible avec *Eclipse 3.7*.
5. Après l'installation d'*Eclipse*, il faut installer les *plugin Android* en suivant *Help | Install new software | Archive*. Choisissez le fichier ZIP de l'ADT et appelez-le *Android plugin*. Passez ensuite dans *Windows | Preferences* et cherchez le fichier extrait d'*Android SDK* (sous-répertoire de *Tools*). Allez alors dans *Windows | Android SDK and AVD Manager* et choi-

sissez *New (Virtual Device)*, p. ex. *Gingerbread* pour la version 2.3.3 (ou *Samsung_GIO* ou équivalent) qui peut vous servir à simuler le programme sur le PC sans téléphone.

6. Téléchargez le répertoire de l'exemple de projet *Bluetoothinterface* [1], mais ne le mettez pas dans l'espace de travail d'Eclipse.

Bluetoothinterface

Programmer une fonction qui, par *Bluetooth*, trouve un autre appareil *Bluetooth*, prend contact avec lui puis échange des messages, est une tâche ardue. Par chance, un exemple en est fourni avec l'Android SDK. Malheureusement, il semble qu'il ne fonctionne pas avec de nombreux appareils *Bluetooth*. On trouve sur la toile un remplaçant pour l'un des fichiers, *BluetoothRfcommClient.java*, qui peut résoudre le problème après adaptation des déclarations [5] [6]. Vous pouvez alors utiliser le projet pour personnaliser l'interface. Après le lancement d'Eclipse, on peut réaliser un nouveau projet à l'aide de *File|New|Android project|* en se servant éventuellement d'un exemple, mais dans ce cas, il existe déjà dans l'espace de travail un projet que nous voulons importer. C'est ce qu'on fait avec *File|Import...|Existing Projects in Workspace*. N'oubliez pas de cocher *Copy projects into workspace*, sinon vous en perdrez l'original. Entrez alors dans le répertoire où vous avez placé le projet téléchargé *Bluetoothinterface*.

Dans l'arborescence de ce projet, vous allez découvrir que différents fichiers .java et .xml sont déjà présents (fig. 3).

Ceux qui nous importent sont le programme principal *BluetoothChat.java*, l'interface pour le téléphone *main.xml* et la déclaration des variables dans l'interface *strings.xml*

Réalisation de l'interface

D'un double-clic sur *main.xml*, on fait apparaître l'interface sur le téléphone (fig. 4). Il est aisé, au moyen des onglets sous l'interface, de passer rapidement de l'interface elle-même au code produit automatiquement.

La disposition horizontale prédéfinie de l'interface reprend, entre autres, une liste des appareils *Bluetooth* détectés, un éditeur de texte pour les sessions de *chat* et un bouton pour expédier du texte. Pour en faciliter la mise en place, il sera souvent plus commode de faire glisser directement un nouveau composant vers la fenêtre *Outline*. Si vous modifiez les dimensions en les faisant glisser sur l'écran alors que le signe décimal du PC est la virgule plutôt que le point, vous risquez d'obtenir un message d'erreur. Vous devrez alors, dans le fichier *main.xml* remplacer la virgule par un point.

Lors de la création d'un bouton-poussoir ou d'un bouton radio, pour définir la procédure à exécuter quand on cliquera dessus, donnez le nom de cette procédure dans la propriété *On click*.

On commande les LED et relais des sorties numériques sur l'adaptateur au moyen de boutons ou des coches placés côte à côte. Pour cela, il y a dans l'écran ou dans la partie *Outline* (c'est souvent plus facile) une structure horizontale (*linear layout*) qui comporte des boîtes à cocher ou des boutons à bascule.

Pour changer, la position de l'interrupteur sur l'entrée numérique est représentée dans l'interface par deux boutons radio *On* et *Off*

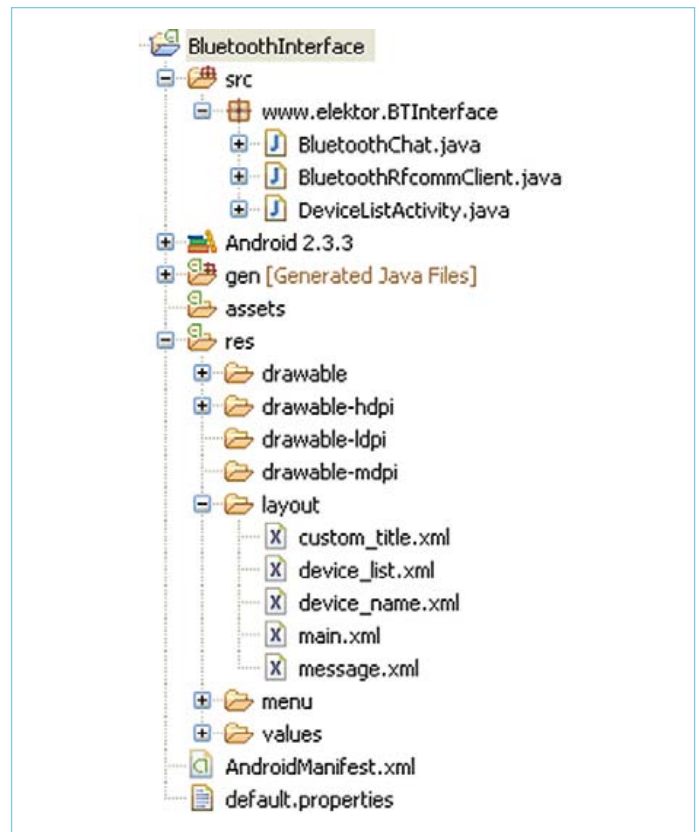


Figure 3. La création des fichiers dans le projet d'interface *Bluetooth*.

(*Radiogroup orientation horizontal*).

La valeur de l'entrée analogique s'affiche au moyen d'une barre de progression horizontale (*Style Horizontal*) et un *textView*, ainsi que sous forme graphique (*image View*).

La sortie MLI (PWM) avec le FET se commande par une glissière (*SeekBar*).

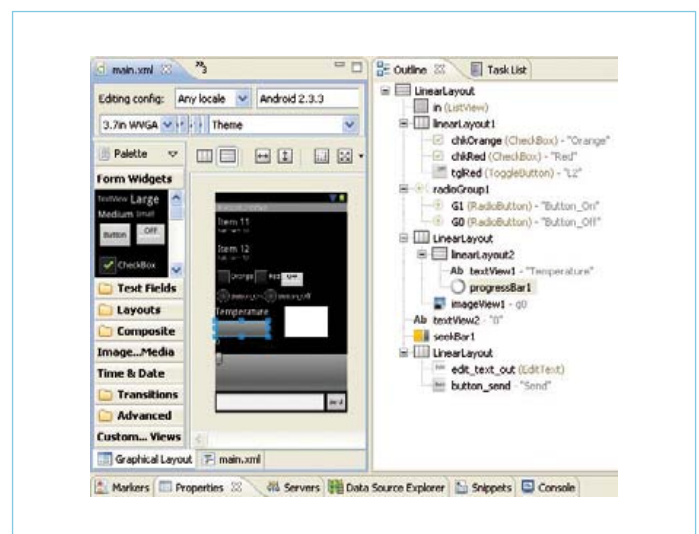


Figure 4. Le dessin de l'écran de l'interface. Il est souvent plus facile d'installer les organes de commande dans l'écran *Outline*.

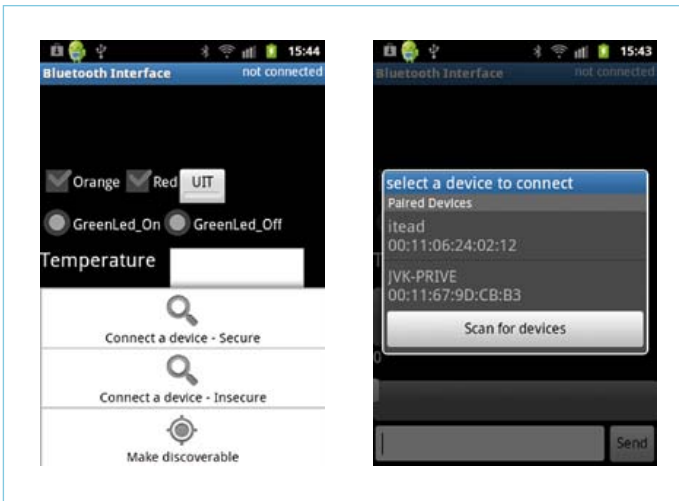


Figure 5. Choisissez *Connect a device – Secure*, après quoi vous pourrez sélectionner l'un des appareils déjà appariés. Vous pouvez aussi rechercher des appareils déclarés détectables. Pour l'appariement (unique), vous devez fournir un code, 0000 ou 1234, par exemple.

Nous n'avons encore rien programmé, pourtant nous pouvons déjà voir à quoi l'interface va ressembler. Il y a aussi une possibilité de simulation sur le PC, avec un téléphone virtuel à l'écran, mais ceci ne permet pas de simuler le *Bluetooth*.

Reliez à présent par un câble USB votre téléphone au PC et sélectionnez le dossier de projet correspondant, cliquez du bouton droit de la souris et choisissez *Run As... | Android Application*. L'application est compilée puis transférée sur votre téléphone.

Si l'émetteur *Bluetooth* est allumé, vous recevrez toutes les informations, qui s'afficheront sur votre téléphone.

Essayez aussi, en communiquant avec une clé *Bluetooth* sur un PC, de faire envoyer et recevoir des données via un programme de terminal ; dans ce programme, p. ex. *Hyperterminal* ou *Advanced Serial*



Figure 6. L'interface au travail. Au-dessus, vous voyez les communications envoyées par Itead (l'adaptateur *Bluetooth*) ou Me (le téléphone). En dessous, vous pouvez encore envoyer manuellement des commandes. Des options pratiques pour le dépannage, mais que vous pouvez bien sûr remplacer pour d'autres applications.

Port Terminal, il faudra commencer par sélectionner le bon port COM et établir la liaison.

Après avoir transféré et lancé l'application, établissez la liaison avec le bouton de menu en bas, à gauche. Après l'invite *connect to device – secure*, choisissez l'appareil *Bluetooth* voulu. Si ça marche, l'avis *connecting* s'affiche et, un peu plus tard, « *connected: nom d'appareil* » (fig. 5).

Dans le fichier *BluetoothChat* le code ressemble à cet exemple :

```
private CheckBox chkD13, chkD11,chkDIM;
// (Si, dans le programme, les propriétés de la boîte
// à cocher sont lues ou écrites, il faut que la
// variable soit déclarée)
```

```
chkD13 = (CheckBox) findViewById(R.id.chkD13);
// (Lors de la procédure 'onCreate' le lien est
// établi entre le nom dans main.xml (l'écran layout)
// et le nom dans le programme)
```

```
public void chkD13Click(View view){
    if (chkD13.isChecked()==true)
        sendMessage("O"); else sendMessage("o");
}
```

La procédure elle-même est très simple. Selon qu'il y a ou non une coche, c'est un O ou un o qui est envoyé pour que le microcontrôleur exécute l'instruction qui met dans l'état voulu la LED2 et le relais 2.

Le code pour envoyer la valeur qui correspond à la position de la barre de progression, au moment où elle est libérée, est un peu plus compliqué. Dès que vous commencez à le rédiger, l'aide Quick fix vous signale les fautes et la plus grande partie du code sera produite automatiquement. Elle ajoute aussi automatiquement à la déclaration *implements OnSeekBarChangeListener*.

```
public void onProgressChanged(SeekBar seekBar, int
progress, boolean fromUser) {
    // TODO Auto-generated method stub
    textView2.setText(Integer.toString(progress));
}
public void onStartTrackingTouch(SeekBar seekBar) {
    // TODO Auto-generated method stub
    textView2.setTextColor(Color.rgb(255, 48, 48));
}
public void onStopTrackingTouch(SeekBar seekBar) {
    // TODO Auto-generated method stub
    textView2.setTextColor(originalTextColor);
    sendMessage("P");
    try{
        Thread.sleep(1000);
    } catch (InterruptedException e) {
        // TODO Auto-generated catch block
```

```

e.printStackTrace();
}
sendMessage (seekBar1.getProgress()+
"\r\n");
}

```

Votre contribution personnelle se résumera à modifier le texte selon la valeur du décalage de la barre de progression, le changement de couleur du texte pendant que vous la déplacez et l'envoi d'un P. Il sera le signal donné au microcontrôleur d'appeler la procédure dans laquelle *Input Pwm_str* lit une ligne contenant la valeur de MLI souhaitée. Après quoi, le téléphone envoie une ligne qui en exprime la valeur, puis un code de fin de ligne.

Il existe aussi une procédure qui réceptionne les messages du microcontrôleur et les traduit en texte. Pour ce faire, on se sert du code suivant :

```

case MESSAGE_READ:
    byte[] readBuf = (byte[]) msg.obj;
    // construct a string from the valid bytes
    in the buffer
    String readMessage = new String(readBuf, 0,
        msg.arg1);
    mConversationArrayAdapter.
add(mConnectedDeviceName+
    ": " + readMessage);
break;

```

Il ne nous reste plus qu'à intercepter la communication et, selon le contenu de nos boutons radio et de la barre de progression, donner la valeur adéquate. On gère les boutons radio avec le caractère G (allumer) ou g (éteindre).

```

if (readMessage.contains("G")==true)
G1.setChecked(true);
if (readMessage.contains("g")==true)
G0.setChecked(true);

```

Pour la barre de progression, on vérifie si un code TXXXXt arrive, dans lequel les quatre X représentent un nombre entre 0 et 1023 ou bien la valeur numérisée de l'entrée analogique. Il faudra évaluer le code total, parce qu'il arrive parfois en deux morceaux, ce qui risque de conduire à une erreur dans la valeur.

```

if (readMessage.charAt(0)=='T')
if (readMessage.contains("t")==true) {
    Tempoud=Temp;Temp=readMessage.indexOf("t");
    OldMessage=readMessage.substring(1,Temp);
    Temp=Integer.parseInt(OldMessage);
}

```

Avec une valeur sous forme de texte (pour la température réelle, pour la barre de progression et le graphique), il y a lieu dans certains cas de la recalculer en fonction de l'échelle et de la formater.

```

RTemp=Temp;RTemp=RTemp/6;// 0-255 vers grdc
// jamais plus grand que le maximum
de la barre de progression
A0.setProgress(Temp*4);
//échelle barre de progression
Temp=Temp/2; //échelle graphique
DecimalFormat formatter = new DecimalFormat("#.#");
textView1.setText(formatter.format(RTemp));

```

On place aussi la valeur de mesure dans le graphique, lequel est effacé quand le nombre de mesures atteint 150. Après quoi, on redessine le cadre et les repères.

```

xcoordoud=xcoord;xcoord+=1;
if (xcoord==1){ //dessiner cadre et niveaux
    paint.setColor(Color.BLUE);
    canvas.drawLine(0,105,150,105,paint);//10 degrés
    canvas.drawLine(0,85,150,85,paint);//15 degrés
    canvas.drawLine(0,65,150,65,paint);//20 degrés
    canvas.drawLine(0,45,150,45,paint);//25 degrés
    canvas.drawLine(0,25,150,25,paint);//30 degrés
    paint.setColor(Color.YELLOW);
    canvas.drawRect(1, 1, 149, 124, paint);
}
if (xcoord==150) {xcoord=0;
    canvas.drawColor(Color.BLACK);}
else
{canvas.drawLine(xcoordoud,
    125-Tempoud,xcoord,125-Temp,paint);
}
OldMessage=readMessage;

```

Pour empêcher que l'application ne démarre avec un bouton du clavier, on a inscrit *soft input mode stateHidden* dans *Androidmanifest | Application | Window*.

Voilà pour l'essentiel. Pour vous permettre maintenant de personnaliser votre interface, nous avons ajouté dans le téléchargement un apk complet, à copier sur votre téléphone. Installez-le et essayez toutes les possibilités décrites.

(120075 - trad. Robert Grignard)

Liens

- [1] www.elektor.fr/120075
- [2] www.oracle.com/technetwork/java/javase/downloads/
- [3] <http://developer.android.com/sdk/index.html>
- [4] www.eclipse.org/downloads/
- [5] <http://projectproto.blogspot.com/2010/09/android-bluetooth-oscilloscope.html>
- [6] <http://code.google.com/p/android-bluetooth-oscilloscope/>

Retour aux sources (3)

Mettez la main à la patte

Burkhard Kainka

L'électronique devient-elle trop complexe pour vous ? Restez branché, cette série d'articles nous ramène aux fondamentaux. En mesurant ici différentes grandeurs caractéristiques d'un transistor, vous verrez qu'un simple multimètre analogique peut rendre votre transistor très bavard.

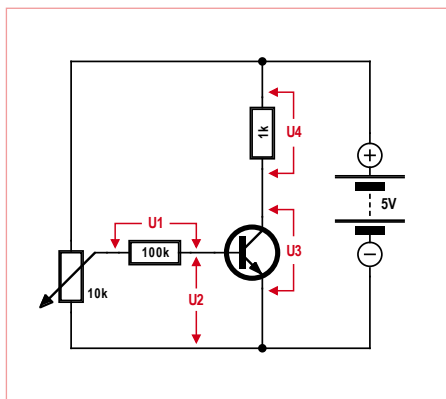


Figure 1. Le montage de mesure.

Tableau : valeurs mesurées avec un transistor BC547B

	U ₁	I _B	U ₂ =U _{BE}	U ₃ =U _{CE}	U ₄	I _C	$\beta = I_C/I_B$
1	0 V	0 μ A	0 mV	5 V	0 V	0 mA	0
2	0 V	0 μ A	400 mV	5 V	0 V	0 mA	0
3	0,07 V	0,7 μ A	573 mV	4,9 V	0,1 V	0,1 mA	143
4	0,15 V	1,5 μ A	595 mV	4,8 V	0,2 V	0,2 mA	133
5	0,26 V	2,6 μ A	612 mV	4,6 V	0,4 V	0,4 mA	153
6	0,47 V	4,7 μ A	629 mV	4,2 V	0,8 V	0,8 mA	170
7	0,90 V	9,0 μ A	646 mV	3,4 V	1,6 V	1,6 mA	177
8	1,77 V	17,7 μ A	665 mV	1,8 V	3,2 V	3,2 mA	181
9	2,63 V	26,3 μ A	679 mV	0,3 V	4,7 V	4,7 mA	179
10	3,54 V	35,4 μ A	681 mV	0,15 V	4,85 V	4,85 mA	137
11	4,32 V	43,2 μ A	683 mV	0,13 V	4,87 V	4,87 mA	113

Le comportement de votre transistor est inscrit dans les courbes caractéristiques données par sa fiche technique. Étudiez-les, c'est instructif, mais quoi de plus formateur que de mettre la main à la... patte ? L'intimité physique complète l'intimité théorique. Mesurez vous-même vos transistors, vous ne les comprendrez que mieux. Les grandeurs intéressantes sont le cou-

rant de base I_B , le courant de collecteur I_C , la tension base-émetteur U_{BE} , et la tension collecteur-émetteur U_{CE} . Si pour faire ces mesures on ne dispose que d'un seul multimètre, plutôt que de passer constamment sur cet appareil du mode voltmètre au mode ampèremètre, il est commode de ne l'utiliser qu'en voltmètre et de ne mesurer que des tensions ; l'intensité des cou-

rants de base I_B et de collecteur I_C pourra être déduite de la valeur des tensions et des résistances présentes.

Sur le montage de la **figure 1**, modifiez la tension d'entrée à l'aide du potentiomètre. Choisissez des valeurs intermédiaires entre 0 V et 5 V. Pour chacune d'entre elles, mesurez et notez les tensions U1 à U4. À partir

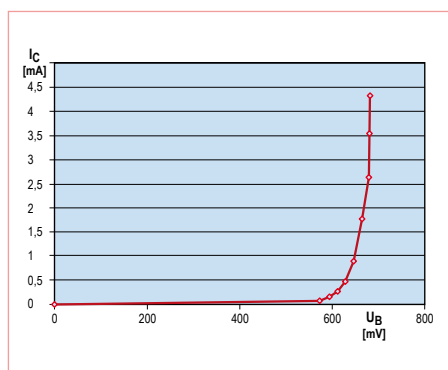


Figure 2. Le courant de base en fonction de la tension base-émetteur.

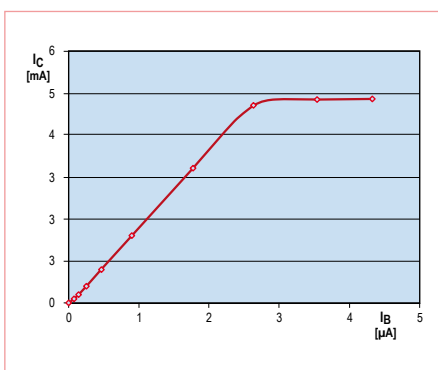


Figure 3. L'intensité du courant de collecteur dépend du courant de base.

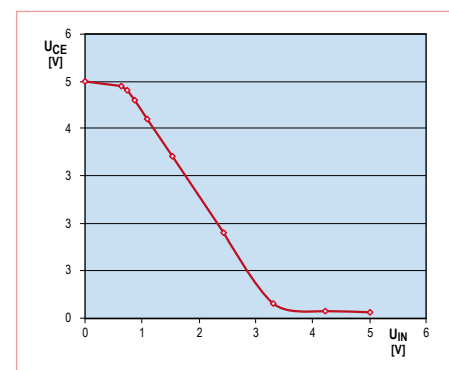
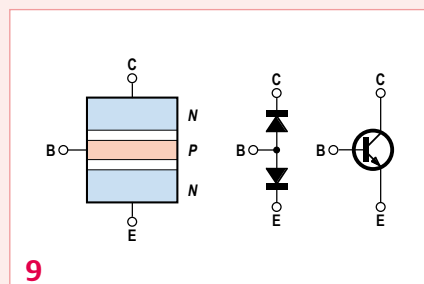


Figure 4. La tension de sortie comme fonction de la tension d'entrée.

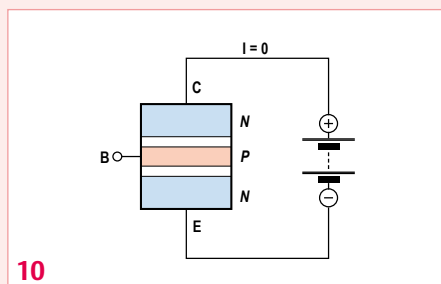
Structure et fonctionnement du transistor

Un transistor est un semi-conducteur à trois électrodes, principalement utilisé comme amplificateur de courant. Composé comme la diode de matériaux semi-conducteurs dopés N et P, il possède une structure à trois couches ordonnées selon les séquences NPN ou PNP. La **figure 9** montre la constitution et le schéma équivalent d'un transistor NPN. Remarquez que ses 3 couches forment deux jonctions PN en opposition.

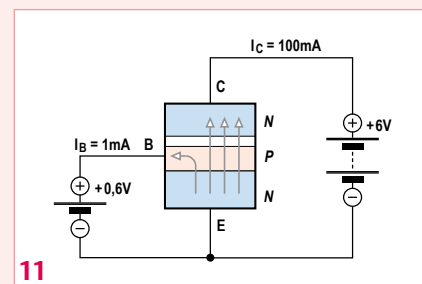
La couche médiane est la base (B), les deux autres sont l'émetteur (E) et le collecteur (C). Reliez tout d'abord le transistor à une source de courant : laissez la base en l'air, et reliez l'émetteur au pôle moins de la source (**figure 10**). Aucun courant ne circule, car la jonction base-collecteur est polarisée en sens inverse.



9



10



11

Connectez ensuite une seconde source de courant entre la base et l'émetteur en reliant la base au pôle plus de la source. Sa tension devra être inférieure à 0,6 V pour que seul un faible courant traverse la diode base-émetteur. Un courant plus intense circule maintenant entre l'émetteur et le collecteur. L'explication tient à l'extrême minceur de la base, une qualité essentielle au bon fonctionnement d'un transistor. Les porteurs de charge de type N sont en effet soumis au fort champ électrique de la jonction base-collecteur dès qu'ils pénètrent dans la base. La plupart de ces porteurs sont entraînés vers le collecteur. Seuls un pour cent de ceux quittant l'émetteur atteignent la base (**figure 11**). Le courant de collecteur est donc quelque cent fois plus intense que le courant de base. Et vous comprenez maintenant pourquoi ce courant est commandé par la tension base-émetteur ou, dit autrement, par le courant de base. Les électrons se déplacent de l'émetteur vers le collecteur, mais en pratique on parle d'un courant allant du collecteur à l'émetteur pour rester en accord avec le sens conventionnel du courant (du plus vers le moins).

de ces valeurs, calculez la valeur des deux courants et du gain β . Le **tableau** montre les résultats obtenus avec un BC547B et trois multimètres numériques reliés à U1, U2 et U3. La grandeur U4 a été calculée à partir de U3, puis l'intensité des courants et le gain.

Conseils pratiques

Comment procéder ? Il est pratique de commencer avec un courant de **collecteur** de 0,1 mA ($U_4 = 0,1$ V), puis de doubler ce courant à chaque nouvelle mesure (U_4 vaudra donc successivement 0,1 V, 0,2 V, 0,4 V, 0,8 V, etc), jusqu'à ce qu'il n'augmente plus. Vos mesures devraient suivre une loi de variation intéressante : aux erreurs de mesure près, le courant de **collecteur** est multiplié par 2 chaque fois que le courant de **base** double. L'accroissement correspondant de la tension base-émetteur est constant : environ 20 mV.

Le gain en courant est donné par le rapport I_C / I_B . D'après le tableau, sa valeur maximale est d'environ 180. Or, en théorie, nous aurions dû atteindre un facteur d'amplification d'au moins 200. Comment expliquer cet écart ? Par la présence de sources d'erreur. Ainsi la résistance interne

du multimètre (10 M Ω) prélève-t-elle une petite partie du courant de base lors de la mesure de U2. Les erreurs de mesure sont normales. Une fois l'ensemble des sources d'erreur et des tolérances (y compris celles des résistances) prises en compte, le gain maximal de notre transistor approcherait 200. Essayez avec le vôtre, peut-être trouverez-vous un gain supérieur.

Qu'elle soit faite avec un papier et un crayon ou le tableur d'un PC, une représentation graphique des données de mesure facilite leur interprétation. Voici ce que nous avons obtenu :

La courbe de la **figure 2** (I_C en fonction de U_B) est similaire à la caractéristique exponentielle d'une diode au silicium : coude vers 0,6 V, puis montée en flèche (exponentielle) du courant. Que nous dit-elle d'autre ? Par exemple que le courant de base – et avec lui le courant de collecteur – n'était pas encore mesurable lorsque la tension de la base valait 400 mV. D'où l'on retiendra que la plage utile de cette tension de base se situe pour l'essentiel entre 0,6 V et 0,7 V.

La **figure 3** (I_C en fonction de U_B) montre qu'en première approximation le cou-

rant de collecteur croît d'abord en même temps que le courant de base, avant de s'infléchir au moment d'atteindre 5 mA ; le transistor est dit saturé lorsque le courant de collecteur n'augmente plus. Si les 5 mA ne sont pas atteints, c'est que la résistance de collecteur limite le courant à 5 mA ($5 \text{ V} / 1 \text{ k}\Omega = 5 \text{ mA}$). Le transistor se comporte comme un interrupteur « à peu près » fermé : il subsiste entre collecteur et émetteur une tension résiduelle d'un peu plus de 0,1 V.

Remarquez aussi que la pente est plus faible (et donc le gain d'autant moins élevé) du côté des très petits courants. Le gain diminue en effet légèrement pour de très petits et très grands courants de collecteur, effet amplifié par une insidieuse erreur de mesure : lors de la mesure de U2 circule un petit courant de mesure qui, lorsque les intensités mesurées sont petites, fait paraître le courant de base plus grand qu'il n'est en réalité.

La **figure 4** montre la tension de sortie (U_{CE}) en fonction de la tension d'entrée ($U_1 + U_2$) définie par la position du curseur du potentiomètre. L'interprétation est immédiate :

une tension d'entrée plus élevée entraîne une tension de sortie plus petite. La raison en est tout aussi claire : lorsque le courant de collecteur augmente, la chute de tension aux bornes de la résistance de collecteur s'accroît.

Contre-réaction

Comment dimensionner son montage lorsque le gain en courant n'est pas connu avec précision ? Rien de plus simple dans le cas d'un étage de commutation. Il suffit de choisir un courant de base pour lequel le circuit fonctionnera encore avec le plus petit gain en courant possible. Dans le doute, augmentez le courant de base : le circuit sera adapté à tous les transistors d'un type donné.

L'affaire est tout autre lorsqu'il s'agit d'amplifier une grandeur analogique. Un courant de base trop fort nous jouerait ici de mauvais tours, car le transistor ne tarderait pas à entrer en régime de saturation. L'idéal, le plus souvent, est un courant de collecteur d'intensité moyenne, qui pourra alors être augmentée ou diminuée. Pour obtenir une telle marge de manœuvre même avec des transistors aux caractéristiques différentes, on emploie une configuration appelée la contre-réaction. On l'obtient en reliant la résistance de base non pas à l'alimentation, mais au collecteur (**figure 5**). Un transistor caractérisé par un gain élevé produira alors une chute de tension plus grande aux bornes de la résistance de collecteur. La tension de collecteur chutera, et avec elle le courant de base. Inversement, les transistors à gain faible seront traversés par un courant de base de plus forte intensité. Au final, cette stratégie convient pour tous les transistors.

La fonction ohmmètre

S'ils ne font pas jeu égal avec les multimètres numériques en termes de précision, les instruments ana-

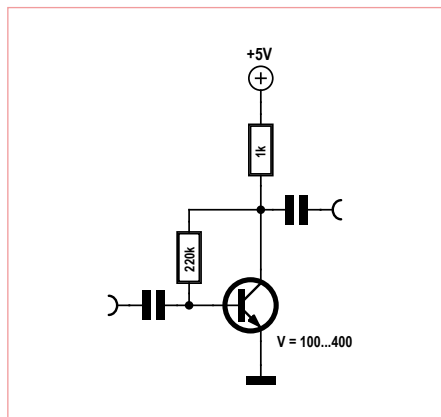


Figure 5. Polarisation par contre-réaction.

logiques à aiguille resteront votre meilleur œil pour lire rapidement un résultat, surtout lorsqu'il s'agit de tester des composants ou de localiser des pannes.

Les multimètres analogiques simples possèdent en général un ou plusieurs calibres pour la mesure des résistances. Des mains exercées sauront les utiliser aussi pour mesurer transistors, diodes ou autres condensateurs. Les piles de ces appareils

ne sont souvent utilisées que par la fonction ohmmètre. Le principe de la mesure de résistance repose sur la mesure d'un courant à tension constante, d'où il résulte que l'échelle d'un ohmmètre analogique n'est pas linéaire. Pour compenser l'usure de la pile, la déviation maximale de l'aiguille, qui correspond à une résistance nulle, doit être alignée sur le zéro de l'échelle graduée à l'aide du potentiomètre de tarage de l'appareil (**figure 6**). L'autre extrémité de l'échelle graduée correspond à des grandeurs infinies.

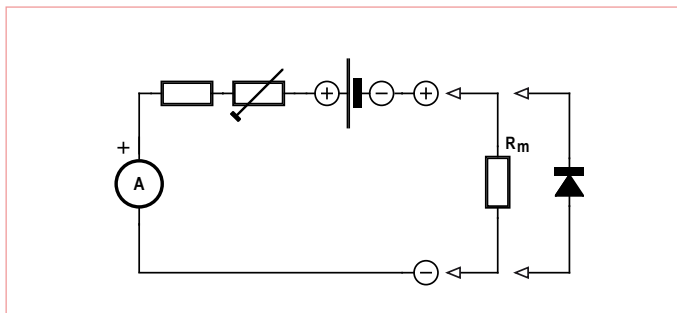


Figure 6. Schéma de principe d'un ohmmètre analogique.

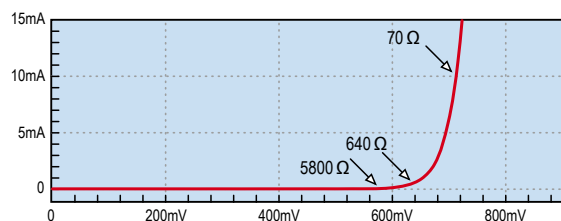


Figure 7. Résistance en continu d'une diode au silicium avec différents courants de mesure.

En fonction ohmmètre, le circuit de mesure interne d'un multimètre analogique inverse la polarité des pointes de touche par rapport à la polarité indiquée pour les modes voltmètre ou ampèremètre : le pôle *plus* correspond donc à la borne *moins* de l'appareil. Souvenez-vous de cette inversion lorsque vous utilisez un tel ohmmètre pour vérifier des diodes ou des transistors.

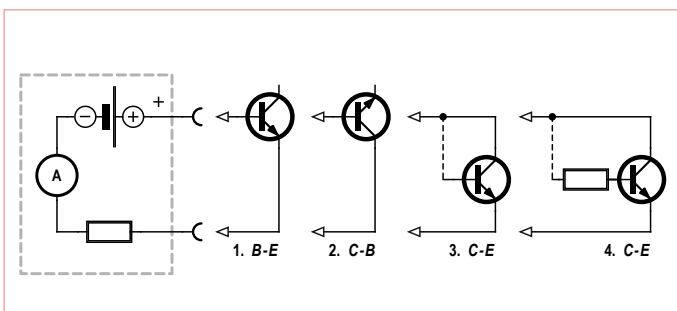


Figure 8. Des mesures sous toutes les coutures.

Lorsque vous vérifiez des diodes, gardez aussi à l'esprit qu'il est impossible d'attribuer à une jonction une valeur de résistance constante. La valeur indiquée par l'appareil de mesure dépend du courant de mesure, et donc du calibre sélectionné. Être limité par les

Testeur de transistor

Un microcontrôleur doté d'un convertisseur A/N peut faire un excellent appareil de mesure. Prenons l'exemple d'un testeur de transistor. Dans ce cas, la mission du microcontrôleur est simple : déterminer le gain en courant. Un ATtiny13 saura s'en acquitter. Le résultat peut être transmis à un PC par liaison série puis affiché sur un terminal.

Le schéma de principe de la **figure 12** est simple. Seule la tension de collecteur est mesurée. La présence d'une contre-réaction permet de mesurer des gains en courant très différents les uns des autres.

```
,Transistor tester
$regfile = „attiny13.dat“
$crystal = 1200000
$hwstack = 8
$swstack = 4          ' 16
$framesize = 4

Dim UC As Word
Dim U1 As Word
Dim U2 As Word
Dim I1 As Word
Dim I2 As Word
Dim V As Word

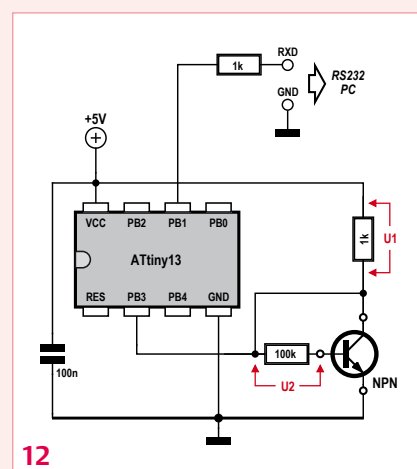
Config Adc = Single , Prescaler = Auto
Start Adc

Open „comb.1:9600,8,n,1,INVERTED“ For Output As #1

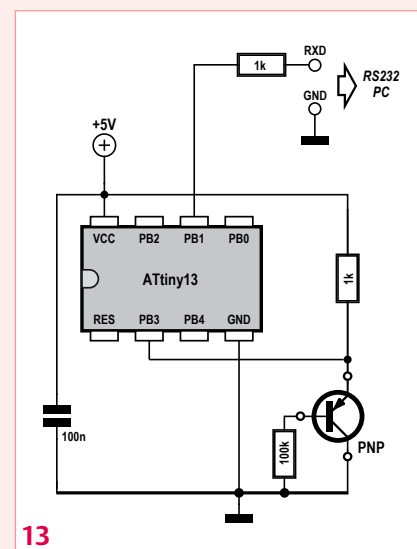
Do
  UC = Getadc(3)          ' PB3=ADC3 -> UC = 0..1023

  UC = UC * 50            ' max 51150 -> 5115 mV
  U2 = UC - 6000         ' 6000 <- U_BE = 600 mV
  U1 = 51150 - UC
  I1 = U1                ' 1 k
  I2 = U2 / 100          ' 100 k
  V = I1 / I2
  Print #1 , V           ' --> RXD
  Waitms 1000
Loop

End
```



12



13

Le programme calcule les chutes de tension U1 et U2 aux bornes des résistances de collecteur et de base. Le courant de collecteur I1 et le courant de base I2 sont déduits de ces valeurs. Le gain en courant est donné par $I1 / I2$. Les variables ont été déclarées comme entiers de type Word pour que le programme puisse s'exécuter dans la petite mémoire de l'ATtiny. Il ne peut se produire ni débordement ni perte de précision due à des résultats intermédiaires trop petits.

Ce même programme permet également de mesurer des transistors de type PNP, mais vous devrez les connecter selon le schéma de la **figure 13**.

outils de mesure n'empêche pas de raisonner : si la tension interne d'un ohmmètre est de 1,5 V, on peut déduire que lorsque l'aiguille est à mi-course, la chute de tension aux bornes du composant testé est d'environ 0,75 V. Changez de calibre de mesure et vous constaterez que la position de l'aiguille ne change qu'à peine. Ce qui s'explique par la caractéristique exponentielle des diodes. Chaque calibre indiquera donc une résistance différente ; la déviation de l'aiguille sera en revanche similaire puisque la chute de tension à ses bornes reste d'environ 0,6 V. Le résultat permet d'inférer le type de la diode. Celle de la **figure 7** devrait ainsi être une diode au silicium.

Vérifier un transistor

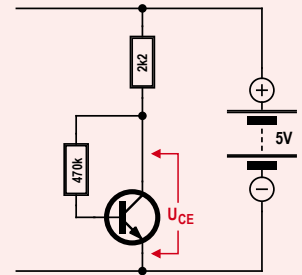
Un simple ohmmètre permet de déterminer le type et l'état d'un transistor. Même d'un transistor dont vous ignorez tout, il vous permettra de deviner le brochage. Trois mesures suffisent à faire passer à un transistor une visite médicale complète. Commencez par mesurer les diodes base-émetteur et base-collecteur (voir **encadré** : Structure et fonctionnement du transistor). Vous pourrez déjà distinguer un silicium d'un germanium et déceler la présence d'éventuels courts-circuits (**figure 8**, schémas 1 et 2). Mesurez ensuite la résistance entre l'émetteur et le collecteur, avec et sans courant de base (3). Base ouverte, un transistor intact ne laissera circuler aucun courant et présentera donc une résistance infinie. S'il existe un court-circuit entre base et collecteur, l'intensité mesurée sera légèrement supérieure à celle de la seule diode base-émetteur.

Le dernier test (4) nécessite un petit courant de base, obtenu avec une résistance base-collecteur ou, puisque généralement on en a toujours une sous la main, avec un doigt, humide, qui ira toucher la base et le collecteur. La déviation de l'aiguille donnera une idée approximative du gain en courant. Notez qu'il existera encore un faible gain si émetteur et collecteur sont inversés. En cas de doute sur les connexions, retournez donc votre transistor.

Contrairement à un ohmmètre analogique, un multimètre numérique mesure

Jeu-concours

À l'aide d'une seule mesure et sans microcontrôleur, sauriez-vous déterminer le gain en courant d'un transistor ? En voici un doté d'une résistance de collecteur de 2,2 k Ω et d'une résistance de base de 470 k Ω reliée au collecteur ; la tension de sortie est notée U_{CE} .



1) Vous mesurez $U_{CE} = 2,8$ V. Quel est le gain ?

- A) environ 152
- B) environ 214
- C) environ 472

2) Vous mesurez $U_{CE} = 0$ V. Qu'en déduisez-vous ?

- D) Le transistor est grillé (second claquage)
- E) La résistance de base n'a pas de contact
- F) La résistance de base vaut 470 Ω et non pas 470 k Ω

3) Vous mesurez $U_{CE} = 4,9$ V. Quelle peut-être l'origine de cette valeur ?

- G) Coupure au niveau du fil du collecteur
- H) Coupure du fil de la base
- I) Émetteur et collecteur sont intervertis

Si vous avez trois bonnes réponses, vous gagnerez peut-être par tirage au sort un kit « Minty Geek Electronic 101 Kit » !

Envoyez avant le 31 mars 2012 votre réponse dans l'objet d'un message électronique adressé à basics@elektor.com, sous forme d'un code composé des lettres correspondant à vos trois réponses, à l'exclusion de toute autre mention. L'objet du message sera lu automatiquement, le corps du message ne sera pas lu du tout..

La solution du précédent jeu-concours sera donnée dans le prochain numéro !

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

la résistance par la chute de tension à courant constant. L'affichage est cette fois-ci linéaire, la plage de mesure est finie, et le réglage du zéro inutile. Autre différence par rapport aux instruments à aiguille, une polarité identique pour toutes les fonctions de l'appareil.

En principe, la fonction ohmmètre d'un numérique permet les mêmes tests de composants que ceux effectués avec un appareil

analogique. Un multimètre numérique possède souvent une fonction supplémentaire pour tester les diodes ; l'appareil indique alors une chute de tension en millivolts ou une valeur proportionnelle à la tension de diode.

(120003 - trad. Hervé Moreau)

RL78 GREEN ENERGY CHALLENGE



Bientôt! Le défi Énergie Verte RL78

Renesas a fait l'impossible pour réduire au strict minimum la consommation des microcontrôleurs de sa famille RL78, qui offrent jusqu'à 41 DMIPS (millions d'instructions par seconde) à 32 MHz, au sein d'un remarquable écosystème de matériel, de logiciel et de fournisseurs tiers. Ces processeurs polyvalents sont au cœur d'une plateforme à faible consommation, idéale pour les applications embarquées à 8 ou 16 bits les plus exigeantes.

Renesas s'associe à *Circuit Cellar* et à *Elektor* pour vous inviter dans l'univers de l'énergie verte : autour du contrôleur RL78 et de la chaîne d'outils IAR, concevez une application remarquable par sa faible consommation. Ne laissez pas échapper cette chance de remporter un des lots d'une valeur totale de 20.000\$ en espèces!



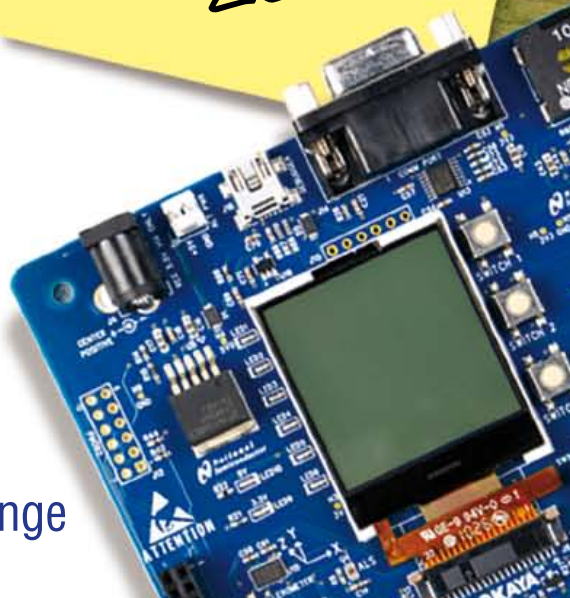
Ce concours est organisé en association avec *Elektor* et *Circuit Cellar*.

Colorez l'électronique en vert et relevez le défi
ÉNERGIE VERTE RL78.

www.circuitcellar.com/RenesasRL78Challenge

20.000\$
en
espèces !

Le concours
sera lancé
le 26 mars
2012



Indicateur de charge pour génératrice de secours

Alimentation domestique sans coupure

Aucun réseau électrique n'est à l'abri d'un aléa météorologique. Et aucune génératrice de secours n'est à l'abri d'une surcharge. Grâce à cet indicateur de charge, votre foyer ou votre résidence secondaire sera toujours correctement alimenté en électricité.

Chuck Hansen (États-Unis)

Au nord-est des États-Unis, où j'habite, des tempêtes de pluie verglaçante et des cyclones entraînent de graves pannes de courant. J'ai donc acheté une génératrice de secours portable, de 2500 W et d'une capacité de surtension de 2800 W. Elle peut alimenter four, réfrigérateur, chauffe-eau, ainsi que prises et éclairages de plusieurs pièces. Mon électricien avait installé un tableau secondaire de transfert d'alimentation de 60 A pouvant relier nos charges au réseau public ou à une source CA additionnelle. Ce tableau possède huit disjoncteurs, et deux indicateurs de charge gradués de 0 à 2500 W qui auraient permis de mesurer la puissance fournie par la géné-

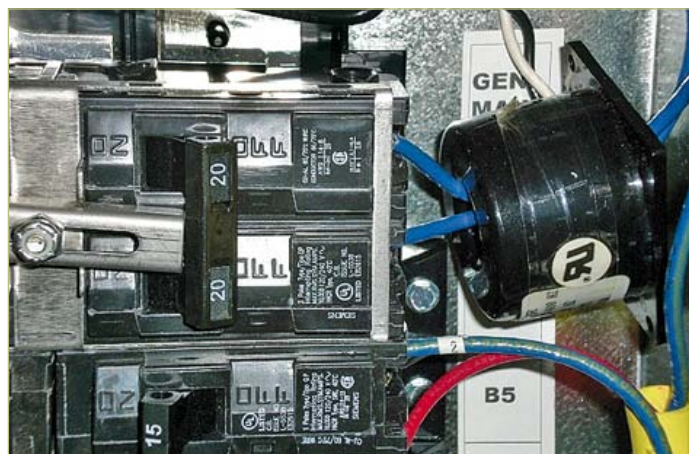


Figure 1. Logement du TC dans le tableau de transfert.



ratrice, mais il est situé au sous-sol, et je voulais connaître depuis l'étage la charge absorbée pour mieux estimer le moment de ravitailler la génératrice.

Fonctionnement

Le tableau de transfert peut être câblé pour la distribution américaine en monophasé 3 fils (120-0-120 V_{CA}), ou pour l'alimentation de la génératrice monophasée (120 V_{AC}) en branchant en parallèle les deux barres omnibus. Les indicateurs du tableau de transfert sont hors service lorsque le commutateur de transfert est relié au tableau électrique principal.

Pour garantir la compatibilité avec les éléments de réglage électronique de nos appareils, j'ai choisi une génératrice qui associe une génératrice à aimant permanent haute tension à un onduleur à signal sinusoïdal monophasé. Elle réduit son nombre de tr/min en mode « économie », et repasse à plein régime lorsque la charge l'exige.

Pour relier la sortie de la génératrice à la maison, je me suis servi d'un câble d'alimentation de 4,6 m et 8 mm², ainsi que d'un connecteur quart-de-tour utilisé pour alimenter un camping-car en 30 A monophasé.

J'ai acheté sur eBay un transformateur de courant (TC) de rapport 150/1, homologué UL et CE, donc conforme à nos codes locaux. Mon électricien a passé à travers le noyau du TC les deux câbles bleus qui alimentent les huit disjoncteurs du tableau de transfert, et ce de façon additive (figure 1). L'intensité du courant secondaire est égale à celle du primaire divisée par le rapport de transformation, ici 150. La puissance nominale du TC est de 5 VA. Il peut donc produire sans saturation une tension secondaire efficace théorique d'environ 31 V_{eff} (charge de 200 Ω) avec la surcharge admissible nominale de 23,33 A de la génératrice. C'est plus que suffisant pour l'indicateur de charge distant.

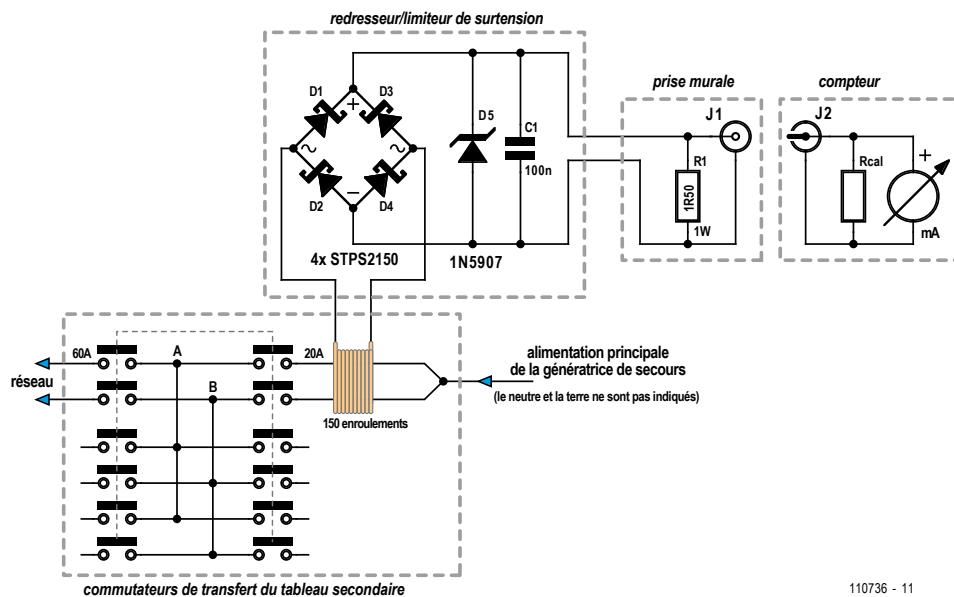


Figure 2. Schéma de principe du tableau de transfert et de l'indicateur de charge.

Publicité

LabWorX 1 Mastering the I²C Bus

➡ Nouveau livre d'Elektor en anglais

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

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

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



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

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

www.elektor.fr/e-choppe

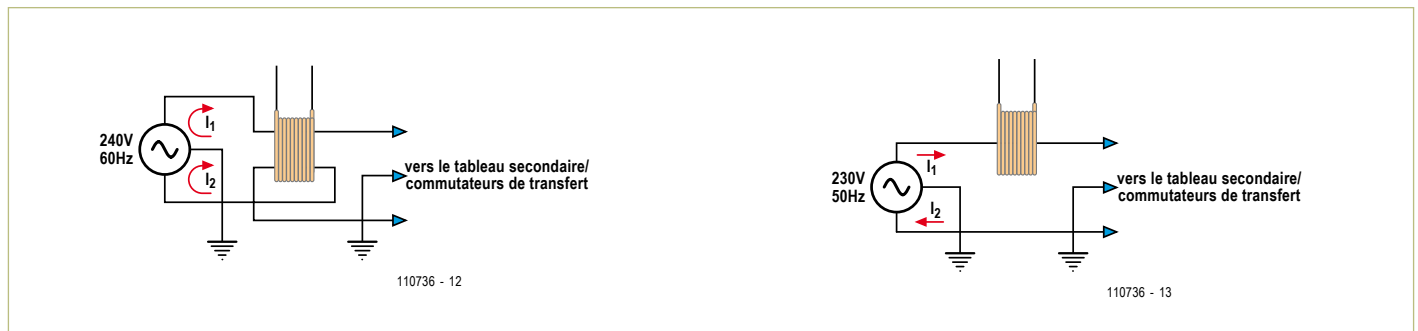


Figure 3a et b. Les deux configurations du TC en fonction du réseau électrique.

Schémas de principe

La **figure 2** montre le schéma de principe du tableau de transfert et de l'indicateur de charge. Notez que pour des raisons de clarté les fils de neutre et la mise à la terre obligatoire ne sont pas indiqués sur le schéma du tableau. Le TC est situé du même côté que la génératrice.

J'ai conçu un boîtier en plastique pour loger le limiteur de surtension et le redresseur. Le secondaire du TC lui est relié. Son bobinage torsadé minimise les champs magnétiques parasites. Les composants sont montés sur une petite platine d'essai. Les deux alternances du courant secondaire sont redressées par les quatre diodes Schottky D1 à D4. La diode D5, une supprimeuse de surtension 1N5907, permet de limiter la tension secondaire au cas où serait accidentellement coupé le câble à deux conducteurs auquel est reliée la résistance de charge de l'indicateur.

Même si le secondaire d'un TC peut sans dommage être mis en court-circuit puisqu'il s'agit d'une source de courant, la tension secondaire serait dans ce cas multipliée par le produit du rapport de transformation et de la chute de tension qui règne le long de l'enroulement primaire situé à l'intérieur du noyau. La tension de court-circuit du secondaire pourrait ainsi être mortelle : le noyau en acier au silicium possède une grande perméabilité magnétique, et la chute de tension aux bornes du primaire sera donc significative pour des valeurs de charges élevées. Le condensateur de 100 nF (C1) filtre toute pointe de tension susceptible d'apparaître dans le signal du secondaire.

R1 est la résistance de charge de 1,50 Ω du TC aux bornes de laquelle est produit le signal de l'indicateur. Elle est située dans une prise murale du rez-de-chaussée, à côté du connecteur BNC (j'ai modifié une prise vidéo standard de type F de 75 Ω). Puisqu'il s'agit d'une application basse tension, le code permettait d'installer dans le mur un coffret à dos ouvert pour monter la prise murale. Ce connecteur BNC ne sera pas confondu avec les autres connecteurs utilisés (RCA, vidéo, RJ-44, Ethernet, etc).

Mesures

Pour l'indicateur de charge, j'ai choisi un appareil de mesure analogique gradué de 0 à 100 mA_{CC}. Je l'ai calibré pour qu'il indique les variations de charge de la génératrice de 0 à 100 %. Le boîtier pupitre en aluminium (voir la **photo**) contient un connecteur d'entrée BNC, l'indicateur de charge, ainsi qu'une résistance R_{cal} branchée aux bornes de l'indicateur, lui relié à la prise murale par un court câble BNC. La tension aux bornes des redresseurs pleine onde ainsi que celle du câblage qui va à la prise murale de l'étage ne créent

aucune erreur au niveau de l'indicateur de charge puisque le TC est une source de courant.

R_{cal} permet de calibrer l'indicateur de façon à ce que la déviation maximale de 100 mA corresponde à la pleine charge de 2500 W de la génératrice. Je me suis servi de deux mini-fours et d'un compteur d'énergie EM-100 pour vérifier la calibration. J'ai utilisé une valeur de 63,4 Ω pour R_{cal} (simple correction de 2 %... mais nous autres techniciens sommes parfois quelque peu fignoleurs). J'ai marqué au feutre rouge la zone de l'indicateur qui va au-delà de 100 % pour visualiser les moments où la génératrice est en mode de limitation de courant.

La tension aux bornes de la résistance de 1,50 Ω située dans la prise murale vaut 233 mV_{CC} à 2800 W. La tension fixe aux bornes de chaque redresseur est de 300 mV (600 mV au total par alternance). Celle qui règne aux extrémités du câble de 24 m et 1,3 mm² vaut 50 mV. La tension secondaire du TC n'est donc que de 883 mV, ou 137 mVA à 2800 W, c'est-à-dire très inférieure à la puissance nominale de 5 VA. Débrancher l'indicateur n'a aucun effet sur le TC puisque la résistance de charge R1 est située à l'intérieur de la prise.

Adaptation aux réseaux 230 V/240 V européens

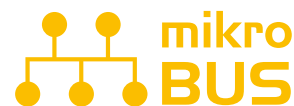
J'ai dessiné deux autres configurations : une pour le réseau électrique européen de 230 V, l'autre pour un réseau nord-américain de 240 V. Le câblage du secondaire du TC ne change pas (figure 2). Vous aurez peut-être à adapter les valeurs de R1 et de R_{cal} en fonction des tensions.

Du fait de la configuration particulière des réseaux 120/240 V_{AC} nord-américains, les courants de phase (notés I₁ et I₂ sur la **figure 3a**) sont non seulement déphasés, mais susceptibles d'être d'intensités différentes. Un des fils de phase doit donc traverser le noyau dans l'autre sens de façon à ce que les courants en opposition de phase ne s'annulent pas.

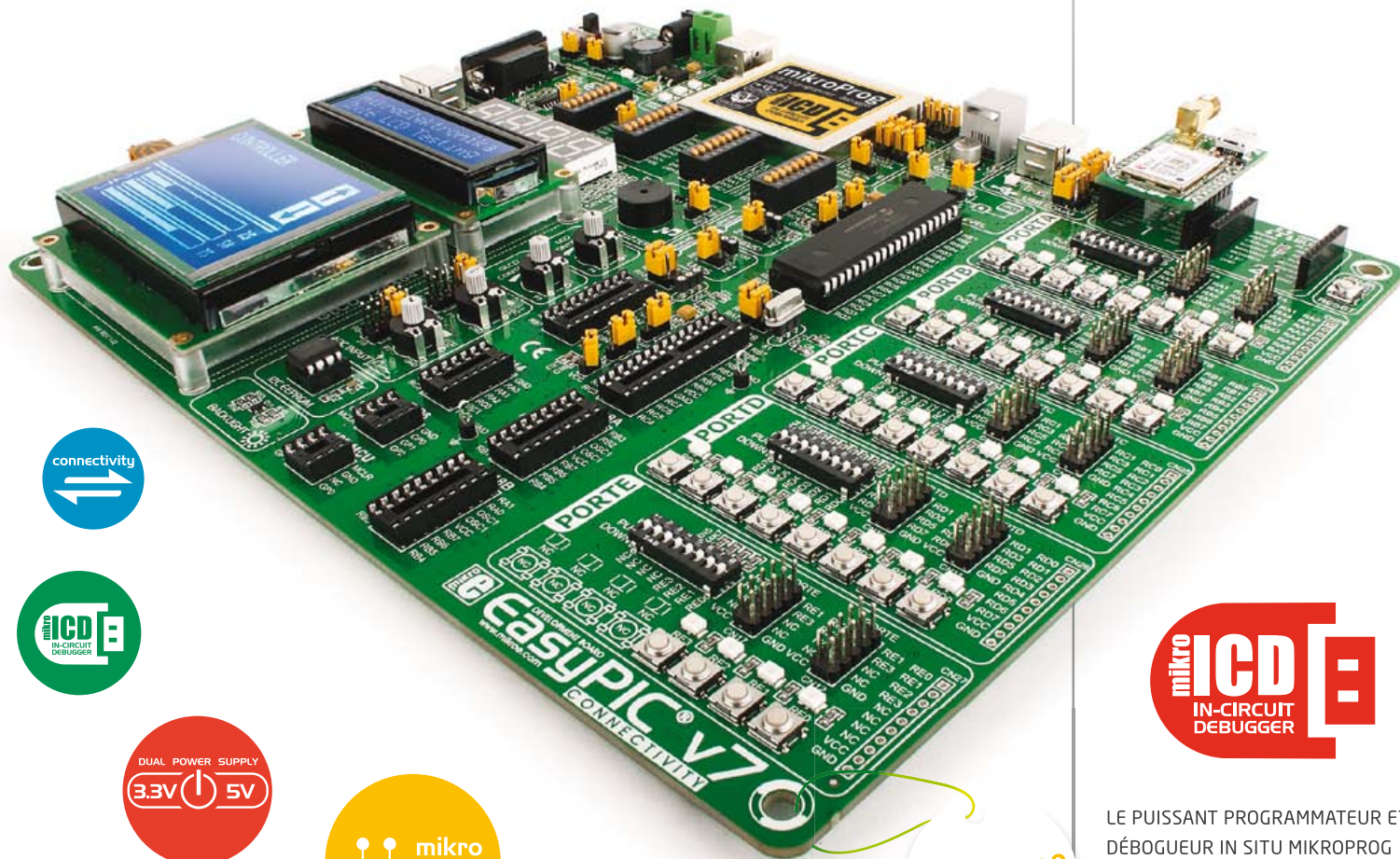
Le raccordement au réseau européen de 230 V_{CA} se fait par fils symétriques avec terre au centre (**figure 3b**), et I1 et I2 sont toujours en opposition de phase et d'intensité égale. Le TC peut être branché selon la configuration précédente (figure 3a), mais il est aussi possible de ne faire passer qu'un seul des deux fils d'alimentation à travers le noyau (figure 3b). Dans ce cas l'intensité du secondaire sera deux fois moindre, et la valeur de R1 devra être augmentée pour atteindre la déviation maximale de l'indicateur.

(110736 - trad. Hervé Moreau)

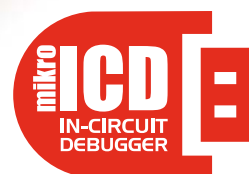
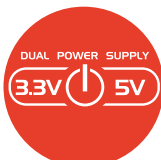
EasyPIC^{v7} connectivity



CONNECTEZ VOTRE **CLICK BOARD™**,
C'EST PRÊT ! L'INTERFACE DE
COMMUNICATION SPÉCIALISÉE
MIKROBUS FACILITE VOS
DÉVELOPPEMENTS ET REND LA
CONNECTIVITÉ AUSSI SIMPLE
QU'EFFICACE. **TOUT EST**
MAINTENANT À PORTÉE DE CLIC!



connectivity



\$149⁰⁰

Qualité, conception et fonctions ultramodernes, tout ce qui a valu à la carte de développement EasyPIC son succès mondial se retrouve dans cette version 7. Sa **connectivité vous étonnera** : 4 connecteurs pour chacun des ports, groupés de façon logique avec leurs LED et boutons associés. Son puissant programmeur et débogueur in situ **mikroProg** gère plus de 250 dispositifs de **3,3 V et 5 V**. Trois types d'afficheurs, EEPROM série, deux capteurs de température, buzzer piézo, broches GND pour oscilloscope, RS-232 et FTDI, connecteur USB et **prise en charge du mikroBus** rendent cette station de développement PIC irremplaçable.

LE PUISSANT PROGRAMMATEUR ET
DÉBOGUEUR IN SITU MIKROPROG
PROGRAMME TOUS LES
**MICROCONTRÔLEURS PIC10, PIC
12, PIC 16 ET PIC18**. QUE VOUS
SOYEZ PROFESSIONNEL OU
DÉBUTANT, SA PRISE EN MAIN
FACILE ET SES PERFORMANCES
EXCEPTIONNELLES VOUS SERONT
INDISPENSABLES.

DUAL POWER SUPPLY

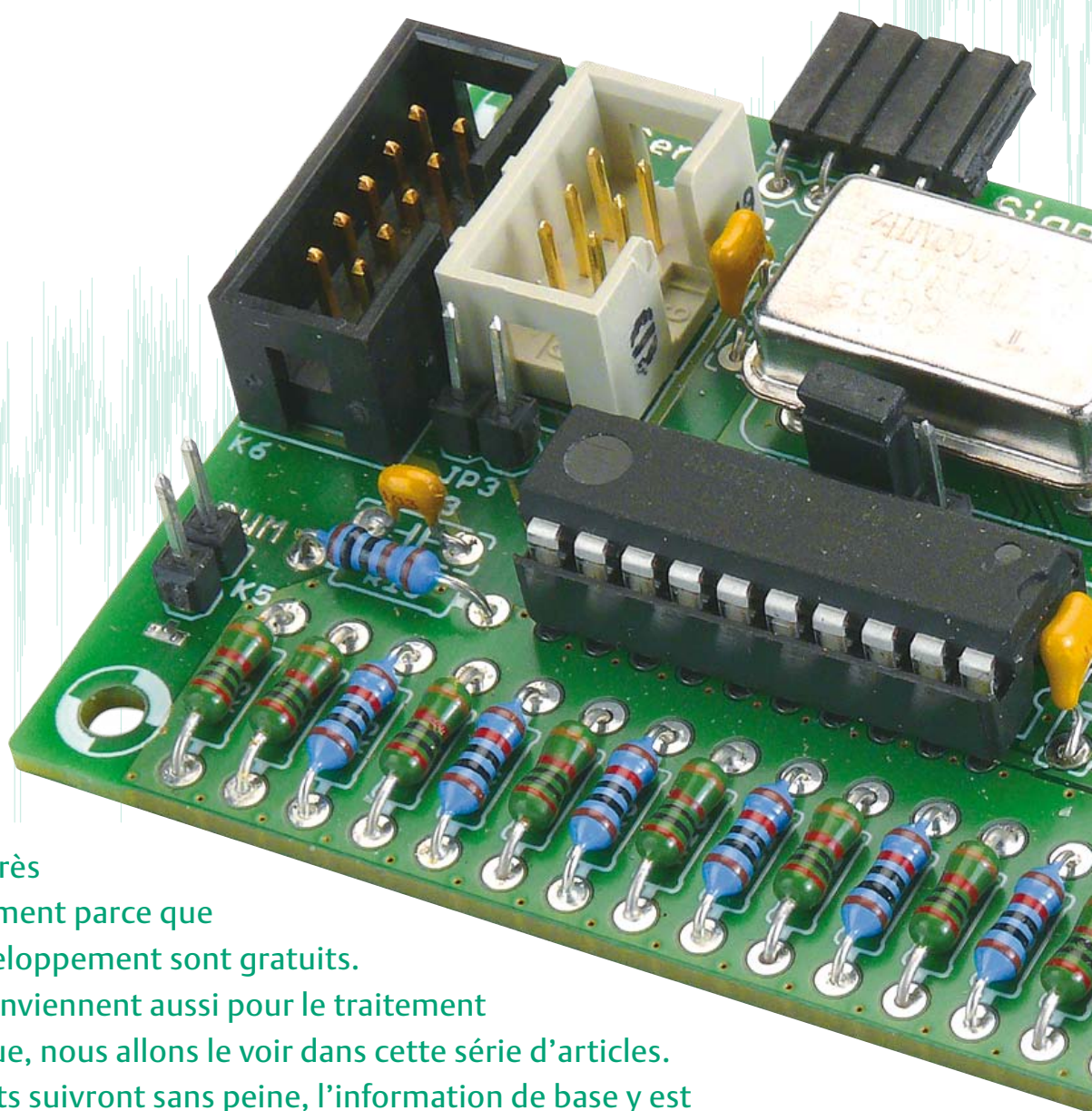


EASYPIC V7 EST LA SEULE CARTE
DE DÉVELOPPEMENT À GÉRER À
LA FOIS LES MICROCONTRÔLEURS
DE 3,3 V ET 5 V. **PLUS DE 250
MICROCONTRÔLEURS** SONT PRIS
EN CHARGE SUR UNE CARTE
UNIQUE GRÂCE À UNE TECHNIQUE
RÉVOLUTIONNAIRE. EASYPIC, LA
CARTE MULTI-CARTES!

Radio logicielle avec

L'ATtiny en générateur de signal ultra

Martin Ossmann



Les contrôleurs

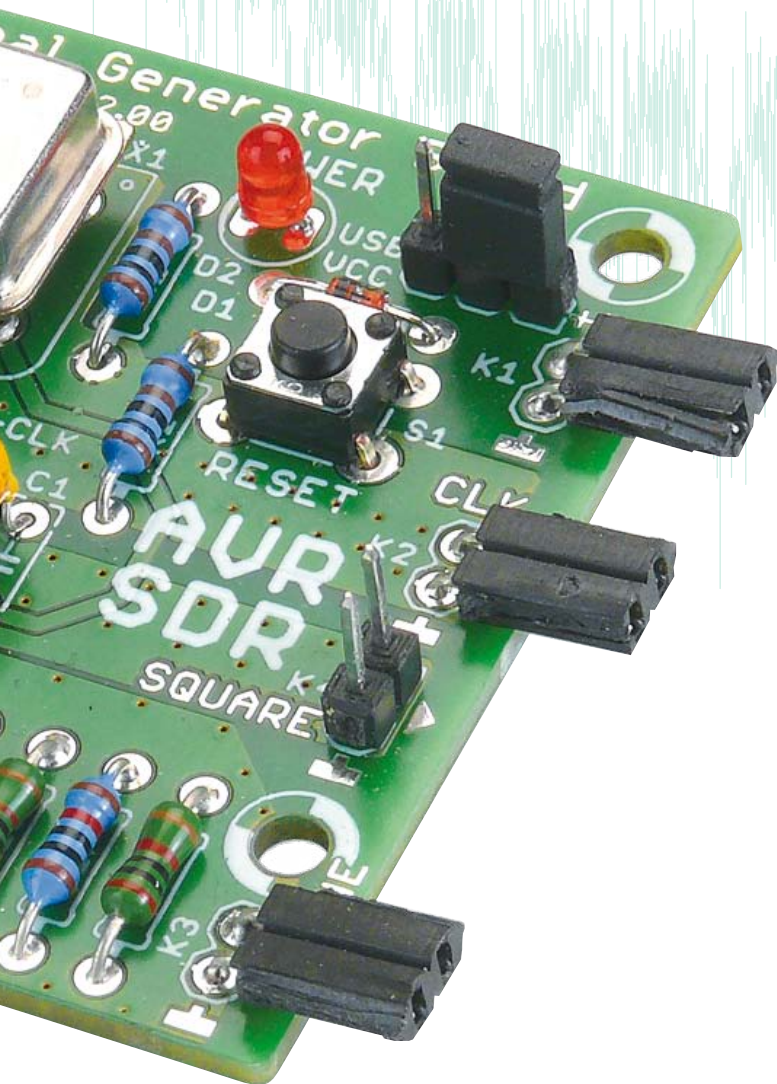
AVR d'Atmel sont très demandés, notamment parce que leurs outils de développement sont gratuits.

Ces processeurs conviennent aussi pour le traitement du signal numérique, nous allons le voir dans cette série d'articles.

Même les débutants suivront sans peine, l'information de base y est expliquée simplement. Fidèle à la tradition elektorienne, l'expérimental sera au cœur de l'action. Pour le matériel à construire soi-même, Elektor pense à votre confort et vous en propose une platine. Quant au logiciel, comme toujours, le code source est disponible sur notre site. En avant pour un générateur de signaux et bien davantage !

AVR (1)

précis pour SDR



Au cours de cette nouvelle série d'articles, nous ne nous limiterons pas à produire des signaux. C'est certes le rôle de la première platine présentée ici, équipée d'un ATtiny2313, d'un oscillateur à 20 MHz et d'un convertisseur numérique/analogique (N/A) à réseau R-2R ; mais nous irons aussi à la chasse aux signaux radio, avec une deuxième platine, pour former un récepteur numérique SDR, une **radio logicielle**. Sur cette première carte, une interface RS232, un afficheur à cristaux liquides ainsi qu'un VCXO à 20 MHz, à coupler plus tard à un standard de référence. La troisième platine sera une antenne ferrite active. Tout le logiciel a été rédigé avec le compilateur WINAVR-GCC en AVR-Studio. Le code source en C ainsi que les réglages des fusibles et le code hexadécimal sont sur le site d'Elektor.

La série se veut résolument expérimentale, avec des générateurs de sinusoïdes et d'ondes carrées, un voltmètre RMS, des recherches sur la modulation en fréquence, en amplitude et en phase, sur les filtres FIR et IIR, des expériences de transmission de données sans fil autant que sur la réception des signaux de DCF77, ceux des stations météo en RTTY, de différents émetteurs de la BBC en grandes ondes et encore bien d'autres explorations. Suivez le guide !

Commençons par un conseil : pour certaines de ces expériences, il faudra proscrire tout éclairage à lampes économiques genre fluo-compacte, dont l'alimentation à découpage perturbe la réception en grandes ondes. Alors pendant vos sessions de radio, éteignez-les et éclairez-vous à la bougie !

La platine du générateur de signaux

Notre générateur de signaux se compose principalement d'un microcontrôleur AVR cadencé à 20 MHz et d'un convertisseur N/A R-2R pour fournir la tension de sortie. Un tel générateur n'a rien de nouveau. Il est simple et ses applications sont innombrables, en particulier pour obtenir des signaux de forme tarabiscotée à des fins de test d'autres appareils. Il produit entre autres des signaux modulés en fréquence et en phase. Bientôt, nous asservirons le générateur de signaux par une horloge, elle-même pilotée par un standard de fréquence, comme DCF77 ou TDF à 162 kHz, ce qui en augmente singulièrement la précision.

La **figure 1** représente le circuit du générateur de signaux. La pièce centrale est évidemment le microcontrôleur ATtiny2313. Sur le port B, un réseau R-2R fait office de convertisseur N/A à 8 bits et on recueille le signal analogique (*SINE*) sur la sortie K3. Ne pas perdre de vue que l'impédance de sortie est de 10 k Ω , une valeur relativement élevée. La sortie OC1A à modulation de largeur d'impulsion (MLI ou *PWM* en anglais) du contrôleur est également accessible de l'extérieur, c'est K4 *SQUARE*. Nous pourrions nous en servir pour disposer

Produits et services Elektor

- Générateur de signaux (platine et tous les composants en kit : 100180-71)
- Convertisseur USB/TTL BOB FT232 (module câblé et testé : 110553-91)
- Programmeur USB AVR (platine avec CMS implantés plus les autres composants : 080083-71)

Téléchargement gratuit des logiciels (fichiers hex et code source)
Infos, téléchargements, produits : www.elektor.fr/100180

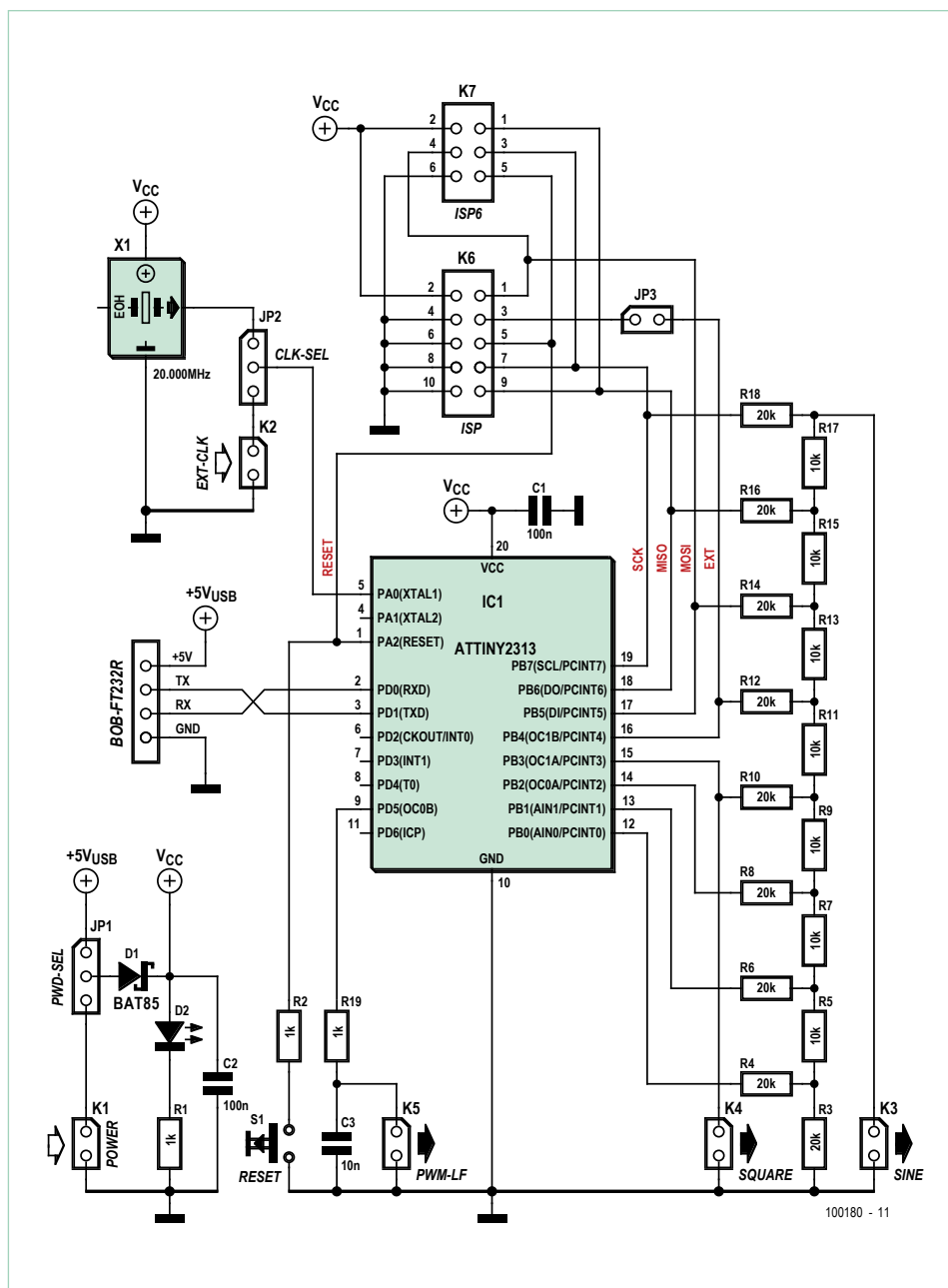


Figure 1. Le schéma du générateur de signaux.

de signaux rectangulaires de quelques centaines de kilohertz, par exemple pour moduler d'autres signaux. D'autre part, la sortie MLI OC0B est reliée à travers le filtre passe-bas R19/C3 au connecteur K5 (*PWM-LF*), ce qui nous fournit une sortie analogique supplémentaire pour des signaux lents.

Le processeur est cadencé par un oscillateur à quartz X1 de 20 MHz. Il est utile de sélectionner un exemplaire précis, à 50 ppm ou mieux. Pour essayer des oscillateurs de différents types ou de différentes fréquences, mieux vaut se servir d'un support. Le cavalier JP2 permet de passer sur un signal d'horloge externe à brancher sur le connecteur K2 (*EXT-CLK*).

On peut configurer en partie le programme du générateur de signaux de l'extérieur en se servant de l'UART du contrôleur. Sur la platine, les broches nécessaires sont rassemblées sur un connecteur

sur lequel on pourra brancher le convertisseur USB/série BOB-FT232R [1], ce qui facilite le raccordement à un PC. Avec JP1, on peut choisir une alimentation par l'USB quand l'appareil en a besoin, ce qui épargne un adaptateur secteur supplémentaire.

Pas de CMS, pas de souci ! L'implantation des composants conventionnels sur la platine (**fig. 2**) est facile. En plus du générateur d'horloge, il convient d'installer aussi le processeur dans un support. Attention à l'orientation correcte des connecteurs de programmation K6 et K7. Lors de la programmation, qui peut s'effectuer p. ex. à l'aide du petit USB-AVR-Prog [2] d'Elektor, n'oubliez pas les fusibles de bits de configuration. Dans ce but, les fusibles et les options du compilateur apparaissent dans le code source.

Le générateur de signal DSS

Notre première application sera la programmation en C d'un simple générateur de sinus. Un temporisateur du contrôleur, déclenché par interruption, fournit l'horloge d'échantillonnage fondamentale. Dans la routine d'interruption, il faut donc toujours produire la valeur de l'amplitude actuelle de l'échantillon (**fig. 3**). Nous appellerons $S[k]$ l'échantillon de rang k . Si $p[k]$ est la phase correspondante, nous aurons :

$$S[k] = \sin(p[k])$$

D'un échantillon à l'autre, la phase progresse chaque fois d'une valeur constante d , appelée **incrément de phase**. Donc :

$$P[k+1] = p[k] + d$$

Sur un générateur de sinus idéal, tous les calculs devraient être effectués avec exactitude, mais cela prendrait trop de temps. Aussi utilisons-nous le principe de la synthèse numérique directe (DDS pour *Direct Digital Synthesizer*). La phase actuelle **DDSp** est ainsi mémorisée avec une précision finie, sous forme d'un nombre de **m** bits, dans un accumulateur de phase. Une période complète de sinusoïde s'étale alors à l'intérieur d'une fourchette de 0 à $2^m - 1$. La présentation et l'addition des incréments de phase s'effectuent avec la même précision. Il faut encore déterminer, pour chaque angle de phase, l'amplitude du sinus correspondante. On peut le faire en utilisant une table dans laquelle on a stocké les valeurs de sinus d'une

Liste des composants

Résistances :

R1, R2, R19 = 1 k Ω
 R5, R7, R9, R11, R13, R15, R17 = 10 k Ω
 R3, R4, R6, R8, R10, R12, R14, R16,
 R18 = 20 k Ω

Condensateurs :

C1, C2 = 100 nF (100 V)
 C3 = 10 nF

Semi-conducteurs :

D1 = diode Schottky BAT85
 D2 = LED verte
 IC1 = ATtiny2313-20PU

Divers :

S1 = bouton poussoir (micro)
 K4, K5 = embase à 2 picots (2,54 mm)
 JP3 = embase à 2 picots (2,54 mm)
 avec cavalier
 JP1, JP2 = embase à 3 picots (2,54 mm)
 avec cavalier

K1, K2, K3 = 2x1 connecteur femelle coudé
 à 2 contacts

BOB = connecteur femelle coudé à 4 contacts
 K6 = connecteur femelle pour ISP,
 2x5 contacts
 K7 = connecteur femelle pour ISP,
 2x3 contacts
 X1 = quartz 20 MHz
 (avec 4x support Harwin H3153F01)
 BOB-FT232R-001 = convertisseur
 USB/TTL BOB (câblé et testé 110553-91)
 Platine : 100180-1
 ou kit complet 100180-71

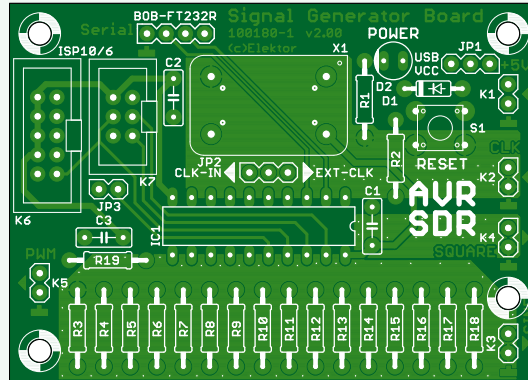


Figure 2. La platine est disponible en kit
 avec tous les composants.

période complète. Si l'on voulait emmagasiner une valeur pour chacun des 2^m angles de phase, la table serait beaucoup trop longue. On n'utilise donc que les n premiers bits pour l'adressage de la table qui, ainsi, ne doit plus compter que 2^n valeurs. Ces valeurs elles-mêmes ne sont pas non plus mémorisées complètement, mais sous forme de nombres $S[k]$ à r bits pour lesquels il ne faut plus qu'un convertisseur N/A à r bits (fig. 4).

Dans notre programme, $m=32$ et $n=8$. Avec un accumulateur de phase de 32 bits, la précision du signal obtenu est très grande, grâce au calcul exact des phases correspondantes. Nous utilisons une table de sinus de $256=2^8$ valeurs et un CN/A à 8 bits ($r=8$). Avec le programme *EXP-SinusGeneratorDDS-T1INT-V01.c* [3] la fréquence fixe réglée est de 1 kHz. On peut observer le résultat à l'oscilloscope (fig. 5) et le listage 1 montre le code de la routine d'interruption.

La chronométrie

Réglons notre horloge DDS sur $f_{\text{DDSClk}} = 100$ kHz. Pour atteindre une fréquence f qui corresponde à un incrément de phase convenable, nous devons calculer :

$$\text{DDSD} = 2^n \times f / f_{\text{DDSClk}}$$

Avec $f = 1\,000$ Hz et $f_{\text{DDSClk}} = 100\,000$ Hz, on arrive à :

$$\text{DDSD} = 2^{32} \times 1\,000 / 100\,000 = 42\,949\,673$$

Précisément la valeur que nous introduisons pour initialiser DDSD dans le programme en C.

La formule permet de constater que le taux d'échantillonnage sera d'autant plus haut que la fréquence produite est élevée. Pourquoi 100 kHz ? Pour le savoir, chronométrons d'abord la routine d'interruption. Comme on le voit dans le listage 1, nous avons, pour ce faire, encadré le code de calcul par l'activation et la désactivation de la ligne de port D.4. Ce qui permet, à l'aide d'un oscilloscope, de déterminer la durée réelle du calcul ; dans ce cas-ci, elle avoisine les 2,2 μ s. Attention, cette valeur ne représente pas la totalité du temps nécessaire au traitement d'une interruption ! En pratique il faudra aussi compter le temps de latence, requis p. ex. par la sauvegarde

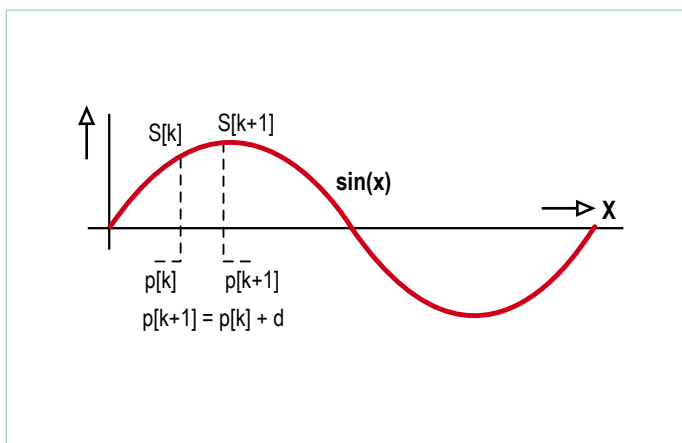


Figure 3. L'échantillonnage d'un signal sinusoïdal.

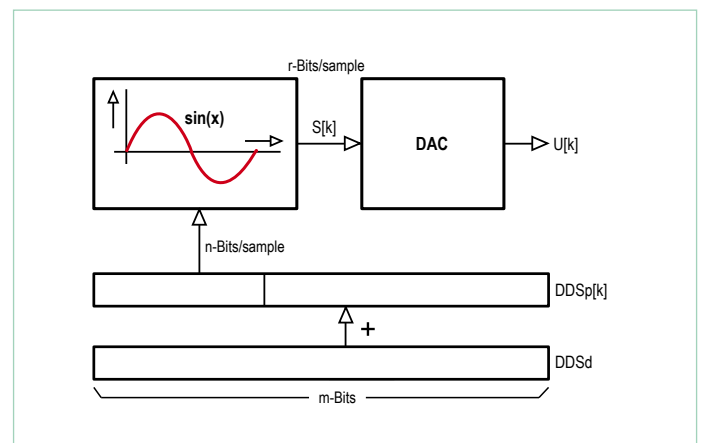


Figure 4. Le schéma du générateur de sinus DDS.



Figure 5. Test du générateur de sinus.



Figure 6. L'arrêt du basculement correspond à la durée de l'interruption.

et le rétablissement des registres.

Le programme principal est une boucle sans fin qui fait basculer une ligne de port, ici PD.5, le plus vite possible. Chaque fois qu'on voit à l'oscilloscope que ce basculement s'arrête, c'est que la routine d'interruption est active (fig. 6).

Dans notre cas, la durée totale du traitement d'une interruption est de 5,4 µs environ. La récurrence maximale reste inférieure à 180 kHz. Pour conserver une certaine marge, nous avons choisi 100 kHz.

Plus la fréquence f du générateur est haute comparée à f_{DDSClk} , plus les offenses au principe DDS dues à la gigue (*jitter*), au bruit et aux ondes parasites affecteront le spectre. En plaçant notre récurrence à 100 kHz, en pratique nous n'arrivons pas à dépasser le 10 kHz. Peut-être pourrions-nous aller plus haut avec l'assembleur ?

Un générateur de sinus turbo-DDS

Pour accélérer le générateur de sinusoïde et atteindre des fréquences supérieures, j'ai réécrit la routine DDS en langage assembleur. Avec une mise en ordre adéquate des variables dans les registres, nous pouvons atteindre une fréquence d'échantillonnage de 2 MHz pour le DDS à 32 bits. La programmation astucieuse (listing 2) utilise le drapeau T comme critère de rupture.

Dans cette combinaison des langages C et assembleur, la table de sinus doit commencer à une adresse absolue. Cette configuration à l'intérieur de WINAVR n'est pas une sinécure pour les débutants. Heureusement pour ceux qui ne désirent pas y apporter de modifications personnelles, il existe toujours la possibilité de graver dans le processeur le fichier hexadécimal tel quel, sans oublier les fusibles de bits. Le projet s'appelle *EXP-SinusGenerator-DDS-ASM-C-V01*.

Pour garder toute souplesse d'utilisation au générateur de sinus, il faut le doter d'un panneau de commande à travers l'interface UART

Listing 1

```
ISR(TIMER1_OVF_vect) {
    PORTD |= _BV(4); // start signalize timing
    PORTB=pgm_read_byte( SIN8+(DDSp>>24)) ; // fetch and output sine-sample
    DDSp += DDSd ; // advance DDS phase DDSp by DDSd
    PORTD &= ~ _BV(4); // end signalize timing
}
```

Listing 2

```
loop:
    add DDSphase0,DDSdelta0 // 1 LSB of 32 bit DDS adder
    adc DDSphase1,DDSdelta1 // 1
    adc DDSphase2,DDSdelta2 // 1
    adc ZL,DDSdelta3 // 1 MSB is in ZL as pointer
    lpm R0,Z // 3 access sine-table
    out PORTB,R0 // 1 out to R-2R DAC at PORTB
    brtc loop // 2 (1) loop until T-flag set by interrupt routine
// 10 cycles in total for one loop
```

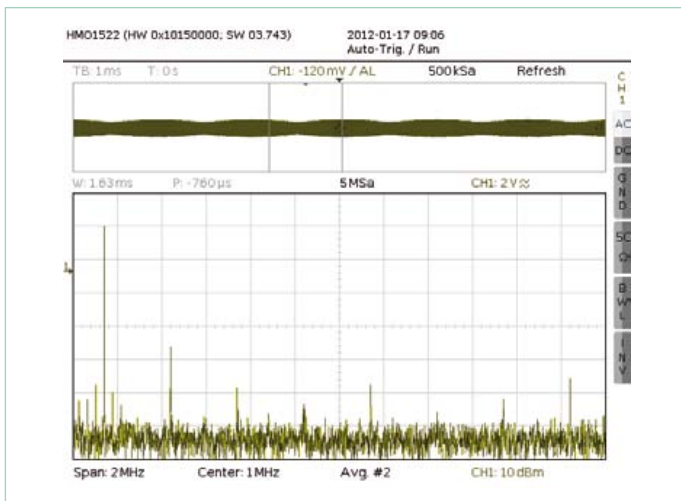



Figure 7. Le spectre du signal produit.

(19200 bauds, 8N1). Dans le programme de terminal, on saisit simplement la valeur chiffrée de la fréquence à produire (suivie de CRLF) et on l'envoie à l'ATtiny. La fréquence maximale à laquelle le signal est encore utilisable se situe autour de 200 kHz. La résolution du réglage de fréquence est théoriquement de :

$$f_{\text{DDSCLK}} / 2^n = 2 \text{ MHz} / 2^{32} = 0,00046... \text{ Hz}$$

Pour se servir de cette possibilité de réglage fin, on inscrit la fréquence sous forme de nombre décimal, avec jusqu'à trois chiffres après la virgule, sauf qu'ici, il faut mettre un point décimal et l'on saisit p. ex. : 1000.045 CRLF.

Calculer les paramètres pour le DDS en fonction de la fréquence voulue demande une grande précision. C'est pourquoi j'ai programmé moi-même les routines arithmétiques, entre autres pour effectuer des divisions avec des nombres à partie décimale.

La **figure 7** représente le spectre entre 0 Hz et 2 MHz d'un signal sinusoïdal produit pour $f = 125,123 \text{ kHz}$. On y aperçoit la présence d'harmoniques, mais à 30 dB sous le signal utile. Il y a en plus un tapis de bruit, inhérent au principe même du DDS.

Si nous étions partis d'un oscillateur à quartz ordinaire, notre générateur serait entaché d'une imprécision de $\pm 100 \text{ ppm}$. Pas question dans ces conditions de produire une fréquence de 100,00005 kHz.

Publicité

www.elektor.fr/kitdebut
 après le succès phénoménal du livre
 de Remy Mallard voici enfin votre
kit d'initiation
 avec plaque d'expérimentation et pince coupante

avec vidéo didactique pour l'assemblage

livre + kit :

le cadeau idéal

pour un(e) débutant(e)

Listing 3

```

uint32_t DDS24 ; // DDS phase, 24 bits used
volatile uint32_t dDDS24 ; // delta for DDS
uint16_t TOP1 ; // integer part of divider for PWM

ISR(TIMER1_OVF_vect) {
    PORTD |= _BV(4) ;
    DDS24 += dDDS24 ; // advance DDS phase
    if (DDS24 & 0x1000000UL) { // check bit 24 for overflow
        ICR1 = TOP1 ; // on overflow PWM width = TOP1+1
    }
    else {
        ICR1 = TOP1-1 ; // else PWM width = TOP1
    } ;
    DDS24 &= 0xffffffffUL ; // make DDS24 24 bits again
    PORTD &= ~ _BV(4) ;
}

```

C'est pourquoi notre générateur dispose aussi d'une entrée extérieure pour une horloge à 20 MHz. Dans un prochain article, je décrirai une horloge qui, à partir d'un émetteur étalon, délivrera une référence ultra précise de 20 MHz, pour produire des fréquences (sinus) d'une grande précision.

Accord de circuits résonants

Dans la suite de cette série d'articles, nous allons, toujours à l'aide d'un microcontrôleur AVR, recevoir et exploiter des transmissions de données dans la gamme des grandes ondes : DCF à 77,5 kHz, France Inter à 162 kHz, BBC à 198 kHz, etc. La plupart du temps, on utilise une antenne ferrite. Pour l'accorder, on peut fort bien mettre à profit notre générateur de sinus. On lui fait alimenter le circuit de la **figure 8** et puis on règle le condensateur variable pour atteindre le maximum.

La différence de phase entre tension de sortie U_{out} et tension d'entrée U_{in} peut nous servir à déterminer si la fréquence de résonance du circuit oscillant est plus haute ou plus basse que la fréquence du signal sinusoïdal qui lui est appliqué. Si la phase de U_{out} est en avance sur U_{in} , c'est que la fréquence du signal est plus basse que celle de résonance du circuit oscillant. Inversement, si U_{out} est en retard sur U_{in} , la fréquence du signal est plus haute que celle de résonance du circuit oscillant. En cas de résonance, U_{in} et U_{out} sont en phase.

Dans le circuit de l'exemple, les valeurs des composants sont données pour une résonance à 125 kHz. Comme bobine L1, c'est une self à noyau en pot qui est employée. Nous utiliserons ce circuit plus tard pour produire des signaux de test à 125 kHz. Le condensateur variable sert à régler la résonance du circuit oscillant sur 125 kHz. Comme signal d'entrée, on peut aussi bien recourir à l'onde de notre convertisseur N/A via K3 qu'au signal rectangulaire de la sortie MLI K4.

Onde carrée MLI avec diviseur fractionnaire

Venons-en à une autre application du principe DDS. Avec un temporisateur à sortie MLI qui produit un signal rectangulaire, on ne peut en principe obtenir qu'une fréquence qui résulte de la division de la première par un nombre entier. Si N est le diviseur et f_{CLK} la fréquence d'horloge, la nouvelle fréquence engendrée sera $f = f_{CLK}/N$. Mais si nous faisons varier alternativement le diviseur, par exemple entre N et N+1, nous produirons aussi des fréquences intermédiaires. Supposons un diviseur qui alterne N et N+1, on divisera finalement en moyenne par N+0,5. Et si nous voulons diviser par

10,333333... nous choisirons le diviseur 11 avec un rapport de « probabilité » $p=0,33333...$, et dans les autres cas $N=10$.

Mais comment réaliser cela en pratique ? Il nous faudrait une procédure qui dise quand il faut diviser par N et quand par N+1.

C'est ici que le générateur DDS à m bits nous vient en aide. Avec un m fort grand, nous pourrions effectuer des opérations très précises. Nous devons pour cela accéder au report (ou dépassement) de l'accumulateur de phase. Le rapport p sous lequel l'accumulateur de phase d'un DDS à m bits génère un report est exactement de

$$p = \text{DDSd} / 2^m$$

et il est possible de le régler avec grande précision par la variable DDSd. Nous pouvons ainsi utiliser le report pour inciter le temporisateur à diviser soit par N, soit par N+1.

Avec p.ex. une horloge à 20 MHz pour produire un signal à la fréquence de 77,5 kHz, il faudra diviser par

$$20\,000 / 77,5 = 258,0645161...$$

Cela signifie qu'il faut basculer entre N=258 et N=259 et cela dans le rapport $p = 0,0645161...$. Avec un DDS à 24 bits, on obtient $\text{DDSd} = p \times 2^{24} = 1\,082\,401$. Le **listage 3** est une routine de service d'interruption du temporisateur qui illustre cette idée.

Le signal rectangulaire résultant saute de part et d'autre de l'oscillation idéale, mais la moyenne correspond exactement à ce que nous voulions.

Le temps d'exécution de la routine complète, latence comprise, est d'environ 6 μ s, nous pouvons donc produire de cette manière des fréquences jusqu'à 160 kHz. Une routine écrite en assembleur nous permettrait d'aller encore plus haut. Comme le générateur d'ondes rectangulaires est d'un usage simple, il nous sera aisé de le commander par un terminal. Le code source [3] s'appelle *EXP-SquareGenerator-DDS-T1INT-V01.c*.

Le principe du diviseur fractionnaire connaît encore bien d'autres applications. Il permet entre autres de dériver de l'horloge d'un processeur un taux d'échantillonnage quelconque et même de le changer ensuite, si nécessaire, à l'aide d'une boucle à phase asservie (PLL).

Le générateur FM

En soi le générateur d'ondes rectangulaires n'est peut-être pas tellement intéressant, mais comme le processeur n'est pas encore saturé par la gestion de la MLI, il lui reste du temps pour faire varier dynamiquement la fréquence. Et voilà notre générateur à modulation de fréquence !

Le service météorologique fédéral allemand [4] émet sur la fréquence de 147,3 kHz en modulation par déplacement de fréquence (*FSK Frequency-Shift Keying*) des bulletins météo, mais à destination de radiotélétypes (*RTTY*, téléscripteur radio). Nous construirons plus tard un récepteur pour ces services radio. Pour essayer et accorder ce récepteur, un signal de test est bien utile. C'est facile à faire avec le diviseur fractionnaire et la MLI. En se guidant sur le débit binaire des données, on effectue des sauts de fréquence.

Programmons d'abord notre émetteur de test pour une fréquence porteuse de $f = 125$ kHz. Pour fabriquer un sinus avec un signal rectangulaire, utilisons le circuit de la figure 8. Nous connaissons déjà la routine d'interruption pour le générateur d'ondes carrées fractionnaires. La routine supplémentaire *SendBit* (listage 4) nous servira à émettre chacun des bits.

D'abord, le temporisateur 0 attend que le nombre de reports atteigne la valeur de *COUNT2*, ce qui revient à diviser la cadence des bits de reports du temporisateur 0 par *COUNT2*. Ensuite, selon les bits à émettre, on règle les valeurs de *deltaDDS24* et *TOP1* (modulation de fréquence). Pour masquer impérativement les interruptions pendant le réglage de ces paramètres, ne pas oublier les instructions *cli()* puis *sei()*. Sans cette précaution, une interruption pourrait intervenir alors qu'un seul des paramètres a changé, ce qui se traduirait par une erreur. On retrouve ces mêmes routines dans le programme *EXP-SQTX-FM-RTTY-V01.c*. Avec l'aide d'autres routines annexes, les nombres sont émis en code pour radiotélétype (Baudot [5]) pour simuler les émetteurs des services météo.

Vous verrez à la figure 9 le spectre de ces signaux FM RTTY : observez les deux pointes, à seulement 50 Hz de part et d'autre de la fréquence de 125 kHz. Il s'agit d'un spectre continu qui s'affaiblit rapidement jusqu'à ± 1 kHz.

Une fois qu'on s'est construit un tel générateur, l'envie est grande de l'essayer pour en vérifier la modulation. L'article du mois prochain sera donc consacré à la construction d'un récepteur numérique.

(100180 – trad. Robert Grignard)

Liens

- [1] www.elektor.fr/110553
- [2] www.elektor.fr/080083
- [3] www.elektor.fr/100180
- [4] <http://wapedia.mobi/de/DDH47>
- [5] http://fr.wikipedia.org/wiki/Code_Baudot
<http://fr.wikipedia.org/wiki/Radiotélétype>

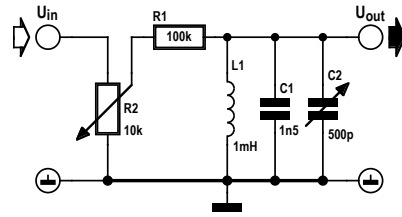


Figure 8. L'accord d'un circuit oscillant.

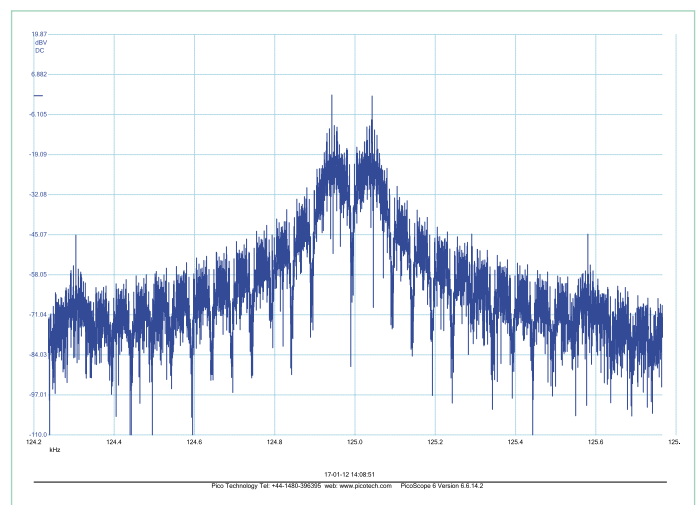


Figure 9. Le spectre de la modulation à ± 50 Hz de la fréquence de 125 kHz.

Listing 4

```
void SendBit(uint8_t theBit) {
    uint8_t k ;
    for (k=0 ; k<COUNT2 ; k++){
        while( ( TIFR & (1 << TOV0) )== 0 ) { }
        TIFR |= (1 << TOV0) ;
    }
    if ( theBit==MARK ) {
        cli() ;
        deltaDDS24=MARK_deltaDDS24 ;
        TOP1=MARK_TOP1 ;
        sei() ;
    }
    else{
        cli() ;
        deltaDDS24=SPACE_deltaDDS24 ;
        TOP1=SPACE_TOP1 ;
        sei() ;
    }
}
```


Protégeons l’USB

Raymond Vermeulen (labo Elektor)

L’interface USB est robuste, mais, sur le plan électrique, il y a des précautions à prendre, notamment en matière de **limitation du courant** sur la ligne d’alimentation +5 V. Pour cela, on a souvent recours à un fusible. Qu’ils soient ou pas d’un type qui peut être réarmé, de tels fusibles manquent de précision. Entre le débit qui peut les traverser sans que rien se passe et le courant minimum pour lequel la rupture est garantie, on constate un écart souvent redoutablement grand. Il peut aller du simple au double !

En outre, il faut sécuriser les lignes de données contre les **décharges électrostatiques** (DES, *ESD* en anglais). Généralement, cette menace est prise en compte sur les circuits USB, mais le dispositif est encombrant et introduit sur la ligne une charge capacitive qui risque de déformer le signal. Pour vous éviter ces deux ennuis, nous proposons ici deux circuits intégrés appropriés.

(120095)

TPD2EUSB30ADRTR

Cette puce est parfaite pour l’écrtage sur les lignes de données. Hormis son nom impossible à retenir par cœur, elle n’a rien de particulier et même son prix est modéré. Si, il y a sa taille, bien sûr : à peine 1 x 1 mm ! Même sur une petite platine, elle s’installera sans difficulté entre deux lignes de données. Le fabri-

cant affirme qu’elle peut s’utiliser sur des lignes différentielles à haute fréquence comme des USB3.0, SATA, PCIe, etc. D’autres puces en font sans doute autant, mais elles déforment davantage le signal. Intéres-

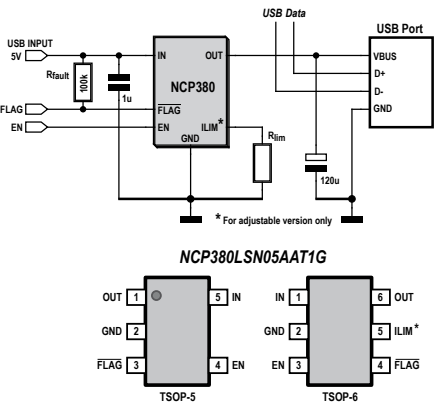
sant aussi : cet intégré se passe de V_{bus} . C’est pratique, puisque sur certaines applications, il n’y a pas de V_{bus} disponible. Avec une tension de claquage aussi basse que 4,5 V, les appareils sous 3,3 V sont étroitement protégés. En revanche, cette puce n’est pas utilisable sur des lignes qui acheminent des tensions plus élevées que 3,3 V ; celles-ci se font heureusement de plus en plus rares, surtout à haute vitesse. Il est étrange que le fabricant inscrive d+ et d- sur le composant, alors que, pour autant qu’on sache, le fonctionnement est identique dans un sens comme dans l’autre, comme le confirme d’ailleurs la fiche de caractéristiques. Peut-être une manière de repérer physiquement les lignes. En tout cas, vos lignes seront protégées contre la DES jusqu’à plus et moins 8 kV et le signal n’en subira qu’une influence minimale, de quoi éviter pas mal de soucis.

PARAMÈTRE	CONDITION	VALEUR
Tension inverse limite	D-, D+	3,6 V
Tension de claquage	$I_{io} = 1\text{ mA}$	4,5 V
Protection DES	D-, D+	8 kV
C_{io-io}	$V_{io} = 2,5\text{ V}$	0,05 pF
C_{io-gnd}	$V_{io} = 2,5\text{ V}$	0,7 pF

Lien : www.ti.com/lit/ds/symlink/tpd2eusb30a.pdf

NCP380LSN05AAT1G

Pour la limitation de courant, j’ai trouvé cette puce dont le prix est de moitié inférieur à celui de produits comparables ! Le dessin reprend le circuit pour vos hôtes USB. Il m’a semblé intéressant d’en faire l’essai du côté secondaire, pour le cas où, selon la loi de *Murphy*, il lui arriverait de consommer trop de courant dans le projet concerné. Si un circuit dispose de la double possibilité de s’alimenter par l’USB et sur une source externe, la puce se déconnecte dès que la tension de sortie dépasse de 100 mV celle d’entrée. Pour l’USB, il est pratique d’utiliser la variante dont la limite de 500 mA est intégrée. Autre avantage : cette puce offre également une protection contre la DES. Et ce n’est pas tout, elle démarre et s’arrête en douceur, ce qui est spécialement utile au moment de la déconnexion. La résistance interne en conduction n’est que de 70 mΩ, alors que sur les fusibles, elle peut se situer dix fois plus haut. En condition de surchauffe, la puce est automatiquement débranchée, même si la tension d’entrée est trop basse. Toutes ces situations d’erreur sont signalées par une broche *drapeau*. Ce genre de puces peut se révéler très utile.

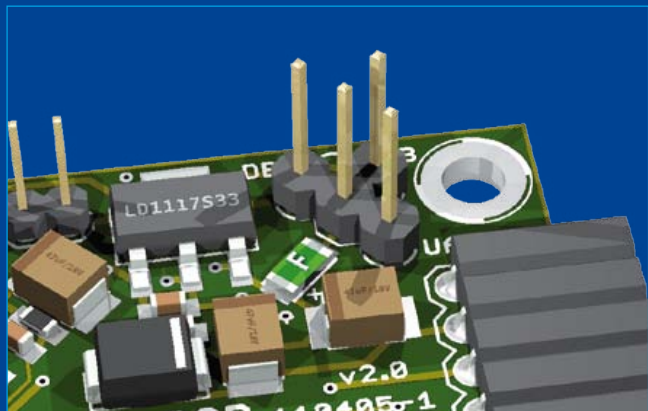


PARAMÈTRE	CONDITION	VALEUR
V_{in}	tension de service	2,5 V...5,5 V
$V_{rev} (V_{out} - V_{in})$	point de déclenchement	100 mV
$R_{DS(on)}$	$V_{in} = 5\text{ V}$	70 mΩ
Limite de courant	$V_{in} = 5\text{ V}$	[min. 0,5 A] [typ. 0,58 A] [max. 0,65 A]

Lien : www.onsemi.com/pub_link/Collateral/NCP380-D.PDF

Le cavalier à trois pattes de l'AndroPOD

À première vue, il en a bien quatre, mais voyons voir...



Position DEBUG

Antoine Authier (Elektor Lab)

L'AndroPOD est notre interface de commande d'un téléphone Android pour communiquer avec d'autres circuits. Une de ses caractéristiques remarquables est la variété des sources d'alimentation possibles : USB device, UART, module de débogage, module RS-485. Une telle flexibilité implique un défi pour le concepteur : comment éviter tout conflit d'alimentation sans recourir à de coûteux composants mécaniques ? D'ailleurs après un long butinage des catalogues de distributeurs et après avoir discuté avec mes amis chinois, force a été de constater qu'il n'est pas facile, voire impossible, de trouver un mignon petit interrupteur à glissière à quatre positions qui résisterait à disons 500 mA, voire 1 A.

Ces questions m'ont incité à réfléchir encore plus que d'habitude lors de la conception du circuit imprimé, afin que le projet reste pratique, facile à utiliser et accessible à tous. La multiplication des cavaliers, composants certes robustes, serait une

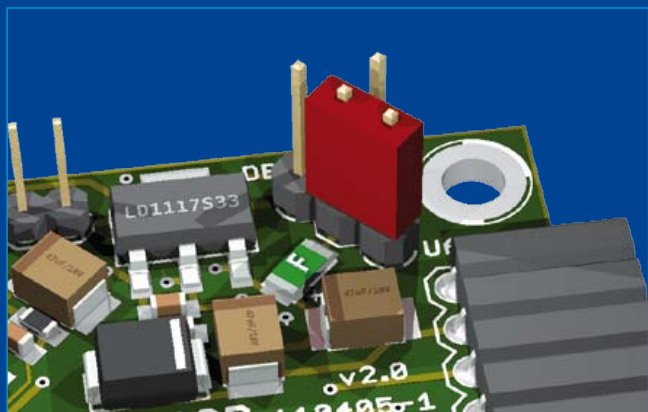
solution... cavalière et surtout dangereuse, car, avec le grand nombre de configurations possibles, une mauvaise manipulation par manque de concentration mènerait tôt ou tard au pire... La solution est venue quand, apitoyé par le labyrinthe de cavaliers dans lequel il me voyait perdu, mon collègue Ton m'a suggéré ingénument : « c'est pas un peu exagéré, quand même ? ». Sans attendre ma réponse, il ajoutait : « et pourquoi pas les mettre en étoile... ? »

Eureka. Cool. Génial. Élémentaire, mon cher Ton ! En effet, cette configuration, a priori un peu exotique, interdit les connexions multiples et n'autorise jamais la sélection que d'une seule source à la fois pour alimenter le montage.

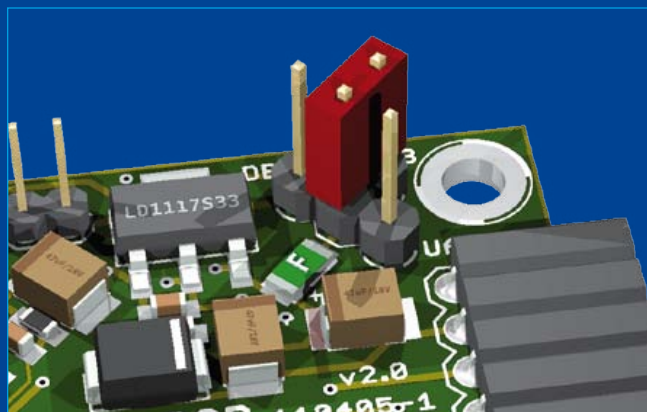
Amusez-vous bien avec l'AndroPOD.

Antoine (merci Ton !)

(120076)



Position USB



Position UART

Alim. découp. ch. filt. DC*

Raymond Vermeulen (Labo Elektor)

Ah les convertisseurs DC/DC, souvent utiles, parfois indispensables. Et quel rendement la plupart du temps ! Comme je suis les traces des vrais pros, j'en utilise de plus en plus.

L'autre jour, je travaillais sur un circuit alimenté par piles pour lequel il m'en fallait un très efficace. J'étais le nez dans le guidon, en train de calculer le condensateur réservoir, quand je me suis souvenu de mes cours sur les filtres passifs et les alimentations à découpage.

Contrairement à celle des régulateurs linéaires de la vénérable série LM78xx, la tension de sortie des alimentations à découpage est affectée d'une forte ondulation résiduelle qui, en aval, perturbe davantage les circuits analogiques que les circuits numériques ; il faut donc adapter le filtrage de cette tension aux besoins de l'application. Les feuilles de caractéristiques des régulateurs à découpage suggèrent souvent une valeur modeste pour le condensateur chargé de filtrer la sortie, par exemple 1 μF , mais en pratique on a tendance à y mettre plutôt 10 μF .

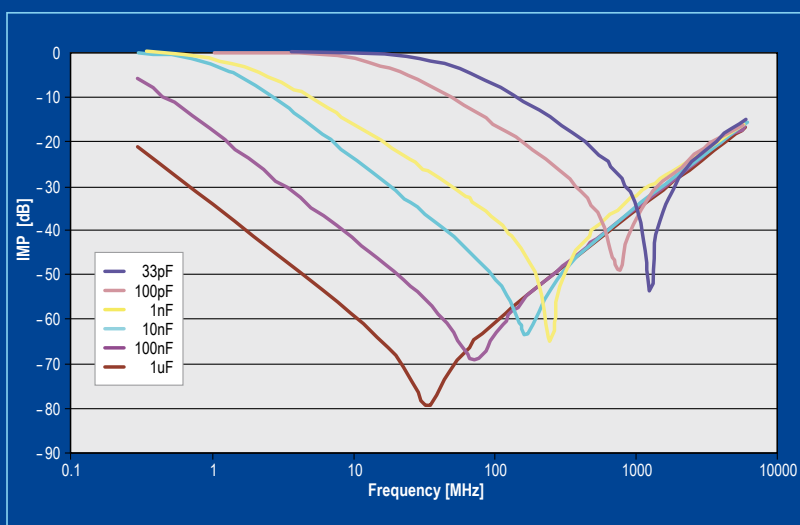
Or, comme chacun le sait, plus la capacité du condensateur sera élevée, plus tôt son comportement sera celui d'une inductance à mesure que la fréquence appliquée augmentera (voir **graphique**). De plus, les alimentations à découpage, du fait de leur principe de fonctionnement, engendrent des ondulations sur

une large plage de fréquences. Ce qui n'arrange rien quand on sait que les perturbations causées par les hautes fréquences du bruit de l'alimentation sont plus fortes que celles que causent les composantes à basse fréquence.

Pour élargir le spectre filtré, on monte en parallèle plusieurs condensateurs de valeurs différentes. Dans notre cas, l'unique condensateur de 10 μF gagnerait à se voir adjoindre en parallèle un modèle de 100 nF, doté de préférence de bonnes caractéristiques en HF, p.

ex. un condensateur céramique X5R à faible résistance série : ainsi, l'atténuation des ondulations porte sur l'ensemble du spectre susceptible de perturber une application.

Si cela ne suffit pas pour la vôtre, ajoutez une petite inductance d'environ 0,1 μH en série avec la sortie, entre les condensateurs de filtrage de 10 μF et 100 nF, créant ainsi un filtre LC. Un filtre RC additionnel est recommandé aussi



près que possible de chacune des broches d'alimentation sur chaque circuit intégré analogique. Chères en composants et en surface de circuit imprimé, ces mesures vous feront économiser de l'aspirine.

* Alimentation à découpage cherche filtre pour le continu

(120144 - trad. Kévin PETIT)

Histoire(s) de prises

Thijs Beckers (Elektor)

Pour alimenter sa rubrique Rétronique, mon collègue Jan Buiting se jette sur toutes sortes de bidules décatis qu'il chine le dimanche matin aux puces, avec un penchant pour les vieilles radios Philips. Récemment, Jan m'a exhibé sa collection de prises anciennes. Trop jeune pour avoir été témoin de leur utilisation,

je reste curieux de leur fonction d'antan. Quelle variété ! Elles méritent un coup de projecteur.

Passons en revue cette robuste connectique d'un autre âge. Y reconnaissez-vous quelque chose de familier ?

La plupart de ces prises ou embases sont polarisées, et munies d'un détrompeur mécanique contre les mauvais branchements.

1. Prise pour haut-parleurs de radio des années 50 avec contact central empêchant le branchement dans une prise secteur 220 V. Voir aussi 4 ci-dessous.

2. Prise solide à trois contacts avec verrouillage. Dimensionnée pour les forts courants ou les tensions élevées. Utilisée pour des haut-parleurs, éventuellement aussi sous une tension de 100 V.

3. Adaptateur *Philips* DIN vers IEC à 5 voies, pour tourne-disque, doté de cinq contacts plaqués argent (nettoyés avec une brosse en fibre de verre). Objet rare.

4. Prise d'enceintes satellites (aigus) pour radios *Philips* de haut de gamme, type B6X62A. Non polarisée ; le point rouge identifie la phase du haut-parleur. La version marron (1) servait pour l'enceinte de graves. La radio possède un filtrage actif à deux voies, séparées à 400 Hz !

5. Version câble du connecteur audio IEC à 5 voies *Philips*, avec son modèle femelle. *Philips* s'obstinait à utiliser ce connecteur qui n'a jamais convaincu ni les clients ni l'industrie. Impossible de l'ouvrir sans outil spécifique.

6. Prise mâle à 3 contacts pour microphones ou haut-parleurs 800 ohms *Philips* avec sa complémentaire femelle. Prévue pour relier des haut-parleurs aux amplificateurs à tubes sans transformateur de sortie (OTL) (vers 1955-1964). Polarisée. Le troisième contact, optionnel, servait au passage des aigus (>400 Hz). Remarquez le câble blindé pour microphone dynamique sur le connecteur femelle.

7. Vieux connecteur d'alimentation secteur sans terre, style IEC. Polarisé.

8. Prise avec contact central pour antenne VHF sur câble plat 240 Ω .

9. Comme 8, mais avec des contacts arrondis et sans contact central.

10. Prise pour antenne TV VHF/UHF sur câble plat 240 Ω . Remarquez les contacts asymétriques. Utilisation exacte inconnue.

11. Comme 8. Polarisé (comment ?). Sûrement pour antenne ; utilisation exacte inconnue.



Ceci n'est encore qu'un petit échantillon des innombrables prises, embases et connecteurs utilisés pendant des décennies, avant l'Ethernet et le WiFi. Et visiblement, ce n'est pas fini. De nouveaux connecteurs ne cessent de faire leur apparition. Les plus courants actuellement sont : USB, USB 3, Firewire à 4 et 6 broches, Thunderbolt, HDMI, DisplayPort, DVI (-A, -D, -I, -D HDCP, mini-, micro-), VGA, S-vidéo,

RCA, TOSLINK (pour S/PDIF), XLR, TRS (jack 2,5 mm, 3,5 mm ou 6,35 mm), banane, SATA, RJ-45 etc. La liste est longue, sans même parler de la kyrielle de connecteurs d'alimentation.

Où s'arrêtera cette avalanche ? Bien malin qui le dira. Dans le capharnaüm de Jan, je pensais retrouver rapidement la trace du premier connecteur. Illusion ! C'est pourquoi j'ai imaginé de vous mettre au défi de retrouver la *Iunctio Antiquissima*, l'Adam et Ève des connecteurs. Envoyez-nous une photo claire du plus vieux couple de connecteurs que vous pourrez trouver, en l'adressant à connectorcontest@elektor.com. Et ne vous étonnez pas de le retrouver dans un prochain article de la rubrique Rétronique.

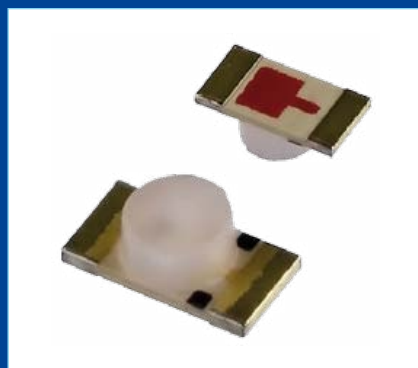
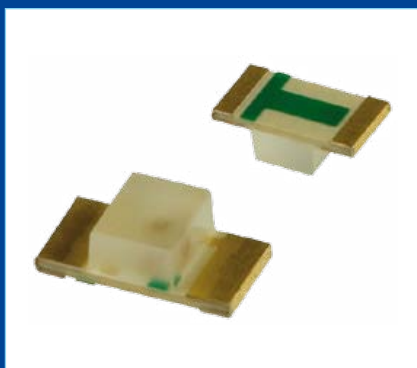
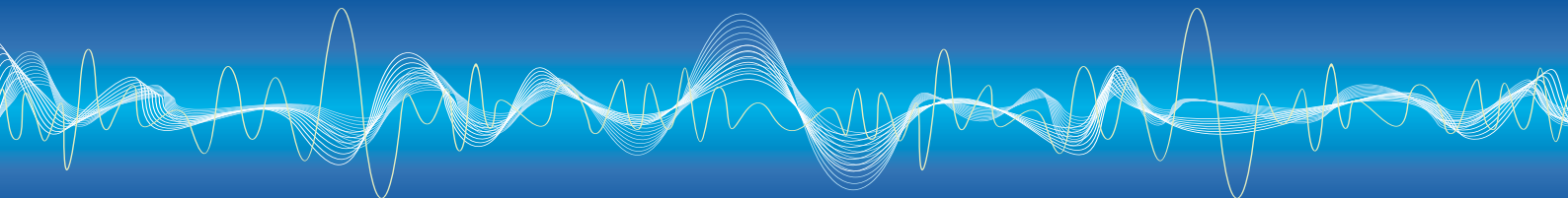
(120027 - trad. Kévin PETIT)

Polarité des LED CMS

Raymond Vermeulen (Labo Elektor)

Il faut redoubler d'attention pour souder les composants quand ils sont polarisés. Des fois, surtout avec les plus petits des CMS,

il n'est pas évident de repérer la bonne patte. C'est en tout cas ce que j'ai remarqué en voulant souder deux LED CMS en boîtier 0603 : impossible de distinguer l'anode de la cathode sur ces fichus boîtiers microscopiques...



J'ai donc décidé de les souder au pif, en espérant secrètement que la chance serait de mon côté. Une chance sur deux. Conformément à la loi de Murphy, évidemment, mes deux LED étaient dans le mauvais sens. Je les dessoude puis les ressoude dans le bon sens, vaguement contrarié de ne pas maîtriser tous les détails de mon circuit. L'affaire aurait pu en rester là si je n'avais pas fini par remarquer, dans la fiche technique de ces LED, qu'un marquage est bel et bien prévu (je m'en doutais), même sur des composants en boîtier 0603.

Un symbole quelconque au dos de la LED en repère l'anode ou la cathode. Personne ne se préoccupe d'un standard et chaque fabricant y va de son petit dessin. Voyez la variété de repères

que j'ai pu rencontrer sur les spécimens trouvés au labo : bandes, points, rectangles et flèches en tous genres, parfois combinés.

Ma conclusion :

n'hésitez pas à jeter un coup d'œil attentif à la fiche technique de vos composants ou utilisez un petit montage de test. C'est l'évidence même pour les pros, mais le conseil ne sera peut-être pas inutile pour les autres. Et il les reconciliera éventuellement avec ces diables de petits composants.

(120145 - trad. Kévin PETIT)

Un câble pour le bus

Jens Nickel

Notre bus, désormais connu sous le nom d'ElektorBus, continue de se développer ; jusqu'ici nous nous sommes contentés d'expérimenter sur table, avec des petits circuits, mais le mois prochain, notre bus passera par la station « 230 V ». Puis nous mettrons en place un bus pilote, grandeur nature, au château d'Elektor (voir le logo de l'ElektorBus). Compte tenu de la taille de la bâtisse, ça nous fera quelques dizaines de mètres. Il était donc grand temps de nous préoccuper de ce qui se passera avec un câble d'une certaine longueur.

Pour mener à bien ce premier essai, nous avons interconnecté deux nœuds expérimentaux [1] et notre interface AndroPOD [2], reliés à un *smartphone* dans le rôle du troisième larron. À l'un des deux nœuds, nous avons connecté une simple photo-résistance dont les valeurs, acheminées par le câble, étaient affichées sur l'écran du téléphone Android.

En fouillant dans les stocks du labo, nous avons mis la main sur un premier rouleau de câble de type *KROSCHU Schaltflex CY Style 2571* [3] à dix conducteurs, dont une paire torsadée. Nous avons procédé à deux essais : d'abord avec un seul conducteur pour chacune des quatre lignes de notre bus : données A et B, 12 V, GND ; puis avec la paire torsadée pour A et B, comme le recommande d'ailleurs le standard RS-485. Le rouleau faisait 36 m :

sur cette longueur déjà respectable, nous n'avons remarqué aucune perturbation, ni dans un cas ni dans l'autre. Aussi bien les signaux que la tension d'alimentation se sont révélés fiables. Suffisamment en tout cas pour commander avec le deuxième nœud un relais alimenté lui aussi par le bus. Mon collègue Raymond Vermeulen, qui pourtant ne s'en laisse pas conter, a trouvé mes signaux parfaits ! Une seconde série d'essais a été faite ensuite avec du câble CAT5E, composé de quatre paires torsadées et blindées. Conformément au standard, nous avons utilisé l'une des deux paires pour les données A et B et court-circuité les deux autres pour les lignes 12 V et GND. Là non plus, aucun problème notable avec la trentaine de mètres disponibles. Notre séance a été filmée [4]. Le bus Elektor repasse le mois prochain. Soyez au rendez-vous.

(120198)

[1] www.elektor.fr/110258

[2] www.elektor.fr/110405

[3] www.kroschu-cable.de/documents/downloads/Schaltflex%202571-07-08%20%5Be%5D.pdf

[4] www.youtube.com/watch?v=rbDSTXNARmw



Adaptateurs DIP pour programmeurs in situ

Alain Litt (France)

Pour programmer un microcontrôleur *in situ*, c'est-à-dire sans le sortir de son circuit, il faut un connecteur spécial sur la carte, qui établira la connexion entre le programmeur *in situ* et le micro. Or, les contrôleurs sont généralement à l'étroit, et l'encombrement d'un tel connecteur l'empêche d'accéder aux broches du composant. La programmation *in situ* devient alors difficile, voire impossible.

Les adaptateurs proposés ici permettent de programmer *in situ*, à l'aide des programmeurs PICKit 2 ou 3 de Microchip, les microcontrôleurs en boîtier DIP sur une carte sans connecteurs de programmation spécifiques, mais son principe est facile à adapter à d'autres familles de micros d'autres constructeurs.



L'astuce consiste à utiliser deux supports tulipes embrochés l'un sur l'autre et à souder les connexions utiles pour la programmation sur le deuxième support, puis à rajouter un connecteur à six broches, repéré pour la connexion au PICKit. Il suffit ensuite d'assembler solidairement les deux supports, et de bloquer les fils de connexion à l'aide d'un peu de colle époxy. Il faudra bien sûr fabriquer un adaptateur



spécifique pour les principaux types de boîtiers existants. Les cartes comportent en général leur alimentation propre, ce qui limite à quatre le nombre de broches à connecter : /MCLR (1), GND (3), PGD (4) et PGC (5). Il faudra évidemment repérer la broche 1 du connecteur à six broches vers le PICKit, soit par un repère de couleur sur le câble, soit par une autre marque.

(120034)

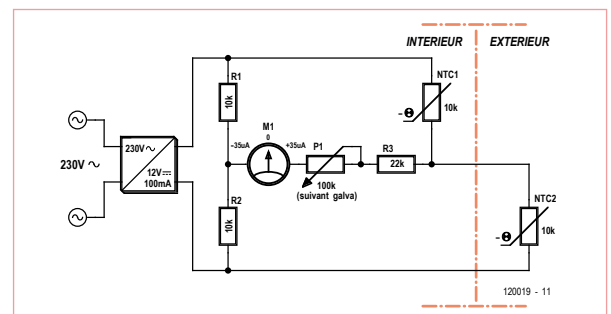
Un thermomètre relatif

Loïc Marty F5UBZ (France)

Voilà une réalisation qui ne requiert que très peu de composants et qui est utile à la maison. Il s'agit d'un thermomètre différentiel qui indique à l'aide d'un galvanomètre à zéro central la différence de température entre l'intérieur d'une habitation et l'extérieur. À quoi ça sert ? Les extravagances météorologiques deviennent de plus en plus fréquentes. Il n'est pas rare qu'en automne et même en hiver, certains jours, la température extérieure dépasse la température à l'intérieur de l'habitation. On peut alors couper le chauffage et ouvrir grand les fenêtres pour faire entrer un peu de chaleur dans la maison. En été, c'est souvent l'inverse, et on peut faire rentrer de l'air frais. Il est amusant de constater qu'au point d'équilibre des températures, il fait aussi chaud dedans que dehors. C'est notamment ce point que ce thermomètre affiche. Il faut donc considérer ce montage comme un indicateur et non pas comme un appareil de mesure étalonné. Néanmoins, au regard des relevés comparatifs avec deux vrais thermomètres, l'indicateur



(gradué $\pm 35 \mu\text{A}$, transformé en ± 35 degrés) est quand même proche de la réalité malgré la non-linéarité des CTN. Le schéma est simple, il s'agit du principe du pont de Wheatstone. À l'équilibre des températures, le potentiel au point milieu entre les deux CTN est égal au potentiel entre les deux résistances de $10\text{k}\Omega$: l'aiguille est en position centrale. Si par exemple la température extérieure est inférieure à celle de dedans, la valeur de la CTN externe sera supérieure à celle de dedans et le potentiel au point milieu des deux CTN est inférieur au potentiel du point milieu des deux résistances : l'aiguille dévie vers la gauche. L'utilisation d'un galvanomètre de quelques



microampères ne perturbe pas trop la mesure. On peut étalonner ce thermomètre par exemple avec un congélateur et la température ambiante pour obtenir des indications tout à fait satisfaisantes et représentatives. Un adaptateur secteur à trois sous permet d'alimenter tout ce petit monde. On trouve des indicateurs à zéro central par exemple chez Selectronic. Bon amusement. Si vous trouvez d'autres applications intéressantes et originales pour un tel indicateur, veuillez nous en informer.

(120019)

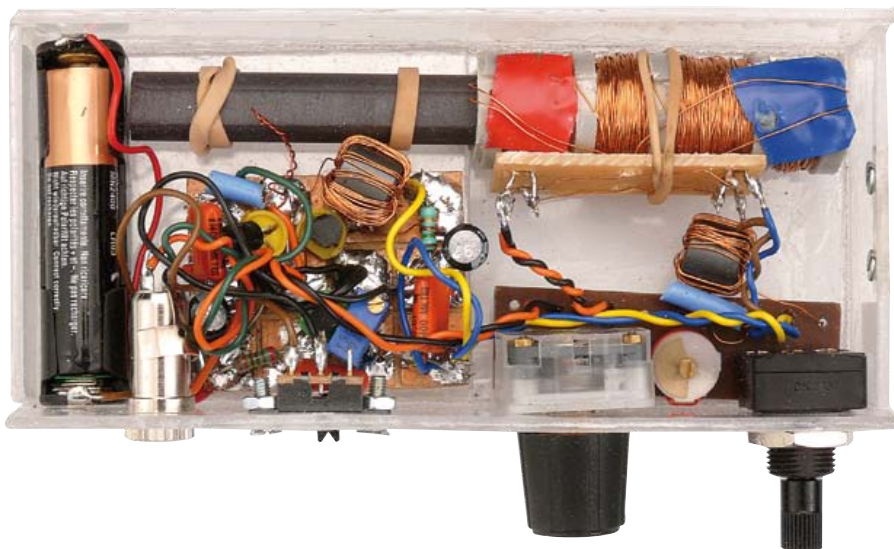
Lien :

Site de l'auteur : www.f5ubz.fr

récepteur Petites Ondes à faible consommation

Dans son numéro de janvier 2012, Elektor vous invitait à goûter aux satisfactions élémentaires de la pratique de l'électronique en renouant avec la réception et la démodulation des ondes radio. Après les Grandes Ondes, voici les Petites Ondes.

Jean-Pierre Redouté



Variante du récepteur Grandes Ondes

Le récepteur décrit page 14 du numéro 403 de janvier 2012 est conçu pour l'écoute des Grandes Ondes seulement, car c'est sur cette gamme que l'on trouve le plus d'émetteurs francophones.

Cependant, dans de nombreux pays, il n'y a que peu, ou pas du tout d'émetteurs sur Grandes Ondes. L'essentiel de la radiodiffusion se trouve sur Petites Ondes. Il nous a donc paru opportun de tester et publier une modification très simple permettant d'obtenir un récepteur PO de mêmes performances.

Rappelons le schéma (figure 1) et le plan de gravure du circuit (figure 2). La photo ci-dessus est celle du prototype dans sa version GO. Pour la description de son fonctionnement et du mode de construction, nous vous renvoyons à l'article mentionné sous [1]. Notre sujet ici est le changement de gamme de réception.

Transformation en récepteur PO

On peut facilement modifier le montage du récepteur de Grandes Ondes de façon à recevoir les Petites Ondes. Il suffit de remplacer les enroulements du cadre en ferrite.

Sur le prototype, L1 est constitué de 65 spires jointives de fil de cuivre émaillé 34swg sur une portion de tube en CPCV 14x1,6. L'inductance obtenue est d'environ 330 μ H.

L2, placée à environ 5 mm, comporte 7 spires jointives de même fil.

Le bobinage a été photographié seul d'abord (figure 3), puis muni de son bar-

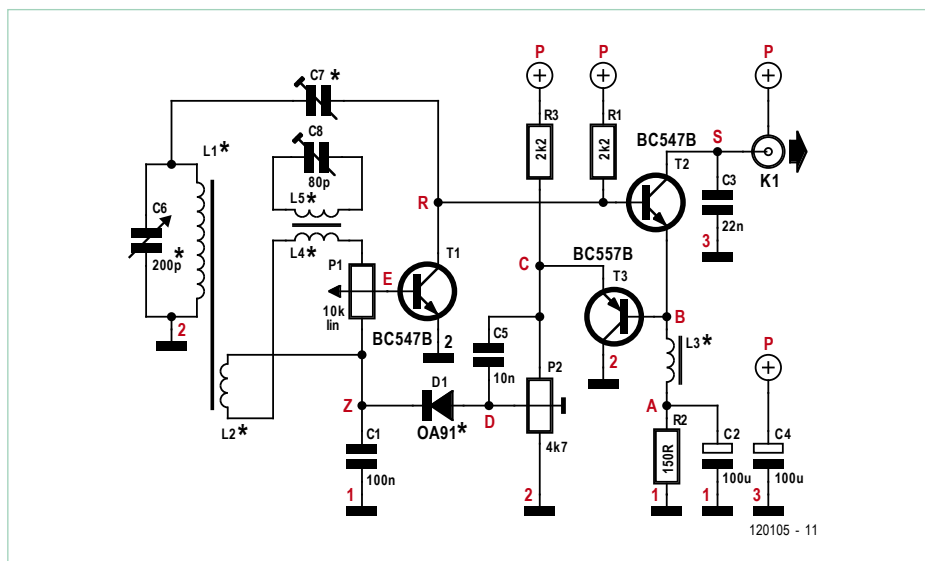
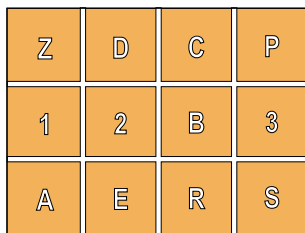


Figure 1. Schéma de principe. Les caractères en rouge correspondent aux carrés du plan de gravure de la figure 2.



120105 - 12

Figure 2. Le plan de gravure

Liste des composants

Résistances :

R1, R3 = 2,2 k Ω R2 = 150 Ω P1 = 10 k Ω potentiomètre linéaireP2 = 4,7 k Ω ajustable

Condensateurs :

C1, C5 = 10 nF

C2, C4 = 100 μ F électrolytique

C3 = 22 nF

C6 = 200 pF variable (voir texte)

C7 = « queue de cochon » avec éventuelle résistance en série (voir texte)

C8 = 80 pF ajustable

Inductances :

L1 à L5 = voir texte

Semi-conducteurs :

D1 = diode au germanium (OA91, AA112, etc.)

T1, T2 = BC547B

T3 = BC557B

Divers :

K1 = mini jack stéréo 3,5 mm

reau de ferrite (**figure 4**). Il se monte en lieu et place du bobinage GO dans le récepteur d'origine sans autre modification.

Toutefois l'écoute de signaux très faibles, provenant d'émetteurs éloignés, nécessite un réglage de réaction très proche de l'oscillation ; le réglage montre alors un manque de progressivité très gênant. Ce défaut a été corrigé par une résistance R4 placée en série avec C7.

Pour compenser, la valeur de C7 a été augmentée (sur le prototype, R4 = 100 k Ω , et C7 est une « queue de cochon » d'une dizaine de centimètres).

Pour l'écoute d'émetteurs proches, la taille du récepteur peut être réduite en utilisant un barreau plus court ; dans ce cas, il faut augmenter le nombre de spires (environ 80 spires pour un barreau raccourci à 5 cm).

Mise au point

Avant toute mise sous tension, vérifier le câblage, les polarités, l'absence de court-

circuit. Puis sans brancher d'écouteur, connecter la pile avec un milliampère-mètre en série : la consommation doit être nettement inférieure à 1 mA ; dans le cas contraire, rechercher le composant défectueux ou le court-circuit responsable de la surconsommation en mesurant sur chaque résistance la tension entre les extrémités (rappel : résistance en k Ω x courant en mA = tension en V).

Polarisation : la pile étant raccordée normalement, brancher le milliampèremètre à la place de l'écouteur ou en série avec lui et régler l'ajustable P2 pour obtenir 1 mA (courant collecteur de T2).

Vérifier que les tensions entre base et émetteur de chaque transistor sont voisines de 0,6 V et que la diode est polarisée dans le sens passant (D positif par rapport à Z). Si le réglage est impossible, vérifier la continuité des liaisons, l'absence de court-circuit,

les valeurs des résistances et le bon état des semi-conducteurs.

Réaction : si les connexions du cadre sont raccordées dans le bon ordre (L4/P1, D1, masse, C7) et si la capacité de C7 est suffisante, le circuit ne doit pas osciller quand P1 est au minimum (côté D1 C1) et osciller quand il est au maximum ; pour le vérifier, régler P1 au minimum ; on doit entendre un léger souffle dans l'écouteur ; augmenter P1 : arrivé à une certaine valeur on doit entendre un net *cloc* et le souffle disparaît presque. Notons que s'il est possible d'écouter une émission en réglant C6 et l'orientation du cadre, la vérification est encore plus facile : le niveau sonore augmente avec la rotation de P1 jusqu'au point de début d'oscillation où apparaissent un sifflement puis, si on insiste, le *cloc* et le silence. Pour cet essai, il est intéressant d'utiliser pour C7 une valeur trop élevée (10 à 15 cm devraient suffire).

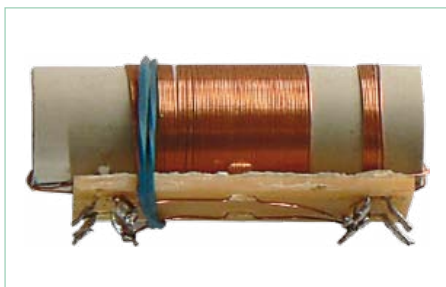


Figure 3. Le bobinage couplé L1 et L2 seul...

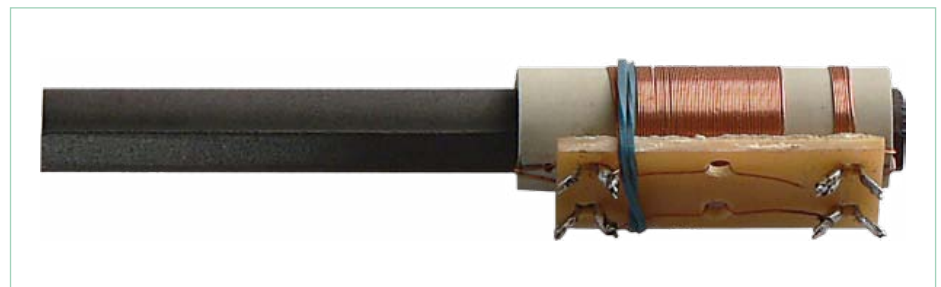


Figure 4. ... et avec son barreau de ferrite.

Calage en fréquence : en construisant soi-même son cadre, on n'a pas forcément la chance d'obtenir directement la couverture exacte de la gamme d'ondes recherchée : l'inductance dépend du type de ferrite, des dimensions et du nombre de spires et de leur agencement ; et pour un bobinage en vrac, la valeur de la capacité parasite n'est pas du tout négligeable, et est à peu près imprévisible.

Les petits condensateurs ajustables à vis situés à l'arrière du condensateur variable C6 donnent une petite possibilité de rattrapage ; une autre possibilité plus importante est offerte par le déplacement des bobines sur le barreau.

Comment caler son récepteur quand on ne possède ni fréquencemètre, ni oscilloscope, ni *grid-dip* ? On va se servir du phénomène décrit dans le paragraphe précédent. Quand notre récepteur entre en oscillation, il se comporte comme un émetteur de très faible puissance et son sifflement peut être capté par un autre récepteur s'il en est très proche. Cet autre récepteur étant réglé sur une fréquence connue, on tourne lentement la commande de C6 ; le passage sur la même fréquence provoquera dans le haut-parleur un *pi-ou-itt* caractéristique ; il suffit alors de répéter l'opération pour plusieurs valeurs de fréquence. Si on a France-Inter, mais pas RTL, il faut réduire le nombre de

spires ; dans le cas contraire, il faut l'augmenter ou ajouter un petit condensateur en parallèle sur C6.

Réglage du taux de réaction : la partie utile de la course de P1 est entre le minimum et le point où commence l'oscillation ; pour une plus grande souplesse de réglage, il est donc intéressant de repousser ce point vers le maximum de P1. Pour cela, il suffit de se mettre à l'écoute d'une émission faible, et de raccourcir peu à peu C7 jusqu'à ce qu'il soit possible de régler P1 au maximum ; veiller à ce que la pince coupante ne crée pas de bavures qui court-circuiteraient C7 !

Informations complémentaires

Pourquoi une bobine de couplage ?

Nous n'entrerons pas dans des détails théoriques, mais il faut retenir qu'un transistor bipolaire (contrairement à un FET) présente une impédance d'entrée faible quand il fonctionne en mode émetteur commun (c'est-à-dire émetteur au 0 V).

Quand le circuit L1/C6 est accordé sur une émission, il constitue pour elle une forte impédance : la tension est obtenue avec un très faible courant ; si on y raccorde une faible impédance, elle se comportera presque comme un court-circuit et la tension diminuera énormément. En revanche si on ajoute une bobine L2, couplée par induction à L1 et comportant un nombre de spires beaucoup plus petit, on dispose entre ses bornes d'un courant plus élevé (sous une tension plus faible) beaucoup mieux adapté à l'entrée du transistor.

Dans ce récepteur, L2 est également nécessaire pour une autre raison : On cherche à obtenir une réaction positive sur l'étage d'entrée, c'est-à-dire ajouter au signal d'entrée une petite fraction du signal amplifié, définie par C7 et R4. Si elle était directement appliquée à la base de T1, elle serait soustraite ; L2 sert à inverser le signal.

Capacité parasite

Dès que l'on met en présence deux conducteurs, on constitue un condensateur (même si sa capacité est très faible).

Effet gênant : Les spires adjacentes d'une bobine présentent une très faible capacité ; pour un bobinage à spires jointives, l'inconvénient est généralement minime ; mais s'il faut des centaines de spires, ce mode de bobinage est inconfortable et encombrant, ce qui conduit à superposer les spires, donc augmenter la capacité qui peut atteindre des dizaines voire des centaines de pF. Très ennuyeux quand le CV ne

vaut que 200 pF ! La parade ? Diviser la bobine en plusieurs galettes ; d'une part, la capacité de chaque galette est inférieure à celle de la bobine monobloc, d'autre part, les galettes étant câblées en série, la capacité globale est égale à la capacité unitaire divisée par le nombre de galettes.

Effet intéressant : Le condensateur en fils torsadés dit « queue de cochon ».

Exemple pratique : prendre environ 40 cm de fil émaillé de 0,45 mm de diamètre, bloquer les deux extrémités côte à côte dans un étau, passer un crayon dans la boucle, maintenir les fils tendus d'une main et tourner le crayon de l'autre, comme pour remonter l'hélice d'un avion à moteur caoutchouc, resserrer les torsions jusqu'à environ une par mm, terminer par un léger coup pour bien tendre ; enlever le crayon et couper la petite boucle : on obtient une double torsade d'environ 16 cm dont la capacité mesurée vaut 40 pF environ.

Bobinage du cadre

Il faut bien reconnaître que notre montage fait un peu bricolage avec son assemblage par élastiques de bureau ! Néanmoins, le récepteur a effectué plusieurs voyages y compris deux expéditions postales sans problème.

On est ici dans le cas le plus simple : le bobinage à spires jointives ; percer à 10 ou 15 mm d'une extrémité un trou dans le tube isolant avec une miniperceuse ; passer dans ce trou depuis l'extérieur une longueur de fil émaillé suffisante pour pouvoir ensuite atteindre la cosse ; enrouler L1 en tournant le tube de la main droite (pour les droitiers) par l'extrémité opposée au trou, et en utilisant la main gauche pour tenir le tube, bloquer le fil qui doit toujours rester tendu, et avec

Réglage du filtre optionnel : si nécessaire, c'est-à-dire si un émetteur puissant et/ou proche gêne la réception des autres (sinon on peut le désactiver en court-circuitant L4 ou même ne pas installer L4, L5 et C8), se mettre à l'écoute d'une émission gênée et ajuster C8 au moyen d'un tournevis isolant pour atténuer le plus possible l'émission indésirable. Si l'extrémité du tournevis est métallique, elle doit être éloignée de la main par plusieurs cm d'isolant sinon le réglage sera faussé par sa capacité parasite dite « effet de main ».

Bonne écoute !

(120105)

- [1] Récepteur grandes ondes à faible consommation, Elektor, janvier 2012, p. 14 www.elektor.fr/110721
- [2] www.elektor.fr/120105

Bibliographie :

- Sir Douglas Hall | *Simplicity and sensitivity with 2 transistors* | *The radio constructor* | avril 1964 | l'ancêtre au germanium
- G. Short | *Silicon Transistor reflex TRF* | *The radio constructor* | janvier 1968 | l'origine d'une multitude de variantes
- C.Sinclair | *Slimline* | *The radio constructor* | septembre 1963 | réaction par retour capacitif sur le circuit d'accord
- R. Haig | *Practical radio circuits* | *Everyday practical electronics* | juin 2003 | filtre coupe-bande et réglage unique HF + BF + réaction

le pouce tasser chaque spire contre la précédente ; les spires doivent être côte à côte sans chevauchement ; deux spires avant la fin, percer un trou dans le tube. Les quatre trous devront être alignés le long de l'axe du tube de façon que tous les fils intérieurs passent le long du méplat du barreau de ferrite. Couper le fil en gardant une longueur suffisante, le passer dans le trou et le sortir du tube côté gauche. Si possible, mesurer l'inductance de L1 (provisoirement en position sur le barreau de ferrite) et corriger le nombre de spires si nécessaire (et, sans oublier d'enlever le barreau, percer un nouveau trou pour s'adapter à la nouvelle longueur).

Provisoirement bloquer la bobine avec du ruban adhésif, percer un troisième trou à 5 mm du deuxième et bobiner L2 dans le même sens selon le même procédé ; seule différence, les deux fils sortent du tube côté droit. Placer la plaquette à cosses contre le tube, tailler les fils à longueur, dénuder, torsader autour des cosses et souder, fils tendus ; il ne reste plus qu'à enfiler le tube sur le barreau, les fils passant dans l'espace entre le méplat et le tube. Quand on sera certain de ne plus rien avoir à modifier, on pourra remplacer l'adhésif par une goutte de colle sur chaque trou.

Comment mesurer une inductance ?

La méthode la plus simple consiste à l'associer à un condensateur fixe de valeur connue (si possible avec précision) et utiliser l'ensemble

dans un oscillateur dont on mesure la fréquence avec un oscilloscope ou un fréquencesmètre (**figure 5**). Ici nous ne pourrions pas échapper aux mathématiques : si on nomme F la fréquence de l'oscillateur en MHz, C la capacité du condensateur en pF, et L l'inductance de la

bobine en μH , ces trois valeurs sont liées par la relation : $(159/F)^2 = L \times C$

Exemple numérique : $L = 4,7 \text{ mH} = 4700 \mu\text{H}$ et $C = 10 \text{ nF} = 10000 \text{ pF}$ correspondent à $F = 0,023 \text{ MHz} = 23 \text{ kHz}$

La valeur de 10 nF a été choisie pour deux raisons : d'une part les fréquences basses obtenues sont compatibles même avec des instruments de bas de gamme, d'autre part les capacités parasites sont proportionnellement beaucoup plus faibles et risquent peu d'influer sur les résultats. Notre oscillateur est réalisé selon un schéma de B. Kainka. On peut difficilement trouver plus simple, et il a une couverture en fréquence absolument phénoménale.

D'autres indications et astuces pour la construction ont été données dans l'article sur la version GO [1] que nous recommandons de relire. D'éventuelles mise à jour ou corrections du présent article seront placées le cas échéant sur le site d'Elektor [2]. Le mois

prochain, nous revenons sur le sujet en remplaçant les ferrites par des cadres à air. Nostalgie garantie.

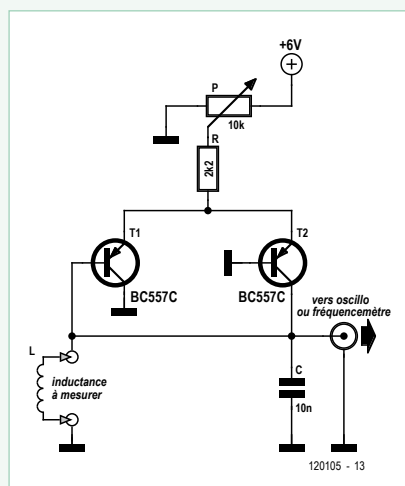
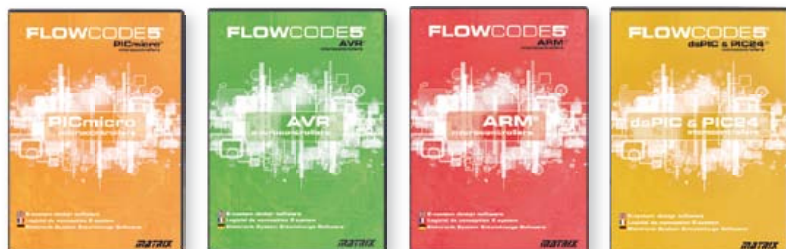


Figure 5. Cet oscillateur permet de mesurer une inductance. Il fonctionne d'ailleurs aussi avec le BC547B.

Flowcode 5 pour concev

FLOWCODE5

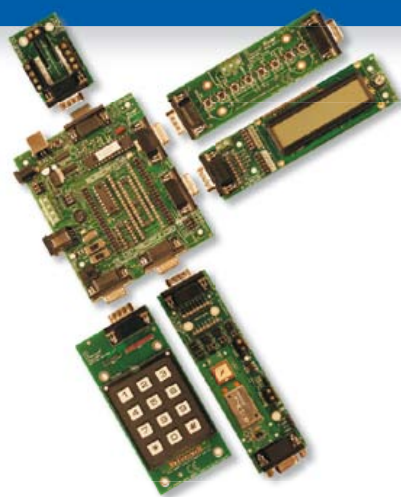


Flowcode 5 est l'un des langages de programmation graphique pour microcontrôleurs (PIC, AVR, ARM et dsPIC/PIC24) les plus avancés au monde. Son avantage principal est de permettre la création de systèmes électroniques et robotiques complexes même si l'on manque encore d'expérience.

Flowcode est utilisé :

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

... en électronique



Les E-blocks sont des circuits électroniques compacts, correspondant chacun à une fonction autonome comme on les trouve dans les circuits embarqués. Il en existe une quarantaine, dont la complexité va croissant, depuis le simple afficheur à LED jusqu'au circuits de programmation, aux modules Bluetooth ou TCP/IP.

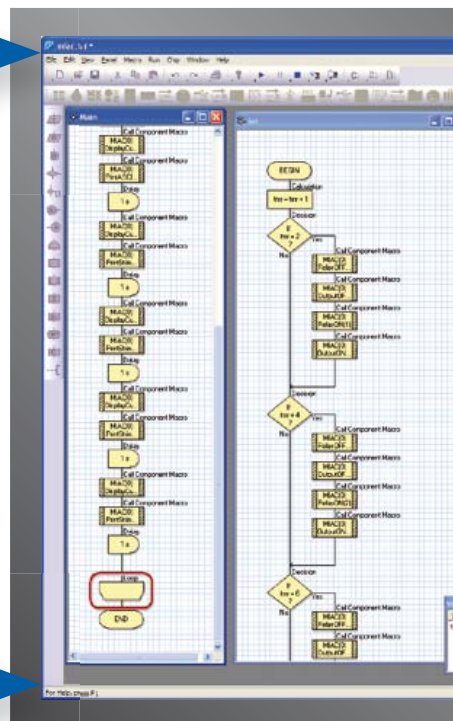
Les E-blocks peuvent être assemblés aisément pour élaborer des systèmes propices à l'apprentissage par l'expérimentation. Ils conviennent aussi pour le prototypage rapide de systèmes complexes. L'ensemble est complété efficacement par une gamme étendue et sans cesse renouvelée de logiciels puissants, et de capteurs variés.

... pour la commande industrielle



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

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



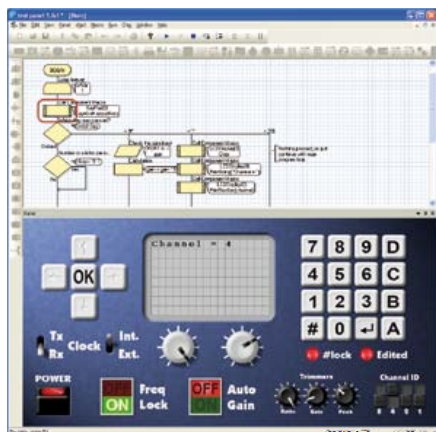
Outil de débogage FlowKit

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

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



... et pour se former...

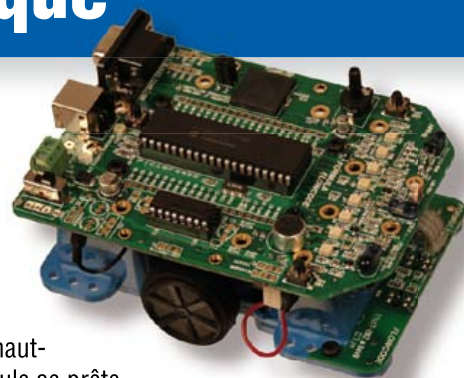


NOUVEAU dans Flowcode 5 :

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

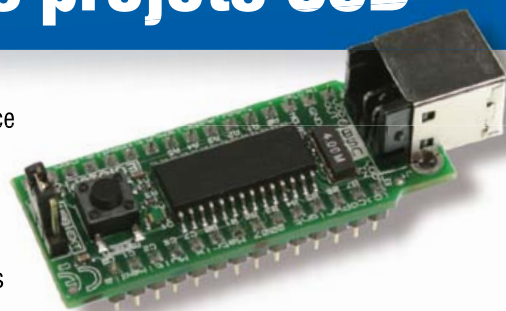
... en robotique

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



... pour les projets USB

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

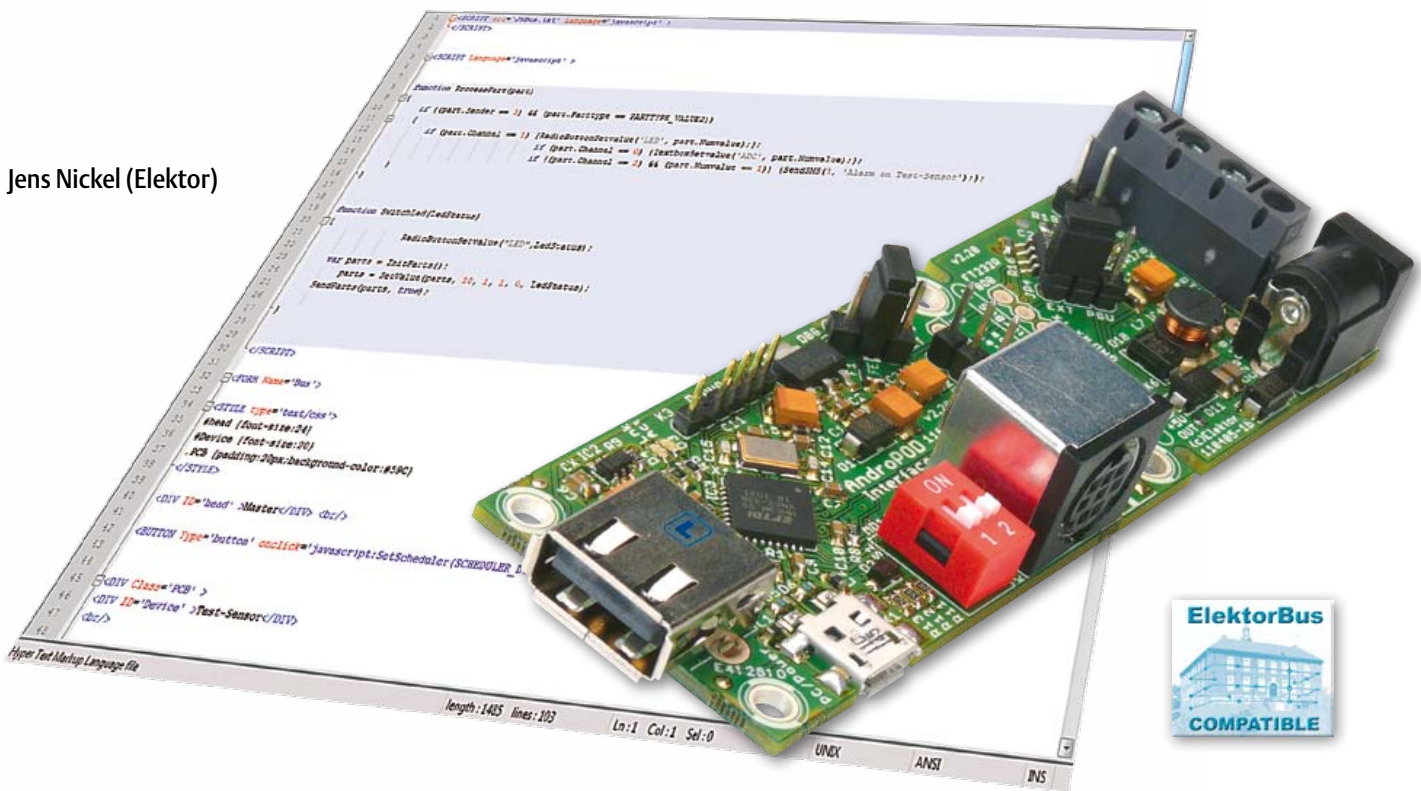


Retrouvez les E-blocks et leur documentation sur :
www.elektor.fr/eblocks

AndroPOD (2)

Interface utilisateur : la souplesse HTML

Jens Nickel (Elektor)



Marier vos circuits électroniques avec un *smartphone* pour qu'ils soient heureux et aient beaucoup d'enfants ? Vous en rêvez, AndroPOD le fait. Les entrailles de notre interface série pour téléphone ou tablette *Android* ont été disséquées le mois dernier. Vous avez apprécié la simplicité de son installation ? Vous aimerez son interface utilisateur écrite en HTML, souple, téléchargeable gratuitement, et vous la personnaliserez puisque son code source est ouvert. Tous à la noce de l'AndroPOD, épisode II.

C'est incontournable : pour commander une électronique depuis votre *smartphone*, il faut une application *Android* dotée d'une interface utilisateur. Comment implanter cette interface ? Comment la personnaliser ? Le puissant cadre d'applications *Android* et le langage de programmation *Java* sont sans doute les réponses auxquels penseront d'abord les développeurs. Programmer sous *Android* n'est toutefois pas évident. Pour le débutant, la courbe d'apprentissage est plutôt raide. L'effort sera payant, ne serait-ce que pour bien comprendre les projets *Android* de plus en plus souvent proposés par Elektor. Pour vous aider, l'un des développeurs de l'interface AndroPOD, Bernhard Wörndl-Aichriedler [1], étudiant à l'université de Hagenberg, en Autriche, a écrit une courte introduction à *Android* [2]. Vous pouvez aussi consulter la documentation de référence du SDK d'*Android* [3], en particulier le guide d'installation de l'EDI Eclipse [4] et de son extension ADT pour *Android*. La première chose à faire au démarrage d'un nouveau projet *Android* est d'implanter les fonctions de base d'AndroPOD (envoi et récep-

tion des octets via l'interface série). Ses développeurs ont à cet effet créé la classe *Java AndroPODConnection*. Vous pouvez la télécharger depuis [5] et la lier à vos projets *Android* (voir Tutoriel). L'étape suivante est la construction de l'interface utilisateur à partir des éléments d'interface graphique *Android* tels que boîtes de dialogue, boutons, etc. La structure d'une application AndroPOD est décrite dans l'encadré. Vous pouvez utiliser son code source comme point de départ de vos propres développements.

Smartphones et PC : même commande

Voici une autre approche : créer l'interface utilisateur à l'aide des langages HTML et *JavaScript*. Écrire une page HTML à l'aide des éléments du langage et programmer une commande en *JavaScript* ne relève pas du tour de force intellectuel. L'interface sera donc plus rapidement créée et sera, en outre, indépendante du système d'exploitation. Elle pourra donc être utilisée sur un *smartphone Android* ou toute autre plate-forme PC. Le recours à une interface série

adaptée reste bien sûr indispensable (par exemple le convertisseur USB/RS485 d'Elektor [6]).

Les pages HTML créées s'affichent en théorie directement dans le navigateur de l'appareil *Android*. Pour des raisons de sécurité, un navigateur standard ne peut toutefois pas recevoir ni envoyer d'octets sur une interface série. AndroPOD a donc besoin d'un navigateur spécial qui puisse à la fois servir d'interface utilisateur HTML et se charger de la communication via l'interface série (**fig. 1**). Nous l'avons écrit, baptisé *ElektorBusBrowserForAndroPOD*, et même publié sur l'*Android Market* de Google. Vous n'avez plus qu'à le télécharger et à l'installer sur un *smartphone* ou une tablette. L'exécutable .apk (l'équivalent sous PC d'un .exe) et le code source sont également disponibles sur le site Elektor. Cette application est déjà installée si vous avez construit le matériel d'essai du premier article [7], mais nous vous recommandons d'effectuer sa mise à jour. Il suffit ensuite de charger sur le téléphone les pages créées par l'application. Aucune difficulté ici, mais voyons d'abord la structure d'une telle page HTML.

Messages simples

Notre application *Android* envoie et reçoit des paquets de données (des messages) de 16 octets, construits conformément au protocole *ElektorMessageProtocol* [6]. Le débit est de 9600 bauds, d'emblée compatible avec nombre de futurs projets Elektor. Ce protocole a été développé pour le bus *ElektorBus* [8], mais n'est en aucun cas restreint au matériel *ElektorBus* ni même à RS-485. L'application fonctionne également lorsque les octets sont lus et envoyés par le connecteur mini-DIN ou les barrettes. Précisons bien que dans ce cas il est impératif de désactiver le pilote RS-485 par retrait du cavalier JP4 !

Dans le cas le plus simple, un message *ElektorMessage* possède la structure suivante (voir aussi la **figure 2**) :

Octet 0	170 = AA _{hex}
Octet 1	0
Octet 2	0
Octet 3	Adresse du destinataire (1...127)
Octet 4	0
Octet 5	Adresse de l'émetteur (1...127)
Octets 6 à 13	Données
Octets 14 à 15	Optionnel ; peut servir à la somme de contrôle / au CRC.

L'adresse par défaut du portable *Android* est 10 (vous pouvez la modifier via le menu de configuration). Le cas élémentaire est celui où l'on ne communique qu'avec un seul correspondant, par exemple avec un circuit à microcontrôleur. Supposons que l'adresse 1 lui soit attribuée : les messages allant du microcontrôleur au *smartphone* porteront la signature 170,0,0,1,0,10,... ; dans le sens inverse elle devra être 170,0,0,10,0,1,...

Les octets de données ne peuvent pas contenir la valeur 170 car elle est utilisée comme signal d'émission. Si cette restriction ou la

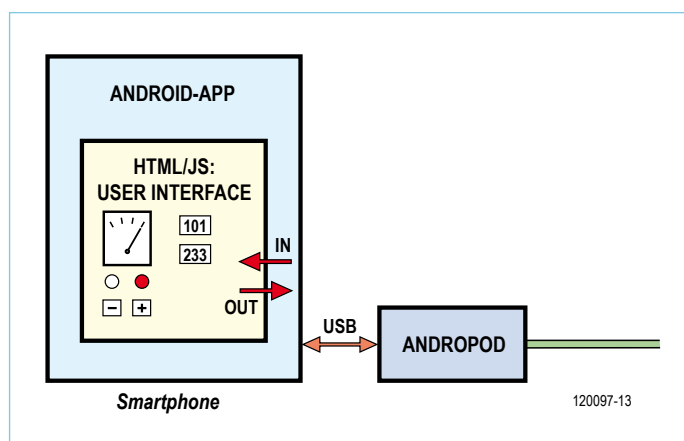


Figure 1. Notre application Android affiche une page HTML. Vous n'aurez que quelques éléments à modifier pour afficher la vôtre.

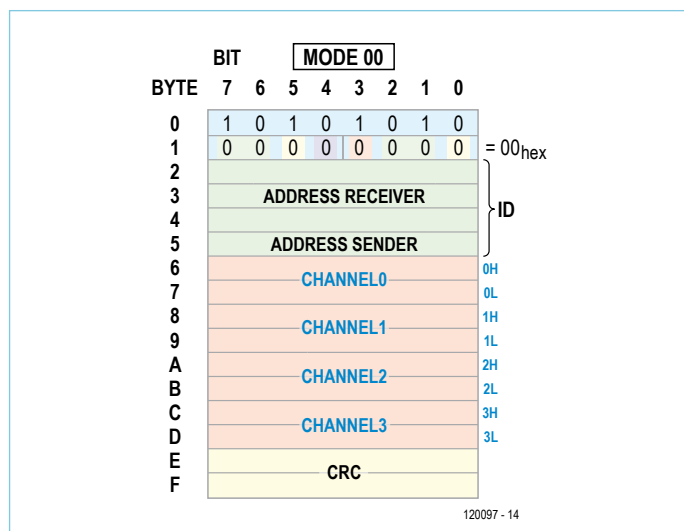


Figure 2. Structure d'un *ElektorMessage* ; sa longueur est fixée à 16 octets.

longueur fixe des messages vous gênent, lancez-vous : écrivez votre propre protocole ou modifiez en conséquence l'application *Android*. À moins de vouloir toucher aux fondements mêmes de l'application (voir l'encadré p.58), point besoin d'être docteur ès *Android* pour changer par exemple la longueur des messages.

Réception...

Lorsque l'application *Android* reçoit 16 octets, elle procède d'abord à un premier décodage des champs « émetteur », « destinataire » et « 8 octets de données utiles ». Le résultat est ensuite transmis à la page HTML de l'utilisateur AndroPOD. Pour terminer le décodage, le développeur prudent évitera ici de s'empêtrer dans des détails de programmation en utilisant JSBus, une bibliothèque *JavaScript* qui peut être liée aux pages HTML. Cette bibliothèque prend le paquet de 8 octets et complète le décodage (**fig. 3**). Seule condition à sa bonne utilisation : que les données aient été structurées selon l'*ElektorApplicationProtocol*. Ce protocole a été conçu pour la transmission des données de mesure (représentées sous la forme de nombres entiers et à virgule flottante), les changements d'unités et d'échelle

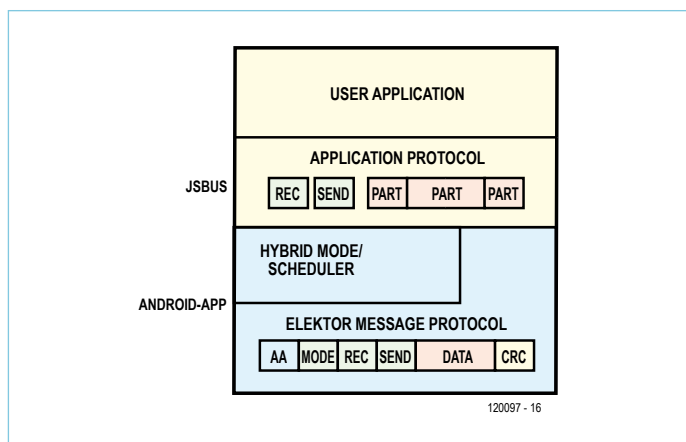


Figure 3. L'App Android se charge d'un premier décodage, JavaScript décode les données utiles.

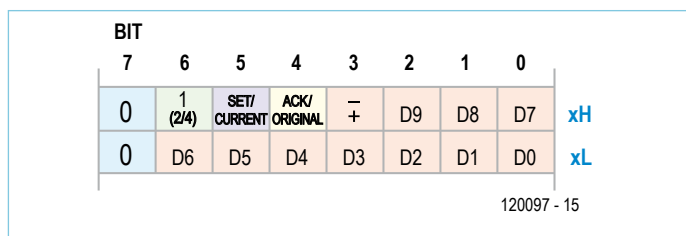


Figure 4. À deux octets de données correspond un intervalle de valeurs comprises entre -1023 et 1023, suffisant pour de nombreuses applications.

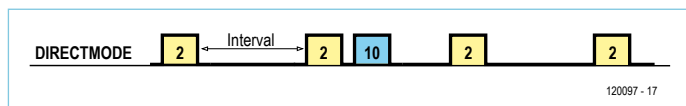


Figure 5. Le mode *DirectMode* évite les collisions de messages pour les communications de type 1:1.

(p. ex. de V à mV), le paramétrage des valeurs de consigne, l'affichage des valeurs limites inférieures et supérieures, plus quelques compléments des verbes « mesurer, commander, régler ». Les futurs projets Elektor (dont un multimètre) l'utiliseront et, comme nous le montrerons prochainement avec l'exemple d'une carte de circuit alimentée en 230 V, ce protocole peut bien sûr commander le matériel d'ElektorBus.

Dans un message (soit huit octets de données utiles), ce sont jusqu'à quatre valeurs de mesures (-1023 à 1023) qui peuvent être transmises simultanément sur quatre canaux (channels), chaque valeur occupant alors deux octets (fig. 4). La bibliothèque JavaScript sépare en quatre unités (parts) le paquet de 8 octets qu'elle reçoit. Ces quatre unités contiendront par exemple les quatre valeurs d'une certaine mesure. La bibliothèque appelle ensuite la routine JavaScript `ProcessPart(...)` qui traitera chacune d'entre elles. Pour insérer comme il se doit cette fonction dans une page HTML, il suffit d'y ajouter ces lignes :

```
<SCRIPT src='JSBus.txt' Language='javascript'
></SCRIPT>
<SCRIPT Language='javascript' >

function ProcessPart(part)
{
    // votre code
}

</SCRIPT>
```

Le corps de la fonction accueillera par exemple le code d'une zone de texte HTML chargée d'afficher la valeur de certaines mesures. La bibliothèque JSBus simplifiera là aussi votre travail. Supposons que vous vouliez afficher la première mesure (Channel=0) dans une zone de texte appelée *MaZoneTexte*. Écrivez simplement :

```
if (part.Channel == 0)
{TextboxSetvalue('MaZoneTexte', part.
Numvalue);};
```

La variable `part.Numvalue` contient toujours la valeur de mesure reçue (de -1023 à 1023). Un aperçu de toutes les fonctions de la bibliothèque JSBus ainsi que d'autres informations sont donnés en [9].

Les pages du site d'Elektor associées au projet ElektorBus offrent en téléchargement différents programmes de démonstration pour microcontrôleurs AVR. Inutile donc de réinventer le boulier, adaptez plutôt le code disponible à vos projets. Nous publierons dans le courant de l'année d'autres programmes pour microcontrôleurs compatibles avec le protocole décrit.

... et émission

Supposons que depuis notre portable nous voulions commander deux LED situées sur les canaux *Channel1* et *Channel2*. Nous utiliserons la fonction JavaScript suivante :

```
function SwitchLed(LedStatus1,LedStatus2)
{
    var parts = InitParts();
    parts = SetValue(parts, 10, 1, 1, 0,
LedStatus1);
    parts = SetValue(parts, 10, 1, 2, 0,
LedStatus2);
    SendParts(parts, true);
}
```

Les variables `LEDStatus1/2` doivent être affectées des valeurs 0 (hors tension) ou 1 (sous tension). Le code crée d'abord un tableau vide d'éléments appelés `parts` (les parties du message). Ces éléments `parts` contiennent chacun la valeur de mise hors ou sous tension des LED à transporter sur les deux canaux. Ils sont ensuite combinés, puis la fonction `SendParts` les envoie dans un seul message. La fonction `SwitchLed` peut être appelée de trois façons : avec un

bouton HTML, en réponse à un message entrant, ou commandée par un temporisateur. Ces trois possibilités sont illustrées dans les articles [9] et [10]. Les fichiers HTML correspondants sont à télécharger depuis les pages associées. Nous recommandons aux débutants d'étudier les morceaux de code généralement très courts avant de les adapter à leurs propres objectifs.

Éviter les collisions

Si vous utilisez l'extension RS-485, vous savez qu'il s'agit d'une communication sur deux lignes de données seulement (semi-duplex). Comment dès lors évite-t-on les collisions de message ? Car si émetteur et récepteur communiquent en même temps, c'est salade de bits au menu assurée pour tout le monde. L'application *Android* prend en charge une grande partie du problème par l'intermédiaire de deux modes appelés *DirectMode* et *HybridMode*.

Le premier est particulièrement adapté aux communications de type 1:1 (p. ex. entre un téléphone mobile et une électronique de mesure). Dans ce mode, l'électronique externe, par exemple un appareil de mesure, envoie ses valeurs ou ses messages d'état à intervalles réguliers. À l'inverse, les messages de commande à envoyer du *smartphone* à l'électronique externe sont envoyés dès réception d'une valeur de mesure (fig. 5). Le protocole *ApplicationProtocol* prévoit du reste qu'un capteur puisse envoyer à intervalles réguliers les valeurs mesurées [10]. Avec un débit de 9600 bauds, des intervalles de 100 ms sont possibles.

Ce *DirectMode* est activé dans une page HTML par le code *JavaScript* suivant :

```
SetScheduler(SCHEDULER_
DIRECTMODE,0,0,0,0,0,0,0,0);
```

Il est également possible de lancer cette commande avec un bouton HTML. Nous l'avons appelé « Activer le DirectMode » dans l'instruction suivante :

```
<BUTTON Type='button' onclick='javascript:
SetScheduler(SCHEDULER_
DIRECTMODE,0,0,0,0,0,0,0,0) ` >Activer le
DirectMode</BUTTON>
```

La syntaxe des langages HTML et *JavaScript* est facile à assimiler. Le lien [9] convaincra les débutants.

Le mode *HybridMode*, lui, est adapté à la communication entre plusieurs participants d'un même bus. Dans ce mode, un superviseur (*scheduler*) attribue des temps de parole aux nœuds du bus. L'application *Android* en contient un. Prenons l'exemple d'une interface AndroPOD reliée à un bus RS-485 avec deux participants appelés 1 et 2. La commande suivante lance le superviseur et lui communique l'adresse de tous les nœuds à prendre en compte :

```
SetScheduler(SCHEDULER_ON,2,1,10,0,0,0,0,0);
```

Jusqu'à huit adresses de nœuds peuvent être passées comme para-

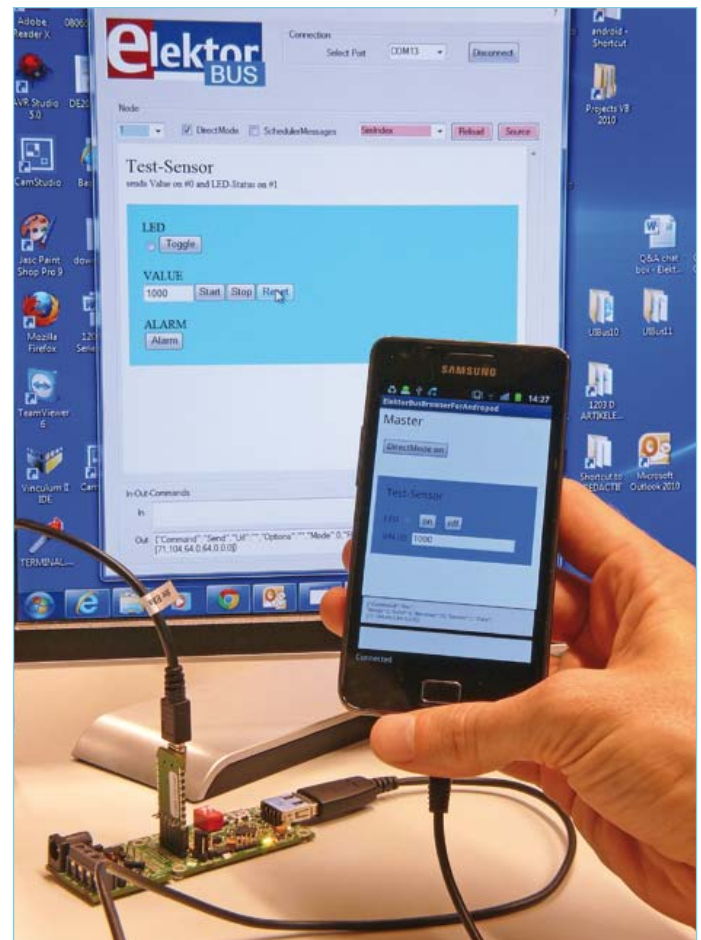


Figure 6. Un premier essai effectué avec un PC relié à AndroPOD, par exemple via le convertisseur USB/TTL BOB FT232.

mètres. Le 10 ne doit pas être oublié puisque le *smartphone Android* a aussi son mot à dire.

Démo

Concrétisons cette approche plutôt théorique à l'aide d'une petite application. Le « nœud expérimental » du projet ElektorBus [6] pourrait servir de matériel d'essai. Autre cobaye possible : un PC relié à l'interface AndroPOD, par exemple via le convertisseur USB/TTL BOB-FT232 [5] (fig. 6). Le fichier zip téléchargeable contient l'exécutable *ElektorBusElectronicsSimulator.exe* [5]. Ce programme sera l'interlocuteur de notre mobile puisqu'il peut envoyer et recevoir des messages. Nous lui ferons endosser le rôle d'un appareil de mesure et le représenterons à l'aide d'une page HTML. Placez le dossier *UIBus* (à télécharger depuis [5]) sur le Bureau de Windows pour que le programme PC puisse trouver la page HTML. Reste à transférer cette page ainsi que la bibliothèque *JavaScript JSBus.txt* dans le téléphone. L'application recherche les deux fichiers dans le dossier *ElektorBusBrowser* de la carte SD (ce comportement par défaut peut être modifié via le menu de configuration). Très bien, mais comment envoyer là-bas nos fichiers ?

Première possibilité pour envoyer des fichiers du PC au *smartphone*, le moyen traditionnel, p. ex. via un câble USB ou *Bluetooth*. Vous devrez désactiver puis réactiver l'ADB (le mode de débogage) si



Figure 7. Le programme PC simule une électronique de mesure externe.

vous utilisez un câble (voir l'article précédent [7]), et vous devrez en outre créer vous-même le dossier pour les fichiers.

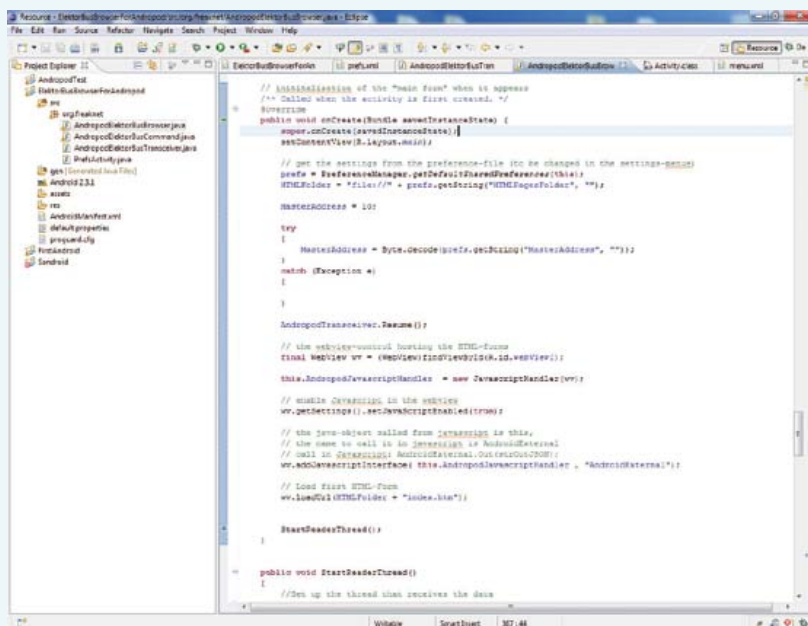
Brancher et rebrancher votre câble vous épuise ? Alors utilisez le programme *AdifController* des développeurs d'AndroPOD (cf. première partie de cet article). Sélectionnez l'onglet *Files*. Le téléphone devrait apparaître sous *Detected Phones* (si ce n'est pas le cas, mettez brièvement l'interface AndroPOD hors tension). Cliquez ensuite sur *Browse*, puis sélectionnez le dossier *UIBus* sur le Bureau. Appuyez enfin sur le bouton *Upload*.

Servez-vous du gestionnaire de fichiers pour vérifier que les fichiers ont été correctement transférés dans la carte mémoire du téléphone.

Essai

Nous sommes prêts à lancer l'application *Android* et le programme PC. Commençons par le second. Notre simulateur d'appareil de

Structure d'une application AndroPOD



Rien de tel qu'un exemple pour expliquer la structure d'une application *Android*. Nous avons choisi *ElektorBusBrowserForAndroPOD*. Vous pouvez télécharger son projet *Android* (pour *Eclipse*) depuis [5]. Le code source est dans le sous-dossier *src*. Nous l'avons déjà dit, le *framework Android* repose sur le langage *Java*. Il en reprend la syntaxe et plusieurs de ses bibliothèques importantes. Quelques connaissances préalables en *Java* sont donc un atout. Si vous êtes débutant, les livres sur *Java* ne manquent pas ; étudiez en particulier les chapitres qui abordent les *threads* et la programmation réseau. Ces sujets maîtrisés, vous comprendrez facilement la structure

d'ensemble de l'application et saurez la modifier à dessein. Connaître *Android* est indispensable pour créer une interface utilisateur native (c.-à-d. réalisée directement avec *Android*). Référence incontournable de votre apprentissage, le site *Android developers* [11]. Le lien [12] vous suggère un bon livre.

Notre application AndroPOD utilise trois classes. La classe *AndroPODElektorBusTransceiver* initialise une connexion TCP sur le port 1337, la ferme et l'établit à nouveau si nécessaire. Les méthodes *TransmitPacket* et *ReceivePacket* envoient ou reçoivent un paquet de 16 octets, avec comme marqueur de début de nouveau message la valeur « 170=AA_{hex} ». Vous devrez adapter le code si vous souhaitez une longueur de message différente.

Les objets de la classe *AndroPODElektorBusCommand* servent de conteneurs aux données d'un message entrant ou à envoyer. Elle contient un tableau d'octets appelé *Rawdata* qui reçoit les 16 octets d'un message. La méthode *parseRawData* décode le paquet. Les variables *Mode*, *Receiver*, *Sender* et *Data* sont alors remplies ; *Data* est de nouveau un tableau qui contient les données utiles du messages (8 octets). Le tableau de 16 octets *Rawdata* est à l'inverse rempli lorsque les valeurs des variables *Mode*, *Receiver*, *Sender* et *Data* sont transmises au constructeur de la classe *AndroPODElektorBusCommand*. La classe se charge également de transcrire ces valeurs dans une chaîne spéciale pour qu'un message décodé puisse être transmis

mesure s'ouvre dans une grande fenêtre (**fig. 7**). Le bon port COM (virtuel) doit être sélectionné avant d'appuyer sur *Connect*.

Le programme représente un capteur qui transmet régulièrement des valeurs. Ce serait p. ex. un compteur électrique. Le bouton *Start* de l'interface HTML envoie les valeurs relevées. Le bouton HTML *Toggle* simule l'allumage et l'extinction d'une LED de contrôle. Essayez. Les valeurs s'affichent sur l'interface *Master* du *smartphone* (**fig. 8**). Avez-vous remarqué le léger retard à l'affichage de la LED de contrôle ? Un changement d'état de la LED n'est en effet pas envoyé immédiatement, mais uniquement lorsqu'une nouvelle valeur de mesure quitte le PC. L'état de la LED et les valeurs de mesure sont transmises à l'intérieur d'un message sur différents *canaux*.

Nous aimerions maintenant que le maître renvoie quelque chose au capteur. Avant de pouvoir transmettre des messages dans l'autre

sens, nous devons tout d'abord activer le mode *DirectMode*. Nous avons placé le bouton correspondant dans l'interface HTML car la résolution d'écran d'un portable est inférieure à celle d'un PC. Les boutons *On* et *Off* de l'interface *Master* servent à définir ou initialiser l'état de la LED de contrôle du capteur. Jouons avec. Observez bien la réaction de la LED lorsque vous appuyez sur l'un ou l'autre. Son comportement ne vous aura pas surpris si vous avez été attentif : le *smartphone* n'envoie pas le changement d'état de la LED tant

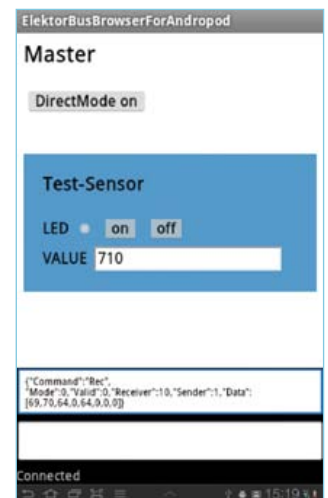


Figure 8. Capture d'écran d'une commande sur *smartphone*.

à la page HTML/JavaScript en cours d'affichage (voir [9] pour en savoir plus sur *InCommand* et *OutCommand*). Un développeur qui souhaite une distribution différente des octets dans un message adaptera cette classe en conséquence.

La classe *AndroPODElektorBusBrowser* gère l'affichage des données reçues, et reçoit en outre les données utilisateur qui déclenchent l'envoi de messages. Ces deux opérations se produisent dans une page HTML pour conserver l'indépendance de la plateforme vantée dans l'article. *AndroPODElektorBusBrowser* veille seulement à ce que cette page HTML soit affichée dans un composant *Android* du type *WebView*. Les messages reçus (plus précisément la chaîne qui en résulte) sont transmis à HTML/JavaScript via la méthode *In(...)*. C'est à l'inverse JavaScript, intégré dans la page HTML, qui passe une chaîne à la méthode *Out()* lorsqu'un message doit être envoyé. Le superviseur (*scheduler*) est lui aussi implanté dans cette classe. La méthode *Out(...)* n'envoie donc pas directement le message mais le tamponne. Le message n'est transmis que sur autorisation du superviseur (le mode *DirectMode* fait exception). Une partie de la classe est dédiée au protocole *ElektorBus*. L'ensemble du code à partir de la méthode *OnCreate* vous intéressera si vous souhaitez créer votre propre application. La classe est en effet une activité *Android*, plus précisément une activité appelée au démarrage de l'application (pour ainsi dire sa fenêtre principale). Les experts d'*Android* savent qu'en plus de *onCreate* (initialisation), les méthodes *OnStop*, *OnPause* et *onResume* doivent être modifiées ; ces trois méthodes contiennent respectivement le code à exécuter lorsque vous quittez l'application, lorsque vous reprenez une activité précédemment interrompue, et lorsque l'application repasse au premier plan.

Autre méthode d'importance pour *AndroPOD*, *StartReaderThread*. Nous la décrivons ci-dessous.

La méthode *onCreate* de la classe *AndroPODElektorBusBrowser* (ou de l'activité que vous avez programmée) est appelée au démarrage de l'application. La ligne de code *AndroPODTransceiver.Resume* active le *Transceiver* (émetteur-récepteur) de la classe *AndroPODElektorBusTransceiver*. La méthode *Resume(...)* ouvre d'abord un *ServerSocket* sur le port 1337. Une autre *thread* est ensuite démarré. Le code de la méthode *run(...)* de la classe *AndroPODElektorBusTransceiver* est alors exécuté. Il est contenu dans une boucle infinie qui vérifie en permanence s'il existe une connexion vers l'interface *AndroPOD*, et l'établit si elle n'existe pas. C'est grâce à ce *thread* que l'application en cours peut insérer l'interface.

Placée à la fin de la routine d'initialisation *onCreate*, la ligne de code *StartReaderThread()* ; démarre le *thread* de lecture proprement dit. Le code correspondant est lui aussi inclus dans une méthode *run()*, quelques lignes plus bas. Sa boucle *while* est exécutée tant que la connexion à l'interface est établie. La ligne *ReceivedMessage = AndroPODTransceiver.ReceivePacket()* ; lit les 16 octets suivants (à partir du bit de départ *AA_{hex}*). Le paquet peut ensuite être traité. Ajoutons pour être complet que le superviseur a lui aussi son propre *thread*.

Certes la technique vous semblera peut-être moins directe que la communication par tam-tam, mais envoyer un message reste dans le fond assez simple : il suffit d'appeler *AndroPODTransceiver.TransmitPacket(...)* (avec comme paramètre un *AndroPODElektorBusCommand*)

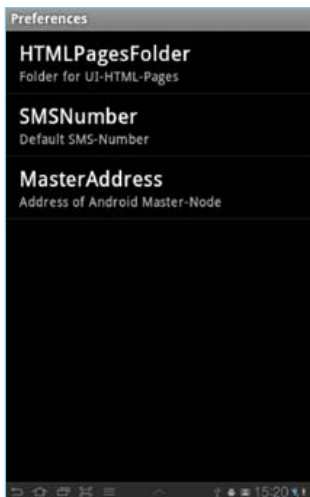


Figure 9. Paramétrage de notre App.

qu'un nouveau message (avec la valeur de mesure et l'ancien état de la LED de contrôle) n'a pas été reçu du PC. Cet expédient est la garantie d'un bus libre.

Essayons maintenant la fonction SMS. Nous nous servons d'un second mobile auquel nous enverrons un SMS d'alarme. Nous devons au préalable communiquer son numéro à l'application. Le paramétrage se fait dans *Settings*, un menu que nous avons déjà utilisé dans le précédent

article [7]. Pour y accéder, appuyez sur l'icône de menu qui se trouve en bas à gauche de l'écran de l'*Android*, puis cliquez sur *Settings*. Nous pouvons régler trois paramètres (fig. 9) :

- Le dossier dans lequel l'application recherche les fichiers HTML et *JavaScript*.
- Un numéro SMS par défaut.
- L'adresse *ElektorMessage* du smartphone *Android* (pré-réglée sur 10).

Le changement de ces paramètres provoque dans de très rares cas le blocage de l'application. Redémarrez-la si cela vous arrive.

Une fois le numéro saisi, cliquez sur le bouton *Alarm* du programme PC pour déclencher la fonction SMS. Un « 1 », qui signale une alarme du capteur, sera transmis par le prochain message envoyé via le canal *Channel2* encore libre.

La ligne *JavaScript* suivante, contenue dans la page HTML affichée sur le mobile, déclenche l'envoi du SMS :

```
SendSMS("1", "Alarme Capteur !");
```

Le paramètre 1 signifie que le SMS est envoyé au numéro par défaut ; vous pouvez toutefois entrer un numéro de téléphone quelconque.

Le programme PC permet de voir le code source des deux pages HTML utilisées. Celle qui représente une électronique de mesure s'appelle *SimIndex*, celle qui s'affiche sur le *smartphone* se nomme *Index*. Choisissez l'une d'entre elles dans le menu déroulant rouge clair, puis cliquez sur *Source* pour en voir le code.

Dans une application réelle, le contenu de certains messages pourrait être critique et leur perte tout autant. Un tel programme devrait donc comporter un mécanisme de confirmation d'envoi/réception pour parer à tout dysfonctionnement possible. Les protocoles *ElektorMessageProtocol* et *ApplicationProtocol* offrent à cet effet plusieurs solutions [8].

Perspectives

Complexe au premier abord, cette approche basée sur HTML présente de gros avantages. Là où programmer une interface utilisateur *Android* en *Java* exigerait temps et concentration, écrire une page HTML/*JavaScript* se fait pour ainsi dire le coude à la portière, sans parler de l'indépendance de la plate-forme ainsi créée (puisque la même interface pourra être utilisée sur *smartphone* et sur PC). Ajoutons que HTML a actuellement le vent en poupe. Attendez-vous d'ailleurs à ce que de plus en plus de programmes PC associés à des projets Elektor soient lancés depuis un navigateur !

Le prochain numéro présentera une « platine d'installation » équipée de deux relais, de deux entrées, et d'un micrologiciel compatible ElektorBus. Elle permettra de commander des éclairages et d'autres récepteurs depuis un *smartphone Android*. Vous aimez envoyer des messages d'état à vos amis ? Vous pourrez bientôt en envoyer à vos ampoules. D'autres projets sont en préparation, notamment dans le domaine de la mesure.

(120097 - trad. Hervé Moreau)

- [1] www.xdevelop.at
- [2] www.xdevelop.at/#category=projects&subcategory=1&anchor=6
- [3] <http://developer.Android.com/sdk/installing.html>
- [4] www.eclipse.org/downloads/
- [5] www.elektor.fr/120097
- [6] www.elektor.fr/110258
- [7] www.elektor.fr/110405
- [8] www.elektor.com/elektorbus
- [9] www.elektor.fr/110517
- [10] www.elektor.fr/110708
- [11] <http://developer.Android.com/guide/index.html>
- [12] „Learning Android”, Marko Gargenta, O'Reilly:
<http://shop.oreilly.com/product/0636920010883.do>

Produits & Services Elektor

- AndroPOD avec extension RS485 ; platine équipée et testée 110405-91
- Convertisseur USB/TTL BOB FT232, équipé et testé 110553-91
- Convertisseur RS485/USB, équipé et testé 110258-91

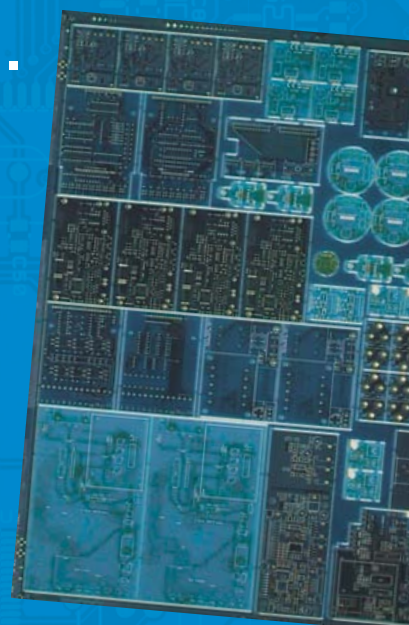
- Câble USB A/micro-B
- Alimentation pour smartphones avec connecteur USB micro-B
- Programmes à télécharger (gratuit)

Ces produits et les téléchargements sont disponibles sur la page associée à cet article : www.elektor.fr/120097

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

Accélérez votre cycle de développement tout en réduisant les coûts grâce nos services pooling rapides et faciles d'accès en ligne:

- Haute qualité professionnelle et prix modiques
- Livraisons rapides pour répondre à vos contraintes
- Soutien technologique performant
- Sans frais de dossier et d'outillages
- Sans minimum de commande à partir d'une pièce
- Sans règlement par avance
- Service pochoirs



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

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

Prix avec TVA Française de 19,6% transport non compris

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

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

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

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

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

- C.I. 1 face sur semelle métallique (SMI) – Isolants/conducteurs $\geq 200\mu\text{m}$
- Cuivre $35\mu\text{m}$ avec isolant $75\mu\text{m}$ sur semelle en aluminium 1.5mm
- A partir de 3 jours ouvrés

On demand – C.I. spéciaux minces et rigides

- 0 à 16 couches isolants/conducteurs $\geq 90\mu\text{m}$
- CI hyperfréquences et haut Tg
- A partir de 2 jours ouvrés

Cap sur l'avionique

recyclons un instrument de vol

Martin Ossmann (Allemagne)

Comme vous peut-être, je passe mes journées dans l'électronique, mais les merveilles mécaniques me fascinent aussi. Dans un entrepôt de surplus, je suis tombé récemment sur un *Horizontal Situation Indicator*, fringant instrument d'avionique, et je n'ai pas résisté à ce chef-d'œuvre d'électromécanique : je l'ai dépiauté, examiné sous toutes les coutures et analysé sa construction et ses fonctions. Embarquement immédiat, décollage vertical.



Dans un article [2] de février 2010, Elektor a déjà évoqué l'entrepôt de surplus [1] d'où vient mon *Horizontal Situation Indicator* [3] ou indicateur de position horizontale. C'est une combinaison d'instruments qui rassemble les affichages du VOR, *VHF Omnidirectional Range*, relèvement magnétique par rapport à un émetteur VHF connu, de l'*Instrument Landing System*, aide à l'atterrissage, et du *Height Indicator*, indicateur d'altitude.

Devant un tel appareil, on se demande s'il est possible d'en faire quelque chose d'utile. C'est ce que cet article va expliquer : sur quelles techniques repose cet instrument, comment animer soi-même les différents indicateurs. Pourquoi ne pas le coupler à un simulateur de vol, par exemple, ou s'en faire une horloge HSI pour le bureau ou une salle de réunion ? Pour admirer le résultat, mettez le cap sur *YouTube*, j'y ai posté une vidéo [4].

Synchros et moteurs

En regardant à l'intérieur de l'appareil par l'arrière (**figure 1**), on découvre la multiplicité de ses composants électromécaniques. Il y a là des parties à couvercle bleu et d'autres avec une capsule de métal nu. Les pièces bleues sont des composants du fabricant de matériel aéronautique et d'avionique Muirhead [5], sur lesquels il est indiqué que trois fils conduisent au côté primaire et deux vers le secondaire. Ces renseignements donnent à penser qu'il s'agit de ce qu'on appelle des synchros [5] ou résolveurs. On en trouve d'ailleurs une belle description en suivant les liens [6] et [7]. Les synchros, on les utilise normalement pour capter une information angulaire et la transmettre à un instrument d'affichage. On imagine bien que c'est le genre d'indication que l'on peut trouver sur une boussole,

par exemple, ou alors sur un gyrocompas de navigation, ici un compas de route. Pour présenter l'information dans le cockpit, il faut un système qui prolonge artificiellement l'axe mécanique, et on utilise un CX, un synchro-transmetteur. Il possède au stator trois enroulements décalés chacun de 120 degrés, qui forment le côté secondaire. Le primaire est constitué d'un bobinage fixé sur l'axe qui tourne, le rotor (cf. **figure 2** à gauche). Celui-ci est alimenté par un

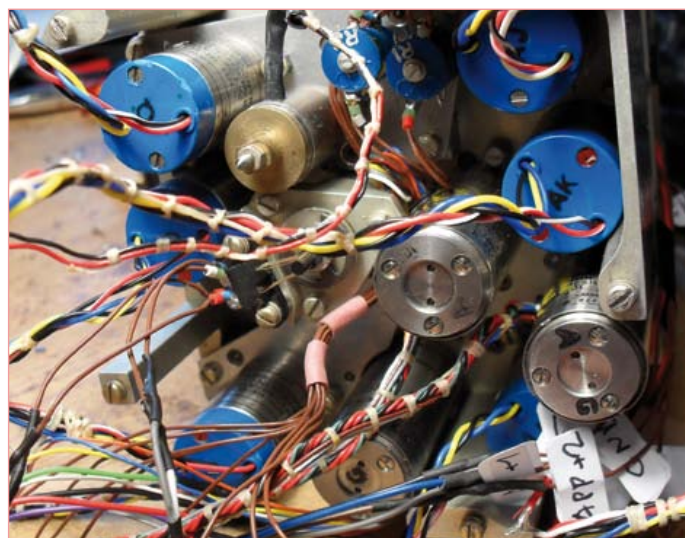


Figure 1. Coup d'œil de l'arrière sur les moteurs et synchros du HSI.

signal sinusoïdal à 400 Hz que l'on peut noter : $U_R(t) = \hat{U} \sin(2\pi t 400)$. Dans les trois bobines secondaires, la tension induite dépend alors de l'angle θ (thêta) qui traduit la position actuelle de l'axe du rotor. Ce qui peut se noter comme suit :

$$U_{S1}(t) = \hat{U} \sin(\theta) \sin(2\pi t 400)$$

$$U_{S2}(t) = \hat{U} \sin(\theta + 120^\circ) \sin(2\pi t 400)$$

$$U_{S3}(t) = \hat{U} \sin(\theta - 120^\circ) \sin(2\pi t 400)$$

Ces trois tensions contiennent à présent l'information angulaire que trois fils suffisent à transmettre plus loin. La partie droite de la figure 2 représente le symbole du CX, le synchro-transmetteur. Pour retrouver l'angle de rotation mécanique à partir des informations électriques des bobines, on réalise un circuit d'asservissement **CT**, un synchro-transformateur, composé d'un synchro-récepteur, d'un amplificateur et d'un servomoteur, comme à la **figure 3**. Le primaire du CT, le stator, est la copie conforme du secondaire du CX, avec trois bobines en étoile écartées de 120 degrés. Le secondaire, cette fois-ci, c'est le rotor.

Aux bornes de la bobine du rotor, on retrouve une tension alternative à 400 Hz dont l'amplitude est proportionnelle au sinus de (A-B). A, c'est l'angle que fait l'axe du CX par rapport à la référence, on peut penser au nord, par exemple, tandis que B, c'est pareil, mais pour le CT. La tension est exactement nulle si les deux angles sont égaux, puisque $\sin(0)$, c'est zéro. On voit tout de suite le parti à tirer de ce signal, il représente l'erreur de positionnement entre les deux axes et on va l'utiliser pour faire tourner le servomoteur jusqu'à ce que B soit égal à A. Et voilà comment tournent les différentes aiguilles dans notre HSI.

Embarquons sur un autre système

Si vous voulez mettre en œuvre ce principe de gouverne, il vous faut générer pour chaque aiguille les trois signaux électriques produits par le CX et construire le circuit d'asservissement. Mais ça peut aussi marcher autrement.

Si vous alimentez les bobines, comme le montre la **figure 4**, au moyen de deux signaux déphasés l'un par rapport à l'autre, il apparaîtra au rotor un signal à 400 Hz dont le décalage de phase correspond exactement à l'angle de l'axe. On peut alors traiter le signal par un convertisseur A/N (analogique/numérique) pour mesurer la phase et commander le moteur en conséquence. Et comme tous les CT peuvent être alimentés avec les mêmes signaux sinusoïdaux, ceux-ci n'ont à être produits qu'une seule fois.

Le générateur de signaux par MLI

Un signal sinusoïdal à la fréquence de 400 Hz, on peut fort bien l'obtenir par modulation de largeur d'impulsion (MLI) sur un processeur. Nous en avons pris un dont la fréquence d'horloge est de 20 MHz et qui peut fournir de la MLI rapide avec une résolution de 8 bits. La fréquence du signal MLI est alors de 20 MHz divisée par 256, soit 78,125 kHz, ce qui est très rapide par rapport à 400 Hz. Comme fréquence d'interruption, nous avons choisi 36×400 Hz, ce qui divise 20 MHz par 1 389. À chaque

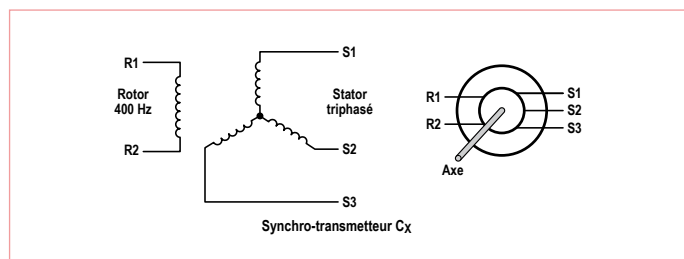


Figure 2. Constitution d'un synchro-transmetteur CX.

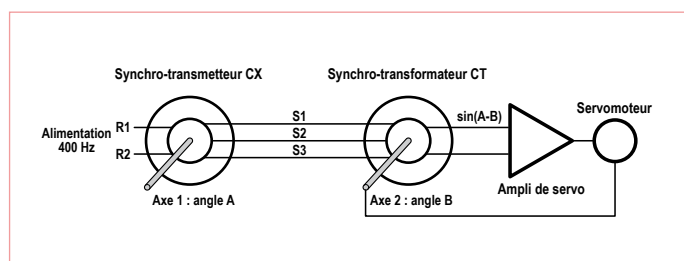


Figure 3. Allonge d'axe électrique.

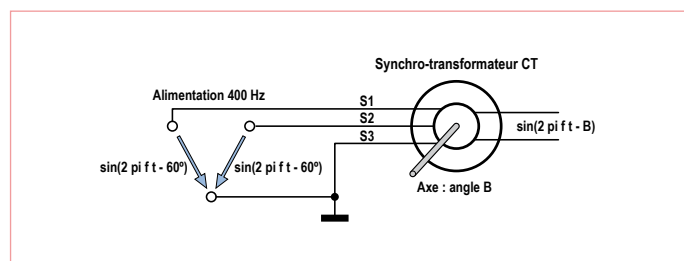


Figure 4. Une commande différente de synchro-transformateur CT.

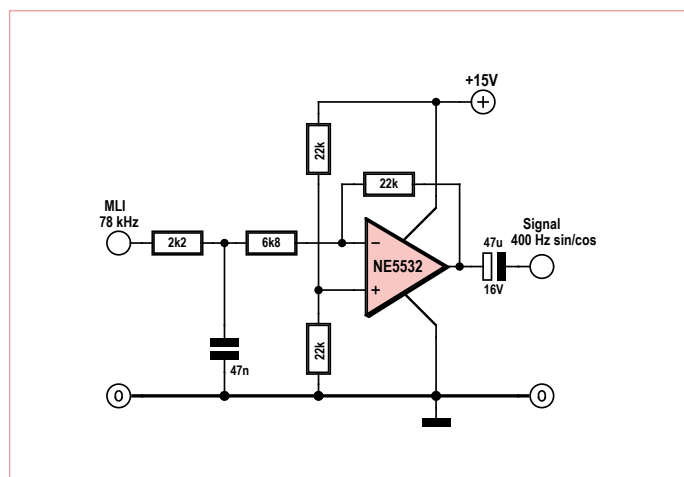


Figure 5. Filtre passe-bas et amplificateur pour le signal à 400 Hz.

DÉMONTÉ

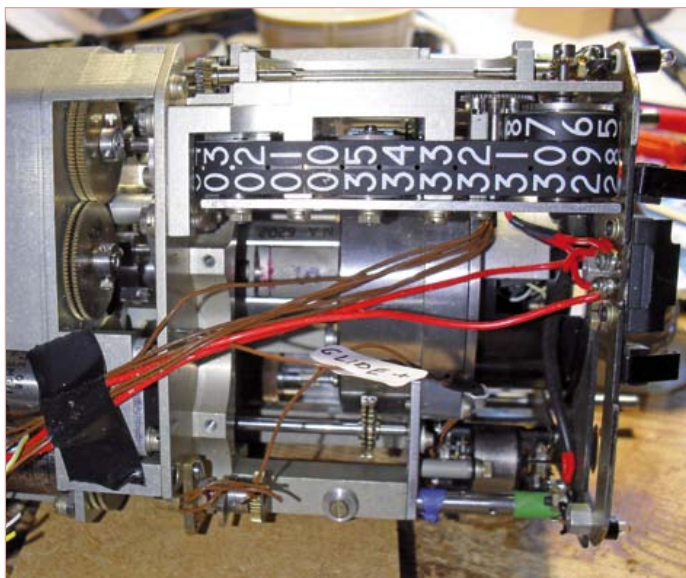


Figure 6. Vue d'en haut sur la mécanique.

interruption, on envoie une nouvelle valeur de sinus ou de cosinus au générateur MLI. Le signal MLI doit encore traverser un filtre passe-bas et être amplifié pour pouvoir exciter les enroulements du CT. La **figure 5** présente le schéma de l'un des deux canaux.

La mesure de phase

Sous quel angle se trouve l'axe mécanique ? Pour le calculer, déterminons la phase du signal du rotor. Appliquons-le au convertisseur A/N que nous allons consulter avec la même routine d'interruption que celle qui produit la MLI, donc pour obtenir $400 \times 36 = 14\,400$ échantillons par seconde. Une période correspond ainsi à 36 valeurs, lesquelles définissent la phase aussi bien que dans le discriminateur d'un récepteur SDR (la radio logicielle). Ensuite, pour déterminer l'angle, les composantes sinus et cosinus en quadrature de phase sont extraites du signal.

De cette façon, on peut obtenir les angles des différentes aiguilles de notre HSI et utiliser ces informations dans des circuits d'asservissement pour les faire virer vers la position souhaitée.

Aux commandes des moteurs

Exercice suivant : faire tourner les moteurs, dont il ne sort pas moins de 10 câbles (chacun). Comme on peut le déduire des inscriptions,

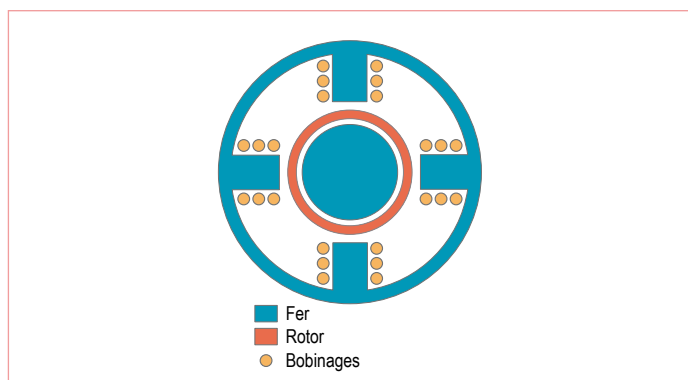


Figure 7. Constitution d'un moteur Drag-Cup.

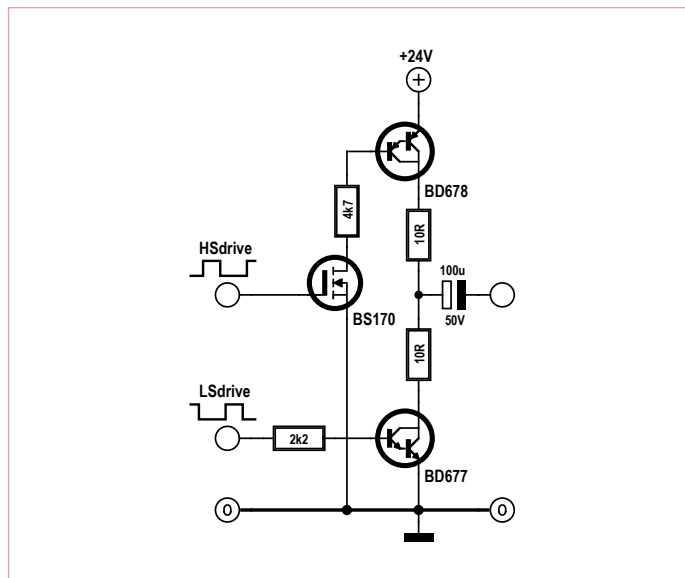


Figure 9. Le demi-pont à 24 V.

quatre d'entre eux conduisent au générateur de tachymétrie. On trouve encore trois autres bobinages à propos d'un desquels on lit « REF 26 V 400c/s ». Il faudra donc le brancher sur du 26 V à 400 Hz. Pas simple de dénicher des informations sur le raccordement de ce genre de moteurs, mais finalement, nous avons quand même trouvé des documents [8].

Manifestement, il s'agit d'un moteur à induit en cloche dit *drag-cup* [9] et d'une génératrice de tachymétrie. L'induit du moteur est construit comme sur un moteur asynchrone, mais le flux magnétique ne tourne pas à l'intérieur de l'induit. Le stator est fait d'un enroulement dit de référence alimenté en permanence par du 400 Hz. Un bobinage double est placé à angle droit avec lui. Si on lui applique un signal à 400 Hz déphasé de 90 degrés par rapport à la bobine de référence, l'induit pivote. Le sens de rotation change avec la polarité. Le bobinage est réalisé en deux parties pour faciliter la construction du circuit d'asservissement.

La **figure 7** représente schématiquement l'agencement d'un tel moteur. L'induit en forme de cloche tourne à l'intérieur d'un noyau de fer. Les deux enroulements placés à l'extérieur produisent le champ tournant [10] comme dans un moteur asynchrone à deux phases.

En faisant ses petites expériences, on s'aperçoit que le moteur tourne aussi quand la tension n'est pas sinusoïdale, mais tout bête-



Figure 8. Un moteur/génératrice tachymètre du HSI.

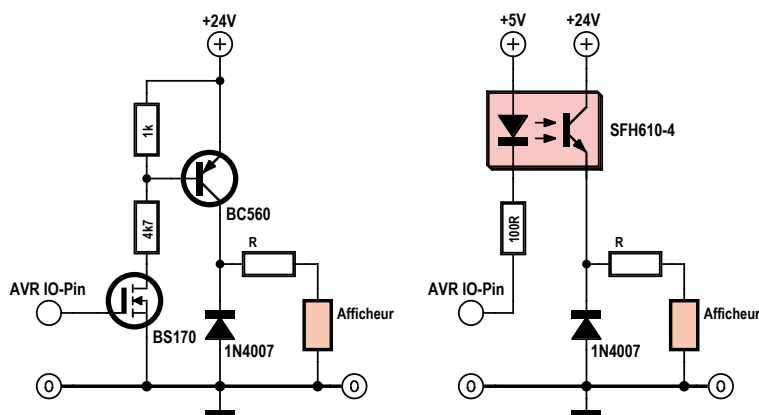


Figure 10. Deux tampons pour 24 V.

ment rectangulaire. Il suffirait alors de construire un simple circuit en demi-pont comme à la **figure 9**. Comme c'est le contrôleur qui distille les signaux de commande, on lui a demandé de laisser un temps mort entre les impulsions de niveau haut et de niveau bas, de manière à empêcher les deux Darlingtons d'être conducteurs en même temps.

Il faut alors deux demi-ponts de ce modèle-là, comme le montre la **figure 11**, pour alimenter un moteur. Son régime peut être réglé par déphasage.

Génératrice tachymètre

Il est aussi intéressant de savoir comment ça marche, une génératrice de tachymétrie. Celle-ci se compose d'une « bobine excitatrice » soumise à un signal sinusoïdal à 400 Hz. Tournée à 90 degrés, il y a une bobine de capteur dans laquelle aucune tension n'est induite quand le moteur est à l'arrêt. Dans le champ, on retrouve un induit en court-circuit comme dans la partie de moteur de la figure 7. Si ce rotor se met à tourner, on pourrait dire qu'il attire le champ d'excitation. Du coup, la symétrie est rompue et, dans le capteur à bobine, une tension décolle, proportionnelle à la vitesse de rotation. Comme il n'y a pas de balais, l'engin est particulièrement endurant.

Indicateur de course CDI

L'afficheur pour le VOR (relèvement magnétique sur balise VHF) est un CDI (*Course Deviation Indicator*). Il s'agit d'une aiguille qui oscille vers la gauche ou la droite pour donner l'ampleur de la déviation par rapport au cap choisi. Techniquement, le CDI se sert d'un instrument à cadre mobile. Pour obtenir des courants positifs et négatifs, nous avons réalisé le circuit dont la **figure 11** vous présente le schéma. Pour une MLI de rapport cyclique égal à 50 %, le pont est à l'équilibre et le CDI se positionne au milieu. Selon que le rapport cyclique sera plus grand ou plus petit, le CDI penchera à gauche ou à droite. Un potentiomètre d'ajustage permet de régler la sensibilité.

Indication To/From

L'indicateur de sens « vers ou d'où » du VOR est aussi réalisé avec un instrument à cadre mobile, commandé toutefois en numérique. On reprend le schéma de la figure 11 et on recommence, mais ici, on peut utiliser une sortie à trois états plutôt que de la MLI. Si la sortie est à haute impédance, le courant fourni est nul et ni le triangle TO ni le triangle FROM n'apparaissent. Envoyez un 0 ou un 1 à la sortie, vous activez alors l'un des deux triangles, dans l'ordre TO ou FROM.

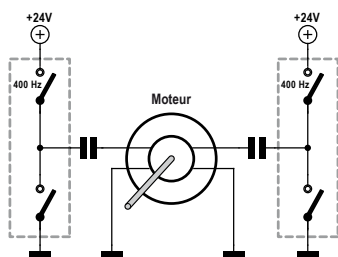


Figure 11. Comment piloter un moteur biphasé.

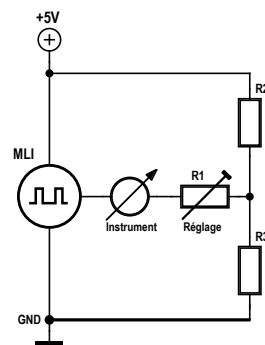


Figure 12. Montage en pont pour attaquer un galvanomètre du CDI.

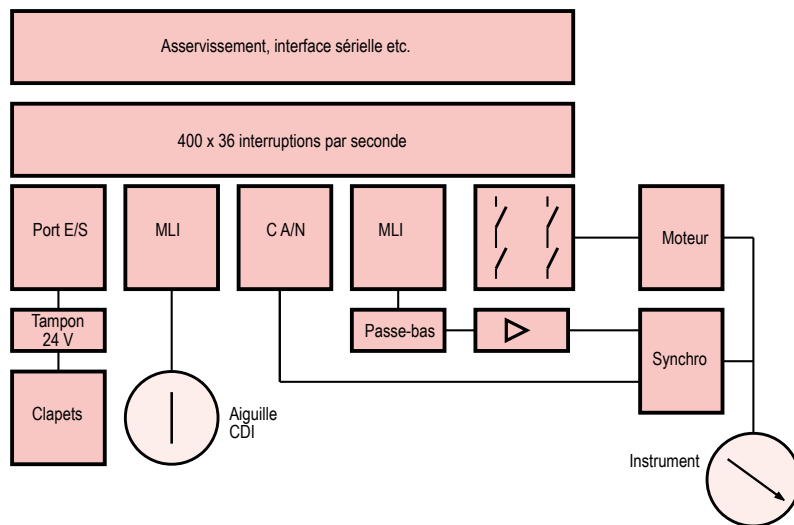


Figure 13. Synoptique : un ATmega644 cadencé à 20 MHz dispose en outre des temporisateurs et du convertisseur A/N nécessaires.

Quoi d'autre comme afficheur ?

Le HSI dispose encore de quelques autres moyens d'affichage que l'on actionne par de simples électroaimants, par exemple les clapets rouges visibles sur la première photo de cet article. Plusieurs de ces éléments ont parfois une masse commune et demandent plus de 12 V. Pour les mettre en oeuvre, on utilisera les circuits tampons de la **figure 10**.

Avec de faibles courants, un photocoupleur à grand gain suffira. Pour des courants un plus élevés, on se sert d'un BS170 comme élévateur de niveau et d'un BC560 comme commutateur. C'est chaque fois la résistance R qui limite le courant.

En finale

Cette volée de circuits proposés vous permet à présent de piloter à distance le HSI complet. La photo ci-contre montre mon proto sur platine perforée (**fig.14**). Elle rend bien l'ambiance joyeusement expérimentale de mon projet. J'espère qu'elle donnera à de nombreux lecteurs l'envie d'aborder ce terrain encore largement inexploré. L'électronique d'aujourd'hui, associée au goût de la récupération, procure encore bien des surprises et des satisfactions à celui qui persévère. À vous maintenant de réaliser votre propre cockpit de simulateur avec de **vrais** instruments d'aéronautique.

Bon vol !

(110756 - trad. Robert Grignard)

Liens :

- [1] www.helmut-singer.de
- [2] www.elektor.fr/090287
- [3] http://en.wikipedia.org/wiki/Horizontal_situation_indicator
- [4] www.youtube.com/user/ossimodding
- [5] www.muirheadaerospace.com/motion-technology/synchros.html
- [6] <http://en.wikipedia.org/wiki/Synchro> ou http://aviatechno.info/syn_tdr.php.html
- [7] www.ddc-web.com/documents/synhdbk.pdf
- [8] www.hnsa.org/doc/neets/mod15.pdf
- [9] www.google.com/patents (US-Patent 3641376)
- [10] http://books.google.fr/books/about/A_Textbook_of_Electrical_Machines.html, de R. K. Rajput, page A-5)

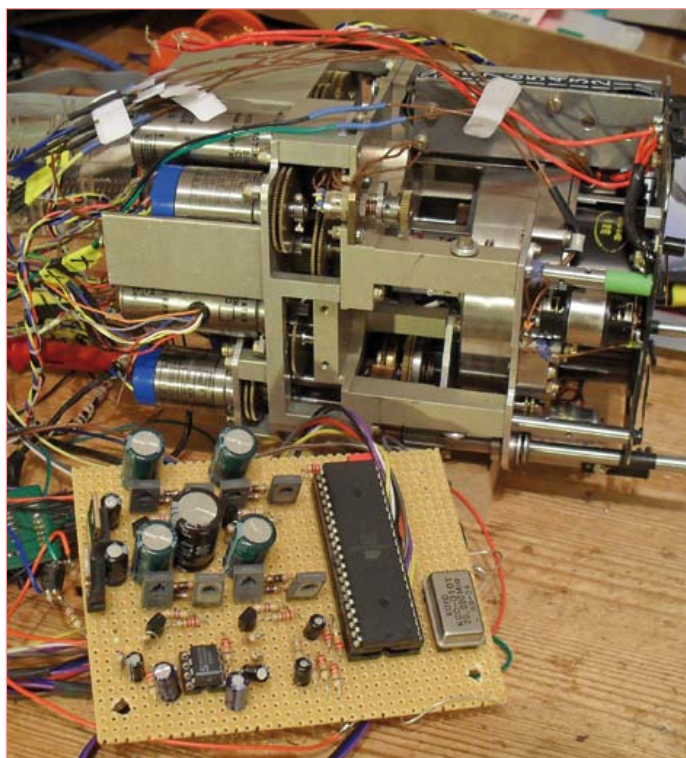


Figure 14. La platine d'expérimentation.

Abonnez-vous à la meilleure revue américaine sur les microcontrôleurs et leurs applications !



La formule d'abonnement qui vous conviendra est sur www.elektor.com/cc-subs

12 parutions par an

Édition numérique (PDF) : 38 \$

Édition imprimée : 63 \$

Les deux éditions : 90 \$

**CIRCUIT
CELLAR®**
THE MAGAZINE FOR COMPUTER APPLICATIONS

Platino dans la peau d'Arduino

Guide d'intégration pour votre propre matériel

Platino, la carte ultra modulaire dotée d'un AVR, présentée par Elektor en octobre 2011, a beau être compatible avec l'environnement de programmation Arduino,



tout n'est pas rose : Arduino n'est pas compatible avec tous les microcontrôleurs que Platino accepte ; il y a du portage dans l'air. Heureusement, toutes les informations nécessaires, éparpillées sur internet, sont ici rassemblées et ordonnées par votre serveur.

Cet article de synthèse exhaustif fait sauter un verrou : la dépendance de l'environnement vis-à-vis du matériel. Il donne accès à la facilité de programmation conférée par l'environnement *Arduino* sans la contrainte d'un matériel directement compatible. Il servira de référence et fera gagner un temps fou à quiconque souhaite se lancer dans une adaptation d'*Arduino* pour son matériel.

Clemens Valens (Elektor)

Vous savez que les microcontrôleurs de l'*Arduino Uno* et de ses prédécesseurs appartiennent à la famille de AVR ATmegaXX8 à 28 pattes d'Atmel. L'*Arduino Mega*, plus gros, utilise un ATmega2560 en boîtier à 100 pattes. Notre Platino est compatible non seulement avec les AVR en boîtier DIP 28 pattes, mais aussi avec les modèles à 40 pattes, qui eux ne fonctionnent pas avec Arduino. Si vous concevez votre propre carte, vous serez tenté de passer à un AVR en boîtier CMS avec encore un autre nombre de pattes, mais incompatible avec Arduino. Dans cet article, je montre comment modifier l'environnement Arduino pour l'utiliser facilement avec votre propre matériel. Laissez Platino vous guider.

Votre matériel dans Arduino 1.0

Il est plutôt facile d'intégrer votre propre matériel compatible Arduino dans l'environnement de programmation du même nom. Tout ce qu'il y a à faire, c'est ajouter un dossier dans le dossier *hardware* de la distribution Arduino (à savoir *arduino-1.0\hardware*). Ce dossier, qu'il conviendra de nommer correctement – ici *platino* – contiendra tous les fichiers spécifiques à votre matériel et, dans certains cas, des sortes de liens vers les fichiers *core* (le cœur ou le noyau, en quelque sorte) d'Arduino.

Afin de se prémunir contre d'éventuels problèmes avec les futures versions de l'EDI Arduino, il est probablement plus simple de copier le contenu du répertoire par défaut du matériel (*arduino-1.0*

hardware\arduino) dans le vôtre (par exemple *arduino-1.0\hardware\platino*) et de ne modifier que vos copies des fichiers. L'inconvénient de cette méthode est que vous n'intégrerez pas automatiquement les futures améliorations des fichiers du noyau Arduino, mais d'un autre côté vous n'aurez rien à modifier quand ceux-ci changeront.

Votre dossier devrait maintenant contenir quatre sous-dossiers :

- Bootloaders
- cores
- firmwares
- variants

et deux fichiers :

- boards.txt
- programmers.txt

Le dossier *bootloaders* contient le(s) chargeur(s) de démarrage (*bootloader*) pour votre matériel. Si votre carte comporte un micro incompatible avec les *bootloaders* tels qu'ils sont inclus, il faudra en modifier (au moins) un. Cette tâche, qui peut paraître décourageante, n'est en réalité pas si compliquée si vous vous cantonnez aux AVR. Nous en reparlerons.

Le cœur d'Arduino est dans les fichiers du dossier *cores* qui définissent les fonctions Arduino, dont certaines devront être adaptées à votre matériel. On peut voir ça comme le *Board Support Package* (BSP), pour utiliser une expression à la mode.

Le dossier *firmwares* contient les exécutables destinés aux cartes qui utilisent un ATmega8U2 pour la communication avec l'EDI. Effacez ce dossier si, comme Platino, votre matériel n'en utilise pas.

Le dossier `variants` est apparu avec Arduino 1.0. Il facilite la définition de plusieurs cartes qui ont beaucoup en commun. Dans le cas de Platino, nous pourrions définir deux variantes, une pour Platino avec un micro à 28 pattes et une pour Platino avec un micro à 40 pattes. Chaque variante possède son propre sous-dossier contenant le fichier `pins_arduino.h` qui se trouvait dans le dossier `cores` des précédentes versions. Si besoin est, vous pourrez y ajouter d'autres fichiers changeant avec les variantes. Effacez le fichier `programmers.txt` ; vous n'en aurez besoin que si vous souhaitez définir votre propre programmeur. Si vous le conservez sans le modifier, il y aura des doublons dans le menu *Tools->Programmer* de l'EDI.

Enfin, le fichier `boards.txt` contient les informations dont l'EDI a besoin pour utiliser les bons dossiers, protocoles, etc. en fonction du matériel sélectionné. Les cartes décrites dans ce fichier apparaissent dans le menu *Tools->Board* d'où elles peuvent être sélectionnées. Éditez ce fichier avant de modifier les autres.

boards.txt

Ce fichier définit les cartes connues de l'EDI. Il doit être édité afin que votre matériel soit reconnu. Pour ce faire, ouvrez-le dans un simple éditeur de texte tel que Bloc-note. Vous y verrez une liste de blocs comme celui du **listing 1** (sans numéros de lignes).

Effacez tous les blocs sauf celui qui correspond à votre matériel. Si vous n'effacez pas les blocs non modifiés, il y aura des doublons dans le menu *Tools->Boards*.

Toutes les lignes du **listing 1** commencent par `uno`. Il s'agit d'un identifiant pour la carte, unique parmi tous les fichiers `boards.txt` présents dans l'installation. L'EDI ne reconnaît qu'une des cartes qui partagent un même identifiant.

L'ordre des lignes n'a pas d'importance.

La ligne 1 contient l'étiquette `name` qui définit le nom de la carte tel qu'il apparaît dans l'EDI. Utilisez un nom explicite ; je ne connais pas la limite en nombre de caractères, mais il y en a certainement une, donc n'abusez pas non plus.

Les lignes 2 & 4 (`upload.protocol` et `upload.speed`) décrivent des paramètres passés à *avrdude*, le programmeur utilisé en standard. Arduino 1.0 en inclut la dernière version qui supporte le protocole `arduino`. Les versions incluses précédemment ne connaissaient pas ce protocole ; il fallait utiliser `stk500` à la place.

La ligne 3 définit la mémoire maximale disponible pour une application utilisateur. Elle est calculée en soustrayant la taille du *bootloader* de la taille mémoire totale. Ici, le micro possède 32 Ko et le *bootloader* occupe 512 octets, ce qui nous donne un maximum de 32,256 octets.

Les lignes 5 à 11 décrivent le *bootloader* et comment le charger depuis l'EDI (*Tools->Burn Bootloader*). Ce ne sont que des paramètres pour *avrdude*. Les fusibles et paramètres de verrouillage dépendent bien entendu du micro que l'on souhaite programmer. Le fichier *bootloader* nommé à la ligne 9 doit se trouver dans le sous-dossier du dossier `bootloaders` précisé à la ligne 8. Si vous ne souhaitez pas programmer le *bootloader* depuis l'EDI, tous ces paramètres peuvent être bidons. Vous pourrez également préciser un *bootloader* qui n'est pas compatible avec le protocole de com-

Listing 1. Une structure du fichier `boards.txt` définissant une carte pour l'EDI Arduino 1.0. J'ai trouvé opportun d'ajouter les numéros de lignes.

```
1  uno.name=Arduino Uno
2  uno.upload.protocol=arduino
3  uno.upload.maximum_size=32256
4  uno.upload.speed=115200
5  uno.bootloader.low_fuses=0xff
6  uno.bootloader.high_fuses=0xde
7  uno.bootloader.extended_fuses=0x05
8  uno.bootloader.path=optiboot
9  uno.bootloader.file=optiboot_atmega328.hex
10 uno.bootloader.unlock_bits=0x3F
11 uno.bootloader.lock_bits=0x0F
12 uno.build.mcu=atmega328p
13 uno.build.f_cpu=16000000L
14 uno.build.core=arduino
15 uno.build.variant=standard
```

munication Arduino (tant qu'*avrdude* sait comment le gérer), tel que *Wiring* par exemple.

La ligne 12 permet de préciser le micro présent sur la carte. Notez qu'il y a des types différents disponibles pour les AVR de type P. C'est important : l'identifiant n'est pas le même que pour ceux qui ne sont pas de type P. Le suffixe optionnel « A » n'a aucun effet.

La ligne 13 donne la fréquence de l'oscillateur utilisé en Hz ; cette valeur correspond à la fréquence du quartz et est nécessaire pour les fonctions *timer* et *UART* d'Arduino. Si vous souhaitez savoir où elle est utilisée, cherchez `F_CPU` dans les fichiers.

Les fichiers *core* devront se trouver dans le sous-répertoire du dossier *hardware* donné à la ligne 14 (ici : `arduino-1.0\hardware\arduino`). La ligne 15 précise le sous-dossier du dossier `variants` qui contiendra le fichier `pins_arduino.h` pour votre carte.

Ces deux dernières lignes peuvent pointer n'importe où, ce qui permet d'utiliser pour votre carte les fichiers *core* d'une autre. Il s'agit des liens dont je parlais plus haut.

Listing 2. Description d'une carte Platino équipée d'un ATmega164p. Les deux dernières lignes permettent que s'assurer que les bons fichiers *core* seront utilisés.

```
platino164p.name=Platino 164p(a) @ 16 MHz
platino164p.upload.protocol=arduino
platino164p.upload.maximum_size=15872
platino164p.upload.speed=115200
platino164p.bootloader.low_fuses=0xff
platino164p.bootloader.high_fuses=0xdc
platino164p.bootloader.extended_fuses=0xfd
platino164p.bootloader.path=optiboot
platino164p.bootloader.file=optiboot_
    platino164p.hex
platino164p.bootloader.unlock_bits=0x3f
platino164p.bootloader.lock_bits=0x0f
platino164p.build.mcu=atmega164p
platino164p.build.f_cpu=16000000L
platino164p.build.core=platino
platino164p.build.variant=ATmegaXX4
```



```

int analogRead(uint8_t pin)
{
    uint8_t low, high;

    #if defined(__AVR_ATmega1280__) || defined(__AVR_ATmega2560__)
        if (pin >= 54) pin -= 54; // allow for channel or pin numbers
    #elif defined(__AVR_ATmega3204__)
        // ...
    #elif defined(ATMEGA_X4) || defined(__AVR_ATmega16__) || defined(__AVR_ATmega32__)
        if (pin >= 24) pin -= 24; // allow for channel or pin numbers
    #else
        if (pin >= 14) pin -= 14; // allow for channel or pin numbers
    #endif

    #if defined(__AVR_ATmega3204__)
        pin = analogPinToChannel(pin);
        ADCSRB = (ADCSRB & ~(1 << MUX5)) | (((pin >> 3) & 0x01) << MUX5);
    #elif defined(ADCSRB) && defined(MUX5)
        // the MUX5 bit of ADCSRB selects whether we're reading from channels
        // 0 to 7 (MUX5 low) or 8 to 15 (MUX5 high).
        ADCSRB = (ADCSRB & ~(1 << MUX5)) | (((pin >> 3) & 0x01) << MUX5);
    #endif

    // set the analog reference (high two bits of ADMUX) and select the
    // channel (low 4 bits). this also sets ADLAR (left-adjust result)
    // to 0 (the default)
    #if defined(ADMUX)
        ADMUX = (analog_reference << 6) | (pin & 0x07);
    #endif
}

```

Figure 1. Extrait du fichier wiring_analog.c, montrant une modification de code pour le portage de Platino.

Le **Listing 2** est une adaptation du **listing 1** pour une carte Platino équipée d'un ATmega164p. J'ai fait de même pour toutes les configurations auxquelles j'ai pensé, ce qui donne un total de 16 cartes dans l'EDI. La variante utilisée est ATmegaXX4 (micros à 40 pattes), j'ai aussi créé une variante ATmegaXX8 pour les micros à 28 pattes.

Un seul objet pour tout Platino

Les autres fichiers *core* sont identiques pour les deux variantes grâce à l'utilisation de `#ifdefs` qui guident le compilateur.

pins_arduino.h

Une fois les cartes décrites correctement, nous pouvons passer à la définition des pattes dans le fichier `pins_arduino.h` du dossier `variants`. Pour l'exemple de Platino avec un ATmega164p, le fichier se trouve dans `arduino-1.0\hardware\platino\variants\ATmegaXX4`.

Vous devrez définir dans ce fichier le nombre de pattes capables de fonctionner comme E/S, le nombre d'entrées analogiques, les pattes qui peuvent faire de la MLI, celles qui sont reliées à un *timer*, le port SPI, etc. Toutes ces définitions sont des macros et des structures de données ; vous aurez besoin de la fiche technique du micro. L'ordre des structures de données détermine les numéros d'E/S numérique dans Arduino. L'E/S numérique 0 est la première définition de chaque structure. Veillez à utiliser le même ordre pour toutes les structures, sinon les pattes et fonctions risquent de se mélanger. Les entrées analogiques sont gérées différemment et pas dans ce fichier.

Avant de modifier quoi que ce soit, assurez-vous de comprendre ce que vous faites. Ce n'est pas sorcier, mais pas anodin non plus. Vous trouverez dans mes fichiers des `#ifdefs` concernant l'ATmega16 et l'ATmega32. J'aurais pu créer une autre variante pour ces micros à 40 pattes, mais les différences étaient trop minimes pour le justifier. Au passage, la définition du micro (`__AVR_ATmega16__` et compagnie) provient de la ligne 12 du **listing 1**.

Modification des fichiers du noyau Arduino (*core*)

Le dossier `cores` (`arduino-1.0\hardware\arduino\cores\`) contient un dossier `arduino` regroupant les fichiers *core* d'Arduino ; au moment où j'écris ces lignes, il y en a 36. Juste pour ajouter

votre carte, cela fait beaucoup, mais, en fait, peu d'entre eux sont à modifier. Pour Platino, j'ai préféré mettre les réels ajouts dans des fichiers à part et n'ai eu à modifier que quatre fichiers (de ma copie de `cores`). Voici mes modifications :

Arduino.h

Ce fichier est inclus par beaucoup d'autres fichiers du noyau et bibliothèques, ce qui en fait l'endroit idéal pour ajouter des liens vers votre code. J'ai écrit mes fonctions Platino en C++ et j'ai donc ajouté un `#include «Platino.h»` à la liste en fin de fichier, juste en dessous de `#ifdef __cplusplus`.

J'ai aussi dû définir les références de tensions `INTERNAL1V1` et `INTERNAL2V56` pour les micros à 40 pattes sélectionnés par un `#elif defined(ATMEGA_X4)`. La macro `ATMEGA_X4` est définie dans le fichier `devices.h` créé par moi et inclus au début du fichier.

Tone.cpp

Normalement on n'a pas à modifier ce fichier, bizarre comme le rappelle le commentaire. Cela dit, et afin de s'assurer de la compatibilité future, il est probablement mieux de le faire quand même. J'ai copié le code existant que j'ai fait précéder d'un `#elif defined(ATMEGA_X4)`.

Wiring_analog.c

La fonction `analogRead` doit être modifiée afin de pouvoir faire correspondre des numéros d'E/S numériques à des entrées analogiques. Il ne s'agit que d'une ligne de code précédée, une fois de plus, par un `#elif defined(ATMEGA_X4)` (**figure 1**).

Wiring_private.h

Ce fichier contient les définitions des interruptions. Les AVR à 40 pattes ont trois entrées d'interruption externes et il faudra ajouter une définition au fichier. Une fois encore, cherchez un `#elif defined(ATMEGA_X4)` pour voir comment j'ai fait.

Vous vous demandez peut-être pourquoi je n'ai fait la modification que pour les micros à 40 pattes. Tout simplement parce que les modèles à 28 pattes sont déjà compatibles avec Arduino et que Platino est compatible avec les Arduinos classiques, tels le Uno : il n'y avait rien à faire.

Ajouter des fichiers au noyau

Si votre carte comporte un périphérique qui n'est pas présent sur les Arduino standards, lisez bien ce paragraphe. Platino intègre un LCD, des poussoirs et/ou encodeurs rotatifs, un buzzer, une LED RVB et des cavaliers de configuration. Ces périphériques peuvent bien sûr être gérés par des bibliothèques (à mettre dans `arduino-1.0\libraries`) — la bibliothèque Arduino *LiquidCrystal LCD* par

exemple fonctionne avec Platino – mais cela rend le portage plus compliqué avec plus de chemins à surveiller. J'ai préféré intégrer les fonctions Platino directement dans l'environnement Arduino et ajouter mon code dans ma copie du noyau.

Ajouter des fichiers au noyau est facile : le compilateur traite tout ce qu'il trouve dans ce dossier. Voici, par ordre alphabétique, ce que j'y ai ajouté et pourquoi :

devices.h

La compilation des fichiers *core* pour Platino requiert des définitions. Étant donné que Platino est compatible avec plusieurs AVR en plusieurs versions, les `#ifdef` ont tendance à devenir longs. J'ai par conséquent décidé de créer deux groupes, un groupe pour les micros à 40 pattes (`ATMEGA_X4`) et un pour ceux à 28 pattes (`ATMEGA_X8`). Les nouveaux micros s'ajoutent ici.

LiquidCrystal cpp & h

La bibliothèque LCD d'Arduino fonctionne avec Platino, je vous l'ai déjà dit, mais une modification est nécessaire pour l'utiliser depuis l'objet Platino (voir plus bas). Je l'ai copiée dans mon dossier *cores* pour faciliter la compilation et la maintenance.

Cette bibliothèque ne possède pas de constructeur par défaut, j'ai donc ajouté la ligne `LiquidCrystal(void) {}` à la définition de la classe dans le fichier d'en-tête. Vous ne connaissez pas C++ ? Bien, si vraiment vous voulez savoir, j'en avais besoin parce que ma classe Platino définit un objet *LiquidCrystal* sans l'initialiser et que la bibliothèque ne comportait pas de fonction pour le faire.

Platino.cpp & h

Pour rendre les périphériques Platino faciles d'accès, j'ai créé un objet Platino pratique sous la forme d'une classe. Créez une instance de cette classe et vous aurez accès à tous les périphériques Platino. Il s'agit en fait d'un *wrapper* (n.d.t : programme d'encapsulation) pour les classes gérant les périphériques, mais un qui sait gérer les cavaliers de configuration de Platino. J'en reparlerai.

PushButton cpp & h

Un objet poussoir a été implémenté dans ces deux fichiers. Vous pourrez accéder directement ou aux poussoirs ou profiter de l'antirebond inclus. La classe de l'encodeur rotatif utilise celle-ci (voir ci-dessous).

RotaryEncoder cpp & h

Les encodeurs rotatifs sont en fait deux (ou trois si on peut appuyer dessus) poussoirs couplés mécaniquement. La classe les gérera en tant que tels. Cette classe n'utilise ni interruptions (entrée de capture) ni *timer* (temporisation, en français) et l'utilisateur doit la cadencer périodiquement, par exemple depuis la fonction `loop` du programme. Attention aux fonctions (éventuellement bloquantes) d'autres bibliothèques ou aux boucles d'attente : elles peuvent ralentir la boucle principale.

Le poussoir optionnel d'un encodeur rotatif n'est pas géré par cette classe ; cela devra être fait, comme avec n'importe quel poussoir, via la classe *PushButton*.



PCB-POOL
L'ORIGINAL DEPUIS 1994
Beta LAYOUT

Pochoir CMS gratuit
Un pochoir CMS est offert avec chaque commande "Prototype"

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

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

PROTEUS P-CAD 2006 TARGEM ALTIUM Designer cadence EDWIN GraphiCode PULSONIX Easy-PC

PCB-POOL® est la marque déposée de **Beta LAYOUT**

www.pcb-pool.com



Schaeffer AG

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

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

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

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

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

Schaeffer AG · Nahmitzer Damm 32 · D-12277 Berlin · Tel +49 (0)30 805 86 95-30
Fax +49 (0)30 805 86 95-33 · Web info.fr@schaeffer-ag.de · www.schaeffer-ag.de

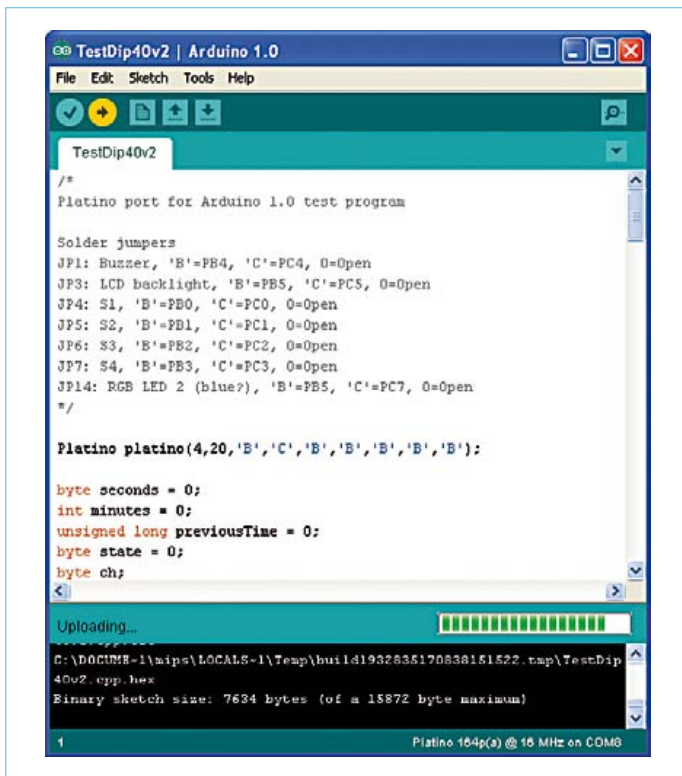


Figure 2. Platino dans Arduino. La carte sélectionnée est visible dans le coin inférieur droit. L'objet Platino est invoqué au début du programme (au centre de la capture d'écran).

Utilisation

Une fois le portage fini, peut commencer l'utilisation. La première étape est de sélectionner la carte depuis le menu *Tools->Boards* de l'EDI. Si vous avez correctement adapté le fichier *boards.txt* et que vous avez accès à un des programmeurs connus de l'EDI, vous devriez pouvoir programmer le *bootloader* via *Tools->Burn Bootloader*. Une fois la programmation faite, nous pouvons passer au programme.

Le plus dur ici sera certainement de trouver un *bootloader* utilisable. J'en ai essayé plusieurs et j'ai trouvé une solution presque universelle ; certains m'ont toutefois posé problème, mais je ne pouvais



Figure 3. Le programme de test de la Figure 2 tournant sur Platino. La valeur du coin inférieur gauche (302) est contrôlée par l'encodeur rotatif à droite ; le timer du coin supérieur droit peut être remis à zéro à l'aide du poussoir de gauche.

pas creuser faute de temps. L'archive qui accompagne cet article contient un *bootloader* pour (presque) tous les AVR utilisables avec Platino – à utiliser à vos risques et périls. Ils ne casseront rien, mais pourraient bien ne rien programmer non plus. Actuellement, j'utilise un ATmega164p avec le *bootloader* fourni et ça fonctionne bien. Le premier programme que vous écrirez vous aidera à valider les attributions des pattes dans Arduino. L'E/S numérique X se trouve-t-elle bien là où elle devrait ? De même pour les E/S analogiques. N'oubliez pas d'essayer la bibliothèque *Arduino SoftwareSerial* pour voir si les interruptions fonctionnent comme prévu. Un coup d'oeil à la fonction *attachInterrupt* ? Si tout ceci fonctionne bien vous pouvez avoir confiance en votre portage. Vous êtes maintenant prêt à démarrer.

Utilisation de la classe Platino

Le simple fait d'instancier la classe Platino vous prendra environ 2,5 Ko de mémoire programme, c'est vrai, mais la programmation sera grandement simplifiée. Écrivez en haut de votre programme (figure 2) :

```
Platino platino(4,20,'B','C','B','B','B','B','B');
```

ce qui veut dire que vous utilisez un afficheur à cristaux liquides de 4 lignes de 20 caractères, et que les cavaliers soudables 1, 3 à 7 et 14 sélectionnent le port B ou C (indiqué par un 'B' ou un 'C'; un cavalier non positionné peut être indiqué par un '0'). Accédez ensuite aux périphériques de Platino comme suit :

```
platino.led(état); // état vaut 1 ou 0
if (platino.pushButton(1)==0) ... // vérifier le
    poussoir 1 (actif au niveau bas)
if (platino.encoder2.tick(counter)!=0) ... // à
    appeler souvent, 1 kHz si possible
platino.beep(fréquence,durée_en_ms);
platino.lcd.setCursor(0,1); // aller à la
    colonne 0 de la ligne 1
platino.lcd.print(«portez ce vieux whisky»);
```

Sympa, hein ?

La ligne 3 mérite quelques explications. Il s'agit d'un appel à la fonction de cadencement de l'encodeur rotatif n°2, qui met à jour *counter* dans le même temps. Lorsque cette fonction renvoie une valeur non-nulle, *counter* a changé. Comme l'indique le commentaire, il faudra appeler cette fonction souvent sous peine de rater des événements.

Bien sûr toutes les fonctions publiques des classes telles que *LiquidLCD* seront utilisables dans votre programme tant que vous n'oubliez pas de les préfixer avec « platino point » (*platino.*).

L'archive spécialement compilée pour cet article contient également quelques programmes de test susceptibles de vous aider à vous familiariser avec la classe Platino.

Amusez-vous bien !

(120094 - trad. Kévin PETIT)



Allez **au bout** de vos **idées!**

Relevez **le défi** DesignSpark chipKIT™

10.000 \$ de prix à votre portée...!

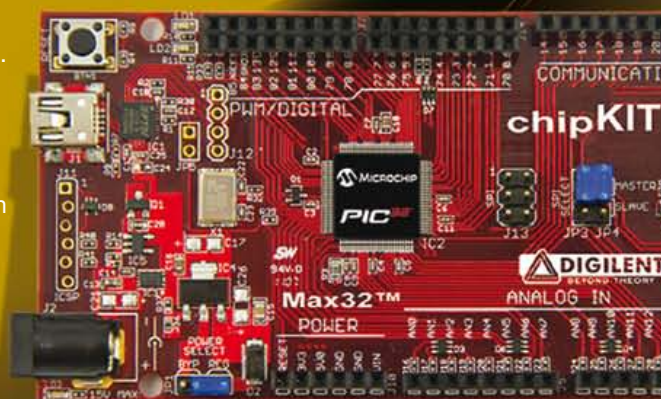
C'est maintenant ou jamais qu'il convient de mobiliser vos ressources créatives. La date de clôture du concours **DesignSpark chipKIT™ Challenge** est en vue. Mettez la dernière main à vos schémas et à vos programmes, puis ficelez bien votre dossier pour en mettre plein la vue au jury!

Est-ce votre projet qui sauvera le monde? En tout cas, il devra contribuer à réduire la consommation d'énergie, ou du moins son gaspillage. Le seul moyen de savoir si vous êtes sur la bonne voie est de participer.

Pour gérer votre projet en ligne, cliquez sur l'onglet My Projects du site www.designspark.com/chipkitchallenge-projects/latest. Poussez sur le serveur tout le matériau nécessaire au jury pour se convaincre de l'intérêt de votre projet, et n'oubliez ni la description ni le code source.

Détails et conseils sur www.designspark.com/chipkitchallenge/faq.

Halte à la procrastination! Le concours DesignSpark chipKIT™ prend fin le 27 mars 2012 à 18.00 GMT.



Règlement et informations détaillées:
www.chipkitchallenge.com

Ce concours est organisé en association avec:



Concours RS DesignSpark ChipKIT™



Ian Bromley (Royaume-Uni)

Dans son dernier mois, le concours de conception d'applications économes en énergie DesignSpark chipKIT™ a déjà permis de voir des projets inattendus et novateurs mais il reste ouvert... si vous êtes rapide...

Le concours touche à sa fin et il ne vous reste que quelques semaines pour tenter de gagner un prix. Lancé dans la bonne humeur en novembre dernier, il vise à encourager les ingénieurs, étudiants et autres amateurs à innover dans le domaine des économies d'énergie, tout en surveillant l'empreinte écologique. Il y a 10.000 \$ à gagner, dont un premier prix de 5.000 \$. À l'apogée du concours, les participants auront certainement bien avancé leur projet basé sur la plateforme compatible Arduino de Digilent, j'ai nommé le chipKIT™ Max32™.

Basée sur un microcontrôleur PIC32 de Microchip, elle permet aux développeurs d'intégrer facilement et à moindre coût de l'électronique dans leurs projets. Le matériel est compatible avec les cartes d'extensions (*shields*) 3,3V et les applications Arduino et peut-être utilisé avec une version modifiée de l'EDI Arduino ; à vous les programmes d'exemple, bibliothèques et références du monde Arduino !

Certains ont toutefois eu quelques problèmes, surtout avec les pilotes USB et la connexion des cartes aux PC. Voici quelques conseils qui devraient être utiles aux nouveaux participants : *primo*, les pilotes pour une large gamme de puces FTDI, fournis dans la bibliothèque d'outils de l'EDI *open source* multiplateforme MPIDE, devraient régler la plupart des problèmes de connexion ; *deuxio*, si vous faites partie des rares personnes à avoir des problèmes avec Windows 7, utilisez le mode « XP » si possible.

Pour nous mettre en bouche, détaillons quelques exemples de ce que vous pourrez trouver sur le site de la communauté en ligne DesignSpark www.designspark.com.

Les propositions couvrent des domaines aussi variés que la domotique, le contrôle ou la surveillance de charge de panneaux solaires ou d'éoliennes. Certains projets sont plus exotiques, comme le contrôleur de planeur sous-marin autonome, ou l'afficheur de consommation instantanée en km/l. Pratique pour conduire « éco » avec une vieille voiture.

Le planeur est très intéressant, la méthode de propulsion employée consomme peu et lui confère une très bonne autonomie. Une carte d'extension, en cours de développement, se chargera de l'interface avec les capteurs et actionneurs. Il y aura un accéléromètre, un gyroscope et un magnétomètre, tous trois sur 3 axes, pour

l'orientation en plongée et un GPS pour déterminer la position absolue en surface. Le propulseur à ballast, qui utilise les variations de la flottabilité, et les actionneurs de contrôle du tangage et du roulis se chargent des déplacements. Ajoutez à ça un capteur à transfert de charges pour mesurer la conductivité de l'eau, la température et la profondeur et les données enregistrées dans la mémoire flash permettent de dresser un profil caractérisé de la salinité.

Autre projet intéressant : un système de contrôle de l'arrosage et des nutriments pour la culture hydroponique. Il surveillera les conditions climatiques et déterminera les besoins des plantes, économisant eau et produits chimiques. La plupart des fermes hydroponiques utilisent des régimes d'alimentation constants, sans prise en compte des conditions environnementales, les économies sont donc quasi assurées.

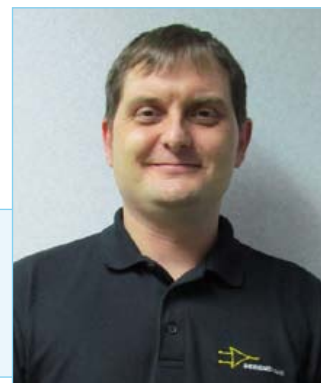
Enfin, un « jardin intelligent » permettant de contrôler la croissance d'une plante avec une intervention humaine minimale, et combinant les sources d'énergie renouvelables, comme l'énergie solaire, à la faible consommation de la carte pour réduire significativement l'empreinte écologique. Le système embarquera divers capteurs d'humidité, température ou luminosité et des actionneurs tels qu'une pompe d'irrigation. Modulaire, il pourra être étendu, par exemple en ajoutant du Bluetooth pour les notifications utilisateur ; une idée que nous adorerions voir se développer cette année, à l'heure du tout internet...

Inutile de dire qu'il ne s'agit que de trois des nombreux projets en cours de développement.

Les projets seront jugés sur l'efficacité énergétique et la qualité de conception du circuit imprimé de la carte d'extension. Les copies sont à rendre le 27 mars 2012 et les gagnants seront annoncés en avril.

(120188 - trad. Kévin PETIT)

Détails et inscription sur : chipkitchallenge.com.

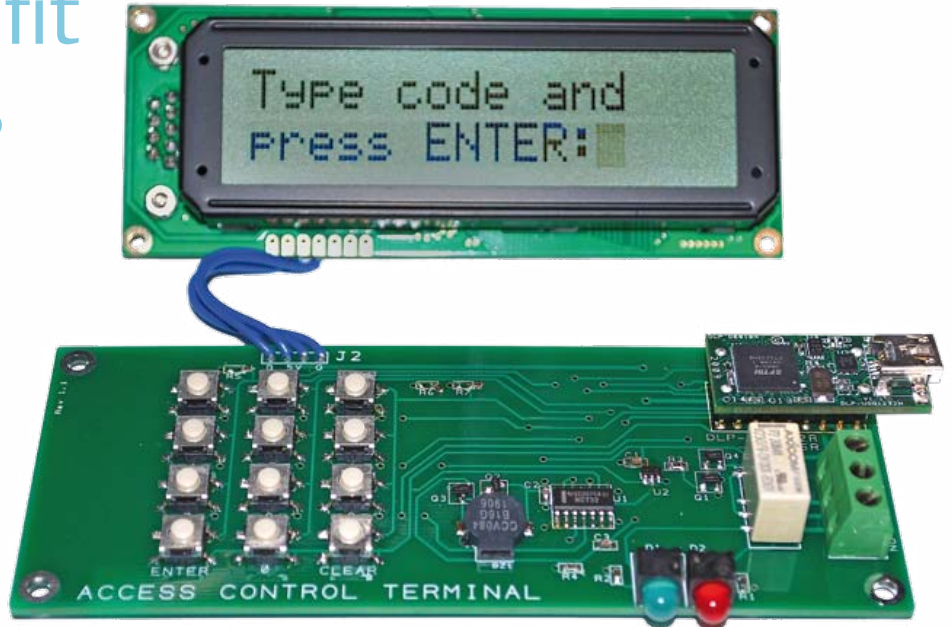


Ian Bromley, ingénieur marketing technique chez RS, dirige le projet logiciel DesignSpark PCB. Avant RS, Ian a travaillé comme consultant en support technique chez Texas Instruments, en plus du travail d'ingénieur de terrain (FAE) commencé immédiatement après l'obtention, en 1994, de son diplôme en d'ingénierie en microélectronique.

Attaque bit à bit du module USB FTDI

pour tirer profit de propriétés méconnues des circuits intégrés USB de FTDI

Don Powrie (États-Unis d'Amérique)



Cet article décrit la conception électrique et les conditions logicielles d'une serrure électronique sans clé, composée d'un pavé numérique, d'un affichage à cristaux liquides, des contacts d'un relais pour déverrouiller la porte et d'une interface USB. Même si cet article plonge dans les entrailles du circuit intégré FT232H de FTDI et son mode *bit à bit*, la technique est compréhensible sans connaissance approfondie de l'USB ni le recours à un microcontrôleur !

Je supposerai pour commencer que le lecteur s'est déjà un peu familiarisé avec la gamme de circuits intégrés USB de FTDI — faciles à utiliser — avant de se pencher sur quelques-unes de leurs caractéristiques les moins connues. Si vous n'avez jamais été en contact jusqu'ici avec ces composants, mon conseil est de potasser leurs caractéristiques et applications à l'aide de quelques-unes de mes publications précédentes [1]. Pour revenir à ce projet, tout le logiciel utilisateur résidera dans une seule application sur le PC hôte, et les seuls circuits intégrés seront le FT232H et quelques portes logiques simples. Le module FTDI utilisé est en vente chez *DigiKey*, *Mouser Electronics* et autres distributeurs FTDI.

Principes du Bit-Banging

Une fois les pilotes USB installés sur le PC et le port ouvert au module USB (j'ai utilisé le

DLP-USB1232H pour faciliter l'assemblage), le mode *Bit-Bang* peut être activé. Le code source en VC++ de ce projet peut être téléchargé de la page [2]. La commande D2XX pour activer le mode *Bit-Bang* est

```
FT_STATUS status = FT_SetBitMode(m_ftHandle, 0x01, 0x01)
```

La « poignée » (*handle*) est renvoyée par la commande d'ouverture du port, le deuxième paramètre indique quelles lignes sont des entrées et sorties, le troisième précise le niveau logique initial des lignes configurées en sorties.

Pour lire le niveau logique des lignes configurées en entrées, il faudrait utiliser la fonction `FT_GetBitMode(m_ftHandle, &data)`. Le paramètre *data* désigne l'état actuel des entrées. Le point important à retenir est que cette fonction envoie l'état instantané des entrées. Inversement, les données écrites

(au moyen de la fonction `FT_Write()`) n'apparaissent pas instantanément sur les broches de sortie. Au lieu de cela, les données apparaissent à une cadence de mise à jour prédéfinie. Si la cadence de mise à jour (ou le débit

en bauds) est de 9600 et que vous envoyez plusieurs octets de données à la fois, chaque octet apparaîtra automatiquement sur les lignes de sortie — un à la fois — toutes les 104 µs jusqu'à ce que le dernier soit passé.

Les puces USB de FTDI ont toujours été capables de fonctionner ainsi. Toutefois, c'est seulement depuis l'apparition de leurs nouvelles puces à grande vitesse que la cadence de mise à jour peut être commandée précisément et que huit flux sériels simultanés peuvent être produits avec un débit en bauds précis pour attaquer des

récepteurs sériels avec un débit stable. Par exemple, le code suivant fixera la cadence de mise à jour pour le débit requis par le module LCD et l'interface TTL que j'utilise dans ce projet :

des données en retour ; c'est parfait pour notre projet puisque nous ne faisons que piloter un afficheur LCD (*Crystalfontz America* réf. CFA632-YFB-KS) avec des données sérieelles aux niveaux TTL et que nous

Une seule variable à huit bits

L'application de l'hôte suit toutes les entrées et sorties, y compris le flux de données vers l'afficheur, avec une seule variable à huit bits. Pour lire l'état logique d'une ligne d'E/S

```
div = 0x8c30;//35888 décimal pour 19200 bauds vers le LCD avec une erreur de 0,6%
status = FT_SetDivisor(m_ftHandle, div);
```

Remarquez que les données sérielles peuvent seulement être **émises** à une cadence définie. Malheureusement, aucune réponse sérielle ne peut être lue sur une ligne d'entrée. Il faudrait utiliser le deuxième canal du circuit intégré USB pour recevoir

n'avons que faire de données en retour. Maintenant que nous avons accès à huit lignes d'entrées-sorties capables de débiter des données sérielles à la cadence désirée, la plate-forme matérielle est prête pour notre projet.

configurée en entrée, vous devez appeler la fonction `FT_GetBitMode()` et appliquer un masque aux données reçues de façon à voir un bit unique. Pour changer l'état logique d'une sortie, vous commencerez par mettre à jour l'état du bit visé dans la variable à huit

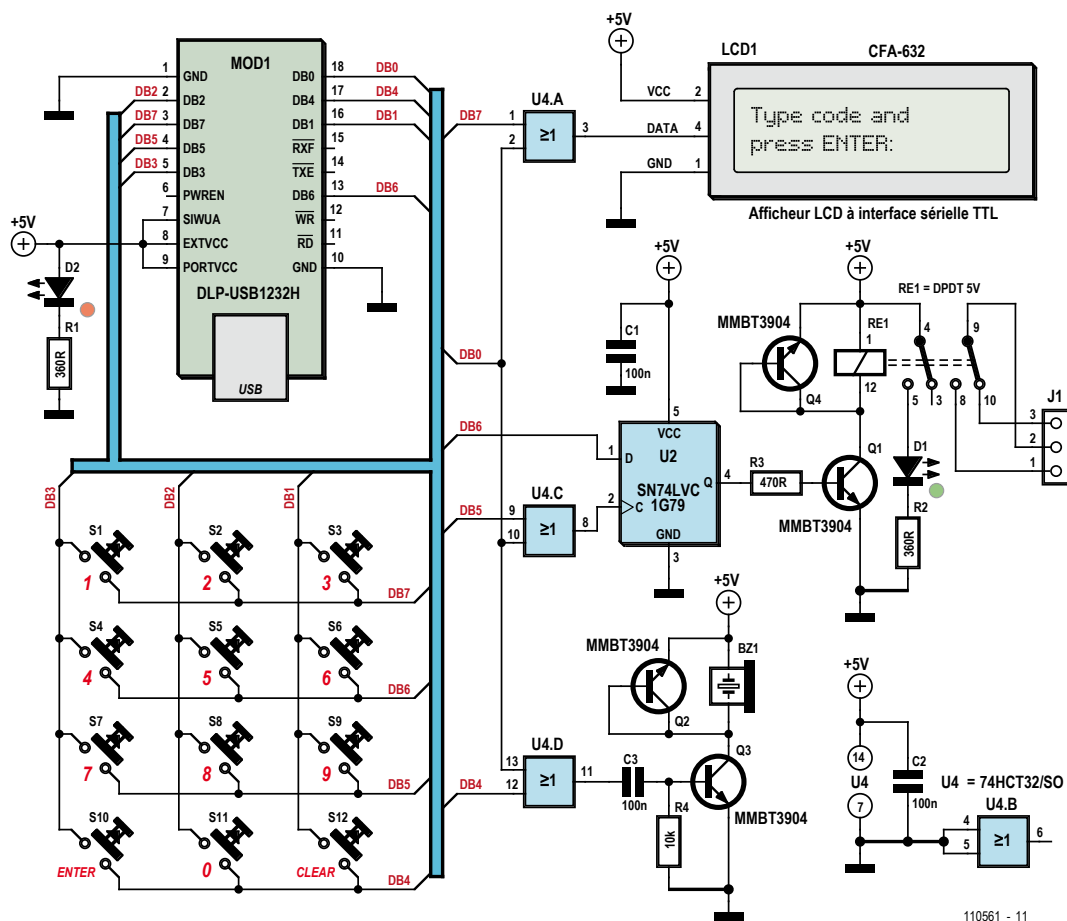


Figure 1 - Attaqué de façon adéquate en mode bit à bit, le module DLP-USB1232H devient l'élément actif d'une serrure codée.

bits, puis vous écrirez l'octet.

Tout va bien jusqu'ici... mais qu'en est-il si vous voulez envoyer un flux sériel de 200 octets à travers l'une des huit lignes d'E/S sans affecter les sept autres ? Bonne question : vous établissez un tampon de 1600 octets. Chaque octet de ce tampon ne diffère du précédent que par un bit, conformément au bit de la sortie sérielle qui doit être envoyé à la prochaine période du temporisateur. Ensuite, vous envoyez le tampon entier d'un coup par la fonction `FT_Write()`. Laborieux ? Certes ! Mais les ordinateurs adorent ça, et vous n'avez à écrire le programme qu'une fois pour émettre de longues chaînes sérielles.

Matériel

Pour la suite, reportez-vous au schéma de principe de la **figure 1**. Pour scruter les douze touches du pavé numérique au moyen des huit lignes d'E/S disponibles, vous portez simplement au niveau bas les lignes des « rangées » DB4, DB5, DB6 et DB7 (une à la fois) et vous lisez l'état des trois lignes des « colonnes » connectées à DB1, DB2 et DB3. Si une touche est enfoncée, la colonne correspondante se signale par un niveau bas sur sa ligne d'entrée.

La ligne DB0 décide si l'hôte lit le clavier ou commande l'afficheur, le relais ou le ronfleur.

Quand DB0 est haut, les portes OU bloquent le passage des données vers ces « récepteurs ». Quand DB0 est bas, le clavier est ignoré et les données peuvent être écrites

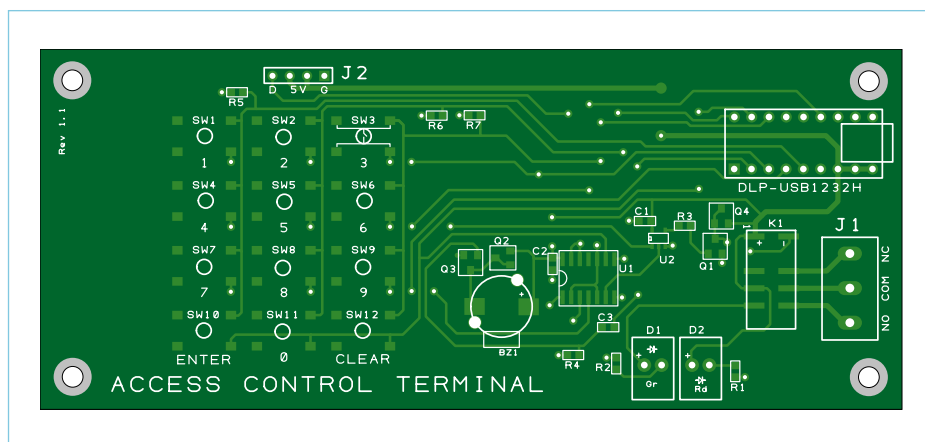


Figure 2 - Implantation des composants du circuit imprimé de la serrure codée (ici à 80% de sa taille réelle). Les fichiers Gerber peuvent être téléchargés du site Elektor [2].

dans les récepteurs par les lignes DB4 à DB7. Vous vous doutez que tenir appuyée une touche du clavier interdira à l'hôte d'écrire sur l'un ou plusieurs des récepteurs. Contournez l'obstacle en attendant dans l'application hôte que toutes les touches aient été relâchées avant de continuer. Vous trouverez toujours un moyen de prendre un système en défaut, si vous décidez d'en chercher un, mais après tout ce système est destiné à garder un intrus à l'extérieur d'une zone protégée. S'il maintient une touche enfoncée, vous êtes sûr qu'il n'entrera pas. Les fichiers Gerber pour la gravure du circuit imprimé peuvent être téléchargés [2]. Le plan d'implantation apparaît sur la **figure 2**.

Bit-Bang++...+?

J'ai d'abord été tenté de présenter un projet dont le matériel aurait été composé de huit afficheurs LCD TTL, tous connectés à un PC hôte, en n'utilisant que les huit lignes d'E/S et le mode *bit à bit*. Cela aurait très bien

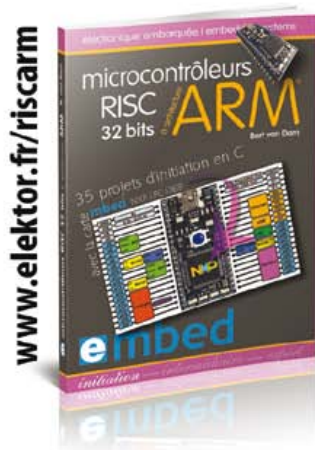
fonctionné, mais sans panache. Cela aurait été plus coûteux aussi. Le mode *bit à bit* peut aussi être utilisé pour des tâches plus ordinaires comme la commande de relais ou de simples entrées-sorties tout ou rien. Les aventuriers essayeront de commander plusieurs périphériques SPI tels des convertisseurs A/N et N/A. Je crois que l'essentiel à retenir de cet article est que vous n'avez pas obligatoirement besoin d'un microcontrôleur — ni du développement logiciel qui l'accompagne — pour interagir *via* l'interface USB avec le monde qui vous entoure. Le mode *Bit-Bang* peut se révéler une solution économique parfaite pour connecter un PC à son environnement avec un système qui ne réclame qu'un logiciel du côté hôte.

(110561 - trad. J.-P. Brodier)

Liens :

- [1] www.dlpdesign.com/pub.shtml
- [2] www.elektor.fr/110561

Publicité



Consonant : préamplificateur de haute qualité (1978)

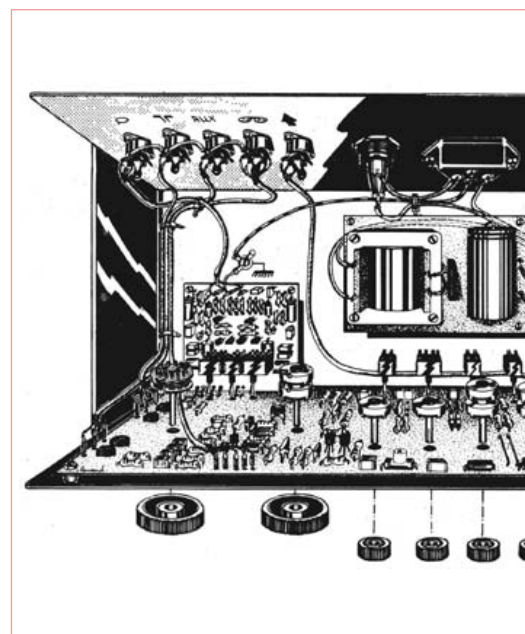
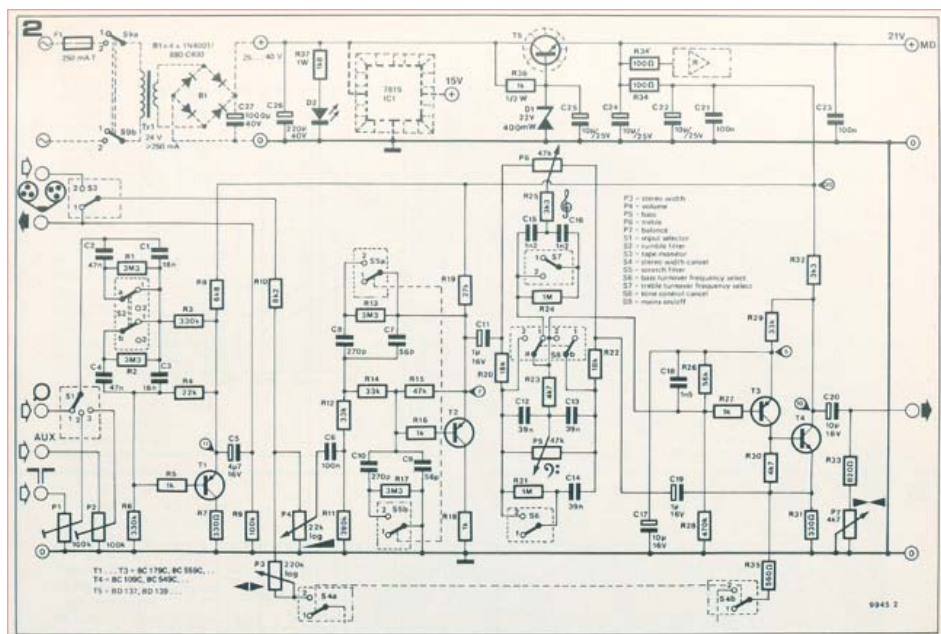
Jan Buiting (Elektor UK/US Editorial)

Mon évocation nostalgique de ce mois commence par un épisode assez récent de la vie d'Elektor, à savoir la dernière journée *Elektor-Live!* L'histoire démarre lors de notre grand raout annuel tenu en novembre 2011, mais elle remonte à l'année 1978 qui est celle de la naissance d'Elektor en France.

L'invité de prestige de l'édition 2011 d'*Elektor Live!* était un pont

un circuit imprimé grand comme ça [à en juger par ses gestes, il devait bien faire 30 cm] pour élargir l'image stéréophonique. Je ne me souviens plus du nom de l'appareil, mais quel circuit !». Wisse est un bleu, sa mémoire elektorienne ne remonte qu'à une demi-douzaine d'années ; il a donc bafouillé quelque chose et s'en est tiré par une pirouette comme le font les intervieweurs.

Plus tard, pour le montage de cet entretien vidéo, nous avons évidemment cherché à mettre un nom sur ce projet oublié. Il retrouverait un lustre inattendu puisqu'il existait encore dans la mémoire



de Philips, M. Henk van Houten. C'est un peu comme si, en France, Elektor invitait le patron de Thales ou d'Alcatel. Le PDG de *Philips Research* est arrivé à l'heure, cela mérite une mention, pour honorer de sa présence l'Evolution, fleuron de l'architecture d'avant-garde néerlandaise des années 60, quand les mots *Philips* et *Hollande* étaient encore synonymes. L'immeuble qui tient du château d'eau et de la soucoupe volante mérite votre attention.

À la grande surprise des deux hôtes elektoiriens chargés de recevoir M. van Houten, celui-ci les entreprit sur des sujets terre-à-terre, comme le soudage des composants, l'isolation des 2N3055 sur leurs radiateurs et le perçage, alors qu'eux avaient préparé pour lui des sujets de conversation de haut vol, comme la macro-économie de l'électronique, avec le jargon financier de rigueur. Dans son entretien [1] avec notre directeur éditorial Wisse Hettinga, l'étonnant Henk évoque ses années de pratique active de l'électronique, mais ne retrouve plus le nom d'un « vieux circuit audio d'Elektor avec

d'un personnage si important qu'on n'imaginait même pas qu'il ait jamais lu un article d'Elektor. Il a fallu la mémoire encyclopédique de notre collègue Harry pour faire le rapprochement entre cette fonction d'élargisseur de l'image stéréophonique et le préamplificateur de 1978 au nom embarrassant : *Consonant*.

L'anecdote en serait restée là si je n'avais pas rapporté sur mon bureau un prototype du Consonant, retrouvé dans les combles, avec son étonnant circuit imprimé de grande largeur, monté derrière la face avant de l'appareil. En découvrant cette antiquité, mes jeunes collègues férus d'audio *vintage* me pressèrent de questions du genre : « on dirait un Sanyo ou un Kenwood ou Sony des années 80, non ? ». Il n'en fallait pas plus pour en faire le sujet de ma rubrique. L'article sur le *Consonant*, beau comme un sou neuf, a été publié sur pas moins de dix pages du numéro d'été anglais de cette année-là, au milieu de l'habituelle mitraille estivale, et en France dans l'édition spéciale numérotée 5-6.

Le nom de *Consonant* s'inscrit dans une tradition de la nomenclature des années 1970 qui, pour dénommer les appareils, puisait volontiers dans le vocabulaire musical : *Crescendo*, *Prelude*, *Stentor*, *Fidelio* (non, celui-là n'est pas un enfant d'Elektor) etc. Ce nom reste une jolie trouvaille qui recèle d'ailleurs quelques subtilités inattendues. En français comme en anglais, la consonance est le contraire de la dissonance. L'étymologie est latine, avec le préfixe *con-*, qui signifie « avec », et le verbe *sonare* qui signifie résonner. Plus prosaïquement, mais toujours dans le domaine du sonore, la consonne est aussi la

compagne de la voyelle, celle *qui sonne avec*. Or en haute-fidélité, on cherche précisément à éviter tout ce qui pourrait *sonner avec la musique* : surtout pas de bruit, de souffle, de ronflement, ni de parasites RTL ou *DeutscheWelle*... qui dépareraient le signal audio que l'on veut pur.

Hormis cette note de lyrisme dans le nom de l'appareil, il n'y a aucune trace de vocabulaire musical dans l'article de 1978 :

Les principales considérations qui ont présidé à l'étude du Consonant ont été :

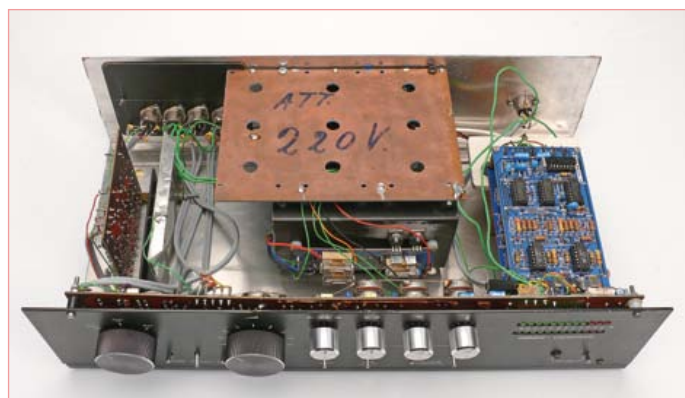
- La comparaison des performances et possibilités offertes avec celles fournies par les meilleures réalisations du commerce,
- La simplicité de la construction et l'utilisation de composants facilement disponibles,
- La disposition logique et claire des organes de réglage, rendant l'utilisation facile.

Jamais l'auteur n'entonne les louanges d'Elektor qui engendra ce brillant rejeton — la description des caractéristiques, pourtant excellentes, reste factuelle, technique, modeste et même sèche. Aucune trace de cette emphase devenue le langage ordinaire de certaines confréries d'audiophiles.

La mise en page du **schéma** ne le mettait pas spécialement en valeur, le majestueux circuit imprimé était mieux servi, de même que les



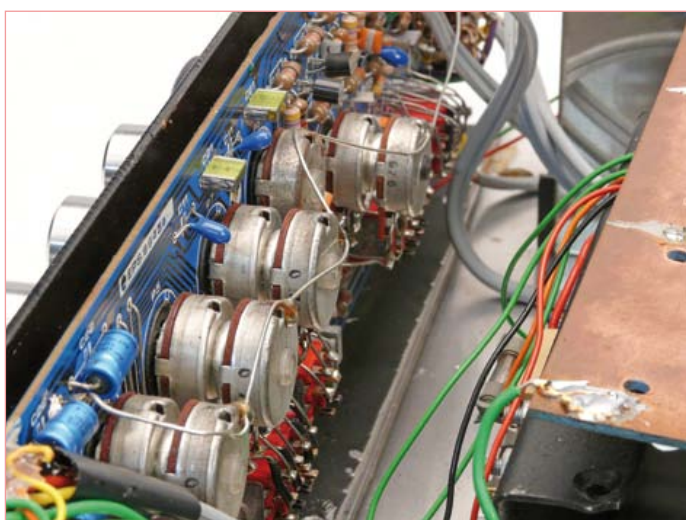
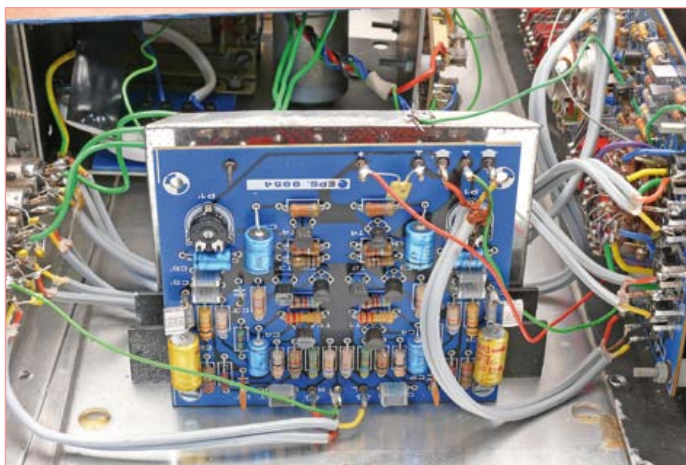
relevés de mesure, effectués avec du matériel *Bruel & Kjaer*, s'il vous plaît (on ne se mouchait pas du pied au labo d'Elektor en ce temps-là). Nos lecteurs apprécieront le schéma du *Consonant* repris ici pour le plaisir de la nostalgie et de la gloriole, avec les **spécifications**. Revenons à nos moutons, ou plus précisément à ce fameux élargisseur d'image stéréo par lequel tout a recommencé. Une image élargie passait pour souhaitable à cette époque, notamment pour don-



Caractéristiques du préamplificateur Consonant Elektor

Réponse en fréquence :	20 Hz à 50 kHz (+0 dB, -3 dB)
Tension de sortie max. :	3,5 V _{eff} (10 V _{cc})
Tension de sortie nom. :	440 mV _{eff}
Rapport signal/bruit :	>72 dB pour 440 mV _{eff} en sortie
Marge de surcharge :	>15 dB au-dessus de 440 mV _{eff} en sortie
Distorsion harmonique tot. :	approx. 0.04% (for 440 mV out)

Séparation des canaux :	>50 dB (à 1 kHz)
Dynamique :	>90 dB
Bruit à la sortie :	environ 0,1 mV _{eff}
Filtres	
Rumble :	60 Hz (-3 dB), 12 dB/octave
Scratch :	10 kHz (-3 dB), 12 dB/octave



ner de l'ampleur à la remastérisation alors fréquente de nombreux enregistrements monophoniques. Le nouvel effet connu alors une vogue qui ne faiblira plus. Aujourd'hui encore des groupes de rock comme *Bloc Party* ou *Editors* en usent et en abusent pour faire gicler leurs riffs de guitare.

Dans le *Consonant*, pour le mettre en service, il faut fermer S4. Les voies D et G sont alors reliées par R35 et P3. En établissant cette liaison entre les émetteurs de T4 et T4', la résistance fixe R35 fait de ces deux étages un amplificateur différentiel. Le signal sur le collecteur de T4 représente donc $(L - kR)$, où k est une constante déterminée par les paramètres du circuit. Le signe négatif exprime la portion de signal en opposition de phase de la voie D. De la même manière, l'expression mathématique de la voie D est $(R - kL)$. L'effet audible de ces signaux en opposition de phase donne cette impres-

sion d'éloignement entre les deux voies stéréophoniques et l'élargissement de l'image perçue par l'auditeur, conformément au canon esthétique de ces années-là. L'effet avait déjà été décrit dans le numéro 1 de l'édition française d'Elektor, page 36.

À la fin des années 70, il était encore de bon ton de construire soi-même à bon prix sa chaîne haute-fidélité — et d'en être fier. Même en anglais les mots *nerd* et *geek* n'existaient pas, et *hobbyist* n'était pas une insulte. Le fait d'implanter tous les composants, y compris les organes de commande, sur le même circuit imprimé passait pour une innovation remarquable : plus de fil de câblage, moins de parasites, moins de bruit. Comme chaque médaille a son revers, il a fallu isoler soigneusement le corps des potentiomètres de la façade métallique afin d'éviter les boucles de masse. Ce détail n'avait pas été mentionné dans l'article initial, mais il apparaît à la fin de l'article de l'édition française, postérieure de quelques mois.

Pour éviter la coupure inévitable entre une page de gauche et une page de droite au milieu du magazine, on a pris soin de publier le majestueux circuit imprimé (370 x 90 mm) sur les pages centrales, car elles appartiennent au même cahier et sont les seules à ne pas souffrir de décalage. Idem pour le plan d'implantation des composants et la face avant. Dans ces années-là, de très nombreux lecteurs gravaient encore eux-mêmes leurs circuits imprimés, et nombreux sont ceux qui n'ont pas hésité à détacher les pages centrales de leur précieux magazine pour transférer le dessin des pistes sur une plaque cuivrée.

Le *Consonant* photographié ci-contre n'a pas remarqué dès la mise sous tension ; il m'a fallu refaire deux soudures, sur un condensateur au tantale et sur un régulateur de tension. Et puis il s'est réveillé de son long sommeil. Dans le même coffret, on trouve un *Preconsonant*, préamplificateur pour disque microsillon, publié dans le premier numéro d'Elektor.

Que dire d'autre du *Consonant* aujourd'hui ? Sa contribution dans la chaîne stéréo n'a pas desservi le signal dans mon salon. Je n'ai ressenti aucun besoin de correction, pas de Baxandall, tous les réglages sont restés en position neutre. Quant à l'élargisseur de stéréophonie, il m'a fait l'effet d'un... effet. C'est-à-dire de quelque chose d'artificiel, qui jure avec mes critères de qualité d'aujourd'hui. J'ai ressorti le vieux 33 tours *Tubular Bells*, de Mike Oldfield. Je pense que si notre invité Henk l'écoutait aujourd'hui dans son salon (qui doit être magnifique, s'il se rapporte à sa prestigieuse ascension sociale), il n'éprouverait sans doute plus le besoin d'élargir l'image stéréophonique comme il le faisait sans sa chambrette d'étudiant à la fin des années 70. Quant à moi, dans ces années-là, je n'avais pas le sou, et certainement pas de quoi me payer le *Consonant*.

(110718)

Liens

[1] vidéo : cherchez la chaîne ElektorIM (sic) sur www.youtube.com

[2] www.elektor.com/110718

Pour célébrer dignement le *Consonant*, Yves Masquelier a soigneusement numérisé pour nous l'article paru en 1978, et disponible désormais en téléchargement gratuit sur notre site.

Hexadoku

Casse-tête pour électroniciens

Combien de grilles Hexadoku différentes de 16 x 16 chiffres est-il possible de composer?

Nous l'ignorons et ne craignons pas pour l'instant d'épuiser toutes ces possibilités. En voici une nouvelle, que vous ne devriez n'avoir encore vue nulle part. Remplissez-la et envoyez-nous votre solution. Vous gagnerez peut-être l'un des quatre chèques-cadeaux Elektor mis en jeu.

Les instructions de ce jeu sont simples. Une grille Hexadoku est composée de chiffres du système hexadécimal, de 0 à F. Du tout cuit pour les électroniciens et les programmeurs ! 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. Si vous trouvez la solution de ce casse-tête, vous pouvez gagner un chèque-cadeau. Inutile de nous envoyer toute la grille, il suffit de nous envoyer la **série de chiffres** sur fond grisé.

Participez et gagnez !

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

À vos crayons !

Où envoyer ?

Envoyez votre réponse (les chiffres sur fond grisé) avec vos coordonnées par courriel, télécopie ou courrier avant le **1^{er} avril 2012** à :

Elektor c/o Regus Roissy CDG – Le Dôme – 1, rue de la Haye

BP 12910 – 95731 Roissy CDG

Courriel : hexadoku@elektor.fr

Les gagnants

La solution de l'Hexadoku du n° 403 (janvier) est : **43ADE**

Le gagnant du **chèque-cadeau Elektor** d'une valeur de **100 €** est : Christian Klems de Nijkerk (Pays-Bas).

Les **3 chèques-cadeaux Elektor** d'une valeur de **50 €** chacun vont à : Antje Völksch. de Bayreuth (Allemagne), Raúl Elguezabal Martínez, de Barakaldo – Vizcaya (Espagne) et Marc Herzog, de Luxembourg (Luxembourg).

Bravo à tous et félicitations aux gagnants !

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

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

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

Morceaux choisis

La bibliothèque des passionnés d'électronique

Mariline Thiebaut-Brodier **Domotique** sécurité - confort - économies

nouvelle édition
revue et augmentée



sécurité - confort - économies

Domotique

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

Ce livre montre comment la domotique gère le fonctionnement des appareils et dispositifs électriques de la maison (tableau d'abonné, électroménager, éclairage, chauffage) et le réseau multimédia domestique (tableau de communication). Cette nouvelle édition traite également de deux aspects essentiels, et en plein développement, de la domotique de demain :

- l'efficacité énergétique : nouvelles étiquettes Énergie, éco-conception, Réglementation Thermique 2012, compteur communicant, etc.
- la gérontechnologie, spécialité des aides techniques aux personnes âgées ou handicapées.

256 pages • 14 x 21 cm • ISBN 978-2-86661-182-8 • 32,50 €

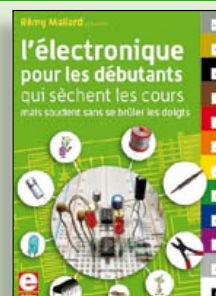


PHOTOVOLTAÏQUE

Alimentation électrique autonome de sites isolés

Rêver de l'autonomie énergétique obtenue par une production photovoltaïque domestique, c'est bien. En calculer la viabilité, c'est mieux. Mais suffit-il de comparer kilowattheure nucléaire (environ 0,08 €) et kilowattheure photovoltaïque (environ 0,22 €) ? L'auteur G. Guihéneuf pense qu'il ne faut pas se contenter d'aligner chiffres et idées. Il décrit trois projets exemplaires et simples d'électrification photovoltaïque de sites ordinaires (cabanon, garage, mobile-home) avec des onduleurs, des régulateurs de charge et un suiveur solaire.

254 pages • ISBN 978-2-86661-181-1 • 32,50 €



Rémy Mallard présente

L'électronique pour les débutants

Par où commencer pour débiter en électronique ? Vais-je m'égarer en explorant l'internet, qui regorge de schémas, mais sont-ils fiables ? Me faut-il un livre avec des montages simples ou plutôt un livre sur les composants ? Après trente ans de pratique, l'auteur de ce livre, resté l'éternel débutant qui réalisait lui-même son premier montage dès l'âge de dix ans, partage ici sa soif toujours vive d'apprendre. Fin pédagogue, il guide les débutants et répond aux questions que trop de livres laissent en suspens : « Quel type de fer à souder acheter ? »... « Un multimètre à 5 € peut-il suffire ? »... « Un oscilloscope est-il indispensable ? ».

317 pages • ISBN 978-2-86661-180-4 • 39,50 €



23 projets ludiques et instructifs
à construire soi-même

Intelligence artificielle

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

238 pages • ISBN 978-2-86661-179-8 • 42,50 €



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

Microcontrôleurs RISC 32 bits à architecture ARM

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

232 pages • ISBN 978-2-86661-178-1 • 39,50 €



Initiation et maîtrise par l'expérimentation

50 applications des microcontrôleurs PIC

Voici 50 projets instructifs et utiles pour vous initier au langage de programmation JAL et maîtriser les microcontrôleurs PIC16 et 18, avec des techniques universelles comme la commande de relais, ou le traitement des signaux émis par divers capteurs (y compris par exemple un codeur rotatif), la communication avec les bus I²C, SPI, RS232, USB, les afficheurs à 7 segments et même le bus CAN.

C'est un ouvrage récréatif et pédagogique : assemblez et utilisez les projets proposés. Les explications claires, les schémas et les photographies vous feront découvrir une activité enrichissante.

394 pages • ISBN 978-2-86661-177-4 • 44,50 €

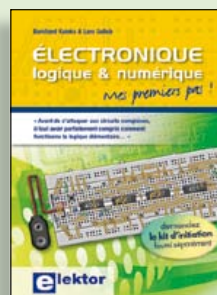


Les microcontrôleurs au service de
l'écologie

Eau de pluie

Cet ouvrage aborde le thème de l'eau sous ses aspects quantitatif (cycle de l'eau, précipitations, prix) et qualitatif (traitement de l'eau potable, assainissement des eaux usées). Il détaille ensuite les différents éléments d'une installation de récupération des eaux de pluie : système aérien, cuve enterrée, pompe, filtres... Récupération et utilisation des eaux de pluie ne sont autorisées que sous certaines conditions et pour certains usages.

160 pages • ISBN 978-2-86661-176-7 • 27,50 €



Mes premiers pas !

Électronique logique & numérique

Regardez les appareils actuels : on y trouve essentiellement des puces, des microcontrôleurs, des dispositifs logiques programmables, etc. Les circuits intégrés simples, comme le 4000 ou le 4093, y sont plus rares. Se familiariser avec de tels circuits reste pourtant la porte d'entrée idéale du monde de la logique. Car c'est d'opérateurs, de bascules bistables et de compteurs que sont faits les montages numériques même les plus complexes. L'apprentissage des notions fondamentales est plus facile lorsqu'il s'accompagne de travaux pratiques ; c'est pourquoi ce livre est associé à un kit d'initiation réunissant une plaquette d'expérimentation universelle et tous les composants requis.

156 pages • ISBN 978-2-86661-175-0 • 29,50 €

Informations
complémentaires
et gamme complète
sur notre site
www.elektor.fr

Elektor / Publitronic SARL
1, rue de la Haye
BP 12910
95731 Roissy CDG Cedex
Tél. : +33 (0)1.49.19.26.19
Fax : +33 (0)1.49.19.22.37
@ : ventes@elektor.fr





Tous les articles de 2011 sur DVD-ROM

DVD Elektor 2011

Ce DVD-ROM réunit tous les articles d'ELEKTOR, le mensuel d'électronique et de micro-informatique appliquées, parus au cours de l'année 2011. Il contient non seulement le texte des articles ainsi que les schémas, mais aussi tous les dessins des circuits imprimés, sous forme de fichiers à haute résolution. Ceci permet à l'utilisateur de modifier à sa guise les dessins existants à l'aide d'un programme adéquat. Dès lors, rien ne s'oppose plus à l'exportation des documents vers un autre format à la convenance de l'utilisateur.

ISBN 978-90-5381-276-1 • 27,50 €



Les articles de la série populaire
Elektor-CC2-AVR réunis sur CD-ROM

CD ATM18 Collection

Voici la compilation sur CD-ROM de la série populaire Elektor-CC2-AVR, soit plus de 25 projets autour de la carte ATM18, les logiciels associés, les typons (PDF) et des informations complémentaires. Le tout est complété par un cours complet de programmation BASCOM-AVR en 6 leçons.

ISBN 978-0-905705-92-7 • 29,50 €



Détecteur de rayonnement

(Elektor novembre 2011)

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

Kit (composants avec la platine, afficheur et microcontrôleur programmé)

Réf. : 110538-71 • 39,95 €



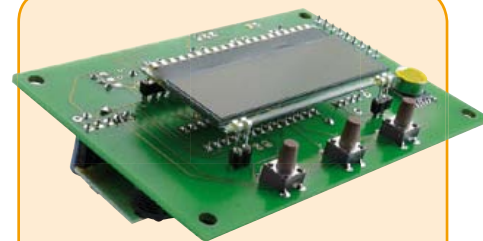
Passerelle USB/série BOB-FT232R

(Elektor septembre 2011)

Ce convertisseur USB/série vous surprendra d'abord par sa taille : pas plus grande que l'embout moule d'un câble USB ! Il vous épatera aussi parce qu'il est pratique, rapide à mettre en oeuvre, réutilisable, multi plate-forme (Windows, Linux, etc.) et pas trop cher pour autant.

Platine montée et testée

Réf. : 110553-91 • 15,00 €



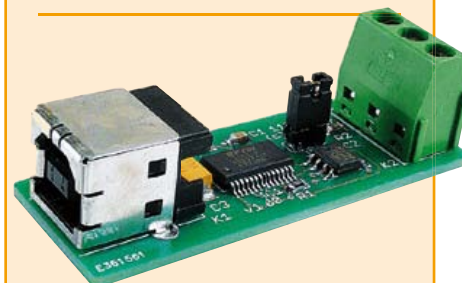
Enregistreur météo USB de longue durée

(Elektor septembre 2011)

Cet enregistreur de données autonome enregistre avec une pile quelque six à huit semaines de données fournies par des capteurs I²C de pression atmosphérique, température et humidité et les affiche sur un LCD. Les résultats, collectés par l'intermédiaire d'un port USB, sont représentés graphiquement sur un PC par GNUplot. L'utilisation de modules de capteurs numériques réduit l'investissement en matériel et évite les réglages.

Kit : platine, contrôleur ATmega88-20PU programmé, capteurs d'humidité et de pression atmosphérique

Réf. : 100888-73 • 34,95 €



Convertisseur USB/RS485

(Elektor juin 2011)

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

Platine montée et testée

Réf. : 110258-91 • 24,95 €

F405 mars 2012

€

+++ Retrouvez sur www.elektor.fr toutes les références disponibles +++

F404 février 2012

AndroPOD (1)

110258-91	Convertisseur USB/RS-485 (monté et testé)	24,95
110405-91	Module avec RS-485 (monté et testé)	www.elektor.fr
110553-91	USB-FT232R (monté et testé)	15,00
120103-92	Câble noir USB 2.0 A mâle vers micro-B, 5 broches, 1,8 m	3,95
120103-94	Bloc d'alimentation moulé 5 V/1 A à connecteur micro USB	8,95

Pico C-Plus et Pico C-Super

110687-41	Microcontrôleur programmé Pico C-Plus (ATtiny2313-20PU)	4,95
110687-42	Microcontrôleur programmé Pico C-Super (ATtiny2313-20PU)	4,95

Retour aux sources (2)

ELEX-1	Platine de prototypage ELEX	5,50
ELEX-2	Double platine de prototypage ELEX	9,95

F403 janvier 2012

Clé USB sur microcontrôleur

110409-1	Platine	10,95
110409-41	Microcontrôleur programmé PIC24FJ64GB002-I/sp dil-28s	14,95

Interface pour sonde lambda à bande large

110363-41	Microcontrôleur programmé ATMEGA8-16AU	9,95
-----------	-------	--	-------	------

Cours DSP audio

110002-71	Platine dotée des CMS + les connecteurs et les barreaux à LED	49,95
-----------	-------	--	-------	-------

Le bus arrive (11)

110258-1	Platine (noeud d'expérimentation)	5,95
110258-1C3	...	3 x Platine (noeud d'expérimentation)	12,95
110258-91	Module convertisseur USB/RS-485	24,95

Le réseau EDF à la loupe

110461-41	AT89C2051-24PU programmé (50 Hz, Europe)	9,95
110461-42	AT89C2051-24PU programmé (60 Hz, Amérique du Nord)	9,95

F402 décembre 2011

Le bus arrive (10)

110258-1	Platine (noeud d'expérimentation)	5,95
110258-1C3	...	3 x Platine (noeud d'expérimentation)	12,95
110258-91	Module convertisseur USB/RS-485	24,95

F401 novembre 2011

Détecteur simple de chauve-souris

110550-1	Platine	9,95
----------	-------	---------	-------	------

Détecteur de rayonnement

110538-41	Microcontrôleur ATmega88PA-PU programmé	10,50
110538-71	Kit (composants avec la platine, afficheur et microcontrôleur programmé)	39,95

Interface JTAG par OnCE

110534-91	Programmeur monté et testé	39,95
-----------	-------	----------------------------	-------	-------

Le bus arrive (9)

110258-1	Platine (noeud d'expérimentation)	5,95
110258-1C3	...	3 x Platine (noeud d'expérimentation)	12,95
110258-91	Module convertisseur USB/RS-485	24,95

F400 octobre 2011

Platine polyvalente pour montages à microcontrôleur(s) AVR

100892-1	Platine Platino	12,95
----------	-------	-----------------	-------	-------

OBD 2 sans fil

100872-71	Platine OBD2-ZigBee dotée de composants CMS plus boîtier et clé USB ZigBee	124,95
100872-72	Platine OBD2-Bluetooth dotée de composants CMS plus boîtier	124,95

Capteurs I²C

100888-71	Capteur d'humidité HH10D	7,95
100888-72	Capteur de pression atmosphérique HP03SA	6,45

vos favoris

livres	1	↑	311 circuits ISBN 978-2-86661-184-2	36,00 €
	2	→	Domotique ISBN 978-2-86661-182-8	32,50 €
	3	↘	L'électronique pour les débutants ISBN 978-2-86661-180-4	39,50 €
	4	→	Intelligence artificielle ISBN 978-2-86661-179-8	42,50 €
	5	↘	Microcontrôleurs RISC 32 bits à architecture ARM ISBN 978-2-86661-178-1	39,50 €
cd & dvd-rom	1	→	CD ECD 6 ISBN 978-90-5381-258-7	29,50 €
	2	↗	DVD Elektor 2010 ISBN 978-90-5381-267-9	27,50 €
	3	↗	DVD The Audio Collection 3 ISBN 978-90-5381-263-1	21,50 €
	4	↘	DVD Wireless Toolbox d'Elektor ISBN 978-90-5381-268-6	32,50 €
	5	↘	CD ATM18 Collection ISBN 978-0-905705-92-7	29,50 €
kits & modules	1	↗	Passerelle USB/série BOB-FT232R Réf. : 110553-91	15,00 €
	2	↘	Déecteur de rayonnement Réf. : 110538-71	39,95 €
	3	↗	Capacimètre Pico C Réf. : 100823-71	82,50 €
	4	↘	Enregistreur météo USB de longue durée Réf. : 100888-73	34,95 €
	5	↘	Convertisseur USB/RS485 Réf. : 110258-91	24,95 €

Commandez tranquillement sur

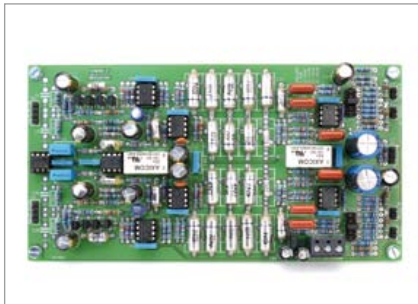
www.elektor.fr/e-choppeou à l'aide du bon de commande encarté
à la fin de la revue. Les commandes en ligne
de livres ou de CD & DVD-ROM bénéficient
d'une remise spéciale de 5%.


Elektor / Publitrone SARL
1, rue de la Haye • BP 12910
95731 Roissy CDG Cedex
Tél. : +33 (0)1.49.19.26.19
Fax : +33 (0)1.49.19.22.37
@ : ventes@elektor.fr



Thermomètre à affichage électromécanique

Même au supermarché on trouve pour quelques euros des thermomètres électroniques à afficheurs à cristaux liquides dont la précision est plus qu'honorable. Pour ce qui est de la lisibilité, c'est une autre histoire. Au-delà de quelques dizaines de centimètres de distance et quand la luminosité est forte ou faible, on n'y voit plus rien. La solution imaginée pour ce thermomètre est spectaculaire : l'auteur propose de récupérer l'affichage électromécanique d'un flipper. C'est un ATtiny2313 qui se chargera de convertir la température captée par un TMP100 en un signal approprié (des impulsions !) pour commander les électroaimants qui actionnent les roues de l'afficheur. Un projet original, ludique, utile et instructif !



Préamplificateur de haut de gamme

Après le remarquable amplificateur de puissance à 64 amplificateurs opérationnels 5532 publié en octobre et novembre 2010 (www.elektor.fr/100124), l'éminent concepteur audio Douglas Self a cogité sur un préampli du même calibre, construit lui aussi autour d'amplis op 5532 montés en parallèle. Des platines ont été dessinées et mises au point au labo d'Elektor pour cette réalisation qui se distingue par les nombreuses possibilités de réglage offertes. C'est par une introduction pour en définir le cadre général que nous ouvrirons le mois prochain une série d'articles consacrée à ce projet. Pour vous mettre la puce à l'oreille, si l'on peut dire, nous reproduisons ci-contre la platine du préampli MD/MC de grande qualité.



Commutation RS-485

L'ElektorBus se prête parfaitement à des applications pratiques, par exemple dans le domaine de la domotique, pour interconnecter des circuits de mesure, de commande et de régulation. C'est dans cet esprit que nous publierons un module de commande de domotique pour l'ElektorBus, équipé de deux relais, d'un microcontrôleur ATmega88 et d'un pilote de ligne RS-485. Ce module permettra de commuter des charges sous 230 V. Deux entrées du contrôleur sont reliées à un bornier auquel il est facile de connecter des interrupteurs ou des poussoirs. Bien entendu, le logiciel de commande de ce module sera compatible avec l'ElektorBus.

Informations sous réserve de modifications. Le numéro d'avril paraîtra le 21 mars.

Pour vous abonner :

Passez par notre site www.elektor.fr/abo, c'est plus rapide et moins cher.

www.elektor.fr www.elektor.fr www.elektor.fr www.elektor.fr www.elektor.fr www.elektor.fr

Elektor en ligne

Sur le site d'Elektor, vous trouvez tous les articles publiés depuis 2000, sous forme de fichiers PDF téléchargeables individuellement, certains gratuitement, d'autres moyennant un modeste paiement forfaitaire. Un résumé de l'article donne une idée du contenu avant de le télécharger, de même que la liste des composants (le cas échéant). Le site propose également les autres ressources liées à chaque article : code source, liens, circuits imprimés, et les corrections ainsi que les mises à jour s'il y en a. L'e-choppe d'Elektor propose de nombreux produits : CD-ROM, DVD, kits, modules assemblés, appareils & instruments, E-blocks, livres. Sans oublier l'indispensable et puissante fonction de recherche.

Également sur le site d'Elektor :

- des nouvelles sur le monde de l'électronique
- un forum des lecteurs
- téléchargement de logiciel et de circuits imprimés
- des offres temporaires très avantageuses
- Foire Aux Questions



DÉFI QUOTIDIEN N°1 : LA RAPIDITÉ

Décrocher en
moins de

10



04 74 68 99 99 Jugez-nous sur pièces !

3 sonneries pas plus avant de vous dire bonjour...
C'est le défi quotidien de chacun de nos conseillers.
Ici, pas de plateforme téléphonique robotisée, vous

êtes accueilli par une interlocutrice ou un interlocuteur
motivé et formé pour satisfaire votre demande avec
précision sans vous faire perdre de temps !

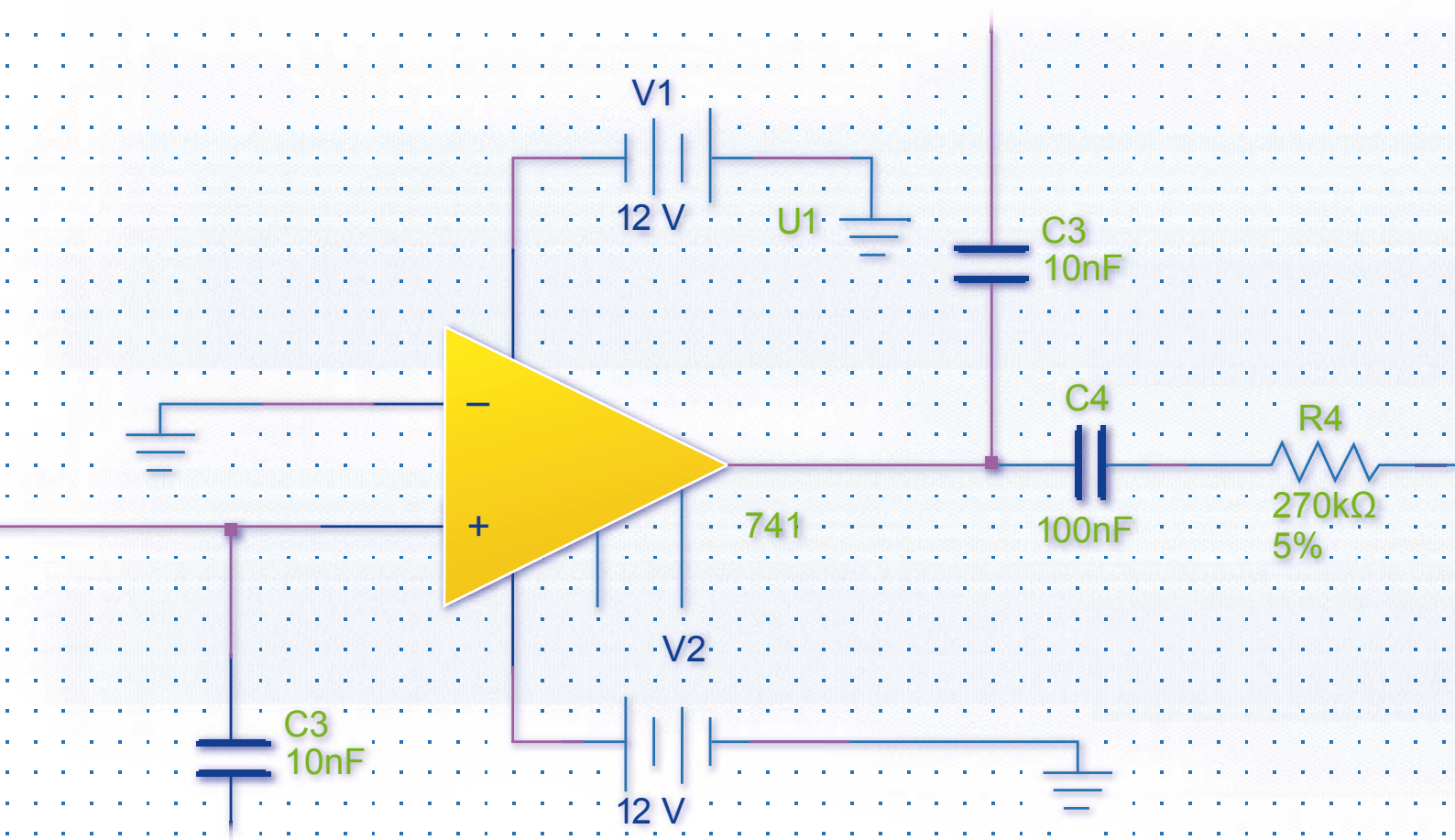
element14
BE IN YOUR ELEMENT

www.element14.com



Votre satisfaction est notre priorité !

Une meilleure conception pour l'enseignement de l'ingénierie



Faites davantage qu'enseigner l'ingénierie. Pratiquez-la.

Enseigner la conception de circuits sans moyen efficace pour passer du concept à l'expérimentation, c'est comme expliquer à une personne comment faire un créneau sans lui laisser le volant. National Instruments propose les matériels et logiciels dont les étudiants ont besoin pour réaliser des expériences, afin d'aller au-delà de la théorie et de la simulation, et de prendre conscience de ce que la pratique de l'ingénierie signifie.



OUTILS PÉDAGOGIQUES

NI LabVIEW

NI myDAQ

NI ELVIS

NI Multisim

>> Découvrez comment NI soutient la prochaine génération d'innovateurs sur ni.com/academic/f

01 57 66 24 24