



elektor

DÉCOUVRIR

CRÉER

PARTAGER

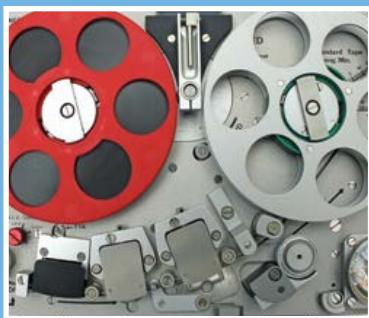
tour du monde des cartes à μ C et kits d'embarquement pour tous



**analyseur de signal
par Ethernet (1)**



**Platino, le retour
par pitié, arrêtez de
réinventer la roue**



**Nagra SN :
l'oreille discrète de
l'espion**

Dans cette édition :
2 projets de labo
2 projets de lecteur
des nouvelles du labo,
des trucs et
astuces ...

initiation aux balises NFC de la famille ST25TA
• duo de LED sur une seule broche de port •
Arduino/Genuino 101 • Eagle : format de fichier
XML • salon CARTES 2015 • compteur de visites pour
vos pages web • hors-circuits : filtrage numérique sans stress, filtres CIC
• corrections, mises à jour et courrier des lecteurs • convertisseur
buck têtu dompté • soudage d'une main sûre : les CMS • ressusciter de
vieux gadgets avec l'électronique la plus récente • horloge aide-mémoire



PARROT RECRUTE DES INGÉNIEURS

EN DÉVELOPPEMENT LOGICIEL, CONCEPTION ÉLECTRONIQUE, TRAITEMENT DU SIGNAL, TEST & VALIDATION...



ARE YOU UP TO IT ?*

recrute.parrot.com

Parrot®

Rejoignez les 900 passionné(e)s qui inventent les drones de demain

* Prêt à relever le défi ?

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

@ : service@elektor.fr

Tél. : (+33) 01.49.19.26.19
du lundi au vendredi de 10h à 13h

Fax : (+33) 01.49.19.22.37

www.elektor.fr | www.elektormagazine.fr

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

Publicité :

Fabio Romagnoli +32 485 65 40 90
fabio.romagnoli@eimworld.com

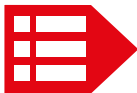
DROITS D'AUTEUR :

© 2016 Elektor International Media B.V.

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

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

Imprimé aux Pays-Bas
par Senefelder Misset - Doetinchem
Distribué en France par M.L.P.
et en Belgique par A.M.P.



Il était une fois ...

An 30 après la bataille de Yavin, Luke Skywalker s'est exilé sur une lointaine planète depuis plusieurs années, laissant derrière lui son droïde R2-D2 en mode « veille ». Le pilote Poe Dameron confie à son droïde BB-8 un morceau de carte qui permettrait de localiser Luke. Les aventures se poursuivent... R2-D2 sorti de son sommeil et BB-8 partagent leurs informations et assemblent la carte indispensable pour atteindre Luke. Un droïde apparu en l'an -32 avant BY qui dialogue avec un congénère né un demi-siècle plus tard, et tout cela sans mise à jour ! C'est vraiment de la science-fiction !¹

Voilà ce que je me suis dit en vidant un grenier où s'accumulaient les vestiges de trente ans d'informatique : Amstrad, Apple II, PC 8086/80286/i386/Pentium, divers supports (disquettes 5,25", 3,5", ZIP...), imprimantes à interface parallèle ou série... Impossible aujourd'hui de se servir de ces *ordinosaures* pourtant tous en état de marche. Tout a changé : les systèmes d'exploitation, les bus de données, les connecteurs... Et nous !

Dans dix ou quinze ans, nos débarras seront à leur tour encombrés d'objets connectés devenus inutilisables. Ils seront tombés en désuétude parce qu'Android aura été relégué par un système d'exploitation encore plus génial, parce que l'USB aura disparu, parce que WiFi et Bluetooth seront ringards...

Les fabricants n'ont même plus besoin de programmer l'obsolescence, ils ont déjà bien du mal à suivre.

Mariline Thiebaut-Brodier

* Pour ceux qui sortiraient de plusieurs mois de retraite à l'écart du monde, cette histoire se déroule dans le septième épisode de *la Guerre des étoiles*, *Le retour de la force*.

Notre équipe

Rédactrice en chef :	Mariline Thiebaut-Brodier (redaction@elektor.fr)
Rédaction internationale :	Thijs Beckers, Jan Buiting, Jens Nickel
Laboratoire :	Ton Giesberts, Luc Lemmens, Clemens Valens (responsable), Jan Visser
Coordination :	Hedwig Hennekens
Ont coopéré à ce numéro :	Thierry Destinobles, Robert Grignard, Jean-Louis Mehren, Denis Meyer, Hervé Moreau, Kévin Petit, Guy Raedersdorf
Service de la clientèle :	Cindy Tijssen
Graphistes :	Giel Dols, Mart Schroijen, Patrick Wielders
Elektor en ligne :	Daniëlle Mertens

- 5 bientôt dans Elektor
- 6 Elektor : guide de connexion(s)
- 24 initiation aux balises NFC de la famille ST25TA avec une pincée d'Arduino
- 28 salon CARTES 2015 : le dernier du nom ?
- 41 agenda mars 2016
- 62 l'e-choppe d'Elektor
- 80 des nouvelles du monde d'elektor
- 82 hexadoku casse-tête pour elektorniciens

DÉCOUVRIR CRÉER PARTAGER

- 8 bienvenue dans la section DÉCOUVRIR
- 9 trucs et astuces duo de LED sur une seule broche de port
- 10 tour du monde des cartes à μ C et kits d'embarquement pour tous
- 22 trucs & astuces pour Eagle (5) le format de fichier XML

DÉCOUVRIR CRÉER PARTAGER

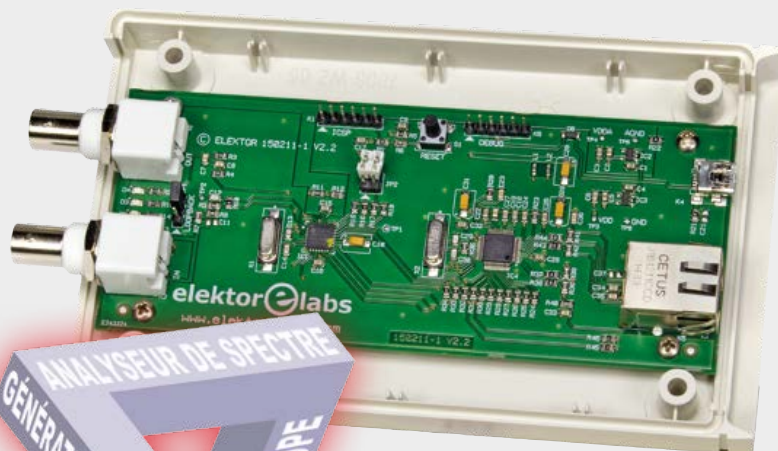
- 31 bienvenue dans la section CRÉER
- 32 compteur de visites pour vos pages web avec écran LCD sans fil
- 34 analyseur de signal par Ethernet (1) dsPIC33 + W5500 = oscilloscope, analyseur de spectre et générateur de signal tout-en-un
- 42 horloge aide-mémoire avec neuf rappels programmables
- 48 hors-circuits avec R. Lacoste le filtrage numérique sans stress : les filtres CIC
- 54 Platino, le retour par pitié, arrêtez de réinventer la roue

Au cours des dernières années, le marché des cartes à microcontrôleur n'a pas cessé de croître, avec son lot d'avantages et d'inconvénients. Une gamme plus étendue, mais l'embarras du choix ; de la concurrence, donc des prix en baisse. Ce tour d'horizon est donc utile, cependant il n'a pas la prétention d'être exhaustif. Vous y trouverez des cartes à 8 bits, mais aussi des kits pour réussir votre entrée dans le monde des 32 bits.



10

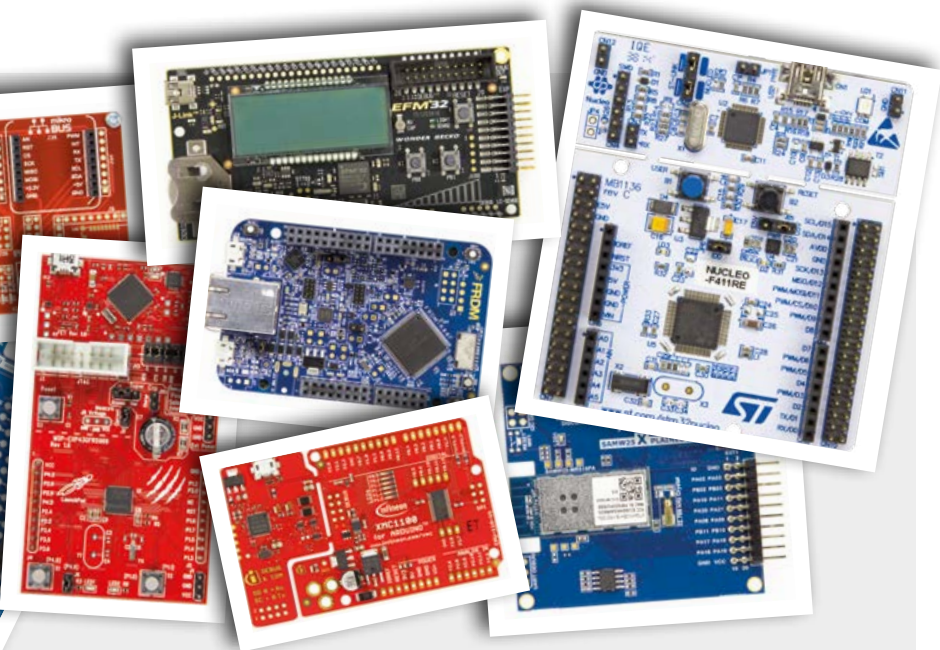
tour du monde et kits d'embarquement



34

analyseur de signal par Ethernet dsPIC33 + W5500 = oscilloscope, analyseur de spectre et générateur de signal tout-en-un

Si vous avez besoin d'un oscilloscope simple, d'un générateur de signal rudimentaire et d'un analyseur de spectre pratique, cet instrument trois-en-un est pour vous. L'ASPE, Analyseur de Signal Par Ethernet, est de surcroît compact et bon marché. À l'origine, je l'ai conçu comme outil d'analyse des systèmes de données échantillonnées, et comme moyen d'appréhender concrètement le potentiel et les limitations de ces systèmes.



des cartes à μ C pour tous

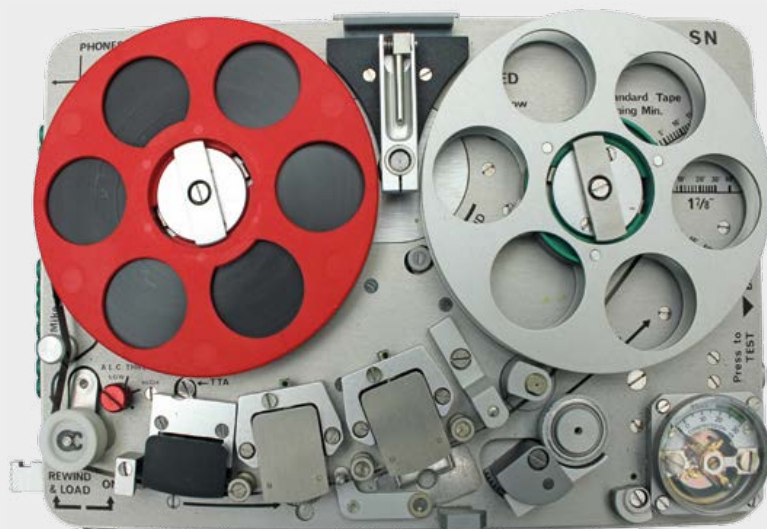
rétronique

NAGRA SN : l'oreille discrète de l'espion

un enregistreur très facile à dissimuler

James Bond l'aurait jaloué, ce petit jouet anodin à première vue, il recèle un enregistreur de son que Q lui aurait présenté avec fierté. Capable même de synchroniser le son avec l'image !

76



66 bienvenue dans la section PARTAGER

67 **Arduino/Genuino 101**
fin de règne pour la carte Arduino UNO ?

70 **projet 2.0**
corrections, mises à jour et courrier des lecteurs

72 **convertisseur buck tête domptée**
... ou peut-être que non ?

74 **le soudage d'une main sûre**
essayez-vous aux CMS

76 **Rétronique :**
Nagra SN, l'oreille discrète de l'espion
un enregistreur très facile à dissimuler

79 **bruits de labo...**
ressusciter de vieux gadgets avec
l'électronique la plus récente

 **bientôt sur ces pages**

Mini-kit Ampli-Op pour MyDAQ

Le mini-kit Ampli-Op pour MyDAQ est un système didactique sur les signaux analogiques. Il comporte des applications rédigées en LabVIEW pour l'étude des amplificateurs opérationnels dans leurs innombrables fonctions, les filtres audio par exemple.

Afficheur à sept segments Leditron

L'originalité de cet afficheur est d'utiliser des filaments de LED pour les segments, de quoi réaliser un numéro de maison lumineux pour sa façade ou un tableau de score.

Journal lumineux défilant Arduino

Une nouvelle manifestation des talents de votre Arduino ? Essayez ce journal lumineux et voyez comment afficher vos propres caractères sur une matrice de 16 x 16 LED.

Sous réserve de modification.

Le numéro d'avril 2016 paraîtra le 22 mars 2016.

la communauté d'Elektor

DÉCOUVRIR

CRÉER

PARTAGER

Elektor, c'est bien plus qu'un simple magazine. C'est une communauté d'électroniciens, du débutant au professionnel, désireux d'apprendre, de concevoir et de partager une électronique qui étonne.



Boutique en ligne d'Elektor: la cave de d'Ali Baba des électroniciens ! 10 % de remise permanente pour les membres GREEN et GOLD. www.elektor.fr



Elektor Magazine : dix numéros par an d'électronique inédite (projets, banc d'essais, cours, trucs et astuces, actualité...). www.elektormagazine.fr



Elektor PCB Service : gravure à la demande de vos circuits imprimés, en un ou plusieurs exemplaires. www.elektorpcbservice.com



Elektor GRATUIT & SANS PAPIERS : lettre d'information hebdomadaire à l'affût des nouveautés. Gratuite : www.elektor.fr/inscription



Elektor Academy : l'apprentissage par la pratique au travers de webinaires, séminaires, ateliers, DVD... www.elektor-academy.com



Elektor Boeken & DVD's: Arduino, Raspberry Pi, microcontrôleurs, Linux et bien d'autres. 10 % de remise pour les membres ! www.elektor.fr



Elektor TV : présentations, comptes-rendus, déballage et évaluation de produits. Pour y voir plus clair. www.youtube.com/user/ElektorIM



Elektor Labs: l'incubateur où partager ses projets et participer à ceux des autres. Nous faisons grandir vos idées ! www.elektor-labs.com

connectez-vous à notre communauté

formule GREEN

92,50 € par an

- ✓ l'accès à l'archive d'Elektor
- ✓ 10% de remise dans l'e-choppe
- ✓ 10 x magazine numérique
- ✗ 10 x magazine imprimé
- ✓ des offres exclusives
- ✓ l'accès à **Elektorlabs**
- ✓ un DVD annuel

connectez-vous

www.elektor.fr/formule-greencard

formule GOLD

127,50 € par an

- ✓ l'accès à l'archive d'Elektor
- ✓ 10% de remise dans l'e-choppe
- ✓ 10 x magazine numérique
- ✓ 10 x magazine imprimé
- ✓ des offres exclusives
- ✓ l'accès à **Elektorlabs**
- ✓ un DVD annuel

connectez-vous

www.elektor.fr/formule-goldcard

formule GRATUITE

- ✗ l'accès à l'archive d'Elektor
- ✗ 10% de remise dans l'e-choppe
- ✗ 10 x magazine numérique
- ✗ 10 x magazine imprimé
- ✓ des offres exclusives
- ✗ l'accès à **Elektorlabs**
- ✗ un DVD annuel

connectez-vous

www.elektor.fr/inscription

78

pays

246853

membres actifs

1031

experts & auteurs

479

publications

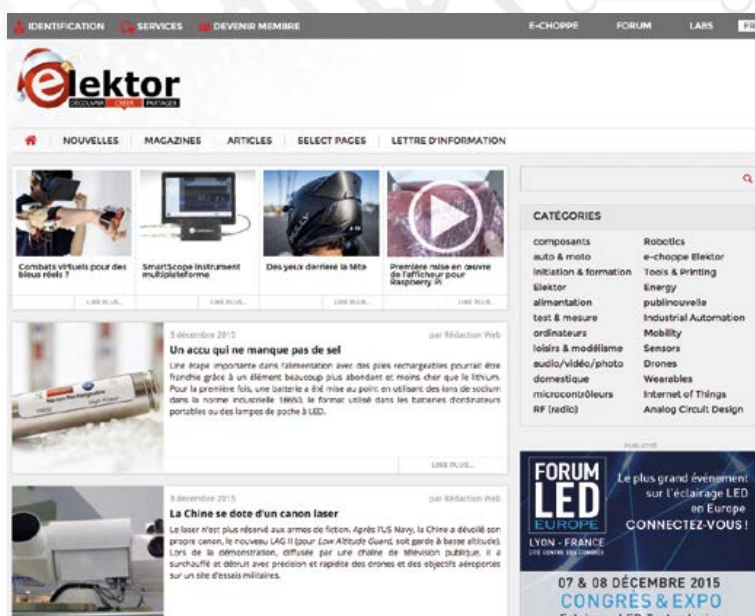
233628

visiteurs (par mois)

www.elektormagazine.fr

Explorez le monde de l'électronique

Entièrement rénové, notre site offre aux mordus d'électronique tout un monde de projets, d'articles et vidéos. Un bouton en haut du menu donne accès aux versions française, néerlandaise, allemande et anglaise du site. Le moteur de recherche intelligent permet de trouver rapidement informations et articles. Avec l'abonnement GREEN ou GOLD, vous pouvez naviguer dans toutes les archives d'Elektor et profiter de services complémentaires : offres et remises exclusives... Vous avez également accès aux préférences de votre compte, y compris l'abonnement au magazine papier et à la lettre d'information Elektor.



Elektor : gratuit & sans papiers

Restez informé, restez passionné !

Comme plus de 120 000 autres électroniciens, grâce à notre lettre d'information hebdomadaire gratuite Elektor, vous êtes au courant des dernières tendances et des récents développements de l'électronique. En outre, les abonnés à Elektor reçoivent régulièrement des offres et remises spéciales pour la boutique en ligne.

Pour vous abonner :
www.elektor.fr/inscription





Mon voyage dans l'Internet des objets

Dans le dernier numéro, je vous ai expliqué que nous nous sommes réparti entre rédacteurs les principaux domaines de l'électronique et que nous ne pouvons pas nous reposer sur nos lauriers, mais plutôt continuer d'apprendre.

Il est vrai que l'Internet des Objets (IdO) fait partie des domaines dont je m'occupe. J'ai déjà beaucoup lu sur ce sujet et vu de nombreux produits sur des salons. Cela m'a permis de me familiariser avec ce domaine, c'est important. J'ai surtout constaté que c'est un gigantesque jeu de construction avec des protocoles, des spécifications et des services qui peuvent à leur tour être combinés avec différentes solutions de type Internet des Objets.

Par quoi nos lec-

teurs sont-ils intéressés ? Quelles sont les solutions les mieux adaptées à des tâches (simples) de commande, mesure et régulation ? Les obstacles à franchir pour ses propres projets ne sont-ils pas insurmontables, que ce soit en matière de compréhension ou de coût ?

Je ne progresserai que si je me concentre d'abord sur quelques protocoles qui me semblent courants. Le classique TCP/IP qui permet de se connecter à des sites web à la maison est une bonne base. Le protocole MQTT qui s'appuie sur le TCP/IP (utilisé par ex. pour les célèbres cartes WunderBar) a l'air particulièrement intéressant. Toutefois je sais par expérience qu'on ne peut se fier à la simple étude des spécifications. Il faut passer à la pratique pour se confronter aux difficultés. C'est ainsi que mon voyage vers l'IdO débute, vous pouvez suivre mon périple sur www.elektormagazine.fr (cherchez les articles « Mon voyage dans le nuage »). Les épisodes de cette série paraissent également (pas régulièrement, mais presque) dans notre lettre d'information hebdomadaire. Si vous connaissez des liens vers des sites intéressants sur l'IdO, partagez-les : écrivez-moi à redaction@elektor.fr !

APP					
	HTTP	MQTT,			
NETWORK	TCP		UDP		
	IPv6/IPv4				
PHY/MAC	6 LoWPAN		WIFI	ETHERNET	2G/3G/4G
	IEEE 802.15.4				

Messenger

Les lecteurs de longue date savent que je souhaite mettre sur pied un petit serveur Messenger pour nos abonnés. Avec une telle station relais, il est aisé de développer sa propre application dans le domaine de la « mesure/commande par l'internet ». Le processus d'identification (*login*) des abonnés devrait suffire à assurer une certaine sécurité. C'est une des raisons pour lesquelles nous n'avons jamais terminé ce projet, en effet la base de données qui contient les informations sur nos abonnés a depuis été déménagée sur un autre système. L'idée n'a pas été abandonnée, elle fait simplement une petite sieste comme nombre d'autres idées ;-). Nous continuons de réfléchir à un moyen de lui donner vie, ce serait une vraie plus-value pour nos lecteurs. Ce projet pourrait reposer sur le protocole MQTT.

Comme d'habitude, je vous invite à me contacter (redaction@elektor.fr) pour me faire part de vos idées. Qu'est-ce qui vous intéresse ? Un moyen simple pour transmettre des messages courts



depuis n'importe où dans le monde, par ex. d'un ordiphone à des circuits électroniques domestiques ? Ou bien échanger des messages avec d'autres utilisateurs ? Elektor doit-il concevoir une solution propre ou soutenir un système déjà existant ? ◀

(150665 – version française : Thierry Destinobles)

trucs et astuces

les lecteurs écrivent aux lecteurs

Encore une solution futée qui facilite la vie des électroniciens



duo de LED sur une seule broche de port

Ralf Schmiedel



Avec les circuits à base de μC , il y a souvent pénurie de broches. Cette astuce permet de piloter, avec une seule broche de port, un duo de LED qui affiche trois couleurs. Ce circuit est dimensionné pour 5 V_{CC}.

La couleur que prend le duo de LED est déterminée par l'état de la broche de port, cf. le tableau.

Cela fonctionne parce que les deux LED verte et rouge du montage ont des tensions directes différentes. Pour les LED rouges, la tension directe (U_F) typique est de 1,6 V ; pour les LED vertes, elle est typiquement de 2,2 à 2,3 V.

Commençons par une broche de port à l'état **haut** : sur l'anode de la LED rouge, on a pratiquement V_{CC} , la cathode commune du duo de LED se trouve ainsi à $V_{CC} - U_{F\text{rouge}} = 3,4$ V. La LED rouge s'allume ($I_{\text{rouge}} = (V_{CC} - U_{F\text{rouge}}) / R1$). Mais, comme la différence de tension entre V_{CC} et la cathode commune des LED est inférieure à 2,2 V, aucun courant ne circule dans la diode verte.

Dans le cas d'une broche de port à l'état **bas**, l'anode de la LED rouge est presque à 0 V ; sur la cathode, on a la tension aux bornes de R1. De ce fait, la LED rouge est polarisée en inverse, et reste éteinte. Il circule maintenant un courant de V_{CC} via R2, la LED verte et R1 vers la masse : $I_{\text{vert}} = (V_{CC} - U_{F\text{vert}}) / (R2 + R1)$.

Pour finir, le cas de la broche de port à l'état haute impédance. Avec une tension d'alimentation de 5 V, les résistances R2 et R3 sont dimensionnées pour qu'il circule un courant sensiblement égal à travers les diodes et R1 vers la masse. Les deux LED sont allumées : le mélange des deux couleurs donne du jaune.

Il est donc possible de visualiser les états de travail typiques de nombreux circuits avec les trois couleurs : vert, jaune et rouge. Si on applique un signal d'horloge au port pour provoquer des changements de couleur, il est même possible de surveiller d'autres états.

Le circuit présente un inconvénient mineur : il y circule en permanence du courant, même lorsque le μC est en sommeil. ◀

(150666 - version française : Guy Raedersdorf)

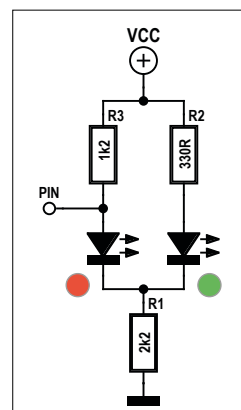


Tableau 1.

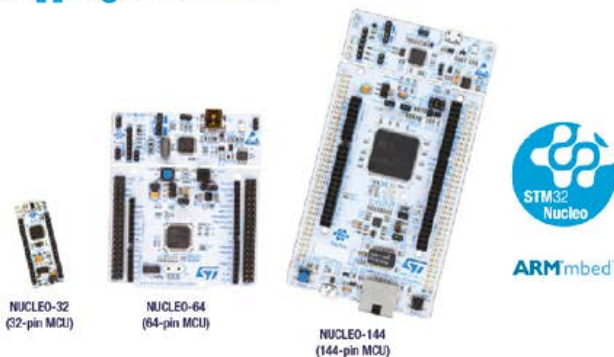
port	couleur
haut	rouge
bas	vert
haute impédance (port désactivé ou entrée)	jaune

Vous avez une solution futée pour arranger une bricole... Une façon bien à vous d'utiliser un composant ou un outil... Vous savez comment résoudre un problème plus facilement ou mieux qu'avec la solution actuelle... Écrivez-nous – chaque astuce publiée vous rapportera 40 € !



Publicité

Prototypage flexible



La plateforme open source **STM32 Nucleo** offre de nouvelles perspectives

www.st.com/stm32nucleo



tour du monde des cartes à μ C et kits d'embarquement pour tous

Viacheslav Gromov

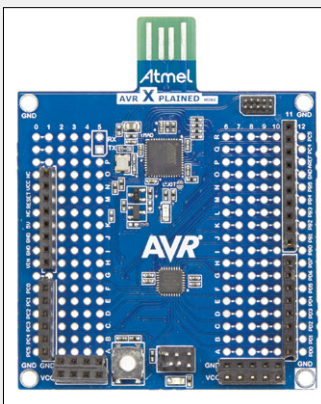
Au cours des dernières années, le marché des cartes à microcontrôleur n'a pas cessé de croître, avec son lot d'avantages et d'inconvénients. Une gamme plus étendue, mais l'embarras du choix ; de la concurrence, donc des prix en baisse. Ce tour d'horizon est donc utile, cependant il n'a pas la prétention d'être exhaustif. Vous y trouverez des cartes à 8 bits, mais aussi des kits pour réussir votre entrée dans le monde des 32 bits.

Notre survol des cartes les plus intéressantes et des kits d'initiation de différents constructeurs parcourt les familles de contrôleurs les plus répandues que l'on peut se procurer facilement chez nous. Toutefois pour apporter une note piquante, nous avons inclus quelques cartes « exotiques ». Bien entendu, nous n'avons pas oublié les outils logiciels indispensables avec toutes

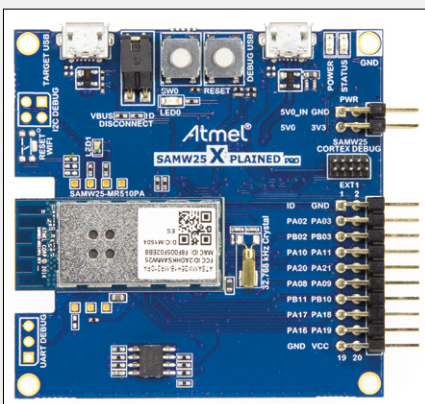
ces cartes. Les environnements de développement indépendants des marques comme Keil ou IAR ne sont pas très bon marché en version complète, mais il y a des versions gratuites pour des longueurs de code limitées. Les fabricants proposent souvent des outils de développement gratuits, nous les avons pointés du doigt.

Atmel

Le fabricant Atmel établi dans la Silicon Valley (Californie), bien connu des mordus de la famille AVR [1] à 8 bits reste



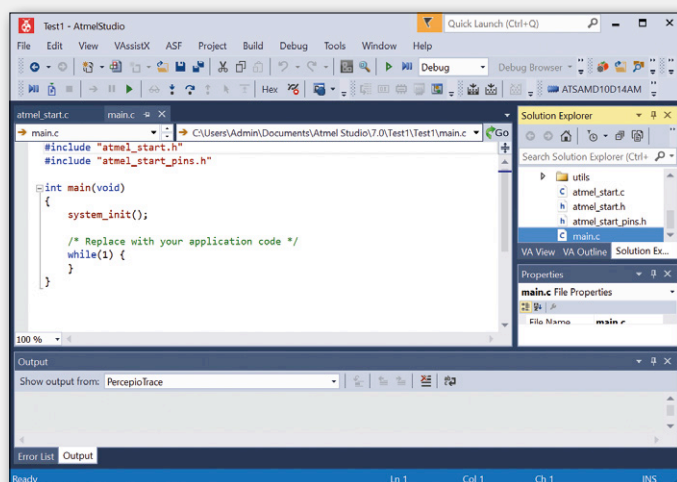
une grosse pointure dans le monde des 8 bits, d'accès facile pour les débutants. Jusqu'il y a peu, les kits de démarrage STK500 et STK600 étaient très populaires, mais, depuis, les cartes Xplained Mini (petites sœurs des très en vogue des Xplained Pro [2] désormais supportées par l'environnement de développement Mbed IDE), sont proposées pour moins

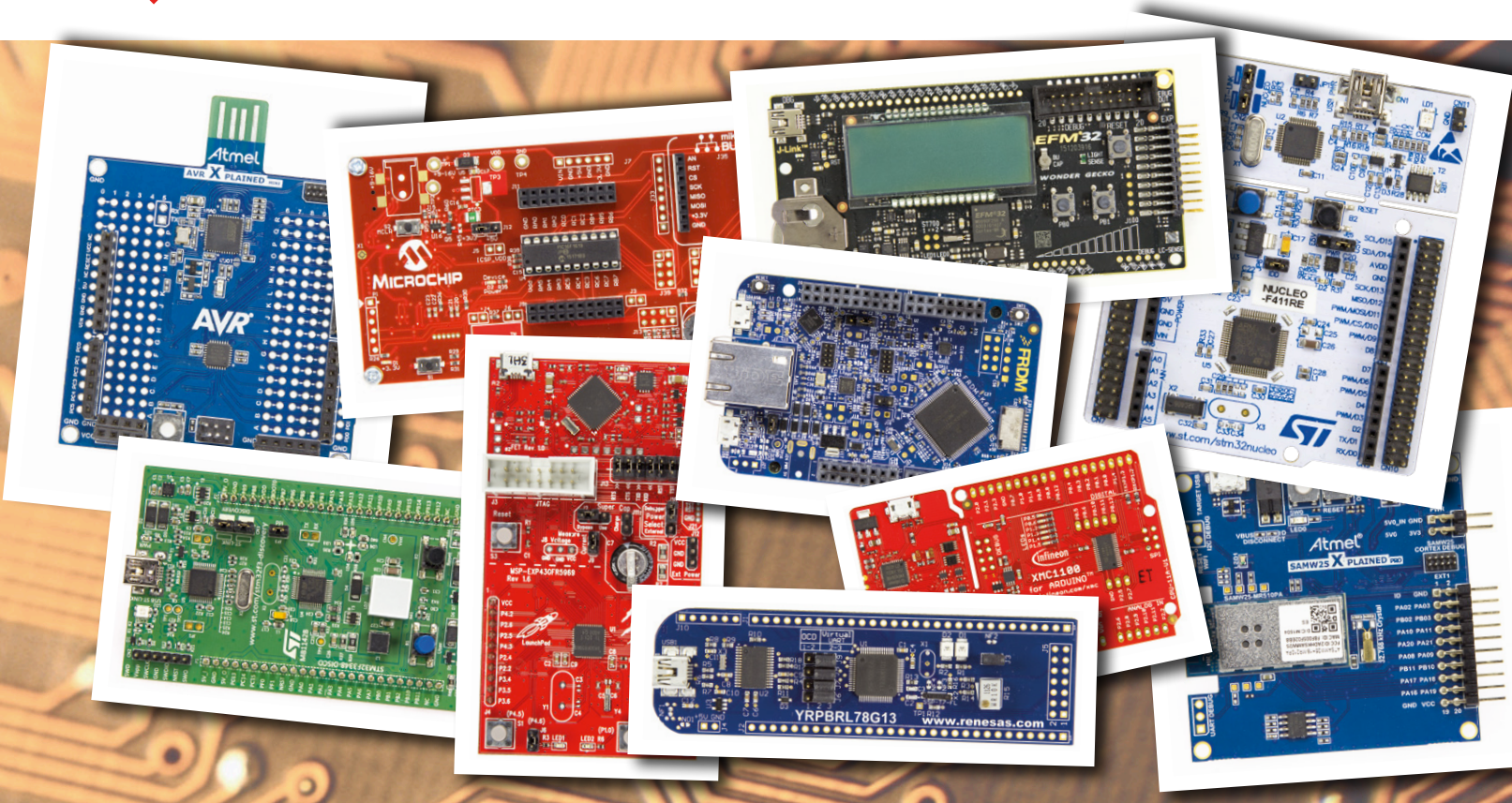


de 15 €. Elles sont disponibles avec processeur AVR, mais aussi avec le Cortex M0+ ARM à 32 bits. Elles disposent toujours du débogueur USB embarqué (EDBG) et il y a aussi deux champs de pastilles libres pour l'expéri-

mentation. On a encore la possibilité d'y souder deux barrettes au format Arduino R3 pour adjoindre différents shields Arduino.

On dénombre à présent cinq types de cartes dans cette famille : une avec ATmega168 ou ATmega168PB, une avec ATmega328P/PB et enfin une avec un microcontrôleur SAMD10. Les contrôleurs avec le suffixe ...B disposent de fonctions additionnelles comme les touches capacitives (QTouch) ou des périphériques améliorés. Grâce à l'Arduino Uno, vous connaissez sûrement l'ATmega328 (20 MHz, 32 Ko de flash, 1Ko d'EEPROM et 2 Ko de RAM). L'ATmega168 est tout pareil, mais avec la moitié de mémoire.





Pour adhérer au cercle des AVR à 8 bits, nous recommandons la carte **ATmega 328PB AVR Xplained Mini**, parce que ce microcontrôleur a tout ce qu'il faut : CA/N, interfaces, temporisateur, etc. Elle représente parfaitement toute la famille AVR. En revanche, pour se lancer dans le monde « Atmel SMART SAMD 32 bits ARM Cortex M0+ », le meilleur choix est la carte Xplained Mini avec le petit ATSAMD10. Depuis peu, il y a aussi le SAM W25, un ARM Cortex M0+ SAM D21 à 32 bits apte au WLAN. Il est utilisé sur la carte

SAMW25-Xplained-Pro [3].

Toutes les cartes Atmel présentées ici se programment avec Studio d'Atmel [4], un environnement de développement gratuit et facile à utiliser. On trouve de nombreuses bibliothèques et extraits de programme intégrés dans l'ASF, *Atmel Software Framework*. On peut aussi créer un projet avec l'outil en ligne Atmel START [5] et les bibliothèques adéquates.

Liens

- [1] www.atmel.com/products/microcontrollers/avr/
- [2] <http://developer.mbed.org>
- [3] www.atmel.com/tools/ATSAMW25-XPRO.aspx

- [4] www.atmel.com/microsite/atmel-studio/

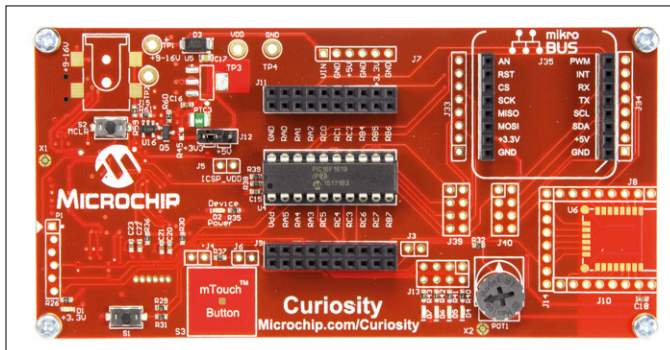
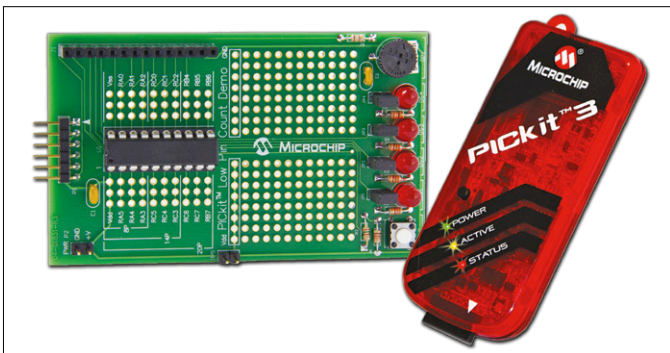
- [5] <http://start.atmel.com>

Microchip

Le fabricant de semi-conducteurs Microchip en Arizona est, avec sa famille de contrôleurs PIC, aussi connu qu'Atmel et ses AVR. Il y a trois groupes de PIC : les PICx à 8 bits, les (ds) PIC2x/3x à 16 bits et les PIC32 à 32 bits, comme il se doit. Vous connaissez certainement la famille des PIC1x à 8 bits. Elle peut monter à la cadence de 64 MHz et dispose de toutes les caractéristiques nécessaires dans la sphère des microcontrôleurs. Elle consomme avec parcimonie tout en fournissant assez de puissance et de périphériques, elle est souvent l'amie des spécialistes du prototype avec ses boîtiers DIP.

Les contrôleurs à 16 bits sont clairement plus puissants, disposent de plus de périphériques, y compris dans le domaine des interfaces, et fournissent, grâce à la DMA et à d'autres adjonctions, une grande puissance de traitement des signaux numériques (DSP). Les microcontrôleurs orientés spécialement vers le DSP s'appellent dsPIC. Certaines bibliothèques rendent les PIC à 16 bits parfaitement aptes à la commande de moteurs.

À la pointe de l'iceberg des PIC trône la famille à 32 bits, les PIC32, encore mieux dotés pour les applications audio,



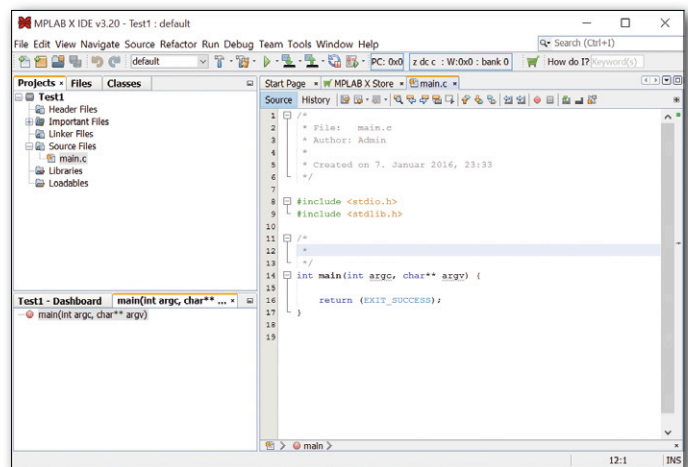
par exemple. Ces contrôleurs ont jusqu'à 2 Mo de mémoire flash et peuvent travailler à des fréquences d'horloge jusque 200 MHz.

Mais revenons aux microcontrôleurs à 8 bits nettement plus dociles. Pour les servir, il existe depuis un certain temps le kit de démarrage **PICKit 3** [7], au prix d'environ 80 €, composé du débogueur PICKit3 avec câble USB ainsi que d'un circuit imprimé sur lequel on peut insérer les deux PIC à 8 bits en boîtier DIP qui font partie de l'ensemble. On relie la carte à microcontrôleur au débogueur pour la programmer et déboguer. Il y a aussi sur la carte un champ de pastilles pour installer un petit circuit personnel et des barrettes de contacts pour donner accès aux broches du contrôleur principal. Un potentiomètre, un bouton-poussoir et quatre LED sont toujours utiles. On trouve aussi sur l'internet un *Guide pour l'utilisateur*, bien écrit (en anglais). En suivant les descriptions, on peut

réellement se familiariser progressivement avec les microcontrôleurs. Le bonus avec le PICKit 3, c'est la possibilité de programmer et déboguer l'ensemble de la famille des (ds) PIC. Donc quand, avec lui, vous avez acquis la pratique des types à 8 bits, vous pouvez continuer l'ascension dans les familles supérieures.

Récemment, on a présenté le **Curiosity Board** [8] voué à la plupart des contrôleurs à 8 bits enfichables. Le programmeur-débogueur USB est construit directement sur la carte. À un peu plus de 18 €, il est nettement moins cher que le kit de démarrage PICKit 3 ; en outre il offre bien plus de possibilités de connexions pour des périphériques externes. Si vous y ajoutez les barrettes de contacts, vous pourrez y brancher facilement les nombreuses cartes Click de MikroElektronika. On peut aussi y installer le module BLE (*Bluetooth Low Energy*) RN4020 de Microchip. Sont déjà présents un bouton-poussoir, quatre LED, une touche sensitive et un potentiomètre pour l'utilisateur.

Question logiciel, il y a un environnement de développement : sur le site de Microchip, on se procure gratuitement le MPLAB X IDE [9]. Après son installation, on peut y inclure de nombreuses bibliothèques pratiques et par exemple le *Code Configurator* [10], un composant additionnel qui simplifie grandement le développement du logiciel.



Liens

- [6] www.microchip.com/pagehandler/en-us/products/picmicrocontrollers
- [7] www.microchip.com/Developmenttools/ProductDetails.aspx?PartNO=DV164130
- [8] www.microchip.com/Developmenttools/ProductDetails.aspx?PartNO=DM164137
- [9] www.microchip.com/pagehandler/en-us/family/mplabx/
- [10] www.microchip.com/pagehandler/en-us/devtools/code_configurator/home.html

NXP

Le fabricant néerlandais de semi-conducteurs NXP a repris l'an dernier la firme texane concurrente Freescale, ce qui fait que tout le portefeuille des contrôleurs Kinetis de Freescale est passé dans le giron de NXP [11]. La famille est répartie en huit groupes, basés soit sur le cœur à 32 bits ARM Cortex M0+, soit sur celui du 32 bits ARM Cortex M4. D'ailleurs,

elle comprend aussi les plus petits microcontrôleurs ARM du monde. Les groupes sont spécialisés dans des applications particulières, par exemple la transmission radio (série Kinetis W) ou l'économie d'énergie (série Kinetis L). Plusieurs groupes, comme celui de la série Kinetis K, sont en fait très universels.

Liens

- [11] www.nxp.com/products/microcontrollers-and-processors/arm-processors/kinetis-cortex-m:KINETIS
- [12] www.nxp.com/products/software-and-tools/hardware-development-tools/freedom-development-boards/freedom-development-platform-for-kinetis-k64-k63-and-k24-mcus:FRDM-K64F
- [13] www.nxp.com/products/microcontrollers-and-processors/arm-processors/kinetis-cortex-m/l-series:KINETIS_L_SERIES
- [14] www.nxp.com/products/microcontrollers-and-processors/arm-processors/lpc-arm-cortex-m-mcus/lpc-cortex-m0-plus-m0/lpc1100-series/lpcxpresso-board-for-lpc11u68:OM13058
- [15] www.lpcware.com/content/project/lpcopen-platform-nxp-lpc-microcontrollers

STMicroelectronics

Le fabricant de semi-conducteurs STMicroelectronics (ST), société internationale de droit français, mise essentiellement sur deux séries différentes de cartes pour la famille puissante et multifonctionnelle STM32 avec ARM Cortex M [16], les cartes Discovery [17] et Nucleo [18].

Les cartes Discovery de divers prix disposent, par rapport aux cartes Nucleo, de périphériques très diversifiés, le plus souvent avec plusieurs LED et boutons-poussoirs ainsi que de nombreux capteurs analogiques et numériques comme un microphone ou un accéléromètre à trois axes. Souvent, la périphérie est adaptée à la spécialité du contrôleur principal, comme un écran E-Ink (papier électronique). En outre, on

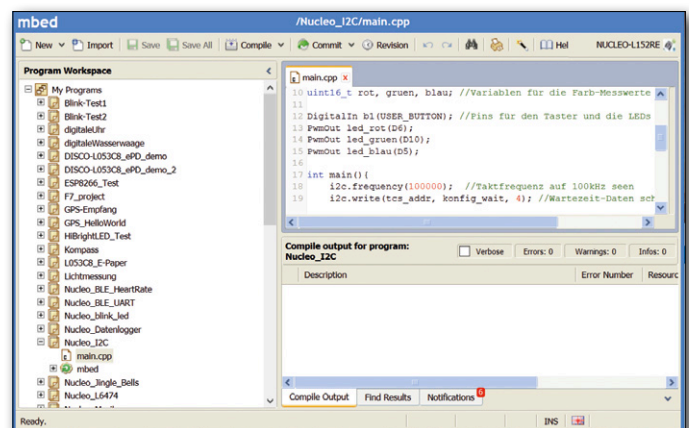
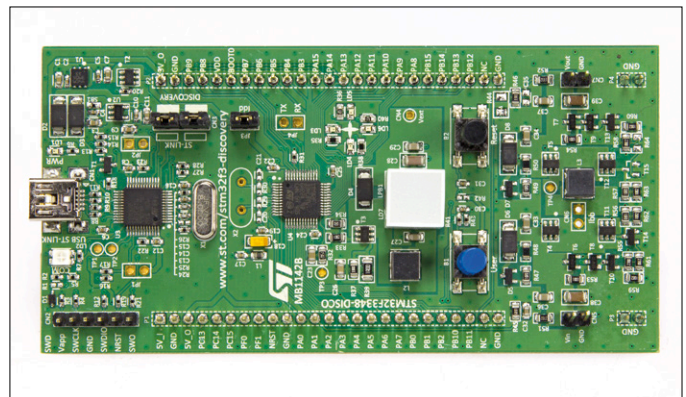
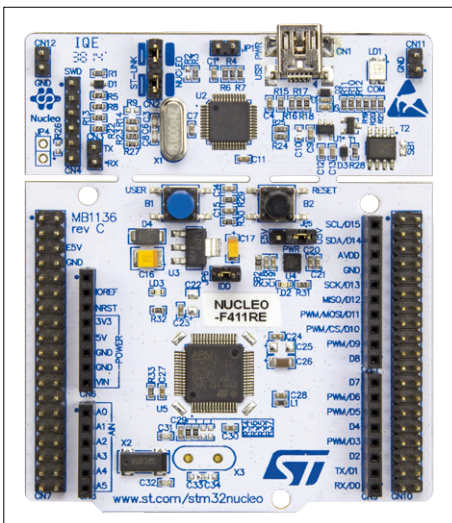
y trouve la plupart du temps des doubles barrettes à picots sur lesquelles on a accès aux broches du microcontrôleur.

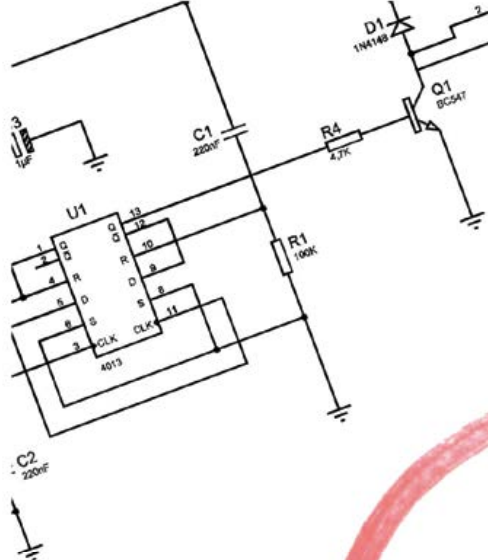
Les cartes Nucleo sont un peu moins fournies, mais c'est ce qui les rend plus uniformes et plus universelles. Elles possèdent comme les Discovery un débogueur USB ST Link/V2-1 que

l'on peut cependant isoler de la carte du microcontrôleur principal. On dispose alors de la place pour construire son propre circuit avec le contrôleur, le régulateur de tension, les boutons-poussoirs (RàZ et autres fonctions à déterminer), et la LED (usage libre). Le débogueur ainsi mis à part, peut servir, comme avec la carte Discovery, à programmer et déboguer d'autres microcontrôleurs *in situ* par le connecteur SWD. Outre les doubles barrettes à picots que l'on retrouve sur presque toutes les cartes Discovery, la Nucleo possède des rangées de contacts compatibles avec Arduino R3 pour se connecter au système Arduino (ST propose d'ailleurs des *shields* intéressants). Il existe à présent une douzaine de cartes Nucleo avec différents microcontrôleurs à 32 bits ARM Cortex M, dont certaines depuis peu en très petit format avec microcontrôleur à 32 broches, depuis celles à très faible consom-

mation jusqu'aux modèles de grande puissance, ce qui fait des cartes Nucleo un bon choix pour débiter. Le mieux est alors une carte Nucleo avec un microcontrôleur bien équipé en périphériques, comme la **Nucleo F411RE** (Cortex M4F, 100 MHz, 512 Ko de flash, 128 Ko de SRAM). Vous disposez alors de tout ce qu'il faut pour vos premiers pas ! Autre solution, la très attrayante carte **Discovery STM32F334C8** (Cortex M4, 72 MHz, 64 Ko de flash, 16 Ko de SRAM) avec une LED de puissance, un convertisseur *Buck-Boost* et bien d'autres choses. Enfin, n'oublions pas la famille STM8 [19], diversifiée et disponible, mais uniquement sur carte Discovery.

Pour la programmation des deux séries de cartes, on se sert surtout des IDE pour ARM très répandus, comme Keil ou IAR, auxquels ST contribue avec un générateur de code très utile, accompagné de toutes les bibliothèques nécessaires, appelé STM32CubeMX pour les microcontrôleurs STM32 [20]. La famille STM8, en revanche, a recours entre autres à l'IDE





INGELEC

SPECIALISTE

du **CORDON SUR MESURE**
Cherche

Projet **ATYPIQUE** et **AMBITIEUX**
Pour relation

LONGUE et **FRUCTUEUSE**

Toutes les propositions seront étudiées

Contact: ingelec@ingelec.net
Tel: 01.60.62.35.25

Depuis 1988, INGELEC se spécialise dans la réalisation de **sous ensembles électroniques**, et plus particulièrement de **cordons surmoulés, élaborés sur-mesure**.

Nous proposons aujourd'hui à nos clients une gamme complète de **produits standards** au meilleur prix. Nous sommes également à leur écoute pour leurs projets de **cordon sur-mesure**.

Petites séries, grandes séries, nous répondons à tous types de besoins, et nous adaptons à de nombreux secteurs d'activité: **Domotique, monétique, médical, industriel, péri-informatique...**



STVD (ST Visual Develop IDE) [21]. Le grand avantage des cartes Nucleo tient depuis longtemps à la possibilité d'utili-

ser Mbed et donc d'accéder à ses très bonnes bibliothèques et aux manuels.

Liens

[16] www.st.com/web/en/catalog/mmc/FM141/SC1169?sc=stm32

[17] www.st.com/web/en/catalog/tools/FM116/SC959/SS1532/LN1848?icmp=ln1848_pron_pr-stm32f446_dec2014&sc=stm32discovery-pr

[18] www.st.com/web/en/catalog/tools/FM116/CL1620/SC959/SS1532/LN1847?s_searchtype=keyword

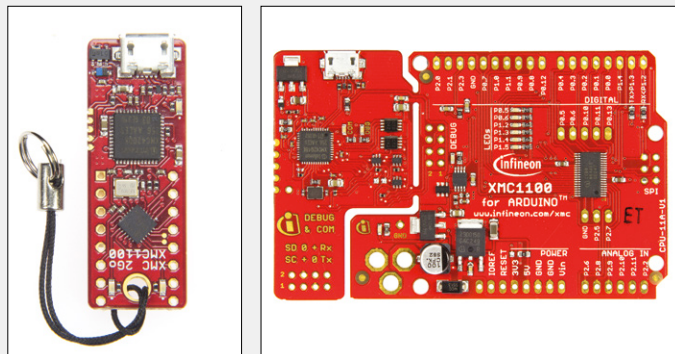
[19] www.st.com/web/en/catalog/mmc/FM141/SC1244

[20] www.st.com/web/catalog/tools/FM147/CL1794/SC961/SS1533/PF259242?s_searchtype=partnumber

[21] www.st.com/web/catalog/tools/FM147/CL1794/SC1807/SS1747/PF210567?s_searchtype=partnumber

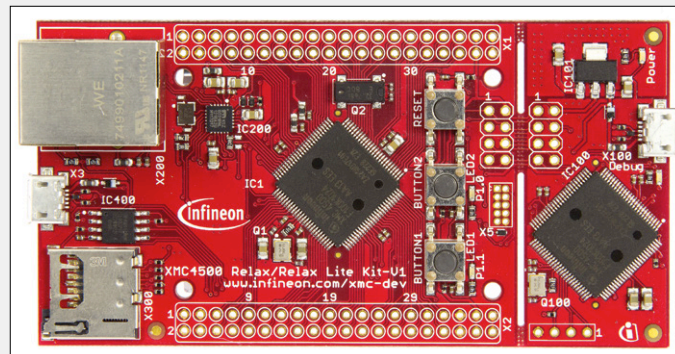
Infineon

Dans le domaine des microcontrôleurs, le fondateur de puces allemand Infineon est surtout connu par sa famille XMC [2]. Elle se subdivise en deux groupes : la série XMC1000 avec le cœur à 32 bits ARM Cortex M0 et la série XMC4000 avec ARM Cortex M4F à 32 bits aussi, le F désigne l'unité à virgule flottante intégrée. Les deux groupes se distinguent par une très grande variété d'interfaces, parfois spéciales comme l'interface de position (POSIF) et MultiCAN dans la série XMC1000,

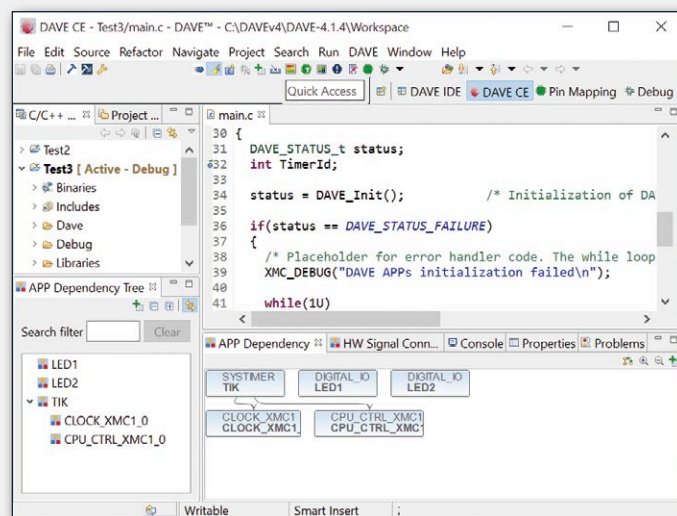


ainsi qu'Ethernet, SD et EtherCAN dans la série XMC4000. Il existe de nombreuses cartes pour cette famille de puces, dont le kit **XMC1100 Boot** (32 MHz, 64 Ko de flash, 16 Ko de RAM) [23] à 20 €, avec débogueur amovible J Link LITE (USB), plusieurs LED libres, ainsi que la place au format Arduino R3 pour les barrettes de connexion à installer soi-même. Infineon propose aussi des *shields* très intéressants. Pour le groupe XMC4000, de son côté, il y a le kit **XMC4500 Relax** (120 MHz, 1 Mo de flash, 160 Ko de SRAM) au prix attractif de 35 €. Il est équipé de plusieurs périphériques, dont un lecteur de carte SD ou une prise micro-USB ; il est aussi disponible en version allégée sans Ethernet [24]. Beaucoup moins chère et plus petite, sans doute la plus petite du marché actuel, à 6 € et également dotée du microcontrôleur du groupe XMC1100, c'est la **XMC2Go** (32 MHz, 64 Ko de flash, 16 Ko de RAM) [25]. Cette carte, de la grandeur d'une clé USB, est aussi dotée du microcontrôleur du groupe XMC1100 et dispose d'un débogueur USB J Link LITE pour la programmation et le débogage. Elle n'a que deux LED libres et 16 connexions, mais cela peut suffire à un petit projet. En y ajoutant des barrettes à picots sur les broches

du microcontrôleur, on peut brancher un BoB et réaliser des projets sans plus devoir souder. Cette minuscule carte ne dispose évidemment pas d'un débogueur amovible comme les kits Boot et Relax, mais en revanche, il y a sur le côté un connecteur SWD (*Serial Wire Debug*) pour programmer et déboguer avec d'autres microcontrôleurs. XMC2Go est une bonne rampe d'accès, par la suite, à des kits plus performants comme Boot ou Relax.



Si nous jetons un coup d'œil au logiciel, les cartes (et généralement les microcontrôleurs XMC) dépendent de l'IDE DAVE, de source Infineon. Il est disponible [26] sans frais, à condition de s'annoncer. Il se base sur des applis, il y en a une



pour presque chaque périphérique, on la glisse dans l'espace de travail et on peut la lier à d'autres applis avec un symbole. On configure chaque appli, et donc chaque périphérique, d'un clic droit et plus tard, on pourra l'attribuer aux broches voulues du microcontrôleur. On peut alors rédiger le code nécessaire et encore composer le noyau du logiciel.

Liens

- [22] www.infineon.com/cms/de/product/microcontroller/32-bit-industrial-microcontroller-based-on-arm-registered-cortex-registered-m/channel.html?channel=db3a30433c1a8752013c3e221b9d004f
- [23] www.infineon.com/cms/en/product/evaluation-boards/KIT_XMC11_BOOT_001/productType.html?productType=db3a30443b360d0e013b8f5163c46f62
- [24] www.infineon.com/cms/en/product/evaluation-boards/KIT_XMC45_RELAX_V1/productType.html?productType=db3a304437849205013813b23ac17763
- [25] www.infineon.com/cms/en/product/evaluation-boards/KIT_XMC_2GO_XMC1100_V1/productType.html?productType=db3a304443537c4e01436cceb5d154f
- [26] www.infineon.com/cms/en/product/microcontroller/development-tools-software-and-kits/dave-version-4-free-development-platform-for-code-generation/channel.html?channel=db3a30433580b37101359f8ee6963814

Texas Instruments

Ce fabricant de semi-conducteurs texan (on s'en doute) est connu pour la famille de microcontrôleur MSP430 à 16 bits [27] bien équipée à tous égards. Elle associe une grande puissance, une construction ordonnée, avec de nombreux périphériques et, qui plus est, s'adapte à la faible consommation. Les cartes de la série LaunchPad sont tout indiquées pour débiter.

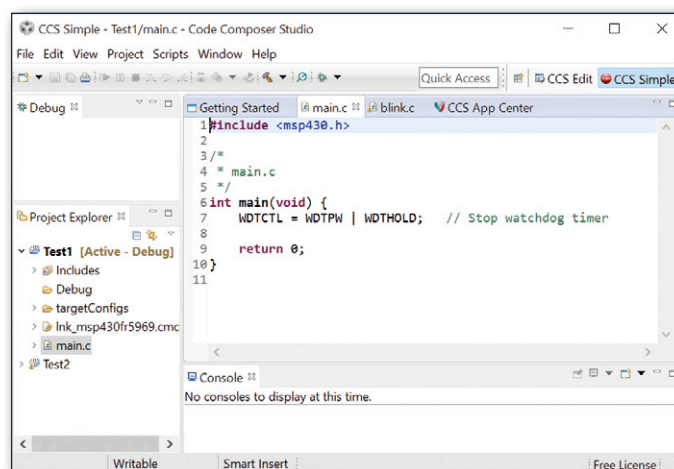
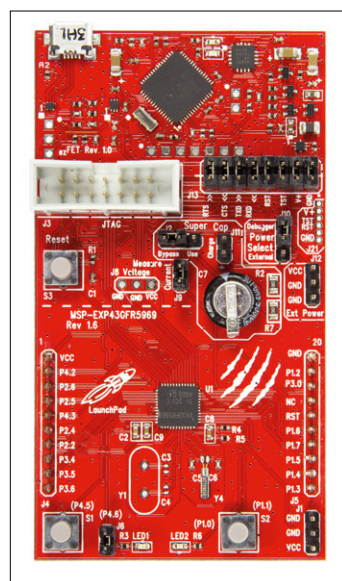
La carte la plus populaire est à environ 15 €, c'est la **MSP-EXP430G2 LaunchPad** [28], avec deux microcontrôleurs MSP430 en boîtier DIP. Mais actuellement il y a encore cinq autres Launchpad dans la famille MSP430, dont le MSP432, comme représentant d'un nouveau groupe à 32 bits, équipé du cœur ARM Cortex M4F. Les nouveaux *Pad* ont beaucoup plus de périphériques, comme un LCD ou une fonction dite *EnergyTrace* qui surveille la consommation pendant le fonctionnement.

Toutes les cartes disposent d'un émulateur eZ FET (USB) pour déboguer et programmer le contrôleur à bord, dont les conducteurs de données peuvent être séparés par plusieurs cavaliers du contrôleur principal CMS pour arriver à programmer aussi un autre microcontrôleur.

Des connecteurs à longues broches sont toujours présents sur les deux bords de la carte pour y brancher l'un des nombreux BoosterPacks [29] de Texas Instruments (TI). Depuis l'écran à encre électronique jusqu'à la

roue tactile, il y a dans la gamme des BoosterPacks tout ce que le concepteur peut rêver d'associer à son projet. Vous pouvez évidemment utiliser aussi ces broches pour vos propres périphériques. De même, il y a toujours au moins un bouton-poussoir et deux LED disponibles. En outre, sur les cartes les plus récentes, on trouve encore des capteurs de température ou d'accélération.

Si vous souhaitez rejoindre cet environnement, vous pouvez choisir un LaunchPad éprouvé tels le MSP-EXP430G2 ou l'un des nouveaux comme le **MSP-EXP430FR5969** [30]. Comme son immatriculation l'indique, la carte est équipée d'un microcontrôleur MSP430FR5969, rythmé à 16 MHz, avec 64 Ko de FRAM, 2 Ko de SRAM. Si un microcontrôleur apte au WLAN vous intéresse, passez à la série Connected des LaunchPads. La série C2000 est orientée vers la commande de moteurs et la série Hercules vers la sécurité, les applications médicales en particulier.



Il existe de nombreuses possibilités pour créer du code pour la famille MSP430. Principalement pour les LaunchPads, il y a un environnement de développement comparable à l'IDE Arduino et gratuit du nom d'Energia [31]. Il est très simple, les fonctions de commande sont très semblables à celle d'Arduino. Il est idéal pour les premières expériences et de petits projets.

En revanche, qui veut développer du logiciel en véritable C et exploiter toutes les fonctions et caractéristiques de la famille MCU, ira chercher le Code Composer Studio (CCS) de TI [32],

basé sur Eclipse. La version gratuite est limitée à 16 Ko pour les microcontrôleurs MSP430 courants et à 32 Ko pour les MSP432. Si vous n'employez pas le compilateur de TI, mais plutôt l'ancien GCC, toutes ces limitations tombent. Depuis peu, TI propose aussi le CCS Cloud [33], une variante de l'autre, simplement déplacée sur Internet. Pour peu que vous ayez accès à la toile, vous pouvez ainsi développer votre code n'importe où, juste dans votre navigateur et à l'aide de deux petits outils, l'extraire du navigateur pour le programmer dans le microcontrôleur et le déboguer directement.

Liens

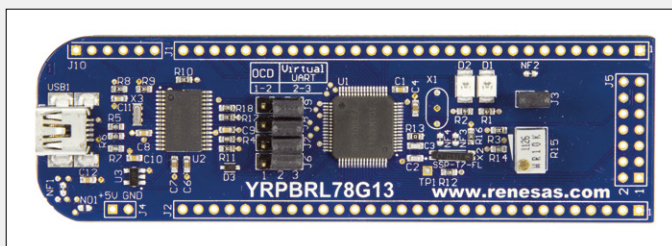
- [27] www.ti.com/lscds/ti/microcontrollers_16-bit_32-bit/msp/overview.page
- [28] www.ti.com/tool/msp-exp430g2
- [29] www.ti.com/ww/en/launchpad/boosterpacks.html
- [30] www.ti.com/tool/msp-exp430fr5969?keyMatch=launchpad%20fr5969&tisearch=Search-EN
- [31] <http://energia.nu/>
- [32] www.ti.com/tool/ccstudio
- [33] <http://dev.ti.com/about>

Renesas

Renesas a été réduit à une portion congrue dans Elektor, parce que ce fabricant japonais est assez peu connu chez nous. Pourtant, il n'y a rien d'étonnant à ce que, pour les familles de microcontrôleur à 16 bits les plus renommées de Renesas RL78 [34], il existe des cartes intéressantes et aisément disponibles ; tout autant que le **Studio e²** basé sur Eclipse, qui fournit un support élaboré puisqu'il contient, entre autres, un générateur de code à télécharger gratuitement, après enregistrement et présentation du demandeur. Comme introduction dans la famille RL78, une carte comme **YRPBRL78G13** [35] est tout indiquée, elle est actuellement en promotion. Sa version RL78/G13 accepte une fréquence d'horloge jusqu'à 32 MHz, possède 64 Ko de flash, 4 Ko de RAM et 4 Ko de flash de données. Ces cartes, Renesas en offre gratuitement à des foires commerciales à des fins publicitaires, mais elles sont en vente pour environ 30 € chez

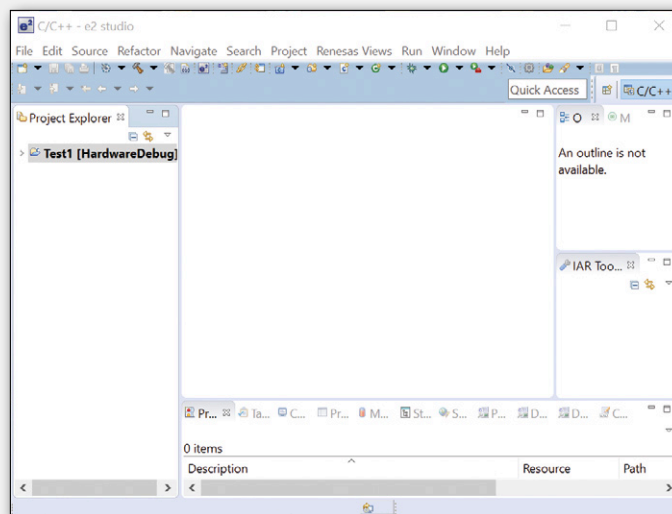
spécialisés, comme celui des RL78/Lxx orientés vers la commande de LCD. En somme, cette famille est bien équipée en périphériques et riche en performances. Si vous prenez la carte YRPBRL78G13, vous avez tout ce qu'il faut pour commencer. Contrairement aux kits de démarrage de Renesas souvent dix fois plus chers, celui-ci n'a pas d'émulateur E1 (programmeur et débogueur USB), mais le câble USB pour la liaison à l'ordinateur et le tournevis pour le réglage du potentiomètre d'ajustage sont fournis.

Pour la programmation et le débogage, il y a sur la carte un OCD, débogueur « on chip » USB. Pour l'utiliser, il vaut mieux avoir téléchargé quelques outils supplémentaires pour le Studio e², comme *Renesas Programming Tool*. Vous trouverez de nombreux outils et une riche documentation dans le DVD joint au kit, ainsi qu'un programme de démonstration qui vous fera découvrir les principales fonctions des RL78.



de nombreux distributeurs. Les microcontrôleurs RL78 sont efficaces et appropriés à l'usage industriel. Dotés de nombreux temporisateurs, entre autres, ils sont tout indiqués pour la commande de moteurs. En outre, ils sont parmi les plus économes en énergie, on le remarque immédiatement d'un coup d'œil à la fiche technique.

La grande famille RL78 compte de nombreux groupes très



Le micrologiciel à cet effet est déjà disponible d'origine sur le microcontrôleur. Plus tard, quand vous voudrez y inscrire votre logiciel, vous n'aurez qu'à mettre les quatre cavaliers J6 à J9 sur le mode OCD.

Il y a une LED libre et un potentiomètre d'ajustage sur la carte ; en outre, de nombreuses broches du microcontrôleur sont accessibles sur les deux bords (J1 et J2). On peut y souder des barrettes à picots pour utiliser la carte comme plaque d'expérimentation. Si avec le temps, vous voulez ajouter un émulateur E1, vous pourrez installer une barrette supplémentaire à l'arrière sur J5. On peut mesurer la consommation de courant des microcontrôleurs d'usage général en remplaçant J3 par un ampèremètre.

Liens

[34] <http://am.renesas.com/products/mpumcu/r178/index.jsp>

[35] http://am.renesas.com/products/tools/introductory_evaluation_tools/renesas_promo_board/yrpbr178g13/index.jsp

[36] <http://am.renesas.com/products/mpumcu/rx/>

[37] http://am.renesas.com/products/tools/ide/ide_e2studio/index.jsp

[38] www.kpitgnutools.com/index.php

Si vous voulez monter à 32 bits chez Renesas, c'est la famille de microcontrôleurs RX [36] qui s'invite. Récemment, la famille Synergy à 32 bits est sortie sur le marché avec les très puissants microcontrôleurs ARM Cortex M qui peuvent travailler jusqu'à 300 MHz.

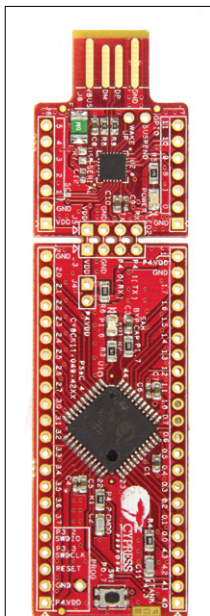
Revenons sur le Studio e² [37] : après l'installation, y compris de la panoplie d'outils de KPIT [38], vous pouvez rédiger un projet en C(++) avec le générateur de code. Il s'occupe pour vous de toutes les configurations, il vous suffira de cliquer sur les périphériques internes nécessaires, d'adapter à vue quelques paramètres et vous n'aurez plus qu'à vous concentrer sur le noyau du programme.

Cypress

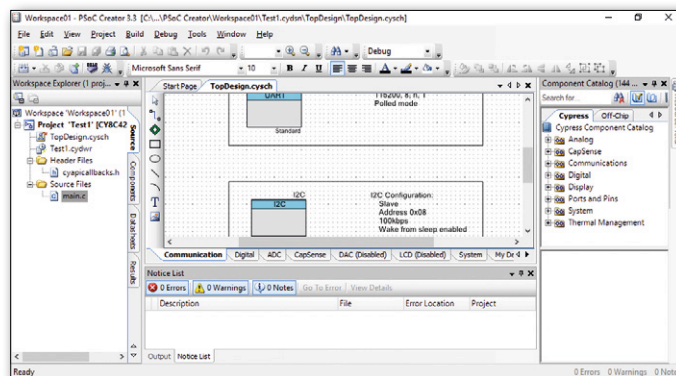
Dans la Silicon Valley aussi, Cypress fabrique des semi-conducteurs, mais avec une certaine originalité, du moins en ce qui concerne sa famille de microcontrôleurs PSoC [39]. Chaque groupe se base sur un cœur différent, ARM Cortex M depuis 8 bits jusqu'à 32 bits. L'universalité et la souplesse de ces microcontrôleurs reposent sur la composition « à la carte » des circuits périphériques. Avec PSoC Creator, vous pouvez organiser votre microcontrôleur comme avec des briques Lego, choisir par exemple le nombre de temporisateurs ou de CA/N et sur quelles broches, dans certaines limites, bien sûr. Ce n'est qu'ensuite qu'il faut rédiger son logiciel, opération qui ne comporte pas de difficultés avec PSoC Creator.

Voulez-vous connaître la souplesse de cet environnement exaltant ? Pour 10 € tout au plus, vous aurez une carte **CY8CKIT-043** [40], **CY8CKIT-049** [41] ou **CY8CKIT-059** [42]. Elles sont très compactes et, à part la LED et le bouton-poussoir habituels, elles n'ont presque pas de périphériques embarqués. Mais il y a de nombreuses broches du microcontrôleur accessibles sur des pastilles rigoureusement alignées sur les bords de la carte où vous pourrez brancher vos périphériques. Et si par exemple vous utilisez la technologie tactile, vous ne devrez pas installer de condensateur de référence, il est déjà sur la carte.

Les deux cartes CY8CKIT-049 (avec le



PSOC 4100 et le PSoC 4200) coûtent environ 5 €. Toutefois, les cartes 049 ne sont équipées que d'un chargeur d'amorçage sur le contrôleur et d'un convertisseur UART vers USB, si bien qu'en plus du PSoC Creator, il faut encore d'autres outils logiciels pour le chargeur d'amorçage. Lors de la programmation, il ne faut pas oublier de charger aussi ce logiciel sous la forme d'une brique au moment de configurer le microcontrôleur. En outre, il n'y a pas de possibilité de débogage. Pour le double du prix, la situation est plus confortable avec d'autres cartes de prototypage. Elles portent un débogueur USB, on peut ainsi laisser tomber le chargeur. Avec le débogueur à bord, on a même le loisir de programmer d'autres



microcontrôleurs PSoC. Si vous n'osez faire que le premier pas, une carte CY8CKIT-049-42xx (Cortex-M0, 48 MHz, 32 Ko de flash, 4 Ko de SRAM) suffira largement, mais si vous préférez déjà réaliser un projet modeste, les autres cartes avec un débogueur amovible sont intéressantes. Évidemment, Cypress propose aussi beaucoup de cartes plus grandes et

plus richement dotées [43], avec depuis peu, un module BLE (Bluetooth Low Energy) embarqué dans le microcontrôleur. Le Creator PSoC est gratuit au téléchargement sur [44].

Liens

[39] www.cypress.com/products/programmable-system-chip-psoc

[40] www.cypress.com/documentation/development-kitsboards/cy8ckit-043-psoc-4-m-series-prototyping-kit

[41] www.cypress.com/documentation/development-kitsboards/psoc-4-cy8ckit-049-4xxx-prototyping-kits

[42] www.cypress.com/documentation/development-kitsboards/cy8ckit-059-psoc-5lp-prototyping-kit-onboard-programmer-and

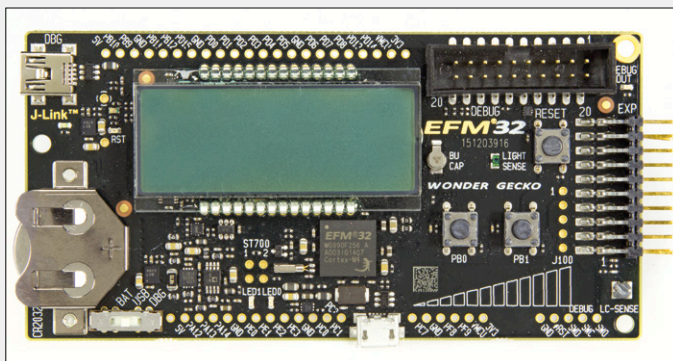
[43] www.cypress.com/products/psoc-creator-integrated-design-environment-ide

[44] www.cypress.com/documentation/development-kitsboards/cy8ckit-044-psoc-4-m-series-pioneer-kit

Comme le programmeur PSoC est compris dedans, tout tient dans le même outil.

Energy Micro

Le fabricant de semi-conducteurs Silicon Labs d'Austin (Texas) a racheté il y a quelques années l'entreprise Energy Micro avec Mikrocontroller-Sparte. Maintenant, les contrôleurs tels que EFM32 [45] sont disponibles avec différents cœurs ARM Cortex M à 32 bits. Les microcontrôleurs « EFM Zero Gecko » sont basés sur un cœur Cortex M0+ à 24 MHz, tandis que les « Wonder Geckos » le sont sur un Cortex M4 à 48 MHz.



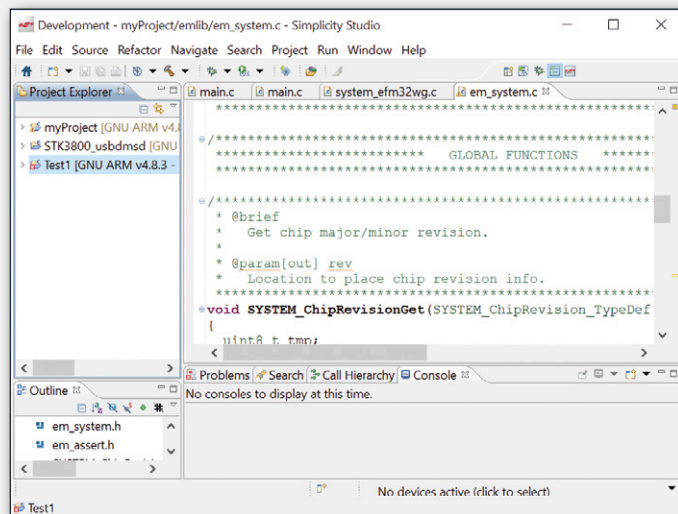
Tous ont ensemble la faculté de travailler avec peu de courant. Malgré cela, ils ne lésinent ni sur les périphériques ni sur les performances. Toutes les interfaces actuelles sont disponibles, certains types sont même capables de piloter un LCD. Les groupes de la série à 8 bits EFM8 [46] obéissent à la même frugalité en courant que leurs grands frères, seulement dans une moindre mesure quand il s'agit de tourner à 48 MHz.

Pour chaque groupe d'EFM32 à 32 bits, il y a un kit de démarrage aux alentours de 30 €. Ils sont tous très différents, mais ils sont tous bien équipés en périphériques et possèdent un débogueur USB sur la carte. De nombreuses broches du microcontrôleur sont accessibles sur les bords du circuit imprimé. On y trouve généralement, en plus des sempiternels boutons-poussoirs et LED libres, des écrans, des

capteurs de lumière, des dispositifs tactiles et bien d'autres choses intéressantes. Chaque carte a sa pile de 3 V pour alimenter sans difficulté ces microcontrôleurs très sobres. Pour programmer la carte ou la déboguer, il n'y a qu'à déplacer un interrupteur à glissière et elle s'alimente sur le port USB. Que vous faut-il de plus ? Si vous optez pour cette famille de microcontrôleurs, mieux vaut acheter une carte avec un microcontrôleur d'une relativement grande puissance comme le kit **EFM32 Wonder Gecko** (Cortex M4F, 48 MHz, 256 Ko de flash, 32 Ko de RAM) [47].

On peut développer le logiciel pour les kits avec Mbed, mais toutes les cartes ne l'acceptent pas encore, ou bien avec le Studio Simplicity [48], lequel est très convivial avec le développeur. On peut configurer les périphériques à vue et il y a de nombreux tutoriels pour accompagner les utilisateurs novices. ◀

(150687 - version française : Robert Grignard)



Liens

[45] www.silabs.com/products/mcu/32-bit/efm32-gecko/Pages/efm32-gecko.aspx

[46] www.silabs.com/products/mcu/8-bit/Pages/efm8.aspx

[47] www.silabs.com/products/mcu/lowpower/pages/efm32wg-stk3800.aspx

[48] www.silabs.com/products/mcu/Pages/simplicity-studio.aspx

DÉCOUVRIR

CRÉER

PARTAGER

elektor•PCB Service

le fruit de la coopération d'elektor et d'eurocircuits



**confiez-nous
la production de vos circuits imprimés,
vous avez tout à y gagner !**

➡ **le prix !**

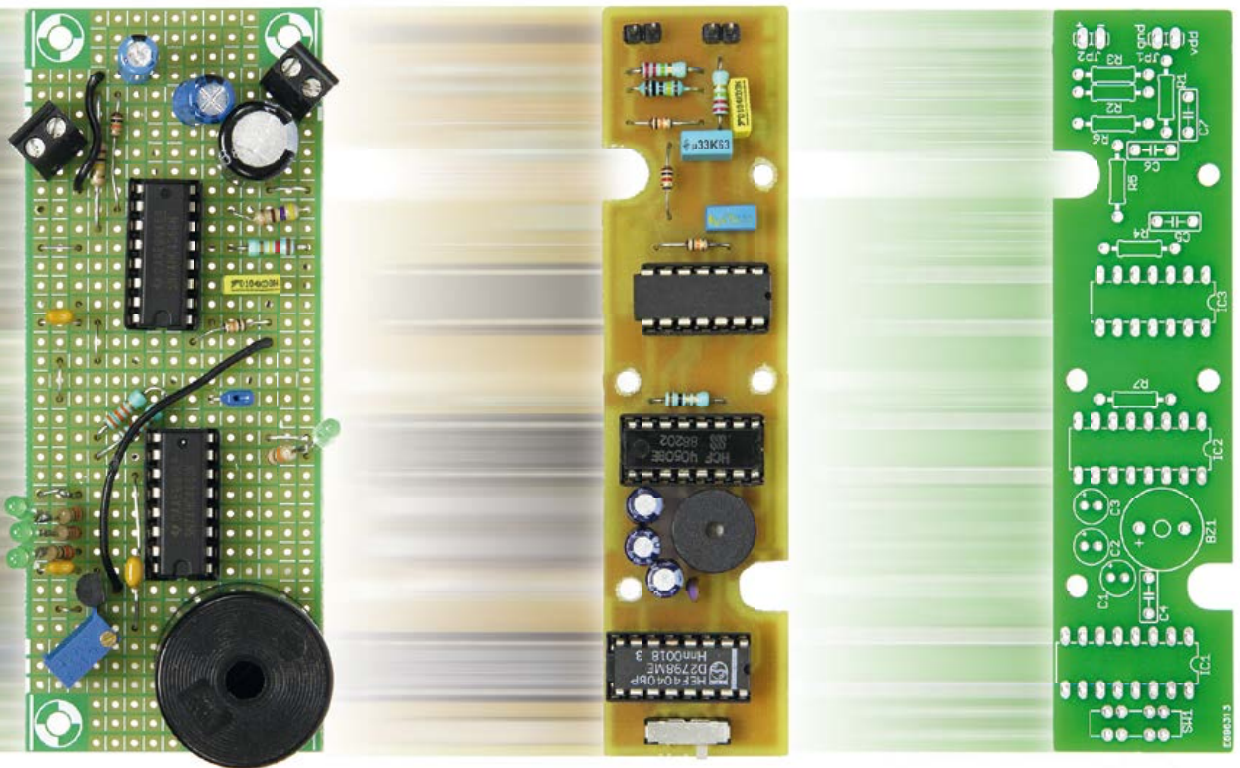
irrésistible...

➡ **la qualité !**

imbattable...

➡ **la fiabilité !**

inégalée...



elektor PCB service est, à l'échelle européenne, le service le plus complet de fabrication sur mesure de circuits imprimés. Ce service commence en ligne, par le confort et l'efficacité d'outils faits sur mesure, étudiés pour vous permettre de visualiser votre commande et de l'analyser *avant* de payer.

- Pour vos débuts, vous utiliserez éventuellement le service de **prototypage sans masque de soudure** ; dans ce cas, vos circuits imprimés simple ou double face sont nus.
- Pour les utilisateurs exigeants, nous proposons les options sans compromis. Le **PCB Visualizer** montre les circuits imprimés tels qu'ils seront livrés, le **PCB checker** procède à une vérification technique de votre circuits imprimés (*design rules check*), et enfin le **PCB configurator** facilite la préparation de la commande.

Des menus bien conçus et un guidage par options accélèrent le processus de commande sans négliger aucun détail. Au moment de passer votre commande, vous savez exactement ce qui sortira de nos machines.



Qu'il soit perso ou pro, confiez votre prochain PCB à :

www.elektorPCBservice.com

trucs & astuces pour Eagle (5)

le format de fichier XML



Neil Gruending (Canada)

Petit précis d'anatomie informatique :
les ailes XML d'Eagle

Depuis sa version 6, Eagle utilise un format XML pour tous les fichiers de conception. Ce qu'il y a de bien avec les fichiers XML, c'est qu'ils sont directement interprétables par l'utilisateur, autrement dit peuvent aisément être lus et, au besoin, modifiés.

Structure d'un fichier XML

Les fichiers au format XML (*Extensible Markup Language*, langage de balisage extensible) sont des fichiers texte qui suivent les règles de formatage de la spécification XML 1.0 du W3C (*World Wide Web Consortium*) [1]. Les détails formels du langage sont nombreux, mais les règles de base plutôt simples. XML est d'ailleurs un surensemble d'HTML.

Ce qui intéresse avant tout l'utilisateur d'Eagle est la façon dont sont créés les éléments XML pour stocker de l'information. Un élément XML est délimité par une balise ouvrante ayant la forme `<élément>` et par une balise fermante notée `</élément>`.

La balise ouvrante peut contenir des attributs décrivant plus précisément l'élément : `<élément attribut="valeur">`. Entre les balises d'ouverture et de fermeture peuvent figurer d'autres éléments imbriqués ou du texte. Si un élément n'est défini que par ses attributs, on peut utiliser une balise vide de la forme

`<élément attribut="valeur" />`, le caractère `/` sert à fermer cette balise unique.

Un élément XML est par définition tout ce qui est compris entre les balises d'ouverture et de fermeture, y compris celles-ci.

Dans le **listage 1** par exemple, extrait d'un fichier de carte Eagle, l'élément `layers` est délimité par les balises `<layers>` et `</layers>` et comprend quatre éléments `layer` imbriqués ; chacun est défini par une balise vide dont les attributs décrivent la couche en question.

Les fichiers XML d'Eagle ne comportent pas de commentaires, mais on peut en placer en les délimitant par `<!--` et `-->`.

Le format XML d'Eagle

Le format de fichier XML d'Eagle est défini dans le fichier `eagle.dtd` du dossier « documentation » du répertoire d'installation. Vous pouvez l'ouvrir depuis l'éditeur de texte d'Eagle, mais je conseille plutôt un éditeur de texte qui a une coloration syntaxique pour XML. Un fichier DTD (*Document Type Definition*) définit un format de fichier XML avec le langage XML lui-même, et sert aussi à vérifier si un fichier XML est correctement formaté.

Les fichiers DTD peuvent être très complexes, mais celui d'Eagle est facile à décrypter. La première section de la DTD (**listage 2**), `Entity`, définit tous les types de données permis ainsi que leurs valeurs. En effet un type peut être une liste de valeurs. La première ligne définit une entité appelée `Int` de type CDATA

Listage 1. Exemple de fichier Eagle XML

```
<layers>
<layer number="1" name="Top" color="4" fill="1" visible="no" active="no"/>
<layer number="16" name="Bottom" color="1" fill="1" visible="no" active="no"/>
<layer number="97" name="Info" color="7" fill="1" visible="yes" active="yes"/>
<layer number="98" name="Guide" color="6" fill="1" visible="yes" active="yes"/>
</layers>
```

Listage 2. Section Entity

```
<!ENTITY % Int          "CDATA" >!-- an integer number -->
<!ENTITY % Real         "CDATA" >!-- a real number -->
<!ENTITY % String       "CDATA" >!-- a character string -->
<!ENTITY % Bool         "(no | yes)">
<!ENTITY % Coord        "%Real;" >!-- coordinates, given in millimeters -->
<!ENTITY % Dimension    "%Real;" >!-- dimensions, given in millimeters -->
<!ENTITY % Layer        "%Int;" >!-- layer number -->
<!ENTITY % GridUnit     "(mic | mm | mil | inch)">
<!ENTITY % GridStyle    "(lines | dots)">
<!ENTITY % WireStyle     "(continuous | longdash | shortdash | dashdot)">
```


Listage 3. Définition des éléments.

```
<!ELEMENT eagle (compatibility?, drawing, compatibility?)>
<!ATTLIST eagle
    version          %Real;          #REQUIRED
    >
    <!-- version: The EAGLE program version that generated this file, in the form V.RR -->

<!ELEMENT drawing (settings?, grid?, layers, (library | schematic | board))>

<!ELEMENT library (description?, packages?, symbols?, devicesets?)>
<!ATTLIST library
    name             %String;        #REQUIRED
    >
```

(*Character Data*, données caractères). Idem pour les entités suivantes, *Real* et *String*. Une entité *Bool* ne peut prendre que deux valeurs *no* ou *yes*, et une *GridStyle* uniquement *lines* ou *dots*. Dans la DTD, on peut utiliser la syntaxe *%type*; pour se référer aux valeurs autorisées. L'entité *Layer* a par ex. la valeur *%Int*; qui est le type CDATA. Eagle convertira le CDATA en un *Int* lorsqu'il chargera le fichier, de sorte que seuls les formats numériques seront implicitement permis.

La section suivante de la DTD, *Element*, définit la façon dont la structure et les données XML seront sauvegardées dans un fichier XML. Le **listage 3** montre comment sont définis quelques éléments et leurs attributs.

La définition d'un élément est délicate, car il faut en lister tous les éléments imbriqués. Ici l'élément *eagle* peut contenir l'élément facultatif *compatibility*, l'élément obligatoire *drawing*, et un autre élément facultatif *compatibility*. Le mot-clé *!ATTLIST* définit les attributs et types d'un élément, et s'ils sont obligatoires, facultatifs, ont une valeur fixe ou une valeur par défaut. Ici l'élément *eagle* a un attribut obligatoire appelé *version* qui doit être un nombre réel. L'élément *drawing* peut éventuellement contenir un élément *settings*, peut

éventuellement contenir un élément *grid*, a un élément *layers*, et un élément *library*, *schematic* ou *board*. La définition des éléments imbriqués se poursuit jusqu'à ce qu'un élément n'ait que des attributs et aucun élément imbriqué.

Exemple réel

Maintenant que vous en savez un peu plus sur la façon dont Eagle utilise XML, vous devriez être en mesure de comprendre le **listage 4**, extrait d'un véritable fichier de schéma (les ... symbolisent du code qui a été retiré).

Bien sûr, en pratique vous n'aurez peut-être jamais besoin d'un fichier XML d'Eagle, mais il est bon d'avoir ce format à disposition puisqu'il permet de nombreux traitements textuels, comme comparer deux fichiers, et qu'il facilite aussi la sauvegarde des fichiers dans un système de gestion de versions. Restez maître de vos données ! ◀

(150639 – version française : Hervé Moreau)

Lien

[1] Spécification XML 1.0 : www.w3.org/TR/REC-xml/

Listage 4. Fichier de schéma (extrait)

```
<package name="DIL20">
<description>&lt;b&gt;Dual In Line Package&lt;/b&gt;&lt;/description>
<wire x1="12.7" y1="2.921" x2="-12.7" y2="2.921" width="0.1524" layer="21"/>
...
<pad name="1" x="-11.43" y="-3.81" drill="0.8128" shape="long" rot="R90"/>
...
<text x="-13.081" y="-3.048" size="1.27" layer="25" rot="R90">&gt;NAME</text>
<text x="-9.779" y="-0.381" size="1.27" layer="27">&gt;VALUE</text>
</package>
...
</packages>
...
</library>
</libraries>
...
</schematic>
</drawing>
</eagle>
```

initiation aux balises NFC de la famille ST25TA

avec une pincée d'Arduino

Martin Cooke (Royaume-Uni)

Avez-vous participé au jeu-concours NFC avec la balise offerte par STMicroelectronics et Elektor dans le dernier numéro ? Un ordiphone Android permettait de découvrir l'UID de la balise. Simple, mais c'est là justement que réside la puissance de la technologie NFC : elle permet de créer facilement des applications où l'utilisateur accède instantanément au web et y effectue des transactions. Vous rêvez de vous y mettre ? Voici comment.

Nous l'annonçons dans les articles de janvier/février 2016 [1] [2], STMicro a étendu sa gamme de balises NFC avec une nouvelle famille baptisée ST25TA. Les balises NFC/RFID de cette série prennent en charge l'interface sans contact *ISO 14443 type A*, la spécification *NFC Forum Type-4*, ainsi que les commandes associées. Les balises NFC de la famille M24SR sont *a contrario* à « double interface », c.-à-d. que l'accès à leur mémoire peut se faire via RF ou interface matérielle I²C. Elles sont aussi compatibles avec l'interface sans contact *ISO 14443 type B*. Le contenu des mémoires des balises ST25TA n'est accessible que par interface RF, mais certaines balises ont un signal *General Purpose Out* (GPO) qui peut déclencher le matériel d'un système embarqué proche. Nous y reviendrons plus bas.

Au commencement était la RF

La distance maximale de lecture/écriture entre une balise et un lecteur dépend de la sensibilité du dispositif de lecture/écriture. Nous avons observé une distance d'environ 4 cm avec une tablette PC Nexus 7. La trace d'oscilloscope de la **figure 1** montre les impulsions d'interrogation d'environ 12 μ s produites par un lecteur NFC fonctionnant à 13,65 MHz. Le débit maximal via une liaison NFC est de 424 Kbits/s. Comme cette valeur n'est pas optimale pour le transfert des gros fichiers, les applications Android pair à pair utilisent habituellement la liaison NFC pour configurer – c.-à-d. établir – la communication via un protocole de bande

passante supérieure, par exemple Bluetooth ou Wi-Fi. C'est la méthode qu'utilise la fonction *Android Beam* (Android version 4.1 et sup.) pour amorcer une liaison Bluetooth et ainsi accélérer le transfert d'un fichier. La fonction *S-beam* de Samsung fait de même, mais initie une liaison Wi-Fi.

Structure de la mémoire d'une balise

Les balises ST25TA prennent en charge l'application *NDEF Tag* définie dans la spécification *NFC Forum Type 4 Tag*. L'espace mémoire de la balise est divisé en trois fichiers :

- le fichier *Capability Container* (CC) ;
- le fichier NDEF (*NFC Data Exchange Format*) ;
- le système de fichiers spécifique à STMicroelectronics.

Ces fichiers sont décrits en détail dans

les fiches techniques des balises. Pour résumer, le fichier CC est en lecture seule et indique la capacité de la balise. Le fichier NDEF est l'espace où un message peut être stocké, l'espace disponible pour ce fichier correspondant à la capacité de mémoire de la balise ; il peut être lu et écrit, et aussi protégé par un mot de passe de 28 bits.

Le système de fichiers est l'espace où vous pouvez configurer la balise, p. ex. sa protection en écriture/lecture par un mot de passe. Vous pouvez aussi initialiser un compteur de 20 bits pour enregistrer le nombre de fois où la balise a été lue ou écrite. Cette information est utile dans le cas p. ex. d'une balise insérée dans l'affiche « intelligente » d'un lieu public. Vous pouvez aussi y définir le comportement du signal de sortie GPO, une fonction de la puce ST25TA02K-P utilisée dans la carte d'évaluation CLOUD-ST25TA. Certaines options de configuration sont irréversibles, donc testez-les avec prudence.

Lecture et écriture

L'appli Android *ST25 Demo App* de STM permet de lire et écrire des données dans les balises ST25TA. Elle fonctionne sur tablettes/ordiphones compatibles NFC et vous pouvez la télécharger sur Google Play ou le site de STM [3]. Il existe d'autres applis NFC capables de lire le contenu des balises ST25TA, mais souvent elles ne permettent pas de modifier certains fichiers propres à STMicroelectronics.

Certaines applis NFC « reconditionnent » une balise. La première fois que celle-ci

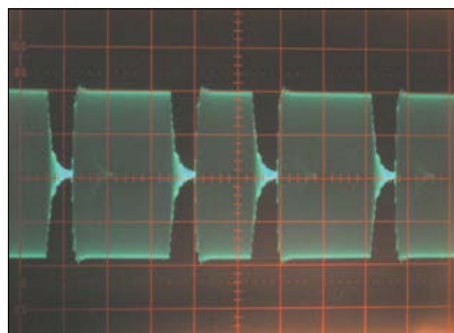


Figure 1. Impulsions d'interrogation d'un lecteur NFC émettant à 13,65 MHz.



est lue, l'UID est stocké. Vous pouvez alors définir l'action que la balise exécutera chaque fois qu'elle sera lue. Il n'y a aucune donnée à stocker dans la balise – cela fonctionne aussi avec les balises non-inscriptibles. Vous pourriez par exemple scanner la balise d'une carte bancaire sans contact (expirée), puis associer les données et actions à exécuter lorsque la balise est lue.

La plupart des passeports électroniques actuels contiennent une balise NFC. Chose étrange, cette balise renvoie un UID différent chaque fois qu'elle est scannée ! En fait ces changements garantissent au détenteur du passeport qu'il ne peut pas être « pisté » avec l'UID associé à son document. Grâce à un générateur de nombres aléatoires embarqué, la fonction RUID (*Random UID*) présente un UID différent à chaque accès sur la puce.

Appli Demo App de STM

Pour lire ou écrire dans la mémoire de votre balise, téléchargez l'appli *Demo App* de STM, puis lancez-la depuis un appareil compatible NFC. Scannez votre balise : vous avez maintenant accès aux données des trois fichiers internes, vous pouvez protéger leur contenu par mot de passe avec les outils de l'appli, et au besoin initialiser le compteur. Le bouton *Compose NDEF* de composition de message a les options suivantes :

Text – le message de type texte entré ici s’affiche sur l’ordiphone/tablette lorsque la balise est lue.

URI – entrez l'*Uniform Resource Iden-*

tifier, soit l'URL ou l'adresse web vers laquelle sera dirigé le destinataire après lecture de la balise.

Contact – entrez vos coordonnées ici : elles seront ajoutées à la liste des contacts du destinataire lorsqu’il scan-
nera la balise. Si la mémoire de la carte
est suffisante, vous pouvez inclure une
image de faible résolution.

M24SR – cette option spécifique à la carte M24SR-Discovery permet d'en commander certaines fonctions matérielles.

WiFi – établit une connexion à un réseau Wi-Fi.

Bluetooth – affiche une liste de clients Bluetooth disponibles pour un envoi de message par Bluetooth.

SMS – entrez le numéro de téléphone et un message SMS. Le message est envoyé au numéro de téléphone.

Email – entrez le courriel du destinataire et le message. Passez la balise devant le lecteur pour envoyer le courriel.

AAR – lanceur d’appli Android. Sélectionnez une appli installée ; l’application enregistrée est lancée à chaque lecture de la balise.

Une fois ces options complétées, placez la balise sous l'appareil pour qu'y soient écrites les données saisies.

Carte d'évaluation CLOUD-ST25TA

La carte d'évaluation CLOUD-ST25TA illustre une façon d'intégrer une balise dans un produit fini à peu de frais. Vous

pouvez en gagner une en participant au jeu-concours NFC Elektor/STMicro [1]. La petite carte d'évaluation montrée séparément sur la **figure 2** et à gauche de la **figure 3** comporte une bobine imprimée d'émission/réception ainsi que des connexions pour l'utilisation du signal GPO. La carte d'évaluation est équipée d'une balise ST25TA02K-P dotée d'une EEPROM de 2 Kbits pour le stockage du message NDEF, et a un débit de 106 Kbits/s. Elle possède aussi une broche de sortie GPO dont l'option *Tools* de l'appli *ST25 Demo App* devrait permettre d'activer certaines options de programmation. L'accès aux registres internes avec un appareil NFC de lecture/écriture permet de modifier les options suivantes de la sortie GPO :

SessionOpen : une session RF est en cours.

MIP : modification en cours du message
NDEF.



Figure 2. Où puiser dans l'intelligence de la carte d'évaluation CLOUD-ST25TA.

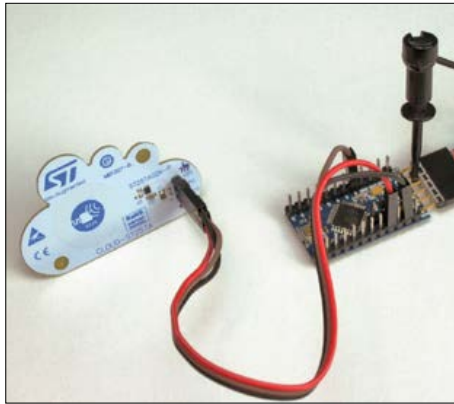


Figure 3. La carte CLOUD-ST25TA configurée pour envoyer un signal de sortie de veille à la Pro Mini.

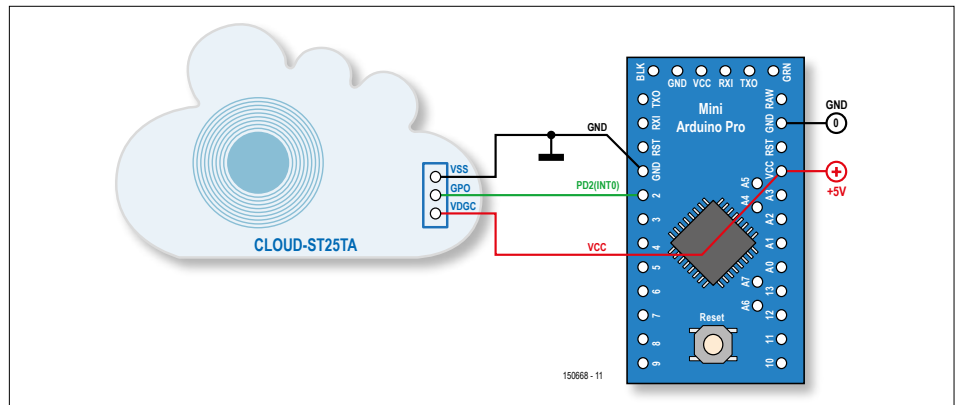


Figure 4. Rien de plus simple que de relier l'Arduino à la CLOUD-ST25TA puisque sa broche GPO accepte des tensions entre 1,65 et 5,5 V.

WIP : écriture en cours.

INT (Interrupt) : l'hôte RF peut forcer la ST25TA02K-P à envoyer une impulsion sur la broche GPO.

State mode : l'hôte RF commande la broche GPO directement.

RF busy : un hôte RF communique avec la ST25TA02K-P.

Field detection : l'intensité du champ RF reçu est suffisante pour communiquer avec la ST25TA02K-P (état par défaut).

Vous pouvez verrouiller ces options de façon permanente en mettant à 1 le bit le plus significatif du champ de commande GPO. Notez que toutes ces options ne sont pas configurables via l'appli. Dans son état par défaut, la sortie GPO est mise au niveau haut lorsque le champ RF reçu a une intensité suffisante. Dans cette configuration, l'état de la broche GPO est plutôt aléatoire puisqu'un générateur de signal de quelques spires réglé entre 11 et 18 MHz peut produire un signal assez fort pour la commuter.

La dernière version de l'appli *Demo App* de ST permet de configurer le comportement de la sortie GPO. Scannez la carte Cloud-ST25TA avec votre ordiphone/tablette, allez sur l'option Tools de l'appli. Vous pourrez y sélectionner l'une des sept options proposées pour la sor-

tie GPO. Appuyez sur *GPO Configure* et passez sur la carte Cloud-ST25TA pour effectuer les modifications. Vous pouvez vérifier l'état de la sortie GPO en scannant à nouveau la carte Cloud-ST25TA et en lisant le système de fichiers dans l'appli. Lorsque la sortie est configurée sur SC (*State Control*), on peut utiliser *Drive GPO* pour changer l'état de la sortie. Le logiciel de développement CR95HF (pour Windows) et la carte d'émission/réception DEMO-CR95HF permettent aussi d'effectuer ces changements.

Programme d'exemple : un ordiphone NFC réveille Arduino !

J'ai écrit une petite application pour la carte Arduino Pro Mini qui illustre l'utilisation de la carte CLOUD-ST25TA dans un environnement embarqué type. Vous pouvez télécharger le code du **lis-tage 1** en [4]. Trois fils suffisent pour relier la carte CLOUD-ST25TA à la Pro Mini (**fig. 4**). Le signal de sortie GPO est connecté à la broche d'interruption INT0 (PD2) de la Pro Mini. La broche GPO peut absorber ou délivrer 0,7 mA sous 1,65 V, et 4 mA sous 5,5 V. Ces valeurs sont suffisantes pour piloter une LED à haut rendement si vous souhaitez en utiliser une. Notez que la carte CLOUD-ST25TA nécessite une alimentation externe puisqu'elle

ne contient que l'élément commutateur. Un circuit alimenté par pile devrait toujours consommer aussi peu d'énergie que possible, particulièrement en mode veille. C'est la raison pour laquelle le code place le processeur en mode *PWR_DOWN* après que la LED a clignoté cinq fois ; de tous les modes de veille possibles, c'est en effet celui qui consomme le moins. Dans son état par défaut, la sortie GPO produit un signal logique haut pour indiquer la présence d'un champ RF ; nous utilisons donc ce drapeau pour sortir de l'état de veille. Lorsqu'on approche un appareil compatible NFC de la carte CLOUD-ST25TA, le système Arduino est activé s'il est en veille. L'exemple est élémentaire, mais il peut servir de base à la construction de projets plus complexes. Nous sommes d'ailleurs impatients de découvrir vos réalisations, donc postez-les sur www.elektor-labs.com ou envoyez-les à la rédactrice en chef (redaction@elektor.fr). Pour les plus ambitieux, STMicroelectronics met à disposition le code source STSW-ST25002 et l'exécutable STSW-ST25001 [5] de l'appli Android *ST25 Demo App*. ■

(150668 - version française : Hervé Moreau)

Liens

- [1] *NFC avec Elektor et ST*, Elektor janvier/février 2016, p. 30 , www.elektormagazine.fr/150593
- [2] *Technologie NFC ST25TA de STMicroelectronics*, Elektor janvier/février 2016, p. 26, www.elektormagazine.fr/150472
- [3] Appli Demo App : www.st.com ou Google Play
- [4] Code d'exemple : www.elektormagazine.fr/150668
- [5] Code source applis ST25 et M24SR : www.st.com/web/catalog/tools/FM147/SC1871/PF262828

Listage 1. Réveil par NFC, code d'exemple

```

/*
  Réveil par NFC
  Ce code s'exécute sur une Pro Mini dont l'entrée INT0 (D2) est reliée
  à la sortie GPO de la carte CLOUD-ST25TA de STM.
  Il fait clignoter la LED 5 fois et met l'Arduino en mode veille. Lorsqu'une
  carte NFC est lue, la LED clignote 5 fois et retourne en mode veille.
*/
#include <avr/interrupt.h>
#include <avr/sleep.h>
#include <avr/power.h>
#include <avr/io.h>
int AwakeTime;                // nb de clignotement de la LED avant mise en veille
int wakePin = 2;              // utiliser INT0

void setup() {
  pinMode(13, OUTPUT);        // la LED clignotante
  pinMode(wakePin, INPUT_PULLUP); // définir D2 comme entrée d'interruption
  byte AwakeTime = 5;        // nb de clignotement avant mise en veille
}

void loop() {
  digitalWrite(13, HIGH);     // allumer la LED
  delay(100);                 // attendre 100 millisecondes
  digitalWrite(13, LOW);     // éteindre la LED
  delay(100);                 // attendre 100 millisecondes
  AwakeTime--;                // décrémenter le compteur de boucle
  if (AwakeTime <= 0) {      // vérifier le nombre de boucles exécutées
    sleepNow();              // si 5 boucles, mise en sommeil
  }
}

void sleepNow(void)
{
  sleep_enable();
  attachInterrupt(0, awakeNow, HIGH); // un champ NFC détecté met GPO à l'état haut
  delay(100);
  set_sleep_mode(SLEEP_MODE_PWR_DOWN); // mode éco « Power down »
  sleep_mode(); // au dodo !
  //----veille---attendre l'interruption de la ST25TA
  //ZZZzzz
  //----debout là-dedans !----une balise NFC a été lue
  sleep_disable();
  detachInterrupt(0);
}

void awakeNow(void)
{
  // code exécuté au réveil
  AwakeTime = 5; // initialisation du compteur et 5 coucous de plus
}

```

salon CARTES 2015 : le dernier du nom ?



Patrick GUEULLE (France)

La 30^e édition de l'évènement mondial en matière de solutions sécurisées pour le paiement, l'identification et la mobilité, préfigure un changement radical de sa dénomination : l'an prochain, il devrait s'appeler TRUSTECH. Souvenons-nous que lors de sa création en 1985, Internet et les téléphones portables n'existaient pas encore, tandis que la carte à puce n'en était qu'à ses balbutiements... que nous suivions déjà de très près !

Un triste anniversaire

Endeuillée par les tragiques attentats de novembre et entourée de pesantes mesures de sûreté, l'édition 2015 de cet incontournable salon a pâti de défections d'exposants jusqu'à présent fidèles. Bien des visiteurs habituels ont également renoncé à s'y rendre, mais nous y étions !

Au fil des années, la part des cartes à puce a progressivement diminué sur les stands (au profit des « éléments sécurisés » au sens large), tandis que dans les allées, on croise aujourd'hui davantage de décideurs que d'ingénieurs, et presque plus d'amateurs ou d'étudiants.

Il n'empêche que c'est là, plus que jamais, que l'on peut vivre des expériences qu'il serait délicat (voire illégal) de tenter dans la vie quotidienne. Surtout quand Milipol (le salon mondial de la sécurité intérieure des États) se tient dans le hall voisin, avec un point de passage bien commode entre les deux. Nous y reviendrons sous peu ! Notre moisson de cette année fut donc tout aussi riche que par le passé, mais plus compliquée à engranger...

Le plus frappant était l'absence des principaux fabricants de lecteurs de cartes : bien que le lecteur PC/SC ou CCID soit

aujourd'hui un produit de grande diffusion peu coûteux et banalisé, on peut craindre que l'objectif d'en trouver un dans chaque foyer ne soit jamais atteint.

D'ailleurs, une marche arrière semble s'amorcer pour les applications « carte » à domicile, par ex. en matière de rechargement de titres de transport en commun ou de paiement en ligne : l'un après l'autre, les navigateurs Internet cessent de supporter les *applets* Java (trop vulnérables ?), servant à faire dialoguer les sites web avec les lecteurs de cartes.

La classe HID monte en puissance !

En revanche, les cartes à puce déguisées en clefs USB s'imposent aujourd'hui dans toujours plus d'applications. Très bon marché (moins de 6 €) comme les cartes prédécoupées de Haplink (anciennement Plug'up) ou plus élaborées (micro-contrôleur Inside Secure incorporé) comme les clefs Keydo de Neowave (**fig. 1**), elles sont volontiers labellisées « FIDO U2F » : *Fast IDentity Online*, un système ouvert d'authentification forte de type PKI (infrastructure à clefs publiques) idéal pour la connexion sécurisée à des sites Internet compatibles, par « validation en deux étapes ». Cela fonctionne déjà, avec



le navigateur Chrome, sur les services en ligne Google, Gmail, Dropbox, Github, en attendant un portage (imminent ?) sur Mozilla Firefox. Et on nous promet pour bientôt des variantes sans contact !

Aujourd'hui, tout est basé sur la *classe HID* (interface homme-machine) de l'USB : pas besoin de pilote, il suffit d'enficher. Le comble, c'est que de plus en plus de lecteurs de cartes à puce se connectent à l'ordinateur (Windows, Linux, Mac, etc.) de cette façon plutôt qu'en PC/SC.

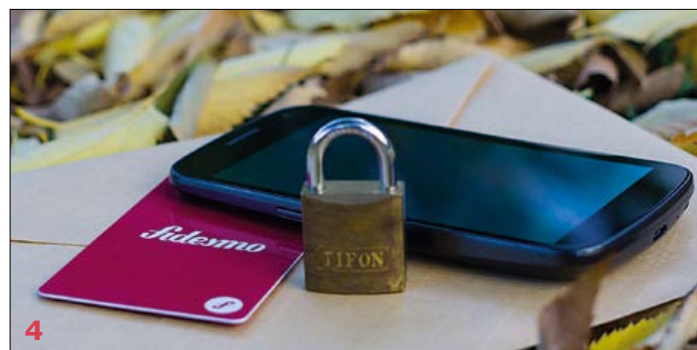
Le nouveau lecteur sans contact ACR 1281U-C2 UID d'ACS (**fig. 2**) est tout simplement vu comme un clavier USB. L'identifiant unique (UID) d'une balise RFID ISO14443 qui lui est présentée est ainsi récupérable par tout logiciel (ou site web) capable de lire une frappe de touches. L'idée n'est pas neuve (cela se faisait déjà couramment en matière de cartes magnétiques, de codes-barres, ou de balises 125 kHz), mais elle facilite spectaculairement le développement d'applications simples. Même les socles RFID pour figurines de jeux vidéo (Skylanders, Disney Infinity, etc.) se connectent aussi aux consoles par HID, ce qui interdit de s'en servir sur PC avec nos outils favoris (par ex. le kit BasicCard), car chaque protocole demeure propriétaire. C'est frustrant, mais cela stimule la curiosité !

Des idées à glaner

Sur un tel salon professionnel, on peut discuter en tête-à-tête avec des industriels qui n'ont habituellement aucun contact

avec le monde des amateurs, même avertis. Ainsi, le néerlandais SMARTRAC (l'un des sponsors du badge « visiteurs ») nous a présenté ses inserts « PRELAM » à bobinages filaires ultraplats (**fig. 3**), servant à la confection de cartes sans contact à double antenne, dont l'une alimente un voyant lumineux. Et alors même que l'on se complique traditionnellement la vie avec des circuits accordés et des doubleurs de tension pour réaliser des détecteurs de champ 13,56 MHz, on découvre qu'une simple LED « haute luminosité » peut être branchée directement aux bornes d'un bobinage apériodique de quatre ou cinq spires de la taille d'une carte de crédit !

Un salon aussi bien fréquenté que CARTES offre également une bonne occasion de rencontrer des innovateurs qui, n'ayant pas les moyens de s'offrir un stand, hantent ceux de leurs confrères. Mattias Eld, cofondateur de FIDESMO (Suède), nous a ainsi parlé (chez LEDGER) de son très séduisant projet, qui a d'ailleurs remporté un Sésame en 2014. En résumé, il s'agit d'une Java Card sans contact *spéciale* (incluant même une émulation Mifare Classic) qui peut être peuplée, *via* un téléphone compatible NFC, d'applications disponibles en ligne dans un *Card App Store* dédié (**fig. 4**). Bien entendu, tous les développeurs sont encouragés à diffuser leurs *Cardlets* par ce canal. Les plus populaires peuvent même être préchargées dans la carte lors de son achat (entre 10 et 15 €) : génération de mots de passe à usage unique (OTP) et chiffrement de messages ou de fichiers (OpenPGP). On parle alors d'une *Fidesmo Android Privacy Card*. À découvrir absolument !



La NFC « V » enfin sur les rails

Les poids lourds de la profession sont parfois tentés de devancer des spécifications sur lesquelles un consensus tarde à se mettre en place. Inside Secure, par exemple, fournissait déjà en 2009 (voir [1]) des puces PicoPass pour des démonstrations très précoces d'applications NFC (communications en champ proche). Mais il s'agissait de balises ISO15693, autrement dit de catégorie « V » comme *Vicinity* (voisinage), incompatibles avec les types 1 à 4 définis par le NFC Forum.



Inside a ainsi publié, dans le cadre de son initiative *Open NFC*, sa propre spécification d'une balise *Type 5*, précisément basée sur son produit PicoPass. Après tout, NXP n'a-t-il pas imposé ainsi ses puces Mifare Classic (note d'application AN1305), parfois rebaptisées *Type 7* ? Même démarche chez STMicroelectronics avec sa famille LRI, proche du *Type 2* (note d'application AN3408) tout comme le Mifare Ultralight (note d'application AN1303).

Cette contribution spontanée peut toutefois créer une certaine confusion, à propos de laquelle Jürgen Böhler (Comité technique du NFC Forum) nous a expliqué que les caractéristiques du type 5 fixées par le document *Open NFC* (gratuit) n'avaient aucunement servi de base à la spécification fonctionnelle (payante) de la balise *Type 5*, tout récemment publiée par le NFC Forum. En pratique, d'ailleurs, cette dernière fait autorité sur la façon dont un terminal doit s'adresser à une balise de type 5, mais ne définit pas directement son organisation interne.

Pour notre part, nous avons relevé des incompatibilités flagrantes entre les deux documents, qui ne favoriseront pas l'interopérabilité.

Affaire à suivre, en tout cas, car l'ouverture de la NFC à la technologie ISO15693 est prometteuse, dans la mesure où les balises et les lecteurs correspondants existent déjà sur le terrain : dans les médiathèques modernes, entre autres ! Pouvoir les solliciter discrètement à l'aide d'un simple téléphone Android ouvre d'étonnantes perspectives, mais révèle aussi de grosses failles de sécurité à colmater...

(150640-I)

Lien

[1] Elektor 02/2010, Le salon Cartes 2009 : www.elektormagazine.fr/091025

[2] Code du reformatage : www.elektormagazine.fr/150640

Quelques lignes de BASIC

Même dans un domaine aussi récent que la NFC « V », le langage ZCBasic de la BasicCard permet de faire beaucoup de choses avec quelques lignes de code. Ainsi, ce petit programme `NFCicode.bas`, compilé (`NFCicode.exe`) avec le kit logiciel gratuit, tente de reformater en balise NFC semblable au *Type 2* (notes d'application AN11032 et AN11042 de NXP), une étiquette équipée d'une puce ICODE-SLI. On en récupère facilement au dos de la couverture de livres réformés par bien des bibliothèques publiques :

```
#Include CARDUTIL.DEF
#Include COMMERR.DEF
Declare Command &HFF &HCA UID(S$)
Declare Command &HFF &H30 ICODE(S$)
Declare Command &HFF &HD6 UBIN(S$)
ComPort=102
REM Pour Omnikey 5321 avec driver 1.2.0.6 / XP (ou sup)
CLS:Call WaitForCard:ResetCard(S$)
:Call CheckSW1SW2
Print:Print
Call UID(P1P2=&H0000,Lc=0,S$,Le=8)
:Call CheckSW1SW2
Print "UID : ";
For F=1 To Len(S$)
C$=MID$(S$,F,1):C=ASC(C$):C$=HEX$(C)
IF LEN(C$)=1 then C$="0"+C$
Print C$;
Next F:Print:Print
S$=Chr$(1)+Chr$(2)+Chr$(0)+Chr$(0)
Call ICODE(P1P2=&H0001,Lc=4,S$,Disable Le)
S$=Chr$(1)+Chr$(6)+Chr$(0)+Chr$(0)
Call ICODE(P1P2=&H0001,Lc=4,S$,Disable Le)
S$=Chr$(1)+Chr$(1)+Chr$(1)+Chr$(1)+Chr$(1)+Chr$(1)+Chr$(1)+Chr$(1)
Call UBIN(P1P2=0,Lc=4,S$,Disable Le)
S$=Chr$(3)+Chr$(0)+Chr$(1)+Chr$(0)
Call UBIN(P1P2=&H0001,Lc=4,S$,Disable Le)
For B=2 To 27
S$=Chr$(0)+Chr$(0)+Chr$(0)+Chr$(0)
Call UBIN(P1=0,P2=B,Lc=4,S$,Disable Le)
Next B
Call WaitForNoCard
```

Ce programme a été spécialement écrit pour le lecteur Omnikey 5321, très répandu chez nos lecteurs et encore facile à trouver sur l'internet.

Même si son fabricant (HID Global) en a récemment abandonné la production, le modèle 5421 va prendre la relève. Le 5021 CL peut également convenir, mais comme il n'est pas à double interface (avec et sans contact), il faut remplacer **ComPort=102** par **ComPort=101** avant la compilation.

bienvenue dans la section **CRÉER**

Clemens Valens, labo d'Elektor

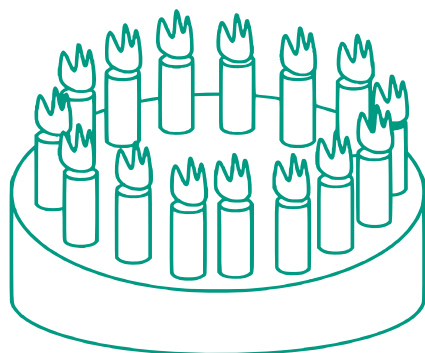
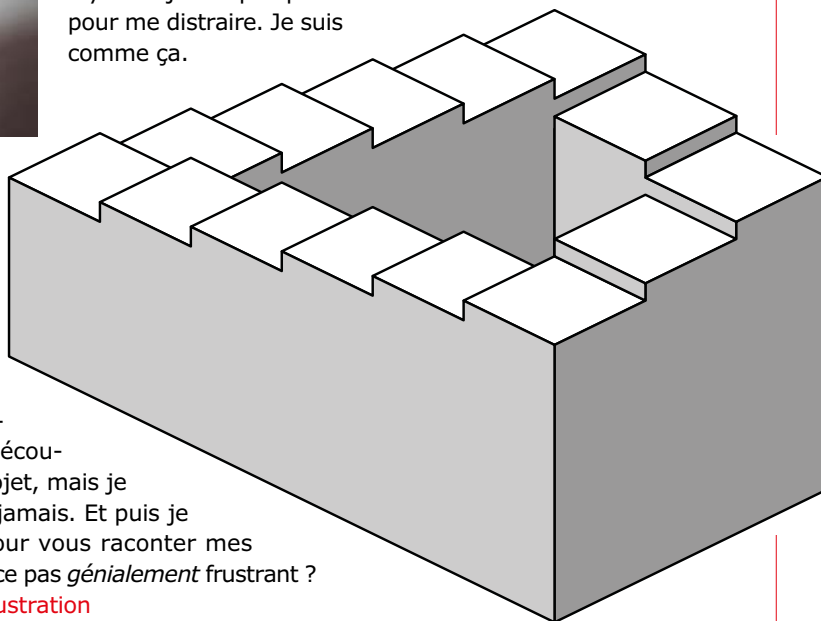


Au bord de la frustration

Ma femme et mes enfants ont du mal à le comprendre, mais je trouve mon métier formidable. Quand on est passionné d'électronique comme moi, il est même génial. Chaque jour je reçois des communiqués de presse annonçant de nouveaux produits et des technologies innovantes, je suis invité à toutes sortes de salons, et je reçois même des échantillons gratuits. « Si vous avez besoin de quelque chose, dites-le-moi » est une phrase que j'entends souvent. D'un autre côté, je trouve ma position assez frustrante. Il y aurait tant de choses que je pourrais créer avec ces produits, si seulement j'avais du temps et de l'argent. Mais voilà, je n'ai ni l'un ni l'autre. Cela dit, même si j'en avais les moyens, il est probable que je n'achèverais jamais le moindre projet. Car il y a toujours quelque nouveauté pour me distraire. Je suis comme ça.

Montrez-moi un horizon ou une porte, j'aurais envie de savoir ce qui se cache derrière. Je suis un papillon. Je volète d'une technologie à l'autre, je saute d'hyperlien en hyperlien, avec toujours le même émerveillement. Mais toujours aussi avec ce sentiment de frustration. Chaque découverte m'inspire une idée de projet, mais je sais que je ne la concrétiserai jamais. Et puis je me souviens : je suis payé pour vous raconter mes découvertes. Finalement, n'est-ce pas *génialement* frustrant ?

<https://fr.wikipedia.org/wiki/Frustration>



WIKIPEDIA15
維基百科十五歲

Viva Wikipédia !

Wikipédia a fêté ses 15 ans au début de l'année. Je trouve cette encyclopédie en ligne super, d'ailleurs je l'utilise beaucoup. Voici ce qu'elle dit sur la frustration positive : *La frustration peut être considérée comme un comportement problématique [...] Dans certains cas positifs, cette frustration s'accroît jusqu'à un niveau durant lequel le patient ne la supporte plus et tente de trouver une solution aux problèmes qui la causent.* Nombre de projets Elektor sont le fruit d'une frustration. Platino, que vous retrouverez plus loin, en est un bon exemple. Je le sais, Platino est mon projet. La carte Platino est aussi sur Wikipédia. Je le sais, c'est moi qui l'y ai mise.

Viva Wikipédia !

https://en.wikipedia.org/wiki/List_of_Arduino_boards_and_compatible_systems#Arduino_footprint-compatible_boards

(150669 – version française : Hervé Moreau)

compteur de visites pour vos pages web

avec écran LCD sans fil

Bert van Dam (Pays-Bas)

Dans cet article, nous vous montrons comment créer une page web sur un serveur Raspberry Pi et afficher le nombre de visites de cette page. En outre ces informations s'afficheront aussi sur l'écran LCD d'une carte Arduino sans fil.



Figure 1. Tablette, carte XinoRF et shield LCD.

Le matériel nécessaire pour ce projet se résume à une carte XinoRF (Arduino Uno avec module radio intégré) et un shield LCD. Comme serveur web, nous nous servons d'un nano-ordinateur Raspberry Pi avec le module *Slice of Radio*. La carte XinoRF et le module Slice of Radio font partie du kit RasWIK (*Wireless Inventors Kit for Raspberry Pi*) disponible dans l'e-choppe d'Elektor [1]. Il faut que le Raspberry Pi soit connecté à votre routeur par câble Ethernet ou Wi-Fi. N'oubliez pas que l'adresse IP est celle de votre Raspberry Pi.

Ce projet a été testé avec la carte SD du livre « Raspberry Pi - 45 applications utiles pour l'électronicien » [2] mais fonctionne très bien avec la carte SD du kit RasWIK. Les autres cartes SD ne disposent pas de réglages pour les liaisons radio et/ou les serveurs Python, il vaut mieux ne pas s'en servir.

Dans cet article, nous supposons que vous savez vous servir du Raspberry Pi, installer des logiciels et saisir des commandes. Si ce n'est pas le cas, nous vous suggérons de lire au préalable le livre mentionné ci-avant.

Le projet

Le Raspberry Pi est doté d'un serveur internet Python simple. La page d'accueil de ce serveur (*index.html*) affiche d'abord une illustration et lance ensuite automatiquement un programme Python sur votre Raspberry Pi. Le résultat des instructions de ce programme s'affiche sur la page *index.html* dans un cadre *iframe* (voir **listage 1**). Il s'agit d'un cadre qui se trouve à l'intérieur d'une page web. Vous pouvez ainsi afficher une page web à l'intérieur d'une autre. Dans notre projet, un programme écrit en Python crée cette seconde page web.

L'interface CGI (*Common Gateway Interface*) permet à la page d'accueil de lancer un programme sur le Raspberry Pi. Pour vous protéger contre tout usage abusif, le projet ne fonctionne qu'avec les programmes qui se trouvent dans un dossier spécial nommé *cgi-bin*. En outre, le propriétaire du Raspberry Pi doit donner au préalable à ces programmes des droits d'exécution. Ainsi des personnes non autorisées ne pourront pas démarrer ces programmes.

La commande HTML *viewport* au début du **listage 1** ne vous est peut-être pas familière. Elle sert à adapter la page aux formats des tablettes et ordiphones (voir **figure 1**). La largeur de l'illustration affi-

chée sur la page est de 286 pixels, nous réglons donc *viewport* sur une valeur légèrement supérieure, soit 300 pixels, pour avoir une petite bande blanche à droite de l'illustration. Vous auriez pu paramétrer *viewport* exactement sur la même largeur, mais l'utilisateur risquerait alors de se demander si une partie de la page n'est pas mangée parce qu'il n'y a pas de bordure.

Le programme *visitor.py* lancé par la page *index.html* va d'abord essayer d'ouvrir le fichier où est conservé le nombre de visiteurs. Ce fichier se trouve sur le serveur Raspberry Pi. Si ce fichier n'existe pas encore, c'est qu'il n'y a pas encore eu de visite et le compteur est mis à zéro. Si le fichier existe bien, la valeur qui s'y trouve est lue. Ensuite, la valeur est incrémentée d'un et enregistrée dans le fichier. Le texte « You are visitor number: » est ajouté devant le chiffre, cette chaîne est envoyée via la connexion sans fil à l'Arduino. Enfin, le programme crée une page HTML qui contient cette chaîne et la renvoie à la page *index.html*, où elle s'affichera dans l'*iframe*. Le **listage 2** ne contient que la dernière partie de ce programme, vous trouverez le code source complet dans les fichiers à télécharger [3].

Vous voyez que le programme « imprime » une page HTML. En réalité, tout ce qu'un programme CGI délivre n'est pas envoyé à l'écran (ou une imprimante), mais est transmis au navigateur sous la forme d'un fichier. Il affiche le fichier sous forme de page, dans ce cas au milieu de la page *index.html* dans le cadre *iframe*.

Listage 1.

```
<HTML>
<HEAD>
  <TITLE>Visitor</TITLE>
  <META NAME="viewport" content="width=300" CONTENT="initial-scale=1">
</HEAD>
<BODY>

<IMG SRC="welcome.jpg" WIDTH="286" HEIGHT="70"
ALIGN="BOTTOM" BORDER="0" NATURALSIZEFLAG="0">

<iframe name="myframe" src="cgi-bin/visitor.py" height="50"
width="100%" frameborder="0"></iframe>

</BODY>
</HTML>
```

Listage 2.

```
# show the HTML page with the
  number of visits
print "Content-Type: text/html"
print ""
<HEAD>
<TITLE>Server Counter</TITLE>
</HEAD>
<html>
<body>
%s
</body>
</html>
"" % comment
```

Le programme *visitor.ino* tourne en permanence sur la carte Arduino. Ce n'est donc pas la page *index.html* qui le lance. Ce programme initialise l'écran LCD et attend ensuite les données envoyées par radio. Ces données sont affichées sur l'écran LCD et réparties entre les lignes. Lorsque le Raspberry Pi envoie le signe tilde (~), l'écran LCD est effacé. Nous savons ainsi que le message est mis à jour. Le code source de ce programme se trouve aussi dans les fichiers à télécharger [3].

Comment procéder ?

Voici un aperçu des étapes à suivre :

1. Transférez le programme *visitor.ino* depuis votre PC sur la carte XinoRF, à l'aide de votre IDE Arduino.
2. Veillez à ce que la distance entre l'Arduino et le Raspberry Pi soit au moins de 50 cm pour la connexion sans fil. Il n'y a en effet aucun câble entre l'Arduino et le Raspberry Pi !
3. Créez un dossier pour ce projet sur votre Raspberry Pi et appelez-le *server*. Dans ce dossier, créez un sous-dossier appelé « *cgi-bin* ». Attention : vous devez utiliser la désignation *cgi-bin* (sans majuscule et avec un trait d'union), sans quoi vous ne pourrez ouvrir aucun programme à distance.
4. Copiez dans le dossier *server* les fichiers suivants : *index.html*, *favi-con.ico* et *welcome.jpg* (fichiers disponibles en [3]).
5. Copiez le fichier *visitor.py* dans le dossier *cgi-bin*. Attention : n'ouvrez pas ce fichier sur votre ordinateur, pas même pour le lire. Si

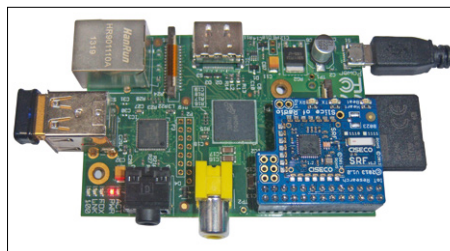


Figure 2. Raspberry Pi et module *Slice of Radio*.

vous le faites quand même, il sera sauvegardé **et** modifié, il ne pourra plus remplir sa fonction (le serveur affichera une fenêtre pour vous dire qu'il n'a pas trouvé le fichier alors qu'on sait qu'il est bien là).

6. Donnez au fichier les droits d'exécution en tapant l'instruction suivante dans le dossier *cgi-bin* :
`chmod u+x visitor.py`
 Si vous oubliez, vous verrez la page *index.html*, mais pas le compteur. Dans la fenêtre où tourne le serveur, vous verrez un message d'erreur.
7. Retournez dans le dossier *server* et lancez le serveur en tapant :
`python -m CGIHTTPServer 8080`
 Il faut le faire depuis le dossier *server*, sans quoi la page *index.html* n'apparaîtra pas.
8. Allez ensuite dans votre navigateur internet sur votre ordinateur ou votre tablette à l'adresse qui suit (n'oubliez pas de remplacer mon numéro IP par celui de votre RPi). Si vous ne vous souvenez plus de l'adresse IP de votre Raspberry Pi, allez sur votre routeur et ouvrez la page des utilisateurs. Vous y trou-

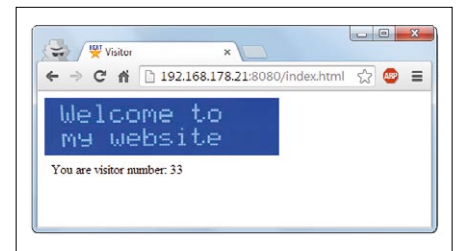


Figure 3. Page web *index.html* affichée sur un ordinateur portable.

verez votre Raspberry Pi et son adresse IP.

<http://192.168.178.21:8080/index.html>

Vous voyez maintenant une page comme celle de la **figure 3**. Chaque fois que la page est mise à jour, le compteur s'incrémente d'une unité (voir la figure 1). ◀

(150551 – version française : Eric Dusart)

Liens

- [1] www.elektor.fr/raswik
- [2] www.elektor.fr/rpi
- [3] www.elektormagazine.fr/150551

L'auteur

Bert van Dam écrit des livres, des cours et des articles consacrés aux microcontrôleurs, à l'électronique, aux ordinateurs, à l'intelligence artificielle et à la programmation.

analyseur de signal par Ethernet (1)

dsPIC33 + W5500 = oscilloscope, analyseur de spectre et générateur de signal tout-en-un

Neal Martini (États-Unis)

Si vous avez besoin d'un oscilloscope simple, d'un générateur de signal rudimentaire et d'un analyseur de spectre pratique, cet instrument trois-en-un est pour vous. L'ASPE, Analyseur de Signal Par Ethernet, est de surcroît compact et bon marché. À l'origine, je l'ai conçu comme outil d'analyse des systèmes de données échantillonnées, et comme moyen d'appréhender concrètement le potentiel et les limitations de ces systèmes.

L'ASPE (NCSA en anglais, *Network Connected Signal Analyzer*) peut numériser des signaux avec des fréquences d'échantillonnage allant jusqu'à 1 MHz. Les signaux numérisés sont ensuite affichés sur PC. Le logiciel associé à l'instrument permet aussi de produire un spectre de puissances dans le domaine fréquentiel. Les variables contrôlant la transformée de Fourier sont paramétrables, les commandes intuitives et les graphiques interactifs. L'instrument fournit également un générateur de signal analogique et numérique, ainsi que six types de fenêtrage pour la transformée de Fourier.

Description

La **figure 1** montre le synoptique des principales composantes du système ainsi que leur emplacement. Le module ASPE conditionne et amplifie le signal d'entrée, puis le transmet à un convertisseur analogique-numérique (CA/N). Le signal numérisé est ensuite envoyé au PC par liaison Ethernet. C'est sur ce même PC qu'a lieu le reste du traitement du signal et où sont affichées les courbes correspondantes. L'ASPE peut en outre produire les formes d'ondes de base : sinus, ondes carrées et triangulaires, ainsi qu'un signal « bruit ». L'onde produite est disponible sur le connecteur de sortie BNC. Vous pouvez

exploiter le signal hors instrument, ou l'envoyer à l'analyseur pour visualiser son chronogramme et sa représentation fréquentielle. Le logiciel sur PC crée l'interface utilisateur (UI) où se paramètre l'instrument. Les graphiques des domaines fréquentsiels et temporels sont eux aussi affichés dans l'UI. Le traitement numérique du signal (fenêtrage, FFT, mise à l'échelle, calculs des puissances) est de même effectué par le logiciel.



Caractéristiques

- Oscilloscope, analyseur de spectre et générateur de signal, **trois-en-un**
- Fréquences d'échantillonnage jusqu'à 1 MHz
- Sous-échantillonnage
- Signal d'entrée max. : 0 dBm (0,225 V_{eff})
- Sensibilité : -80 dBm (22,5 μV_{eff})
- Connexion Ethernet
- Code source ouvert

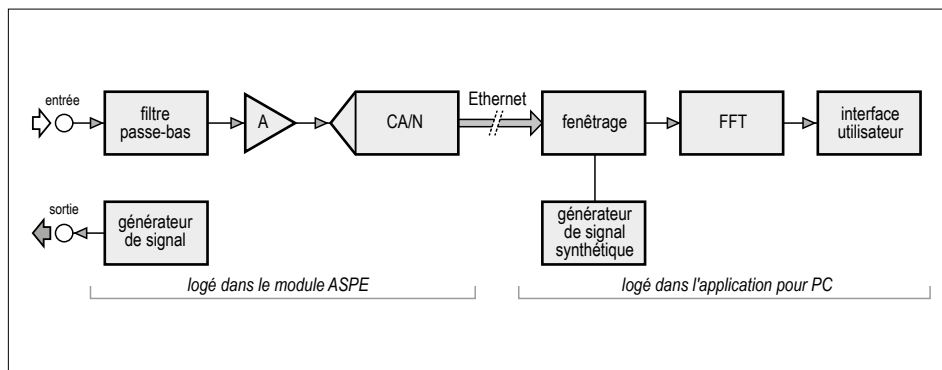
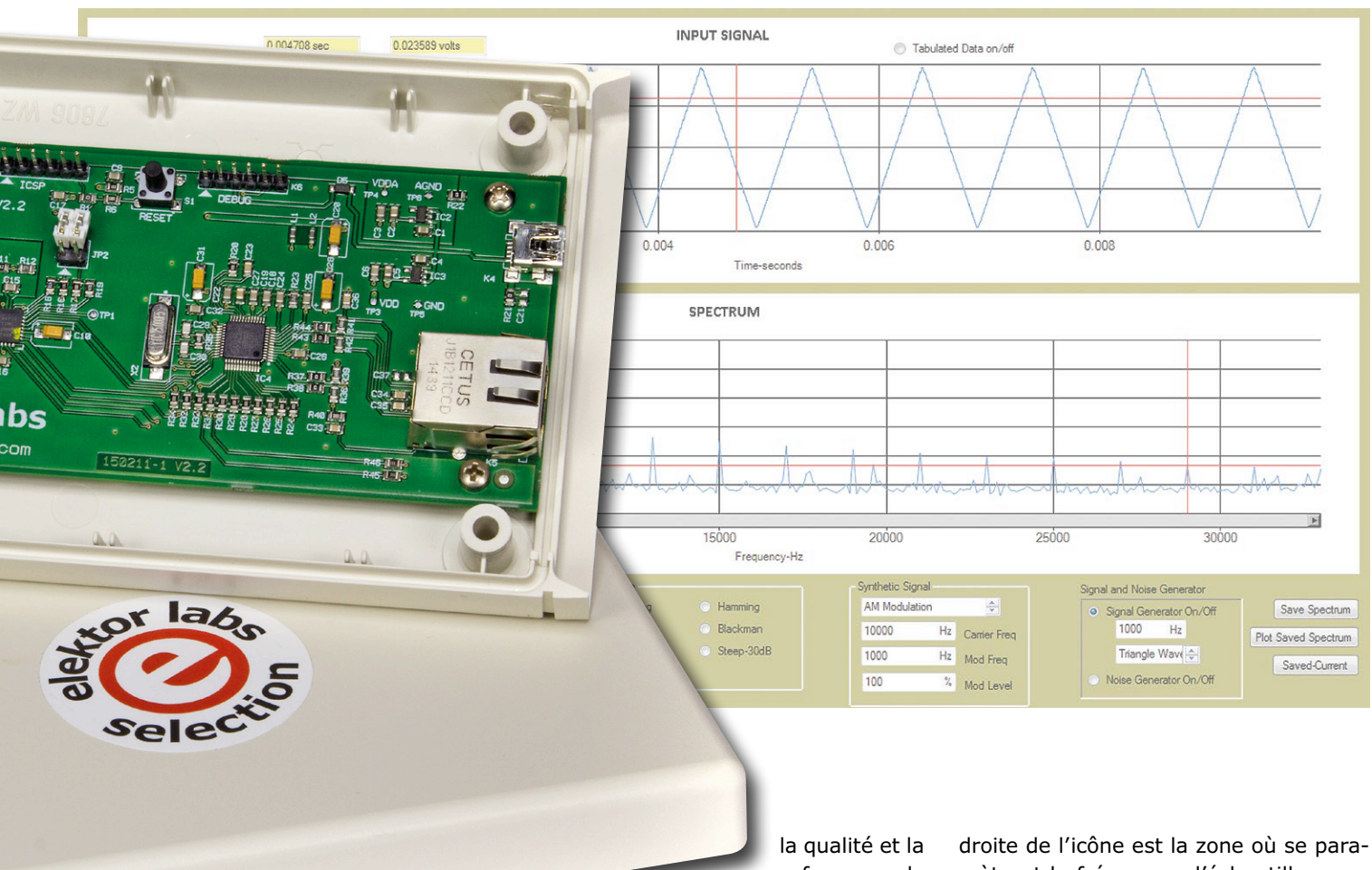


Figure 1. Diagramme fonctionnel de l'ASPE.



la qualité et la performance de l'échantillonnage et de la transformée de Fourier.

L'icône du coin inférieur gauche montre si l'analyseur est connecté au PC via le réseau local, ainsi que son adresse IP. La boîte de commande *FFT Controls* à

droite de l'icône est la zone où se paramètrent la fréquence d'échantillonnage (F_s), le nombre de points à inclure dans la FFT (N), la résolution du CA/N (*ADC bits*) et le nombre de spectres consécutifs à utiliser pour la moyenne (*Averages*).

Un clic sur le bouton *single FFT Real Data* déclenche l'acquisition de N échantillons

L'application fournit un second générateur, le générateur de signal synthétique. Il produit des signaux numériques qui peuvent être passés à la moulinette FFT (*Fast Fourier Transform*, transformation de Fourier rapide). Cela permet d'utiliser le logiciel sur PC sans que le module ASPE ne soit connecté. Les signaux synthétiques peuvent être un signal modulé en amplitude (AM), un signal modulé en fréquence (FM), une onde carrée partielle, ou un signal de test. La fréquence et la modulation sont réglables. Les signaux synthétiques sont dépourvus de bruit, donc très pratiques pour l'expérimentation. J'expliquerai plus tard comment ajouter des options supplémentaires au générateur de signal synthétique.

L'interface utilisateur

L'interface du programme (fig. 2) a été conçue pour que l'utilisateur bénéficie d'un maximum de souplesse dans le contrôle des paramètres qui affectent de façon critique les systèmes de données échantillonnées, et qu'il voit leur effet sur

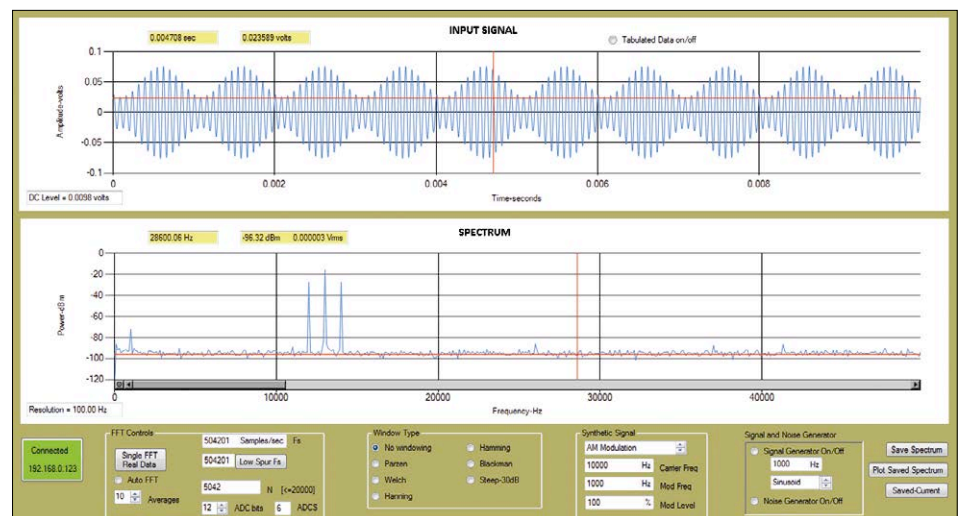


Figure 2. Tracés obtenus avec l'entrée de l'ASPE reliée à un générateur de signal qui délivre un signal modulé en amplitude (AM). La fenêtre du haut montre la représentation temporelle du signal, celle du bas son spectre de fréquences. On distingue nettement la porteuse de 13 kHz et les bandes latérales de 1 kHz.

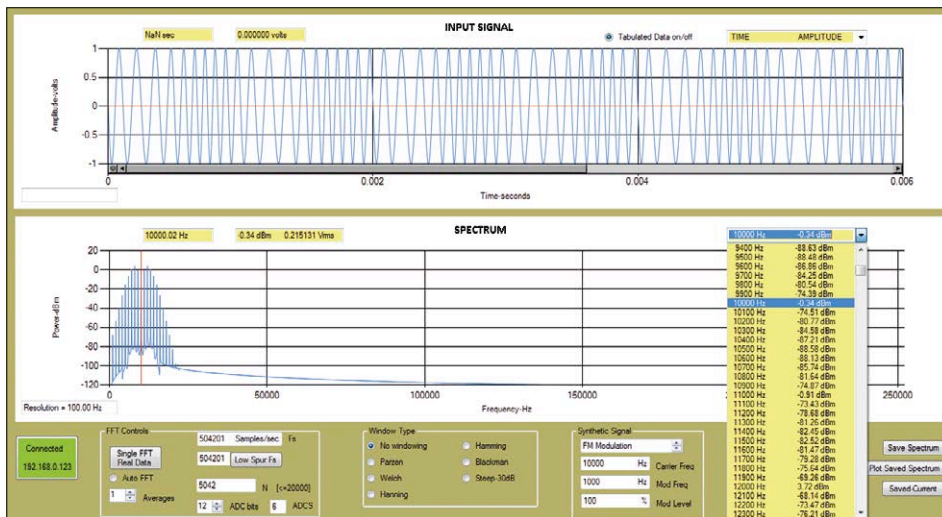


Figure 3. Le signal analysé ici est un signal modulé en fréquence (FM) « idéal », produit par le générateur de signal synthétique du logiciel. Le chronogramme du haut montre la modulation de la porteuse. Le tracé du dessous est le spectre de puissances correspondant. On distingue clairement la fréquence centrale, l'espacement entre les lignes et la largeur de bande qui caractérisent le signal FM.

du CA/N et une FFT. Le domaine temporel du signal est représenté dans la fenêtre du haut, le spectre de fréquences dans la fenêtre du bas. J'expliquerai plus loin comment les paramètres de la boîte *FFT Controls* agissent et interagissent, de même que le rôle de *ADCS* et de *Low Spur Fs*. Si on active le bouton *Auto FFT*, le système effectue l'échantillonnage et la transformation des données en continu. Comme dit plus haut, le sélecteur *Averages* permet de choisir le nombre de spectres consécutifs à inclure dans la moyenne afin de réduire la variance du bruit. La boîte de commande qui jouxte *FFT Controls* est la *Windows type*. Elle permet de choisir entre sept types de « fenêtres » pour le lissage des données précédant la FFT. Là encore je reviendrai plus tard sur le pourquoi et le comment de ces fenêtrages.

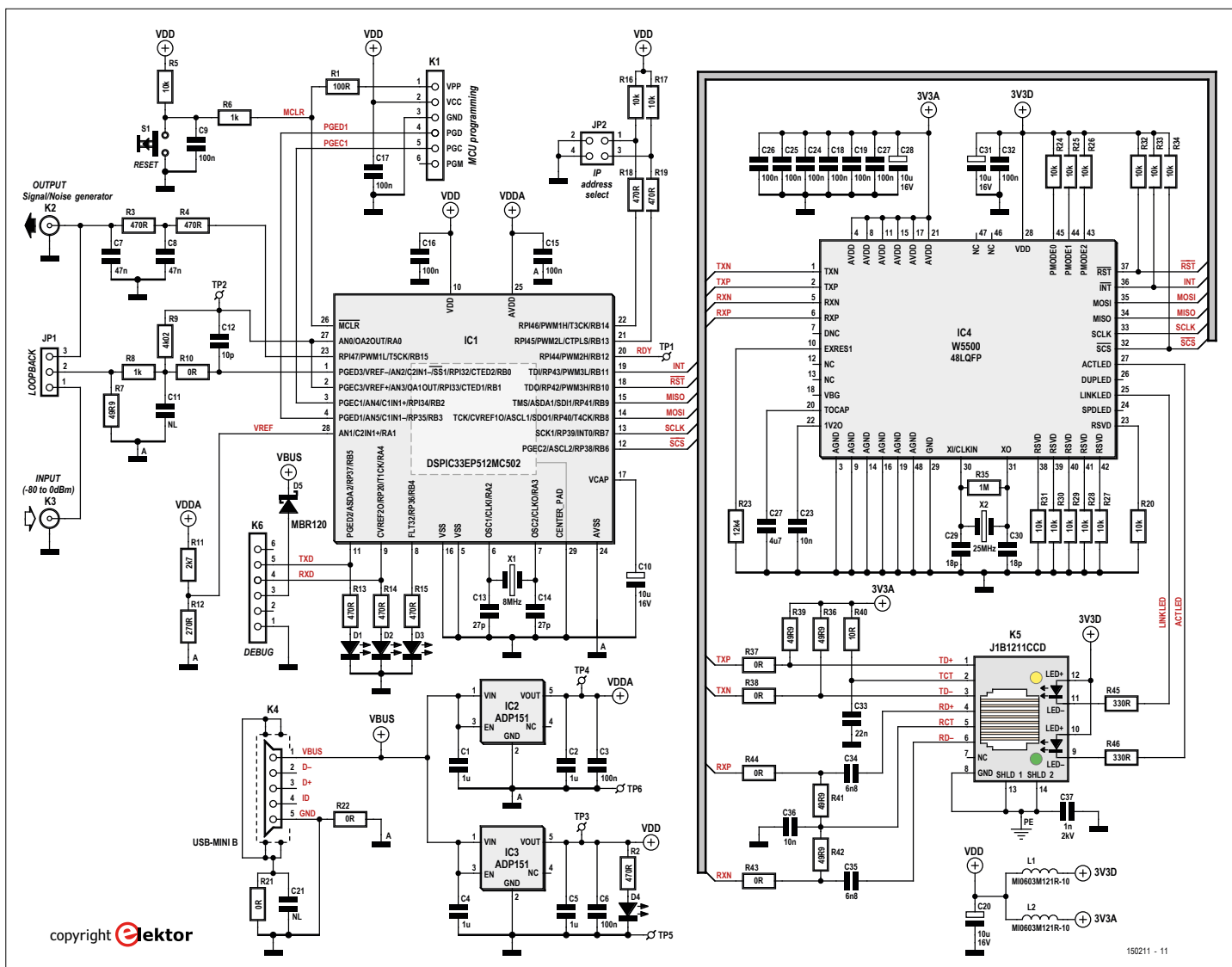


Figure 4. Schéma de l'ASPE



Le zoom multi-échelle permet un niveau de détail impressionnant

Vient ensuite l'interface *Synthetic Signal*. Elle permet de produire des tableaux de nombres pour les ondes synthétiques AM, FM et carrée, ainsi qu'un signal de test spécifique servant à vérifier le système. Lorsqu'on clique sur l'élément supérieur de cette boîte, une FFT est effectuée sur le type de signal sélectionné ; les valeurs de fréquence et de modulation du signal sont celles des autres paramètres sélectionnés.

La dernière boîte de commande est *Signal and Noise Generator*, où l'on peut sélectionner l'onde carrée, triangulaire, sinusoïdale ou bruitée que produira le microprocesseur de l'ASPE. Le signal analogique produit est disponible sur la sortie BNC K2. Il peut être envoyé à une application externe ou à l'entrée de l'analyseur.

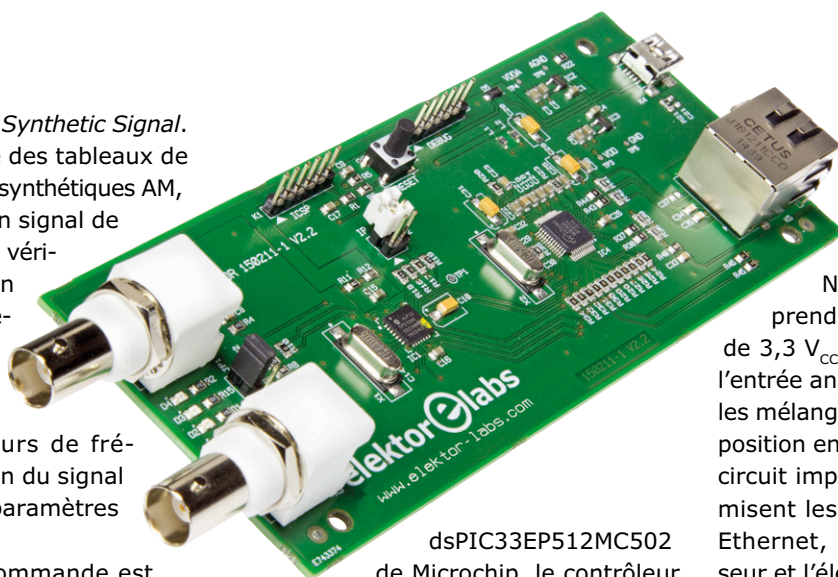
Les boîtes jaunes situées au-dessus des graphiques montrent les valeurs qui correspondent à la position du curseur, soit le temps et l'amplitude pour la fenêtre supérieure, la fréquence et les niveaux de puissance pour la fenêtre inférieure. Le bouton radio *Tabulated Data* permet d'afficher tous les points représentant le temps et la fréquence sous forme de tableau. Ce type d'affichage est activé sur la **figure 3**.

Enfin, l'UI dispose d'une fonction de zoom très puissante, comme vous pouvez le constater sur la fenêtre *Spectrum* de la figure 2. Ce zoom multi-échelle permet de visualiser les tracés avec un niveau de détail impressionnant.

Notez que l'application PC peut être chargée et lancée sur un PC sans que l'ASPE n'y soit nécessairement relié. Dans ce cas l'accès aux données réelles de l'ASPE est bien sûr impossible, mais on peut utiliser les commandes de l'interface utilisateur ainsi que le générateur de signal synthétique pour se familiariser avec le fonctionnement de l'instrument.

Matériel

Le schéma du module ASPE est reproduit sur la **figure 4**. Les principaux composants du circuit sont le



dsPIC33EP512MC502 de Microchip, le contrôleur Ethernet W5500 de WIZnet, et les régulateurs de tension à faible tension de déchet et faible bruit ADP151 d'Analog Devices.

Le dsPIC33 a été choisi pour plusieurs raisons. L'une d'entre elles est qu'il peut exécuter 60 MIPS (à l'aide de la PLL interne), autrement dit ce PIC est bien adapté aux applications CA/N à grande vitesse. Une autre est que ses 50 Ko de RAM permettent de stocker de larges blocs de données, donc évitent le recours à une mémoire externe. Son amplificateur intégré plaide aussi en sa faveur, puisque tout système d'échantillonnage nécessite une interface analogique. Pour la précision de la fréquence, on utilise un quartz externe de 8 MHz.

Le dsPIC33 a aussi l'avantage d'offrir les périphériques de qualité attendus pour une telle application : des temporisateurs/compteurs (*timers*) pour le contrôle précis de la production des signaux et de l'échantillonnage, une fonction MLI (*PWM*) pour la production des signaux et du bruit, un module SPI pour la communication avec le W5500, un CA/N à 10/12 bits rapide pour les conversions A/N, et un amplificateur opérationnel pour le filtrage anti-repliement et l'amplification en entrée.

La puce W5500 établit la communication Ethernet avec le PC. Elle est commandée par le port SPI du dsPIC33 au moyen de quelques lignes de commande (*RDY* et *Reset*). L'interface électrique entre le contrôleur W5500 et le port Ethernet est fournie par le connecteur Ethernet

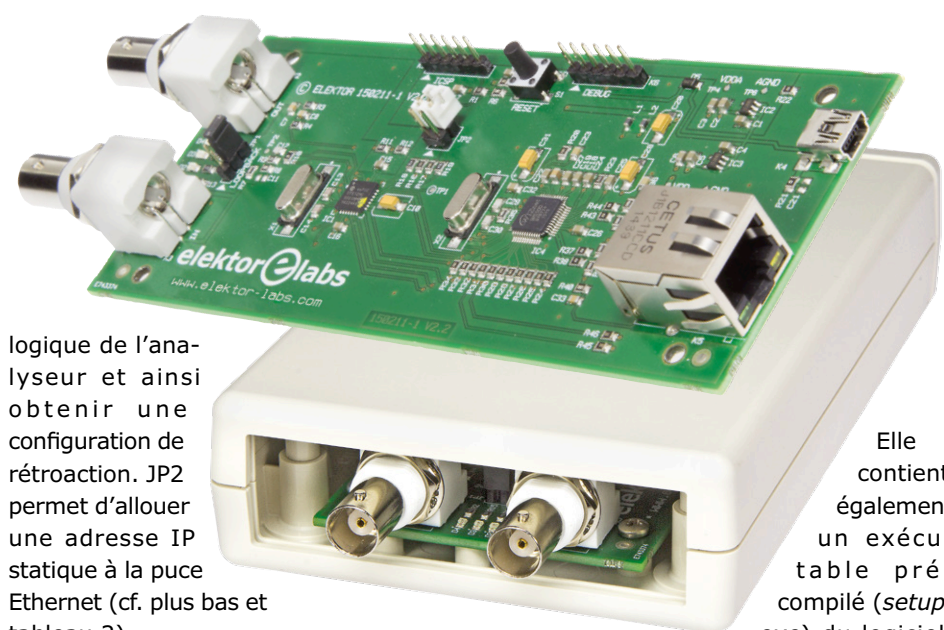
J1B1211CCD ; il contient tous les transformateurs nécessaires à une connexion directe au réseau.

Notez que le circuit comprend deux régulateurs de tension de 3,3 V_{CC} séparés. Leur présence isole l'entrée analogique du CA/N et minimise les mélanges de bruit numérique. La disposition en étoile des plans de masse du circuit imprimé ainsi que L1 et L2 minimisent les couplages entre les signaux Ethernet, les signaux du microprocesseur et l'électronique de l'étage d'entrée analogique.

Le régulateur ADP151 a été choisi pour son très faible facteur de bruit (9 μV_{eff}) et sa très faible tension de déchet (140 mV). Un bruit faible garantit un signal pur sur le CA/N, tandis qu'une faible tension de déchet permet de se passer de radiateur. Un ADP51 a besoin d'un filtrage externe très simple pour fonctionner. L'entrée de 5 V pour les régulateurs est délivrée par un connecteur mini-USB standard. Notez que les résultats de l'instrument seront meilleurs si le câble mini-USB est relié à un adaptateur USB autonome plutôt qu'à un port USB du PC. L'ASPE est en effet un instrument très sensible, capable de détecter des signaux de quelques microvolts, et dans certains cas le port USB du PC peut introduire des parasites inférieurs à -80 dBm dans le spectre de sortie. Le chargeur de mon iPhone, pour prendre mon exemple, élimine ces parasites de très bas niveaux.

Lorsqu'on branche l'analyseur sur un port USB, le dsPIC33 s'initialise et est prêt à être utilisé. On peut aussi initialiser le μC avec le bouton *reset* (S2) de la carte. Parmi les autres connecteurs de la carte figurent un connecteur de programmation pour PIC (K1), une prise BNC (K3) pour appliquer des signaux analogiques, une sortie BNC (K2) pour le générateur de signal du dsPIC33, et un port d'E/S série (K5) pour le débogage. Deux cavaliers sont présents. JP1 (*loopback*) permet de relier directement le générateur de signal du dsPIC33 au nœud d'entrée ana-

▶ Voir $22,5 \mu V_{eff}$? Pas de problème.



logique de l'analyseur et ainsi obtenir une configuration de rétroaction. JP2 permet d'allouer une adresse IP statique à la puce Ethernet (cf. plus bas et tableau 2).

Il y a quatre LED sur la carte. La LED4 indique que l'analyseur est sous tension. LED3 signale une connexion Ethernet (clignotement rapide) ou une connexion du PC vers l'ASPE (clignotement lent). LED2 indique un transfert de données, et LED1 sert pour le débogage.

Logiciel et micrologiciel

Vous pouvez télécharger le micrologiciel et le logiciel pour PC depuis le site d'Elektor (archive 150211-11.zip en [1]). Le micrologiciel a été écrit en C avec l'EDI MPLAB-X (version gratuite de Microchip). Le programme pour PC a été écrit en C# avec l'EDI Visual Studio 2010 Express (version gratuite de Microsoft). L'archive contient les codes sources C et C# complets, vous pouvez donc ajouter des fonctions au système si vous le souhaitez.

Lorsque vous l'exécutez, l'application *WindowsFormsApplication1.exe* s'installe automatiquement. Vous devez avoir le *framework* Microsoft .NET 4.0 (ou sup.) installé sur votre PC. Si vous ne l'avez pas, vous pouvez le télécharger depuis le site de Microsoft ; il est inclus par défaut dans Visual Studio 2010 Express.

Échantillonnage

Je vais maintenant décrire le système de numérisation de l'ASPE (fig. 5) de façon à ce que vous compreniez le traitement du signal numérique sous-jacent et que vous l'utilisiez correctement. Voyons d'abord comment est commandé le CA/N (ADC). La fréquence d'horloge du CA/N s'obtient en divisant celle du contrôleur (60 MHz) par une constante (ADCS + 1). La valeur ADCS peut être choisie depuis l'UI, mais avec quelques restrictions. Timer3 est utilisé pour déclencher le démarrage d'une

conversion A/N. La période d'horloge du CA/N est notée T_{AD} dans la fiche technique du dsPIC, et elle est calculée ainsi :

$$T_{AD} = (ADCS + 1) / 60 [\mu s]$$

Pour le CA/N configuré en 10 bits, la fiche technique du dsPIC indique 75 ms comme valeur minimale de T_{AD} , et 117 ns pour le CA/N configuré en 12 bits. Ces valeurs entraînent pour ADCS une valeur minimale de 4 pour un CA/N sur 10 bits, et de 6 pour un CA/N sur 12 bits. En pratique, on peut exploiter le CA/N à une cadence beaucoup plus rapide que ça si on est prêt à sacrifier un peu de précision pour l'amplitude. Sur la base d'une durée de $15 T_{AD}$ nécessaire à une conversion A/N à 10 bits, et d'une durée de $17 T_{AD}$ pour 12 bits, le **tableau 1** montre les possibilités de l'analyseur pour différentes valeurs d'ADCS. Si vous utilisez des valeurs d'ADCS ou des fréquences d'échantillonnage qui font tourner le CA/N hors de sa cadence nominale, un indicateur jaune s'allume dans l'UI. Les valeurs du tableau garantissent un fonctionnement efficace de l'ASPE, avec une dégradation minimale des performances. Notez bien que la fréquence d'échantillonnage doit être inférieure à 500 000 Hz lorsque le générateur de signal de l'ASPE est activé, sinon le microprocesseur du générateur n'a pas le temps de s'actualiser et des erreurs apparaissent dans le spectre de fréquences.

En effet les déclenchements de Timer3 ne doivent pas se produire au milieu d'un cycle d'horloge du CA/N, car un déclenchement qui survient entre deux fronts d'horloge n'est pas suivi d'effet avant la fin du cycle T_{AD} , ce qui entraîne une fluctuation des temps de démarrage du CA/N. On peut éviter cette situation en choisissant pour la fréquence d'échantillonnage des multiples entiers des cycles T_{AD} . Les deux équations utilisées pour garantir cette relation sont :

$$F_{s(10bit)} = 1 / (12 + 3 + K) \times (T_{AD})$$

$$F_{s(12bit)} = 1 / (14 + 3 + K) \times (T_{AD})$$

avec F_s la fréquence d'échantillonnage (« s » pour *sampling*), K un nombre entier, 12 et 14 les nombres de cycles T_{AD} requis pour effectuer la conversion A/N (A/D), et 3 la valeur qui prévoit le retard le plus défavorable entre le déclenche-

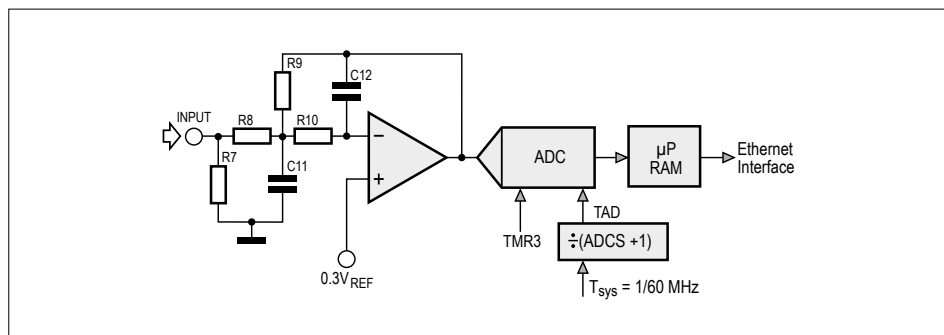


Figure 5. Le système de numérisation de l'ASPE.

ment de Timer3 et le démarrage effectif du CA/N. Lorsqu'on sélectionne une fréquence F_s dans l'IU, l'application calcule la valeur F_s la plus proche à utiliser pour éliminer les fluctuations. C'est cette valeur qui est utilisée lorsqu'on appuie sur le bouton *Low Spur Fs* de la

Tableau 1. Fréquences d'échantillonnage max. en fonction d'ADCS

ADCS	T_{AD} [ns]	F_s max. [kHz] (10 bits)	F_s max. [kHz] (12bits)
3	67	1000	882
4	83	800	705
5	100	666	588
6	117	571	504

Liste des composants

Résistances

Défaut : CMS 0805, 5 %, 0,1 W
 R10, R21, R22, R37, R38, R43, R44 = 0 Ω
 R40 = 10 Ω
 R7, R36, R39, R41, R42 = 49,9 Ω , 1 %
 R1 = 100 Ω
 R12 = 270 Ω
 R45, R46 = 330 Ω
 R2, R3, R4, R13, R14, R15, R18, R19 = 470 Ω
 R6, R8 = 1 k Ω
 R11 = 2,7 k Ω
 R9 = 4,2 k Ω , 1 %
 R5, R16, R17, R20, R24, R25, R26, R27, R28, R29, R30, R31, R32, R33, R34 = 10 k Ω
 R23 = 12,4 k Ω , 1 %
 R35 = 1 M Ω

Condensateurs

Tous CMS 0805
 C12 = 10 pF
 C29, C30 = 18 pF
 C13, C14 = 27 pF

C37 = 1 nF, 1000 V
 C34, C35 = 6,8 nF
 C23, C36 = 10 nF
 C33 = 22 nF
 C7, C8 = 47 nF
 C3, C6, C9, C15, C16, C17, C18, C19, C22, C24, C25, C26, C32 = 100 nF
 C1, C2, C4, C5 = 1 μ F
 C27 = 4,7 μ F
 C10, C20, C28, C31 = 10 μ F 16 V, tantale
 C11, C21 = non montés

Inductance

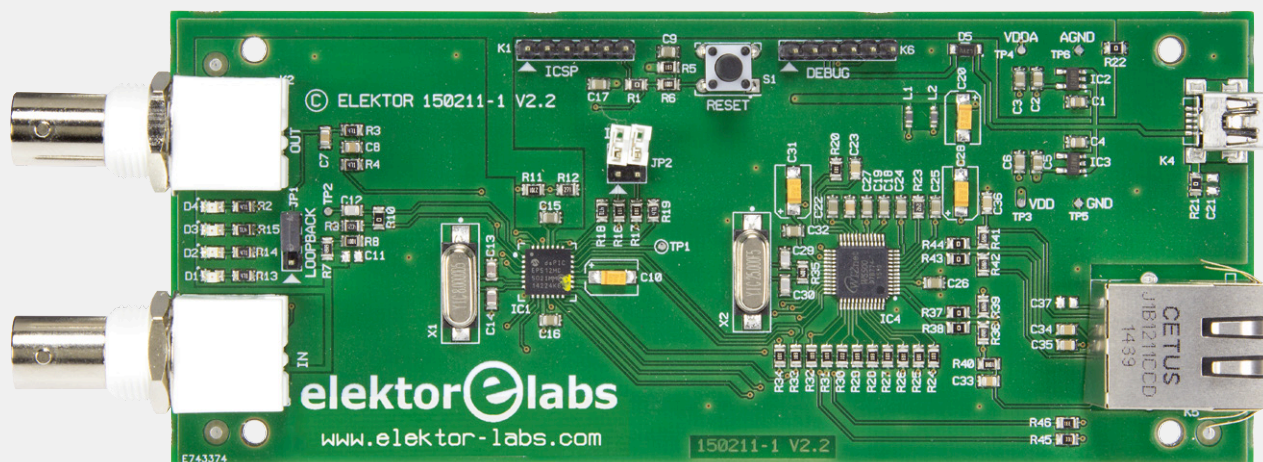
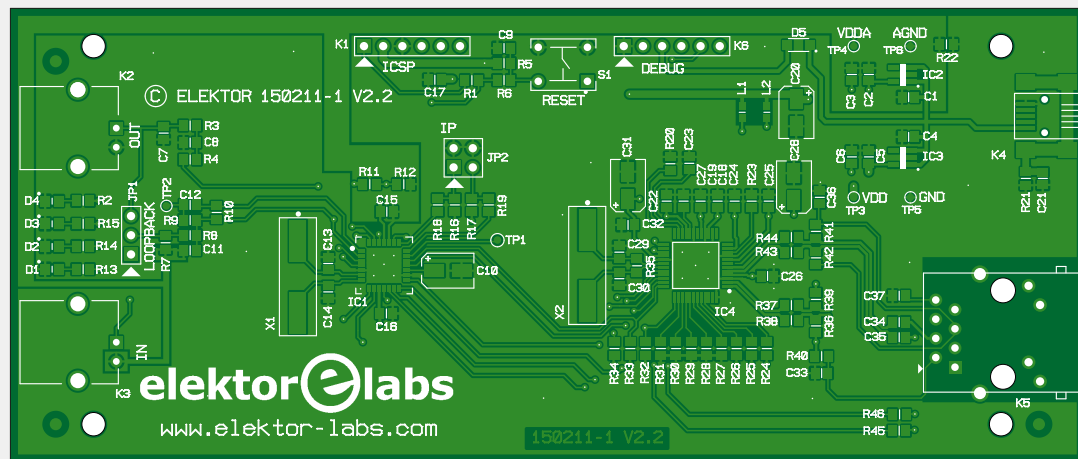
L1, L2 = ferrite, 600 Ω , 1 A

Semi-conducteurs

D1, D2, D3, D4 = LED, rouge
 D5 = MBR120LSFT1G
 IC1 = DSPIC33EP512MC502-I/MM
 IC2, IC3 = ADP151
 IC4 = W5500 (WIZnet)

Divers

JP1 = barrette mâle à 3 broches, pas de 2,5 mm, verticale
 JP2 = barrette mâle à 4 broches (2 x 2), pas de 2,5 mm, verticale
 K1, K6 = barrette mâle à 6 broches, pas de 2,5 mm, verticale
 K2, K3 = prise BNC, angle droit
 K4 = connecteur mini-USB type B, blindé
 K5 = connecteur magnétique RJ45 avec Ethernet 10/100 Base TX et LED
 S1 = poussoir, 6 x 6 mm
 X1 = quartz 8 MHz
 X2 = quartz 25 MHz
 Cavaliers pour JP1 et JP2
 Boîtier : OKW Shell-type Cases O 155, vers. I (modèle coquille), 150211-71 ((www.elektor.fr)
 Circuit imprimé nu, 150211-1 (www.elektor.fr)
 Module assemblé, 150211-91 (www.elektor.fr)



boîte *FFT Controls* afin de minimiser les instabilités.

Jetons à présent un œil sur les composants de l'étage d'entrée analogique. Nous y trouvons un filtre anti-repliement ainsi qu'un amplificateur. L'amplificateur opérationnel est interne au dsPIC33, les composants passifs sont externes. Le filtre est à rétroactions multiples, j'ai utilisé l'outil *FilterLab 2.0* de Microchip pour calculer les valeurs de ses composantes. J'avais d'abord conçu un filtre passe-bas à deux pôles avec une fréquence de cou-

pure de 500 000 Hz, ce que reflète le dessin du CI, mais j'ai réalisé assez tôt que l'amplificateur opérationnel du dsPIC33 avait un produit gain-bande de 6 MHz.

Cela signifiait que les signaux de fréquences supérieures à la fréquence d'échantillonnage maximale du CA/N pouvaient aisément traverser l'amplificateur, quoiqu'avec des amplitudes légèrement atténuées. Donc, si on augmente la fréquence de coupure du filtre anti-repliement, on peut voir des signaux de fréquences bien plus élevées que la

fréquence d'échantillonnage du CA/N. Cet accès aux fréquences supérieures est utilisé dans le sous-échantillonnage (que j'expliquerai plus tard) et permet de tirer avantage du repliement. J'utilise donc actuellement un filtrage très doux pour avoir cette bande passante large, et pour cela j'ai éliminé le second pôle du filtre. R10 vaut donc zéro ohm, et C11 n'est pas chargé. J'ai en outre déplacé le pôle à rétroactions multiples qui restait, de façon à ce que les signaux de fréquences supérieures puissent passer dans le CA/N. C12 est un condensateur de 10 pF qui place ce pôle près de 4 MHz. Parmi les résistances qui entourent l'amplificateur opérationnel, R7 (50 Ω) fixe l'impédance d'entrée utilisée dans la plupart des analyseurs de spectre. Le gain de l'amplificateur est :

$$R9 / (R8 + R7) = 4000 / (1000 + 50) = 3,809$$

La référence de 0,3 V polarise l'entrée jusqu'à une valeur d'environ 1,44 V, ce qui permet d'appliquer des signaux bipolaires sur l'entrée.

Avant de conclure, quelques mots à propos des niveaux minimal et maximal que l'ASPE peut traiter.

L'entrée maximale est déterminée par le point au-delà duquel l'amplificateur d'entrée et le CA/N commencent à saturer. Avec le gain de 3,809 et la polarisation de 1,44 V, la saturation se manifeste dans le spectre à 0,225 V_{eff} (0 dBm). Ce n'est pas vraiment une limitation, car pour travailler avec des signaux plus forts on peut toujours ajouter un atténuateur simple à l'étage d'entrée.

Beaucoup plus important est le niveau minimal acceptable. Pour comprendre ce qui le définit, faisons un petit détour par le bruit de fond de la FFT, autrement dit le plus petit niveau de signal observable dans le spectre de Fourier. Sa valeur théorique est donnée par :

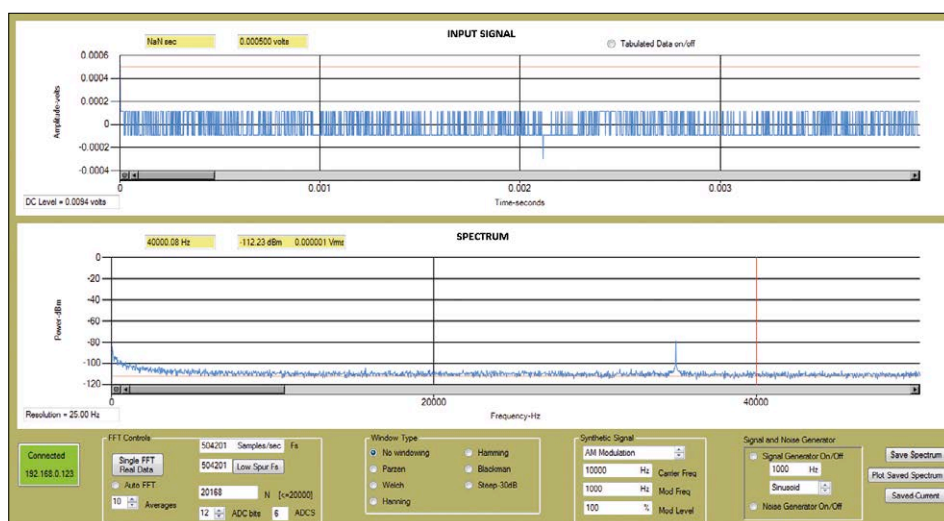


Figure 6. Le gain de traitement apporté par la transformation de Fourier permet d'observer des signaux de très bas niveaux. Sur le tracé du bas, on peut ainsi observer un signal de 35 kHz et -80 dBm (22,5 μV_{eff}) ! Notez a contrario le peu d'informations que fournit le chronogramme pour ce mini-signal.

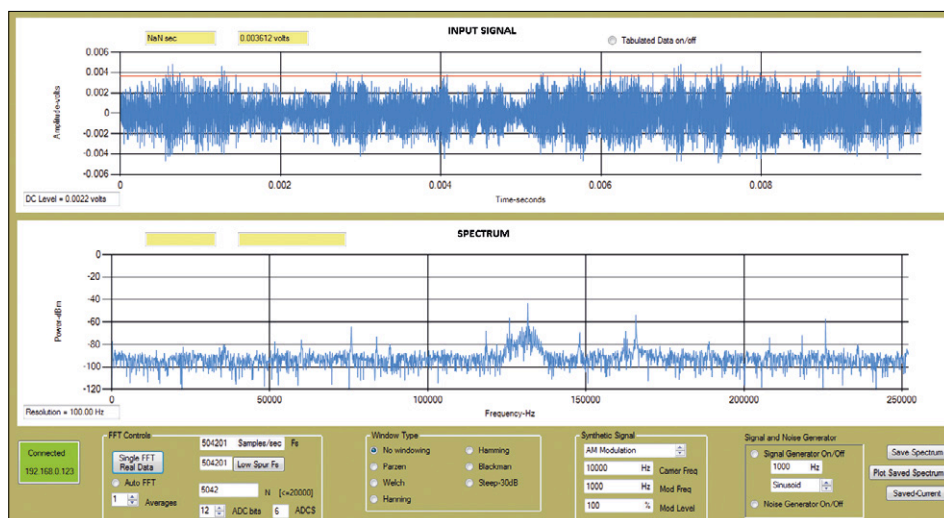


Figure 7. Ici j'ai utilisé une technique appelée sous-échantillonnage pour analyser le signal de 1140 kHz émis par une station radio AM située près de chez moi (Boise, Idaho). J'ai relié à l'entrée de l'ASPE un fil de 3 m servant d'antenne, et je n'ai échantillonné qu'à environ 500 kHz, bien en dessous de ce que vous pensiez peut-être nécessaire. J'expliquerai dans le prochain article pourquoi ce signal est centré sur 131,598 kHz.

Tableau 2.
Adresses IP statiques de W5500

JP2 broche 3	JP2 broche 1	IP allouée
GND	GND	192.168.1.123
GND	ouvert	192.168.0.122
ouvert	GND	192.168.2.123
ouvert	ouvert	192.168.0.123

bruit de fond =
S/B du CA/N + gain de traitement

Le rapport signal sur bruit (S/B) du CA/N dépend de la résolution (nombre de bits) du CA/N et vaut $6,02 \times (\text{nombre de bits}) + 1,76$ dB.

Le gain de traitement dû à la FFT (*gain processing*) dépend du nombre d'échantillons dans la FFT et vaut :

$$10 \log_{10}(\text{nombre d'échantillons} / 2) \text{ [dB]}$$

Donc nous obtenons avec p. ex. un CA/N à 12 bits et une FFT de 4096 échantillons :

$$\text{bruit de fond FFT} = (6,02 \times 12 + 1,76) + 10 \log_{10}(4096 / 2) = 107 \text{ dB}$$

Cette valeur est purement théorique : à ce bruit de fond s'ajoutent en pratique le bruit de fond des composants, les couplages parasites entre signaux, le bruit des pistes du CI, etc. Sur les exemples de cet article, la région du spectre allant de -80 dBm à -100 dBm est baignée de signaux parasites, alors que les tracés sont nets au-dessus de -80 dBm. C'est la raison pour laquelle le niveau minimal du signal d'entrée de l'ASPE devrait être de -80 dBm (**fig. 6**). Et c'est plutôt pas mal pour un circuit aussi simple !

Démarrage rapide

Pour utiliser l'ASPE, connectez-le simplement à un réseau local via un routeur, et ouvrez l'application PC de l'archive, elle tentera de détecter l'ASPE sur l'une des adresses IP allouées.

À la mise sous tension de l'instrument ou après une initialisation, le microprocesseur attribue au W5500 l'adresse IP statique qui correspond à la configuration du cavalier JP2 (**tableau 2**). Les quatre adresses IP du tableau couvrent la plupart des modèles actuels de routeurs auxquels l'ASPE pourrait être connecté. Pour que la connexion Ethernet puisse être établie, vous devez sélectionner l'adresse IP dont les trois premiers champs correspondent à ceux du routeur auquel l'ASPE est relié. Au cas peu probable où votre routeur serait trop exotique pour ne pas être adressable par ce moyen, un simple changement du code corrigerait la situation.

Notez que vous devez redémarrer l'ASPE chaque fois que vous modifiez la configuration de JP2. Pensez également à mettre l'instrument sous tension avant de lancer l'application PC.

Dans le prochain épisode

Je vous ai décrit le matériel et l'utilisation de l'Analyseur de Spectre Par Ethernet. Dans le prochain article, je vous dévoilerai la mécanique logicielle et les dessous mathématiques de l'instrument, connaissances qui vous permettront d'utiliser au mieux votre analyseur (**fig. 7**), mais aussi de le modifier et de l'adapter à vos besoins. *ASPE plus !* ◀

(150211 – version française : Hervé Moreau)

Lien

[1] www.elektormagazine.fr/150211

AGENDA

				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

mars 2016

Mars 2016

♦ 6^e conférence internationale sur l'énergie photovoltaïque

07 au 09/03 – Chambéry
www.siliconpv.com/home.html

♦ JEC World (matériaux composites)

08 au 10/03 – Paris
www.jecomposites.com

♦ Salon Educatec Educative Salon professionnel de l'Éducation

09 au 11/03 – Paris
www.educatec-educative.com

♦ NIDays

10/03- Paris

*Nous y serons ! Nous serons
heureux de vous accueillir
sur notre stand.*

<http://france.ni.com/nidays>

NIDays

♦ Salon du livre

17 au 20/03 – Paris
www.salondulivreparis.com

♦ 26^e Semaine de la presse et des médias dans l'école®

21 au 26/03 – partout en France

*Le magazine Elektor sera distribué
gratuitement dans les établissements
qui en ont fait la demande.*

www.clemi.fr



♦ Cloud Computing World Expo

23 au 24/03- Paris
www.cloudcomputing-world.com

♦ Salon Systèmes embarqués / M2M / Objets connectés / Affichage

23 au 24/03 – Paris
www.salons-solutions-electroniques.com

♦ Salon Microwave & RF

23 au 24/03 – Paris
www.microwave-rf.com

♦ Salon Analyse industrielle

30 au 31/03 – Paris
www.analyse-industrielle.fr

Un événement oublié ?

Vous organisez une conférence, un salon... ou bien vous participez à un séminaire ou tout autre événement qui aurait sa place ici, partagez cette information avec tous les lecteurs. Envoyez-nous tous les détails à redaction@elektor.fr.



horloge aide-mémoire

avec neuf rappels programmables

Martien Schot (Pays-Bas)

Certains traitements médicaux demandent une prise très régulière de médicaments. Le genre de réveil proposé ici ne se limite pas à une, voire deux sonneries par jour, mais il produit jusqu'à neuf rappels quotidiens. Un dispositif idéal pour vous rendre la vie plus facile et libérer votre esprit.

Un réveil ordinaire sonne une, voire deux fois, par jour et toujours à la même heure quel que soit le jour de la semaine. Si vous avez besoin de plusieurs alarmes quotidiennes, il vous faut une batterie de réveils pour respecter vos obligations. Le problème est résolu avec cette horloge compacte que l'on peut construire facilement avec un minimum d'expérience du soudage. Elle se compose d'un circuit

imprimé de base surmonté d'un afficheur LCD ou OLED. On la manipule à l'aide de boutons-poussoirs et l'heure toujours exacte est fournie par un récepteur DCF77.

Le circuit

Ce circuit repose sur un microcontrôleur ATmega328P, c'est IC2 sur la **figure 1**. Il distribue les données sur un afficheur

LCD ou OLED à deux lignes de 16 caractères, d'un modèle courant avec les connexions sur la face supérieure ; il mesure 80 x 36 mm. Sur le LCD, on règle le contraste avec R7. Le réveil multi-alarme ne demande guère plus de composants, à part un quartz à 16 MHz pour assurer la stabilité de son horloge interne.

Comme il a bien raison d'aimer se faire

entendre, il dispose également d'une section audio spécialisée, basée sur IC3, un LM386. Lui aussi se satisfait d'un entourage restreint, où R6 règle le niveau sonore produit par un petit haut-parleur de 8 Ω branché sur X2.

Et le module récepteur DCF pour la référence temporelle ? Il reste à distance et viendra se brancher sur le connecteur X3. Les boutons-poussoirs S1 et S2 servent au maniement de l'horloge aide-mémoire, S1 pour choisir le mode, régler les alarmes, tandis que S2 (AL/Nav) intervient dans trois fonctions : le réglage des alarmes (navigation), l'arrêt de la sonnerie et la commutation vers la supervision des impulsions des signaux horaires (on y reviendra dans le logiciel). Le bouton de mode S1 affiche le menu de réglage des heures d'alarme dans la ligne du bas. Le champ actif clignote, il commence par le numéro de l'alarme, puis le jour de la semaine, l'heure et l'indicateur de direction, voyez la **figure 2** et le mode d'emploi.

On alimente le circuit sur un bloc secteur qui fournit une tension continue stabilisée de 5 V. Ces alimentations à découpage sont actuellement bon marché et à haut rendement énergétique. La diode D1 protège le circuit d'une inversion (temporaire) de polarité.

Le logiciel

Le logiciel a été développé à l'aide d'une carte Arduino UNO et de l'environnement de développement intégré (IDE) d'Arduino associé. Le **tableau 1** indique les correspondances entre les broches d'E/S d'Arduino et celles de l'ATmega.

En gros, le logiciel se compose des parties suivantes, appelées par la boucle principale (*void loop*).

- *InsideClock* comprend l'horloge interne et la comparaison avec l'heure d'alarme.
- *DCF_receiver* couvre la partie réception de DCF et le décodage des signaux.
- *ModeButton* lance la fonction *Alarmedit* complétée d'un atténuateur de rebond des contacts et affiche les données de l'alarme n°1 dans la ligne du bas.
- *NavigateButton* est appelée quand *Alarmedit* est lancée. Cette routine dispose aussi d'un atténuateur de rebond.
- *AlarmHandler* n'est appelée que lors du

Caractéristiques techniques

- jusqu'à neuf alarmes par jour
- séquence identique tous les jours ou spécifique par jour de semaine
- résolution temporelle de 15 min pour le réglage des alarmes
- séquence d'alarmes mémorisée en EEPROM
- la sonnerie se répète toutes les 3 min pendant un quart d'heure au besoin
- horloge interne synchronisée sur DCF 77
- fonction de surveillance des signaux de DCF
- double vérification de l'exactitude des informations reçues
- construction aisée, aucun CMS

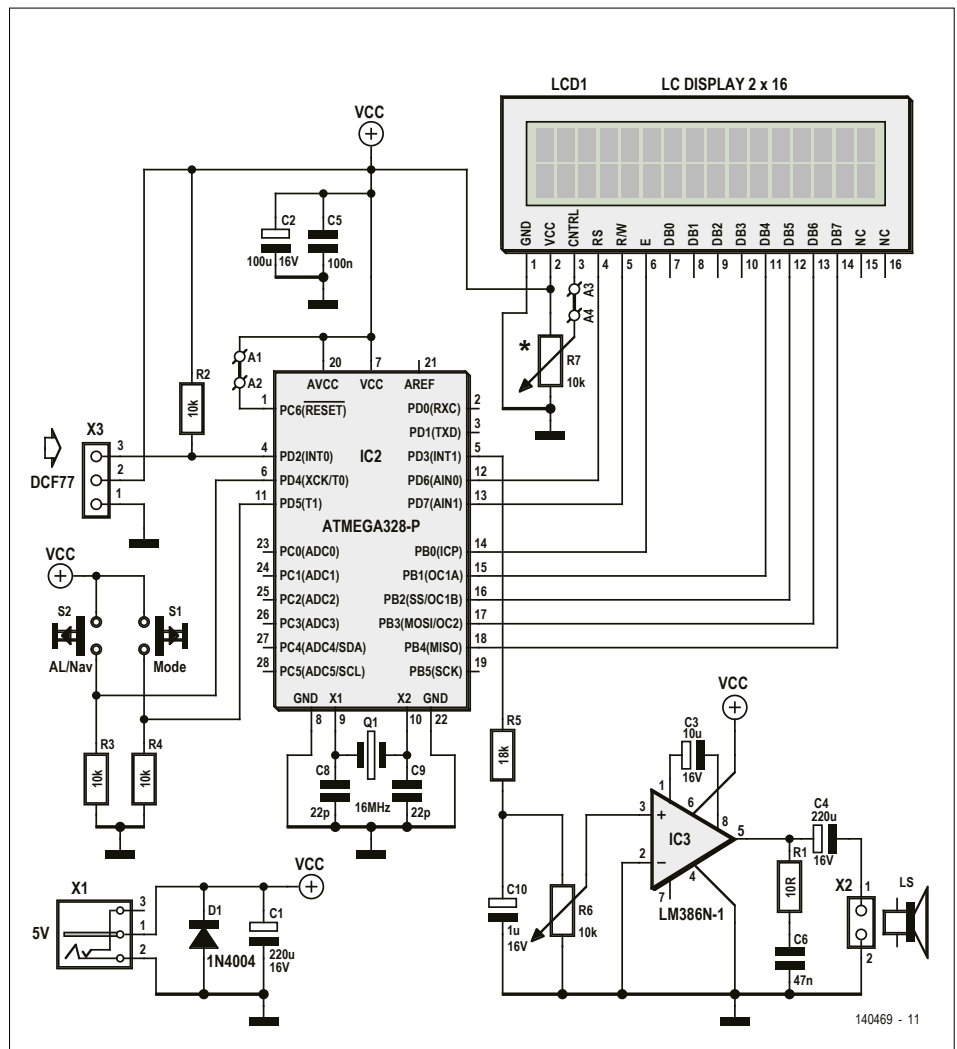


Figure 1. Le circuit repose essentiellement sur un microcontrôleur ATmega, un afficheur à deux lignes et un petit ampli audio.

Listage 1.

```
int TotalMinutes = (WkDay*1440)+(Hour*60)+Minute;
if (TotalMinutes==ExpectTotal)CorrectReceive=true;
else CorrectReceive=false;
if (TotalMinutes==11519) ExpectTotal=1440;
else ExpectTotal= ++TotalMinutes ;
```



Figure 2. Appuyez sur S1, vous verrez à la deuxième ligne de l'afficheur le menu de réglage des alarmes.

Mode d'emploi

Après la mise sous tension, attendre la synchronisation de l'horloge sur DCF77.
Réglage des alarmes :

- Appuyer sur le bouton de mode S1. La ligne du bas affiche :
#1 XXXXXX 00:00|

#1 numéro de l'alarme
choix de 1 à 9 (le 1 clignote).

XXXXXX jour de l'alarme (actuellement désactivé)
choix entre
Quotid, LunDi, MarDi, MercDi, JeuDi, VendDi, Samedi, Dimanc, XXXXXX

00:00 heure d'alarme
à régler en trois étapes : dizaines d'heures, unités d'heures et quarts d'heure

| une ligne verticale indique que l'on peut changer avec S2 le numéro de l'alarme qui clignote
- Appuyer encore sur le bouton de mode S1. La barre verticale devient horizontale. Le bouton d'alarme S2 permet alors de passer au champ XXXXXX suivant et de le faire clignoter. Nouvelle pression sur S1 pour redresser la barre et choisir par exemple Quotid. La séquence d'alarmes sera alors identique tous les jours de la semaine.
- On peut aussi régler l'heure 00:00 de la même manière.
- À la fin de la ligne, on atteint la barre _ qui se met à clignoter. Si la ligne introduite est correcte, il ne reste qu'à l'inscrire en EEPROM au moyen du bouton de mode S1 qui fait aussi quitter le mode d'édition.
- S'il y a encore une ligne à introduire ou à modifier, au lieu de pousser sur S1, on pousse sur S2 pour revoir la ligne introduite ou écrire la suivante.
- Même procédure quand on veut modifier les alarmes.

Fonctions des boutons-poussoirs

Le bouton de mode S1 ne sert qu'à modifier les alarmes.

Le bouton AL/Nav S2 a trois fonctions :

- naviguer comme décrit ci-dessus
- arrêter une alarme
- appeler la fonction de surveillance

déclenchement d'une alarme.

Dans *InsideClock*, la fonction *millis()* produit les secondes desquelles seront dérivés les calculs des minutes, heures et jours de semaine.

Dans *DCF_receiver*, on convertit les deux longueurs d'impulsion en zéros et uns. Ils seront ensuite transformés en valeurs décimales à afficher, contrôle de parité compris. Quand une trame temporelle complète a été correctement reçue, on calcule l'instant d'arrivée de la trame suivante.

La durée calculée tient compte de la somme des minutes passées depuis le début de la semaine, lundi à 00:00 h, voyez-en le calcul dans le **listage 1**. Normalement, on ajoute 1 à la somme pour la minute suivante, mais entre le dimanche à 23:59 et le lundi à 00:00, la somme des minutes passe de 11 519 à 1 440, ce qui représente un jour entier. Si la somme préalablement calculée est égale à la valeur de minutes reçue, le temps ainsi obtenu est correct, on peut l'utiliser pour synchroniser l'horloge interne et évidemment afficher l'heure. La ligne du bas revient également. L'indication que la réception s'est bien déroulée apparaît aussi. Le **figure 3** montre l'heure courante et la date sur deux lignes.

En *ModeButton*,

- Alarmedit* se lance
pendant *Alarmedit*, on change la barre horizontale en verticale à la position 15
- Alarmedit* s'arrête.

En *NavigateButton*, on appelle *Right* (barre horizontale) ou *UP* (barre verticale). La routine *Right* est reproduite dans le **listage 2**.

Les données des alarmes sont enregistrées dans un tableau à deux dimensions « Alarms[9][4] ». Il contient neuf enregistrements de quatre champs chacun. Ils sont numérotés de 0 à 8 pour les enregistrements et de 0 à 3 pour les champs. Chaque enregistrement englobe un ensemble complet de données d'alarme :

- champ 0 = jour ; 0 = chaque jour, 1 à 7 = lundi à dimanche, 8 = XXXXXX.
- champ 1 = dizaine des heures ; 0, 1, 2.
- champ 2 = unité des heures ; 0 à 9.
- champ 3 = quarts d'heure ; 0 = 00,

Listage 2.

```

void RIGHT()
{
  ++CurPosH;
  lcd.setCursor(CurPosH,1);
  switch (CurPosH)
  {
    case 0: CurPosH= 1; break; //goto 1=Alarmnumber
    case 2: lcd.setCursor(1,1); lcd.print(Ala_Num+1);
            CharakStr=AlarmDaySt[Alarms[Ala_Num][0]]; CurPosH=3; break; // DOW
    case 4: lcd.setCursor(3,1); lcd.print(AlarmDaySt[Alarms[Ala_Num][0]]);
            CharakStr=String(Alarms[Ala_Num][1]); CurPosH=10; break; // ten minutes
    case 11: lcd.setCursor(10,1); lcd.print(String(Alarms[Ala_Num][1]));
            CharakStr=String(Alarms[Ala_Num][2]); CurPosH=11; break; //minutes
    case 12: lcd.setCursor(11,1); lcd.print(String(Alarms[Ala_Num][2]));
            CharakStr=QuarterStr[Alarms[Ala_Num][3]]; CurPosH=13; break; //Quarters
    case 14: lcd.setCursor(13,1); lcd.print(QuarterStr[Alarms[Ala_Num][3]]);
            CharakStr=" "; CurPosH=15; break; //write data
    case 16: CurPosH= 1; CharakStr=String(Ala_Num+1); break; //goto Alarmnumber again
  }
  lcd.setCursor(CurPosH,1);
}

```

1 = 15, 2 = 30, 3 = 45.

Comme les champs se composent de 1, 2 ou 6 caractères, on n'a pas utilisé la fonction standard de clignotement qui n'affecte qu'un seul caractère, mais on a écrit une routine qui permet de les faire clignoter tous ensemble, selon leur nombre. Voyez la partie *void CharBlink* du **listage 3**.

Pour simplifier la comparaison avec le temps réel, nous avons créé un format spécial (**listage 4**) qui diffère de celui utilisé lors de la vérification des données horaires reçues.

Notre temps réel (*RealTime*) est égal à la somme des minutes plus 100 fois les heures plus 10 000 fois les jours de semaine. Le format de la somme pour une alarme (*AlarmSum[i]*) est le même que celui du temps réel.

Les jours de semaine en temps réel ont une valeur entre 1 et 7 ; dans les alarmes, ces valeurs vont de 0 à 8. Donc quand on met dans le champ 0 la valeur XXXXXX (soit la valeur 8), il n'y aura jamais d'alarme.

Quand on a choisi « Quotid » (0), pareil tous les jours, la somme pour l'alarme est toujours inférieure à 2 360 (24:00 h). Dans ce cas, il faut y ajouter la valeur du jour de semaine pour pouvoir effectuer la comparaison avec le temps réel.

AlarmHandler. On l'appelle quand, dans le listage 4, *Alarm=true* est exécuté dans



Figure 3. En temps normal, l'afficheur contient l'heure exacte et la date. Le logiciel concerné est disponible en quatre langues.

Listage 3.

```

void CharBlink()
{
  if (millis()-LastOne > 400) // 0,4 sec
  {
    LastOne=millis();
    switch (CharakStr.length())
    {
      case 6: if(Charak=="") Charak=CharakStr;
              else Charak=""; // 6 characters days
              break;
      case 2: if(Charak=="") Charak=CharakStr;
              else Charak=""; // 2 characters quarters
              break;
      case 1: if(Charak=="") Charak=CharakStr;
              else Charak=""; // 1 character number, minutes
              break; // ten minutes
    }
    lcd.print(Charak); lcd.setCursor(CurPosH,1);
  }
}

```


Listage 4.

```

if (IntSec == 0 && AlarmEdit == false)
{
  RealTime=IntMin+(IntHour*100)+(IntWkDay*10000L);
  for (byte i = 0; i<9; i++)// 9 alarms
  {
    if(AlarmSum[i]< 2460) CompareSum=AlarmSum[i]+(IntWkDay*10000L);
    else CompareSum=AlarmSum[i];
    if (CompareSum==RealTime){Alarm=true;ParticuAla=i;TriggerTime=RealTime;}
  }
}

```

la ligne du bas. L'alarme correspondante est également affichée dans la ligne du bas. Durant 15 min, la petite mélodie de l'alarme se répète alors toutes les 3 min,

à moins d'appuyer sur le bouton S2 pour arrêter le processus.

Le logiciel met à profit la bibliothèque EEPROM d'Arduino, présente dans le dos-

sier de bibliothèques standard d'Arduino. Au premier démarrage du logiciel, l'EEPROM est vierge (pleine de FF) et il faut y inscrire d'abord les 36 octets pour les neuf

Liste des composants

Résistances ¼ W :

R1 = 10 Ω
 R2 à R4 = 10 k Ω
 R5 = 18 k Ω
 R7 (*seulement avec LCD), R6 = 10 k Ω pot. ajust. hor.
 A1-A2 = pont de câblage
 A3-A4 = pont de câblage

Condensateurs :

C1, C4 = 220 μ F/16 V radial
 C2 = 100 μ F/16 V radial
 C3 = 10 μ F/16 V radial
 C5 = 100 nF
 C6 = 47 nF
 C7 = 10 nF
 C8, C9 = 22 pF
 C10 = 1 μ F/16 V tantale

Semi-conducteurs :

IC2= ATMEGA328P
 IC3 = LM386
 D1 = 1N4004

Divers :

Affichage = LCD 2 x16 caractères ou OLED (p.ex. Winstar WEH01602 ou EA W162-X3LG)
 Q1 = quartz 16 MHz
 X1 = connecteur d'alimentation encartable, broche centrale 2 mm
 X2 = borne à 2 vis au pas de 5,08 mm
 X3 = borne à 3 vis au pas de 5,08 mm
 S1, S2 = bouton-poussoir (p.ex. Conrad réf. 700046-89)
 module récepteur DCF (p.ex. Conrad réf. 641138-89)

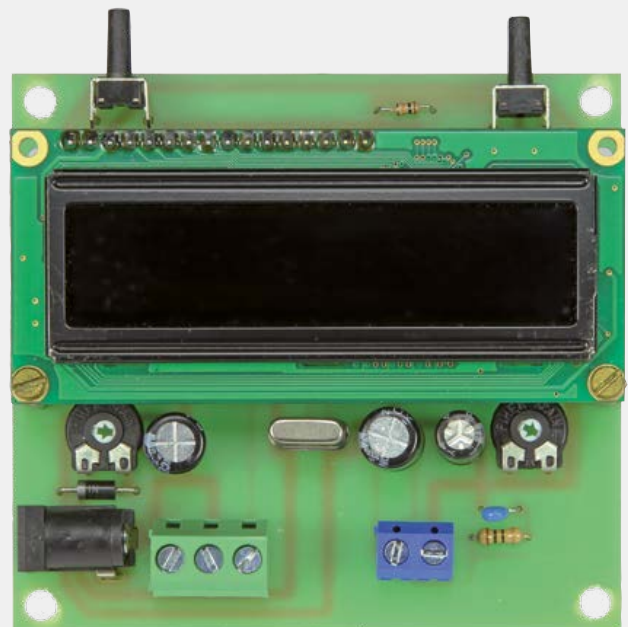
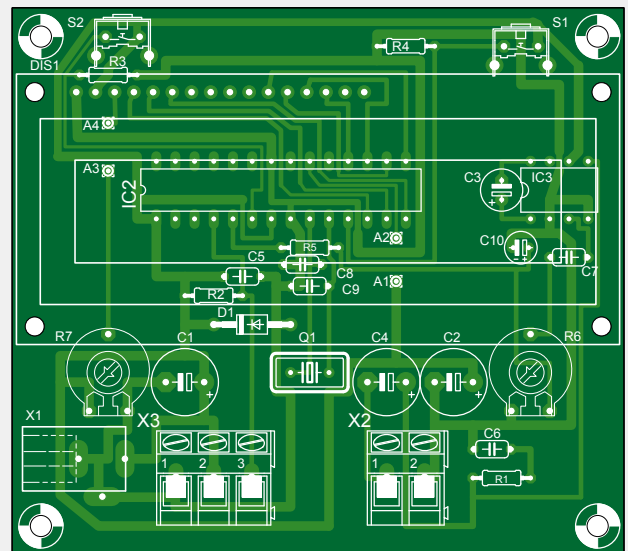


Figure 4. Voici le circuit imprimé conçu par l'auteur pour l'horloge aide-mémoire. Au prix de deux ponts de câblage (A1-A2 et A3-A4), un circuit à simple face suffit.

alarmes. Le champ 0 est toujours mis à 8 de manière à inhiber toutes les alarmes. On remplit le reste des champs de zéros (cf. **listage 5**). Toutes les alarmes sont ainsi nulles comme l'alarme n°1 : **# 1 XXXXXX 00:00**.

Nous avons encore ajouté une fonction de surveillance qui s'active avec S2 (AL/Nav). Elle affiche les informations sur les impulsions reçues. Par exemple : **P13 0=100 mS** ou **P23 1=200 mS**. Pxx est le numéro de l'impulsion, suivi par 0 ou 1 selon la longueur en millisecondes de l'impulsion. Cette longueur peut varier et dériver de 10 % environ. Une nouvelle action sur S2 ramène à l'affichage normal. La commande de l'affichage a recours à la bibliothèque *LCD_OLED_FourBit* qui gère aussi bien un LCD qu'un écran OLED. Avant de programmer votre ATmega328P au moyen d'une carte Arduino, ajoutez dans le dossier *libraries* d'Arduino le dossier *LCD_OLED_FourBit*. Mettez-y :

- LCD_OLED_FourBit.h
- LCD_OLED_FourBit.cpp
- keywords.txt

Ces dossiers, ainsi que le logiciel en quatre langues, sont disponibles au téléchargement sur la page du projet [3]. Créez un dossier *HelloLcdOledChar* et insérez-y *HelloLcdOledChar.ino*.

La construction

La **figure 4** représente le circuit imprimé que l'auteur a développé et dont le tracé est également disponible sur le site du magazine [3]. C'est un petit circuit imprimé simple face de 7,2 x 8,1 cm. L'implantation des composants est simple. Nombre d'entre eux seront surmontés par l'afficheur, surveillez donc bien leur hauteur lors du montage. Les boutons-poussoirs sont dotés d'une longue tête pour dépasser de la face avant du boîtier. Le bouton de mode est installé un peu plus en retrait pour éviter d'y toucher par mégarde quand on arrête une alarme. Le couplage de l'afficheur au circuit imprimé s'effectue au moyen de bouts de fil ou d'embases mâle et femelle. Libre à vous d'opter pour la version LCD ou OLED (p.ex. WEH01602 de Winstar), le logiciel marche avec les deux et les broches sont identiques.

Le haut-parleur est à monter en bonne et

Listage 5.

```
byte Adress=0;//EEPROM init 36 bytes
if (EEPROM.read(Adress)== 0xFF)//("empty=FF")
{for (byte i = 0; i < 9; i++)
  {for (byte j = 0; j < 4; j++)
    {if (j == 0) EEPROM.write(Adress, 8);
     else EEPROM.write(Adress, 0);
     ++Adress;
    }
  }
}
```

Tableau 1.

Correspondance entre les broches d'E/S d'Arduino et celles de l'ATmega.

ARDUINO	6	7	8	9	10	11	12
ATmega	12	13	14	15	16	17	18
ARDUINO DCF77PIN = 2	ATmega broche 4						
ARDUINO SPEAKPIN = 3	ATmega broche 5						
ARDUINO NAVIGBUT = 4	ATmega broche 6						
ARDUINO MODEBUTT = 5	ATmega broche 11						

due place dans le boîtier, raccordé à X2. Le récepteur DCF, logé dans une petite boîte en plastique, doit être raccordé à X3 par un câble à trois conducteurs d'environ 50 cm. La numérotation sur le circuit imprimé va de gauche à droite, alors que sur la platine de Conrad, elle va de droite à gauche, il faut donc croiser les fils de liaison.

À propos de ce récepteur DCF, il y a lieu de se méfier de son environnement, éloigner le câble de raccordement au secteur, éviter la proximité du téléviseur, de l'ordinateur et de toute source de parasites, même des éclairages à tube fluorescent. ◀

(140469 – version française : Robert Grignard)



Liens

- [1] Informations sur le signal DCF77 : <http://fr.wikipedia.org/wiki/DCF77>
- [2] Brochage ATmega : www.hobbytronics.co.uk/arduino-atmega328-pinout
- [3] www.elektor-magazine.fr/140469

hors-circuits

le filtrage numérique sans stress

les filtres CIC

Robert Lacoste (Chaville)

Le précédent article de cette série traitait de filtres numériques très classiques, les filtres à réponse impulsionnelle finie ou FIR. Ils permettent de synthétiser n'importe quelle réponse en fréquence, mais nécessitent parfois un processeur musclé pour le calcul. Ce mois-ci, je m'intéresse aux filtres dits CIC (pour *Cascaded Integrator Comb*). Ces filtres, beaucoup moins gourmands en ressources de calcul que les FIR, mais moins flexibles, sont souvent utilisés pour dégrossir le travail dans un système de traitement numérique du signal. Pas d'affolement, un CIC peut se résumer au calcul d'une moyenne glissante... mais d'une façon astucieuse et très efficace. Cet article aborde une notion fondamentale dans le traitement numérique du signal : le multi-débit.

Un exemple typique...

Commençons par une application pratique avec un signal audio monophonique de qualité hi-fi, échantillonné à 44,1 Ké/s (soit 44 100 échantillons par seconde, c'est la fréquence d'acquisition d'un CD audio). Comme le dit Monsieur Nyquist, une telle fréquence d'acquisition permet de représenter tout le spectre, du continu à un peu moins de $44,1 / 2 = 22,05$ kHz, parfait pour

de l'audio avec un filtre anti-repliement assez raide. Supposons que vous souhaitez traiter numériquement ce signal pour détecter des fréquences très basses, disons inférieures à 100 Hz. Les applications d'un tel traitement sont nombreuses : musique électronique, détection de vibrations....

Comment faire ? Il faut d'abord filtrer le signal pour ne garder que la bande utile,

c'est-à-dire la bande DC à 100 Hz, soit $100 / 44\,100 = 0,4\%$ du spectre total. C'est faisable avec un filtre FIR, mais un filtre aussi étroit nécessitera beaucoup de « taps » (cf article [1]), peut-être 256 taps. Le filtre FIR effectuera donc 256 multiplications et additions pour chaque échantillon audio, c'est-à-dire 44 100 fois par seconde. Cela fait un peu plus de 22 millions d'opérations mathématiques par seconde. Ce n'est pas simple !

Techniques multi-débits

Comme dans cet exemple, les signaux de la vie réelle sont souvent des flots d'informations volumineux dont seule une infime partie est utile. L'objectif est toujours de réduire ce flot le plus vite possible, pour traiter plus efficacement l'information réellement utile. Ici, le taux d'échantillonnage de 44,1 Kéchant/s est imposé par la bande de fréquence 0-20 kHz du signal audio hi-fi, mais les informations intéressantes sont contenues sur une plage de 100 Hz seulement. Par conséquent 99,6% de l'information est inutile, et 99,6% des opérations mathématiques sont donc effectuées pour rien. Comment améliorer cette situation ? Il suffit de mettre en œuvre des algorithmes multi-débits (*multi-rate* pour les anglophones). De quoi s'agit-il ? Le signal initial a un débit élevé, mais une faible proportion d'informations utiles. L'objectif est de réduire, étape par étape, le taux d'échantillonnage tout en conservant les informations utiles et en augmentant donc la densité de l'information (**figure 1**). Tant que le débit est très rapide, les algorithmes numériques utilisés doivent être très simples pour ne pas surcharger le processeur, tandis que des algorithmes beaucoup plus sophistiqués peuvent être déployés une fois le débit réduit. Bien sûr, pour les applications de

production de signaux, on peut utiliser les mêmes techniques en sens inverse, pour augmenter progressivement le débit d'un signal avec une réduction de la densité de l'information. Sur le plan de la programmation, gérer dans un même microcontrôleur ou FPGA des flux de données avec des débits différents, et ce en temps réel, nécessite un peu de doigté, c'est la joie du multi-débit, mais cela nous entraînerait un peu loin.

Décimation

En résumé : nous avons un signal échantillonné à 44,1 Kéchant/s contenant un spectre audio de DC à 20 kHz, mais nous ne sommes intéressés que par la bande DC à 100 Hz. Nous souhaitons donc réduire rapidement le taux d'échantillonnage à un peu plus de $2 \times 100 = 200$ échant/s. En traitement du signal, l'opération, vraiment simple, de réduction du taux d'échantillonnage est appelée décimation. Comment réduire le taux d'échantillonnage par ex. de 44 100 à 220 échant/s ? $44\ 100 / 220 = 200$, il suffit de garder un échantillon sur 200 et de jeter les 199 autres ! C'est parfaitement légitime, mais à une condition : il faut, avant de procéder à cette décimation, s'assurer que le signal peut être représenté de manière fiable avec ce taux

d'échantillonnage de 220 échant/s, c'est-à-dire qu'il ne doit pas avoir de composantes de fréquences supérieures à $220/2 = 110$ Hz. Sinon, après décimation, le signal sera imparfait à cause du pénible phénomène de repliement spectral. Il faut donc, avant la décimation, appliquer un filtre passe-bas de fréquence de coupure 100 Hz sur le signal. Argh, retour au problème initial : il faut appliquer un filtre passe-bas sur le signal échantillonné à 44,1 Kéchant/s avant de pouvoir réduire son débit via une décimation ! Nous avons donc besoin d'un algorithme de filtrage passe-bas qui consomme très peu de ressources du processeur...

Moyennes glissantes

Vous avez sans doute observé qu'effectuer une moyenne permet d'éliminer les « bruits » dans des données, c'est-à-dire de ne conserver que les variations lentes d'un signal en éliminant les variations rapides. En d'autres termes, une moyenne, ou plus précisément une moyenne glissante, est un moyen simple pour réaliser un filtre passe-bas. L'algorithme d'une moyenne glissante est ultra simple : prendre N échantillons consécutifs du signal et calculer leur moyenne (ou simplement leur somme, ce qui est la même chose à un facteur multiplicatif N près),

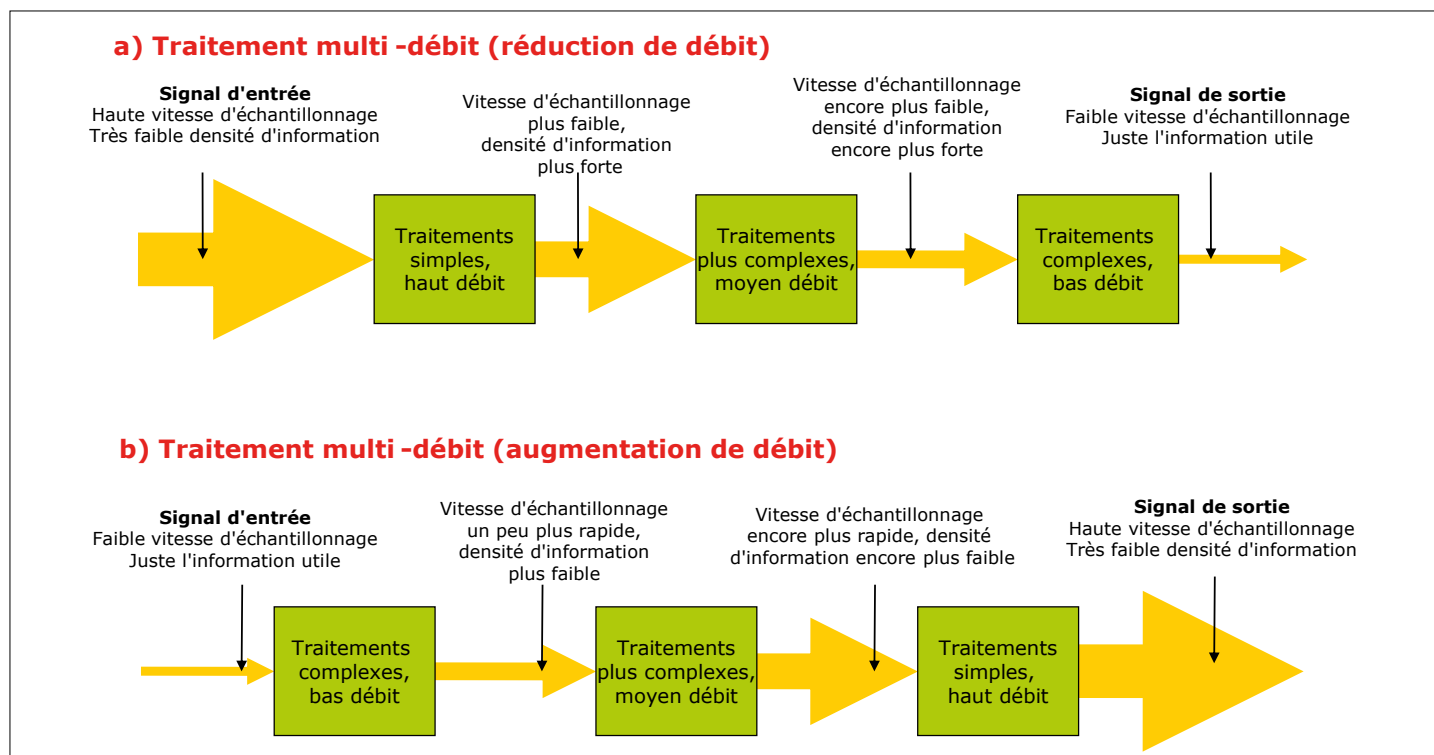


Figure 1. **a)** Le traitement multi-débit est une technique qui permet de réduire graduellement la fréquence d'échantillonnage d'un signal tout en augmentant la densité des informations utiles. **b)** Réciproquement la même approche est utilisable pour augmenter une fréquence d'échantillonnage.

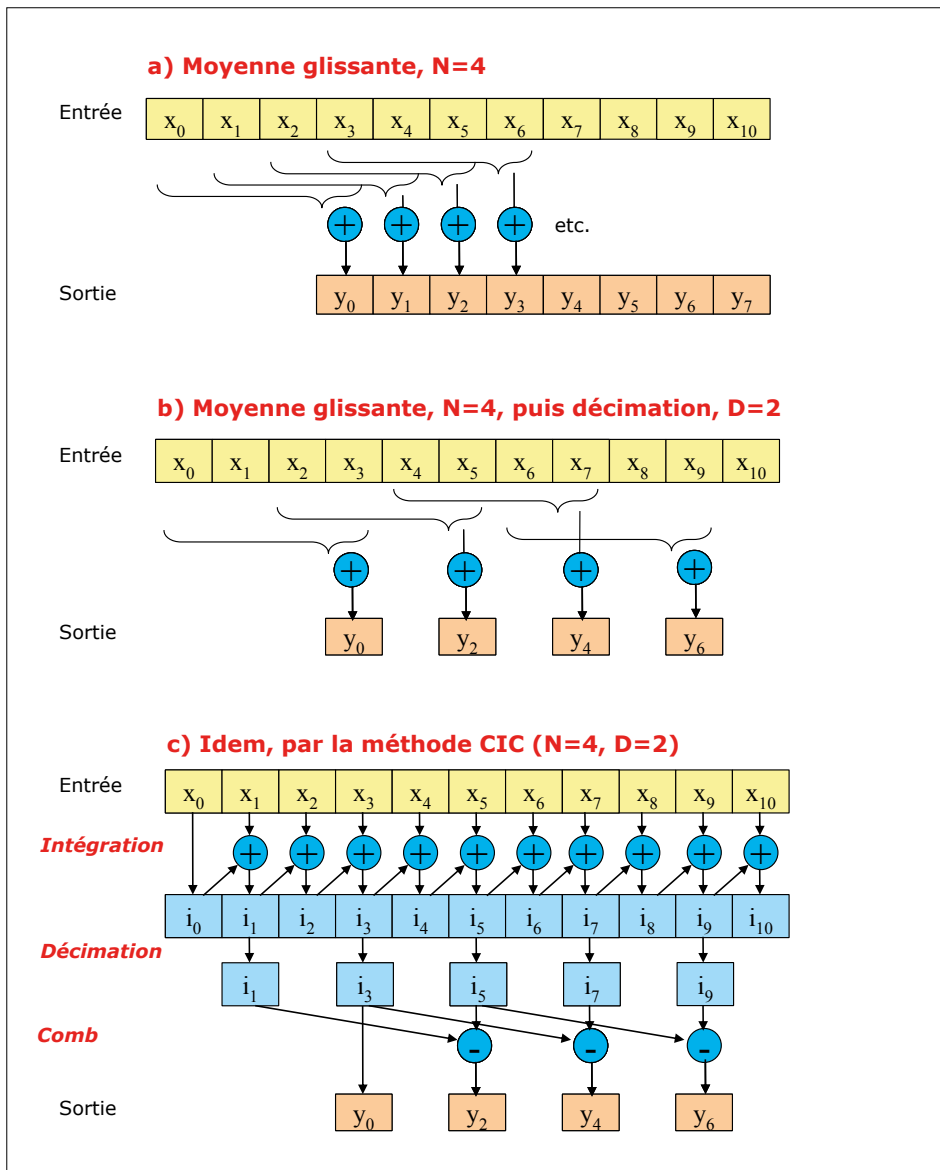


Figure 2. **a)** Une moyenne glissante, ici sur $N = 4$ points successifs. **b)** Une opération de décimation, ici avec $D = 2$, revient à ne garder qu'un point sur deux. **c)** Un filtre CIC donne exactement la même valeur de sortie, mais d'une manière plus efficace.

cela donne le signal filtré. La **figure 2a** vous montre un exemple avec $N = 4$. Comme une moyenne glissante ne nécessite rien d'autre que des additions,

nous pouvons l'utiliser pour éliminer à moindres frais les hautes fréquences du signal : ici, un choix approprié pour N permettra de ne pas trop modifier les

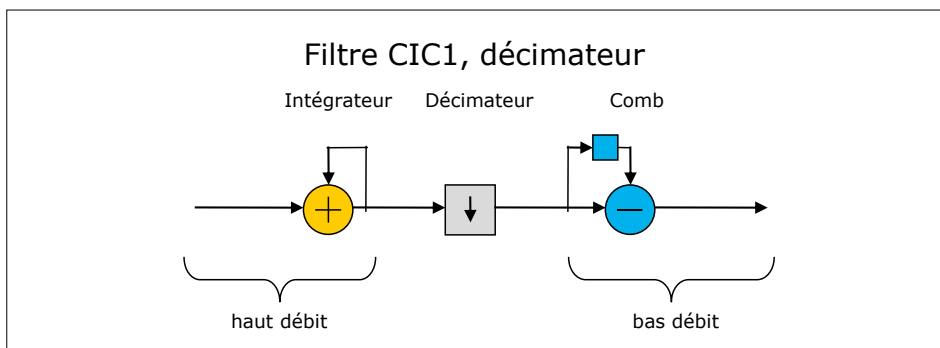


Figure 3. Un filtre CIC d'ordre 1 (CIC1) comporte trois étapes : intégration, décimation et comb.

fréquences inférieures à 100 Hz tout en atténuant les plus hautes fréquences. $N = 256$ points serait une bonne base de départ. Ensuite une étape de décimation serait légitime, par ex. d'un facteur $D = 64$, ce qui ramène le débit à 689 éch/s. Évidemment si vous réfléchissez deux secondes, vous comprendrez qu'il est inutile de calculer toutes les moyennes glissantes si ensuite on doit décimer. Il suffit de calculer celles qu'on veut conserver après décimation. Ici, il faudra calculer la somme de $N = 256$ échantillons et ce $44\,100 / 64 = 689$ fois par seconde, soit $256 \times 689 = 176\,384$ additions par seconde ; on est loin des 21 millions d'opérations initiales !

La **figure 2b** illustre ce mode de calcul dans le cas d'une moyenne glissante sur quatre échantillons ($N=4$) puis d'une décimation d'un facteur $D = 2$. Il suffit de calculer une moyenne tous les deux échantillons. Mais regardez à nouveau la figure 2b : cet algorithme n'est pas très efficace, car nous calculons plusieurs fois la même opération. Par exemple, la première valeur de sortie est $x_0 + x_1 + x_2 + x_3$ et la deuxième est $x_2 + x_3 + x_4 + x_5$: soit deux fois la somme $x_2 + x_3$. Ce phénomène sera encore plus accentué si la moyenne est calculée sur un plus grand nombre de points, ce qui est généralement le cas. Si nous calculons une moyenne glissante pour $N = 256$ points avec une décimation d'un facteur 64, certaines additions seraient répétées 16 fois. Comment optimiser ce calcul ?

Le filtre CIC enfin !

Commençons par imposer que la longueur N de la moyenne soit un multiple du facteur de décimation D , ce qui n'est pas une grande concession. Cela nous amène à ce qu'on appelle un filtre CIC, illustré sur la **figure 2c** dans sa forme la plus simple : le CIC à un étage (« CIC1 »). Mathématiquement, ce filtre donne exactement le même résultat qu'une moyenne glissante suivie d'une décimation, mais il est beaucoup plus efficace sur le plan algorithmique. L'idée est très simple. Par ex. avec $N = 10$, le 100^e échantillon filtré est égal à la somme des échantillons x_{101} à x_{110} . Plutôt que de calculer cette somme, pourquoi ne pas effectuer la somme des échantillons x_0 à x_{110} et soustraire la somme des échantillons x_0 à x_{100} ? Le résultat sera bien sûr exactement le même.

Bien entendu, c'est absurde de procé-

der de la sorte à la main et pour un seul échantillon, mais c'est judicieux pour le filtrage de tout un signal, car il n'y a aucune addition inutile. Comme illustré sur la figure 2c, l'algorithme optimisé est un processus en trois étapes. Tout d'abord, il faut calculer la somme de tous les échantillons précédents. C'est facile, il suffit d'ajouter chaque nouvel échantillon à un accumulateur. Pour les matheux, sommer les échantillons revient à faire une intégration. Ensuite on fait la décimation, c'est-à-dire qu'on ne fait rien, sauf pour un échantillon tous les D pour lequel une valeur de sortie doit être calculée. Pour chaque valeur de sortie à calculer, on fait enfin la soustraction entre l'accumulateur actuel et l'une de ses valeurs précédentes, via une pile FIFO (*first in, first out*) de N/D éléments. Si l'on prend $N = D$, la pile FIFO ne contient qu'un seul élément. Un peu de vocabulaire : l'addition ou la soustraction de deux valeurs décalées dans le temps est appelée « comb » en traitement du signal ; pour les matheux, c'est très proche d'une différentiation. Vous comprenez maintenant pourquoi un tel filtre est appelé CIC (*Cascaded Integrator Comb*). La **figure 3** montre la structure du filtre CIC1 que l'on vient ainsi de créer.

Encore une fois le résultat est strictement le même que celui d'une simple moyenne glissante, mais il nécessite beaucoup moins d'opérations de calcul. En reprenant notre exemple avec une moyenne glissante sur $N = 256$ points suivie d'une décimation d'un facteur $D = 64$, l'algorithme CIC1 effectue une addition pour chaque échantillon d'entrée, soit 44 100 additions par seconde, puis une soustraction pour chaque échantillon de sortie, soit $44\,100 / 64 = 689$ soustractions par seconde. Ce qui fait au total $44\,100 + 689 = 44\,789$ opérations par seconde, contre **176 384** avec un *bête* calcul de moyenne glissante, et ce pour exactement le même résultat. Pas mal, non ?

Caractéristiques des filtres CIC

Après les bonnes nouvelles, passons aux mauvaises. Quelle est la réponse en fréquence d'un tel filtre CIC1 ? La même réponse en fréquence qu'une moyenne glissante bien sûr, puisqu'ils sont numériquement identiques, mais quelle est-elle ? Malheureusement elle est assez éloignée de celle d'un filtre passe-bas parfait. Il y a trois manières de la trouver : mathématiquement, par simulation ou bien intuiti-

Le petit script en SciLab

```
// longueur de la simulation
LENGTH=1024;

// Taille de la fenêtre
D=8;

// Réponse impulsionnelle du filtre (rectangulaire)
imp=zeros(1:LENGTH);
imp(1:D)=1/D;

// Réponse en fréquence correspondante
freqresponse=abs(fft(imp));

// Tracer la réponse, en échelles log et lin
subplot(1,2,1);
plot2d((1:LENGTH/2)/LENGTH,freqresponse(1:$/2));
xtitle('Moving average filter - D=8 (linear scale)');
subplot(1,2,2);

db=20*log10(freqresponse+1e-200);
plot2d((1:LENGTH/2)/LENGTH,db(1:$/2),rect=[0,-25,0.5,0]);
xtitle('Moving average filter - D=8 (dB)');
```

tivement. Commençons par les maths. Vous vous souvenez de mon article sur les filtres FIR [1] ? Un filtre réalisé avec une moyenne glissante revient à multiplier N échantillons successifs par une constante $1/N$ et à faire la somme du tout. C'est

donc ni plus ni moins qu'un filtre FIR de N *taps* avec tous les coefficients égaux à $1/N$, c'est-à-dire un filtre FIR dont la réponse impulsionnelle est une impulsion rectangulaire de longueur N et de hauteur constante. Vous rappelez-vous que

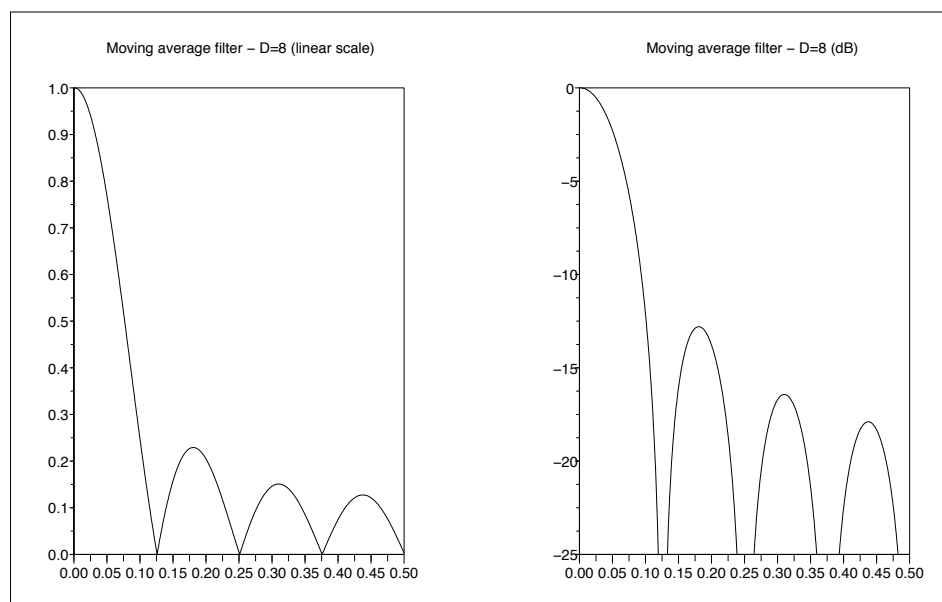


Figure 4. Voici la réponse en fréquence d'un filtre CIC1 avec une fenêtre pour la moyenne de huit échantillons et un facteur de décimation égal à huit, avec une échelle linéaire (à gauche) et une échelle logarithmique (à droite). Mathématiquement c'est une courbe $\sin(x)/x$, avec un lobe principal de type passe-bas, mais également des lobes secondaires.

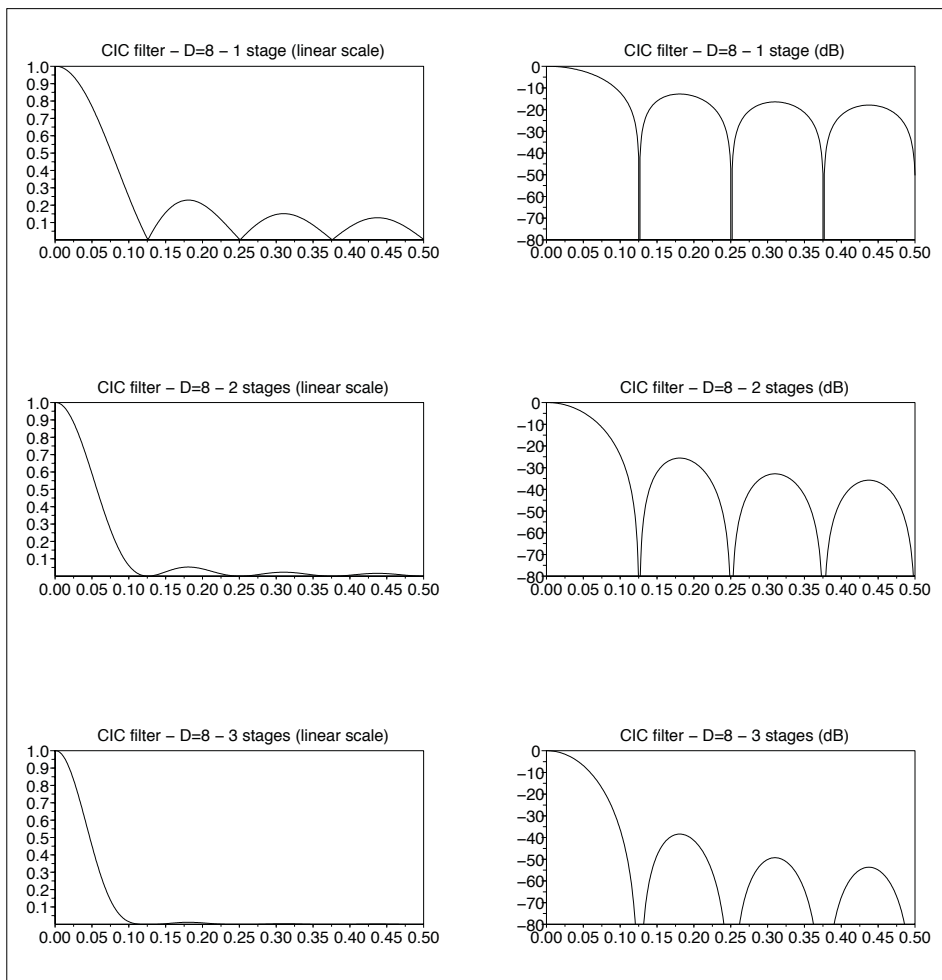


Figure 5. Mettre en cascade plusieurs étages CIC identiques permet de réduire les lobes latéraux, au prix d'un nombre de calculs légèrement plus élevé. Ici une comparaison de filtres CIC1 (en haut), CIC2 (milieu) et CIC3 (en bas).

la réponse en fréquence d'un filtre FIR est la transformée de Fourier de sa réponse impulsionnelle ? Et la transformée de Fourier d'une impulsion rectangulaire est une fonction très commune en traitement du signal, appelée sinus cardinal et égale à $\text{sinc}(x) = \sin(x) / x$. Vous voulez voir sa forme ? J'ai écrit pour cela un petit script en SciLab qui calcule la réponse en fréquence d'un filtre CIC1 avec $D = N = 8$. Le code est simple, je suis sûr que vous le comprendrez, même si vous n'êtes pas familier de SciLab ou Matlab.

Téléchargez SciLab sur votre PC (si ce n'est pas déjà fait), exécutez ce script et vous obtiendrez le graphe de la **figure 4**. Comment comprendre cette réponse intuitivement ? La réponse en fréquence a un premier lobe ressemblant comme prévu à un filtre passe-bas : l'atténuation augmente avec la fréquence, jusqu'à devenir infinie (gain nul) à une certaine fréquence. Pour $D = 8$, ce point est à une fréquence $f = 1/8 = 0,125$ fois le taux

d'échantillonnage. Réfléchissez, cela est tout à fait normal : si le signal d'entrée est une sinusoïde exactement à cette fréquence, alors sa période sera égale à la longueur de calcul de la moyenne, donc cette moyenne sera toujours nulle !

Toutefois, et c'est la première mauvaise nouvelle, lorsque la fréquence d'entrée augmente au-delà de cette fréquence alors le filtre n'est plus aussi bon : il y a un deuxième lobe, avec atténuation de seulement 13 dB, puis une nouvelle valeur nulle puis un troisième lobe, etc. C'est la caractéristique d'un filtre réalisé par moyenne glissante : l'atténuation est maximale à chaque fois que la période du signal d'entrée est un sous-multiple de la durée de la fenêtre de calcul de la moyenne.

La seconde mauvaise nouvelle est que la bande passante d'un filtre CIC1 est loin d'être plate. Regardez une nouvelle fois

la figure 4 : l'atténuation n'est de 0 dB seulement qu'à la fréquence 0 (DC), puis elle augmente régulièrement. C'est donc bien moins bon qu'un filtre passe-bas de type Butterworth ou autre, qui permet d'obtenir une réponse assez plate dans la bande passante. C'est le prix à payer pour l'efficacité... Ceci dit rien ne vous empêche de corriger ce comportement par un peu plus de calcul. Pour ceux que cela intéresse, il suffit d'ajouter, après la décimation, un filtre FIR ayant une courbe de réponse précisément inverse de celle décroissante du CIC, cela permet de retrouver une réponse plate dans la bande passante ! Un tel filtre est bien sûr nommé *invsinc*. La bonne nouvelle est que ce filtre de correction sera réalisé après décimation, et donc sans gros besoins en ressources de calcul.

Revenons aux filtres CIC. Choisir convenablement les paramètres N et D permet d'optimiser la réponse du filtre CIC1, mais il présentera toujours les mêmes lobes parasites. Que faire si cela ne suffit pas pour votre application ? La solution consiste à mettre en cascade plusieurs filtres CIC1 : leurs lobes parasites seront atténués d'autant. Cela revient à calculer des moyennes de moyennes de moyennes etc. La **figure 5** montre les réponses en fréquence obtenues en mettant en cascade 1, 2 ou 3 filtres CIC1 dans le cas $N=D=8$.

Dans la pratique, mettre en cascade plusieurs filtres CIC1 est réalisable d'une manière astucieuse, surtout s'ils ont les mêmes paramètres N et D . Par exemple si vous associez deux filtres CIC1, vous avez les six étapes suivantes : intégration, décimation, *comb*, puis intégration, décimation, *comb*. Comme toutes ces opérations sont linéaires, rien n'interdit de les exécuter dans un autre ordre, et l'on obtient la structure du filtre CIC2 représenté en **figure 6a** : les deux intégrations sont réalisées d'abord, puis une seule décimation, puis les deux opérations *comb*. Bien sûr rien ne vous empêche de construire sur le même modèle des filtres d'ordre plus élevés de type CIC3, CIC4, etc.

Avant d'oublier : les filtres CIC peuvent aussi être utilisés dans l'autre sens, pour augmenter la fréquence d'échantillonnage d'un signal. Si, si, ça peut servir. Pour cela

il suffit d'une part de remplacer le décodeur par sa réciproque (appelé interpolateur), et d'inverser les opérations : *comb*, puis interpolateur, puis intégrateur (figure 6b). Un interpolateur dans ce contexte n'est rien de plus qu'un verrou : chaque échantillon est simplement dupliqué N fois. Vous pouvez le vérifier, un tel algorithme réalise en fait très exactement une interpolation linéaire entre chaque échantillon d'entrée.

Mise en œuvre

En pratique, comment utiliser un filtre CIC dans votre prochain projet ? Tout d'abord vous l'avez peut-être déjà fait sans le savoir : de très nombreux circuits intégrés exploitent en interne des filtres CIC. C'est le cas de la majorité des composants faisant appel à du sur-échantillonnage : CODECs audio, synthétiseurs de fréquence de type DDS, convertisseurs analogique/numérique sigma-delta, récepteurs radio avec traitement numérique des signaux de bande de base, etc. À titre d'exemple, côté « hautes performances », allez fureter sur l'internet et regardez la notice du composant AD6636 (Analog Devices). Ce monstre intègre six filtres CIC à haute vitesse (jusqu'à 150 Méc/s...), suivis de filtres FIR configurables dont vous aurez compris la fonction si vous avez lu ce qui précède. Chaque filtre CIC est en outre précédé par un mélangeur numérique permettant de centrer le filtre sur une fréquence donnée. Étudier la notice d'un tel composant permet de comprendre pas mal de notions...

Bien sûr, vous pouvez coder vous-même un filtre CIC, que ce soit avec quelques lignes de VHDL dans un composant programmable type FPGA, ou bien avec quelques lignes de C dans un DSP, voire dans un microcontrôleur. En effet les filtres CIC ne demandent que très peu de ressources de calcul : quelques opé-

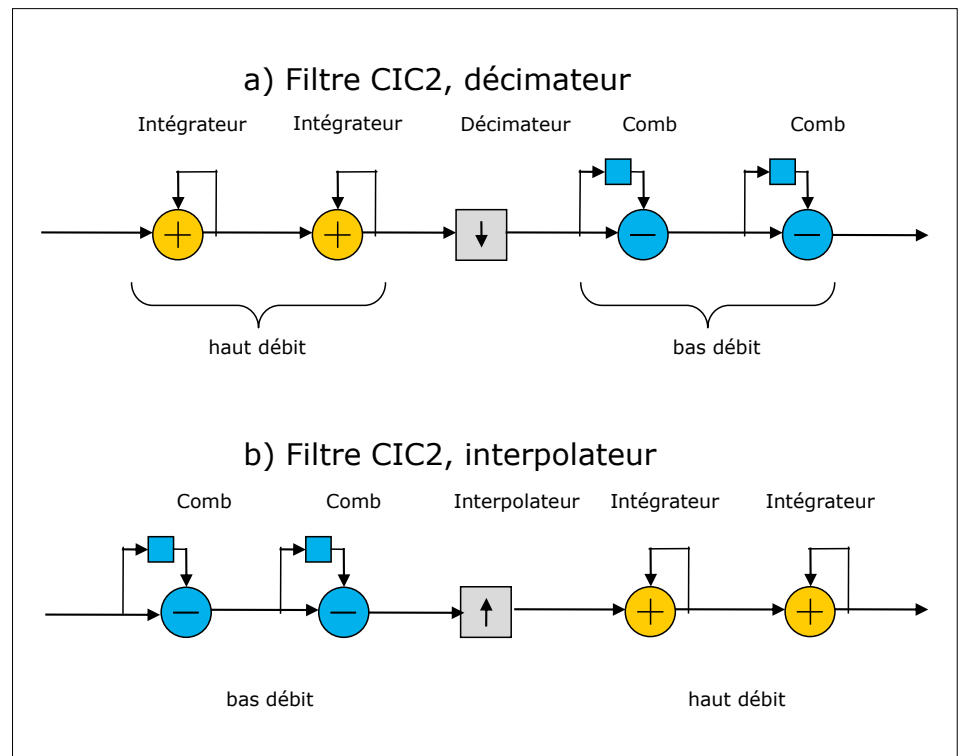


Figure 6. a) Regrouper les étapes intégration et *comb* de deux filtres CIC1 successifs est un moyen efficace pour réaliser un filtre CIC2. b) Réciproquement un filtre CIC peut servir à augmenter la fréquence d'échantillonnage d'un signal, simplement en échangeant l'intégrateur et le *comb*, et en remplaçant le décimateur par un interpolateur.

rations par échantillon seulement. Ainsi même un petit microcontrôleur à 8 bits permet de réaliser des filtres FIR sur plusieurs dizaines de milliers d'échantillons par seconde. Vous pouvez ajouter un ou deux ordres de grandeur avec un microcontrôleur à 32 bits de type Cortex M3, et ce pour moins d'un euro...

Pour conclure

Comme vous l'avez compris, les filtres CIC ne sont ni plus ni moins que des moyennes glissantes réalisées élégamment. Si l'on accepte leur réponse en fréquence un peu chahutée, ils permettent de concevoir des filtres numériques passe-bas très peu gourmands en

ressources, ainsi ils trouvent leur place dans toute étape de réduction (ou d'augmentation) de la fréquence d'échantillonnage d'un signal. Dans mon prochain article, je clôturerai cette mini-série sur les filtres numériques avec une classe de filtres très performants, mais un peu plus délicats à utiliser : les filtres à réponse impulsionnelle infinie (IIR).

(150636)

Cet article a été publié dans la revue *Circuit Cellar* (n°231, octobre 2009).

Liens

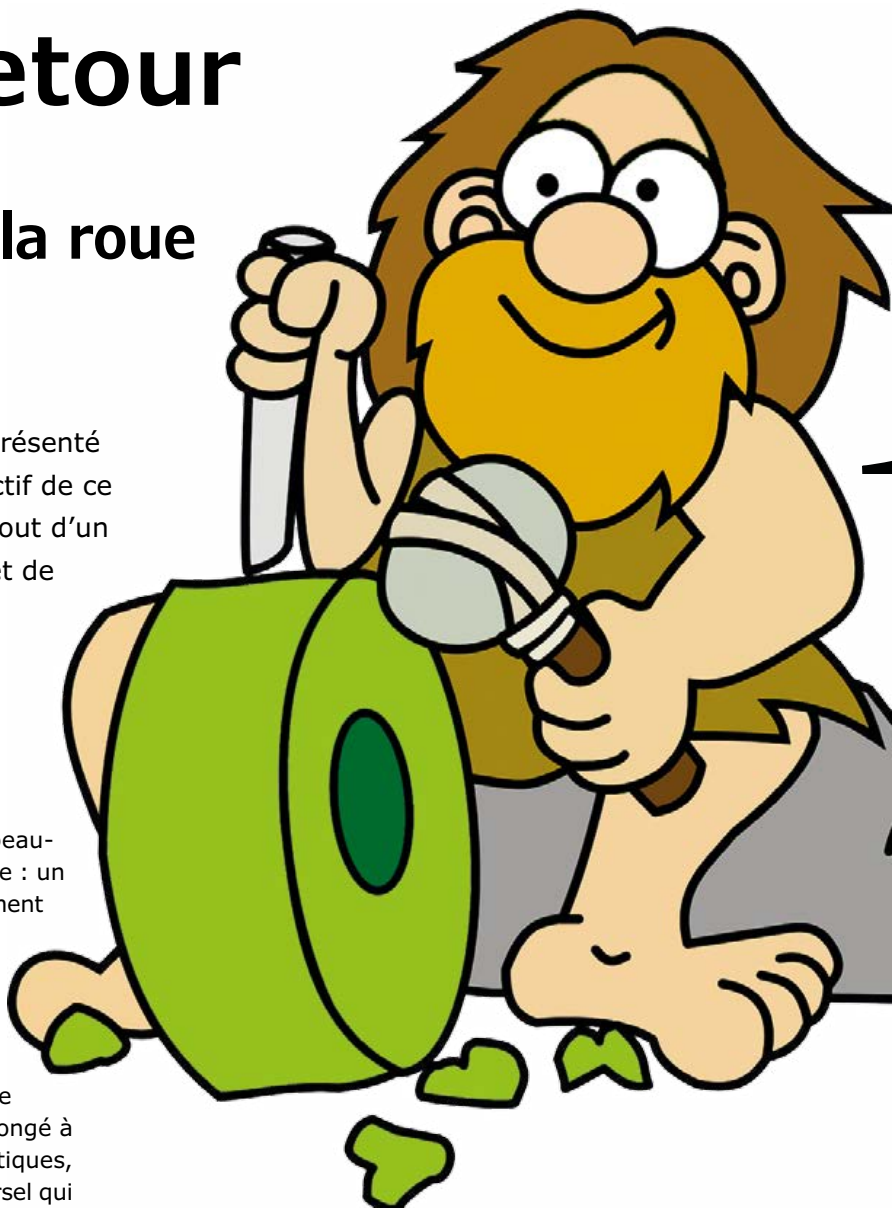
- [1] *Hors-circuits – le filtrage numérique sans stress : les filtres FIR*, Elektor 01/2016, www.elektormagazine.fr/150635
- [2] *CIC Filter Introduction*, Matthew P. Donadio, <http://users.snip.net/~donadio/cic.pdf>
- [3] *Understanding CIC Compensation Filters*, note d'application 455, avril 2007, Altera Corporation, www.altera.com/literature/an/an455.pdf
- [4] *Understanding cascaded integrator-comb filters*, Richard Lyon, Embedded Systems programming, www.embedded.com/columns/technicalinsights/160400592?_requestid=511057
- [5] www.scilab.org

Platino, le retour par pitié, arrêtez de réinventer la roue

Clemens Valens (labo d'Elektor)

C'est en octobre 2011 que nous vous avons présenté Platino. Ouvertement inspiré d'Arduino, l'objectif de ce circuit imprimé polyvalent était de faciliter l'ajout d'un microcontrôleur, d'un afficheur, de poussoirs et de codeurs rotatifs à un projet. Les années ayant prouvé le bien-fondé du concept Platino, nous avons décidé d'améliorer la carte.

Pour rappel, la carte Platino est née du constat que beaucoup de projets à microcontrôleur ont la même ossature : un microcontrôleur, un LCD, quelques boutons (habituellement quatre) et un étage d'entrée analogique. Cet étage diffère selon les projets, mais le reste – abstraction faite du modèle du contrôleur – se retrouve plus ou moins à l'identique à chaque fois. De plus nous avons remarqué que chaque concepteur tendait à écrire sa propre interface utilisateur en partant de zéro, une perte de temps et d'énergie à nos yeux. Nous avons donc songé à une solution répondant à la plupart des situations pratiques, et plutôt à une carte polyvalente qu'à un « circuit universel qui



Caractéristiques de Platino v1.4

- Compatible ATmega8, ATmega16, ATmega32, ATmega48, ATmega88, ATmega164, ATmega168, ATmega324, ATmega328, ATmega644, ATmega1284
- LCD alphanumérique 2 x 16, 4 x 16 ou 4 x 20 car.
- Jusqu'à quatre poussoirs
- Jusqu'à deux codeurs rotatifs
- Buzzer et LED RGB
- Régulateurs de tension embarqués de 5 V et 3,3 V
- Clavier séparable
- Dimensions optimisées pour le boîtier Bopla 26160000
- Paquet Platino pour l'EDI Arduino
- Bibliothèque et chargeurs d'amorçage compatibles Arduino pour tous les µC (sauf ATmega48)
- Connecteurs d'extension compatibles avec les shields Arduino
- Embase pour connecteur FTDI ; permet la programmation de l'ATmega328 depuis l'EDI Arduino et fournit une interface de communication série
- Code source ouvert, matériel ouvert

peut tout faire ». Résultat, Platino [1] avec des spécifications satisfaites sur toute la ligne :

- microcontrôleur AVR ATmega d'Atmel en boîtier DIP à 28 ou 40 broches
- compatible Arduino
- LCD alphanumérique 2 x 16, 4 x 16 ou 4 x 20 car.
- jusqu'à quatre poussoirs
- jusqu'à deux codeurs rotatifs
- buzzer
- LED RGB
- régulateur de tension de 5 V embarqué
- connecteurs d'extension
- clavier séparable
- dimensions optimisées pour le boîtier Bopla modèle 26160000

Nous tenons à ce que notre carte soit compatible avec Arduino car cet environnement offre un grand nombre de bibliothèques et d'outils gratuits et faciles à utiliser. Nous avons donc créé pour Platino une bibliothèque pour l'EDI Arduino, des chargeurs de démarrage pour tous les contrôleurs compatibles

OUI, ON DIRAIT LA MÊME
MAIS CELLE-LÀ
ELLE EST VERTE !



Arduino, et expliqué en détail comment intégrer le tout dans l'EDI Arduino de l'époque [2].

La situation ne s'améliore pas

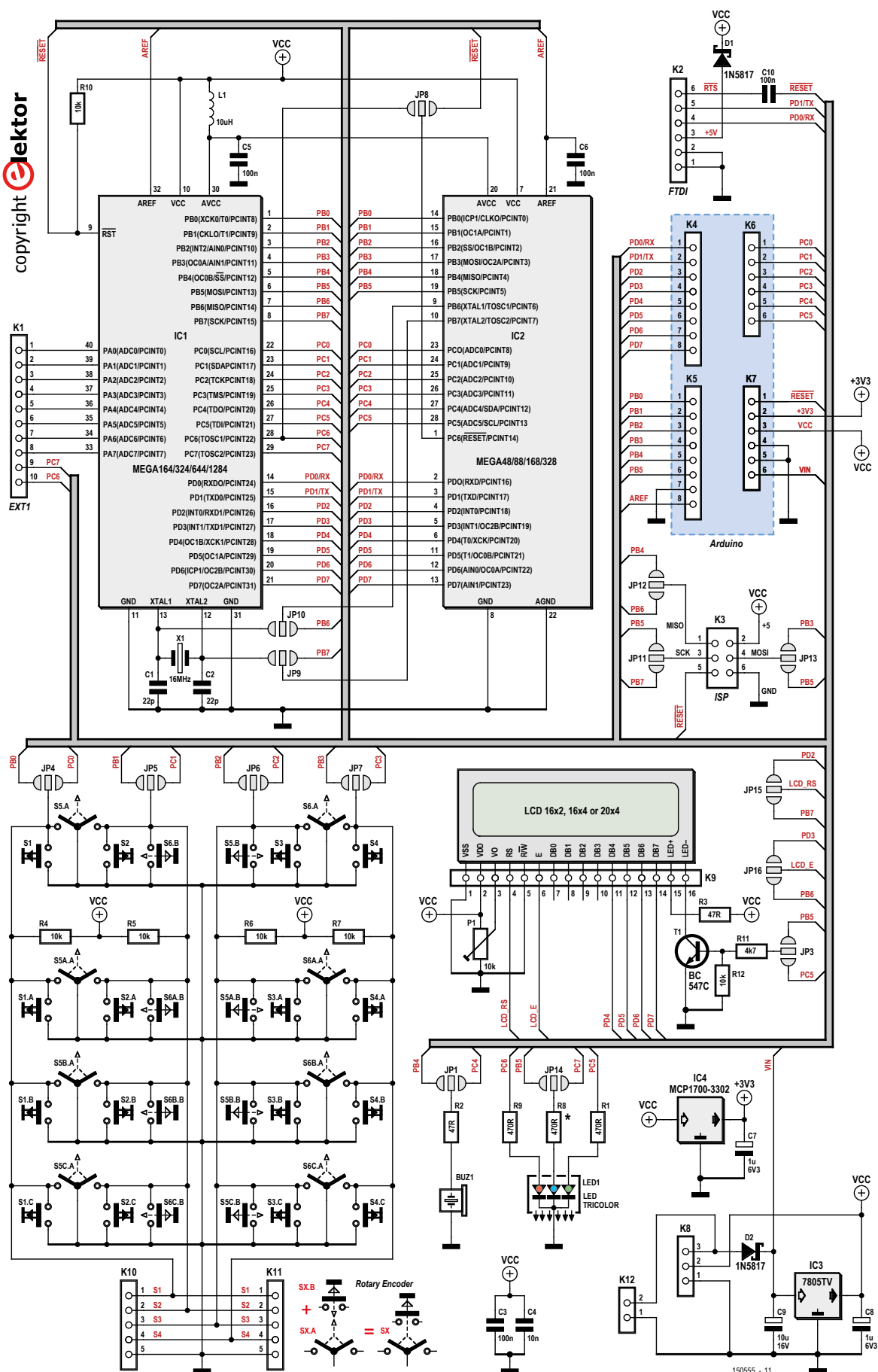
De l'eau a coulé sous les ponts depuis la naissance de Platino en 2011. La popularité d'Arduino ne s'est pas démentie, bien au contraire, et Platino a trouvé sa place dans de nombreux projets. Cependant, presque cinq ans plus tard, le constat qui nous avait poussés à créer Platino demeure. La situation a même empiré à nos yeux, car l'époque où les concepteurs élaboraient d'élégants circuits autour de leur microcontrôleur préféré semble révolue. Aujourd'hui pullulent des empilements de *shields* enfichés à qui mieux mieux sur des cartes Arduino, le tout, pour ne pas dire le tas, donnant d'affreux circuits, mal conçus, pleins de fonctions redondantes et uniquement adaptés à des boîtiers faits sur mesure. Souvenez-vous, Arduino Uno n'est rien d'autre qu'une carte de liaison (une BoB, *break-out-board*) à ATmega328, dotée d'un convertisseur série-USB et à l'encombrement peu pratique.

Platino revue et corrigée

Si le temps nous a permis d'enrichir notre expérience avec Platino, il a aussi révélé quelques problèmes. Rien de grave,

rien d'urgent, leur résolution pouvait attendre. L'occasion s'est présentée lorsqu'une ingénieure du labo travaillant sur un analyseur de protocole de communication série a réalisé que la carte Platino de son instrument lui faciliterait grandement la vie si l'accès au second port série disponible sur la broche 40 des AVR était amélioré. Peu de temps auparavant était sortie la version 1.6.2 d'Arduino avec un *Gestionnaire de carte* qui simplifiait l'installation des cartes indépendantes comme Platino. Cette conjonction des astres nous a décidés à rénover Platino, sa bibliothèque et ses chargeurs de démarrage. Voici les éléments revus (référez-vous au schéma de la **fig. 1**) :

Régulateur 3,3 V embarqué – Platino n'avait à l'origine qu'un régulateur de 5 V, mais les connecteurs pour *shields* Arduino ont toujours eu une broche de 5 V et une de 3,3 V. Nous pensions que si vous aviez besoin d'une carte alimentée en 3,3 V, vous pouviez soit remplacer le régulateur de 5 V par un de 3,3 V, soit ajouter une alim de 3,3 V à un *shield*. Un cavalier (JP2) permettait de connecter la broche 3,3 V du connecteur du *shield* à l'alimentation de la carte. JP2 a disparu, la nouvelle carte a un régulateur de 3,3 V dédié à l'alimenta-

copyright **elektor**

tion du connecteur de 3,3 V du *shield*. Cette modification rend Platino 100 % compatible Arduino.

Plus de conflits d'alimentation – un câble adaptateur série-USB est nécessaire pour programmer Platino « à la Arduino », d'où la présence d'un connecteur (K2) compatible avec les câbles FTDI. Ce type de câble délivre toutefois également 5 V via une broche qui était reliée directement à l'alimentation V_{CC} de la carte. Nous avons placé une diode Schottky en série avec cette broche de façon à éviter toute connexion directe avec la sortie du régulateur de 5 V de Platino. Incidemment, le nouveau bornier K12 rend plus pratique la connexion avec une alimentation externe.

Un deuxième port série – les puces ATmega à 28 broches n'ont qu'un seul récepteur-émetteur universel (a)synchrone (USART) – l'objet *Serial* en langage Arduino – relié aux broches PD0 (RXD) et PD1 (TXD). Les puces à 40 broches en ont toutefois deux (*Serial0* et *Serial1* selon la dénomination Arduino), le second est relié à PD2 (RXD1) et PD3 (TXD1). Sur l'ancienne carte, PD2 et PD3 étaient utilisées pour les signaux RS et E du LCD, et empêchaient donc l'usage simultané de l'autre USART. Les cavaliers JP15 et JP16 de la nouvelle Platino permettent de relier le signal RS du LCD à PB7 (JP15) et le signal E à PB6 (JP16). Sur les puces à 28 broches, PB6 et PB7 sont souvent occupées par le quartz, donc dans ce cas vous configurerez sans doute la carte de façon à utiliser les anciennes connexions. Sur les puces à 40 broches, PB6 et PB7 sont libres, seulement utilisées pour la programmation *in situ* (ISP).

Circuit d'initialisation – Sur les cartes pré-Uno, le signal d'initialisation (*reset*) émis par l'EDI pouvait provenir soit du signal DTR du port série, soit de son signal RTS. Dans le cas du signal RTS, une impulsion était créée et une résistance sur la ligne *Reset* du μC suffisait. Toutefois, le câble FTDI ne transmet que le signal RTS. Comme ce signal est actif en même temps que le port série est aussi actif, nous avons utilisé un condensateur afin de n'utiliser que le front descendant du signal pour initialiser le μC . Pour gérer les deux cas, l'ancienne Platino avait R13, marquée comme résistance, mais avec une double empreinte résistance/condensateur. Nous avons abandonné la prise en charge DTR et remplacé R13 par C10, toujours un condensateur.

Sérigraphie – bien que nous ayons été tentés d'ajouter des motifs tribaux à Platino et d'en faire la première carte à poils, nous n'avons pas cédé à la mode des barbes et tatouages : nous nous sommes contentés d'imprimer des indications claires et pratiques sur les deux faces. Nous avons en particulier marqué la position et la fonction des cavaliers à souder. Les majuscules A, C ou D désignent le port auquel se connecte un cavalier (**tableau 1**).

Nous avons également revu le dessin de la carte afin de résoudre les problèmes de communication qui survenaient à haut débit,

Figure 1. Le schéma semble compliqué en raison des nombreuses options de configuration de Platino. Même si vous voyez 24 poussoirs, vous pouvez n'en monter que 4 !

Un mot sur l'ATmega48

Avec seulement 4 Ko de mémoire flash et 512 octets de RAM, l'ATmega48 est le petit rabougri de la famille ATmega. A priori compatible avec Platino puisqu'il s'agit d'un contrôleur AVR à 28 broches, sa (trop) petite mémoire n'a toutefois pas d'espace réservé à un chargeur d'amorçage. Vous ne pourrez donc l'utiliser avec l'EDI Arduino que si vous avez un programmeur ISP AVR reconnu par l'EDI. Le bouton *Téléverser* ne fonctionne pas avec ce μC .

en particulier avec les chargeurs d'amorçage de certains contrôleurs moins courants. Désormais tous les *bootloaders* fonctionnent parfaitement à 115200 bauds.

Le paquet Platino

Comme Arduino, Platino est bien plus qu'une carte. C'est une alchimie matérielle et logicielle pouvant piloter toutes sortes d'applications. Le logiciel est un ingrédient de la polyvalence de Platino, voyons ce qui a été fait de ce côté-là.

Arduino, qui a considérablement évolué au fil des ans, est désormais compatible avec de nombreuses chaînes d'outils, architectures de processeurs et cartes de différents fabricants. L'EDI a notamment été retravaillé pour faciliter la gestion des cartes d'extension indépendantes. Nous avons maintenant un outil bien pratique, le *Gestionnaire de carte*, disponible depuis le menu *Outils/Type de carte*.

Le *Gestionnaire de carte* permet d'installer directement dans l'EDI le paquet d'une carte indépendante. Autrement dit plus de fichiers à copier dans différents dossiers, plus de fichier *boards.txt* à modifier, le Gestionnaire s'occupe de tout, du moment que nous lui fournissons le paquet attendu. Un paquet installé peut être désinstallé sans laisser de traces (du moins, espérons-le).

Tableau 1. Cavaliers.
Placez-les avant de monter les composants.

cavalier	fonction	position 1	position 2
JP1	buzzer	PB4	PC4
JP2	supprimé		
JP3	rétroéclairage LCD	PB5	PC7
JP4	S1/S5.A	PB0	PC0
JP5	S2/S5.B	PB1	PC1
JP6	S3/S6.A	PB2	PC2
JP7	S4/S6.B	PB3	PC3
JP8	PC6 (28 broches)	Reset	PC6
JP9	PB7 (28 br. uniquement)	quartz	PB7
JP10	PB6 (28 br. uniquement)	quartz	PB6
JP11	ISP SCK	PB5 (28 br.)	PB7 (40 br.)
JP12	ISP MISO	PB4 (28 br.)	PB6 (40 br.)
JP13	ISP MOSI	PB3 (28 br.)	PB5 (40 br.)
JP14	LED	PB5	PC7
JP15	LCD RS	PD2	PB7
JP16	LCD E	PD3	PB6

Liste des composants

Résistances

Par défaut : 5%, 0,25 W

R2, R3 = 47 Ω

R1, R8, R9 = 470 Ω

R11 = 4,7 k Ω

R4, R5, R6, R7, R10, R12 = 10 k Ω

P1 = ajustable 10 k Ω , horizontal

Condensateurs

C1, C2 = 22 pF, pas de 2,5 mm

C4 = 10 nF, pas de 2,5 mm

C3, C5, C6, C10 = 100 nF, pas de 2,5 mm ou 5 mm

C7, C8 = 1 μ F, 50 V, pas de 2,5 mm

C9 = 10 μ F, 50 V, pas de 2,5 mm

Inductance

L1 = 10 μ H

Semi-conducteurs

D1, D2 = 1N5817

LED1 = LED RGB, 5 mm, cathode commune

T1 = BC547C

IC3 = MC7805

IC4 = MCP1700-3302E/TO

Divers

IC1 = support CI DIP40

IC2 = support CI étroit DIP28 (largeur 7,6 mm)

X1 = quartz, 16 MHz, 18 pF

BUZ1 = buzzer, 12 mm

S1, S2, S3, S4 = bouton-poussoir, Multimec RA3FTH9

S5, S6 = codeur rotatif, Alps EC12E2424407 ou équivalent

K1 = embase femelle 10 voies, pas de 2,5 mm, verticale

K2 = barrette à 6 broches, pas de 2,5 mm, verticale

K3 = barrette à 6 broches (2x3), pas de 2,5 mm, verticale

K4, K5 = embase femelle à 8 voies, pas de 2,5 mm, verticale

K6, K7 = embase femelle à 6 voies, pas de 2,5 mm, verticale

K8 = embase mâle à 3 voies, pas de 2,5 mm, verticale

K9 = embase femelle à 16 voies, pas de 2,5 mm, verticale

K12 = bornier pour CI à 2 voies, pas de 3,5 mm

Circuit imprimé Platino v1.4 (réf. 150555-1 sur elektor.fr)

Boîtier Bopla type 26160000

MATÉRIEL DISPONIBLE SUR L'E-CHOPPE

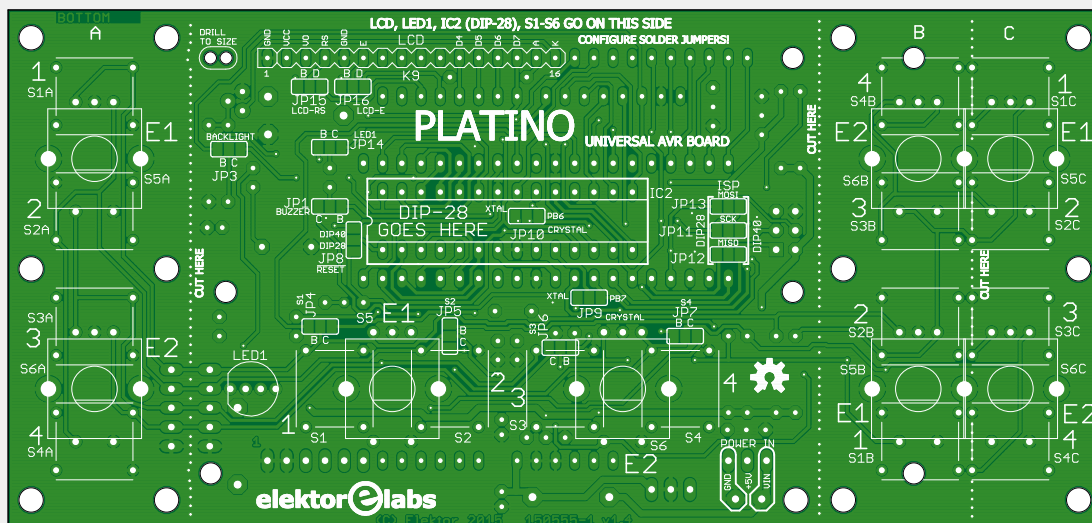
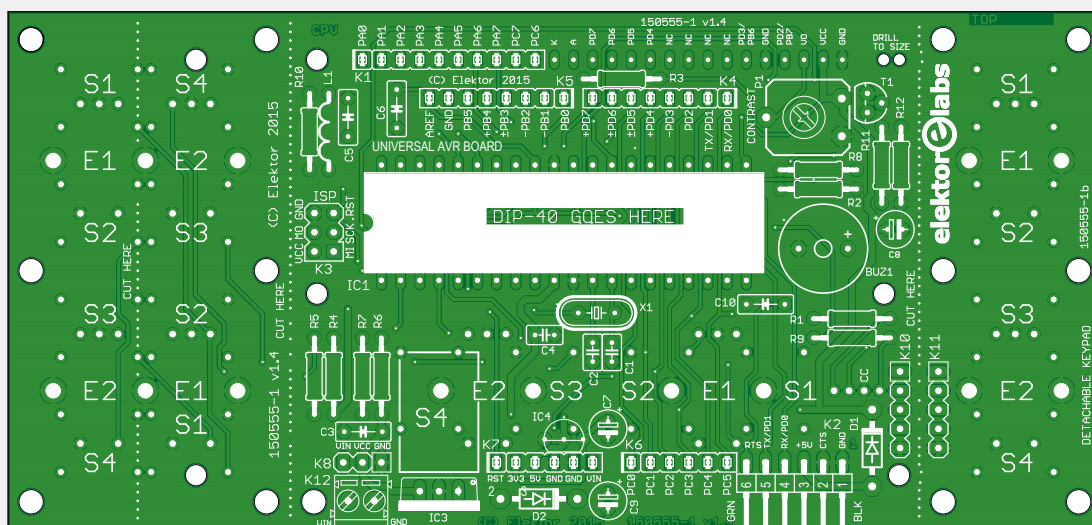
150555-1 – circuit imprimé, sans composants

140433-1 – circuit imprimé de la carte d'extension

Platino, sans composants

129009-1 – circuit imprimé de la platine polyvalente

Arduino, sans composants



Un paquet contient un fichier JSON regroupant les informations relatives à une ou plusieurs cartes, ainsi qu'un fichier ZIP contenant tout ce dont a besoin l'EDI pour compiler les croquis (*sketchs*) d'une carte particulière. L'utilisateur fournit le fichier JSON au Gestionnaire, qui télécharge ensuite le ZIP, vérifie les sommes de contrôle et installe le paquet. La carte nouvellement installée est alors disponible depuis le menu *Outils/Type de carte*.

Elektor a créé deux paquets, un pour Arduino 1.6.5, l'autre pour 1.6.6 et sup. Vous les trouverez sur la page Github d'Elektor.Labs [3]. Notez que le Gestionnaire est (pour l'instant) absent de l'EDI Arduino maintenu par Arduino.org (v1.7.x), donc choisissez l'EDI d'Arduino.cc. Ouvrez le menu *Fichier/Préférences*, et dans la boîte *URL de gestionnaire de cartes supplémentaires* entrez (à l'identique !) :

https://raw.githubusercontent.com/ElektorLabs/arduino/master/package_elektor-labs.com_ide-1.6.6_index.json

(Vous pouvez aussi télécharger le fichier JSON sur un disque local et entrer son chemin d'accès préfixé par *file:///*).

Fermez la fenêtre des préférences, ouvrez le *Gestionnaire de carte*, et sélectionnez *Partenaire* dans le menu déroulant *Type* (en haut et à gauche de la fenêtre). Si l'option *Partenaire* n'apparaît pas, fermez et relancez le Gestionnaire, puis re-sélectionnez *Partenaire*. Vous devriez voir une entrée *Elektor AVR boards by Elektor.Labs*. Cliquez dessus, puis sur le bouton *Installer*. Cliquez sur *Fermer*, ouvrez à nouveau le menu *Outils/Type de carte* et sélectionnez *Platino*. Sélectionnez ensuite le processeur qui équipe votre Platino depuis *Outils/Processeur*. Attention, certains AVR ont des types différents (A, P, PA, rien), vous devez choisir le bon.

Et voilà, votre carte est prête à être programmée avec le plein soutien des bibliothèques et chargeurs d'amorçage installés. Notez que la bibliothèque Platino ajoute du code à vos croquis même si vous ne faites appel à aucune de ses fonctions. Si vous ne voulez pas l'utiliser, sélectionnez *Platino without the library* depuis *Outils/Type de carte*.

Pense-bête pour l'assemblage

La configuration et l'ajout de composants sur Platino dépendent de votre projet et sont de votre ressort. Voici une liste de points de contrôle qui vous aidera à éviter les erreurs courantes (surtout si vous en respectez l'ordre) :

1. Si vous souhaitez utiliser Platino avec Arduino, songez à vous procurer un câble série-USB FTDI de 5 V. Pour vos tests/débugages, nous vous conseillons aussi de monter au moins une LED reliée à R8, et de placer le cavalier JP14 sur la position « B ». Une autre possibilité est d'utiliser le rétroéclairage du LCD si vous avez placé le cavalier JP3 en position « B ».
2. Prenez le temps de bien réfléchir à la configuration de votre système, en particulier au processeur et aux périphériques dont vous aurez besoin.
3. Notez le placement de vos cavaliers à l'aide du **tableau 1**.
4. Placez les cavaliers. Tous sont montés sur la face soudure de la carte. Placez-les tous, même si certains ne vous servent pas.
5. Montez les composants. Rien d'ardu, mais il y a quelques pièges à éviter :
 - les μC à 28 broches (IC2) sont montés du même côté que le LCD, les boutons et la LED. Tous les autres composants, y

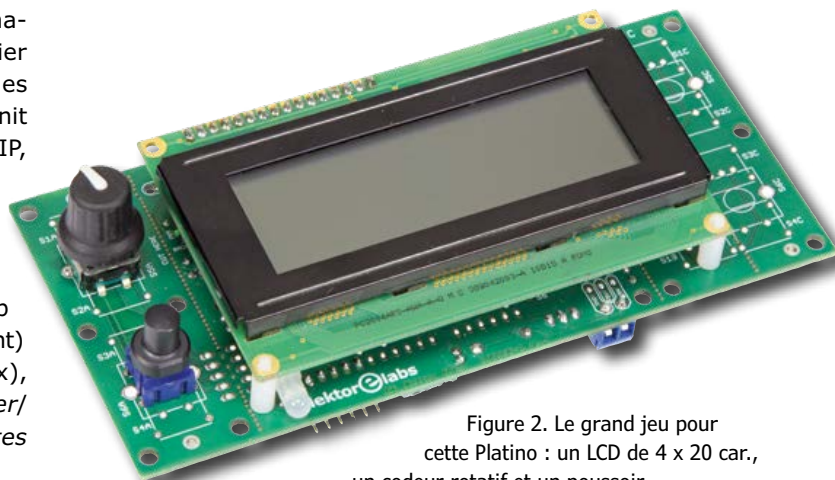


Figure 2. Le grand jeu pour cette Platino : un LCD de 4 x 20 car., un codeur rotatif et un poussoir.

compris le μC à 40 broches (IC1), se montent sur l'autre face.

- Le cavalier JP10 est à l'intérieur de l'empreinte IC2 du μC à 28 broches, donc placez-le avant de monter (une embase pour) le μC .
- Le boîtier métallique du quartz X1 peut mettre en court-circuit les traces qui passent dessous. En théorie le vernis vert prévient ce risque, mais pour plus de sécurité vous pouvez fixer le quartz sur le circuit avec de l'adhésif double-face avant de le souder.
- Si vous montez les poussoirs et les codeurs rotatifs sous un LCD 2x16 car., faites-le avant de placer IC3, IC4 et C7 sur l'autre face.
- IC3, C7, C8 et C9 sont relativement hauts et peuvent interférer avec des cartes d'extension s'ils sont montés verticalement. Nous avons donc laissé de l'espace pour un montage horizontal. Pliez les fils des composants avant de les souder, le montage sera propre.
- T1 pilote le rétroéclairage du LCD. Si vous ne l'utilisez pas, ne montez ni T1, ni R11/R12, leur absence évitera les interfé-

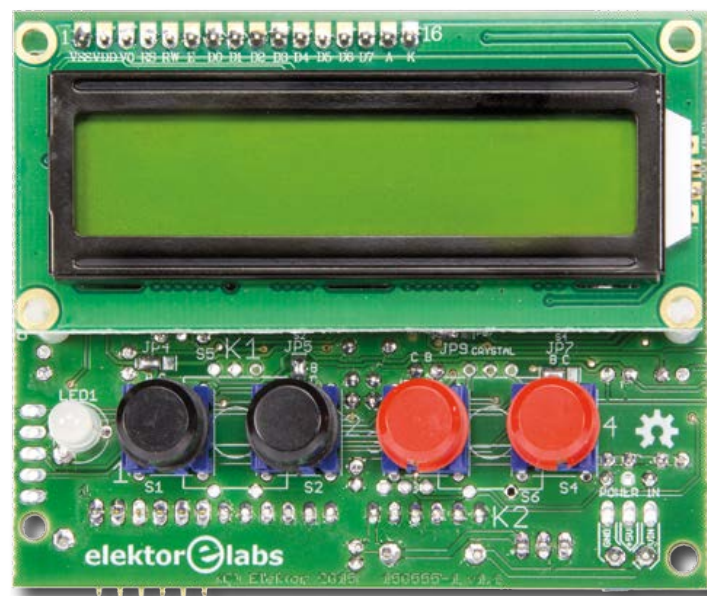
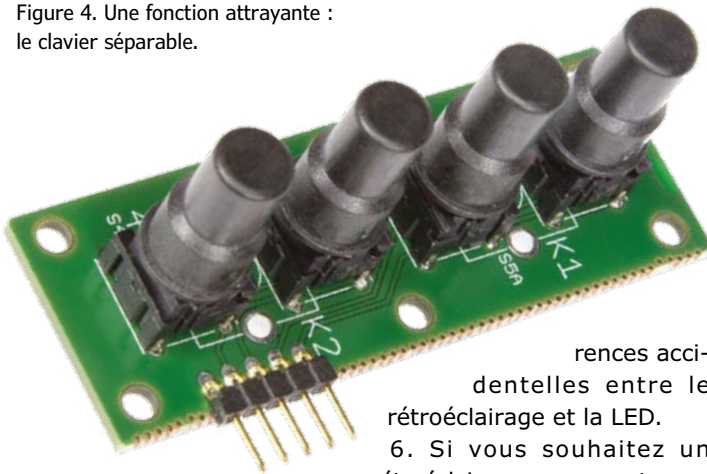


Figure 3. Vous pouvez couper les parties superflues de la carte pour créer un système compact avec un LCD 2 x 16 car. et quatre poussoirs.

Figure 4. Une fonction attrayante : le clavier séparable.



rences accidentelles entre le rétroéclairage et la LED.

6. Si vous souhaitez un rétroéclairage permanent, soudez un fil (ou placez un cavalier) entre l'émetteur et le collecteur de T1.

Si vous souhaitez utiliser Platino avec Arduino, vous devez d'abord programmer le chargeur d'amorçage dans le µC. Vous pouvez le faire depuis l'EDI Arduino si vous avez installé notre paquet Platino. Il vous faudra un programmeur AVR (qui peut être une carte compatible Arduino, Platino p. ex., programmée avec le croquis *Exemples/11.Arduino.ISP*). Sélectionnez d'abord le µC cible de la Platino depuis *Type de carte* et *Processeur* du menu *Outils*, puis sélectionnez votre programmeur sous *Outils/Programmeur*. Reliez le programmeur au connecteur ISP (K3) de Platino, et cliquez sur *Outils/Graver la séquence d'initialisation*.

7. Déconnectez le programmeur et reliez le câble FTDI. Sélectionnez le port correct depuis *Outils/Port*. Si vous avez monté la LED de débogage comme sugg-

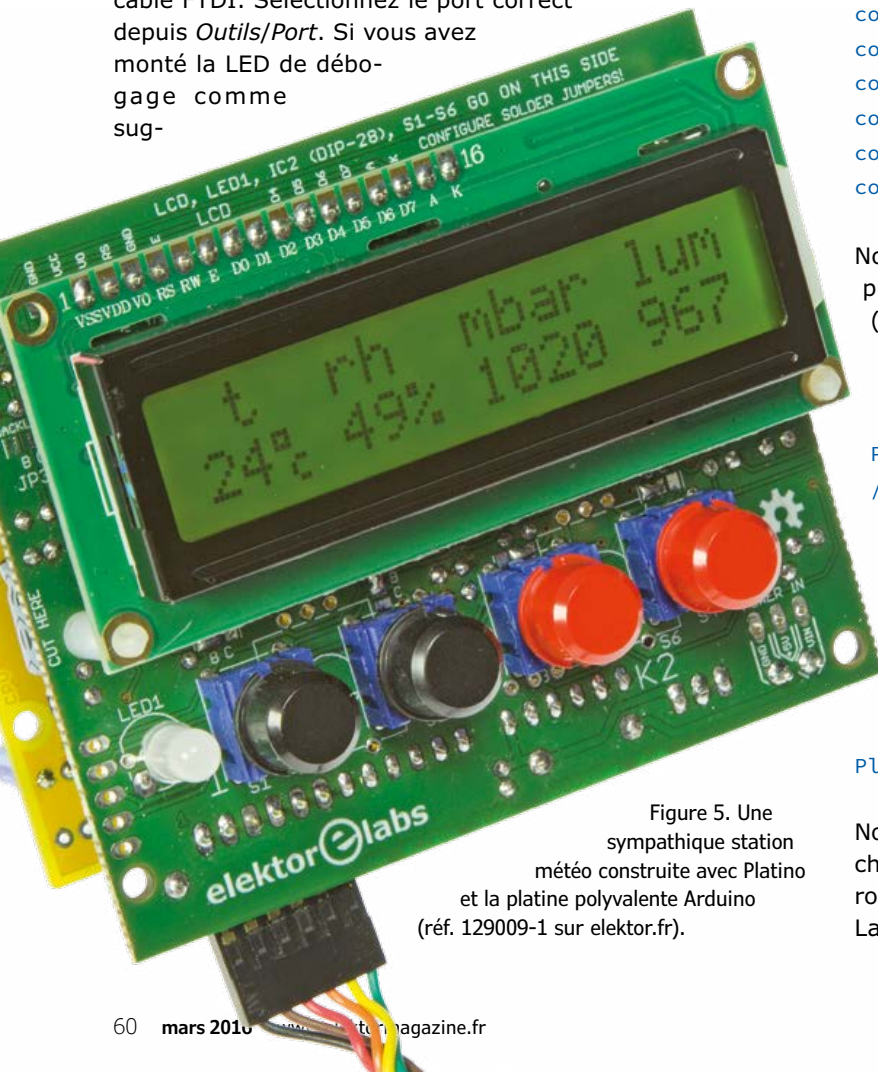


Figure 5. Une sympathique station météo construite avec Platino et la platine polyvalente Arduino (réf. 129009-1 sur elektor.fr).

géré au point 1, chargez le croquis *Exemples/01.Basics/Blink*. Si tout s'est bien déroulé, la LED clignote à une fréquence d'un demi-hertz.

La bibliothèque Platino

Nous avons écrit une bibliothèque qui facilite l'intégration des périphériques de Platino. Vous pouvez télécharger son code source depuis GitHub [4]. Nous y avons ajouté un croquis d'exemple. L'installation manuelle de la bibliothèque est déconseillée, préférez le paquet Platino décrit ci-dessus, plus simple et plus rapide !

Une fois installé, le paquet Platino est intégré à l'EDI et il n'est donc plus nécessaire d'inclure des fichiers d'en-tête « Platino » dans les croquis.

La bibliothèque fournit un objet `Platino` utilisable dans les croquis, mais auparavant nous devons préciser la configuration des cavaliers. Pour cela nous utilisons la commande `begin` :

```
// Configurer Platino.
Platino.begin(jp1,jp3,jp4,jp5,jp6,jp7,jp14,jp15,jp16);
```

Les paramètres `jpX` représentent les cavaliers. Pour une meilleure lisibilité du code, nous les déclarons comme constantes au début du croquis, avant la commande `begin` :

```
// Constantes pour la déclaration des cavaliers
const uint8_t jp1 = 'B'; // buzzer
const uint8_t jp3 = 'C'; // rétroéclairage
const uint8_t jp4 = 'B'; // S1
const uint8_t jp5 = 'B'; // S2
const uint8_t jp6 = 'B'; // S3
const uint8_t jp7 = 'B'; // S4
const uint8_t jp14 = 'B'; // LED
const uint8_t jp15 = 'D'; // LCD RS
const uint8_t jp16 = 'D'; // LCD E
```

Nous devons également indiquer à la bibliothèque quels périphériques sont présents sur la carte, à savoir les poussoirs (s'il y en a), le LCD, sa taille, etc. Pour cela nous utilisons les fonctions `hasXXX` :

```
// Activation des périphériques de la carte
Platino.hasDisplay(4,20,true,true);
// a un LCD 4x20, montre l'écran d'accueil Platino
Platino.hasBacklight(); // a un rétroéclairage
Platino.hasKnob(1);
// codeur rotatif (avec poussoir)
Platino.hasPushbutton(3);
// poussoir 3 ou poussoir du codeur rotatif 1
Platino.hasPushbutton(4); // poussoir 4
Platino.hasLedRgb(); // LED RGB
Platino.hasBuzzer(); // buzzer
```

Nous avons privilégié la simplicité et la lisibilité du code lors du choix du nom des fonctions, raison pour laquelle les codeurs rotatifs, p. ex., sont simplement appelés *knobs* (boutons). La valeur par défaut des fonctions `hasXXX` est `true` si aucun

argument ne leur est transmis, autrement dit le périphérique XXX est supposé être présent par défaut. Par souci de clarté, vous pouvez signaler dans le code (et à la bibliothèque) qu'un périphérique donné est absent :

```
Platino.hasBacklight(false); // câblé ou absent
```

Une fois le matériel déclaré, vous pouvez commencer à l'utiliser :

```
Platino.backlight(true); // rétroéclairage allumé
Platino.ledRgb(0,0,0); // LED RGB éteinte
Platino.display.clear(); // effacer l'écran
Platino.bEEP(1000,100); // 1 kHz, 100 ms
```

Notez la syntaxe de l'appel à la fonction `clear()`. La bibliothèque Platino utilise pour le LCD les fonctions de la bibliothèque Arduino *LiquidCrystal*, et pour les appeler il faut utiliser le préfixe `Platino.display`.

Le rebond des contacts des poussoirs ou codeurs (*knobs*) est traité par la bibliothèque, donc vous pouvez directement lire un bouton activé, comme ici :

```
// Envoie la nouvelle valeur du codeur rotatif 1
// au traceur série d'Arduino
if (Platino.knobChanged(1)==true) Serial.
println(Platino.knobRead(1));
// Remise à zéro du compteur du codeur 1
// après pression du poussoir 3
if (Platino.pushbuttonRead(3)==PUSHBUTTON_DOWN)
Platino.knobWrite(1,0);
```

Après lecture, un poussoir est mis par défaut à l'état `PUSHBUTTON_IDLE` car c'est ce qu'il y a de plus pratique pour les détections simples de pression. Si toutefois vous voulez faire quelque chose lorsqu'un bouton est pressé, son état ne doit pas être effacé après sa lecture. Dans ce cas, vous devez mettre le deuxième argument sur `false` :

```
while (Platino.pushbuttonRead(3,false)==PUSHBUTTON_
DOWN) faire_un_truc();
// Effacer l'état du bouton une fois l'action terminée
pushbuttonClear(3);
```

Le prétraitement des poussoirs se fait dans la fonction `tick` appelée par la routine d'interruption `Timer0`, jamais dans un croquis. Cette routine décompte des microsecondes et des millisecondes et est utilisée par les fonctions Arduino `millis`, `micros` et `delay` (pas par `delayMicroseconds`) ; normalement vous ne devriez donc pas y toucher. Ce n'est que si vous modifiez `Timer0` que vous devrez appeler vous-même `Platino.tick`, de préférence à une fréquence d'environ 1 kHz.

Notez qu'un codeur rotatif basique équivaut à deux poussoirs. Si un codeur rotatif a aussi un poussoir, il équivaut à trois poussoirs. Par conséquent vous pouvez avoir :

- 0 codeur et jusqu'à 4 poussoirs.
- 1 codeur et jusqu'à 2 poussoirs.

- 2 codeurs sans poussoir et pas d'autres boutons-poussoirs.

La LED RVB, enfin, est commandée de façon « binaire », c.-à-d. qu'une couleur ne peut être qu'allumée ou éteinte. La bibliothèque Platino ne dispose pas de fonction MLI (*PWM*), il vous faudra l'écrire vous-même si vous souhaitez jouer avec plus de trois couleurs.

Comme celles de la vraie vie, notre bibliothèque est un bon endroit pour en savoir plus. Consultez donc son code source. Le fichier *Platino.h* est un bon point de départ.

Elle n'attend plus que vous

La carte Platino est une plateforme idéale pour les projets

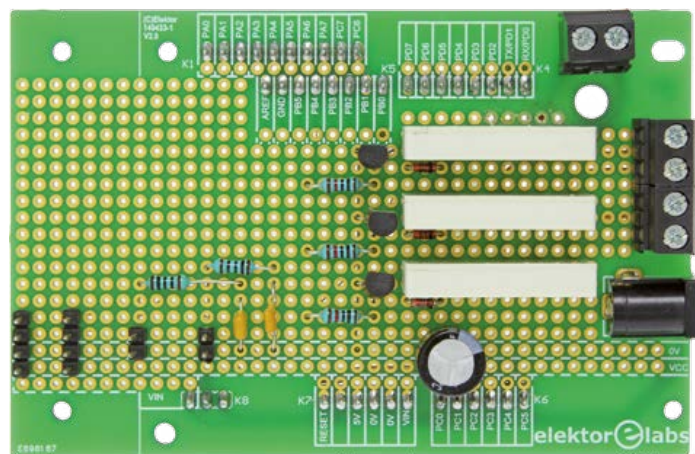


Figure 6. La carte d'extension Platino, pratique pour expérimenter et assembler des applications maison.



Arduino nécessitant une interface humaine. C'est aussi une carte à microcontrôleur modulaire, polyvalente, riche d'options à exploiter et explorer au moyen d'autres outils et langages. Quel projet est votre projet ? ◀

(150555 – version française : Hervé Moreau)

Liens

- [1] www.elektormagazine.fr/100892
- [2] www.elektormagazine.fr/120094
- [3] <https://github.com/ElektorLabs/Arduino>
- [4] <https://github.com/ElektorLabs/150555-Platino/>
- [5] www.elektormagazine.fr/150555
- [6] www.elektor-labs.com/platino



bienvenue dans votre e-choppe

Elektor recommande



Kit Tiny Tesla

Sur l'internet, je trouve souvent des vidéos de gros générateurs Tesla qui produisent des étincelles incroyablement longues, semblables à des éclairs. Construire un tel dispositif est fastidieux et son utilisation n'est pas sans risque, je ne me suis jamais lancé dans un tel projet. Mais lorsque j'ai vu Tiny Tesla, cette mini-version de transformateur Tesla, je me suis dit : « il me le faut ». Ce petit générateur est

fourni sous forme d'un kit dont une bobine secondaire avec ses 600 spires enroulées sur un support transparent. Il suffit de bobiner quelques spires sur la bobine primaire. La carte est prête en un tournemain et la construction mécanique est très simple. Essayez... et oui, les étincelles s'envolent jusqu'à 10 cm.

Vive les jouets électroniques !

Harry Baggen, labo d'Elektor

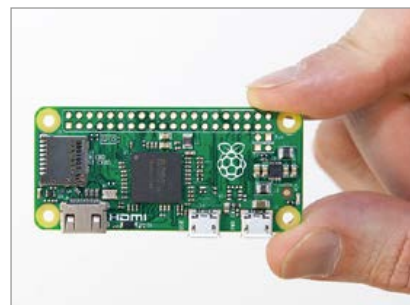
www.elektor.fr/tinytesla-kit



Vos favoris :

1. Raspberry Pi Zero

www.elektor.fr/raspberry-pi-zero



2. LabNation SmartScope

www.elektor.fr/labnation-smartscope

3. Abonnement Formule Green

www.elektor.fr/green

4. Domotique avec Zelio Logic

www.elektor.fr/domo-zelio

5. Red Pitaya V1.1

www.elektor.fr/red-pitaya-instrument

6. T-Board bien ARMée

www.elektor.fr/t-board-arm

7. Red Pitaya V1.1

www.elektor.fr/red-pitaya

8. Carte d'E/S Android

www.elektor.fr/ardroid-i-o-board

DVD Elektor 2015



Ce DVD-ROM réunit tous les articles d'ELEKTOR parus au cours de l'année 2015, en français (mais aussi en anglais, allemand et néerlandais). Il contient le texte des articles ainsi que les schémas, et tous les dessins des circuits imprimés, sous forme de fichiers à haute résolution. Une fonction de recherche dans la table des matières vous permet de trouver immédiatement l'article souhaité.



Prix (membres) : 24,75 €

www.elektor.fr/dvd-2015

Bien entendu - itinéraire d'un audiophile



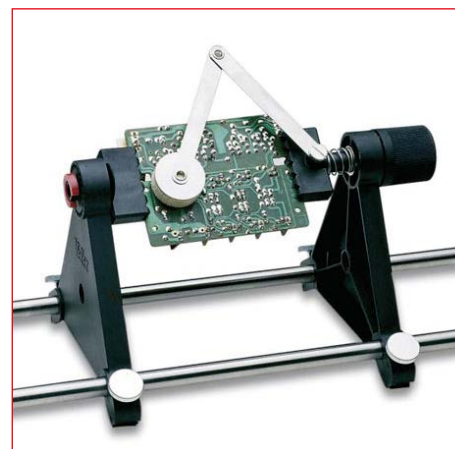
La musique ne se réduit pas à un paquet de bits comprimés, téléchargé furtivement puis écouté en *surround Dolby*. L'auteur analyse ici l'acte d'écouter et développe sur plus de cent pages une approche globale du lieu d'écoute. À quoi bon un matériel de rêve si on néglige le milieu dans lequel il est censé faire des merveilles ? Une autre centaine de pages s'intéresse aux enceintes acoustiques et une centaine enfin à l'électronique à tubes.



Prix (membres) : 47,70 €

www.elektor.fr/bien-entendu-french

Support pour circuit imprimé Weller



Le support pour circuit imprimé ESF-120 ESD de Weller est un châssis qui permet d'immobiliser un circuit imprimé sur lequel on veut monter et souder/dessouder des composants. Il est équipé d'un dispositif de serrage à ressort orientable sur 360° (par pas de 15°) et d'un bras mobile avec coussinet qui permet de maintenir en place les composants lorsqu'on retourne le circuit imprimé pour le soudage.



Prix (membres) : 71,96 €

www.elektor.fr/weller-pcb-holder



Ce DVD-ROM contient tous les numéros d'Elektor parus durant les années 2000 à 2009.

Elektor propose à ses lecteurs des montages électroniques de conception professionnelle et aisément reproductibles. N'hésitez pas à compléter votre schémathèque dans les domaines les plus divers de l'électronique et de l'informatique appliquées : alimentation, audio, auto, domestique, hautes fréquences, microcontrôleurs... !

Les 2200 articles d'Elektor réunis sur ce DVD-ROM sont présentés par ordre de parution (mois/année). Ils sont également classés par ordre alphabétique et par rubriques. Un index général permet d'effectuer des recherches dans tous les fichiers PDF du DVD-ROM même quand ils ne sont pas ouverts.



DVD-ROM Elektor 2000 à 2009

Retrouvez tous les articles parus dans Elektor de 2000 à 2009 !

Analyseur d'impédance

Carte d'extension pour Red Pitaya

e-choppe Elektor en ligne

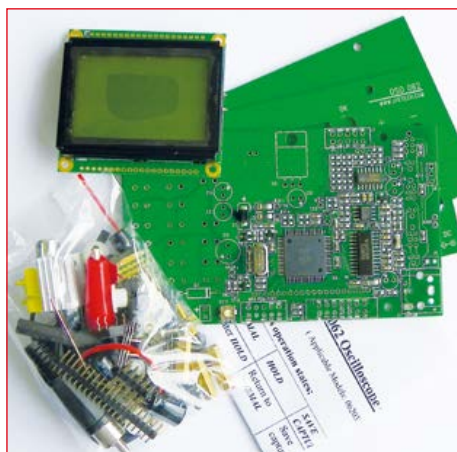
Votre panier d'achats pour l'électronique



Prix (membres) : 80,10 €

www.elektor.fr/dvd-elektor-2000-2009-fr

DS0062 - Oscilloscope à LCD en kit



DS0062 - Oscilloscope en kit : réalisez un oscilloscope avec écran à cristaux liquides. Ce kit contient le circuit imprimé DS0062 avec tous les composants CMS déjà soudés et le microcontrôleur pré-programmé (toutes les cartes sont testées avant leur emballage), ainsi que tous les composants traversants à monter sur la carte principale, toutes les pièces mécaniques (vis, entretoises), une sonde de mesure et la notice d'utilisation en anglais.



Prix (membres) : 48,56 €

www.elektor.fr/062-lcd-oscilloscope-diy-kit

Compilation BL600



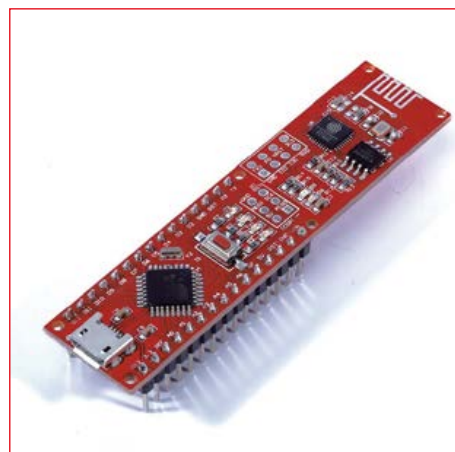
Compilation BL600 en PDF : retrouvez les huit articles parus dans Elektor sur le BL600, un module de communication radio à consommation ultra-faible. Avec la carte de liaison pré-assemblée, vous pourrez accéder facilement et rapidement aux principaux signaux du module BL600 (I2C et son capteur de température, SPI et son convertisseur numérique-analogique) pour réaliser un thermomètre extérieur sans fil ou bien un interrupteur 230 V commandé.



Prix (membres) : 7,16 €

www.elektor.fr/e-select-compilation-bl600

La carte IoT WiFi



La carte IoT WiFi est une carte de développement facile à utiliser et entièrement programmable, idéale pour le monde de l'Internet des Objets. Avec de simples commandes AT, vous pouvez par ex. lancer un serveur web, paramétrer le module en point d'accès ou établir une liaison avec un réseau domestique. À noter : interfaces I²C, UART, SPI ; mémoire Flash de 32 Ko ; EEPROM de 1 Ko ; 18 broches d'E/S à usage général (6 MLI, 6 sorties analogiques).



Prix (membres) : 26,96 €

www.elektor.fr/pretzel-board



par **Oliver Stanfield**

I bought my Egg-bot last year. I first read the news item and later saw a video and I just could not stop smiling! I was not in time for Easter but I can tell you: there's loads of occasions to put this little plotter to work, just think of birthday breakfasts, Christmas balls, glasses and even a personalized bowl.

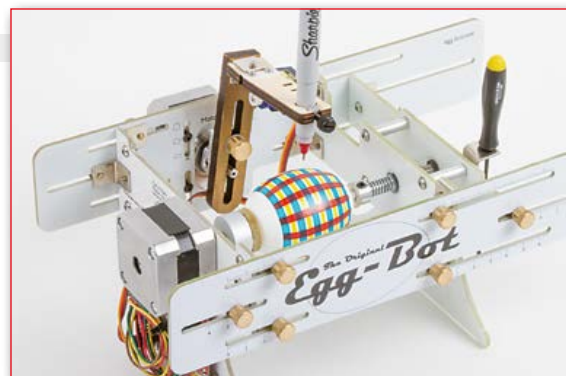
There are accessories you can get (for engraving etc.) but in its basic form the Egg-bot is a plotter that prints with pens (you get a few but you can use many). You can run it using Inkscape to draw vector graphics on many curved items. It is easy to use but do consider using multiple colors on an item requires switching pens for each color. That makes it somewhat slow and I ended up using the Egg-bot for single-color items most.

I had been using Inkscape before. It's open-source and simply the best software to turn any image found online into a great looking vector so I knew where to start but I assure you, it is nothing major to learn.

The Egg-bot kit itself is fairly easy to make. I am handy around the house but not an electronic engineer by trade. But I had enough skills to get the job done and the kit is well documented, so don't let that be a show stopper.

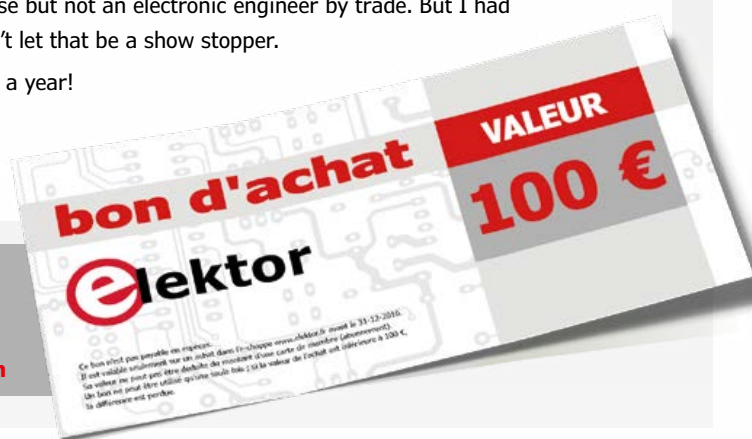
The biggest downside to the whole affair is that Easter only comes once a year! Still, I'll be one popular dad this year!

www.elektor.fr/eggbot



... vous aussi envoyez-nous votre avis sur votre produit favori d'Elektor ; si nous le retenons pour une publication, vous recevrez en échange un bon d'achat de 100 € à valoir dans l'e-shoppe d'Elektor...

J'essaye : www.elektor.fr/rotm



layar
INTERACTIVE PRINT

SCANNEZ CETTE PAGE ET REGARDEZ LA VIDÉO

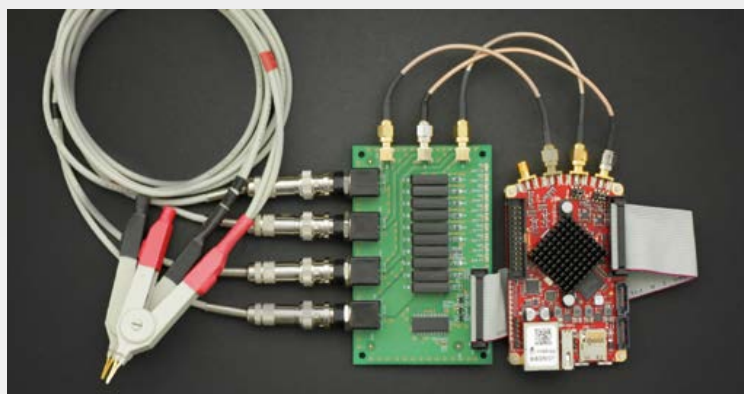
- 1 
Téléchargez l'appli gratuite Layar
- 2 
Scannez cette page
- 3 
Découvrez le contenu interactif





Cette carte d'extension vous permettra d'exploiter l'analyseur d'impédance de Red Pitaya (LCR-mètre) en vous passant de résistance de dérivation. La carte sélectionne la résistance parallèle adéquate dans un tableau de valeurs pour obtenir les mesures les plus précises possible.

Les données de la carte d'extension transitent sur une interface I²C. L'isolement galvanique est réalisé avec des relais soudés au centre de la carte.



Matériel fourni :

- Carte d'extension
- Câble en nappe – pour le raccordement à la carte Red Pitaya
- Câbles avec connecteurs SMA pour les lignes de signal raccordées à la carte Red Pitaya
- Sonde avec connecteur BNC pour raccorder le dispositif sous test

Attention : cette carte d'extension est livrée sans la carte Red Pitaya ! La carte Red Pitaya est vendue séparément.



Prix (membres) : 287,20 €

www.elektor.fr/impedance-analyzer-board

DVD-ROM Elektor 2000 à 2009

Retrouvez tous les articles parus dans Elektor de 2000 à 2009 !

Analyseur d'impédance

Carte d'extension
pour Red Pitaya

e-choppe Elektor en ligne

Votre panier d'achats
pour l'électronique

DVD Elektor 1990-1999



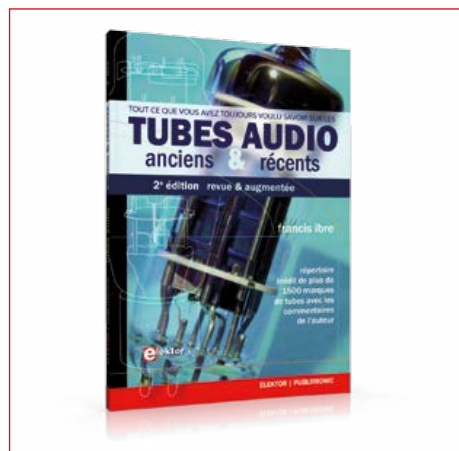
Ce DVD-ROM contient tous les numéros d'Elektor des années 1990 à 1999. Complétez votre schémathèque dans les domaines les plus divers (alimentation, audio, auto, domestique, hautes-fréquences, microcontrôleurs...) ! Les 2 200 articles d'Elektor réunis sont présentés par ordre de parution (mois/année), mais aussi classés par ordre alphabétique et par rubriques. Un index général permet d'effectuer des recherches dans tous les fichiers PDF.



Prix (membres) : 80,10 €

www.elektor.fr/dvd-elektor-1990-1999-french

Tubes Audio Anciens & Récents



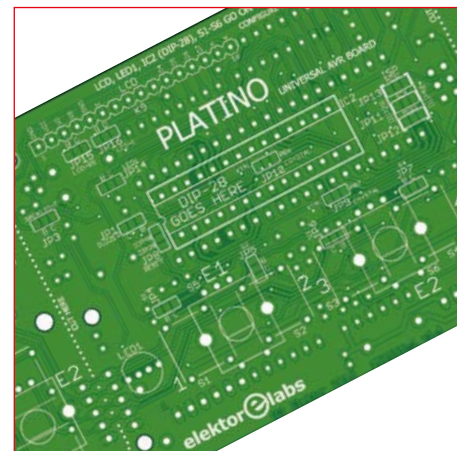
Découvrez ici ce qui se cache au cœur des « lampes », à travers une centaine de photos et d'illustrations, un tour d'horizon des productions actuelles et un historique des fabricants anciens les plus connus. Apprenez à reconnaître et choisir les tubes NOS les plus réputés, grâce aux codes et à leurs particularités de construction. Ce livre est aussi un formidable voyage dans l'histoire des grandes sociétés actuelles d'électronique et leur genèse.



Prix (membres) : 39,15 €

www.elektor.fr/tubes-audio-anciens-recents

Platino V1.4



Platino est une carte AVR universelle, compatible Arduino, créée par Elektor pour les contrôleurs ATmega à 28 et 40 broches. Son circuit imprimé a été remanié (version 1.4). Un régulateur de 3,3 V s'ajoute à celui de 5 V. L'accès au second UART des microcontrôleurs à 40 broches est plus facile. L'alimentation est raccordée au bornier à vis supplémentaire. Enfin la sérigraphie des composants a été peaufinée sur les deux faces du circuit imprimé.



Prix (membres) : 13,50 €

www.elektor.fr/platino-v1-4

PARTAGER

CRÉER

DÉCOUVRIR



Thijs Beckers (Elektor NL)

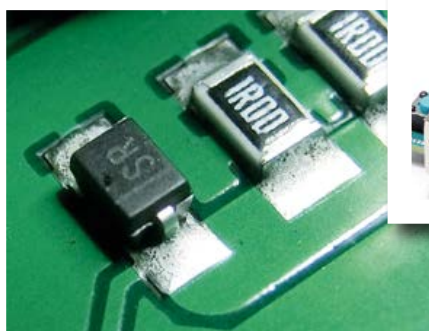
« Il aurait fait envie
à James Bond »



Du thermomètre à tubes Nixie au pré-ampli de guitare piloté par MIDI avec un Raspberry Pi 2, des projets que vous pouvez découvrir sur le site de notre labo, jusqu'à la présentation de la nouvelle carte Arduino/Genuino 101 : la section Partager contient toujours de quoi vous surprendre.

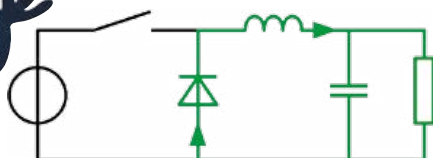
Bien entendu vous retrouverez la traditionnelle rubrique Rétronique. Ce mois-ci, cet article lève le voile sur le Nagra SN, un enregistreur miniature des années 60, en avance sur son temps, conçu pour les espions. Plus tard cet appareil a été proposé au grand public et même utilisé par les ingénieurs du son sur les tournages de films.

Du côté nouveauté, nous avons ausculté la petite dernière de la famille Arduino : la Genuino 101. Elle a été présentée à la Maker Faire de Rome en 2015 (16 au 18 octobre 2015). Notre labo vient de terminer son examen approfondi. Est-ce réellement la digne remplaçante de la très populaire Arduino Uno, comme on le pense chez Arduino ? Lisez notre banc d'essai !



Dans le numéro de novembre 2015, nous vous avons appelé à l'aide pour la conception d'une nouvelle alimentation à découpage : *convertisseur buck têtue au comportement erratique*. Vous avez été nombreux à répondre. Nous vous adressons à tous de chaleureux remerciements. Nous avons rassemblé et commenté les contributions les plus intéressantes.

Enfin je pense qu'il y a un article à ne pas manquer : l'art de souder les



CMS. Quel que soit votre âge, attrapez votre fer et commencez à souder sans peur des CMS. Harry a sélectionné pour vous des vidéos sur cette discipline. Lancez-vous ! ◀

(150677)

Arduino/Genuino 101

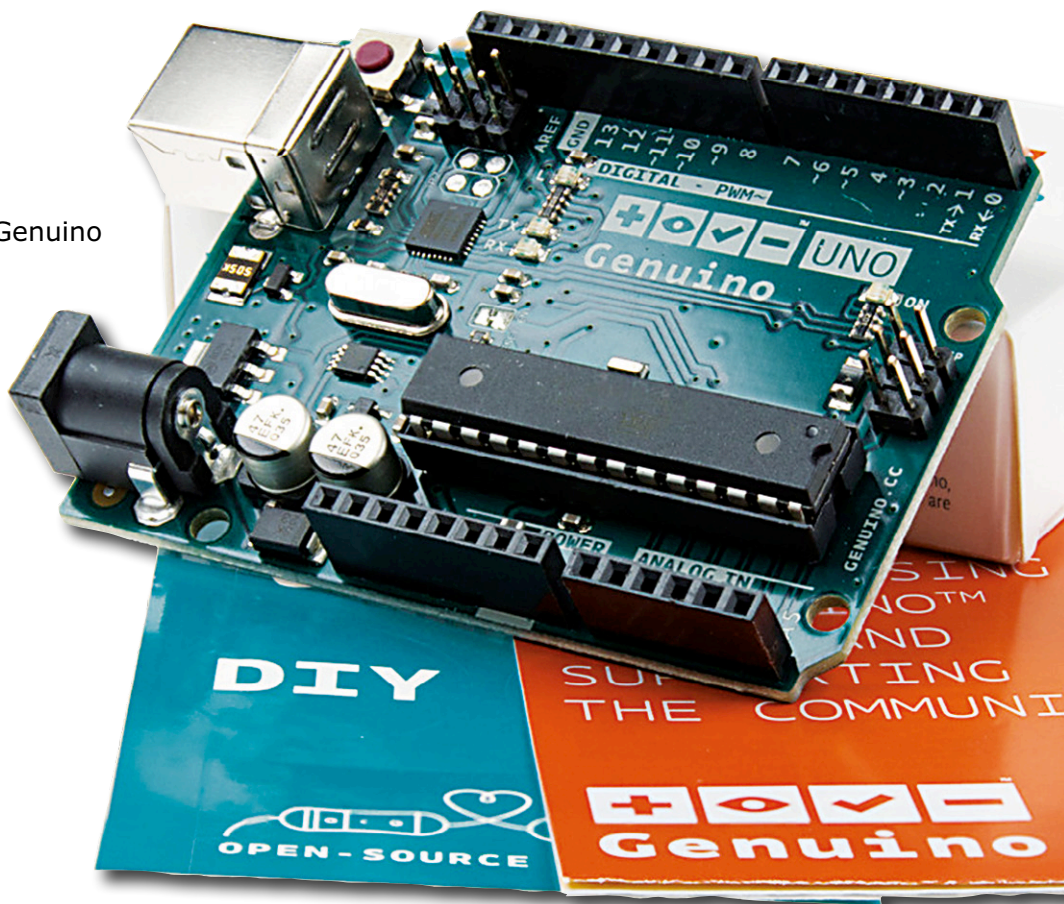
Fin de règne pour la carte Arduino UNO ?

Clemens Valens (labo d'Elektor)

Présentée à la Maker Faire de Rome de 2015, la nouvelle carte Arduino/Genuino 101 est maintenant disponible. Elle repose sur le système sur puce Curie d'Intel. Je l'ai testée pour vous.

La carte Genuino 101 est disponible dans notre boutique en ligne (www.elektor.fr)
www.elektor.com/genuino-101-powered-by-intel

Note : ce banc d'essai a été réalisé sous Windows 7 édition familiale, SP1.



La carte Arduino/Genuino 101 est *soutenue* par arduino.cc. Elle est commercialisée aux États-Unis sous le nom d'Arduino 101, alors qu'en Europe elle est appelée Genuino 101. De quoi en embrouiller plus d'un, surtout quand on sait qu'il existe aussi une Arduino Industrial 101 vendue à travers le monde par arduino.org. Comme je suppose que vous me lisez depuis l'Europe, je l'appellerai donc Genuino 101 par la suite, ou simplement 101.

Selon arduino.cc, la 101 a été conçue pour succéder à la Uno. C'est plutôt surprenant, car ces deux cartes sont à des années-lumière l'une de l'autre. Alors que la Uno repose sur un (impressionnant, je l'admets) microcontrôleur à 8 bits, la 101 embarque deux cœurs de 32 bits sous la forme d'un système sur puce (SoC) Curie d'Intel. Un de ces cœurs est un processeur x86, donc génétiquement parlant la 101 est sans doute plus près d'un PC que de la Uno. La 101 dispose par ailleurs d'au moins dix fois plus d'espace mémoire que la Uno.

Une autre différence, et pas des moindres, est que la Uno s'apparente plutôt à une carte de liaison pour l'Atmega328 (avec un convertisseur série-USB), alors que la 101 dispose de ses propres périphériques, par ex. un accéléromètre/gyroscope à six axes et un module radio Bluetooth Low Energy (BLE) (avez-vous remarqué l'antenne sur le circuit imprimé ?) Et tandis que vous avez les pleins pouvoirs sur le microcontrôleur de la Uno, l'accès au Curie est (pour l'instant) plus que restreint.

Bien que la 101 repose sur du matériel *open source*, peu de gens seront capables de construire la leur. Au moment de l'écriture de cet article, cliquer sur le lien « Eagle Files in .ZIP » [1] donnait accès à une archive avec un fichier BRD (compatible Cadence Allegro et non pas Eagle). Utilisez l'outil Allegro Free Physical Viewer 16.6 pour l'ouvrir. Les schémas sont disponibles en PDF.

Notez que la 101 n'est pas 100 % compatible avec la UNO. Elle n'a que quatre

sorties MLI (contre six pour la UNO) et elle est alimentée en 3,3 V. Des adaptateurs de niveau rendent les broches d'E/S compatibles 5 V, toutefois les entrées analogiques passent également par un adaptateur de niveau, ce qui signifie que leurs signaux sont limités à 3,3 V.

Côté EDI en revanche, vous ne noterez guère de différences. Vous devez utiliser la version 1.6.7 ou sup. d'arduino.cc [2] (**pas** la 1.7.x d'arduino.org) sinon vous ne pourrez pas télécharger et installer le compilateur et les outils nécessaires à l'exécution et à l'écriture des croquis. Pour les installer, ouvrez le Gestionnaire de carte (*Outils* -> *Type de carte*) et cherchez dans la liste déroulante l'entrée *Intel Curie Boards*. Cliquez dessus, puis sur le bouton *Installer* (**fig. 1**). L'EDI téléchargera environ 250 Mo, et après le processus d'installation vous serez prêt à coder. Le paquet est installé sous `<votre-chemin-utilisateur>\AppData\Local\Arduino15\packages\Intel\`.

Toujours dans l'EDI, sélectionnez la carte et le port série qui ont été créés par les pilotes de la carte. Ouvrez l'exemple Blink, cliquez sur *Téléverser*, et admirez le clignotement de la (mini) LED ! Il se passe a priori exactement la même chose qu'avec la Uno, mais si vous activez les avertissements du compilateur vous noterez quelques différences. Un croquis vide compilé occupe par ex. env. 30 Ko (15 % de l'espace programme disponible) alors qu'avec la Uno ce sont 450 Ko et 1 %. La programmation est elle aussi différente, avec un véritable indicateur d'état. Lorsque vous sélectionnez la 101 comme carte, un nouvel exemple pour Curie apparaît sous *Fichier -> Exemples*. La page arduino.cc parle d'un exemple Curie-BLEHeartRateMonitor [3], mais il n'est pas disponible sous mon EDI. Cela dit on peut tout de même copier le code de la page *tutorial* pour le tester. Le code s'exécute, mais il s'agit plus d'une sorte d'oscilloscope que d'un moniteur cardiaque. Pour que l'exemple fonctionne, vous devez installer la boîte d'outils nRF de Nordic sur votre appareil Android ou iOS. L'appli est utile, mais je m'attendais plutôt à une appli pour Intel ou Arduino. Autre exemple à tester : *RawImuDataSerial* (*Fichier -> Exemples -> CurieImu -> RawImuDataSerial*). Malheureusement, ce croquis crache des données à un débit trop rapide pour être suivi par l'œil humain. Cela dit l'EDI 1.6.7 a un nouvel outil appelé *Traceur série* (menu *Outils*), alors pourquoi ne pas en profiter ? Il suffit de commenter la ligne 124

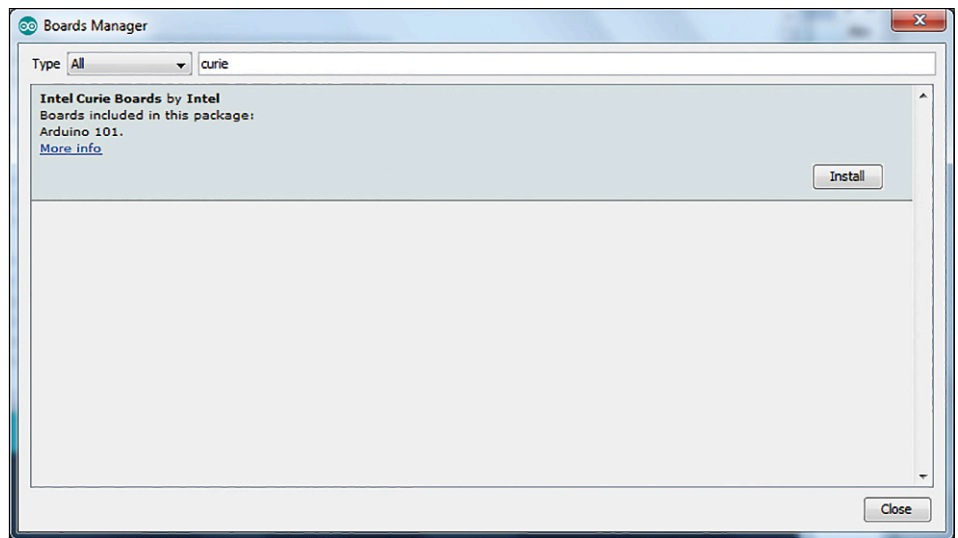


Figure 1. Utilisez le gestionnaire de cartes pour installer les pilotes de la carte Genuino 101. Cliquez sur *Intel Curie Boards* pour faire apparaître le bouton *Installer*.

(sur mon code d'exemple) de la fonction *loop*, comme ceci :

```
// Serial.print(<a/g:\t>);
```

Compilez le croquis. Pour une raison ou une autre, il faut attendre quelques secondes avant de pouvoir ouvrir le Traceur série. Une fois le traceur lancé, attendez à nouveau quelques secondes pour voir les données s'afficher. Tournez et agitez la carte pour influencer le tracé des courbes (**fig. 2**).

Une séquence de calibration est exécutée avant le démarrage du croquis. Pour cela la carte doit reposer à plat, immobile et

sur le dos. Le câble USB peut être gênant, si c'est le cas posez quelque chose d'un peu lourd sur la carte, une tasse de café (vide !) par ex., pendant à peu près 5 s. Retirez le poids avant de secouer la carte (mais devais-je vraiment le préciser ?) Le site arduino.cc fournit un autre exemple de mesure inertielle qui implique le langage Processing [4]. Le tutoriel suggère l'installation de la bibliothèque Madgwick avec le Gestionnaire de bibliothèques (*Croquis -> Inclure une bibliothèque -> Gérer les bibliothèques*) mais je ne l'ai pas trouvée dans ce menu. Je l'ai donc téléchargée depuis Github [5], puis j'ai utilisé l'option *Croquis -> Inclure une*

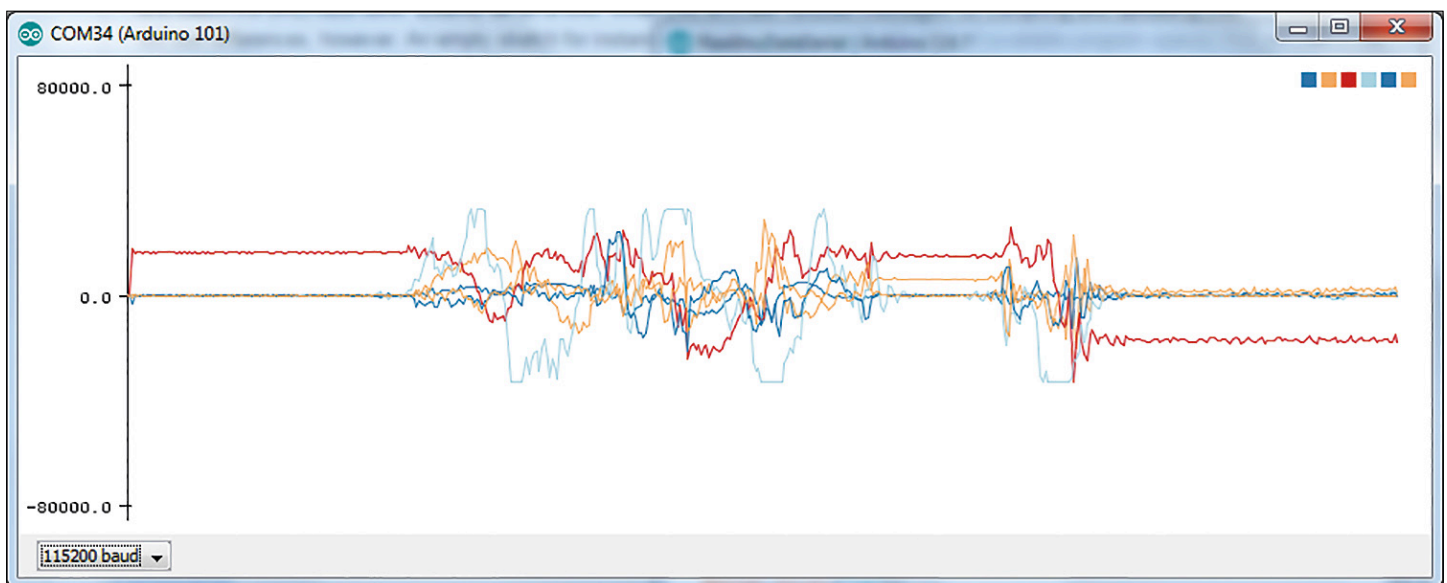


Figure 2. Le nouveau Traceur série montre les mouvements de la carte en temps réel.

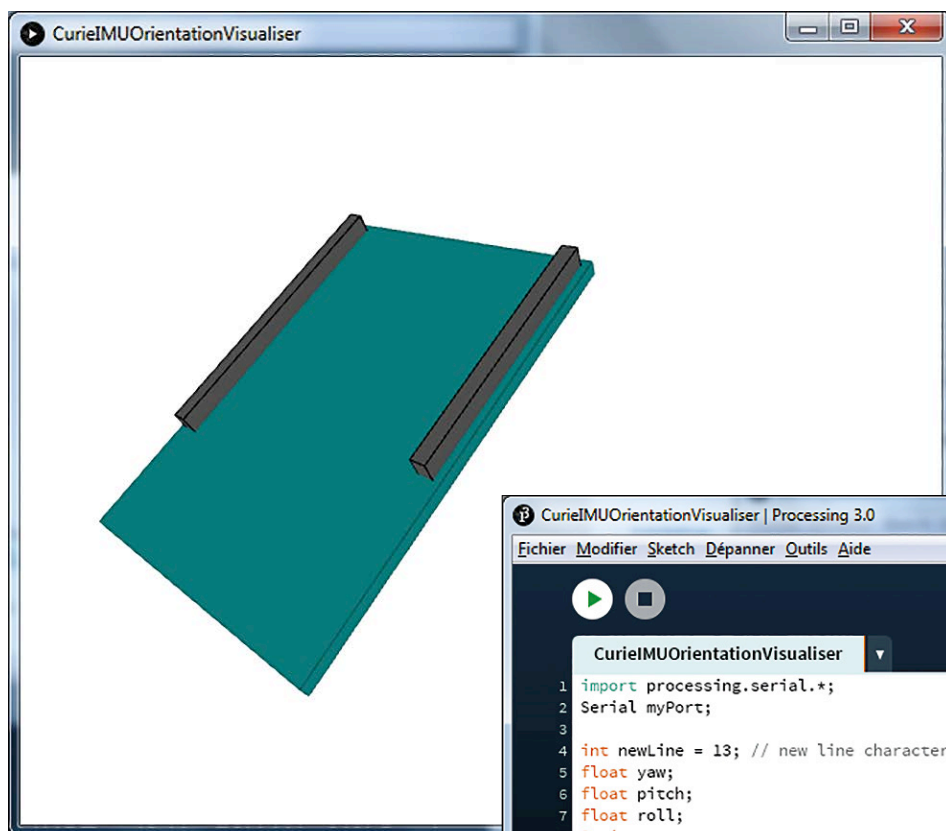
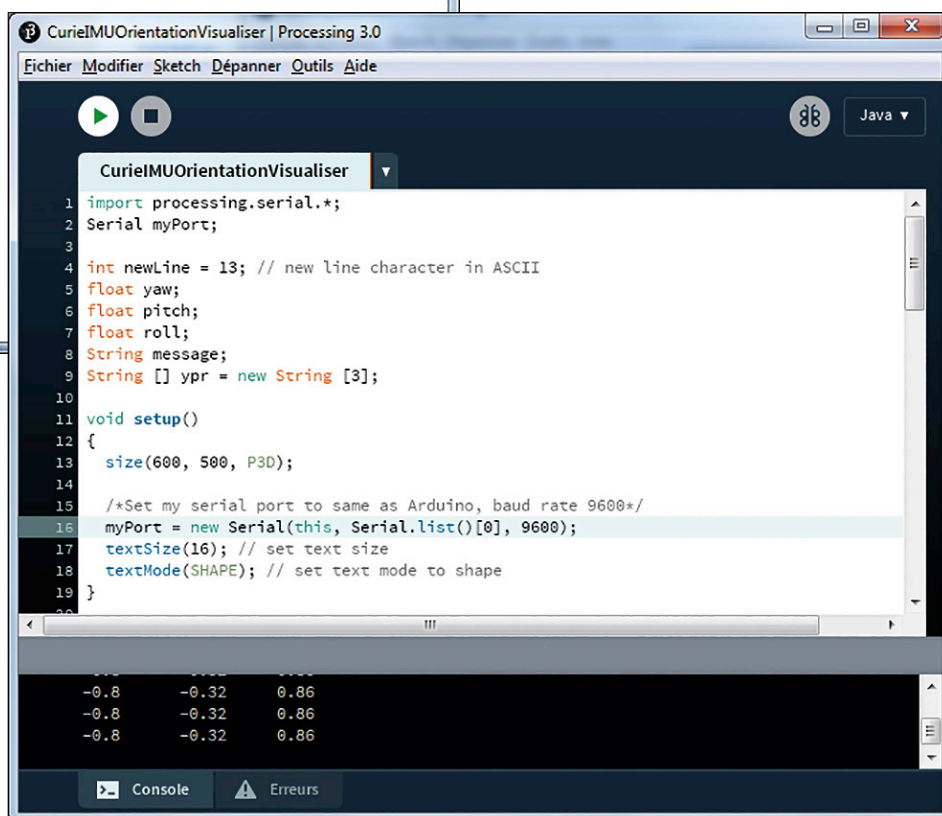


Figure 3. Vue en 3D de la Genuino 101 dans Processing. Régler le port COM n'est pas aussi simple que le commentaire le laisse croire.

bibliothèque -> Ajouter la bibliothèque .ZIP. J'ai copié le code Arduino dans l'EDI, le code Processing dans l'EDI de Processing, et là, tadaaaam ! ... nada. Juste une erreur disant `Method «glClearDepthf» not available`. J'ai cherché une solution sur le net, et découvert qu'il ne fallait pas utiliser la version actuelle de Processing (3.0.1, 23 octobre 2015) mais une ancienne.

L'installation de Processing 3.0 (30/09/2015) a résolu le problème du `glClearDepthf`, mais je me suis retrouvé avec une exception `Array index out of bounds` sur mon numéro de port COM (34). J'ai pu sortir de cette nouvelle impasse depuis Windows en forçant le numéro de port à 2. Après avoir débranché puis rebranché la 101, ni l'EDI Arduino ni l'EDI Processing n'ont protesté, mais ça ne marchait toujours pas. Je voyais bien des données sur le Moniteur série lorsque j'envoyais des « s » à la 101, mais Processing ne recevait que des zéros. Un débogage supplémentaire m'a fait réaliser que l'attribution du numéro de port COM dans le croquis Processing est un peu plus



compliqué que ce que le tutoriel veut bien nous faire croire, donc je vous conseille vivement de lancer d'abord l'exemple Processing affichant la liste des ports série disponibles [6]. Pour ce qui me concerne, il me fallait entrer zéro, l'indice du COM2 de la liste renvoyée pour mon ordinateur. Après ça, l'exemple de mesure inertielle a marché.

Lorsqu'on tourne la carte, l'image de la carte dans Processing tourne aussi. Cool. La ressemblance n'est pas parfaite, mais ce n'est pas vraiment le but (fig. 3).

Et voilà qui concluait ma première prise en main de la nouvelle Genuino 101. En attendant la sortie du système d'exploitation en temps réel Curie (son lancement est prévu pour mars 2016), on peut

dire que la Genuino est une bonne carte compatible Arduino, avec Bluetooth 4.0 et capteur inertielle à six axes. Elle n'est certes pas compatible à 100 % et est un peu chère, mais en contrepartie elle offre plus de fonctions. La programmation des croquis est aussi facile qu'avec la Uno, mais l'accès au matériel est restreint. Espérons que cette carte révélera bientôt son véritable potentiel.

(150728 – version française : Hervé Moreau)

Liens

- [1] www.arduino.cc/en/Main/ArduinoBoard101
- [2] www.arduino.cc/en/Main/Software
- [3] www.arduino.cc/en/Tutorial/Genuino101CurieBLEHeartRateMonitor
- [4] <https://processing.org>
- [5] <https://github.com/arduino-libraries/MadgwickAHRS>
- [6] https://processing.org/reference/libraries/serial/Serial_list.html

testeur de transistors Platino

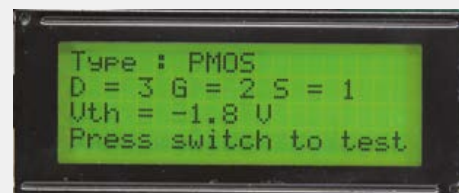
Elektor 03/2015, p. 62 (130544)

Une erreur s'est glissée dans le code source du testeur de transistors Platino.

La ligne 178 doit être :

`If Ischarwaiting = 1 Then...`

Le logiciel téléchargeable ici www.elektormagazine.fr/130544 a été corrigé.



projet 2.0

corrections, mises à jour et courrier des lecteurs



générateur de fonctions par DDS

Elektor 11/2015, p. 42 (150210)

Dans le schéma, C6 vaut 1000 μ F, 50 V. Cependant, comme le dit la liste des composants, C6, pourra être un électrochimique de 470 μ F, 50 V.

LED-Matrix-Player

Elektor 01-02/2016, p. 69 (150510)

Le schéma du LED-Matrix-Player indique pour IC5 un 74LVC1G74 – mais ce doit être un 74LVC1G97.



LCR-mètre 0,05%

Elektor 03/2013, p. 44 (110758)

Une mise à jour pour le projet du LCR-mètre a été postée sur le forum anglais : <http://forum.elektor.com/viewtopic.php?f=2698573&t=2716100>. Il existe une nouvelle version du micrologiciel 3.1.0, qui permet d'effectuer tous les réglages en mode autonome. Seule une mise à jour du micrologiciel requiert une connexion à un PC. De plus, les menus ont été étoffés. Pour accéder aux menus avancés, appuyez simultanément sur les touches Freq- et Freq+ avant la mise en route.

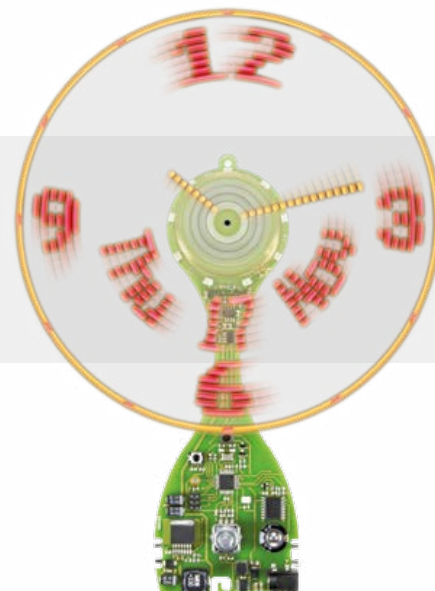
L'auteur attend vos commentaires sur le forum.

clac clock - Cadran Lumineux Interactif Circulaire

Elektor 12/2013, p. 44 (120732)

L'auteur de l'horloge *clac clock*, David Ardouin, a poursuivi le développement de son logiciel et a également conçu un nouveau boîtier.

La mise à jour du logiciel ainsi que les documents relatifs au boîtier peuvent être téléchargés à la page www.elektormagazine.fr/120732.





tubes alimentés sur batteries projet 2.0

Elektor 10/2015, p. 17 (150366)

Elektor 12/2015, p. 79 (150488)

Commentaires au courrier de M. Ruprechtsberger : il faut faire la distinction entre tubes alimentés par batteries et tubes basse tension. Les premiers étaient alimentés par piles uniquement, tant pour le courant d'anode que la tension de filament et étaient destinés à des appareils portatifs. Les seconds étaient, comme le dit l'auteur, conçus pour une utilisation dans des « supers » de voitures avec alimentation directe par le système de bord du véhicule. Comme leur puissance de chauffage est élevée, identique à celle de tubes classiques (6,3 V/0,3 A pour l'EF 98 par ex.), ils ne peuvent pas être alimentés en continu par batterie : trop volumineux et trop lourd.

Pour en savoir plus : VALVO Berichte Heft V Band 2, p. 35 sqq, Hamburg 1959.

Uwe Menrath



shield décodeur de morse

Elektor 01-02/2016, p. 52 (140086)

Il y a moyen de réaliser un décodeur de morse audible plus facilement que ne le décrit cet article. Il y a quelques années, j'ai construit, avec un ATmega16 et très peu de câblage externe, un décodeur aux caractéristiques suivantes : connexion directe d'un microphone à électret (grâce au préamplificateur de l'ATmega16), décodage du signal de morse de 1 000 Hz, LCD bon marché, analyseur de spectre de 300 à 1 800 Hz par pas de 100 Hz avec barographe sur l'écran LCD, faible consommation d'énergie. Pour en savoir plus (en allemand) : www.acs.uni-duesseldorf.de/~aurich/ham/Morsedec/index.html.

Volker Aurich



ainsi soient les LED !

Elektor -01-02/2016, p. 10 (150577)

Au sujet de l'encadré « Attention aux lampes à LED ! », j'aimerais souligner un autre critère : le parasitage par le *transverter*. J'ai chez moi cinq exemplaires de lampes à LED 12 V/3 W qui, en fonction, interdisent toute réception VHF dans un rayon de 4 m ; plus que du bruit. Au début, j'ai pensé à un mauvais déparasitage d'un transformateur électronique que j'ai remplacé par un transformateur toroïdal avec redresseur et 7812, mais le problème est le convertisseur élévateur de la lampe LED.

Avec d'autres lampes à LED, la réception OC devient impossible (je suis radioamateur, indicatif DM4ST). À plusieurs reprises déjà, l'Agence fédérale des réseaux (*Bundesnetzagentur*) a retiré de telles lampes de la circulation. Ainsi, la variante à condensateur de ballast est encore la meilleure, au point que, vu l'absence de tout parasite, notre brebis galeuse en devient presque vertueuse.

Thomas Stelzner



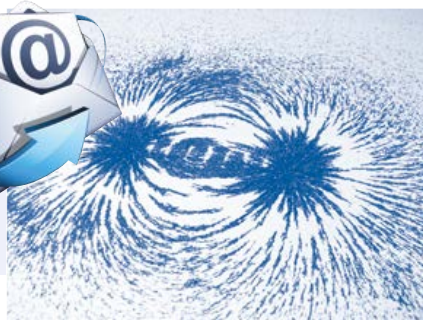
simulation de champ magnétique avec FEMM

Elektor 12/2015, p. 20 (130565)

Il y a quelques années, j'ai également écrit un script pour le calcul de bobines à air. Vous pouvez le télécharger ici :

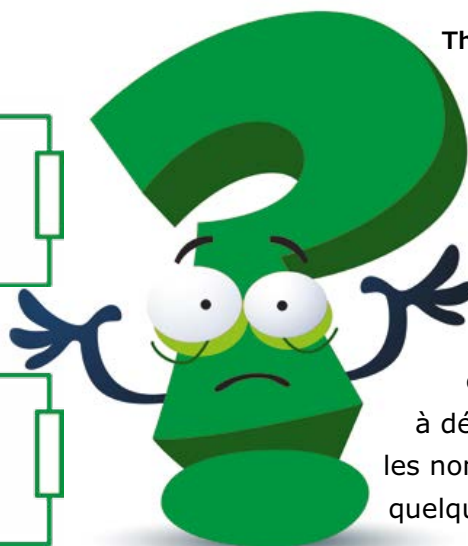
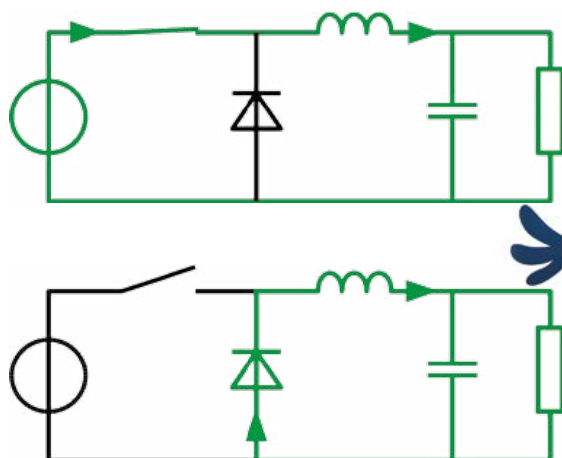
http://staff.itam.lu/feljc/electronics/femm/FEMM_Inductance_Interactive4.lua

Jean-Claude Feltes



convertisseur *buck* têtu dompté ... ou peut-être que non ?

Thijs Beckers (Elektor)



Dans le numéro de novembre 2015, nous vous avons appelé à l'aide pour la conception d'une alimentation à découpage. Nous avons examiné les nombreuses réponses reçues. Voici quelques commentaires intéressants.

Notre circuit utilise un LM2677T-ADJ de TI. Nous en reproduisons ici le schéma d'origine, **non pas** le schéma définitif disponible sur le site du labo d'Elektor [1]. Nous voulions une alim de labo avec une tension de sortie réglable entre 0 et 30 V et un courant d'au moins 3 A sur toute la plage de réglage. Comme l'indique la fiche technique et comme nos essais l'ont confirmé, ce circuit intégré peut délivrer 5 A. Jusqu'à ce que nous appliquions une charge impulsionnelle au circuit. À notre grande frustration, nous n'avions plus que 3 A de courant moyen. Nous avons alors lancé notre appel à l'aide. Les choses ont bien avancé depuis. Passons en revue des approches et suggestions intéressantes.

F. Vau propose une diode de roue libre costaude à la sortie. Ce n'est pas une mauvaise idée, mais cette diode n'a pas d'effet sur la stabilité. Elle sert uniquement de protection et est inactive en fonctionnement normal.

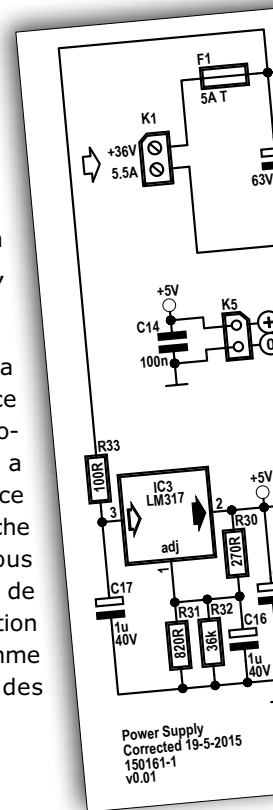
La proposition de *A. Broussal* est de monter un condensateur de quelques μF entre la broche 5 de IC2B et la masse, ce qui conduirait à la désactivation de la boucle de régulation. Sur notre circuit, C4 se charge déjà du filtrage HF. Nous avons également pris à cœur sa remarque quant aux faibles valeurs de C11 et C12 : des valeurs plus élevées rendent instable l'ensemble de la régulation. Une simulation faite avec le logiciel WEBENCH de TI le suggérait d'ailleurs.

Nous avons également pris en compte la remarque sensée de *M. Schreiber* : ajouter une rétro-réaction locale avec une résistance entre les broches 1 de IC2A et 12 de IC2D, et un condensateur entre le drain de T2 et la broche 12 de IC2D pour améliorer le comportement de la boucle de rétro-réaction. C8 peut alors être supprimé. Nous réfutons sa suggestion d'un « surdimensionnement » de T2, vu qu'il faut, à cet endroit, $R_{DS,ON}$ la plus faible possible pour minimiser la tension de sortie

en cas de limitation du courant. Avec un BS170, celle-ci est inférieure à $2\ \Omega$ pour $V_{GS} = 4\ \text{V}$.

W. Dijkman a suggéré de mettre un circuit d'amortissement en parallèle sur la sortie : deux grosses diodes (montage anti-parallèle) en série avec une résistance et un condensateur. Comme nous ignorons quel type de régulation utilise le LM2677T, il est difficile de préjuger des conséquences. Nous ne l'avons pas essayé expérimentalement, mais prédire un comportement correct du circuit tient de la divination. Par contre, cette remarque de *M. Dijkman* est pertinente : le rapport cyclique minimal devrait être faible pour les tensions de sortie basses. Le rapport cyclique à une tension de sortie de 2 V et un courant de 100 mA est alors si faible qu'une impulsion ne dure que 40 ns. Nous avons en effet constaté qu'à des tensions et des courants très faibles, le rapport cyclique est « instable » : la largeur de l'impulsion n'est pas constante. La tension de sortie, elle, est parfaite.

A. de Beun souligne notre silence sur la conception de la boucle de régulation, parce que les pôles et les zéros de la boucle de rétro-réaction doivent être placés correctement. Il a raison. Cependant, comme nous ignorons ce qui se passe dans le CI lui-même et que la fiche technique n'en dit rien, nous préférons nous taire. En cours d'essais, nous avons essayé de supprimer l'ensemble de la boucle de régulation et d'utiliser un simple diviseur de tension comme rétro-réaction, avec, malheureusement, des résultats similaires.



B. Bakker nous a dit qu'il avait, grâce à l'aide d'un ingénieur d'application de LT, compris un peu le fonctionnement interne du régulateur *buck* qu'il utilisait dans un circuit où il avait adapté avec succès la rétro-réaction à son schéma. Cela nous intéresse beaucoup !

A.C. Vogel pense que la fréquence limite de R3/C5 est trop basse ($20k/2n2 = 3,7 \text{ kHz}$). Notre avis : il ne sert à rien de rehausser cette fréquence parce que la fréquence de résonance de L1/C10,11,12 est de 3,4 kHz ; nous avons déterminé les valeurs de R10 et C5 empiriquement. Au passage, l'influence de C5 est très faible, vu la variation limitée du gain de la rétroaction, de +2 à +1. M. Vogel se demandait aussi si la capacité de 220 μF à l'entrée était suffisante. Outre le fait que nous supposons que l'alimentation CC est **stable**, ce condensateur sert juste au découplage HF pour l'alimentation CC ; l'augmenter ne changera rien. Comme sa résistance série (ESR) est relativement élevée, nous avons décidé de monter deux condensateurs en parallèle pour réduire l'ESR de moitié, mais cela n'a rien changé au comportement de l'alim.

P. Weiske lui aussi suspectait la boucle de rétro-réaction ; il pourrait peut-être y avoir, à la sortie de IC2A, une onde carrée dont le déphasage serait incompatible avec l'entrée de rétro-réaction de IC1. À notre avis, cela est sans importance, la fréquence limite pour R15, R16 et C6 est de 41 Hz. La tension sur IC2 tient plus d'une tension continue que d'une tension carrée. Nous comprenons sa remarque : dimensionner L1 pour que la différence entre la tension d'entrée et la tension de sortie soit maximale (même pour la plus petite tension de sortie possible). Toutefois la valeur de L1 n'est qu'un compromis, car il est impossible d'optimiser sa valeur pour toutes les valeurs de sortie. Les simulations WEBENCH suggèrent, pour différents courants et tensions de sortie, une valeur entre 5,6 et 33 μH !

H. Weiß a, lui, réalisé une alimentation à base de LM2678, circuit proche du LM2677 et cela sans problèmes. Il nous a fourni trois indices :

1. La constante de temps pour la broche de contre-réaction est trop grande ; avec C4 et R2, on a $T = 100 \text{ ns}$. Cela nous semble en fait assez rapide, 100 ns correspondant à 10 MHz...
2. Peut-être que la tension auxiliaire de 5 V est gênée par des impulsions et que ladite tension tombe trop bas ? Envisager de remplacer R33 par une diode, et augmenter C15 et C17 ? Cela non plus ne nous a pas permis d'identifier le coupable. R33 sert uniquement au découplage HF et la tension d'entrée pour IC3 peut descendre jusqu'à 8 V sans que le CI ne soit mis en difficulté. C15 et C17 ont été augmentés à 10 μF , sans différence notable.
3. Peut-être que la broche ON/OFF de IC1 est activée par des impulsions sur la tension d'entrée ? Cela aussi nous l'avons essayé, avec une capacité additionnelle exagérément grosse sur la tension d'entrée. Mais comme les résultats de mesure propres et stables sur cette ligne d'alimentation le laissent prévoir, c'est sans effet.

R. Ohlin remarque qu'une charge impulsionnelle de 50% à la fréquence de résonance du filtre de sortie constitue vraiment le pire scénario et que dans le cas d'un circuit similaire avec alimentation linéaire, il serait à prévoir des courants jusqu'à 7,5 A. Nous en sommes conscients ; c'est pourquoi nous n'avons, dans les spécifications de l'alimentation, mentionné « que » 3 A, ce que l'alimentation peut fournir en *toutes* circonstances. C'est un peu dommage vu qu'avec une charge « simple », elle fournit facilement 5 A (pour cela, faire passer à 10 k Ω R24 et R25 du schéma définitif, R17 et R18 du schéma reproduit ici), mais c'est contraire aux habitudes du labo d'Elektor. Nos spécifications sont honnêtes et correctes.

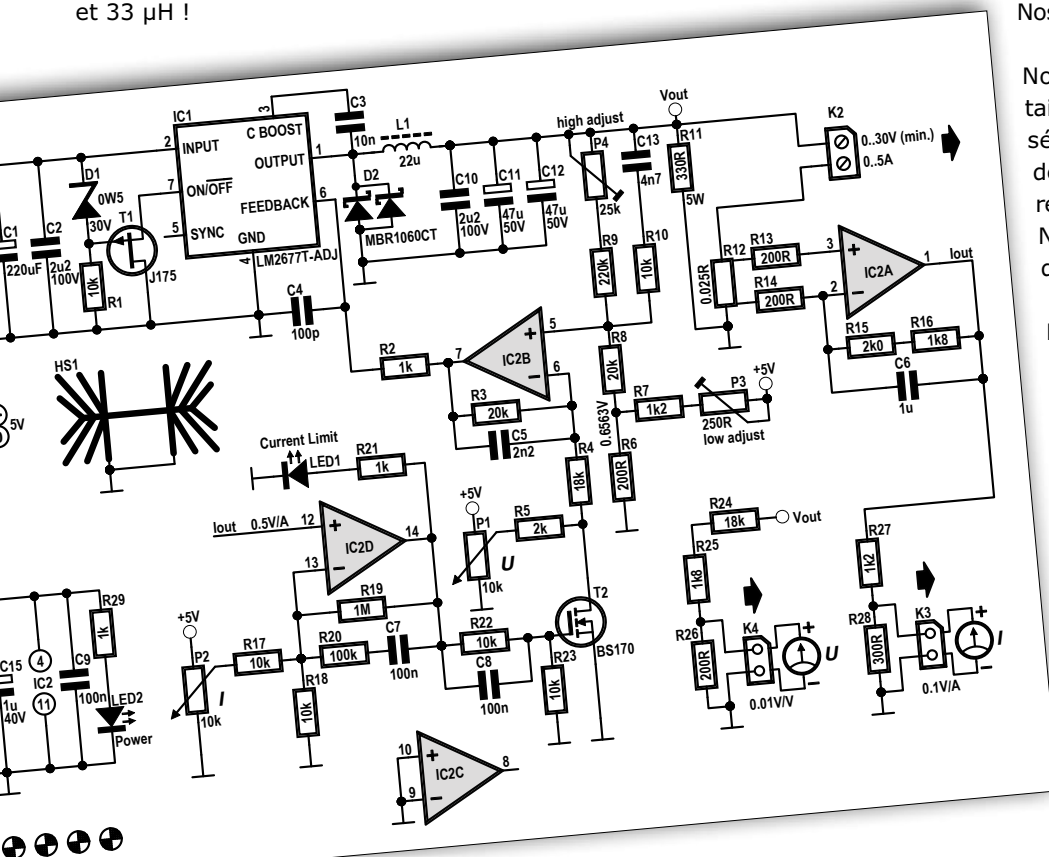
Nous avons également reçu des commentaires sur C3 qui méritent encore d'être passés au crible. Bien que 10 nF soit la valeur de l'application standard, une valeur différente pourrait donner de meilleurs résultats. Nous n'avons pas essayé, donc si l'envie d'expérimenter vous prend...

Notre conclusion ? Eh bien, nous pensons que ce circuit intégré n'est pas le composant idéal pour cette application et nous nous trouvons dans l'obligation d'indiquer 3 A comme intensité maximale du courant de sortie. Néanmoins, c'est toujours une bonne alimentation qui n'a pas de raison d'avoir honte de ses caractéristiques. Et bientôt, vous retrouverez cette alimentation dans le magazine !

(150680 - version française : Guy Raedersdorf)

Lien

[1] www.elektor-labs.com/node/4720



le soudage d'une main sûre

essayez-vous aux CMS

Harry Baggen (labo d'Elektor)

Bien souder est tout un art, particulièrement avec les plus petits CMS. Elektor a déjà décrit à maintes reprises comment tirer son épingle du jeu sans grande expérience ni outillage spécialisé. Le tout est d'oser faire le premier pas. Quelques séries de photos et de vidéos vous donneront plus d'assurance.

Les composants électroniques actuels n'ont généralement plus de fils de raccordement. L'avantage des composants montés en surface (CMS ou SMD), c'est d'être plus petits que les autres et d'occuper moins d'espace sur le circuit imprimé. Mais avec eux, il est plus ardu de réaliser un prototype à la main, comme de réparer des circuits équipés de CMS. C'est avant tout une question de pratique et de soin au travail. Les circuits construits au labo d'Elektor aussi contiennent la plupart du temps des CMS et l'on reste souvent ébahis des résultats que nos techniciens obtiennent d'une main sûre avec de fines pinces à souder, sans faire usage du four à refusion, dont nous disposons aussi, bien entendu.

Ces dernières années, nous avons souvent reçu des appels à l'aide de lecteurs qui avaient du mal à installer des CMS ou même qui n'osaient pas se lancer. Or, le mieux est d'essayer soi-même. Commencez avec un vieux circuit imprimé et quelques résistances, et soudez-les de votre mieux. Vous remarquerez vite que vous y arrivez. Pour les puces à nombreuses pattes, il y a différentes méthodes et astuces pour obtenir un bon résultat. Elektor a publié de nombreux articles sur le sujet, mais le mieux est de pouvoir suivre une démonstration sur la manière de s'y prendre, dans une vidéo éloquent, par exemple. Rai-

son pour laquelle nous avons cherché sur l'internet des séries de photos et des vidéos qui montrent bien comment traiter différentes sortes de CMS. C'est une matière qui passionne beaucoup de monde, on en trouve des centaines, si ce n'est des milliers, de vidéos. Après avoir visionné une cinquantaine de ces vidéos, nous en avons sélectionné quelques-unes qui peuvent vraiment vous aider à démarrer.

Les photos

Pour se familiariser avec le sujet, on commence par quelques photos. La plupart des lecteurs ont probablement déjà de l'expérience du soudage des composants à fils, sinon, les débutants trouveront quantité de renseignements sur la toile.

Le site de Curious Inventor [1] donne une bonne information sur le travail des CMS. Une page décrit l'outillage nécessaire et des photos montrent comment souder les diverses sortes de boîtiers CMS, une résistance et des puces PLCC, QFP et QFN. Une page traite aussi de l'utilisation du four à CMS, mais on n'y est pas encore.

Autre guide photographique pratique sur les CMS, c'est chez Infidigm [2]. Des séries de photos exposent comment souder, mais aussi dessouder différents types de composants selon plusieurs méthodes éprouvées.



Les vidéos

Maintenant que vous savez à quoi servent les divers outils, tels que le stylo à flux et la seringue de pâte à souder, il est utile de regarder les vidéos qui les font intervenir. Ce sont des articles que vous devrez certainement vous procurer pour travailler sur les CMS. En outre, on peut vous conseiller d'acheter un pistolet à air chaud d'une centaine d'euros si vous devez régulièrement opérer sur des CMS.

Les trois premières vidéos [3], [4] et [5] sont toutes de John Gammell, un instructeur professionnel qui a des années d'expérience dans l'enseignement du soudage manuel. Il vous y explique comment faire en pratique. Tout amateur ne doit pas forcément agir comme un spécialiste en son genre, mais il est toujours intéressant de connaître la bonne manière de procéder, quitte à trouver des variantes personnelles. Dans les deux premières vidéos, *Techniques de soudage à la main – montage en surface*, il montre les différentes façons de souder des CMS à nombreuses pattes. Tout cela a l'air si simple qu'on en oublierait qu'il faut beaucoup d'expérience pour réaliser d'aussi belles soudures sur CMS. La partie 3 concerne le soudage du boîtier DPAK. John a posté beaucoup d'autres vidéos instructives sur YouTube, pour les trouver, il suffit d'y chercher son nom.

Le contenu de ces vidéos représente à peu près le fin du fin de ce qu'on peut espérer obtenir comme résultat. Aussi, les deux dernières vidéos, plus longues, mettent-elles en scène des électroniciens qui ont leurs propres méthodes de travail sur

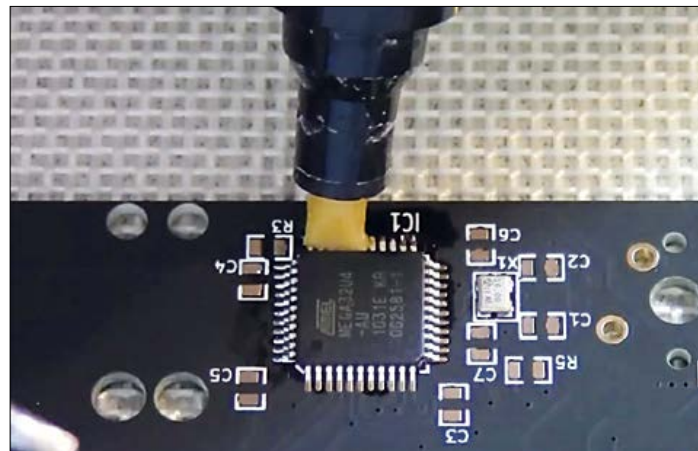
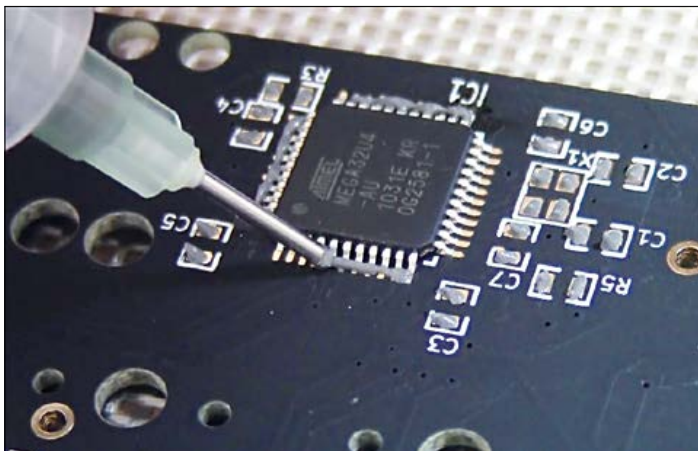
les CMS. Dans la [6], *Comment souder des CMS*, Ryan Edwards de Sparky's Widgets montre comment il arrive à dessouder des CMS d'un circuit imprimé pour les souder sur un autre à l'aide d'un fer à souder et d'un pistolet à air chaud. Particulièrement instructif ! Les vidéos [7] *Soudage de CMS à l'air chaud* et [8] *Soudage de CMS au fer à souder* permettent de suivre en détail la manière d'installer des CMS de différentes sortes, avec toute la procédure de l'application de la pâte à souder, le positionnement des puces et même l'installation d'un connecteur USB. Au milieu de la première de ces vidéos, il y a des parasites dans l'image, mais ils n'enlèvent rien à la qualité du contenu.

Ces vidéos ont de quoi vous inciter à essayer les techniques illustrées. C'est en tout cas ce que nous vous souhaitons ! ◀

(150682 – version française : Robert Grignard)

Liens

- [1] http://store.curiousinventor.com/guides/Surface_Mount_Soldering
- [2] www.infidigm.net/articles/solder/
- [3] www.youtube.com/watch?v=5uiroWBkdFY
- [4] www.youtube.com/watch?v=hINp_g68mh4
- [5] www.youtube.com/watch?v=L_DIpkIxXcI
- [6] www.youtube.com/watch?v=z7Tu8NXu5UA
- [7] www.youtube.com/watch?v=2Z7nCAxS2Rg
- [8] www.youtube.com/watch?v=OaOaRaGGdMc

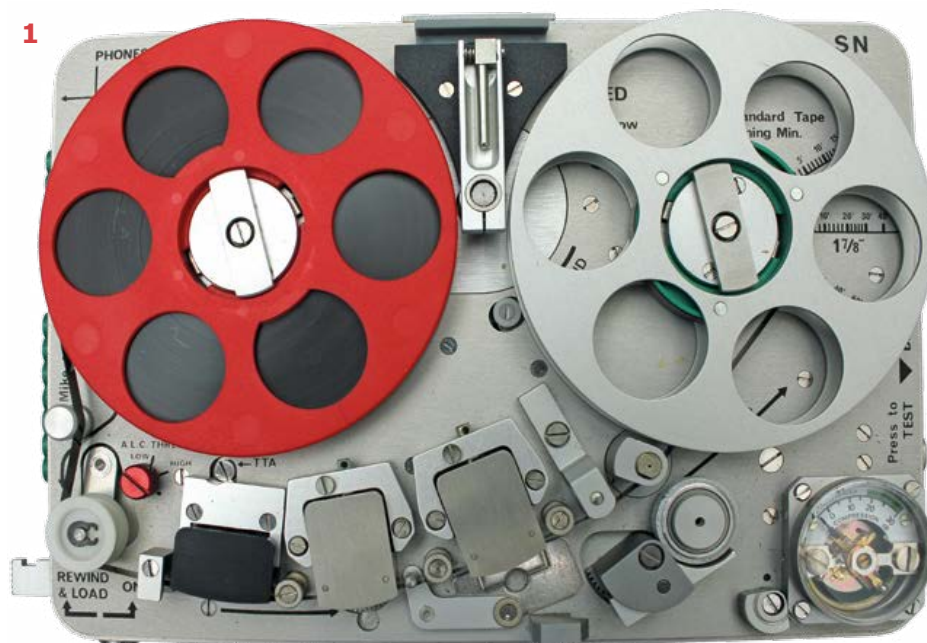




Nagra SN : l'oreille discrète de l'espion

un enregistreur très facile à dissimuler

Peter Beil (Allemagne)



James Bond l'aurait jaloué, ce petit jouet anodin à première vue, il recèle un enregistreur de son que Q lui aurait présenté avec fierté. Capable même de synchroniser le son avec l'image !

Ce sont les Services secrets américains qui, à la fin des années 1950, ont contacté le constructeur suisse haut de gamme Kudelski pour la fabrication d'un enregistreur miniature. À cette époque, les enregistreurs « furtifs » utilisaient le fil d'argent comme support pour sa compacité. Mais la diaphonie et l'étroitesse du spectre sonore rimaient avec médiocrité de la reproduction.



À partir de 1960 et pendant une bonne dizaine d'années, à l'insu de l'homme de la rue, le Nagra SN (pour **s**érie **n**oire) a été commercialisé auprès de clients sélectionnés, avant d'atteindre le grand public.

Le format habituel des bandes magnétiques d'alors ne convenait absolument pas au but à atteindre (**figure 1**). On a donc fait des bandes de 3,81 mm de large et des bobines adaptées, avec un code de couleur selon l'épaisseur de la bande, pour des enregistrements standard, longue durée et double durée. Cette dernière permettait une heure et demie d'enregistrement, mais en usage professionnel, c'est la bande standard (**figure 2**) qu'on utilisait. Les bobines ont un diamètre de 68 mm avec un verrouillage, si bien que l'appareil peut même fonctionner à l'envers, posé sur le couvercle (**figure 3**). Alimenté par deux piles AA (R6), il offre cinq heures d'utilisation. Aujourd'hui encore, les dimensions sont saisissantes : 147 mm sur 100,5 mm, épaisseur 28 mm, pour un poids de 574 g.

Caractéristiques et fonctions

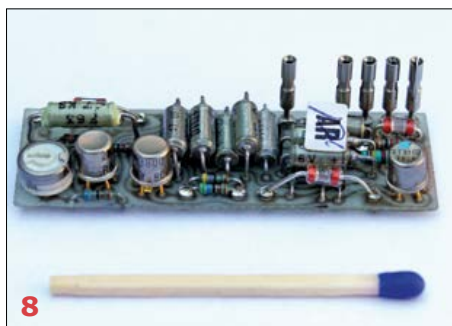
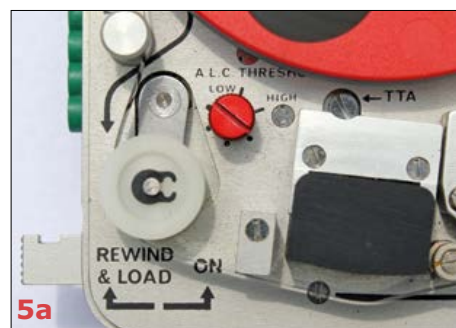
La vitesse de défilement de 9,5 cm/s ne semble pas exceptionnelle, mais les caractéristiques techniques le sont : réponse en fréquence de 60 Hz à 15 kHz, rapport signal/bruit de 62 dB, pleurage et scintillement de $\pm 0,1$ % selon norme DIN et une plage de tempéra-

ture de fonctionnement de -40 °C (!) à $+70$ °C. Pour des applications particulières, on peut descendre à la vitesse de 4,75 cm/s (**figure 4**) pour une réponse en fréquence de 80 Hz à 8 kHz et presque quatre heures d'enregistrement, qui s'effectue en monophonie pleine piste ; une version stéréo est apparue plus tard.

Pour permettre l'utilisation nomade sans commande automatique du gain, l'appareil possède un compresseur qui l'assure sous une forme étonnamment propre. Cette technique inhabituelle remonte sans doute aux origines de l'appareil destiné aux Services secrets, qui tenaient à capter même les sons les plus faibles. En usage professionnel, elle peut devenir embarrassante, aussi a-t-on rendu le seuil ajustable (**figure 5a**) et sur le côté droit, on a ajouté un galvanomètre pour l'indiquer ainsi que pour évaluer l'état des piles (**figure 5b**). Il y a aussi pour le Nagra SN une série de microphones adaptés à différents usages (**figure 6**).

La précision suisse pour se faire petit

À l'intérieur, on admire une précision d'horlogerie. Les circuits imprimés sont tous enfichables, vissés l'un à l'autre pour la fiabilité (**figure 7**). Toute l'électronique est faite de composants discrets (**figure 8**) et pour cause, s'il y avait déjà des puces en ce temps-là, elles ne couvriraient pas le domaine audio.





Faute de place, les concepteurs ont dû renoncer à certaines fonctions et finasser pour en incorporer d'autres. Rembobiner la bande se fait rapidement au moyen d'une manivelle rétractable (**figure 9**). Pas d'inverseur enregistrement/lecture non plus : micro branché, on enregistre, débranché, on reproduit. Un levier met en position le mécanisme (**figure 10**) et une tête magnétique distincte permet une vérification au casque de ce qu'on enregistre.

Un accessoire extérieur (**figure 11**) équipé d'un indicateur de niveau et d'un bouton actionné du pouce permet de régler précisément l'amplitude pour faire du Nagra SN un enregistreur à bande « normal ».

En poche

Depuis 1971, on a pu se procurer une version modifiée de l'appareil qui a fort intéressé les ingénieurs du son occupés au tournage de films. Leur hantise venait des microphones sans fil de l'époque dont la fiabilité laissait à désirer, ils travaillaient dans la maudite bande VHF entre 60 et 80 MHz. Impossible d'enregistrer dans un avion, à proximité d'émetteurs de radio, dans les salles où la propagation était mauvaise, ni là où se produisaient des interférences RF. Alors, ils glissaient simplement dans la poche de l'acteur l'enregistreur dont l'autonomie couvrait largement la durée de la séquence. Et



Qu'est-ce qu'une fréquence pilote ?

C'est un signal dont la fréquence est très précisément établie à 50 ou 60 Hz pour synchroniser le son avec l'image dans un film. Comme une bande magnétique est inévitablement sujette à de petites fluctuations, une tonalité de référence, issue de la caméra ou d'une autre source pilotée par quartz, est enregistrée simultanément à la prise de son. Lors de la lecture, la vitesse est asservie à la tonalité enregistrée. Alors, même les plus petites dérives dans la vitesse de lecture sont corrigées. De nos jours, c'est un code temporel qui en fait office.

encore lors des prémices de l'enregistrement vidéo, quand les caméscopes ne délivraient pas un son convenable, le Nagra servait souvent d'enregistreur audio pour fournir une piste son de bonne facture.

Adjonctions

Dans sa configuration primitive, l'appareil n'était pas destiné à synchroniser le son avec l'image. Il lui manquait une référence de fréquence comme le montage de la **figure 12**. Plutôt que d'utiliser l'habituelle fréquence pilote à 50 Hz enregistrée par une tête supplémentaire, cet auxiliaire ajoute une composante à 10 Hz noyée dans le signal audio.



Lors de la lecture en studio, le Nagra se place sur un adaptateur approprié (**figure 13**) qui multiplie par 5 (ou par 6 en zone à 60 Hz) le signal pilote et l'asservit à la référence. Un filtre passe-haut d'une fréquence de coupure de 30 Hz l'empêche d'interférer avec le signal audio enregistré. Les photos des **figures 14** et **15** montrent toutes les commandes et les connecteurs sur l'adaptateur.

Conclusion

Même si mon Nagra SN ne sert plus depuis plus de vingt ans, il reste, dans sa simplicité, un bel appareil dont je ne pourrais pas imaginer me défaire ! ◀

(150674 - version française : Robert Grignard)

ESTD 2004

www.elektor.tv



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



bruits de labo

ressusciter de vieux gadgets avec l'électronique la plus récente

Voici une série de projets qui combinent anciennes et nouvelles technologies de façon intéressante. Ces projets sont des sources d'inspiration. Pourquoi ne pas ressusciter de vieux gadgets avec l'électronique d'aujourd'hui ?

Préampli à tubes pour guitare commandé par Raspberry Pi

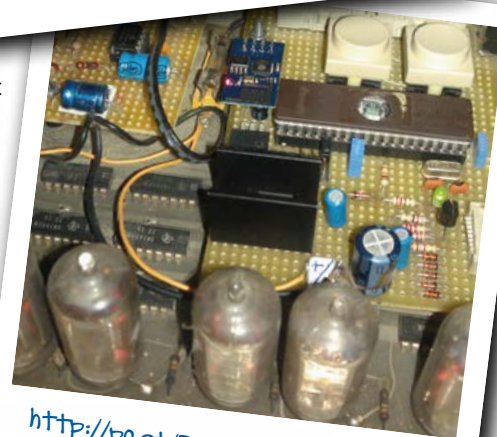
L'auteur cherchait une commande intelligente pour son ampli de guitare à tubes, pour changer rapidement les réglages tout en jouant de l'instrument, par ex. pour passer du son normal à la distorsion, ou mettre en mémoire des réglages et en rappeler d'autres. Inséré dans une chaîne MIDI avec d'autres instruments, son préampli devait répondre aux commandes issues de logiciels MIDI (*looper* ou séquenceur) ou de pédales et d'autres appareils MIDI. Écrit en C pour garantir la meilleure portabilité et un temps de latence court, son logiciel tourne sur un Raspberry Pi 2 qui offre des entrées-sorties et des ports de communication en pagaille.



<http://po.st/RPiGuitarPreController>

Émulation de signal horaire DCF77 avec ESP8266 et Arduino

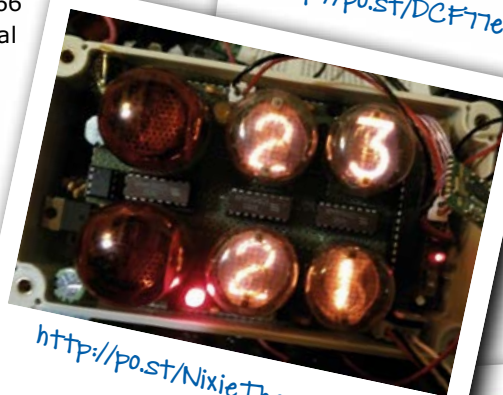
Il y a 20 ans, l'auteur a modernisé une vénérable horloge DCF à tubes Nixie construite avec un 87C51 et un récepteur de signaux horaires DCF77. Ensuite, la réception de DCF77 s'est progressivement détériorée (peut-être du fait de la pollution électromagnétique causée par les alimentations à découpage ?) au point que l'auteur a décidé de remplacer le signal horaire HF par un client NTP : maintenant, un simple module ESP8266 connecté à son réseau sans-fil domestique émule le signal radio horaire.



<http://po.st/DCF77emulator>

Double thermomètre à Nixie avec horloge GPS et alarme de boîte à lettres

Au départ, l'auteur voulait offrir à son père (qui aime les vieux trucs) un thermomètre intérieur-extérieur avec quatre Nixies. Pour le module extérieur, il fallait en plus une alimentation solaire et une communication RF. Et comme la boîte à lettres de ses parents est à bien 60 m de leur maison, il s'est dit qu'un témoin de présence de courrier serait bienvenu. Et puisque l'affichage est à 2 x 2 chiffres, autant en profiter pour donner l'heure. Pour qu'elle soit toujours juste, pourquoi ne pas prévoir un récepteur GPS...

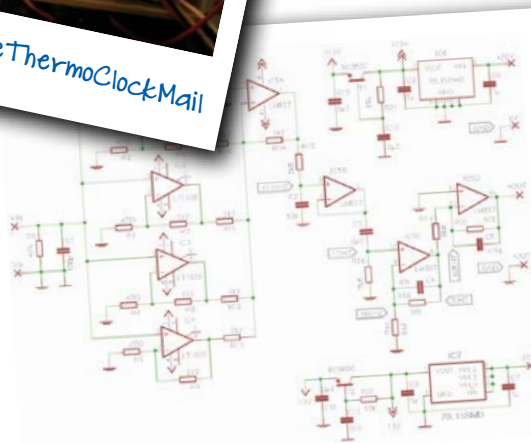


<http://po.st/NixieThermoClockMail>

Supra 2.0 : préampli phono moderne

Il y a plus de 25 ans, Elektor proposait un préampli phono MC/MM à très faible bruit. La performance consistait à mettre en parallèle huit transistors à faible bruit bon marché (BC550/BC560) ce qui donnait un grand circuit avec vingt transistors (par canal !). Aujourd'hui, l'existence d'amplificateurs opérationnels rapides et optimisés comme le LT1028 rend caduque cette approche. Il est à peu près impossible de repousser davantage les limites, puisqu'on bute sur le bruit intrinsèque des résistances. Quel est le prix d'une telle excellence ? ◀

(150671)



<http://po.st/supra2>

compilées par **Aniek Reuling**

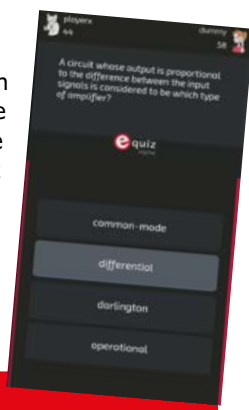
Bons plans de Noël 2015

La campagne *Bons plans de Noël 2015* d'Elektor a une fois encore été un franc succès. Sapin de Noël programmable pour vous amuser ou livre pour apprendre, vous avez profité de toutes nos promotions. Les articles les plus populaires auront été les DVD couvrant une décennie d'Elektor, soit plus de 2000 articles représentant 7000 pages d'électronique sur 1 mm d'épaisseur !



Elektor lance l'appli E-Quiz

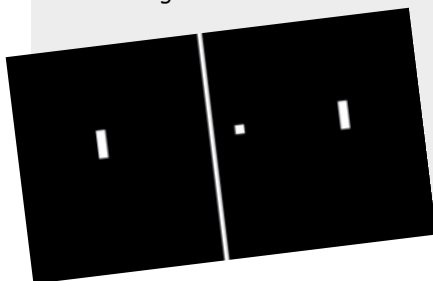
Nous restons fidèles à notre slogan *Découvrir, Créer, Partager*. Et comme s'amuser avec l'électronique fait partie de notre ADN, nous avons créé un quiz d'électronique pour Android. Le lancement de la version alpha est pour bientôt. Pour être tenu au courant, écrivez à fabio.romagnoli@eimworld.com.



READ ONLY MEMORY

Le magazine Elektor et son éditeur sont fiers de leur histoire. Cette rubrique montre pourquoi.

Pong, un des premiers jeux vidéo, était populaire dans les années 70. Saisissant la balle au bond, Elektor publia en 1975 le circuit d'un **jeu de tennis pour TV**. L'utilisateur pouvait ajouter un adversaire automatique et ainsi se désocialiser tout en gardant les doigts occupés. L'écran affichait la ligne de partage du terrain qu'ont la plupart des jeux de balle. L'article expliquait comment paramétrer les deux lignes horizontales contre lesquelles la balle pouvait rebondir, ainsi que celles définissant la sortie de la balle.

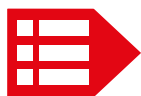
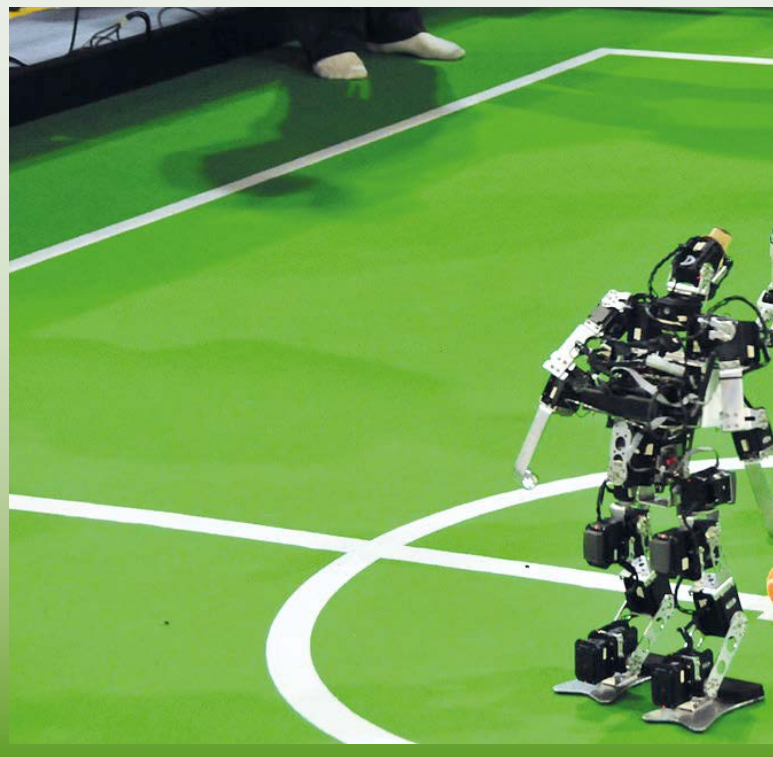


Ce que nous apprennent nos frères artificiels

Elektorethics par Tessel Renzenbrink

C'est par cette comparaison que Joscha Bach [1] a commencé son fascinant exposé à la *Chaos Communication Conference* [2], fin 2015 à Hambourg. Bach a étudié la philosophie puis l'informatique à l'université Humboldt de Berlin. Aujourd'hui cogniticien dans le cadre d'un programme Harvard consacré à la dynamique de l'évolution, il conjugue ses deux pôles d'intérêt en étudiant l'humain sous l'angle de l'intelligence artificielle.

Bach a codé la simulation d'un monde lorsqu'il travaillait sur des robots footballeurs. « Au départ nous avions de vrais robots qui utilisaient les données de leurs capteurs pour mettre à jour le modèle que nous leur avions implanté. » Mais les robots sont coûteux et encombrants. « Nous avons donc simulé sur ordinateur le terrain et la physique de façon à ce que les données produites soient à peu près les mêmes que celles de la réalité, puis placé l'esprit du robot dans un corps virtuel. Et ça fonctionne tout aussi bien ; je veux dire du point de vue du robot, puisqu'il ignore tout de la réalité et n'a donc aucun moyen de distinguer le réel du virtuel. La seule chose qu'il perçoit, c'est la structure des données de son interface générale. »



INDISCRÉTIONS • En début d'année, plusieurs membres d'Elektor ont rejoint la nouvelle antenne délégation d'Elektor s'est rendue à la conférence Embedded World tenue à l'Exhibition Centre fabricants ont rejoint l'équipe d'Elektor

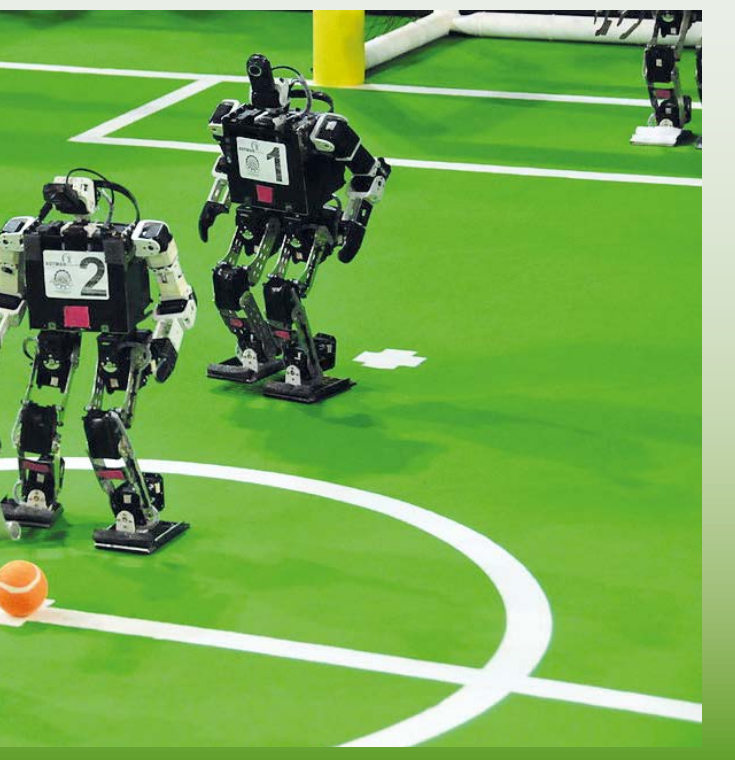
Nous sommes l'expression d'un code. Et d'un code pas si long que ça. Le génome – l'ensemble de notre ADN – tient sur un CD-ROM. Nous sommes moins complexes que le système d'exploitation Windows 10.

C'est la même chose pour nous. Nos esprits créent un modèle du monde d'après les données de nos cinq sens. Selon Bach : « Lorsque vous découvrez que vous êtes dans la même situation que ces robots, vous réalisez que vous êtes une sorte de robot biologique qui n'a pas d'accès direct au monde. Qui n'a jamais vu la matière et les gens. Qui a simplement perçu des fragments d'information auxquels le cerveau doit donner un sens. » La conférence *Computational Meta-Psychology: an Artificial Intelligence Exploration into the Creation of Meaning* peut être visionnée sur le site CCC [3]. Je vous la recommande : Bach explique le fonctionnement du cerveau des génies, que la religion est un virus de l'esprit, et pourquoi l'inadaptation sociale des « nerds » est pour la première fois un avantage du point de vue de la sélection naturelle.

[1] <https://twitter.com/Plinz>

[2] https://ccc.devsn.se/congress/2015/wiki/Main_Page

[3] https://media.ccc.de/v/32c3-7483-computational_meta-psychology



allemande de la jolie ville d'Aix-la-Chapelle • Une de Nuremberg • Plusieurs nouveaux auteurs et

PROFIL D'EXPERTE (en herbe)

Elektor est au cœur d'un réseau de plus de 1 000 experts et d'auteurs engagés dans la publication de livres, d'articles, de DVD, de webinaires et autres événements. Coup de projecteur !

Nom : **Katie Denton**

Âge : **12 ans**

Études : **actuellement au collège (en 5^e, en 3^e pour les maths)**



Katie, qu'est-ce qui t'a poussée à te mettre à l'électronique ?

Le Raspberry Pi qu'on m'a offert il y a quelques années ; faire clignoter ses LED en Python m'amusait. Il y a un an, j'ai voulu une imprimante 3D et mon père m'a dit oui, mais seulement si je la construisais moi-même. Je l'ai fait et elle a été présentée sur le blog de RPi. J'ai alors été invitée au programme *Girls & Boys Mentoring* de la Nasa. C'était parti.

Qui sont tes idoles (scientifiques) ?

Ada Lovelace, qui a écrit le premier programme informatique lorsqu'elle travaillait sur la machine analytique de Babbage, et Marie Curie, la première femme à avoir été récompensée par un Nobel, et la seule personne à en avoir reçu deux dans des disciplines scientifiques différentes.

Connais-tu d'autres filles de ton âge qui s'intéressent à l'électronique ?

Juste quelques-unes, il n'y a pas assez de filles qui s'intéressent à l'électronique ou à la robotique. J'essaie de changer ça. Au collège, j'ai formé une équipe de filles qui font de la robotique, et j'ai aussi mon blog *MakerKatie* qui montre ce que les filles peuvent faire quand on leur en donne l'occasion.

Est-ce qu'un club international de filles passionnées d'électronique t'intéresserait ?

Carrément, oui ! C'est dur de trouver des filles qui partagent la même passion. J'aimerais un forum pour partager mes idées et apprendre des autres.

Tu as un blog, tu aimerais écrire un livre aussi ?

Je préfère le blog. Sur un blog on peut mettre des vidéos et montrer du code actif.

Qu'espères-tu accomplir durant les cinq années à venir ?

Survivre au lycée, intégrer une fac prestigieuse comme le MIT ou Stanford, apprendre encore plein de choses en électronique et programmation, et réussir des projets vraiment sympas.

Si je t'offrais 500 \$ de bons d'achat pour l'e-choppe, qu'achèterais-tu et pourquoi ?

D'abord des livres de robotique et d'intelligence artificielle. Ce sont des domaines d'avenir qui m'intéressent. Ensuite le kit de 37 capteurs pour RPi, je trouve toujours une raison d'utiliser plus de capteurs. Puis je ferais un stock de cartes de prototypage. Une RPi avec une caméra, parce que c'est toujours marrant, et j'adorerais avoir une seconde Red Pitaya pour réaliser des projets avec son FPGA, ses entrées analogiques rapides et ses 4 Go de RAM. ◀

(150675 – version française : Hervé Moreau)

hexadoku casse-tête pour elektorniciens

Taille tôt, taille tard, mais taille en mars !

Le printemps arrive, il est temps de sortir au jardin pour planter, nettoyer, traiter... L'hiver a été doux, il ne faudra peut-être pas trop tarder à finir les opérations de taille. Toutefois le mois de mars alterne les journées ensoleillées et celles arrosées par des giboulées. En attendant une éclaircie, restez au sec et remplissez votre grille mensuelle d'hexadoku. Le nettoyage de printemps pourra reprendre plus tard.

Une grille hexadoku est composée de chiffres du système hexadécimal, de 0 à F. Remplissez le diagramme de 16 x 16 cases de telle façon que **tous** les chiffres hexadécimaux de 0 à F (0 à 9 et A à F) n'apparaissent **qu'une seule et unique fois** dans chaque rangée, colonne et carré de 4 x 4 cases (délimités par un filet gras).

Certains chiffres, déjà placés dans la grille, en définissent la situation de départ.

Pour participer, inutile de nous envoyer toute la grille, il suffit de nous donner **la série de chiffres** sur fond grisé.



Participez et gagnez ! Nous tirerons au sort **trois bonnes réponses** internationales reçues dans les délais ; leurs auteurs recevront chacun un chèque-cadeau d'une valeur de **50 €** à valoir dans l'e-choppe d'Elektor. À vos crayons !

Envoyez votre réponse (les chiffres sur fond grisé) avec vos coordonnées par courriel, avant le **24 mars 2016** à l'adresse **hexadoku@elektor.fr**

Les gagnants

La solution de la grille du numéro de décembre est **6AE32**

Les trois bons Elektor d'une valeur de 50 € vont à :

Jean-Paul **Winberg** (France), Haythem **Zidi** (France), Wolfgang **Köberl** (Allemagne).

Bravo à tous les participants et félicitations aux gagnants !

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

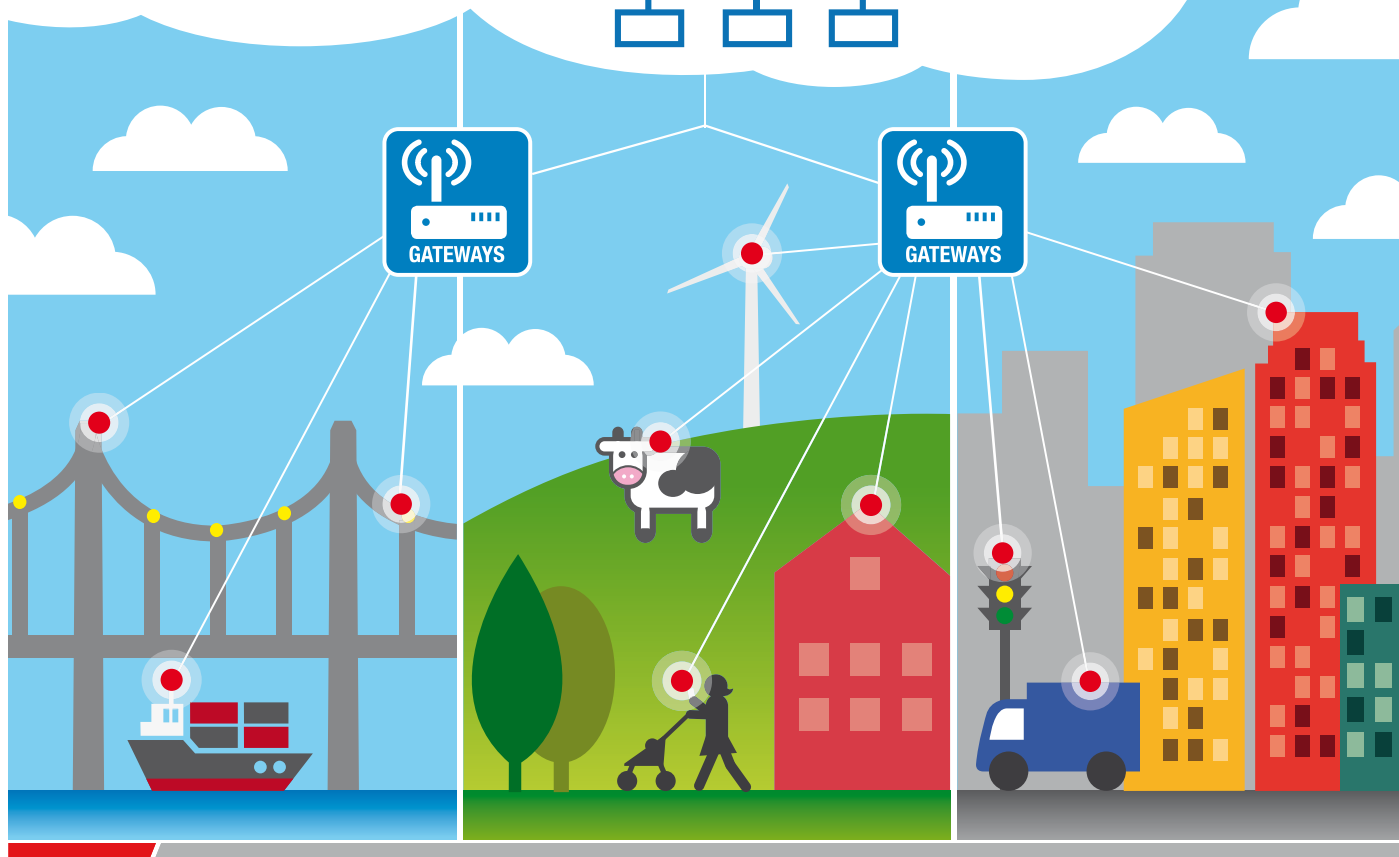
C	A	F	6	0	7	2	B	D	5	4	8	E	1	3	9
0	2	4	E	9	8	5	D	1	3	7	B	C	A	6	F
9	D	5	1	6	A	E	3	2	C	F	0	4	B	7	8
7	3	B	8	C	F	1	4	6	9	A	E	0	2	5	D
8	9	1	5	A	E	3	7	0	6	B	2	D	C	F	4
A	6	C	F	B	D	8	1	4	7	9	3	5	E	2	0
B	E	D	3	2	0	4	5	8	F	C	A	1	6	9	7
2	4	0	7	F	6	C	9	5	1	E	D	3	8	B	A
1	5	6	9	E	2	7	8	C	A	D	F	B	4	0	3
3	8	A	4	D	B	6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

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

Longue portée et faible consommation réunies

dans les modems LoRa faciles à utiliser

Contrôleur réseau basé dans le nuage



Face au marché de l'Internet des objets en plein essor, Microchip propose la solution longue portée LoRa™ (Long Range) pour répondre à la demande croissante en nœuds d'extrémité pour la connectivité longue portée, en faible consommation pour les produits alimentés par batterie et en solutions économiques favorisant le déploiement à grande échelle. La solution de Microchip, le RN2483, utilise une modulation numérique à étalement du spectre et un protocole propriétaire dans la bande de fréquence RF inférieure à 1 GHz pour permettre :

- ▶ Longue portée : > 15 km
- ▶ Faible consommation : jusqu'à 10 ans de longévité des batteries
- ▶ Capacité réseau élevée : jusqu'à 1 million de nœuds

Le RN2483 est un modem 433/868 MHz (inférieur à 1 GHz) entièrement certifié LoRa qui fait office de terminal de nœud d'extrémité sur une infrastructure réseau LoRa. Ce modem au facteur de forme compact intègre la pile de protocole complète LoRaWAN™ ainsi qu'un microcontrôleur PIC®, et est facile à configurer à l'aide de commandes ASCII simples via l'interface UART, ce qui réduit énormément le temps de développement.



Modem longue portée LoRa
inférieur à 1 GHz 433/868 MHz
(version européenne)

microchip
DIRECT
www.microchipdirect.com

 **MICROCHIP**

www.microchip.com/LoRa

Le nom et le logo de Microchip, le logo Microchip et PIC sont des marques déposées de Microchip Technology Incorporated aux États-Unis et dans d'autres pays. Le nom LoRa et le logo associé sont des marques commerciales de Semtech Corporation ou de ses filiales. Toutes les autres marques commerciales ci-dessus mentionnées sont la propriété exclusive de leurs propriétaires respectifs. © 2016 Microchip Technology Inc. Tous droits réservés. DS70005220A. MEC2054Fre01/16

Une qualité professionnelle à prix discount!

reichelt.fr
elektronik

Sprays de contact chimie –
et l'électronique fonctionne



Puissant nettoyant de
contact dissolvant les oxydes

Les recherches scientifiques attestent :
KONTAKT 60 dissout les couches d'oxydes
tenaces, les résistances au contact sont réduites.

- nettoie les contacts qui s'auto-corrodent
- repousse la poussière
- rétablit le flux du courant

Pots	Prix/l	N° de commande	Prix
Spray 100 ml	33,53	KONTAKT 2010	3,35
Spray 200 ml	30,67	KONTAKT 202	6,13
Spray 400 ml	20,90	KONTAKT 203	8,36



Sécurité
contrôlée :
Sprays froids
inflammables

Les sprays froids contiennent des
mélanges froids purs de haute qualité
ayant un effet de refroidissement. Ils sont
utilisés pour la recherche d'erreurs
thermiques dans l'électronique, pour le retrait
à froid et la congélation rapide de petites
surfaces et pour contrôler le fonctionnement
des sondes thermiques.

Effet de refroidissement jusqu'à max -52 °C. Conformément
au contrôle de sécurité, dans le cas d'une utilisation conforme,
il n'y a pas de risque d'inflammation ni d'explosion.

Pots	Prix/l	N° de commande	Prix
Spray 200 ml	35,09	KONTAKT 316	7,02
Spray 400 ml	29,83	KONTAKT 317	11,93

S'ABONNER MAINTENANT !

Newsletter

Recevez chaque semaine les
toutes dernières informations

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



- ✓ Plus de 45 ans d'expérience
- ✓ Envoi en 24 heures
- ✓ Plus de 50 000 produits

Les langues de notre boutique:

Payment
Methods:

SOFORT
BANKING



VISA

PayPal



POSTE À SOUDER WX

Un poste – de nombreux avantages

Puissant poste à souder à 1 canal de 200 W. Idéal pour les applications solaires,
la technique HF, la technique LED, les conducteurs bandes plates Tapping & Bussing
et pour les circuits imprimés en aluminium.

- temps de chauffe rapide
- contrôle visuel du processus grâce à la fonction signal LED
- haute stabilité/précision thermique : $\pm 2^\circ\text{C}$
- supporte les outils jusqu'à 1x 200 W
- détection automatique de l'outil

WELLER WX 1

323,53

Unité d'alimentation WX 1, 200 W / 230 V



WELLER WX 1010

494,96

Le kit **WELLER WX 1010** contient :

Unité d'alimentation WX 1, 200 W/230 V,
Fer à souder WXP 120 (120 W, 24 V),
1x support de sécurité

ACCESSOIRES COMPATIBLES WX :

Kits Support &
Fer à souder



KIT avec WXP 65, 65 W

Le petit fer à souder rapide de 65 watts convient très bien pour les opérations de
soudage fin avec un besoin en chaleur plus important.

Bestellnummer:

WELLER WXP65SET **190,84**

Power-Response

KIT avec WXP 120, 120 W

Cet outil universel permet d'effectuer des opérations de soudage sur des petites
et des grosses pièces avec un besoin en chaleur élevé.

Bestellnummer:

WELLER WXP120SET **171,09**

Power-Response

KIT avec WXP 200, 200 W

Le plus puissant des outils WX. Idéal pour les opérations de soudage avec un besoin
en chaleur supérieur, par ex. pour des backplanes LED (fonds de panier informatique).

Bestellnummer:

WELLER WXP200SET **210,04** Set

WELLER WXP200 **194,12** LötKolben einzeln

Power-Response

KIT avec WXMT, 2x 40 W

Pincette très fine et étroite. Optimal pour souder et dessouder de très petites
pièces SMD.

WELLER WXMTSET **257,14**

Active-Tip

Commander maintenant! **www.reichelt.fr**

Assistance téléphonique en anglais : **+49 (0)4422 955-360**

Prix du jour ! Prix à la date du : 1.2.2016

Prix en € hors T.V.A., frais de port en sus
reichelt elektronik, Elektronikring 1, 26452 Sande (D)