

e lektor

numéro double | édition spéciale | 132 pages
sélection variée de projets électroniques de classe

thermomètre sans fil Bluetooth Low Energy BL600 • T-Board sans fil • passerelle entre cartes à puce et lecteur PC/SC • Beep : testeur logique sonore • générateur de fonctions • commande gestuelle sur Raspberry Pi

synthétiseur musical J2B



avec
webinaire
en ligne
gratuit

plate-forme musicale ouverte

- eBoB ChipCap2 : capteur d'humidité/de température ChipCap2
- eBoB ADS1115 : 4xCA/N 16 bits I²C
- eBoB-SCAP : passerelle USB ↔ ISP
- programmeur USBprog 5.0
- alimentation VariLab 402 [2]
- **nouveau** : µC ARM pour débutants - 1^{ère} étape



2^e édition revue et augmentée du livre

L'électronique pour les débutants

NOUVEAU : troisième kit d'initiation disponible

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

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

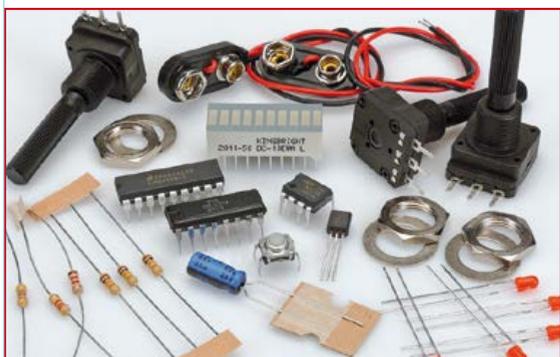
laissent en suspens. L'auteur démystifie l'électronique en n'utilisant que ce qu'il vous faut de théorie pour aborder la pratique : identifier les composants et leur rôle, les récupérer, les tester et les ranger ; lire un schéma ; choisir ses outils ;

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

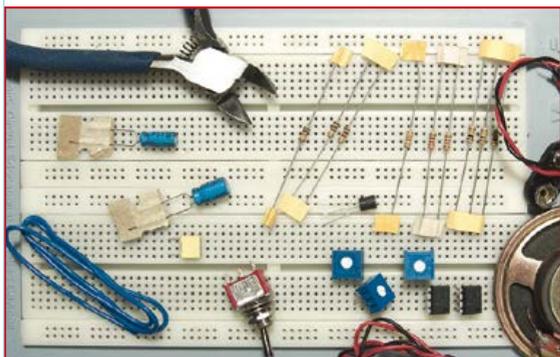
Kit n°1 : orgue | 119016-73 | **24,50 €**



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

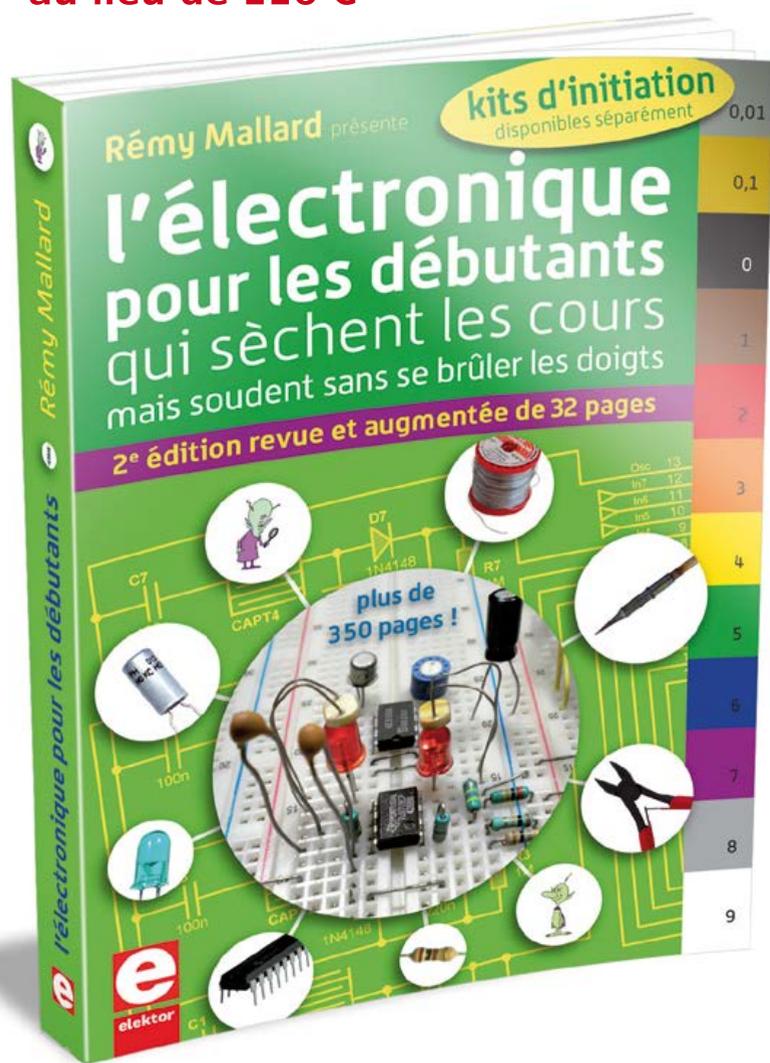


Kit n°1 : sirène | réf. 119016-71 | **24,50 €**



offre spéciale :

livre + trois kits = 99 €
au lieu de 116 €



elektor

ISBN 978-2-86661-186-6

édition revue et augmentée | 352 pages - **42,50 €**

info et commande : www.elektor.fr/debut

abonnez-vous à Elektor devenez membre GOLD!



Les avantages de la formule Gold :

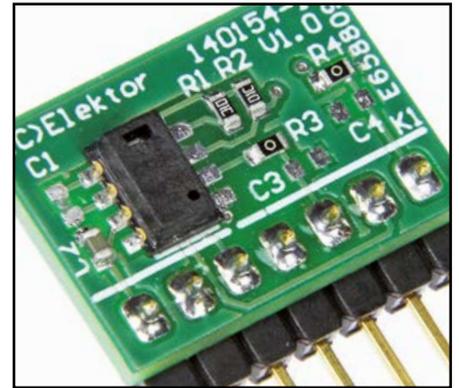
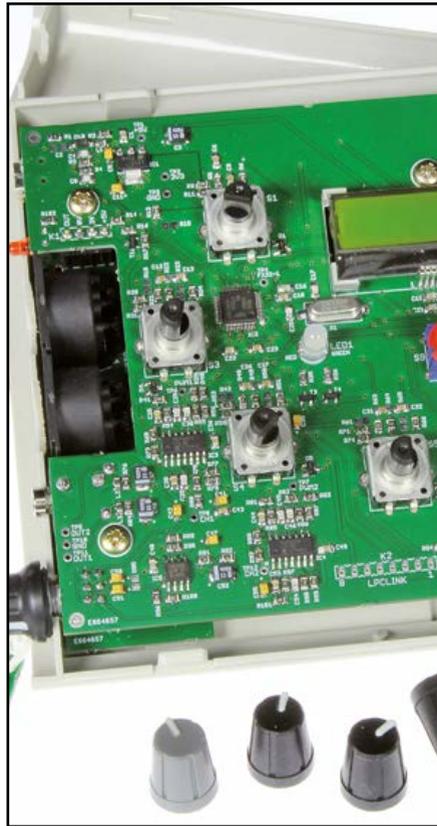
- 8 numéros et 2 numéros doubles sur papier et sous forme numérique
- 1 DVD-ROM annuel
- 10% de remise sur TOUS les produits sur www.elektor.fr
- l'accès direct en ligne à Elektor.LABS, le laboratoire virtuel d'Elektor
- des projets électroniques inédits exclusifs par voie de courrier électronique
- et enfin le bonus exclusif : une reliure gratuite (sur demande) pour des projets inédits.



VOUS N'AVEZ PLUS BESOIN DU MAGAZINE IMPRIMÉ ?
Optez pour la formule **Green**
l'édition numérique (PDF) d'Elektor
sans papier ni DVD-ROM
économique et écologique
... avec les mêmes avantages !



Demandez votre carte de membre sur www.elektor.fr/membres



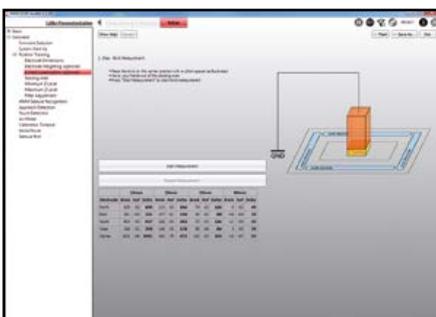
● communauté

6 de nous à vous

● labs

71 une capa et puis ça va

122 kit Hillstar de Microchip
puce de commande tactile et gestuelle



● projets

30 GestIC & 3D Touchpad :
manip. n°2

commande gestuelle sur RPi

32 passerelle entre cartes à puce et
lecteur PC/SC

ausculter les cartes à puce
synchrones avec un lecteur PC/SC

38 CC2-eBoB

capteur d'humidité/de température
ChipCap2

44 BoB ADS1115

4 voies de conversion A/N à 16 bits
sur I2C

46 BOB-SCAP

Une passerelle du passé vers le
futur : USB <-> ISP

50 émetteur synchro FM

oscillateur Clapp

51 récepteur synchro FM

à oscillateur Clapp

52 pilote de LED riquiqui pour
lampes de poche

remplacer l'ampoule par une LED

55 lampes à incandescence

de l'art de planter les bulbes

59 amplificateur de signal
pour oscilloscope USB

conçu entre autres pour *PicoScope*

60 générateur de sinus à filtre RII

code compact pour petits
microcontrôleurs

64 T-Board sans fil

XBee, Bluetooth, WiFi...
écoutez Radio Plaque-à-Trous

67 adaptateur de niveau 3 états

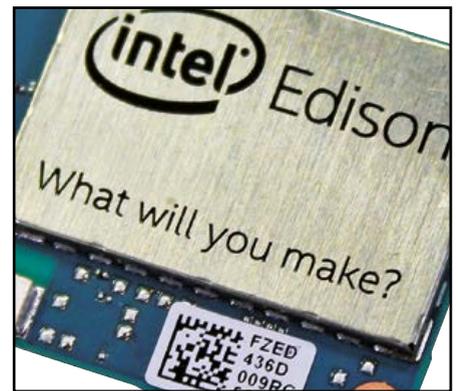
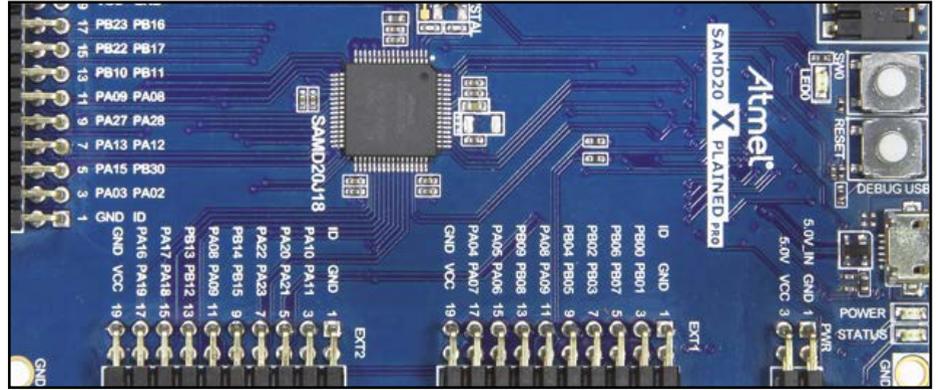
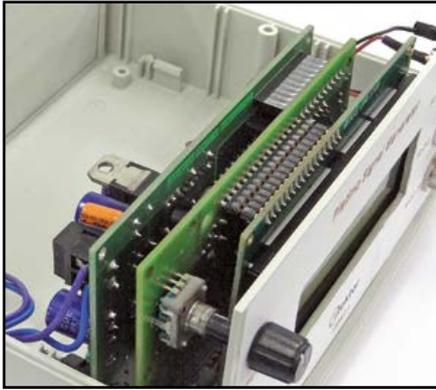
utile avec Arduino

68 Beep : testeur logique sonore

aussi pour signaux rapides

72 thermomètre sans fil Bluetooth
Low Energy

Affichage de la température à
distance sur votre téléphone tactile



- 80 générateur de fonctions Platino**
sinus, dents de scie, rampes... aux manettes
- 88 synthétiseur J2B**
plateforme musicale numérique ouverte
- 100 alimentation VariLab 402 (bis)**
la carte microcontrôleur/affichage
- 105 tension de sortie précise**
en tenant compte de la tension de référence interne du régulateur
- 106 programmeur USBprog 5.0**
programmeur open source + interface web
- 111 transistors tétrodes**
drôle de composant n°12
- 112 microcontrôleur ARM pour néophytes** 1^{ère} étape
une carte, un logiciel et un 1er programme pour passer de 8 à 32 bits
- 124 Design Spark 17e jour :**
plus sur les composants

● industrie

- 8 cap sur l'innovation**
Le numérique :
quels impacts pour l'industrie ?
- 12 salon CARTES SECURE CONNEXIONS 2014**
de notre envoyé spécial P. Gueulle
- 22 Edison d'Intel**
aimez-vous le boudin de pingouin ?

● Magazine

- 16 le marquage CE pour les béotiens**
- 26 EveryCircuit :**
l'électronique virtuelle
Mettez le doigt dans l'électronique
- 126 Digi-Disco 1978**
jusqu'au bout du 7406
- 129 bientôt dans Elektor**
- 130 hexadoku**
casse-tête pour elektorniciens



38^{ème} année, n° 439/440
janvier/février 2015

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

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

@ : 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.elektor-magazine.fr

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

DROITS D'AUTEUR :
© 2015 Elektor International Media B.V.

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

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

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

Qui veut aller loin ménage sa monture !

Avez-vous remarqué, sur la couverture de ce numéro, le mot « *BoB* » apparaît dans pas moins de trois titres d'articles ? Ne s'agirait-il pas d'une erreur ?

D'accord, c'est un numéro double, mais en quoi cela explique-t-il cette prolifération ? Si ce n'est pas une bourde, c'est peut-être un signe, mais de quoi ? Eh bien, c'est une tendance. Elle n'est pas accidentelle du tout, c'est même le résultat d'un choix délibéré d'Elektor. C'est aussi le début d'une série.

Le *BoB* est pour l'électronicien le remède magique contre la miniaturisation des composants et plus précisément de leurs broches. Le terme de *BoB*, pour *Break out Board* est bien commode, il sonne bien, mais reste cryptique si on ignore le sens figuré de cette désignation. En anglais, *to break out* renvoie entre autres à l'idée de *s'échapper*. Le *BoB* ne désigne donc pas une carte détachable (*snap-off board* ou *punch-out board*), mais une carte qui rend accessible des signaux qui autrement ne le sont pas en raison – le plus souvent mais pas nécessairement – de la miniaturisation extrême des broches. Autrement dit, pour filer la métaphore, le *BoB* met au large des signaux jusqu'alors emprisonnés. Si on adopte une dénomination française, *carte de liaison* par exemple, on saisit d'emblée que le *BoB* établit le lien entre le monde extérieur (au pas de 2,5 mm !) et chacune des lignes d'un circuit si serré qu'elles sont inaccessibles à nos doigts. Un autre nom français évocateur a été proposé pour désigner ces nouveaux bijoux : *monture*.

Désormais, au fil des mois, Elektor proposera donc de nouvelles montures pour toutes sortes de circuits intégrés (p. 44), de capteurs (p. 38) et de modules intéressants, mais destinés à l'industrie et peu compatibles de ce fait avec un environnement d'expérimentation. Ces cartes de liaison, ou ces *BoB* si vous préférez, permettront à tous les électroniciens d'utiliser facilement des composants dernier cri sans que leur forte miniaturisation n'en compromette la maniabilité. Un très bon exemple de cet avantage nous est fourni par le remarquable module *Bluetooth Low Energy* BL600 utilisé sans *BoB* p. 72 de ce numéro dans le **thermomètre d'extérieur Bluetooth** de Jennifer Aubinais. Une superbe application, mais un module encore difficile à manipuler aujourd'hui. Le mois prochain, ce module-là aura lui aussi sa monture *eBoB*. Et bien d'autres suivront.

Bonne lecture !

Denis Meyer

(R)APPEL : Elektor cherche des auteurs de LIVRES sur l'électronique et les domaines connexes : audio, domotique, robotique, micro-informatique, énergies renouvelables, modélisme...

Partagez votre savoir, votre passion et le fruit de votre créativité ! Nos productions s'adressent aussi bien aux amateurs qui souhaitent se perfectionner qu'aux professionnels qui aiment ce qu'ils font.

En tant qu'auteur, vous en tirerez d'immenses satisfactions, mais aussi des revenus substantiels.

Notre équipe

Rédacteur en chef :	Denis Meyer (redaction@elektor.fr)
Rédaction internationale :	Harry Baggen, Jan Buiting, Jaime González-Arintero, Jens Nickel
Laboratoire :	Thijs Beckers, Ton Giesberts, Luc Lemmens, Mart Schrooijen (illustrations) Clemens Valens (responsable), Jan Visser, Patrick Wielders (vidéo)
Coordination :	Hedwig Hennekens
Ont coopéré à ce numéro :	Robert Grignard, Hervé Moreau, Kévin Petit, Guy Raedersdorf, Mariline Thiebaut-Brodier
Service de la clientèle :	Cindy Tyssen & Vanessa Noville
Graphistes :	Giel Dols
Elektor en ligne :	Daniëlle Mertens



**France**

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

**United Kingdom**

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

**USA**

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

**Germany**

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

**Netherlands**

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

**Spain**

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

**Italy**

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

**Sweden**

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

**Brazil**

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

**Portugal**

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

**India**

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

**Russia**

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

**Turkey**

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

**South Africa**

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

**China**

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

Notre réseau



VOICE COIL



audioexpress



vous connecte à



Nos annonceurs



Embedded World 2015

www.embedded-world.de 25

Microchip

www.microchip.com/get/eu3DTouchPad . . . 132

NIDays 2015

www.nidays.fr 63

Pico

www.picotech.com/PS303 131

Reichelt

www.reichelt.fr 59

Pour placer votre annonce dans le prochain numéro d'Elektor

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

Vos correspondants

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

cap sur l'innovation

Le numérique : quels impacts pour l'industrie ?



Programme CAP'TRONIC

Fondée par le CEA et Bpifrance, et financée par le Ministère de l'économie, des finances et de l'industrie, l'association JESSICA France est chargée de la mise en œuvre du programme CAP'TRONIC. Ce programme national d'accompagnement a pour objectif d'aider les PME françaises, quel que soit leur secteur d'activité, à améliorer leur compétitivité grâce à l'intégration de solutions électroniques et de logiciel embarqué dans leurs produits.



En 2013, CAP'TRONIC a aidé plus de 2700 PME issues aussi bien du secteur de l'électronique que des secteurs traditionnels. Les ingénieurs électroniciens de CAP'TRONIC, présents dans toute la France, apportent une aide concrète et ciblée : choix technologiques, mise au point du cahier des charges, accompagnement dans la phase de prototypage électronique... Leur travail de soutien peut prendre d'autres formes : séminaires, ateliers...[1]

Le 15 octobre dernier, la Cité Internationale Universitaire de Paris accueillait la 2^e édition de la journée *CAP sur l'innovation*, organisée par CAP'TRONIC et la Direction Générale des Entreprises.

Mariline Thiebaut-Brodier (elektor)

Trophées CAP'TRONIC 2014

Cette journée d'échanges et de témoignages de chefs d'entreprises était aussi l'occasion de découvrir les lauréats des Trophées CAP'TRONIC 2014. Ils récompensent les plus belles innovations des PME qui ont été accompagnées par le programme et qui se sont engagées à développer le produit concerné. Après une présentation rapide des projets, un trophée a été remis à l'une des trois sociétés retenues, dans chacune des quatre catégories. Un cinquième trophée a été attribué suite au vote du public.

Catégorie Produit à l'usage du grand public

- **NIGILOC** (Dax, [3]), **lauréat du vote du public** : le système **NigiBike & NigiScoot** permet de géolocaliser son vélo ou son scooter grâce à un système embarqué et à une application sur ordiphone. Lorsque le vélo ou le scooter se déplace, le système vérifie la présence du téléphone portable du propriétaire ; en cas d'absence, ce dernier est informé du vol de son deux-roues et de sa localisation.
- **STARWAY** (Luynes, [4]) : le **Vélo Facile** est un vélo à assistance électrique qui s'adresse à tous ceux qui veulent disposer d'un moyen de transport propre. Pour simplifier son utilisation, le vélo fonctionne sans dérailleur (remplacé par une courroie), ni changement de vitesse.
- **WISTIKI** (Paris, [5]), **lauréat 2014** : le **Wist** est un petit accessoire de la taille d'une pièce de deux euros qui permet de retrouver ses objets. Il suffit d'accrocher ou de coller un Wist sur les objets à pister. Une application gratuite sous Apple ou Android permet de connaître la dernière position GPS de l'objet enregistrée par son ordiphone ; ensuite, lorsqu'on est à proximité de l'objet, on peut le faire sonner.

Catégorie Industrie et services

- **ADEUNIS RF** (Crolles, [6] [7]) : **Vokkero** est un système de communication audio pour les arbitres ou les équipes

d'intervention en milieu industriel qui ne nécessite aucun relais fixe. Il est équipé d'un filtre de bruits breveté, une parade à tous les problèmes de communication dans un environnement bruyant (stades, BTP, industrie).

- **ARCURE** (Paris, [8]), **lauréat 2014** : **Blaxtair** contribue à améliorer la sécurité autour des engins de chantier. Il s'agit d'une caméra intelligente qui détecte les piétons et les localise avec précision. Cela permet d'éviter les alarmes intempestives. Blaxtair s'adapte à tous types d'engins mobiles dans les secteurs de la construction, des mines et carrières, ou de la logistique : chariot élévateur, compacteur, chargeuse, niveleuse, pelle hydraulique, camion, machine guidée, etc.
- **BOA CONCEPT** (Saint-Etienne, [9]) : cette société propose un **convoyeur programmable intelligent**. Il est composé d'éléments standard, donc réutilisables, raccordés sur site en mode *plug&play*, comme des briques *Lego*. Il n'y a ni câblage sur site, ni automate central ou encore de logiciel à programmer (*Warehouse Control System* (WCS)) : l'utilisateur peut modifier tout seul et facilement son installation, la topologie est reconnue automatiquement. Il peut également assurer sa maintenance par échange standard des composants.

Catégorie eco-innovation

- **ECHY** (Champs-sur-Marne, [10]), **lauréat 2014** : cette société a développé un **système d'éclairage hybride** qui permet d'introduire la lumière naturelle du soleil à l'intérieur des bâtiments, grâce à un transport du flux lumineux via des fibres optiques. Lorsque la lumière naturelle n'est plus suffisante (la nuit par exemple), un éclairage électrique à LED prend le relais automatiquement et progressivement.
- **ECOBATEC** (Montélimar, [11]) : **BatTriBac Pro** est une borne de collecte intelligente des batteries automobiles usagées. Son électronique embarquée permet d'assurer les fonctions de diagnostic, de tri sélectif, de traçage ainsi qu'une gestion documentaire des batteries collectées. Ce diagnostic technique permet de déterminer la filière de traitement la plus adaptée (régénération ou recyclage) pour tous les types de batteries.
- **SEWOSY** (Bernolsheim, [12]) : cette société a conçu un



bandeau électromagnétique à faible consommation qui permet d'ouvrir/fermer des portes d'entrée souvent sollicitées (immeubles, bâtiments tertiaires). Ce système permet d'économiser plus de 70% d'énergie par rapport aux autres bandeaux électromagnétiques présents sur le marché : on passe de 12 W à 1 W.

Catégorie Jeune entreprise (société de moins de 3 ans)

- **BODYCAP** (Hérouville-Saint-Clair, [13]) : **e-celsius** est une petite gélule, qui permet, une fois avalée par le patient, de mesurer sa température à intervalle régulier. Les données de ce capteur électronique miniaturisé sont transmises par radio (433 MHz) à un moniteur. Ce suivi en continu permet de surveiller la température des patients pendant les interventions puis pendant la période post-opératoire.
- **NAÏO TECHNOLOGIES** (Toulouse, [14]), **lauréat 2014** : **Oz** est un petit tracteur électrique, capable de désherber mécaniquement (sans produits chimiques) entre les rangées de cultures de manière totalement autonome. Si on lui adjoint une remorque, il peut servir à transporter des légumes au moment de la récolte.
- **NOVITACT** (Lacroix-Saint-Ouen, [15]) : **Feeltact** est un bracelet connecté qui permet d'émettre et de recevoir des informations en s'affranchissant de l'ouïe et de la vue. Le porteur du bracelet Feeltact émet des messages en pressant des boutons, il reçoit des messages sous forme de vibrations par le

biais de vibreurs situés tout autour du poignet. Les messages mis en œuvre sont facilement interprétables en jouant sur la durée, l'intensité et la séquence des vibrations. Ce nouveau concept est destiné aux professionnels pour la transmission d'information dans des situations difficiles, voire dangereuses (convoyeurs de fonds, agents de sécurité...).

Atelier de fabrication numérique

Cette journée s'articulait autour de tables rondes avec pour fil conducteur « le numérique ». Il est ressorti de la discussion sur les objets connectés que les entreprises doivent réfléchir aux services à proposer autour de ce type d'objets. En effet, la marge sur la vente de ces objets est souvent très faible. La majeure partie du bénéfice devrait être réalisée sur les abon-

nements proposés avec ces objets. C'est également une révolution puisqu'un lien permanent est établi entre le fabricant et le consommateur.

La recherche d'un modèle économique est également une des questions évoquées lorsqu'on parle d'ateliers de fabrication numérique. La DGE a présenté les résultats de son étude sur le sujet [2]. Petite précision de terminologie : le terme *fab lab* (abréviation de *Fabrication Laboratory*) s'applique aux ateliers qui respectent la *Charte des Fab Labs* initiée par le MIT. En résumé, les *fab labs* sont un réseau mondial de laboratoires locaux, qui rendent possible l'invention en donnant aux individus accès à des outils de fabrication numérique. La Charte impose entre autres que ces ateliers soient ouverts au public et contribuent aux projets des autres *fab labs*.

Les différentes structures (111 au total) interrogées par la DGE pour l'enquête ne satisfont pas toutes les critères de la Charte, on y trouve des ateliers ouverts (*makerspace, hackerspace...*), des ateliers de fabrication numérique (au sens large) et des lieux ouverts d'innovation. Leurs points communs sont :

- outils numériques et à commande numérique disponibles à faible coût
- prototypage, réparation, adaptation, production d'objets physiques
- accessibilité et ouverture
- forte dimension communautaire



- collaboration, ce qui soulève des questions sur la propriété industrielle.

Cette étude a permis de définir quatre catégories d'ateliers :

- Atelier institutionnel : structure de service public qui met à disposition des machines et accompagne des projets. Ces ateliers sont généralement fréquentés par des étudiants et/ou des chercheurs, parfois ouverts au grand public.
- Atelier d'entreprise : structure de prototypage. C'est un espace qui doit favoriser la créativité et l'innovation des salariés, par un retour au travail manuel par exemple. Généralement il est fermé au grand public.
- Atelier professionnel : structure de prototypage et de production pour un public professionnel. Cela permet aux entreprises d'utiliser des machines qu'elles ne peuvent pas acquérir ou entretenir.
- Atelier tout public : la production n'y est qu'un prétexte pour apprendre. Le prototypage et l'innovation y sont

également présents.

L'apparition de lieux mixtes, à la fois professionnel et grand public, répond entre autres au souci de trouver un équilibre financier. Ces lieux d'échange apportent des services connexes aux entreprises : recrutement de nouveaux talents, formation des collaborateurs à de nouvelles compétences, renforcement de la motivation des collaborateurs. En conclusion, le phénomène des *fab labs* est jeune, en pleine mutation, et ne se résume pas aux imprimantes 3D.

140506

Liens

- [1] Séminaires, ateliers et formations proposés par le programme CAP'TRONIC et JESSICA FRANCE : www.captronic.fr
- [2] Etude de la DGE – Etat des lieux et typologie des ateliers de fabrication numérique : Fab Labs : www.entreprises.gouv.fr/secteurs-professionnels/etat-des-lieux-et-typologie-des-ateliers-fabrication-numerique-fab-labs
- [3] www.nigiloc.com
- [4] www.levelofacile.com
- [5] www.wistiki.com

- [6] www.adeunis-rf.com
- [7] www.vokkero.com
- [8] www.arcure.net
- [9] www.boaconcept.com
- [10] www.echy.fr
- [11] www.ecobatec.fr
- [12] www.sewosy.eu
- [13] www.bodycap-medical.com
- [14] www.naio-technologies.com
- [15] www.novitact.com

Clemens Valens

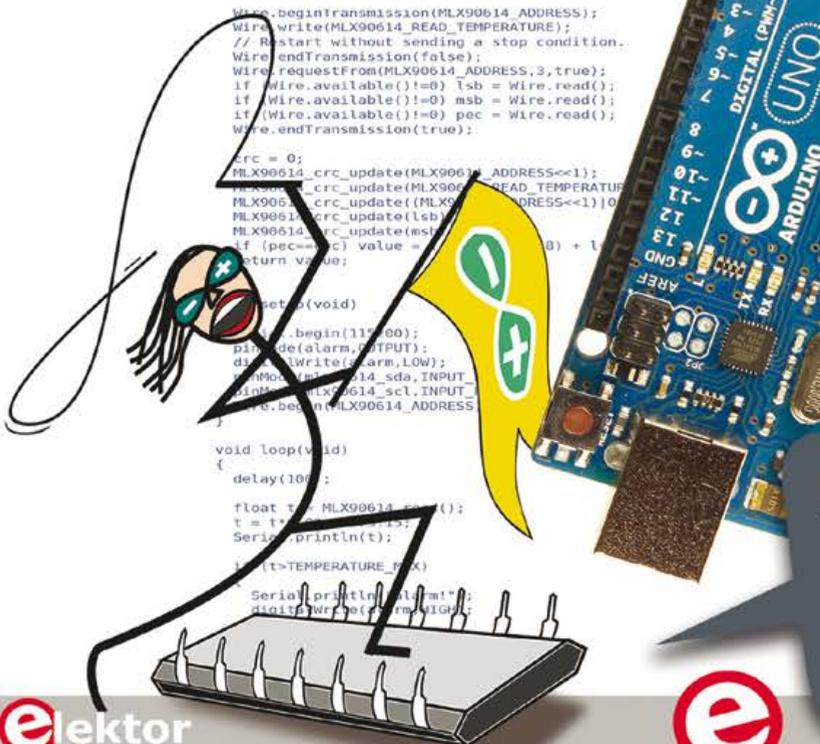
maîtrisez les microcontrôleurs à l'aide d'Arduino

2^e
édition
revue et augmentée

www.elektor.fr/arduino

+35 pages

+ nouvelle carte d'expérimentation polyvalente



elektor

www.elektor.fr/arduino



Témoignage :
« je prête
volontiers
mes bouquins,
mais celui-là...
je veux pas
m'en passer »

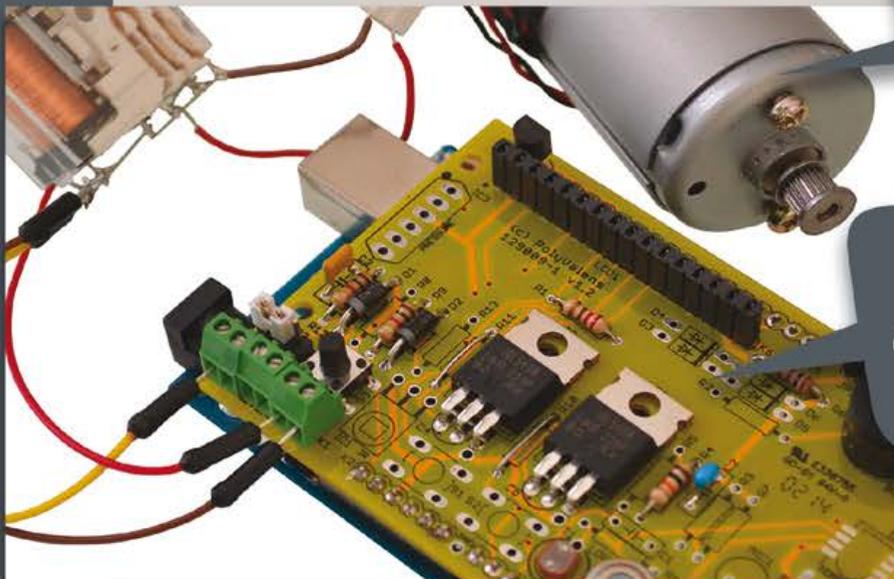
Bernard P.

2^e édition revue
et augmentée
par l'auteur

avril 2014

nouvelles
applications

nouvelle carte
d'expérimentation
polyvalente



385 pages - 42,50 €

isbn 978-286661-195-8

salon CARTES SECURE CONNEXIONS 2014

plutôt calme, et pourtant...



Un peu terne – murmurerait-on entre habitués – cette 29^e édition du salon CARTES, pourtant placée sous le signe plutôt porteur de la mobilité : peu de files d’attente, circulation fluide dans les allées (mais c’était voulu !) et quelques grandes firmes européennes à la voilure ostensiblement réduite. En revanche, 19 noms d’exposants sur 450 commençaient cette année par « Shenzhen », contre 13 en 2013 et seulement 9 en 2012. Un signe ?

Patrick Gueulle (Étretat)

À notre modeste niveau, la moisson fut bonne, avec des interlocuteurs disponibles et un environnement favorable à nos enquêtes et expériences de tout poil...

PROMESSES NON TENUES

Un salon international offre une bonne caisse de résonance à l’annonce d’innovations sensationnelles. Certaines arriveront sur le marché à peine quelques mois plus tard, d’autres ne verront jamais le jour ou seulement sous une tout autre forme (ou firme). En effet, les promesses des commerciaux sont rarement en phase avec les travaux des ingénieurs. Il n’est pas rare que certaines avancées susceptibles de déranger l’ordre établi soient tuées dans l’œuf, en matière de paiement électronique, par exemple.

Parallèlement, des engagements ou des usages remontant à de nombreuses années se trouvent mis à mal, sans faire nécessairement beaucoup de vagues, tant le consommateur a pris la déplorable habitude d’avaloir des couleuvres. Voici deux exemples : Autrefois vendus avec une durée de validité illimitée, les tickets de bus ou de métro à usage unique se périment désormais en deux à trois ans, dès lors qu’ils sont

de type « sans contact ». Et on n’en avertit pas souvent les usagers !

Expressément vendues comme rechargeables, les télécartes de l’opérateur historique français ne le sont plus, pour leur part, depuis juillet 2014. Et tant pis si leur date limite de validité (ou celle de codes de rechargement achetés quelques mois auparavant) n’est pas encore atteinte. Cela pourrait bien annoncer le commencement de la fin pour les cabines publiques, une des premières applications vitrine des cartes à puce...

Mais il y a mieux ! Un argument clef en faveur du *sans contact* a toujours été l’unicité absolue de l’UID (d’où le U initial de ce sigle), cet identifiant individuel gravé en usine dans chaque puce.

Le développeur utilisant des produits Mifare (**NXP**) a toujours cru pouvoir compter, en toute confiance, sur la promesse formelle que jamais deux puces ne sortiraient de l’usine avec le même identifiant.

Depuis quelques années, cependant, l’industriel laissait entendre qu’il serait à court d’UID à 4 octets, dont le nombre maximum (compte tenu de certaines valeurs réservées) est d’environ 3,7 milliards (voir sa note d’application AN10927). Cela donne, au passage, une idée de l’énormité des quantités mises en circulation...

Plutôt que d'abandonner la production du très populaire Mifare Classic pourtant craqué, contrefait, et rattrapé par de nouvelles générations à UID de 7 ou même 10 octets, il a été fort cavalièrement décidé de réemployer, pour de futures séries de composants, des UID « très anciens » (sans autre précision) pourtant garantis uniques !

Le risque de doublons est mince sur le terrain, mais la question n'est pas là : la morale de l'histoire, c'est qu'on ne peut se fier à ce que promettent les fournisseurs ou émetteurs de produits sécurisés, et qu'il vaut mieux prendre les choses en mains soi-même.

En utilisant, au besoin, des UID aléatoires (*Random*), commençant par un octet 08h, et en faisant reposer l'identification sur des moyens cryptographiques. Cela se fait très facilement avec une BasicCard sans contact, par exemple, dont la dernière version (capable d'émuler les Mifare Classic...) existe désormais sous la forme de minces jetons en époxy stratifié, de type porte-clefs. Presque une première mondiale en matière de puces RFID à système d'exploitation ouvert !

QUELQUES LIGNES DE BASIC

Comme tous les ans, nous avons promené sur le salon une BasicCard sans contact, indiscrete, programmée d'une façon pour le moins particulière : sommairement déguisée, cette fois-ci en DESFire, intéressant produit NXP intermédiaire entre Mifare Classic et cartes à microprocesseur multi-application (mais à UID véritablement unique de 7 octets).

Profitant du fait que la puce équipant la ZC7.5 (Rev. D) est précisément d'origine NXP (une SmartMX), nous avons développé un code source ZCBasic agissant (avec prudence !) au tréfonds de son protocole de communication (**listing**). Le résultat était au rendez-vous : la BasicCard ainsi programmée a été acceptée ici ou là comme un *tag* NFC de type 4, aussi bien en lecture qu'en écriture.

Moyennant une programmation un rien plus complexe qui exploite la faiblesse d'une application largement déployée, nous avons même pu faire reconnaître notre ZC7.5 comme un vrai passe sans contact, par l'applet de consultation et de rechargement d'un réseau de transport en

```
#include Misc.def
#pragma Uid(double)
#pragma Ats(hb = Chr$(&H80))
#pragma Sakatqa(&H20 , &H44 , &H03)
Public Sl$ As String * 2 , Id$ As String * 2 = &HD2 , &H76 , Px1$ As String * 3 = &H85 , &H01 , &H00
Public Px2$ As String * 3 = &H85 , &H01 , &H01 , CAP$ As String * 2 = &HE1 , &H03 , NFC As Byte = 0 , N As Byte = 1
Public R1$ As String * 7 = &H04 , &H01 , &H01 , &H01 , &H00 , &H18 , &H05 : REM(C)2013 , Patrick Gueulle
Public R2$ As String * 7 = &H04 , &H01 , &H01 , &H01 , &H03 , &H18 , &H05
Public R3$ As String * 14 = &H04 , &H05 , &H06 , &H07 , &H08 , &H09 , &H80 , &H00 , &H00 , &H00 , &H00 , &H00 , &H42 , &H08
Eeprom Cc$ As String * 48 = Chr$(&H00) + Chr$(&H17) + Chr$(&H10) + Chr$(&H00) + Chr$(&H3b) + Chr$(&H00) _
+ Chr$(&H34) + Chr$(&H04) + Chr$(&H06) + Chr$(&He1) + Chr$(&H04) + Chr$(&H08) + Chr$(&H00) + Chr$(&H00) + Chr$(&H00) _
+ Chr$(&H04) + Chr$(&H06) + Chr$(&He1) + Chr$(&H05) + Chr$(&H08) + Chr$(&H00) + Chr$(&H00) + Chr$(&H00) + Chr$(&H00) + String$(25 , 0)
Eeprom Ndef$ As String * 2048 = Chr$(&H00) + Chr$(&H03) + Chr$(&Hd0) + String$(2045 , 0)
Eeprom Ndefff$ As String * 2048 = Chr$(&H00) + Chr$(&H03) + Chr$(&Hd0) + String$(2045 , 0)
Command &H00 &HA4 Sel(s$ , Disable Le)
Sw1sw2 = &H6A82 : Sl$ = LEFT$(S$, 2)
If Sl$ = Id$ And Lc = 7 And Right$(s$ , 3) = Px1$ Then Sw1sw2 = &H9000 : MID$(Cc$ , 3 , 1) = Chr$(&H10) : NFC = 1
If Sl$ = Id$ And Lc = 7 And Right$(s$ , 3) = Px2$ Then Sw1sw2 = &H9000 : MID$(Cc$ , 3 , 1) = Chr$(&H20) : NFC = 1
If S$ = Cap$ Then Sw1sw2 = &H9000
If S$ = Mid$(cc$ , 10 , 2) Then Sw1sw2 = &H9000
If S$ = Mid$(cc$ , 18 , 2) Then Sw1sw2 = &H9000
End Command
Command &H00 &HB0 Rbin(lc = 0 , S$)
Sw1sw2 = &H6982
If Nfc = 0 Then Exit Command
If Sl$ = Cap$ Then Sw1sw2 = &H9000 : S$ = Mid$(cc$ , P1p2 + 1 , Le)
If Sl$ = Mid$(cc$ , 10 , 2) Then Sw1sw2 = &H9000 : S$ = MID$(NDEF$ , P1P2 + 1 , LE)
If Sl$ = Mid$(cc$ , 18 , 2) Then Sw1sw2 = &H9000 : S$ = MID$(NDEF$ , P1P2 + 1 , LE)
End Command
Command &H00 &HD6 Ubin(s$ , Disable Le)
Sw1sw2 = &H6982
If Nfc = 0 Then Exit Command
If Mid$(cc$ , 15 , 1) = Chr$(&Hff) Then Sw1sw2 = &H6900 : Exit Command
If Sl$ = Cap$ Then Sw1sw2 = &H9000 : Mid$(cc$ , P1p2 + 1 , Lc) = S$
If Sl$ = Mid$(cc$ , 10 , 2) Then Sw1sw2 = &H9000 : MID$(NDEF$ , P1P2 + 1 , LC) = S$
If Sl$ = Mid$(cc$ , 18 , 2) Then Sw1sw2 = &H9000 : MID$(NDEF$ , P1P2 + 1 , LC) = S$
End Command
Command &H90 &H60 Inf(s$)
Call Suspendsw1sw2processing()
N = 1 : S$ = R1$ : ResponseLength = Len(s$) : Sw1sw2 = &H91AF
End Command
Command &H90 &HAF More(s$)
N = N + 1 : Call Suspendsw1sw2processing() : Sw1sw2 = &H9100
If N = 2 Then S$ = R2$ : Sw1sw2 = &H91AF
If N = 3 Then S$ = R3$
ResponseLength = Len(s$)
End Command
```

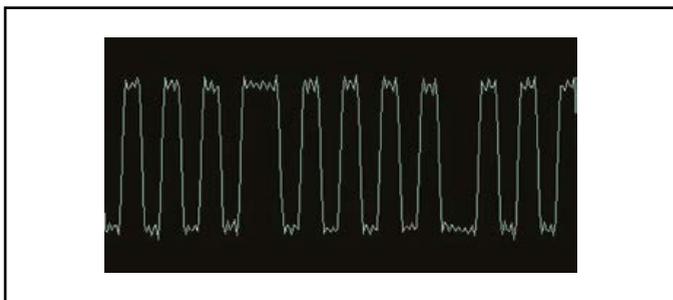
commun. Et en lui attribuant une date de validité de notre choix ! Pourtant, nous n'avons imité aucune des fonctions de sécurité du DESFire...

Lecteurs de cartes pour téléphones et tablettes tactiles

On se souvient avec nostalgie de quelques modèles de téléphones portables dits «bi-fente» : l'une pour la carte SIM (plein format, à l'époque !) et l'autre pour une carte bancaire à puce. De véritables pionniers en matière de paiement sur mobiles. C'est à un concept similaire que plusieurs fabricants proposent d'adapter les *smartphones* (Android ou iOS, notamment) grâce à de nouveaux lecteurs de cartes originaux. L'ACR32 d'**Advanced Card Systems** (Hong-Kong) réunit ainsi sous seulement 60 x 45 x 16 mm, un lecteur de cartes à puce qui a fait ses preuves, et un lecteur de cartes magnétiques. Sans compter qu'une variante (ACR35) existe pour les cartes sans contact NFC.



Deux interfaces sont disponibles : l'habituel connecteur USB assurant une compatibilité PC/SC et CCID (pilote générique), mais aussi et surtout un jack 3,5 mm à 4 circuits. Enfiché dans le connecteur audio d'un *smartphone* ou d'une tablette (ou d'un autre appareil compatible), ce jack assure les communications avec le lecteur par échange bidirectionnel de données Manchester venant moduler une porteuse d'environ 2500 Hz. Comme pour dialoguer avec un capteur externe ou un Arduino, en somme.



La sortie *casque* de droite (contact n° 2) est réservée à un signal de *réveil* du lecteur, économisant ainsi sa batterie Li-ion incorporée, rechargeable par le port USB. Celle de gauche (contact n° 1) véhicule les données vers le lecteur, tandis

que l'entrée *micro* (contact n° 4) rapatrie les données qui en proviennent.

Dans le cas d'une carte magnétique, les données lues sur la ou les pistes (jusqu'à deux) sont transmises à l'hôte sous la protection d'un chiffrement AES-128, voire DUKPT, faute duquel elles seraient évidemment faciles à intercepter.

De quoi imaginer des applications monétiques (ou autres !) pour tous les pays, depuis ceux dont les cartes bancaires ne possèdent pas de puce, jusqu'à ceux qui ont déjà déployé le *sans contact*.

Bien que documenté dans le projet universitaire *HiJack*, le protocole de communication sur jack audio n'est cependant pas normalisé, ce qui fait que les applications développées pour tel ou tel lecteur ne seront, en général, pas utilisables avec ceux de marques concurrentes, déjà assez nombreux sur le marché.

D'autres périphériques pour *smartphones* et tablettes font leur apparition cette année, utilisant cette même interface qui pourrait bien finir par devenir aussi populaire (mais infiniment moins rapide) que l'USB : le minuscule dispositif WAVE du fabricant israélien **OTI**, par exemple, est un émulateur de tag NFC, autonome ou connecté, prometteur dans les domaines du paiement sans contact ou de la billettique, car il peut se substituer aux éléments sécurisés intégrés aux téléphones ou aux cartes SIM, pas toujours dignes de confiance. Ceci est tellement vrai que, de son côté, **INSIDE Secure** a développé un élément sécurisé purement logiciel, le MatrixSSE !

Attention aux cartes SANS CONTACT !

L'an dernier, nous avons fait d'inquiétantes découvertes avec la complicité involontaire d'**Ingénico**, grand spécialiste des terminaux de paiement par carte bancaire : après saisie par le commerçant du montant à débiter, le client disposait de 4 min pour présenter sa carte sans contact ! Un hors-temps excessif pour un mode de paiement prisé surtout pour... sa rapidité ! Un délai en tout cas suffisant pour scanner les passants dans une file d'attente ou dans les transports en commun à l'aide d'un terminal autonome, en leur infligeant subrepticement des débits frauduleux jusqu'à 20 €, voire plus dans d'autres devises.

La réponse à nos demandes d'explications, laborieusement obtenue longtemps après la publication de notre compte-rendu de visite de CARTES 2013, laisse rêveur : « ... conformément aux recommandations du GIE CB » !

Plus tard, une démonstration qu'une grande chaîne de télévision nous avait demandé d'organiser a été décommandée à la dernière minute. Qu'on soit dérangé en haut lieu par notre scénario d'un détournement à la dérobée ne le rend que plus plausible.

Dans ces conditions, nous ne pouvons qu'inciter nos lecteurs (et plus généralement le grand public) à s'équiper d'étuis blindés, dont les cartes ne devront sortir qu'au moment de payer (rapidement) par un geste volontaire.

Pas besoin pour cela de changer de porte-cartes ou de portefeuille : **ASK** sait produire, en masse et à très faible coût, des tickets de blindage en papier *Priva'C*, comme les billets de transport en commun jetables et donc souples, qu'il suffit de ranger tout contre la carte à protéger.

On devine, par transparence, les spires en court-circuit concentriques imprimées dans l'épaisseur, avec une encre conductrice à base d'argent .

Vérification faite, c'est redoutablement efficace de 10 à 30 MHz. Qu'attendent les banques pour en offrir à leurs clients ? En dépit de la demande ferme des autorités, et au vu des faibles quantités commandées, les banquiers traînent les pieds ! À qui profite l'incurie ? Il est vrai que la crainte de fraude favorise la vente d'assurances complémentaires.



Une autre approche, présentée par l'italien **Card Tech**, consiste à intégrer dans la carte elle-même un capteur biométrique souple (du norvégien **IDEX**) avec sa batterie. Sa puce RFID ne s'activera ainsi que si elle est tenue par son utilisateur légitime. C'est séduisant sur le plan technique, forcément un peu cher, mais riche d'applications dans de multiples domaines, bien au-delà de la monétique : les cartes d'électeur, par exemple.

PRISMicide : sécurité ouverte

Tout le monde s'accorde sur la nécessité d'outils de sécurité pour contrer les attaques de pirates isolés ou de programmes gouvernementaux peu scrupuleux (scandale PRISM). Pourtant, il n'existait jusqu'à présent aucune solution ouverte (*open source*) utilisant des éléments sécurisés fiables comme les cartes à puce.

Imaginée par un groupe de professionnels issus d'entreprises françaises qui ont souvent fait parler d'elles dans nos colonnes (Neowave, Plug'up, etc.), la plate-forme de sécurité PRISMicide est composée de cartes à puce *open source* et d'un mini-

lecteur portable *open hardware* utilisable en Bluetooth par les *smartphones* ou les tablettes, et en USB par les ordinateurs de bureau.

En bon français, *open source* et *open hardware* signifient que le code source, les schémas électroniques, et même les plans du boîtier (pour imprimantes 3D) seront rendus publics. Véritable révolution dans ce domaine très opaque, c'est un bien sympathique pavé dans la mare !

PRISMicide se prête idéalement au chiffrement de données (dont les SMS), à la signature électronique, à l'authentification en ligne ou à la messagerie ultra-sécurisée. La première application, dont on parlait déjà avec enthousiasme sur le stand de la maison du Bitcoin, sera un portefeuille sécurisé



(*Bitcoin hardware wallet*). PRISMicide pour Bitcoin permet de garder le contrôle des clés privées en les stockant dans une carte à puce, sans dépendre d'un quelconque service en ligne. Les clés sont ensuite utilisées directement à l'intérieur de la carte, et ne sont plus jamais exposées aux ordinateurs ou *smartphones* servant à préparer les transactions. Bref, le petit lecteur portable sécurisé rend impossible toute attaque. Avant de valider une opération, l'utilisateur doit insérer la carte dans son lecteur qui affiche sur son propre écran les détails (infalsifiables) de la transaction demandée. C'est aussi sur ce lecteur que l'utilisateur doit entrer le code PIN pour utiliser la carte. On reconnaît ici le parcours de paiement courant avec une carte bancaire traditionnelle.

Ce terminal de paiement personnel (en phase de prototypage) sera utilisable en toute confiance, y compris sur des ordinateurs ou *smartphones* non sécurisés, ainsi que chez des tiers (commerçants...). Encore un projet dérangeant à suivre avec attention.

(140428)

le marquage CE pour les béotiens

Ne fuyez pas, continuez la lecture de cet article, vous y apprendrez peut-être des choses surprenantes. Je resterai le plus pragmatique possible, en évitant tout charabia. Promis.

Robert Lacoste (Chaville)

Dans mes articles précédents, j'ai abordé des sujets hautement techniques : adaptation d'impédance, réflexion de signaux, routage de circuits imprimés, etc. Ce mois-ci, on change de registre avec la réglementation et plus particulièrement le marquage CE. Un sujet réputé austère, dont certains se gaussent, arguant que la réglementation ne fait qu'énoncer des évidences ou des exigences applicables uniquement aux grands fabricants. Ils ont tort...

Il est impossible de faire le tour d'un tel sujet en quelques pages. Vous y trouverez des bases pour comprendre, mais pas assez pour faire de vous des experts en normalisation. 99 % de ce que vais raconter est vrai, mais vérifiez toujours auprès de sources officielles...

Marquage CE ?

En pratique le marquage se concrétise par le petit logo CE normalisé (**fig. 1**) qui doit être visible sur la plupart des produits vendus en Europe. Certaines familles de produits sont exemptées, mais la quasi-totalité des appareils électroniques est concernée. Sa taille aussi est normalisée : au moins 5 mm de haut. CE veut dire *Conformité Européenne* (et non *China Export* comme certaines mauvaises langues le laissent entendre).

Déblayons quelques idées reçues : ce marquage CE est obligatoire, mais il n'est nullement un gage de qualité. Tous les produits vendus doivent l'arborer, même ceux dont le concepteur aurait mieux fait de choisir un autre métier. Il n'est pas

non plus une marque d'origine : un produit conçu aux États-Unis et fabriqué en Asie devra être marqué CE pour être vendu en Europe. Enfin, le marquage CE n'est pas un gage de validation du produit par un tiers : comme je l'expliquerai, le fabricant d'un produit peut très bien procéder lui-même à son marquage CE.

Que signifie ce logo ? Simplement que le fabricant du produit garantit à l'acheteur que ce produit est conforme aux « exigences essentielles des directives européennes applicables ». Relisez cette phrase : elle dit que le marquage CE est un engagement du fabricant, sous sa propre responsabilité. Point.

Avant de découvrir ces exigences essentielles, éclairons un point important : vous pourriez décider d'arrêter de lire cet article en vous disant que vous n'êtes pas concerné parce que votre produit ne sera pas vendu. Que nenni. Dans les textes, il n'est pas question de « produit vendu », mais de produit « mis sur le marché ou mis en service », et c'est beaucoup plus large ! Votre prototype ne sera pas soumis au marquage CE tant qu'il reste un prototype sur votre établi. Par contre le marquage CE devient en toute rigueur applicable dès que vous donnez ce prototype à Tonton pour qu'il l'installe chez lui, car il y sera « mis en service ». De même un produit loué à ses utilisateurs est concerné, car il est « mis sur le marché » même si ce n'est pas sous la forme d'une vente. Il règne un flou autour de la notion de prototype, mais mieux vaut être prévenu...

La notion de « fabricant » mérite aussi un petit commentaire : si le produit n'est pas physiquement conçu et/ou fabriqué en Europe, l'importateur ou le distributeur peuvent, aux yeux de la loi, être considérés comme fabricant. C'est en particulier le cas si le produit n'a pas été marqué CE par le fabricant proprement dit. En tout état de cause, c'est celui qui est en Europe qui est responsable... De même, en général, la responsabilité incombe à celui dont le nom est marqué sur le produit (le « vendeur » mettant le produit sur le marché), même s'il a sous-traité la conception à X et la fabrication à Y.

Exigences essentielles

Que sont donc ces exigences essentielles dont le marquage CE atteste la conformité ? Ce sont des exigences très générales, définies dans des directives européennes (sortes de



Figure 1 : Le bien connu logo CE

Quand le législateur ouvre son parapluie, le justiciable a les pieds mouillés

lois à l'échelle communautaire). En pratique, le marquage CE implique le respect de trois directives « techniques » qui concernent la quasi-totalité des appareils électroniques. Elles traitent respectivement de la sécurité électrique (directive dite « basse tension »), de la compatibilité électromagnétique (directive CEM) et de l'utilisation le cas échéant des ondes radio (directive RTTE).

D'autres directives liées à l'environnement sont également applicables (limitation des substances dangereuses comme le plomb, gestion des déchets, écoconception, etc.), mais je n'en parlerai pas dans cet article. Enfin, certaines catégories de produits impliquent le respect de directives européennes spécifiques : machines tournantes, jouets, appareils médicaux, etc. La **figure 2** vous donne un résumé succinct.

Pour être concret, voici un petit extrait de la directive 2004/108/EC concernant la CEM :

Les équipements doivent être conçus et fabriqués, conformément à l'état de la technique, de façon à garantir :

a) que les perturbations électromagnétiques produites ne dépassent pas le niveau au-delà duquel des équipements hertziens et de télécommunications ou d'autres équipements ne peuvent pas fonctionner comme prévu;

b) qu'ils possèdent un niveau d'immunité aux perturbations électromagnétiques auxquelles il faut s'attendre dans le cadre de l'utilisation prévue qui leur permette de fonctionner sans dégradation inacceptable de ladite utilisation.

C'est vague, hein ? Vous voulez un autre exemple ? Voici un extrait de la directive basse tension 2006/95/EC :

Des mesures d'ordre technique sont prévues afin que (...) les personnes et les animaux domestiques soient protégés de façon adéquate contre les dangers de blessures ou autres dommages qui peuvent être causés par des contacts directs ou indirects;

Ces textes, librement disponibles sur le web, ne disent donc absolument pas comment concevoir un produit conforme, c'est la responsabilité du fabricant ! Enfonçons le clou : Le marquage CE n'implique pas le respect de normes précises, mais il n'en reste pas moins qu'il faut démontrer qu'on respecte ces exigences essentielles, et c'est compliqué, car ces exigences sont floues...

Normes harmonisées

Heureusement, personne n'a en général à s'occuper directement des exigences essentielles. En effet, des organismes de normalisation ont créé des documents appelés « normes

Directives CE applicable à une large gamme de produits électroniques	
2006/95/EC	Directive «basse tension» (sécurité)
2004/108/EC	Compatibilité électromagnétique (CEM)
1999/5/EC	Équipement radiofréquence et de télécommunication (R&TTE)
Directives CE applicables à certaines classes de produits spécifiques	
2009/48/EC	Jouets
2006/42/EC	Machines
93/42/EEC	Dispositifs médicaux
204/22/EC	Instruments de mesure
94/9/EC	Appareillage utilisable en atmosphère explosive (Atex)
95/16/EC	Ascenseurs
Etc	
Directives applicables à quasiment tous les produits, mais non directement liées au marquage CE	
2002/95/EC	RoHS – Limitations d'emploi de substances dangereuses dans l'équipement électrique et électronique
2002/96/EC	DEEE – Déchets d'équipement électrique et électronique
2009/125/EC	EuP – Exigences en matière d'écoconception pour les produits liés à l'énergie

Figure 2 : Voici les principales directives européennes. Les trois premières sont applicables à quasiment tout produit électronique

harmonisées ». Ces normes décrivent, de manière très précise, des plans de test, des limites à respecter, etc. La règle est la suivante : S'il existe des normes harmonisées applicables à votre produit, par exemple pour la CEM, et si celui-ci respecte ces normes, alors il y a présomption de conformité aux exigences essentielles de la directive CEM.

Il vaut mieux relire la phrase précédente une seconde fois. Rien n'oblige à respecter une norme harmonisée, seul le respect des exigences essentielles est obligatoire. Par contre si on ne s'appuie pas sur une norme harmonisée, alors le travail à faire est en pratique beaucoup plus compliqué, car il faut démontrer une conformité à des exigences très générales. De même les rédacteurs des normes ont ouvert leur parapluie et ne garantissent pas que certains produits pourraient ne pas respecter les exigences essentielles même s'ils respectent les normes, mais je n'ai jamais rencontré ce cas...

Maintenant la mauvaise nouvelle : C'est au fabricant d'un produit de rechercher les normes harmonisées applicables pour un produit donné, or il y en a des milliers [3]. Prenez par exemple les normes harmonisées associées à la sécurité électrique, vous en trouverez un peu plus de 300 (**figure 3**). Comment choisir la bonne ? Pour un projet professionnel, le plus simple et le plus sage est de demander l'avis d'un expert. Et idéalement, dès le démarrage du projet, car rien n'est plus pénible que de ne s'apercevoir qu'une norme est applicable que quelques jours avant le lancement commercial d'un produit ! Une autre solution moins sûre, mais moins chère, est de

regarder quelles normes ont été appliquées pour des produits similaires au vôtre... mais en vérifiant quand même que ces normes sont toujours d'actualité.

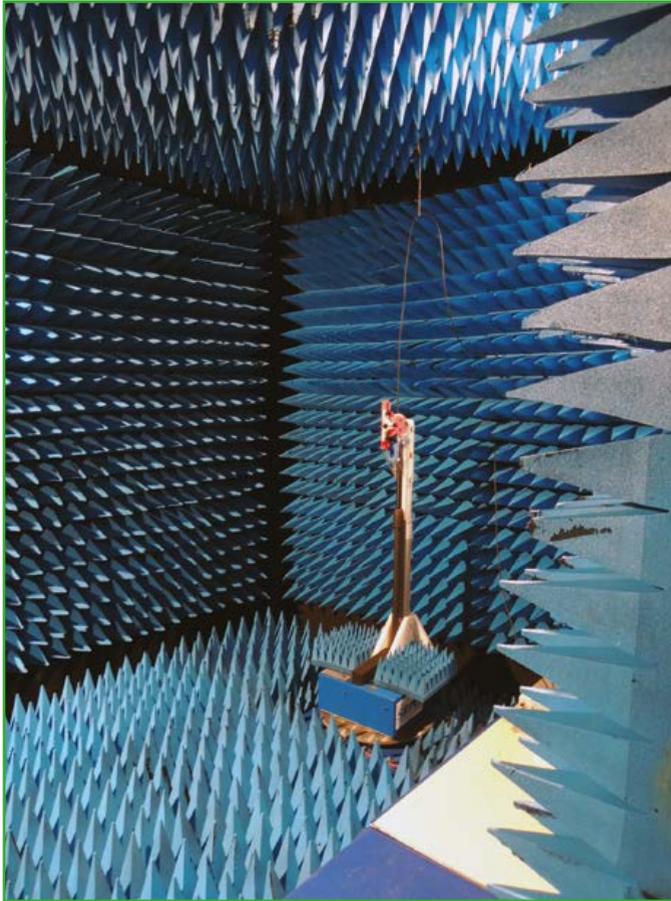
Sécurité électrique

La directive « basse tension » s'applique à tout appareil raccordé à (ou sur lequel règnent) des tensions de 50 à 1000 V AC ou de 75 à 1500 V DC. C'est le sens de « basse tension », par opposition aux systèmes haute tension au sens des électriciens... Cette directive est donc applicable en particulier à tout produit alimenté par le secteur. Une nuance amusante : c'est en toute rigueur aussi le cas pour tout produit radio, même s'il s'agit d'un émetteur de 10 mW alimenté par pile.

Prenons un périphérique d'ordinateur, disons une imprimante. Dans la liste des normes harmonisées associées à la directive basse tension, vous trouverez que la norme EN60950-1 est applicable aux « Matériels de traitement de l'information ». Ça semble bien adapté... Vous aurez alors à acheter la norme en question, par exemple auprès de l'AFNOR en France. Eh oui, les directives européennes sont librement accessibles sur le web, mais les normes se payent. Comptez 50 à 100 € par norme... Vous devrez alors lire l'ensemble des exigences de sécurité à respecter pour concevoir cette imprimante : distances d'isolation, simple ou double isolation, mises à la terre, tenue des câbles, etc. Mais cette norme ne s'arrête pas aux risques électriques : il faudra aussi vérifier qu'il n'y a pas de points accessibles trop chauds, de risques de se coincer un doigt, etc.

EN 41003:1998	Particular safety requirements for equipment to be connected to telecommunication networks
EN 60065:2002	Audio, video and similar electronic apparatus - Safety requirements
EN 60204-1:2006	Safety of machinery - Electrical equipment of machines -- Part 1: General requirements
EN 60215:1989	Safety requirements for radio transmitting equipment
EN 60335-2-5:2003	Household and similar electrical appliances - Safety -- Part 2-5: Particular requirements for dishwashers
EN 60335-2-59:2003	Household and similar electrical appliances - Safety -- Part 2-59: Particular requirements for insect killers
EN 60598-2-7:1989	Luminaires -- Part 2: Particular requirements -- Section 7: Portable luminaires for garden use
EN 60730-2-14:1997	Automatic electrical controls for household and similar use -- Part 2-14: Particular requirements for electric actuators
EN 60825-1:2007	Safety of laser products -- Part 1: Equipment classification and requirements
EN 60950-1:2006	Information technology equipment - Safety -- Part 1: General requirements
EN 60950-21:2003	Information technology equipment - Safety -- Part 21: Remote power feeding
EN 61010-1:2001	Safety requirements for electrical equipment for measurement, control, and laboratory use -- Part 1: General requirements
EN 61558-1:2005	Safety of power transformers, power supplies, reactors and similar products -- Part 1: General requirements and tests
Etc.	

Figure 3 : Quelques exemples de normes harmonisées associées à la directive de sécurité « basse tension ». Il y en a plus de 300...



En général la vérification de conformité aux normes de sécurité électrique consiste à cocher une check-list, mais certains essais sont à faire. C'est en particulier le cas si l'alimentation secteur n'est pas un produit acheté sur étagère et déjà conforme : absences de risque en cas de court-circuit d'un condensateur par exemple, ou tenue au feu du boîtier si celui-ci n'est pas déjà qualifié. La liste peut être longue...

Compatibilité électromagnétique

La sécurité électrique n'est qu'une petite partie des exigences du marquage CE. L'aspect le plus emblématique est la CEM. Les normes CEM couvrent en fait cinq aspects plus ou moins indépendants :

- Les émissions rayonnées : Le produit ne doit pas émettre d'ondes radio trop fortes, qui pourraient parasiter d'autres appareils. Ce test nécessite de placer le produit dans une chambre anéchoïque et de le faire fonctionner en l'orientant dans toutes les directions face à une antenne de mesure. Si par exemple la fréquence de l'oscillateur du microcontrôleur rayonne à un niveau trop fort, le produit sera déclaré non conforme.
- Les émissions conduites : De la même manière, il faut vérifier que les différents câbles d'entrée/sortie du pro-

duit ne conduisent pas trop de parasites. En général, si ce test échoue, il y a fort à parier que le test d'émission rayonnée échouera aussi, raison pour laquelle il est judicieux de le faire en premier...

- L'immunité rayonnée : Là on vérifie que le produit se comporte correctement lorsqu'il est mis en présence d'un champ électromagnétique fort (simulant par exemple l'approche d'un téléphone portable). Plusieurs classes de produits sont définies, mais il doit au moins fonctionner normalement dès que le champ perturbateur est coupé... et sans nécessiter d'enlever et de remettre la pile !
- L'immunité conduite : De la même manière, on vérifie que le produit résiste à des perturbations injectées sur ses câbles entrés/sorties : Applications de signaux RF forts ou de simulations de surtensions.
- L'immunité aux décharges électrostatiques (ESD) : Encore un test drôle. On applique des décharges de 4 à 8 kV à tous les points accessibles du produit. Si par exemple un radiateur ou une vis de fixation n'est pas mise à la masse avec soin et s'approche d'un composant électronique sensible, alors le test est... euh, destructeur. On garde donc en général ce test pour la fin, mais il n'en reste pas moins qu'il doit réussir !



Comme pour la sécurité électrique, il existe des centaines de normes CEM et il n'est pas simple de savoir laquelle est applicable pour un produit donné. Dans l'exemple de l'imprimante que j'ai pris plus haut, les bons choix seraient probablement les normes EN55022 associées aux normes d'immunité de la famille EN61000. Pas simple, hein ?

Liens

- [1] European Council Directives : <http://eur-lex.europa.eu>
- [2] http://ec.europa.eu/enterprise/policies/single-market-goods/cemarking/faq/index_en.htm
- [3] www.newapproach.org/
- [4] http://en.wikipedia.org/wiki/CE_mark
- [5] Harmonized standard database : <http://ec.europa.eu/enterprise/policies/european-standards/documents/harmonised-standards-legislation/list-references/#ch2>
- [6] "EMC for product Designers", Tim Williams, Elsevier/Newnes, ISBN 0-7506-8170-5

Produits radio ?

Ce n'est pas fini. Dans le cas des produits radio, une autre classe de normes est applicable : celles relatives à la directive RTTE pour le « bon usage » du spectre RF. Les normes applicables sont bien connues des développeurs de produits radio et portent par exemple les doux noms de EN300220 (émetteurs/récepteurs en dessous 1 GHz), EN300328 (2,4 GHz type Wifi ou Bluetooth), etc. Les tests à réaliser concernent surtout la partie émetteur : le produit doit émettre à la bonne fréquence et ni trop fort ni trop longtemps, ne doit pas polluer les canaux d'à côté, ne doit pas émettre trop de parasites, etc. C'est moins connu, mais un récepteur doit aussi être testé : il doit par exemple suffisamment bien résister à des signaux parasites dans les canaux voisins. Vous pourriez demander pourquoi, mais le principe est logique : un mauvais récepteur impliquerait une mauvaise utilisation du spectre radio, car l'émetteur associé devrait transmettre à un niveau plus fort que nécessaire...

Et la paperasse ?

Quelques mots pour finir sur le volet administratif. Vous seriez le fabricant d'un produit électronique que vous souhaitez mettre sur le marché en Europe. Vous avez identifié, seul ou avec un peu d'aide, les normes applicables à ce produit. Pouvez-vous faire les essais vous-même ? La réponse est oui, sauf pour certains produits « à risque » (produits médicaux, atmosphères explosives, etc.). Ça ne veut pas dire que ces essais peuvent être bâclés : pour les mener, il vous faudra des appareils de mesure spécifiques, bien calibrés, et une bonne dose de savoir-faire. C'est la raison pour laquelle les laboratoires d'essai spécialisés sont florissants. Rien ne vous interdit non plus de faire certains tests vous-même et d'en sous-traiter d'autres. Vous pouvez aussi ne pas réaliser certains tests si vous estimez que le risque de non-conformité est nul. Mais il faudra le justifier et documenter cette justification dans le dossier technique.

Une fois les tests réalisés, vous devrez constituer le dossier technique du produit. Il comprendra une description précise du produit (schémas, nomenclatures, circuits imprimés, plans

mécaniques, mais aussi logiciels embarqués s'il y en a) et les résultats des tests. Ce dossier technique doit être conservé au moins 10 ans après la date de vente du dernier exemplaire du produit : il peut être exigé en cas de contrôle.

Enfin, vous devrez rédiger un certificat de conformité CE (une simple page signée) par lequel vous certifiez que le produit est conforme et vous listez les normes applicables. Ce certificat doit être à la disposition de chaque client (en général dans le manuel de l'utilisateur, dans la partie en tout petits caractères...).

Est-ce la dernière étape ? Bien sûr que non... Un produit n'est que rarement figé. Il évoluera, certains composants deviendront obsolètes, etc. Vous devrez prouver qu'il reste conforme aux normes du marquage CE. Au minimum, vous devrez analyser l'impact des changements, faire au besoin quelques tests de non-régression, et surtout le documenter dans le dossier technique. Ah, les normes aussi peuvent évoluer, et vous devrez vérifier que le produit reste conforme aux dernières versions des normes en vigueur tant que le produit est commercialisé ! Cela peut impliquer de nouveaux tests, mais aussi peut être des changements dans le schéma...

Pour conclure

Trouvez-vous que les normes semblent faites pour embêter les électroniciens et enrichir les labos de test ? Ce n'est pas complètement faux, mais ces normes ont aussi apporté aux consommateurs des produits de bien meilleure qualité. Prenons l'exemple de la CEM. Allumez un PC récent à côté d'un récepteur radio, tout se passe (à peu près) bien. Faites de même avec l'un des premiers micro-ordinateurs, un Apple II par exemple. Vous serez surpris. Entretemps les fréquences d'horloge ont pourtant été multipliées par 1000, mais les produits actuels perturbent bien moins que leurs ancêtres. Grâce aux directives et aux normes !

J'espère que cet article ne vous a pas découragé. Les prochains aborderont des sujets de nouveau plus techniques !

(140511)

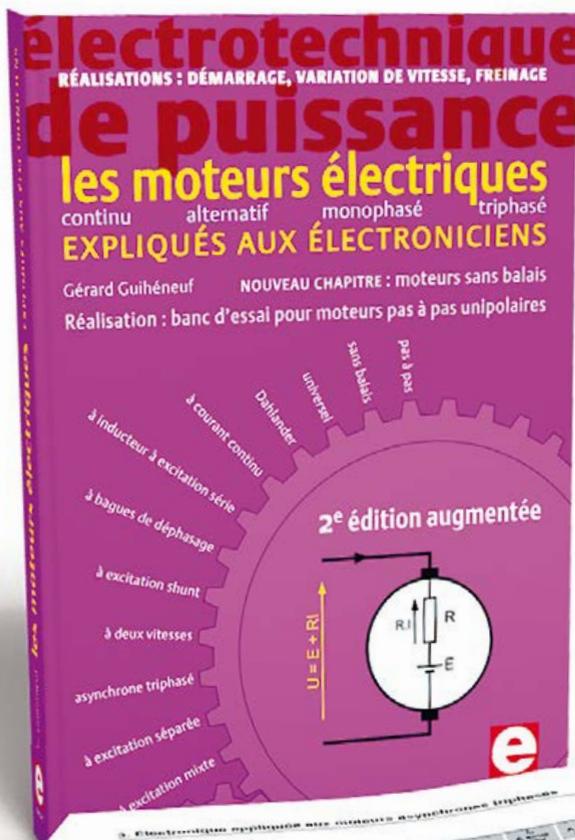
Nouvelle édition augmentée

100 pages supplémentaires !

électronique de puissance

les moteurs électriques

expliqués aux électroniciens



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

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

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

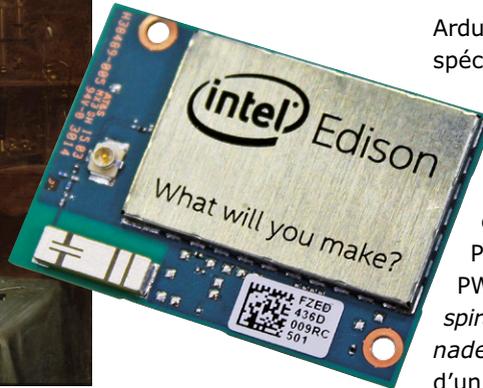
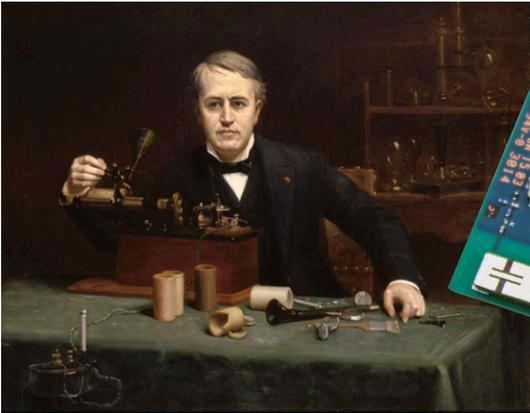
nouveau chapitre + 100 pages

ISBN 978-2-86661-194-1
432 pages | 46,50 €



www.elektor.fr/moteurs

Edison d'Intel : aimez-vous le boudin de pingouin ?



Clemens Valens
(Elektor.Labs)

Fin septembre 2014, moins d'un an après avoir présenté la carte Galileo (maintenant en révision 2), Intel lançait son petit module Edison. Présenté comme une plateforme bon marché, prête à l'emploi et à usage général, ce module vise un large éventail de concepteurs. Il entend séduire des esprits pionniers à l'œuvre dans leur garage façon Steve & Steve, aussi bien que des fabricants d'électronique grand public et des entreprises travaillant pour l'Internet des Objets ou l'électronique portable.

Il suffit de jeter un œil aux spécifications (**encadré**) pour se rendre compte de l'impressionnant jeu de fonctions embarqué d'Edison. Ses entrailles évoquent celles d'un téléphone tactile (son SoC serait celui d'un *smartphone*), et il ne serait sans doute pas difficile d'en construire un à partir de ce module.

Yocto Linux

Edison est livré avec Yocto installé. Le projet Yocto fournit un ensemble de ressources pour créer et personnaliser facilement une distribution Linux pour l'embarqué. L'utilisateur type d'Edison n'en aura pas besoin, mais les utilisateurs avancés pourront profiter de la flexibilité de Yocto pour personnaliser leur système Linux.

Même Edison utilise Arduino

Le module Edison n'est guère exploitable sans carte de liaison (*breakout board*, BoB). Intel propose donc deux kits matériels : le kit Edison pour

Arduino et la carte Breakout Board Edison. Leurs spécifications sont fournies séparément. J'utiliserai ici le kit pour Arduino.

La BoB Arduino est compatible à 99 % avec le brochage Arduino 1.0. La différence de 1 % s'explique par l'absence de deux canaux MLI (PWM) : quatre au lieu de six. Pour y pallier, la BoB est équipée d'un *swizzler* PWM (d'après le nom d'un *hotdog découpé en spirale pour une meilleure absorption de la marinade dans laquelle baigne la saucisse*), c.-à-d. d'un bloc de cavaliers permettant de router les 4 signaux MLI d'Edison vers l'une ou l'autre des 6 broches MLI d'Arduino. Ce système d'aiguillage semble une bonne idée si l'on oublie les cartes d'extension ayant besoin de plus de 4 signaux de MLI matérielles (cela dit vous pouvez bien sûr faire de la MLI logicielle), mais il y a un hic : les sorties MLI d'Edison ne peuvent pas être reliées à n'importe quelle broche Arduino, autrement dit si vous routez une de ces sorties vers une autre broche Arduino que celle prévue, cette dernière n'est pas connectée et devient inutilisable. N'utilisez donc le *swizzler* qu'en cas de besoin.

D'autres subtilités de ce genre peuvent mener l'innocent droit aux ennuis ; si vous envisagez de travailler sérieusement avec BoB, je vous conseille donc de lire attentivement toutes les pages du guide *Intel Edison Kit for Arduino Hardware Guide*.

Démarrage

La première étape passe par une séance de téléchargements : l'EDI Arduino pour Edison (et Galileo), des pilotes de FTDI pour le port série USB, des pilotes d'Intel pour la carte Edison, ainsi que l'image de Yocto. À l'exception des pilotes FTDI, tous ces fichiers peuvent être téléchargés depuis la page Edison du site d'Intel. Dénichez deux câbles micro USB en attendant la fin des téléchargements. Si comme moi vous n'en possédez qu'un seul, vous pouvez utiliser (au prix de nombreux échanges de câbles en ce qui me concerne) un câble micro USB et un adaptateur de 12 V. L'assemblage du kit ne devrait poser aucun problème à l'Elekteur que vous êtes, mais

en cas de difficultés vous pouvez suivre les instructions du site d'Intel [1].

Prêt ? C'est parti :

1. Installez tous les pilotes sur votre ordinateur avec les droits d'administrateur.
2. Copiez l'image de Yocto sur le module Edison ; pour cela reliez un câble micro USB à J16, le connecteur micro USB situé à côté du commutateur SW1. Après avoir relié le câble au PC, le module est reconnu comme clé USB. Assurez-vous que ce périphérique est vide (supprimez éventuellement tous les fichiers présents) puis décompressez-y l'image Yocto.
3. Flashez le module Edison avec l'image Yocto. Pour cela branchez le câble micro USB dans le connecteur J3 placé à l'angle de la carte. Ouvrez le gestionnaire de périphériques de Windows et notez le numéro du port série créé ([1] instructions relatives à Linux et Mac). Ensuite, depuis un émulateur de terminal comme PuTTY ou Tera Term, ouvrez ce port avec pour paramètre `115200n81`. Appuyez deux fois sur Entrée pour faire apparaître l'invite de connexion (*edison login:*). Entrez *root*, validez, puis entrez *reboot ota* et validez. Le processus est terminé lorsque réapparaît l'invite (après environ 2 min).
4. Décompressez l'archive de l'EDI Arduino et lancez *arduino.exe*.
5. Reconnectez le câble micro USB au connecteur J16. En plus du périphérique de stockage USB est créé un port COM virtuel Intel Edison (COM33 pour moi), le port série que vous utiliserez pour programmer Edison.
6. Depuis l'EDI Arduino, cliquez sur *Outils*, puis sur *Type de carte*, et sélectionnez le module Edison.
7. Dans le menu *Outils*, cliquez sur *Port série* puis sélectionnez le port qui correspond au port COM virtuel Intel Edison (COM33 pour moi).
8. Dans le sous-menu *Exemples* du menu *Fichier* choisissez *01.Basics*, puis *Blink* pour charger le croquis *Hello World*.
9. Cliquez sur le bouton *Téléverser* et patientez jusqu'à ce que clignote la LED DS2 (placée au milieu de la carte, près de J4).

Ajouter le Wi-Fi

Vous avez maintenant entre les mains l'équivalent d'une carte Arduino complexe et presque entièrement compatible avec elle. Connectons Edison à un réseau Wi-Fi :

Spécifications du module Edison

- SoC Intel de 22 mm comprenant un processeur Atom d'Intel cadencé à 500 MHz, à double cœur et double fil d'exécution, et un µC Quark d'Intel à 100 MHz et 32 bits
- 40 ports d'E/S à usage général (GPIO), configurable p. ex. comme interface de carte SD, 2x UART, 2x I²C, SPI, I²S, 4x PWM, USB 2.0 OTG
- 1 Go de RAM (LPDDR3)
- 4 Go de mémoire flash (eMMC)
- Wi-Fi double bande
- Bluetooth 4.0
- Alimentation : 3,3 V à 4,5 V
- Consommation en mode veille : 13 mW sans radio ; 21,5 mW avec Bluetooth 4.0 ; 35 mW avec Wi-Fi
- 35,5 × 25,0 × 3,9 mm
- 0 °C à 40 °C
- Yocto Linux v1.6

Spécifications de la carte Edison pour Arduino

- Linux, Wi-Fi, Bluetooth et croquis Arduino pris en charge
- E/S compatibles avec Arduino (exception : 4 PWM au lieu de 6 PWM)
- 20 broches d'E/S numériques, y compris 4 (pas 6) broches comme sorties MLI/PWM
- 6 entrées analogiques
- UART (Rx/Tx)
- I²C
- Embase ICSP à 6 broches (SPI)
- Connecteur micro USB OU (via un commutateur mécanique) connecteur standard USB de type A pour hôte USB
- Port micro USB (relié à l'UART)
- Connecteur pour carte SD
- Prise alimentation CC (7 à 15 V CC en entrée)

Spécifications de la carte Breakout Edison

- Légèrement plus grande que le module Edison
- Conçue pour exposer les E/S natives de 1,8 V du module Edison
- Matrice d'E/S constituée de points de soudure traversants au pas de 0,1 pouce
- USB OTG avec connecteur micro USB de type AB
- Interrupteur alimentation USB OTG
- Chargeur de batterie
- Pont USB vers UART avec connecteur USB de type B
- Prise alimentation CC (7 à 15 V CC en entrée)

1. Reliez le câble micro USB au connecteur J3 et ouvrez le port série (étape 3 précédente) depuis un émulateur de terminal (Tera Term ou PuTTY). Entrez `115200n81` comme paramètre de port, appuyez deux fois sur Entrée pour appeler l'invite de connexion, tapez *root* et validez.

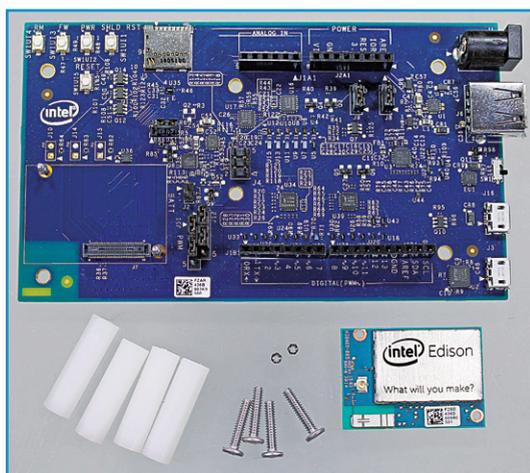


Figure 1.
Le kit Edison pour Arduino.
Et vous, qu'en ferez-vous ? Partagez votre enthousiasme avec la rédaction et les lecteurs d'Elektor.

2. Entrez `configure_edison --setup` (notez les deux tirets simples) et validez pour vous en remettre aux bons offices de l'assistant de configuration. Pour sauter une étape, appuyez sur Entrée.
3. Un message final devrait vous annoncer que vous pouvez vous connecter au module à l'adresse IP 192.168.x.x dans un navigateur. Vous devriez tomber sur une page d'accueil. Rien chez moi. Mon routeur a bien listé le module, mais sans nom d'hôte attribué ☹.

Edison cherche tête d'ampoule

D'après mon routeur, mon Edison n'a donc aucun nom d'hôte. La commande `hostname -fqfd` a retourné l'erreur `Host name lookup failure`, et

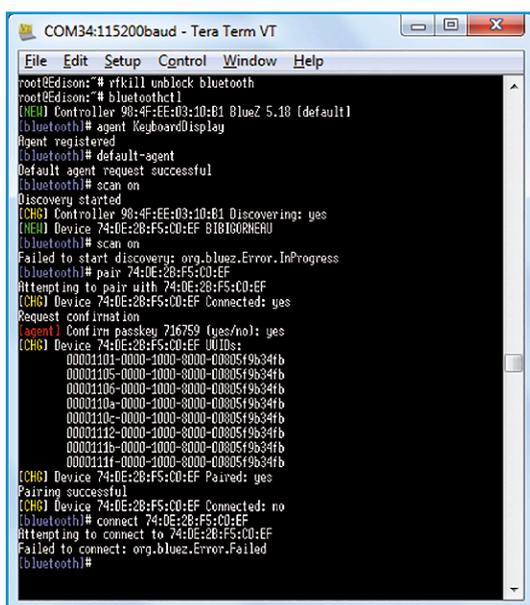


Figure 2.
Activation du Bluetooth
sur le module Edison. Tout
s'est bien déroulé sauf... la
connexion.

un `ping` vers Google a renvoyé `bad address 'www.google.com'`. J'ai bien trouvé sur Internet des utilisateurs de Raspberry Pi désespérés par le même problème, mais aucune des propositions proposées ne m'a aidé, soit qu'elles semblaient sans effet, soit que la commande suggérée n'était pas reconnue par Yocto. Si vous avez une solution, merci d'éclairer ma lanterne. En attendant, je continuerai à maudire le pingouin.

Les bienheureux qui auront connecté leur navigateur à leur Edison peuvent lancer les croquis d'exemple Arduino pour le Wi-Fi. Ceux-ci se compilent sans problème si l'on paramètre correctement les chaînes `ssid` et `pass` dans les croquis. Le module se comporte alors comme une carte Arduino dotée d'un *shield* Wi-Fi.

Faute d'exemple Arduino pour le Bluetooth, je me suis rabattu sur un tutoriel du site Intel expliquant comment activer Bluetooth. Ça marche, mais je n'ai pas réussi à faire dialoguer module et PC et ce tuto ne dit pas comment combiner les Bluetooth Arduino et Edison. Pour le moment Bluetooth semble donc réservé aux programmeurs Linux.

Conclusion

Edison est réussi. Il offre de la puissance en veux-tu en voilà, mais comment libérer ce potentiel ? Sans Wi-Fi il n'y a pas grand chose à en tirer. Sans BoB non plus. Il manque une documentation et des exemples de code faciles à comprendre pour les débutants. Edison conviendra surtout à ceux qui sont nourris au sang de pingouin et savent comment combiner toutes sortes de paquets Linux en méta-applications ; pour ceux assez intrépides pour modifier la distro Yocto installée par défaut (oui, *distro*, j'ai appris un nouveau mot) et pour ceux qui sont à l'aise avec un éditeur de texte comme *vi*.

Reste ce détail qui me chiffonne : Edison est spécifié pour 40 °C max. Les pingouins aiment le frais, mais pour un module qui se veut utile pour l'embarqué vestimentaire, ce 40 °C a de quoi refroidir, surtout si comme moi on mesure 33 °C dans la poche de son pantalon pour une température extérieure de 21 °C...

(140439-I - version française : Hervé Moreau)

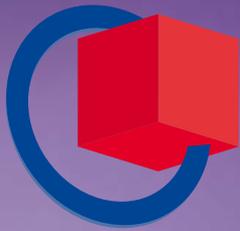
Lien

[1] Edison : www.intel.com/content/www/us/en/do-it-yourself/edison.html

Inscrivez-vous dès maintenant et réservez vos badges !

embedded-world.de

Nuremberg, Allemagne
24 – 26.2.2015



embedded world 2015

Exhibition & Conference

... it's a smarter world

LA rencontre du monde de l'embarqué !

Le plus grand salon mondial des technologies embarquées établit le dialogue entre les acteurs de l'embarqué.

Ne manquez pas cet événement qui met l'accent sur les contacts et le networking international et lance de nouvelles tendances.



Médias partenaires

elektroniknet.de

computer-automation.de

energie-und-technik.de

MEDIZIN-und-elektronik.DE

Markt & Technik
DIE UNABHÄNGIGE WOCHENZEITUNG FÜR ELEKTRONIK

DESIGN & ELEKTRONIK
KNOW-HOW FÜR ENTWICKLER

elektroniknet.de
Elektronik
Fachmedium für Industrielle Anwender und Entwickler

Elektronik automotive
Fachmedium für professionelle Automobilentwicklung

ENERGIE & TECHNIK
Fachmedium für Energieeffizienz

Computer & AUTOMATION
Fachmedium über Automatisierungstechnik

MEDIZIN & elektronik
Fachmedium für Elektronik in der Medizintechnik

Organisateur du salon professionnel

NürnbergMesse GmbH

Tel +49 (0) 9 11.86 06-49 12

visitorservice@nuernbergmesse.de

Organisateur des congrès

WEKA FACHMEDIEN GmbH

Tel +49 (0) 89.2 55 56-13 49

info@embedded-world.eu

NÜRNBERG / MESSE

EveryCircuit : l'électronique virtuelle

Harry Baggen
(Elektor)

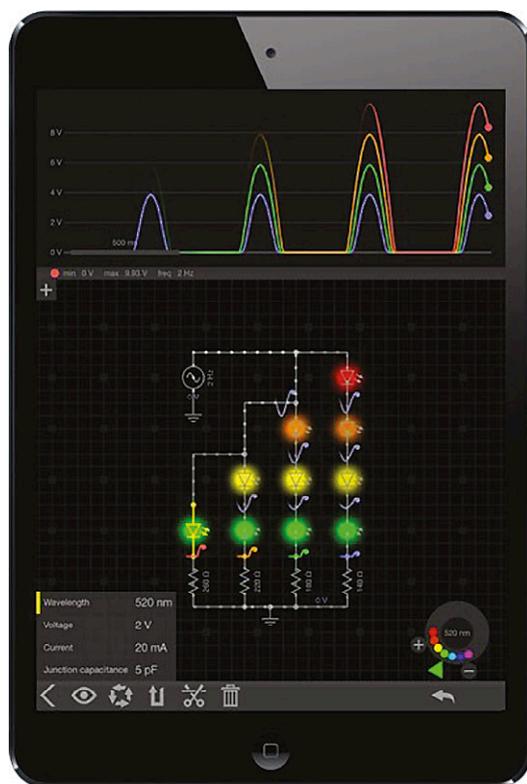


Figure 1.
Simulation avec EveryCircuit
sur tablette iPad.

Mettez le doigt dans l'électronique avec cette belle application de simulation

L'électronique ? Pas de panique !
Avec l'application EveryCircuit tout
devient clair comme de l'eau de roche,
tellement simple que vous pourrez
suivre les électrons sur l'écran ou la ta-
blette. Ce simulateur interactif permet
de tester un circuit, sans fer à souder,
en quelques minutes.

Qui dessine encore ses schémas ou ses circuits imprimés à la main ? Pourquoi se priverait-on des merveilleux outils de conception offerts par l'informatique ! Désormais les téléphones et les tablettes tactiles font partie de l'arsenal de l'électronicien, pas seulement comme calculette ou comme source d'information (fiches techniques). Il existe des programmes de saisie de schéma et de simulation convenables sur tablette. Leur palette s'enrichit rapidement et les progrès sont de plus en plus convaincants.

EveryCircuit est une **app**(lication) de simulation des circuits électroniques. Elle est différente des programmes de simulation classiques, car elle cherche à illustrer simplement le comportement d'un circuit sans trop se soucier de détails techniques. Ainsi, l'écran affiche en temps réel l'évo-

lution de divers signaux (tensions et courants) dans le circuit ; en cours de simulation, on peut modifier les propriétés du circuit. Disponible pour Android et iOS, elle tourne aussi sous Windows et Linux (avec Chrome). Lors d'un banc d'essai rapide, nous avons essayé le programme sur une tablette Android et sur un PC sous Windows, avec le navigateur Chrome. Le fonctionnement de la version pour tablette ou téléphone tactiles est similaire, mais on utilise les doigts au lieu de la souris.

Il existe une version d'évaluation d'EveryCircuit, pour faire connaissance [1], son champ de saisie du schéma est cependant limité à six composants. L'achat d'une licence (pour un an, disponible dans l'e-choppe d'Elektor [2] lève cette limitation et ouvre l'accès à la communauté des utilisateurs

et au stockage en ligne des schémas. Ah ! les nuages, les merveilleux nuages... vos schémas y seront ainsi disponibles pour plusieurs périphériques, tablette et PC p. ex. sans restriction, sauf si vous leur donnez l'attribut « privé ». Autrement, ils sont partagés avec la communauté déjà riche en circuits des utilisateurs d'EveryCircuit.

Le programme au travail

Ce qui frappe au démarrage est le petit nombre d'organes de commande. Ça tombe bien car (pour l'instant) il n'y a pas non plus de manuel. Commencez par de petits pas. Tout en haut, une barre de composants avec les types les plus courants : sources de tension et de courant, résistances, transformateurs, transistors, interrupteurs, portes, et même un temporisateur 555. Il n'existe pas de bibliothèque avec des numéros de type, mais on peut modifier les propriétés de chacun des composants du schéma.

Après un clic sur un élément, celui-ci apparaît dans la zone de dessin où on pourra l'agripper pour le mettre à la bonne place. Établissez ensuite les interconnexions en cliquant sur les extrémités de deux composants. Une fois un composant sélectionné, il apparaît, sur la barre au bas de l'écran, des symboles permettent de le faire pivoter, de le mettre en miroir, de le supprimer

ou d'en adapter les propriétés. Un clic sur la clé plate affiche tous les paramètres ajustables du composant, modifiables à l'aide d'une molette (à droite). Le symbole d'œil permet de choisir la forme d'onde que doit visualiser la simulation. En cas d'action sur un composant, le courant s'affiche ; si l'on touche une connexion, c'est la tension. Si on clique une nouvelle fois le composant sélectionné ou à un endroit vide du champ de saisie de schéma, on verra s'afficher en bas deux triangles, « t » et « f » (ce dernier uniquement dans la version enregistrée). Ils désignent deux options de simulation : « transitoire » et « fréquence ». Un clic sur l'un d'eux lance la simulation. Lors de l'analyse CA (« f »), un diagramme de Bode est affiché, comme le font d'autres programmes de simulation.

Ce qui fait vraiment l'intérêt d'EveryCircuit, est l'analyse transitoire, car elle permet de voir le comportement du circuit et le trajet du courant à travers les composants. Sélectionnez une source de tension, une résistance ou une LED et vous verrez immédiatement l'intensité du courant qui la traverse ou la chute de tension à ses bornes. De nombreux composants réagissent à l'écran comme ils le font dans la réalité. Les LED ne s'allument que si le courant qui les traverse est suffisant ; on peut aussi actionner les interrup-

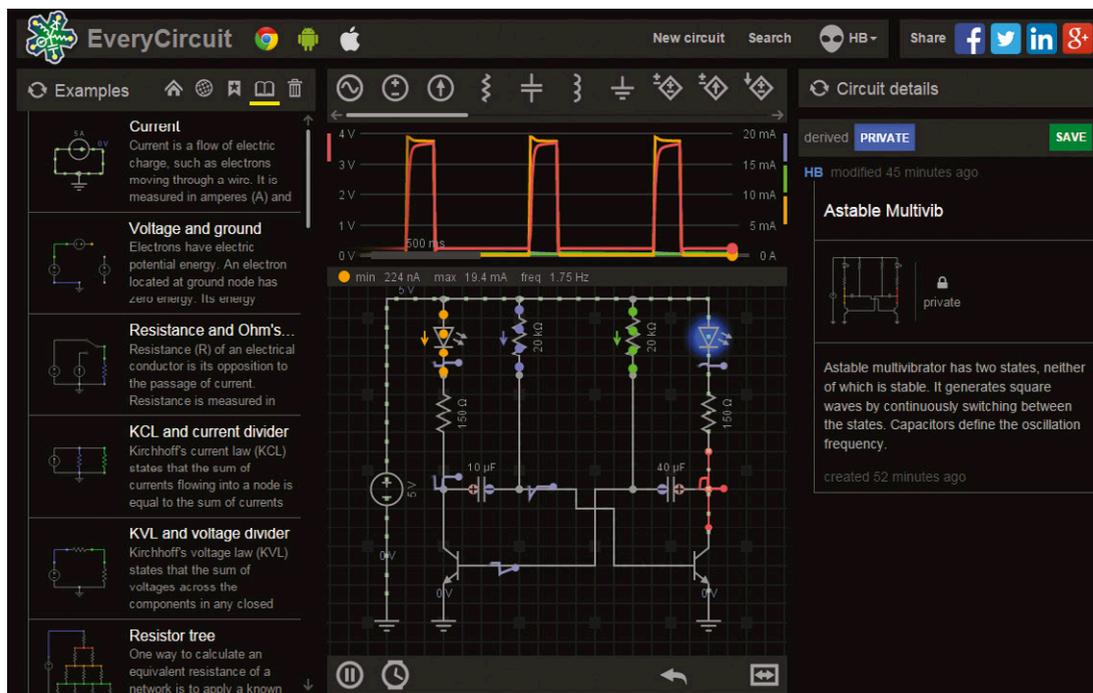


Figure 2. Page d'accueil d'EveryCircuit dans le navigateur Chrome. On trouve à gauche les schémas d'exemple ou d'utilisateurs, au centre on voit la fenêtre du schéma avec les signaux de simulation et à droite le projet en cours avec description.

teurs en cours de simulation. Ces caractéristiques font d'Everycircuit le programme idéal pour les néophytes en électronique. On peut rapidement essayer bien des choses et voir immédiatement ce qui se passe... ou pas ! Un minimum de connaissances du fonctionnement des composants électroniques est souhaitable, car le programme lui-même est muet à ce sujet.

Outre les schémas partagés par la communauté, il y a une liste d'exemples pour se faire la main ou découvrir un peu plus l'électronique.

Voir les électrons

EveryCircuit est très utile pour la simulation de circuits, surtout les petits. Les manipulations sont simples une fois qu'on s'y est habitué. Grâce à l'affichage en temps réel des tensions et des courants, ainsi qu'à la possibilité de modifier les composants du schéma en cours de simulation, un débutant saisira instantanément le fonctionnement d'un circuit comme s'il y voyait circuler les électrons. Pour qui découvre l'électronique, c'est le moyen idéal pour acquérir de l'expérience dans la conception de circuits électroniques avant même de les réaliser en pratique. En revanche, pour ce qui concerne les réglages des paramètres des semi-conducteurs, c'est une autre paire de manches pour le néophyte.

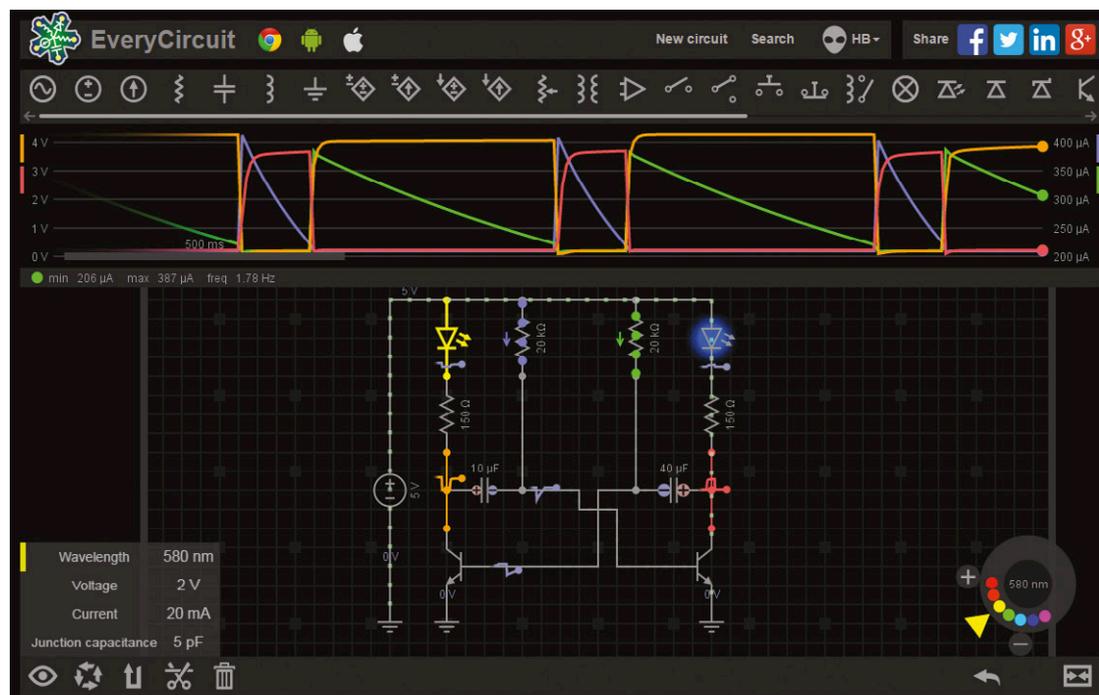
Ce programme ne prétend pas se substituer aux grands programmes de simulation, mais c'est un régal de s'en servir. Même pour un électronicien expérimenté qui, à l'occasion, testera tel ou tel petit schéma avec EveryCircuit, juste pour voir ce qui se passe, sans se soucier de détails techniques.

(140334 - version française : Guy Raedersdorf)

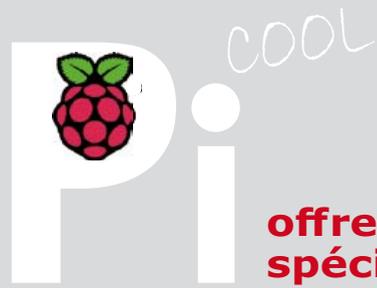
Liens

- [1] www.everycircuit.com
(ou Play Store « everycircuit free »)
- [2] La licence annuelle pour EveryCircuit dans l'e-Choppe Elektor ne coûte que 7 €, ou 6,30 € pour les membres abonnés d'Elektor : www.elektor.fr/everycircuit-app

Figure 3.
Le bouton d'expansion (↔) fait disparaître les colonnes de gauche et de droite : le schéma devient plus lisible. À gauche, une petite fenêtre s'ouvre, elle donne les propriétés d'une LED ; à droite, la molette permet de modifier divers paramètres.



nouveau livre www.elektor.fr/rpi



Raspberry Pi

45 applications pour l'électronicien

utiles

offre spéciale

l'alliance de la programmation et de l'électronique

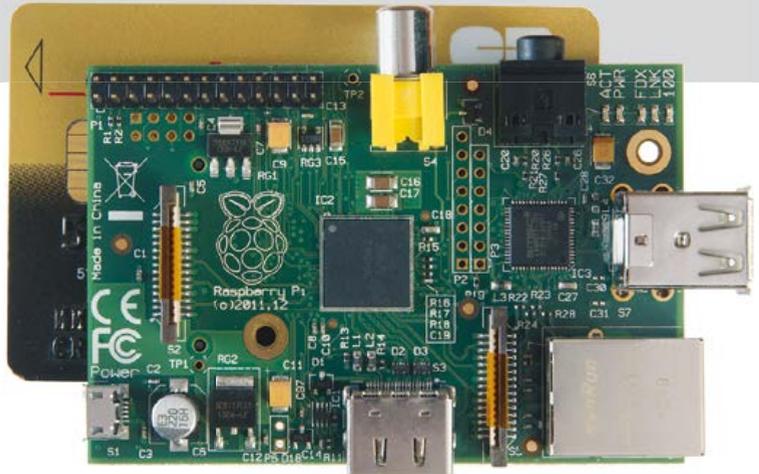


photo Jean-Paul Brodier



Avec le *Raspberry Pi*, pour quelques dizaines d'euros, vous disposez d'un ordinateur complet auquel vous pouvez connecter simplement toutes sortes de montages électroniques. Ce livre montre un des points forts – sinon la raison d'être – du *Raspberry Pi* : l'alliance de la programmation et de l'électronique. Le logiciel est gratuit. Après une introduction brève au système *Linux* et à la programmation en *Bash*, *Python* et *Javascript*, l'accent est mis sur *Python*, mais sans approfondir. L'auteur expose seulement ce qui vous est nécessaire pour comprendre les 45 projets captivants et les adapter à vos besoins. Du clignotant alternatif à la régulation de température, en passant par la commande de moteurs électriques, le traitement de

signaux analogiques et un luxmètre ; mais aussi des projets compliqués comme une régulation de vitesse de moteur, un serveur avec CGI, des applications client-serveur et des programmes *Xwindow*. Ce livre est un manuel de T.P. avec explications claires, schémas et photos de l'implantation sur une plaque d'essai. C'est aussi un cours : les solutions choisies sont expliquées. En réalisant les montages vous-même, vous apprendrez beaucoup sur le *Raspberry Pi*, *Python* et les composants utilisés ; vous pouvez aussi modifier ou combiner les projets et les étendre selon vos souhaits. Ce livre est enfin un ouvrage de référence. Même quand vous aurez réalisé tous les projets, il gardera longtemps une place à côté de votre *Raspberry Pi*.

Logiciel disponible en téléchargement gratuit ou installé sur une carte SD vendue séparément



ISBN 978-2-86661-196-5 | 296 pages - 37,50 €

info et commande : www.elektor.fr/rpi

GestIC & 3D Touchpad : manip. n°2

bougez pour gagner au 2048 sur RPi

Thomas Lindner & Hung Nguyen
(Microchip GestIC® Team, Allemagne)



Cette fois-ci, c'est à un Raspberry Pi que nous allons connecter la puce de contrôleur *MGC3130 touch & gesture* pour y jouer au 2048 (exactement 2¹¹), un jeu qui, dit-on, pousse à l'addiction. Si vous débarquez dans la série, sachez qu'il y a trois choses à retenir :

1. c'est comme un cahier de Travaux Pratiques, notes assez disparates et succinctes ;
2. le matériel utilisé, on ne le trouve que dans l'e-choppe d'Elektor [1] ;
3. il vaut mieux lire l'article précédent [2] et le premier épisode [3].

Recette

Commencez par vérifier les connexions du matériel [3].

Le **tableau 1** récapitule la cartographie des broches d'E/S requises.

Assurez vous d'avoir :

1. le dernier Raspbian, c-à-d Raspbian Debian Wheezy, version : sept. 2014, date de sortie : 09/09/2014
2. le programme Python : 2048_with_Hillstar_Gesture_Port.py (avec la correction du bogue de la cartographie du port) [4].
3. MGC3130 FW + paramétrage : Hillstar Gesture Port V1.2.4 to Raspberry Pi Demo 2048.enz (note version 1.2.4) [4].

Préparation de Python

Commençons tout de suite avec *IDLE Integrated Development Environment* (IDE) qui comprend la panoplie *Tkinter GUI*.

L'installation Raspbian fournit IDLE pour Python 2.7 et Python 3. Notre démo GestIC se base sur Python 2.7. Lancez IDE dans une console Linux comme administrateur pour avoir un accès direct au matériel GPIO (**fig. 1**).

```
# sudo idle
```

On exécute directement le premier programme Python de la console shell comme ceci :

```
>>> print "Hello Python"
```

Accéder aux GPIO de RPi est aussi simple que ça. Chargez le module GPIO Python, initialisez les broches et lisez les signaux dans votre programme, cf. **List. 1**.

Listage 1

```
>>> import RPi.GPIO as GPIO
>>> GPIO.setmode(GPIO.BCM)
>>>
>>> GPIO.setup(17, GPIO.IN, pull_up_down
= GPIO.PUD_DOWN)
>>> GPIO.setup(18, GPIO.IN, pull_up_down
= GPIO.PUD_DOWN)
>>>
>>> while true:
>>>     if GPIO.input(17)==1:
>>>         print "GPIO 17 is HIGH"
>>>     elif GPIO.input(18)==1:
>>>         print "GPIO 18 is HIGH"
```

Si le matériel est bien configuré, le programme écrit une ligne après chaque geste (**fig. 2**).

2048 4 RPi

2048 est un jeu de réflexion inventé en mars 2014 par Gabriele Cirulli, un Italien de 19 ans, concepteur sur internet. Il faut faire glisser des carrés numérotés sur une grille de 4 x 4 de manière à en obtenir un de valeur 2048. Le jeu de Gabriele a attiré 4 millions de visiteurs en moins d'une semaine. Vous en trouverez les règles sur [5]. En réalité, 2048 n'est pas la limite ; les lecteurs d'Elektor devraient pouvoir atteindre le nombre le plus élevé possible : 131 072, soit 2¹⁷.

La version de Python que nous utilisons pour cette réalisation sur RPi a été publiée sur le blog

Tableau 1.

MGC3130	RPi	gestes
EIO1	GPIO17	← coup d'est → ouest
EIO2	GPIO18	↑ coup de sud → nord
EIO3	GPIO25	→ coup d'ouest → est
EIO6	GPIO23	↓ coup du nord → sud

de Hrvoje [6] et est ici combinée aux signaux de la gestuelle captés par le MGC3130.

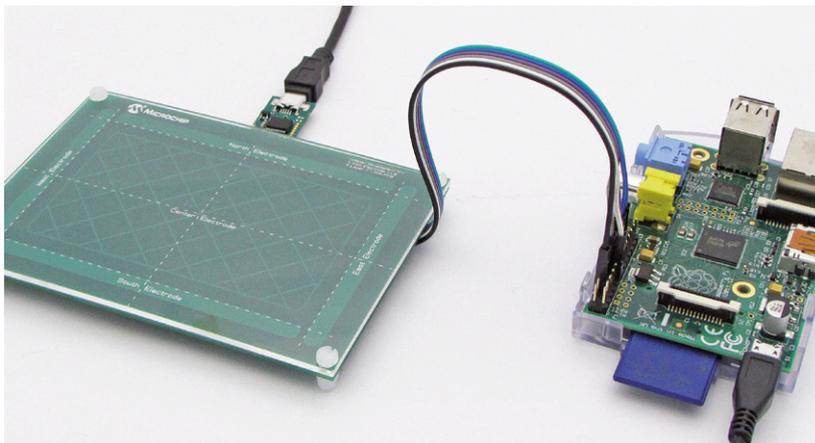
Il ne vous reste qu'à tout mettre ensemble. Téléchargez le code complet pour l'installation sur la page d'Elektor [4] et copiez le fichier dans le RPi puis lancez-le à partir du shell IDLE Python (fig. 3).

Le fichier s'ouvre dans une fenêtre d'édition et on lance le programme par Run dans le menu ou simplement par F5. Vous y jouerez alors par des mouvements dans les quatre directions cardinales au-dessus des électrodes Hillstar. C'est amusant (fig. 4).

Prochainement

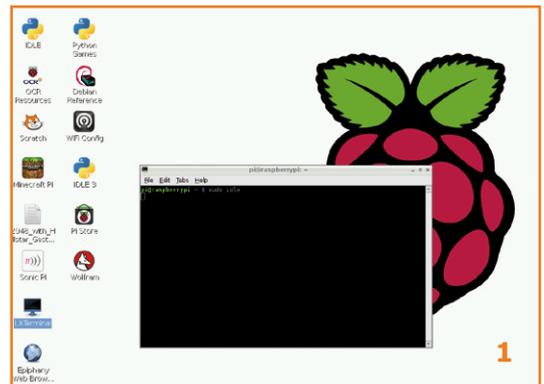
La prochaine fois, nous jouerons au 2048 avec la plaque tactile 3D contenue également dans l'offre spéciale Elektor/Microchip [1]. Nous irons aussi fouiller dans l'interface USB et le kit de développement 3DTouchpad (SDK) et nous utiliserons la plaque 3D par le port USB du RPi.

(140513 – version française : Robert Grignard)

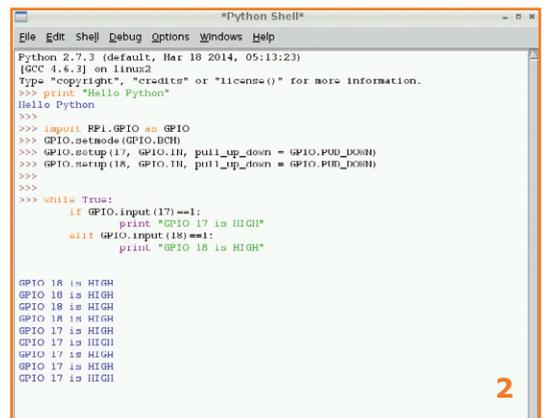


Liens

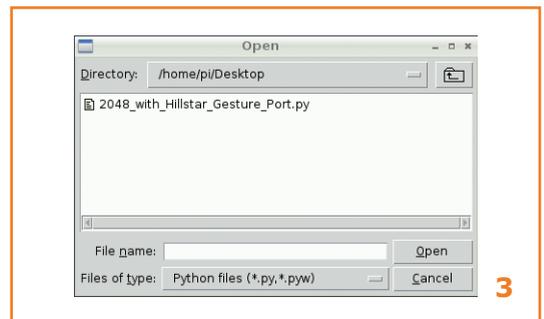
- [1] kit d'évaluation Microchip Hillstar GestIC et Touchpad 3D www.elektor.com/hillstar
- [2] www.elektor-magazine.fr/140408 : commande par gestes 3D pour µC et PC, nov. 2014
- [3] www.elektor.fr/140423 : GestIC & 3D Touchpad manip. n°1, déc. 2014,
- [4] www.elektor-magazine.fr/140513 : GestIC & 3D Touchpad : manip. n°2
- [5] [http://fr.wikipedia.org/wiki/2048_\(jeu_vidéo\)](http://fr.wikipedia.org/wiki/2048_(jeu_vidéo)) : 2048 (jeu vidéo)
- [6] <http://blog.hrvoje.org/blog/2014/09/20/a-simple-2048-clone-in-python> : blog Hrvoje



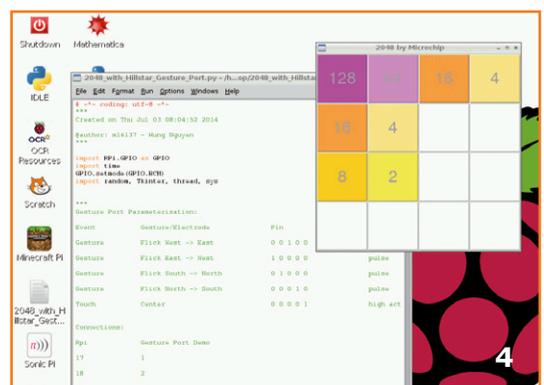
1



2



3



4

ausculter les cartes à puce synchrones avec un lecteur PC/SC

passerelle entre cartes à puce et lecteur PC/SC



Patrick Gueulle
(Étretat)

PC/SC pour le meilleur et pour le pire
Pratiquement tous les lecteurs de cartes à puce (avec ou sans contact) disponibles aujourd'hui (pour quelques euros et même gratuitement si on se débrouille bien !) se connectent à l'ordinateur par USB et sont conformes au standard *Personal Computer Smart Card (PC/SC)*. Autant dire programmables de la même façon quels qu'en soient la marque et le modèle ! Ils rendent les cartes à microprocesseur accessibles de façon transparente au travers du système d'exploitation (Windows, Linux, MacOS) grâce à un pilote spécifique, ou même générique s'ils sont en plus compatibles *Chip Card Interface Device (CCID)*. Pourtant, la programmation d'applications avec carte à puce à partir d'un langage tel que le C (ou Java) n'est pas pour autant à la portée du débutant. Heureusement, le kit logiciel (gratuit !) de

Les lecteurs de cartes à puce PC/SC, sont incontournables pour expérimenter avec les cartes *asynchrones*, autrement dit à microprocesseur. Opérer sur les simples cartes *synchrones* (télécartes et autres cartes jetables) est paradoxalement plus problématique, car la disparition des ports série et parallèle a fait perdre son intérêt à la construction de lecteurs maison. Mais non, mais non... ça n'a jamais été aussi intéressant, à condition de savoir comment s'y prendre et d'avoir le bon outil.

la BasicCard [1] permet de faire des choses étonnantes en quelques lignes de Basic : explorer une carte bancaire EMV (*Europay Mastercard Visa*) ou une carte Vitale, reparamétrer astucieusement la carte SIM d'un téléphone portable, évaluer la sécurité d'un porte-monnaie électronique ou d'un passe de transport en commun, etc. À l'inverse, tous les lecteurs PC/SC ne traitent pas de façon identique les cartes à mémoire les plus simples : chaque fabricant fournit (ou non) une *Application Programming Interface (API)* spéciale, compatible avec un choix limité de cartes synchrones, et avec laquelle on communique d'une façon propriétaire.

Un prolongateur intelligent

Mettons tout le monde d'accord avec cet adaptateur équipé d'un microcontrôleur, dialoguant

d'une part avec la carte *synchrone* au plus bas niveau protocolaire possible, et d'autre part avec n'importe quel lecteur PC/SC auprès duquel le microcontrôleur se fait passer pour une carte à puce *asynchrone*. D'un côté, un morceau de circuit imprimé (simple face) de 8/10 mm tient lieu de fausse carte qu'il suffit d'engager dans le lecteur PC/SC, de l'autre un connecteur de carte classique. Et entre les deux le microcontrôleur dont le microprogramme assure la traduction simultanée entre les micro-instructions comprises par les cartes synchrones et les *Application Protocol Data Units* (APDU) échangées par les lecteurs PC/SC avec des programmes applicatifs écrits en ZCBasic.

Nous envisageons deux cas de figure principaux : les cartes apparentées aux télécartes françaises (T1G puis T2G), et les cartes de type Eurochip apparentées à la *Telefonkarte* allemande. Malgré l'incompatibilité délibérée entre les deux pour cause de concurrence, bien des applications françaises sont basées sur le modèle européen (cartes de lavage de voitures des stations-service BP, certaines cartes de fidélité, etc.). Les différences affectent le brochage de leurs micromodules (**fig. 1 et 2**), et le chronogramme des états logiques (ces fameuses micro-instructions).

La **figure 3** décrit le protocole «6 contacts», *alias* «3 fils», pour lire et écrire dans les télécartes françaises, que la **figure 4** permet de comparer avec le protocole «5 contacts», *alias* «2 fils», des cartes Eurochip. Dans les deux cas, masse (GND) et alimentation (Vcc) sont sur les contacts ISO 5 et ISO 1, et les données série sur ISO 7. Les différences affectent les signaux de remise à zéro (RST) et d'horloge (CLK). Les puces de T2G se servent du contact ISO 4 (inutilisé par Eurochip) pour leur r. à z. fiable. Sans parler du fameux fusible de protection en écriture des T1G, que l'on pouvait griller (ou non...) en appliquant une tension élevée sur ISO 8.

La tension de programmation Vpp (21 V pour les T1G), c'est de la préhistoire. Les lecteurs PC/SC sont désormais incapables de la produire. Le contact ISO 6 a été réaffecté à des usages différents, comme p. ex. le bus monofil *Single Wire Protocol* (SWP) de certaines puces compatibles *Near Field Communication* (NFC). Pour communiquer avec la plupart des cartes synchrones en circulation, notre microcontrôleur,

un PIC 16F84 (**fig. 5**) gère quatre contacts du connecteur. Un seul suffit pour dialoguer avec le lecteur PC/SC, dont les signaux CLK et RST pilotent directement ses entrées d'horloge et de r. à z., tandis que quelques lignes E/S restent disponibles pour des variantes. L'alimentation est assurée par le lecteur et donc par le port USB de l'ordinateur auquel il est connecté.



Figure 1. Disposition des broches du micromodule sur une carte de type T2G



Figure 2. Disposition des broches du micromodule sur une carte de type Eurochip.

ISO 6 : V _{pp} (T1G)		ISO 5 : GND	
ISO 7 : sortie données		ISO 1 : V _{cc}	
ISO 8 : fusible			
ISO 2 :	ISO 4 :	ISO 3 :	Micro-instruction :
0	0		RESET
0	1		UP
1	1		PROGRAM (0 → 1)

Figure 3. Les micro-instructions sur une carte de type T1G.

ISO 7 : données		ISO 5 : GND	
ISO 6 : non connecté		ISO 1 : V _{cc}	
ISO 2 :	ISO 3 :	Micro-instruction :	
1		RESET	
0		UP	
	0	PROGRAM (1 → 0)	
0		enchaîner ces deux séquences	

Figure 4. Les micro-instructions sur une carte de type Eurochip.

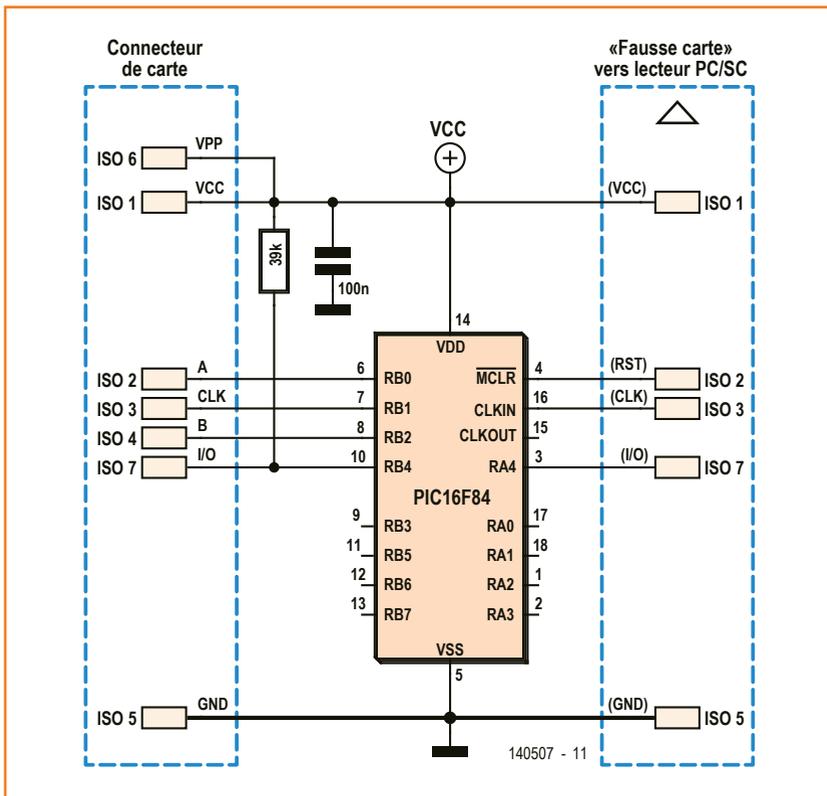


Figure 5. Le microcontrôleur PIC se fait passer pour une carte asynchrone.

Réalisation pratique

Plutôt qu'une carte double face à trous métallisés, j'ai opté pour deux circuits imprimés, l'un d'une épaisseur ordinaire de 16/10 mm (fig. 6) et l'autre, découpé avec précision, sur de l'époxy mince de 8/10 mm (fig. 7). C'est un support DIL 18 à wrapper qui assure l'interconnexion

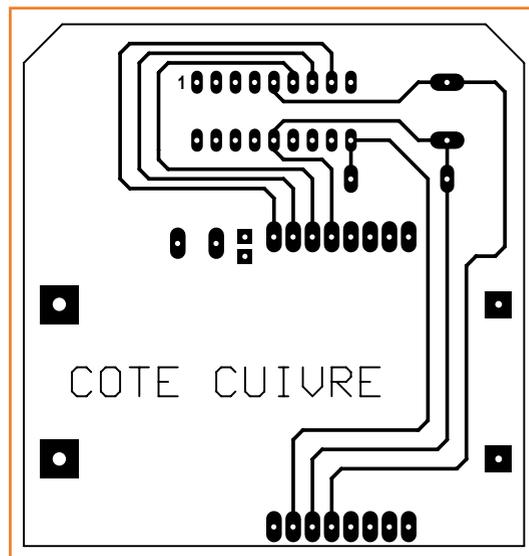


Figure 6. Dessin du circuit imprimé épais qui porte le connecteur de cartes.

entre les deux cartes. Après soudage sur la première carte, ses longues broches très rigides seront insérées dans les trous de la seconde carte, préalablement équipée de ses composants selon le plan de la fig. 8. Le μC ne se sert pas du contact «carte présente» du connecteur de carte à puce, dont on pourra donc utiliser indifféremment un modèle à contact NO ou NF. Comme seuls sont connectés les huit balais de contact correspondant aux cartes avec puce en position ISO, le circuit imprimé est compatible avec différents modèles de lecteurs. Cela convient à toutes les T2G, aux dernières séries de T1G, et naturellement aux Eurochip.

La programmation du PIC 16F84 doit être conforme au fichier **MIXTE.HEX** [1] (oscillateur en mode XT, WDT et PWRTE désactivés), faute de quoi le montage ne fonctionnerait pas. À ce stade, il est possible de procéder à un premier test sans carte : insérée dans un lecteur PC/SC, la fausse carte devra être reconnue comme une carte asynchrone $T=0$ renvoyant une *Answer To Reset* (ATR) minimaliste : **3F 00**.

Ce cas peu commun est conforme aux normes ISO 7816, et ordonne au lecteur d'alimenter la carte sous 5 V et de lui envoyer une fréquence d'horloge de 3,58 MHz, même s'il est capable d'aller plus vite. Toutes les temporisations internes du microprogramme étant dérivées de cette valeur, si elle n'était pas respectée, la pagaille serait assurée !

Micrologiciel

Écrit en assembleur (fichier **MIXTE.ASM**) pour d'évidentes raisons de rapidité d'exécution, le programme du PIC comporte deux parties distinctes, qui communiquent constamment entre elles par **call** et **ret**. La mise au point de la «bibliothèque $T=0$ convention inverse» [2] a été délicate. C'est elle qui fait passer le PIC pour une carte à puce asynchrone, acceptée par n'importe quel lecteur PC/SC, ancien ou récent. Ses routines sont utilisables pour d'autres usages, p. ex. pour émuler une carte existante. Ce concept proche de celui des *Gold cards* est supplanté aujourd'hui par la BasicCard, infiniment plus facile à programmer : en Basic et non plus en assembleur, avec toutes les ressources d'un puissant *système d'exploitation ouvert* et d'un outil de développement professionnel entièrement gratuit.

Après les initialisations, la première tâche dont s'acquitte cette partie asynchrone du programme

est de répondre au signal de *reset* envoyé par le lecteur PC/SC (début du listing source [2]). C'est simple grâce à l'appel, à deux reprises, de la routine **even** servant à l'envoi d'un octet comportant un nombre pair de bits à 1. Auparavant, la ligne de données (ISO 7) de la carte est mise en mode *sortie* (routine **tx** de la bibliothèque) et le retard réglementaire est respecté avant la transmission de l'ATR (**3F 00**). Juste après, ISO 7 est remis en mode *entrée* (routine **rx**), qui est son état de repos. Le PIC guette alors les octets en provenance du lecteur, qu'il va systématiquement analyser afin d'y reconnaître les commandes qu'il sait exécuter.

Chacune de celles-ci se présente sous la forme d'une *Transport Protocol Data Unit* (TPDU) de 5 octets, structurée de la façon suivante : **CLA, INS, P1, P2, LEN**.

CLA est la classe ISO de la commande, INS son code opératoire, P1 et P2 deux *références*, et LEN le nombre d'octets du bloc de données que le lecteur envoie (cas d'une commande *entrante*) ou attend (cas d'une commande *sortante*).

Dans notre application, les octets CLA, P1, P2, et LEN sont ignorés et peuvent donc avoir des valeurs quelconques, à l'exception de LEN qui doit être à 00 pour éviter que le lecteur attende des données que la carte ne lui enverra jamais. À toute commande conforme, la carte répond par un compte-rendu de deux octets (SW1 et SW2) dont la valeur la plus commune est **90 00**, indiquant sa bonne exécution. Mais notre PIC peut aussi répondre **6F 00** s'il reçoit un octet INS qu'il ne connaît pas.

Voici le vocabulaire limitatif des codes INS que reconnaît notre application :

- 42h : *Reset* d'une carte 5 contacts (protocole Eurochip)
- E2h : *Reset* d'une carte 6 contacts (protocole Télécarte)
- 44h : Avance du compteur d'adresse de 1 bit (5 contacts)
- E4h : Avance du compteur d'adresse de 1 bit (6 contacts)
- 22h : Écriture (mise à 0) du bit courant (5 contacts)
- D2h : Écriture (mise à 1) du bit courant (6 contacts)

Liste des composants

- 1 PIC 16F84 (programmé avec MIXTE. HEX)
- 1 résistance : 39 k Ω
- 1 condensateur : 0,1 μ F
- 1 connecteur de cartes à puce
- 1 support à wrapper DIL 18
- circuits imprimés simple face 8/10 et 16/10 mm

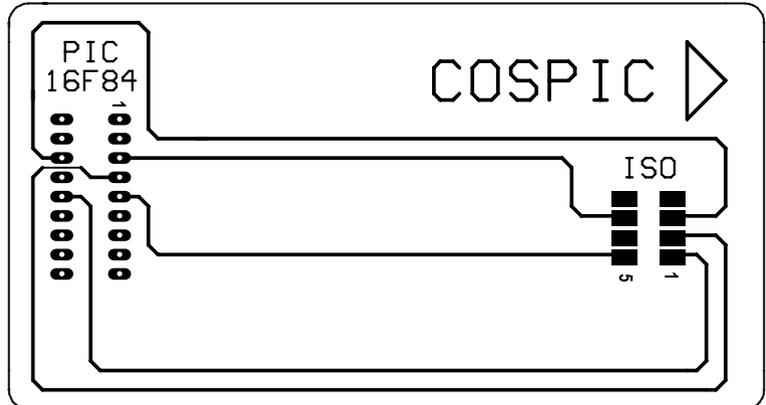


Figure 7. Dessin du circuit imprimé fin qui se fait passer pour une carte asynchrone.

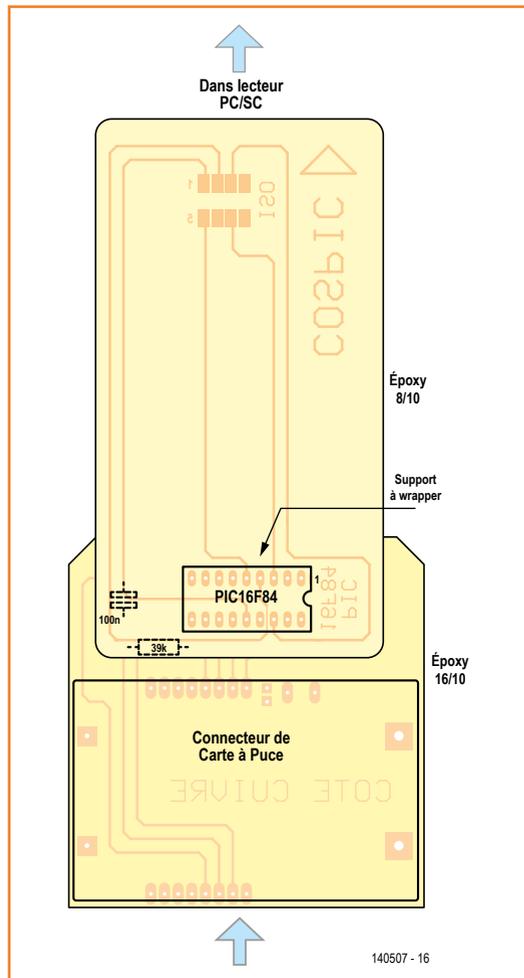


Figure 8. L'interconnexion des deux cartes est assurée par les longues broches du support à wrapper du PIC, soudées sur l'une et l'autre cartes.

Sauf erreur fatale entraînant son mutisme, le PIC répond invariablement **90 00** aux commandes de *reset* et de programmation, mais exploite une possibilité peu connue de la norme ISO 7816-3 pour remonter les données lues dans la carte synchrone lors d'une commande d'avance du compteur d'adresses (UP).

Les deux quartets dont se compose l'octet SW2 (SW1 étant à 90h) peuvent ainsi prendre les valeurs hexadécimales A (bit lu à 1) ou 5 (bit lu à 0). Le premier indique l'état du contact ISO 7 de la carte avant la commande, et le second son état après celle-ci. Cela peut paraître superflu, mais cette redondance ne coûte rien et offre la possibilité au logiciel pilotant le lecteur de procéder à une vérification de cohérence lors de la lecture séquentielle de nombreux bits, ou de s'assurer du succès d'une opération d'écriture. Le reste du microprogramme produit les micro-instructions des **figures 3** et **4**, en respectant au mieux les spécifications des fabricants des puces de cartes synchrones. Il peut être considéré comme prêt à l'emploi, mais le code source est là [2] pour que nos lecteurs le modifient ou le complètent...

Logiciels applicatifs clé en main

La complexité des protocoles de bas niveau étant gérée par le microprogramme du PIC, les logiciels applicatifs peuvent être extrêmement simples, surtout si on les développe dans un langage comme le ZCBasic du kit BasicCard. Quelques lignes de cet étonnant Basic compilé *spécial cartes à puce* suffisent pour lire l'intégralité du contenu d'une puce Eurochip ([2] code source **E2.BAS** et exécutable Windows **E2.EXE**).

```
#include CARDUTIL.DEF
#include COMMERR.DEF
ComPort = 101
Declare Command &H22 &H42 RST(S$,Disable Le)
Declare Command &H22 &H44 UP(S$,Disable Le)
Call WaitForCard : CLS
ResetCard : Call CheckSW1SW2
Print"Lecture EUROCHIP" : Print
Call RST(S$) : Call CheckSW1SW2
FOR F = 1 TO 16
FOR G = 1 TO 8
FOR H = 1 TO 4
S$ = "" : Call UP(S$)
```

```
IF SW2 = &HA5 OR SW2 = &HAA Then Print"1";
IF SW2 = &H5A OR SW2 = &H55 Then Print"0";
NEXT H:Print" ";
NEXT G:Print
NEXT F:Print
Call WaitForNoCard
```

Il suffit d'insérer la carte synchrone dans le connecteur de l'adaptateur, lancer le programme, puis d'introduire la fausse carte dans le lecteur PC/SC.

Deux lignes seulement à changer (les codes INS des commandes utilisées), et nous pourrons faire de même avec toute carte apparentée aux télécartes françaises (code source **T2.BAS** et exécutable Windows **T2.EXE**) :

```
Declare Command &H22 &HE2 RST(S$,Disable Le)
Declare Command &H22 &HE4 UP(S$,Disable Le)
```

Tenter de reconstituer la signification de ces bits est un exercice passionnant, au sujet duquel il a été beaucoup publié depuis les années 90 (voir Elektor n° 231 de février 1997 page 12, « Lecteur-encodeur de cartes à puce synchrones ». Aujourd'hui, nous en savons largement assez pour écrire un court programme capable de déterminer le solde d'unités d'une carte Eurochip, en faisant le total (pondéré) de ses compteurs (code source **CE.BAS** et exécutable Windows **CE.EXE**) Dans le cas d'une *Taxcard* de l'opérateur téléphonique Swisscom, il s'affichera par exemple 500 unités (centimes) si le crédit restant est de 5 francs suisses. Avec une carte de station-service BP, c'est le nombre d'unités de lavage restant à consommer qui apparaîtra, sachant que les cartes les plus courantes contiennent, lorsqu'elles sont neuves, 12, 24, ou 36 unités.

C'est avec les télécartes dites rechargeables d'Orange (France) que l'on fera les expériences les plus amusantes ! Un programme à peine plus compliqué retrouvera le code à gratter au verso sans avoir à toucher à sa protection opaque (code source **TR.BAS** et exécutable Windows **TR.EXE**). Il est inscrit en clair dans la puce, de même que le numéro de téléphone (secret ?) que les publiphones appellent automatiquement pour le transmettre au serveur vocal chargé d'établir les communications. En effet, avec ces télécartes de troisième (et dernière...) génération, le crédit d'unités ne réside plus dans la puce, exposé à toutes les tentatives de fraude, mais dans un sys-



Figure 9.
Le prototype de l'auteur.

tème infor-
matique centralisé.

Il ne faut pas s'impâter après l'affichage des quatre premiers chiffres, car le programme doit lire 224 bits *pour rien* avant d'arriver aux suivants. Or, avec un lecteur paresseux, cela peut prendre jusqu'à une dizaine de secondes !

Saurez vous percer le mystère de leur codage, amis lecteurs ? Nous allons vous mettre sur la voie avec le vidage suivant :

```

1000 0001 0100 0000 0000 0000 0010 0110
1000 0101 0101 0110 0001 0110 0000 1100
0000 0000 1010 1010 0010 0110 0010 0010
0000 1000 0011 0110 0100 1000 0011 0100
1111 1111 1111 1111 1111 1111 1111 1111
1111 1111 1111 1111 1111 1111 1111 1111
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0111 0101 1111 0101 0000 0000 0000 0000
1111 0101 1111 0101 1111 1111 1111 1111
0011 0010 0001 1000 1000 0110 1001 0110
1000 1001 1010 1010 0000 0000 0000 0000
1111 0101 1111 0101 1111 1111 1111 1111
1111 0101 1111 0101 1111 1111 1111 1111
1111 1111 1111 1111 1111 1111 1111 1111
1111 1111 1111 1111 1111 1111 1111 1111
    
```

Celui-ci a été obtenu avec la variante pour T2G de notre utilitaire de lecture de cartes Eurochip (code source **T2.BAS** et exécutable Windows **T2.EXE**).

Voici deux indices : le code de la carte est **26223218869689**, et le numéro de téléphone du serveur **08 36 48 34 00**. Contrairement au **08 09 38 20 00**, il n'est pas gratuit depuis une ligne fixe...

(140507)

Liens

[1] www.basiccard.com

[2] www.elektor-magazine.fr/140507



L'auteur

Patrick Gueulle est ingénieur radioélectronicien et informaticien. Il est aussi journaliste indépendant, auteur d'une quarantaine d'ouvrages de vulgarisation de techniques de pointe auprès d'un public d'amateurs avertis et de développeurs professionnels. Spécialisé dans les cartes à puce, il a dirigé ses efforts dans le sens de l'interopérabilité : conception d'outils permettant à chacun de lire et écrire dans les cartes en circulation, et mise au point de cartes « alternatives » utilisables dans les terminaux existants. Inconditionnel des systèmes d'exploitation ouverts, il développe principalement sur BasicCard.

CC2-eBoB

capteur d'humidité/de température ChipCap2



Luc Lemmens
(elektor.labs)

Si les superbes capteurs d'humidité/température ChipCap2 (ou CC2) d'*Amphenol Advanced Sensors* se laissent souder à la main, leur empreinte ne passe malheureusement dans aucune plaque d'essais. Le prototypage serait facilité par une mini-carte avec un connecteur au pas standard de 2,5 mm (0,1 pouce) qui donnerait accès, de l'extérieur, à toutes les broches du capteur. Notre deuxième eBoB (*e(lektor)Break out Board*) vient de naître.

La précision du capteur d'*Amphenol Advanced Sensor*, étalonné et testé individuellement, est de $\pm 2\%$ pour une humidité relative comprise entre 20 et 80%, ou de $\pm 3\%$ sur l'ensemble de la plage. Un composant simple à utiliser, sans étalonnage additionnel ni compensation de température. Un rêve ! Prêt à l'enficher dans une carte d'expérimentation ? Oui, grâce à notre e-BoB.

Tout droit sorti d'une cave

Cette petite carte a été conçue pour un projet sur le site elektor.labs [1]. Son objet : la mesure, en intérieur et extérieur, de l'humidité relative (HR) et de la température (T°). Selon les valeurs mesurées on commande l'ouverture d'une fenêtre et la marche d'un ventilateur afin de garder dans une plage acceptable l'humidité d'un espace confiné,

une cave (à vin). Pour les critères de ventilation, le système calcule et prend en compte également les niveaux d'humidité absolue. En outre, la température intérieure est surveillée pour une protection contre le gel.

Ce concept original faisait appel à deux capteurs d'humidité/température novateurs : HYT939 d'*IST (Innovative Sensor Technology)*. Vu le prix de ces composants, nous avons préféré des capteurs d'humidité cinq fois moins chers : ChipCap2 d'*Amphenol*. Comme les capteurs des deux marques peuvent être connectés au bus I²C — et qu'ils semblent même compatibles (même adresse I²C par défaut, etc.) — ils devraient faire l'affaire pour notre système de ventilation, au prix de modifications insignifiantes du logiciel. Nous n'avons pas (encore) étudié le HYT939 dans ses moindres détails.

Interfaçage analogique et numérique

Le CC2 est utilisable en mode I²C (type CC2D de notre application) ou PDM (*Pulse Density Mode* = mode de densité d'impulsions ; CC2A) ; dans ce dernier cas, deux broches distinctes délivrent — après filtrage passif — des signaux de T° et RH analogiques (versions disponibles – **tableau 1**). Notre carte d'évaluation est conçue pour les versions analogique et numérique de ces capteurs — cependant cela impliquera le remplacement de certains composants — nous travaillerons ici, principalement, avec I²C, donc un CC2D.

Plots du capteur

La tension d'alimentation (2,7 à 5,5 V, selon la version) est appliquée sur les îlots (non pas des broches) d'alimentation VDD et VSS (GND, contacts 7 et 6). Ces plots devraient être découplés par un condensateur de 220 nF. V_{CORE} est une tension interne ; à connecter à la masse uniquement via un condensateur de 100 nF.

Les transferts des données ont lieu sur le plot SDA (3) ; le plot SCL (4) synchronise la communication entre CC2 et le µC. CC2 comporte un oscillateur interne avec compensation de T° qui fournit une base de temps générique ; il utilise un protocole de communication compatible I²C avec prise en charge des taux de transmission de 100 à 400 kHz. Des résistances de polarisation (*pull-up*) externes forcent le signal de commande au niveau haut.

Les sorties d'alarme (plots 1 et 8) sont utiles pour vérifier si la valeur mesurée par le capteur est inférieure ou supérieure à des valeurs de RH pré-programmées. L'alarme pourra pour piloter une charge à drain ouvert reliée à VDD, ou fonctionner en *push-pull* intégral (*full push-pull*). Pour une application à haute tension, des bro-

Tableau 1. CC2 (ChipCap2) – les versions

Désignation	Description
CC2A25	analogique, 2%, 5 V
CC2A23	analogique, 2%, 3,3 V
CC2D23S	numérique, mode veille, 2%, 3,3 V
CC2D25S	numérique, mode veille, 2%, 5 V
CC2D23	numérique, 2%, 3,3 V
CC2D25	numérique, 2%, 5 V
CC2D35	numérique, 3%, 5 V
CC2A33	analogique, 3%, 3,3 V
CC2D33S	numérique, mode veille, 3%, 3,3 V
CC2D35S	numérique, mode veille, 3%, 5 V
CC2D33	numérique, 3%, 3,3 V
CC2A35	analogique, 3%, 5 V

ches d'alarme commanderont les périphériques externes (**fig. 1**).

Les deux sorties d'alarme sont utilisables simultanément, le cas échéant en combinaison avec I²C. Dans l'EEPROM interne du CC2A, il est possible de configurer les seuils d'activation/désactivation des alarmes, le type de sortie (drain ouvert, *push-pull*, par ex.) et le niveau de sortie actif, voir plus loin. Une fois la configuration de ces broches effectuée via I²C, le capteur est autonome : un système de contrôle d'humidité de base sans µC.

Un front montant sur la broche EOC/Ready indique que de nouvelles données sont prêtes sur l'interface I²C. La broche Ready reste haute jusqu'à la réception d'une commande de récu-

Spécifications du ChipCap2 (CC2)

Capteur : humidité relative (RH%)

- Résolution : 14 bits (0,01%RH)
- Précision : ±2,0 %RH (20–80%RH)
- Répétabilité : ±0,2 %RH
- Hystérésis : ±2,0 %RH
- Linéarité : <2,0 %RH
- Temps de réponse : 7,0 s (τ 63%)
- Coefficient de temp. : max 0,13 %RH/°C (à 10–60 °C, 10–90%RH)
- Plage de service : 0–100 %RH (hors-condensation)

- Dérive à long terme : <0,5 %RH/an (conditions normales)

Capteur : température (°C)

- Résolution : 14 bits (0,01 °C)
- Précision : ±0,3 °C
- Répétabilité : ±0,1 °C
- Temps de réponse : 5,0 s (τ 63%)
- Plage de service : –40 à 125 °C
- Dérive à long terme : <0,05 °C/an (conditions normales)

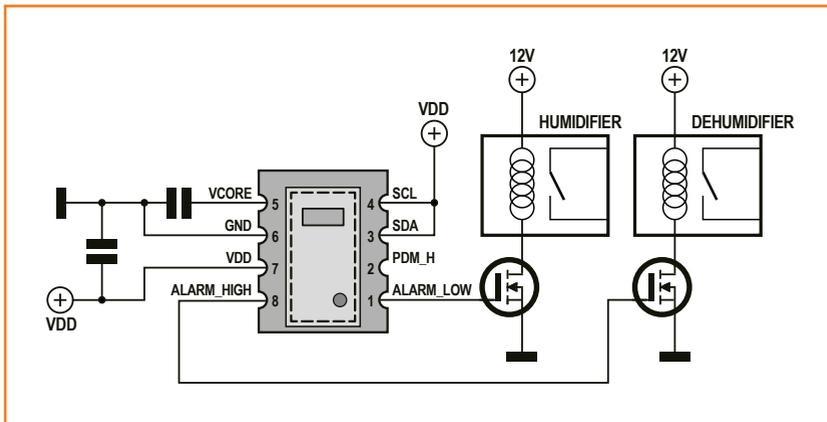


Figure 1.
CC2 (ChipCap2) en application autonome.

pération de données (DF = *Data Fetch*) ; elle le reste, même en cas de mesures supplémentaires effectuées avant le signal DF. Le bit Ready_Open_Drain du mot d'EEPROM Cust_Config fixe le type de la sortie : *push-pull* ou drain ouvert. Généralement, la communication point-à-point se fait en *push-pull*. Si une application exploite plusieurs modules, l'option drain ouvert permet d'interconnecter tous les sous-ensembles à un seul fil et une résistance *pull-up* en utilisant un bus.

Pour plus d'informations sur le mode PDM, voir le *ChipCap 2 Application Guide* [2]. Dans ce mode, le plot ALARM_L est utilisé pour le signal de T°, seule la sortie ALARM_H reste disponible.

Des plots aux broches

Le matériel du CC2-eBoB (fig. 2) est rustique : une petite platine pour rendre les broches du capteur accessibles sur une embase mâle à 7 contacts au pas de 2,5 mm. Pour la plupart des applications I²C 4 broches (SDA, SCL, VDD et GND) suffiront. Les trois autres (EOC/Ready, AL_H et AL_L) seront omises pour économiser de la place. Le découplage des plots de l'alimentation et

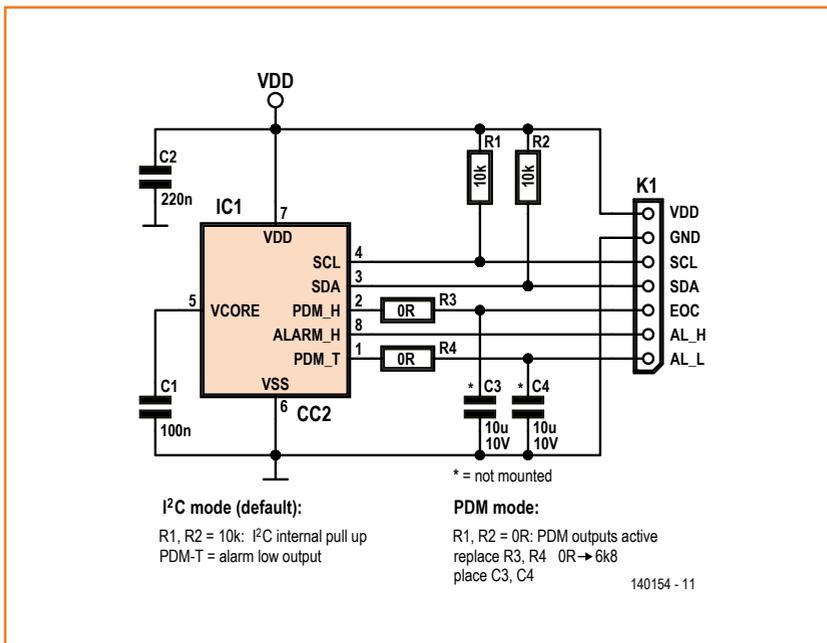


Figure 2.

Qui dit mini-puces dit mini-schémas ! Nous découvrons ici le capteur d'humidité et de température CC2A dans sa configuration la plus dépouillée.

Modes de mesure

Le CC2 (ChipCap2) est programmé en usine pour fonctionner en mode *Sleep* (veille) ou *Update* (mise à jour). En mode *Sleep*, il attend l'envoi de commandes du maître avant d'effectuer des mesures.

Récupération de données en Mode Update

On utilise I²C pour récupérer des données du registre de sortie numérique via une commande de récupération de données (DF = *Data Fetch*). La détection de données prêtes peut prendre la forme soit d'une interrogation ou d'un suivi de la broche Ready. Les bits d'états de DF disent si les données sont valides ou non, voire périmées. Une fois un cycle de mesure terminé, il est possible de récupérer les

données valides. Si la récupération des données suivante est effectuée trop tôt, les données seront les mêmes que lors de la récupération précédente, mais avec des bits d'état indiquant qu'elles sont périmées. On pourra utiliser un front montant sur la broche Ready pour signaler la présence de données valides.

Récupération de données en Mode Sleep

Le noyau du CC2 n'effectuera de conversion qu'à la réception d'une commande de demande de mesure (MR = *Measurement Request*) ; sinon, le CC2 reste hors-tension. L'émission de commandes MR n'est possible que via I²C ; on ne dispose donc pas du mode *Sleep* en mode PDM.

Liste des composants

Résistances :

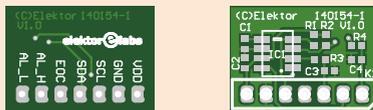
R1,R2 = 10 k Ω 1% 100 mW (0603)
R3,R4 = 0 Ω 100 mW (0603)

Condensateurs :

C1 = 100 nF 16 V 10% X7R (0603)
C2 = 220 nF 10 V 10% X7R (0603)

Semi-conducteurs :

IC1 = CC2D35, capteur d'humidité/de température numérique (Amphenol Advanced Sensors)



Divers :

K1 = barrette SIL à 7 contacts au pas de 2,5 mm (0,1 pouce)
CC2-eBoB monté, prêt à l'emploi, dans l'e-choppe d'Elektor 140154-91

Figure 3.
Vue recto-verso de la platine du CC2-eBoB.

de VCore respecte les prescriptions de la fiche technique.

La carte comporte des résistances de 10 k Ω sur les lignes SDA et SCL. Il est possible de connecter jusqu' à quatre CC2-eBoB au bus sans modifications. Si vous avez besoin de plus de capteurs, dessoudez leurs résistances R1 et R2.

On pourra remplacer les résistances de 0 Ω R3 et R4 par des valeurs appropriées et C3, C4 peuvent être ajoutés pour filtrer les signaux de T° et RH lorsque le CC2 est en mode PDM. Pour activer PDM, R1 et R2 seront court-circuitées, c.-à-d. SDA et SCL connectées à VDD.

Logiciel

Les spécificités des fonctions des **Modes de mesure** et du **Mode de commande**, apparaissent dans leurs **encadrés** respectifs.

Les périphériques I²C ont des adresses fixes sur le bus I²C — nombre d'entre eux ont une ou plusieurs broches d'adresses externes pour raccorder plusieurs périphériques identiques à des adresses uniques sur le même bus. Le CC2 possède une adresse I²C prédéfinie stockée dans son EEPROM interne, dans le registre de Configuration personnalisée (Cust(om)_Config(uration), cf. tableau). Avec le paramétrage par défaut, tous les composants répondent à la même adresse modifiable uniquement par logiciel.

Nous avons découvert une bibliothèque Arduino (et Python) de Richard Wardlow [3], qui contient la plupart des définitions et des fonctions nécessaires pour communiquer avec les capteurs CC2D en mode I²C. On y trouve aussi un exemple de *sketch* Arduino pour écrire certains paramètres dans le capteur et lire les valeurs d'humidité/température, affichées ensuite par le moniteur de l'environnement de programmation d'Arduino. C'est parfait pour tester ce BoB. Il faudra que les signaux I²C SDA et SCL soient connectés aux

broches correspondantes de l'Arduino.

La bibliothèque contient aussi des fonctions de gestion de la broche READY (fin de conversion) du CC2A, mais nous préférons interroger les bits d'état du capteur via I²C pour savoir si les valeurs d'humidité/température sont prêtes.

Cette bibliothèque ne supporte malheureusement pas la reprogrammation de l'adresse I²C, c'est pourquoi nous avons écrit un simple *sketch* (CC2A_set_I2C_address.ino) [4]. Nota : la broche d'alimentation du BoB doit être connectée à une broche de sortie numérique (PB0) de l'Arduino contrôlée par logiciel. Ce *sketch* comporte la définition de deux constantes :

La fringale d'Amphenol

Amphenol (American Phenolic Corp., 1932) est synonyme de connecteurs. Au nombre de leurs produits les plus célèbres citons la prise SO-239 et la fiche PL-259 correspondante et leur invention de la fiche BNC, où BNC signifie *Bayonet Neill-Concelman*. En 2013 Amphenol a fait l'acquisition d'Advanced Sensors Business de GE pour 318 M\$. Auparavant, ils ont quasi-absorbé Bendix (USA), Jaybeam Wireless (Japon), SV Microwave (USA), Teradyne (USA) et SEFEE (France).

CC2A en mode PDM

Bien que, pour cette application du capteur d'humidité/température, l'accent soit mis sur le mode I²C, le CC2-eBoB peut également être utilisé en mode PDM. Dans ce cas, les broches 1 et 2 délivrent des impulsions, images des valeurs de T° et de RH mesurées par le capteur. Après un filtrage passif du 1^{er} ordre (via R4/C4 et R3/C3, respectivement, cf. le guide d'application CC2A), des conversions A/N permettent de déterminer les valeurs.

Pour utiliser ce mode, les deux lignes I²C sont connectées à VDD. Dans ce cas, sur nos eBoB, R1 (SCL) et R2 (SDA) ne sont plus des résistances de 10 k Ω , mais des courts-circuits (ponts).

Mode de commande

Les commandes de ce mode (Cf. **tableau 2**) ne sont prises en charge que pour le protocole I²C. Les capteurs ChipCap2 ont une EEPROM interne pour l'enregistrement des paramètres associés aux plots d'alarme (seuils, configuration de broches, etc.), la configuration du plot Ready, la longueur de la fenêtre de commande et l'adresse I²C du capteur.

Après la mise sous tension, on dispose d'une courte fenêtre de commande pendant laquelle il est possible d'activer ce mode par envoi d'une commande *Démarrer mode de commande*. L'utilisateur peut alors accéder à la plage d'adresses EEPROM entre 0x16 à 0x1F (**tableau 3**) jusqu'à

l'envoi d'une commande *Démarrer mode normal*, le capteur revient alors au mode normal.

Notez que la modification des paramètres en EEPROM — ou du moins l'entrée en mode de commande — doit être effectuée immédiatement après la mise sous tension de votre circuit. Autre option : activer la broche VDD du capteur par logiciel si elle est connectée à une broche de port d'un µC p. ex. ; dans ce cas, le capteur peut être réinitialisé à chaque fois que vous devez modifier sa configuration pendant l'exécution du programme. Consultez le **tableau 4** si vous souhaitez personnaliser le fonctionnement du CC2.

Tableau 2. Liste des commandes et encodages

octet de commande 8 bits de commande (Hex)	3 ^e et 4 ^e octets 16 octets de données (Hex)	description	temps de réponse
0x16 à 0x1F	0x0000	Lecture des adresses en EEPROM 0x16 à 0x1F Après émission et exécution de cette commande, il doit y avoir récupération de données.	100 µs
0x56 à 0x5F	0xYYYY (Y=données)	Écriture aux adresses en EEPROM 0x16 à 0x1F Les 2 octets de données seront écrits à l'adresse spécifiée par les 6 bits de poids faible (LSB) de l'octet de commande.	12 µs
0x80	0x0000	Start_Norm Termine le mode de commande et passe au mode normal	
0xA0	0x0000	Start_CM Démarrer Mode de Commande : sert à entrer dans le mode d'interprétation de commande. Start_CM n'est valide que pendant la fenêtre de commande de mise en fonction.	100 µs

Tableau 3. Plage des adresses EEPROM (partie)

Mot EEPROM	Plage bits	IC par défaut	Nom	Description et notes
16HEX	13:0	0x3FFF	PDM_Clip_High	Limite écrêtage PDM haute
17HEX	13:0	0x3FFF	PDM_Clip_Low	Limite écrêtage PDM basse
18HEX	13:0	0x3FFF	Alarm_High_On	Alarme haute, seuil pour activation
19HEX	13:0	0x3FFF	Alarm_High_Off	Alarme haute, seuil pour désactivation
1AHEX	13:0	0x3FFF	Alarm_Low_On	Alarme basse, seuil pour activation
1BHEX	13:0	0x3FFF	Alarm_Low_Off	Alarme basse, seuil pour désactivation
1CHEX	15:0	0x0028	Cust_Config	Configuration Client
1DHEX	15:0	0x0000	Réservé	Mot réservé : ne pas modifier
1EHEX	15:0	0x0000	Cust_ID2	ID Client octet 2
1FHEX	15:0	0x0000	Cust_ID2	ID Client octet 3

- CURRENT_I2C_ADDRESS = 0x28 (réglage par défaut !)
- NEW_I2C_ADDRESS = 0x22 (ou toute autre valeur unique comprise entre 0x00 et 0x7F).

Le *sketch* change l'adresse par une simple écriture à l'adresse EEPROM 0x1C (registre de configuration personnalisée du CC2D). Nous écrivons dans un registre à 16 bits et l'adresse I²C se trouve dans ses 7 bits de poids faible. Tous les bits de

poids plus fort sont à zéro à la sortie d'usine, la plupart d'entre eux sont affectés à des paramètres spéciaux des broches d'alarme configurables par l'utilisateur. Ce *sketch* remettra ces paramètres aux valeurs par défaut.

Une petite étiquette avec la nouvelle adresse I²C collée sur le dos de la CC2-eBoB n'est sans doute pas une mauvaise idée !

(140154 – version française : Guy Raedersdorf)

Liens

[1] Basement Ventilation:

www.elektor-labs.com/project/basement-ventilation-system-control-unit-140432.14275.html

[2] ChipCap2 Application Guide:

www.digikey.com/document-redirector?doc=http://www.digikey.com/Web%20Export/Supplier%20Content/GESensing_45/PDF/ge-sensing-chipcap2-application-guide.pdf

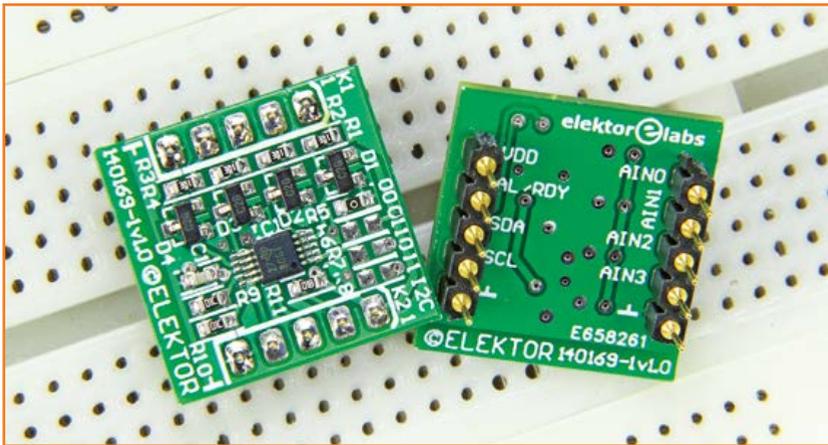
[3] Bibliothèques Arduino & Python : <https://github.com/circuitsforfun/ChipCap2>

[4] Page de ce projet : www.elektor-magazine.com/140154

Tableau 4. Attribution des bits de Cust_Config				
plage par défaut	IC par défaut	nom	description et notes	
6:0	0101000	Device_ID	Adresse esclave I ² C	
8:7	00	Alarm_Low_Cfg	Configurer la broche de sortie Alarm_Low :	
			Bits	Description
			7	Polarité Alarme 0 = active haut 1 = active bas
8	Configuration de sortie 0 = push-pull intégral 1 = drain ouvert			
10:9	00	Alarm_High_Cfg	Configurer la broche de sortie Alarm_High :	
			Bits	Description
			9	Polarité Alarme 0 = active haut 1 = active bas
10	Configuration de sortie 0 = push-pull intégral 1 = drain ouvert			
12	0	Ready_Open_Drain	La broche Ready est : 0 = push-pull intégral 1 = drain ouvert	
13	0	Fast_Startup	Définit la longueur de la fenêtre de commande : 0 = fenêtre de commande 10 ms 1 = fenêtre de commande 3 ms	
15:14	00	Réservé	ne pas modifier – Conserver les paramètres d'usine	

BoB ADS1115

4 voies de conversion A/N à 16 bits sur I²C



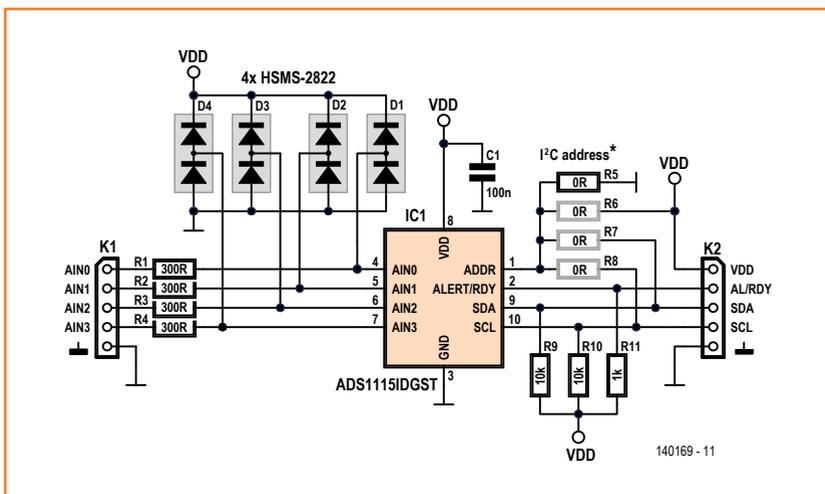
Ton Giesberts (Elektor Labs) &
Jan Buiting (Elektor)

Pour les microcontrôleurs (et pour certains de leurs utilisateurs), le monde analogique ne devient intelligible qu'une fois numérisé. C'est exactement le rôle de cette carte, petite mais quelle précision !

Les CAN et CNA sont les entremetteurs indispensables entre les mondes étrangers de l'analogique et du numérique. Notre récepteur d'ELFes et notre module CA/N [1],[2] publiés récemment semblent sans lien tant du point de vu du domaine que du public visé. Ils ont pourtant quelque chose en commun : la puce de conversion A/N ADS1115 de Texas Instruments [3], parfaite pour concevoir un circuit d'entrée générique pour un système d'acquisition de données relativement lent (< 1 kHz) mais précis.

Dans notre article module CA/N, la puce ADS1115IDGSGT avait atterri sur une carte (réf. 130485-1) mixte (CMS/traversants) en compagnie d'un ampli op, de quelques connecteurs et d'un pilote de LED. Le tout avait été conçu pour être relié facilement à une large gamme de cartes, d'Elektor ou pas : série EEC/Gnublin, serveur web Xmega et le *shield* d'extension. Le récepteur d'ELFes sort du lot : une application presque tout analogique qui vous procurera des heures de plaisir à fouiller dans le bruit.

Figure 1.
L'essence même de l'ADS1115-BoB (Breakout Board) est le faible nombre de composants qu'elle utilise.



Le problème des nouvelles puces...

... logées dans des boîtiers insoudables à la main est récurrent. Encouragés par le bon accueil reçu par les deux articles cités et par considération pour tous ceux qui meurent d'envie de s'amuser avec une vraie puce texane, nous avons décidé d'*elektoriser* la petite ADS1115IDGSGT, en la plaçant sur une carte BoB* minimaliste mais bien conçue, disponible dans l'e-choppe d'Elektor.

BoB à poil

Comme le montre le schéma (**fig. 1**), il est facile d'en faire un très flexible CAN sur 16 bits à 4 voies : c'est l'essence même d'une carte BoB. Nous avons suivi les recommandations de

* BreakOut Board — une telle carte facilite l'accès à tous les signaux d'un circuit complexe, inaccessibles autrement tant le boîtier est petit.

la feuille de caractéristiques de TI pour R1-R4 et D1-D4 (protection des entrées). Les résistances de $300\ \Omega$ limitent le courant de fuite des diodes Schottky à une valeur négligeable équivalente à $\frac{1}{2}$ LSB.

L'ADS1115 est une puce I²C pouvant répondre à quatre adresses, d'où la présence des résistances $0\ \Omega$ de configuration (cavaliers) R5 à R8 sur la broche ADDR. Une seule des résistances devrait être montée, R5 par défaut. Des informations sur la sélection de l'adresse esclave sont données dans [2] et [3].

La **figure 2** montre le schéma d'implantation et la liste des composants de l'ADS1115-BoB, qui mesure $19,1 \times 19,7$ mm. Dans notre e-choppe, nous proposons l'ADS1115-BoB prête à l'emploi, ce qui n'empêchera pas ceux d'entre vous qui ont les outils et les ressources pour reproduire le projet chez eux de réaliser leurs désirs masochistes.

Les broches étiquetées de la carte se trouvent en dessous, ce qui permet d'enficher l'ADS1115-BoB sur une plaque d'essais. Une des rangées comporte les entrées du CAN et la masse, l'autre les alimentations (VDD et masse), les lignes I²C et la ligne Alert/Ready. Si vous n'avez pas l'intention d'utiliser l'ADS1115-BoB sur une plaque d'essais, alors de la barrette sécable standard suffira pour les connecteurs.

Cet article de présentation ne traite que de la partie matérielle, mais vous trouverez des exemples de code dans [2] ainsi que des explications sur comment se lier à une bibliothèque. Le tout se trouvait déjà dans une archive associée à notre module CA/N mais pourra également être téléchargé depuis la page de cet article-ci [4].

(140169 - version française : Kévin Petit)

Liens

- [1] réception d'ELFes, Elektor sept. 2014, www.elektor-magazine.fr/140035
- [2] module CA/N, Elektor sept. 2014, www.elektor-magazine.fr/130485
- [3] Doc. de l'ADS1115 : www.ti.com/lit/ds/symlink/ads1115.pdf
- [4] Le logiciel (identique à 130485-11.zip): www.elektor-magazine.fr/140169



Liste des composants

Résistances

R1-R4 = $300\ \Omega$ 1 % 0,1 W, CMS 0603
 R5-R8* = $0\ \Omega$ 1 %, 0,1 W, CMS 0603
 R9,R10 = $10\ k\Omega$ 1 %, 0,1 W, CMS 0603
 R11 = $1\ k\Omega$ 5 %, 0,1 W, CMS 0603
 *n'en monter qu'une à la fois : R5 par défaut

Condensateurs

C1 = $100\ nF$ 10 %, 16 V, 0603 X7R

Semiconducteurs

D1-D4 = HSMS-2822-TR1G (Avago Industries)
 IC1 = ADS1115IDGST (Texas Instruments)

Divers

K1,K2 = barrette sécable 5 voies au pas de 2,54 mm
 Ou pour montage sur une plaque d'essais :
 K1,K2 = connecteur SIL 5 voies à broches rondes au pas de 2,54 mm, Harwin D01-9923246

Carte préassemblée : e-choppe réf. 140169-91
 En option : C.I. seul, e-choppe réf. 140169-1, v1.0

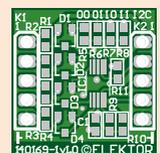
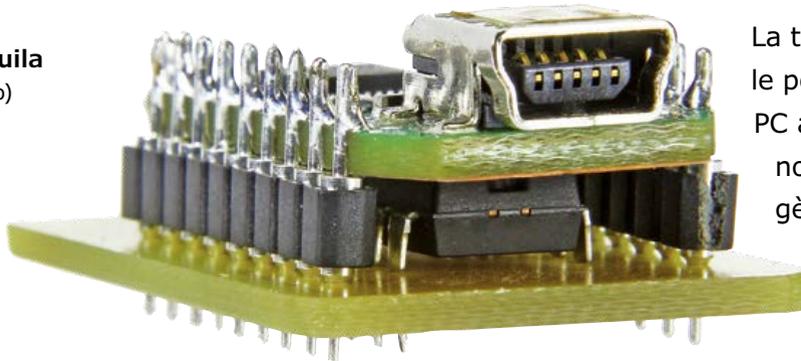


Figure 2.
Schéma d'implantation des composants de l'ADS1115-BoB (taille réelle). Rassurez-vous, vous n'aurez pas à souder ces minuscules CMS vous-même : la carte est disponible préassemblée.

BOB-SCAP

Une passerelle du passé vers le futur : USB ↔ ISP

André Aguila
(Burkina Faso)



Si, comme moi, vous n'êtes pas (encore) équipé pour travailler avec les composants montés en surface, vous utilisez sans doute des microcontrôleurs en boîtier DIP de chez Atmel. Pour communiquer avec ces microcontrôleurs-là, il ne reste que la liaison série, ce qui tombe mal puisque, comme chacun sait, les ordinateurs ne sont malheureusement plus équipés de tels ports. Oui, jeune lecteur qui fais de grands yeux en lisant ceci, il a existé il n'y a pas si longtemps des PC équipés de ports série ! Ce n'était pas si mal et ça s'appelait même le bon vieux temps !

Préliminaires

Pour résoudre ce problème de communication, il y a l'**interface USB-série** comme celle par laquelle j'ai pendant quelques années connecté à mon PC mon enregistreur de données (oui, jeune homme, c'est ce que tu appelles *data logger*) ou encore mon programmeur d'AVR Atmel. Une solution qui présente plusieurs inconvénients, dont celui de l'encombrement. Outre l'interface USB-série, déjà encombrante elle-même, il faut, avec ce type de liaison, se résoudre à prévoir une alimentation spécifique, du moins si on veut éviter de trop en demander au port USB (courant < 500 mA). De plus, il faut ajouter au montage une D-sub 9 broches et un MAX232 pour convertir les signaux au standard RS-232 du port série à ceux au standard TTL du microcontrôleur. C'est un comble, quand on sait que l'interface USB-série s'empresse de reconverter les signaux TTL en RS-232 !

La tendance générale est de privilégier le port USB pour faire communiquer un PC avec un microcontrôleur. Or, bon nombre de ces petits prodiges ne gèrent pas encore le port USB comme une seconde nature. D'où l'intérêt d'une conciliation USB-ISP.

Après avoir fait la connaissance puis utilisé quelques passerelles BOB-FT232R [1] d'Elektor, j'ai imaginé une solution d'interface USB-série plus satisfaisante à mon sens. Ces petits modules peuvent être assez facilement ajoutés à des appareils existants, tels mon enregistreur de données et mon programmeur, dont j'ai donc remplacé les D-sub par des prises USB mini-b (celles des BOB) bien moins encombrantes. En supprimant les MAX232 et en connectant directement les BOB aux μC , j'ai constaté la possibilité d'un débit bien plus élevé entre PC et μC : 921.600 bauds, alors qu'avec le MAX232 je plafonnais à 128 kbauds. Enfin, l'alimentation 5 V des modules est maintenant prélevée directement sur le BOB lui-même. Le câble USB-A-mâle-vers-mini-b est désormais le seul lien entre un module et le PC.

BOB & SCAP

Ces qualités du BOB m'ont inspiré une autre application – qui est le sujet proprement dit du présent article -- que j'ai appelée BOB-SCAP puisqu'il s'agit d'une interface pour le « SCAP, un programmeur AVR simple ». Sous ce titre, Elektor a publié son numéro de juillet/août 2011 [2] une interface série qui proposait de résoudre le problème de *la poule et de l'œuf*, bien connu de qui construit son premier programmeur AVR : ce programmeur a lui-même besoin d'un AVR programmé, ce pour quoi il faut un programmeur ! Dans l'article cité, SCAP reliait simplement un port série au connecteur ISP à six contacts d'Atmel (**fig. 1**).

Comme passerelle USB-série (**fig. 2**), j'ai choisi un BOB [3], car le simple câble FTDI ne conviendrait pas, faute de signaux DCD et DTR du port RS-232, requis par SCAP.

Dans le circuit publié en 2011, deux résistances étaient nécessaires pour limiter le courant entre le port série et l'AVR (fig. 1). Ce courant à limiter est occasionné par les tensions sur le port RS-232 de +15 V et -15 V alors que l'AVR n'accepte des tensions qu'entre 0 et 5 V (TTL). Mon BOB-SCAP (fig. 2), en utilisant un BOB comme liaison à l'ISP (K1), peut se passer de ces résistances puisque les broches du BOB sont en TTL.

Pour rester compatible avec le programmeur SCAP, il faut en revanche inverser tous les signaux du BOB ; en effet, +5 V représente un '1' en TTL alors que c'est -15 V en RS-232, de même 0 V représente un '0' en TTL, mais c'est +15 V en RS-232. Pour inverser trois signaux, tout inverseur TTL peut faire l'affaire. J'ai utilisé un simple 7404.

Porte-BOB

Pour obtenir une construction assez robuste mécaniquement, j'ai confectionné un porte-BOB (**fig. 3**) qui accueille d'une part le 7404, couvert par le BOB soudé entre une double rangée de picots. D'un côté la mini-fiche USB du BOB, de l'autre le connecteur ISP K1. Yapuka programmer.

Programmons

Pour programmer un AVR avec BOB-SCAP, on utilisera le programme *open source* AVRDUDE en ayant pris soin de rajouter les lignes ci-dessous dans le fichier `avrdude.conf`.

Notez que le signal *reset* doit être défini dans le fichier de configuration d'AVRDUDE même s'il n'est pas utilisé, dans notre cas la ligne *reset* est à la masse.

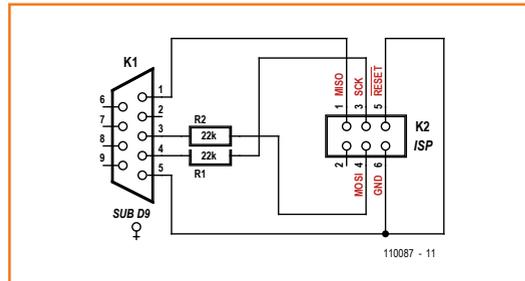


Figure 1.
Pour mémoire, le schéma de l'interface série pour SCAP publiée en 2011.

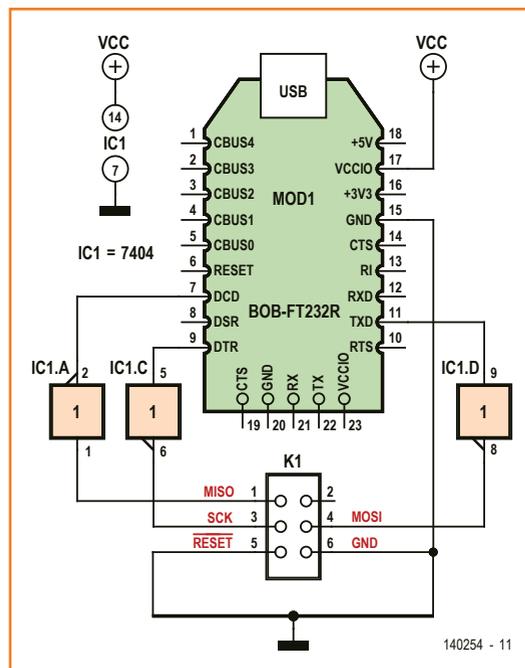


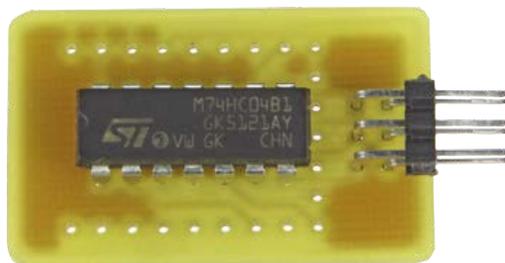
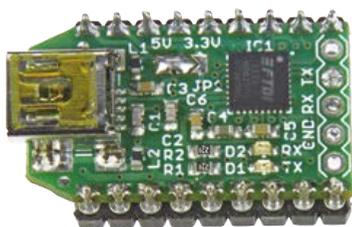
Figure 2.
Associé à trois inverseurs, le BOB FTDI d'Elektor permet de réaliser une interface USB-série pour le programmeur SCAP. NB : Avec un 7404, le circuit fonctionne sans condensateur de découplage ; avec un 74HC04, il serait sage d'en rajouter un sur ses broches d'alimentation. Sur un HC, il ne faut d'ailleurs pas laisser en l'air l'entrée de l'inverseur inutilisé !

```
# -----
# BOB-FT232R - Serial Cheap AVR Programmer (BOB-SCAP)
# reset=rts sck=dtr mosi=txd miso=dcd
programmer
  id   = "bob-scap";
  desc = "BOB-FT232R - Serial Cheap AVR Programmer, reset=rts sck=dtr mosi=txd miso=dcd";
  type = serbb;
  reset = 7;
  sck   = 4;
  mosi  = 3;
  miso  = 1;
;
# -----
```



L'auteur

Diplômé d'Électronique Électrotechnique Automatique par l'université Paul Sabatier de Toulouse (1985), j'ai obtenu un doctorat en Signaux, Circuits, Systèmes à l'Université des Sciences et Techniques du Languedoc (1991). Depuis 1993, ma famille et moi sommes basés à Ouagadougou, où je travaille dans une association qui met par écrit les langues minoritaires, mène des programmes d'alphabétisation et traduit la Bible dans ces langues. Mes centres d'intérêt personnels sont surtout les microcontrôleurs et la programmation en C++. La pratique de l'électronique est réservée à mes loisirs.



Voici un exemple de ligne de commande d'utilisation d'AVRDUDE :

```
avrdude -P com7 -p m8 -c bob-scap -i 600
-U flash:w:test.hex:i
```

Il s'agit ici de sélectionner le programmeur BOB-SCAP grâce au paramètre `-c bob-scap` et de programmer la mémoire flash d'un ATmega8 (paramètre `-p m8`) avec le fichier hexa `test.hex` via BOB-SCAP relié à COM7 (paramètre `-P com7`). Le paramètre `delay` fixe l'horloge sur SCK à 600 µs avec `-i 600`.

La programmation avec BOB-SCAP est assez longue, mais on ne demande pas les mêmes performances à un programmeur d'appoint qu'à un programmeur de production.

Avant d'utiliser le BOB, n'oubliez pas d'y souder le cavalier JP1 qui détermine sa tension de service (5 V ou 3,3 V) ; il est situé entre le connecteur USB et la puce FTDI. Pour choisir le 5V, souder le contact central à celui situé côté connecteur USB. Je l'ai oublié deux fois ! Les BOB n'ont pas grillé mais ne jouez pas avec le feu !

(140254)

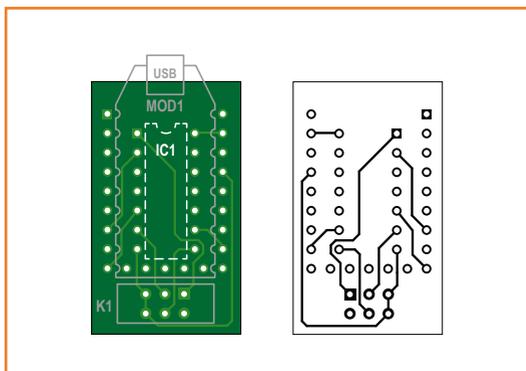


Figure 3. Ce porte-BOB permet d'obtenir une construction robuste à peine moins compacte que le BOB lui-même.

Liens

- [1] Passerelle USB/série BOB-FT232R 110553-91, Elektor sept. 2011 p. 48 www.elektor-magazine.com/fr/110553
- [2] SCAP - programmeur AVR simple, Elektor juillet 2011 www.elektor-magazine.com/fr/110087
- [3] Passerelle USB/série BOB-FT232R 110553-91 www.elektor.fr/ft232r-usb-serial-bridge-bob-110553-91

Après avoir magnifiquement expliqué **l'électronique aux débutants qui sèchent les cours**, dans son livre qui porte le même titre, **Rémy Mallard** revient avec un nouveau livre dans lequel il présente ...

les microcontrôleurs PIC pour les débutants

qui veulent programmer **sans patauger**

NOUVEAU



Ce livre initie à la programmation des microcontrôleurs PIC avec des applications pratiques qui vont bien au-delà du simple chenillard à LED et couvrent un grand nombre de besoins. Après une introduction (pas trop longue) aux principes essentiels de la programmation, tu apprends à interfacer des capteurs avec un microcontrôleur, à acquérir et stocker des données, ou encore à établir une liaison USB ou Ethernet pour transmettre ces données. Rémy regroupe par chapitres les informations théoriques et pratiques nécessaires à la réalisation de chaque montage décrit.

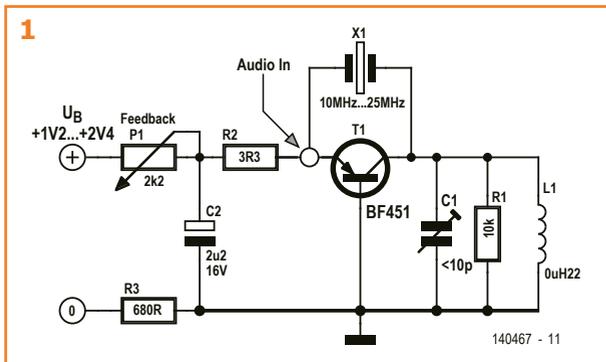
Après la lecture, gagné par la bonne humeur communicative de l'auteur et fort de sa longue expérience (qu'il partage volontiers), tu n'auras qu'une seule envie : aller plus loin, créer toi-même des montages encore plus ambitieux (transposer le code dans d'autres langages, le porter sur d'autres plates-formes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

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



ISBN 978-2-86661-193-4
48,50 €

émetteur synchro FM oscillateur Clapp



Hans-Norbert Gerbig
(Allemagne)

À un quartz. Le circuit de la **figure 1**, les Allemands l'appellent *oscillateur capacitif à trois points*. Effectivement, les condensateurs sont disposés en triangle, mais par ici, nous avons l'habitude de lui attribuer le nom de son inventeur : James Clapp. Ce circuit-ci est en plus stabilisé par un quartz. On peut le synchroniser dans une gamme spécifique de signaux à basse fréquence (BF). Dans ce domaine, il participe à tous les changements de fréquence du signal pilote. On peut élargir le domaine de sortie en ajoutant une résistance en parallèle au circuit oscillant. L'oscillateur travaille sur sa propre fréquence et son amplitude reste constante. Le domaine dans lequel évolue la fréquence est l'image fidèle du

Facile de construire un émetteur à fréquence modulée de petite puissance pour tester des récepteurs FM. Encore mieux quand il est bon marché, simple et pourtant très stable ; pas difficile : il est asservi

signal BF, la déviation à gauche et à droite est pratiquement exempte de toute modulation d'amplitude parasite. Le résultat est une modulation de fréquence !

La basse fréquence qui module est couplée au circuit de base à très faible impédance, 3,3 à 4,7 Ω , sur la résistance d'émetteur. La valeur de cette résistance est importante pour la préaccentuation du spectre aigu à l'émission. Lors de la démodulation dans un récepteur FM normal, les fréquences élevées du spectre audio sont atténuées de manière à ce que le son soit rééquilibré jusqu'à 15 kHz et les parasites réduits.

L'oscillateur n'a forcément qu'une très faible puissance HF, elle se chiffre en picowatts. Le circuit de base est très stable, ce qui le rend peu sensible à l'effet de main : un blindage n'est pas nécessaire. Un récepteur FM situé à proximité immédiate peut sans difficulté capter le faible rayonnement des harmoniques modulées en fréquence de l'oscillation du quartz.

D'ailleurs, on peut fort bien remplacer sans plus le transistor HF PNP par un NPN, HF lui aussi, il n'y a qu'à inverser les polarités de C2 et de l'alimentation. Pour la stabilité de l'oscillation, il faut régler P1, mais aussi veiller à ce que C1 soit adapté à L1 et X1.

(140467 - version française : Robert Grignard)

récepteur synchro FM à oscillateur Clapp

Hans-Norbert Gerbig
(Allemagne)

À l'évocation d'un récepteur FM, vous pensez probablement double superhétérodyne et démodulateur complexe. On peut faire plus simple et plus amusant. Avec un oscillateur Clapp en montage à base commune, vous avez le récepteur et le discriminateur.

Le principe de l'oscillateur utilisé dans l'émetteur synchro FM décrit ci-dessus va nous servir ici à construire le récepteur qui, s'il semble plus compliqué, ne l'est pas réellement. Au fond, c'est un oscillateur à entraînement et réaction. Il occupe la partie gauche du schéma de la **figure 1**, centré

sur T1. À droite, on a simplement l'amplification BF et la désaccentuation.

Démodulateur synchro

Un oscillateur sinusoïdal peut subir un effet d'entraînement par une fréquence incidente située dans un domaine de part et d'autre de sa fréquence propre et se synchroniser sur elle. Une fois accroché, l'oscillateur suit toutes les variations de fréquence, BF comme HF, du signal extérieur.

L'oscillateur est un montage en base commune du transistor PNP T1, avec C2 qui fournit ici la réaction en HF. Sa fréquence propre est déterminée par l'inductance L1 et la diode à capacité variable D1 dont la valeur est ajustée par le potentiomètre P1. Le domaine d'entraînement est d'autant plus large et donc la résolution plus étroite que le circuit résonant est amorti. L'amortissement dépend de la résistance R2 en parallèle sur le circuit oscillant ; il permet d'élargir la bande de fréquences accessible. En l'absence de R2, le récepteur est donc très sélectif.

Propriétés

La caractéristique essentielle de ce circuit provient de sa nature d'oscillateur entraîné : sa tension de sortie est très constante et les parasites dominants sont atténués. En outre, l'oscillateur ne se

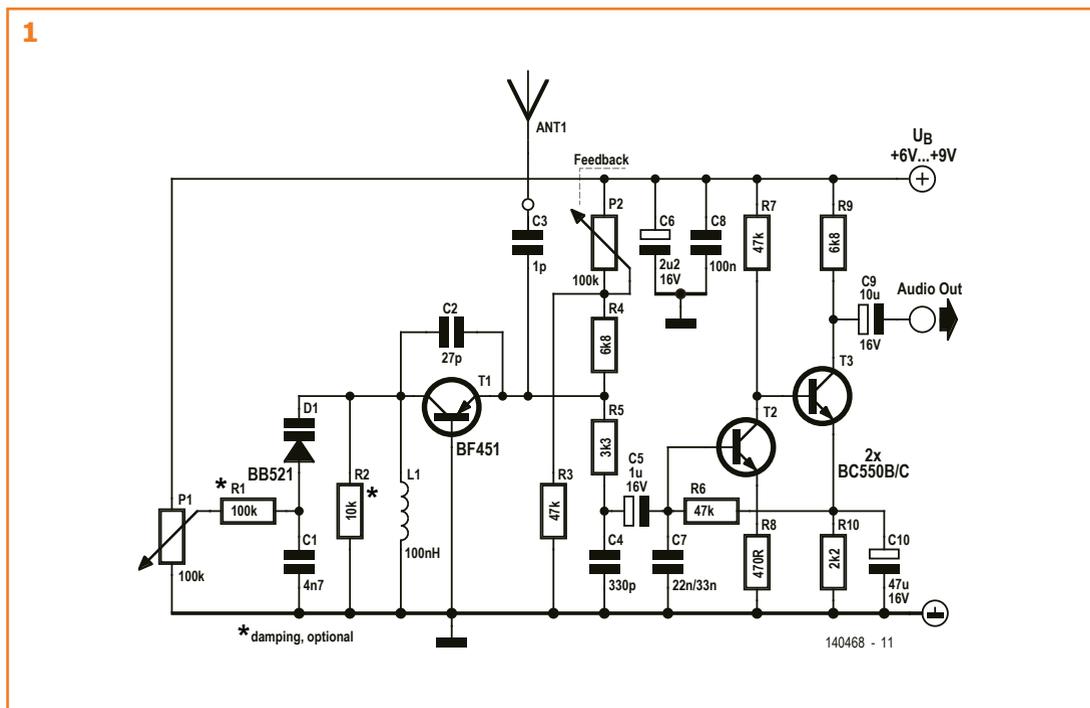
synchronise que sur les fréquences des émetteurs puissants. D'autres fréquences de réception de plus faible amplitude, y compris sur le même canal, ne peuvent pas l'entraîner à leur suite et n'ont ainsi aucune influence sur le démodulateur. Des émetteurs plus faibles, même avec une force de champ de 70 % de la station souhaitée, sont totalement éliminés. C'est la raison de son exceptionnelle sélectivité.

Les fluctuations audio proportionnelles sont prélevées sur l'émetteur du transistor HF T1. La résistance R5 et les condensateurs C4 et C7 filtrent en passe-bas les résidus de HF. Mais C7 a encore une autre fonction : compenser la préaccentuation systématique du spectre aigu du signal audio sur les émetteurs FM.

Les transistors T2 et T3 constituent un amplificateur simple du signal audio. Vous pouvez ajouter en sortie un petit étage final à circuit intégré ou brancher directement un casque d'écoute à haute impédance ou une oreillette.

Le montage en base commune, comme sur l'émetteur Clapp, est peu sensible à la réaction et donc à l'effet de main : lui non plus n'a besoin d'aucun blindage.

(140468 - version française : Robert Grignard)



pilote de LED riqui pour lampes de poche

remplacer l'ampoule par une LED



Peter Krüger
(Allemagne)

Que l'ampoule d'une petite lampe de poche soit alimentée par deux piles de 1,5 V ou une seule n'y change rien, la tension délivrée reste insuffisante pour alimenter une LED blanche. Pour cela il faut un convertisseur élévateur. Tout riquiqui. Ça tombe bien, outre sa simplicité, c'est l'une des qualités de celui proposé ici.



Figure 1.
Le convertisseur élévateur et la LED nichés dans un culot à vis E10. La petite pile bouton de type AG5 utilisée ici (1,5 V et 60 mAh) peut alimenter la LED durant une heure.

Je me suis demandé comment transformer en lampe torche à LED une mini-lampe de poche à ampoule. Le rendement d'une LED est bien meilleur que celui d'une ampoule, et à intensité rayonnée égale une pile alimentant une LED aura donc une durée de vie plus longue. Le courant consommé par une LED blanche impose toutefois une alimentation de plus de 3 V. Il existe bien des solutions toutes prêtes, mais on se heurte vite à des problèmes d'encombrement lorsqu'on cherche à les utiliser. J'ai donc décidé de construire moi-même ce dont j'avais besoin. Après tout, un seul transistor suffit pour construire un convertisseur élévateur auto-oscillant. Le circuit final est pour

le moins dépouillé, et si petit que toute l'électronique, LED comprise, tient dans le culot à vis d'une ampoule ordinaire (**fig. 1**). Le résultat est un substitut d'ampoule bien pratique.

Convertisseur élévateur

Le convertisseur élévateur (**fig. 2**) est réduit au minimum : un transistor, une résistance, deux condensateurs et une indispensable bobine. La prise intermédiaire de cette bobine est reliée à deux enroulements, AB pour la réaction positive, BC pour le stockage de l'énergie. Comme la tension d'alimentation est appliquée au point B, les signaux des deux enroulements sont déphasés de 180°. Le circuit oscille en raison de la réaction positive conjuguée au déphasage de 180° apporté par le transistor.

Plus précisément : à la mise sous tension le courant traverse l'enroulement AB, puis R1 et C2 parallèle, et enfin la base de T1, qui devient passant. Par suite le courant qui circule de B vers C gagne le collecteur de T1, ce qui a pour effet d'amener le potentiel du point A à une valeur supérieure à $+U_B$. T1 est alors totalement saturé via C2. Le courant qui circule entre B et C augmente alors jusqu'à ce que le noyau soit saturé, ce qui prend 2 à 3 μ s avec le dimensionnement de ce circuit. À partir de là le courant n'augmente



plus, ce qui fait baisser le potentiel du point A. T1 passe à l'état bloqué en quelques nanosecondes, et l'énergie magnétique stockée dans le noyau de la bobine se décharge en à peine 2 μ s à travers la LED blanche. Ensuite le cycle reprend.

Avec les composants utilisés, la fréquence d'oscillation vaut entre 200 et 250 kHz. Le circuit oscille jusqu'à ce que $+U_B$ chute sous la tension de seuil de 0,55 V de la jonction BE de T1. Le circuit est configuré pour que le courant qui traverse la LED, et donc la luminosité de celle-ci, diminue en même temps que la tension de la pile – comme avec une ampoule.

Montage

Un noyau de bobine relativement petit convient, car le pilote de LED n'a que 75 mW env. à fournir. Une petite perle de ferrite est donc amplement suffisante. La miniaturisation imposait l'utilisation de composants CMS pour T1, R1, C1 et C2. L'ensemble de l'électronique niche ainsi parfaitement dans le culot à vis E10 d'une ampoule pour lampe de poche. Le montage nécessite des pincettes et une dextérité de brodeuse, car les composants se câblent en l'air (**fig. 3**). Il y a plusieurs petites choses à noter afin d'obtenir un assemblage semblable à celui de la photo. L'élément central est la bobine L1, dont vous ne trouverez hélas aucun modèle préfabriqué. Il vous faudra donc la bobiner vous-même. L'enroulement BC est constitué de 14 spires, suivies de 5 spires pour former l'enroulement AB. Le diamètre du trou de la perle de ferrite ne fait que 1,3 mm, et il faut y passer du fil de cuivre émaillé de seulement 0,15 mm ; patience et pincettes sont de rigueur. Étudiez le plan de bobinage (**fig. 4**) ; les points correspondent à ceux du schéma (fig. 2). Passez d'abord l'enroulement de 14 spires, puis celui de 5 spires. Ensuite, torsadez et soudez les deux connexions pour réaliser la prise intermédiaire (point B sur la figure 2). La **figure 5** rend bien compte de la petitesse de la bobine.

Le schéma de câblage (**fig. 6**) vous aidera à obtenir le même assemblage que celui de la photo. J'ai inséré la LED dans une rondelle en plastique

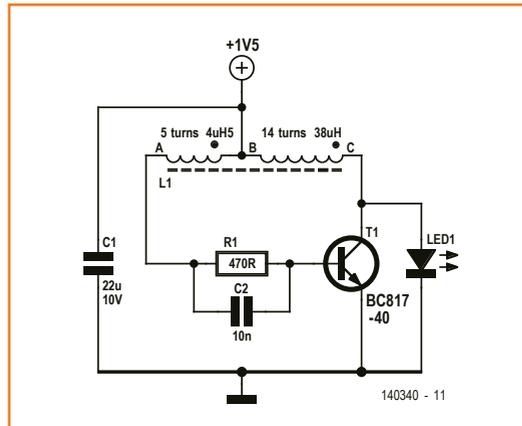


Figure 2. Schéma du convertisseur élévateur : un transistor, une bobine, une résistance, deux condensateurs, et rien de plus !

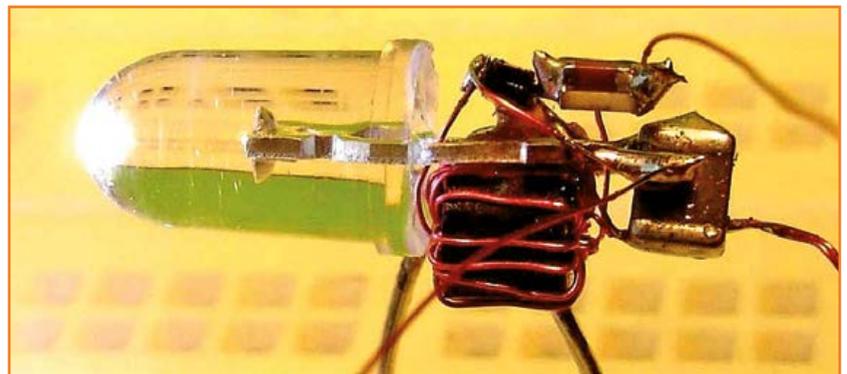


Figure 3. Pas de carte ici, le câblage se fait en l'air.

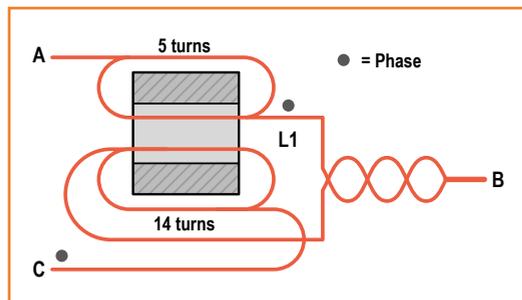


Figure 4. Schéma d'enroulement de la bobine L1. Les points sont ceux de la fig. 2.



Figure 5. La bobine de l'auteur. Vraiment riquiqui.

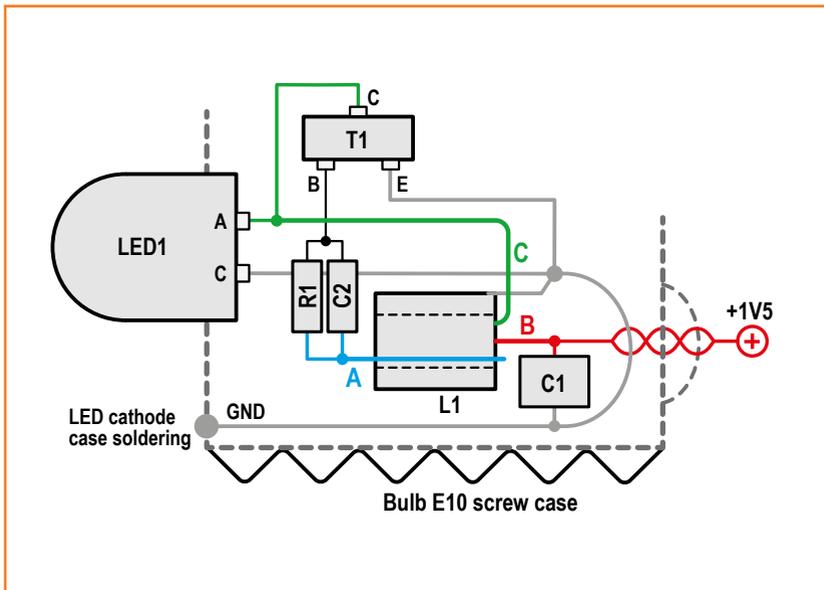


Figure 6. Schéma de câblage de la LED et du pilote.

rouge de diamètre interne de 5 mm, puis soudé et collé l'ensemble sur le culot à vis E10, apprêté séparément.

Une pile et c'est dans la poche

Le montage convient aux petites torches ou lampes de poche alimentées par une petite pile bouton de 1,5 V (comme celle de la figure 1), ou bien par une ou deux piles/accumulateurs AA ou AAA. Une pile ordinaire AA de 2000 mAh fournit une bonne quarantaine d'heures d'éclairage. Si la lampe fonctionne avec deux piles, l'une d'elles devra être remplacée par une pile factice, car le pilote ne doit pas être alimenté par plus de 1,5 V = 1 pile ! Pour cela, coupez un bâton de colle

Liste des composants

Résistance

R1 = 470 Ω, 0805

Condensateurs

C1 = 22 μF / 10 V, multicouche, 1206

C2 = 10 nF / 25 V, multicouche, 0805

Inductance

L1 = perle de ferrite, enrobée, longueur 3 mm, ø 3,5 mm, trou 1,3 mm, p. ex. Farnell 2643001501

Semi-conducteurs

LED1 = LED, blanche, 5 mm, p. ex. Nichia

NSPW500DS b2W

T1 = BC817-40, SOT23

Divers

Fil de cuivre émaillé 0,15 mm

Ampoule et culot E10 de récupération

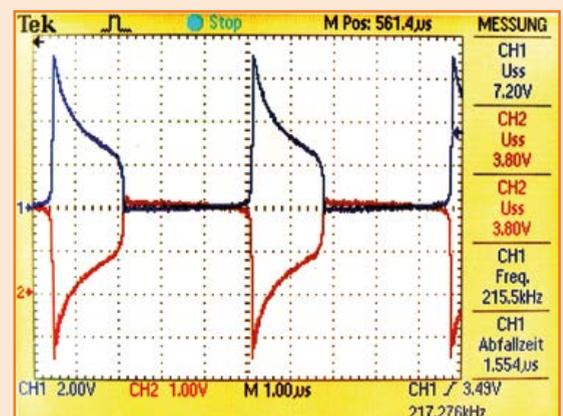
chaude de même dimension que la pile, puis percez-le dans le sens de la longueur. Passez un morceau de fil de cuivre dans le trou, soudez deux plaquettes métalliques aux extrémités, et voilà, votre pseudo-pile prête.

(140340 - version française : Hervé Moreau)

Oscillogramme

La copie d'écran ci-contre montre le comportement de la tension en T1. La tension de collecteur (CH1, en bleu) augmente jusqu'à 7,2 V au moment de la commutation. Sur la base (CH2, en rouge) se développe un pic de tension négative d'environ -3,3 V (= $U_{CC} - U_{BE}$). Cette tension provoque une commutation rapide et reste dans les caractéristiques du transistor.

La tension de collecteur élevée étonne de prime abord. On la comprend mieux si l'on réalise que les connexions de la LED présentent encore de petites valeurs d'inductance (la tension aux bornes de la puce-LED est inférieure) et que la pointe de courant est nettement supérieure au courant admissible de 25 mA. C'est sans danger pour la LED, car à une fréquence de commutation élevée, la charge reste toujours acceptable grâce à l'intégration thermique.



amplificateur de signal pour oscilloscope USB

Mon oscilloscope USB *PicoScope* est doté d'un (fantastique) générateur de signaux de $2 V_{CC}$ maximum.

La plupart du temps, ce maximum me suffit mais pas toujours. En plus j'aime bien régler le zéro et l'intensité du signal à l'aide d'un potentiomètre. J'ai cherché en vain une solution toute prête, alors j'ai décidé que mon fer à souder s'en mêlerait.

Mon circuit comprend trois parties. Celle construite autour d'IC1.A sert au réglage du zéro et l'interrupteur S1 permet d'activer le décalage (*Offset*). La tension de ce décalage se règle avec P1. La partie construite autour d'IC1.B déphase de 180° le signal du générateur de signaux, et P2 sert à régler son amplitude. La troisième partie assemblée autour d'IC2 est l'étage amplificateur, câblé en tant qu'additionneur. La tension de décalage y est amplifiée environ trois fois ($R8/R6$) ; le signal du générateur de fonctions délivré par IC1B est quant à lui amplifié environ huit fois ($R8/R7$). Au final il est possible de régler l'amplitude et le décalage du signal de sortie entre

$-12 V$ et $+12 V$. La plage de fréquence va du signal continu à environ $1 MHz$.

J'ai câblé deux connecteurs BNC en parallèle afin qu'une entrée d'oscilloscope puisse également être reliée au circuit à tester.

La tension est délivrée par un module Traco de $\pm 15 V$. Je l'ai utilisé comme solution de facilité, parce que j'en avais un. Il est assez cher, mais pratique. Sinon vous pouvez assembler une petite alimentation avec un transfo secteur, un pont redresseur, un condensateur électrolytique tampon et deux régulateurs de tension. Le circuit consomme moins de $100 mA$.

(140346 - version française : Hervé Moreau)

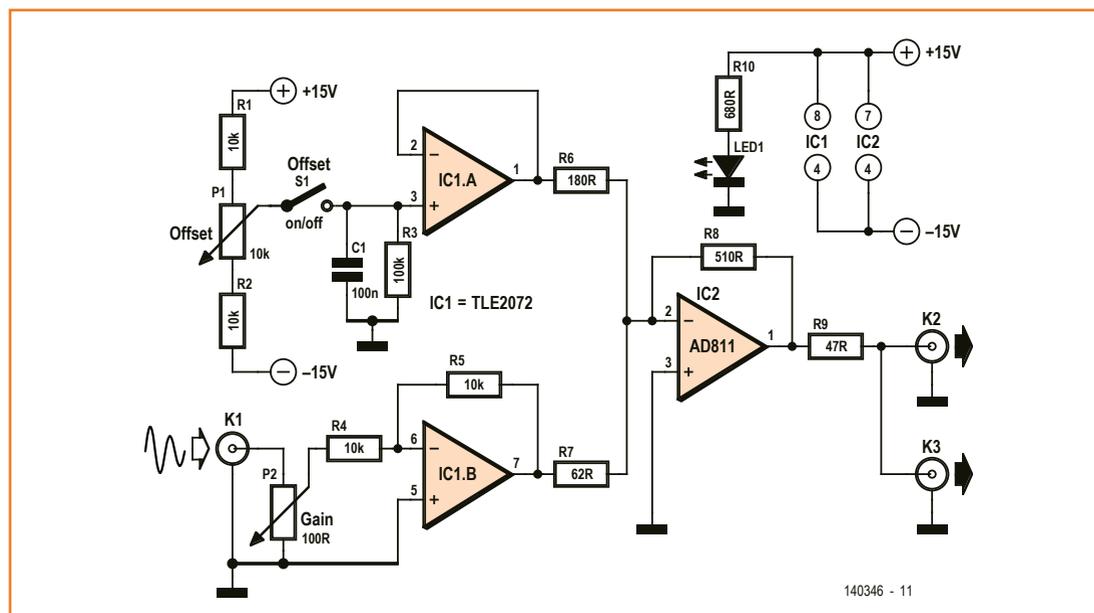


Figure 1.
Schéma de l'amplificateur
de signal avec réglage du
décalage.

lampes à incandescence de l'art de planter les bulbes

Peter E. Tiefenthaler
(Allemagne)

Les lampes à incandescence sont assurément une espèce en voie de disparition, mais si on les considère comme des « fusibles avec limitation automatique du courant et témoin de surcharge intégré », elles pourraient survivre encore quelques années à la pression de sélection eurocratique !



Les lampes à incandescence peuvent faire plus qu'éclairer. Vues comme limitatrices de courant ou de puissance, elles offrent dans de nombreuses situations des avantages que n'ont pas les fusibles « normaux » : alors qu'un fusible voit sa résistance passer brutalement de zéro à l'infini lorsque le courant atteint une certaine intensité, une ampoule peut limiter en douceur la montée d'un courant.

Sa luminosité peut même servir à détecter une surintensité. Elle peut elle aussi griller en cas de surcharge et, surtout, elle est peu chère et facile à manier.

Plus qu'une bête ampoule

Le filament métallique d'une ampoule se comporte comme une thermistance CTP : il offre d'abord une faible résistance à froid (à basse tension), puis une résistance de plus en plus grande à mesure que le courant croît et que le filament s'échauffe. Cet effet a été p. ex. exploité dans les oscillateurs à pont de Wien pour stabiliser l'amplitude.

La résistance à froid d'une ampoule classique de 60 W est d'environ 70 Ω . Un courant d'environ 260 mA la traverse lorsqu'elle brille sous 230 V, ce qui correspond à une résistance à chaud de quelques 900 Ω . Pour les calculs, on peut donc estimer en première approximation que le rapport entre résistance à chaud et résistance à froid vaut 10. La **figure 1** montre la caractéristique résistance-tension d'une ampoule de phare automobile de 12 V. La relation exacte entre V et R dépend du type de l'ampoule.

Les lampes à filament de carbone se comportaient quant à elles comme des CTP.

Limitation de courant et de puissance

Utiliser une lampe à incandescence comme fusible n'a rien de révolutionnaire. L'astuce est simplement tombée dans l'oubli, même parmi les électroniciens expérimentés. À l'époque des radios à lampes, un cordon secteur doté d'une ampoule de 60 W protectrice était pourtant un outil de réparation irremplaçable (**fig. 2**). Ce système fait ses preuves aujourd'hui encore lorsqu'il s'agit de vérifier un appareil dans un état inconnu, de le mettre précautionneusement sous tension et/ou de reformer ses condensateurs électrolytiques. De façon presque standard, le circuit d'alimentation de nombreux lampemètres américains comporte une résistance-talon et une ampoule servant de « fusible témoin » (**fig. 3**). Les effets des fluctuations de tension restent ainsi sous contrôle. Une autre ampoule sert de fusible lors de la polarisation.

Les lampes à filament de certains amplificateurs à tubes ont la même fonction. Leur résistance limite toute montée inadmissible du courant de la grille écran, et elles tiennent lieu de vumètre en s'allumant. L'alimentation du courant de plaque de la plupart des amplificateurs à tubes est dénuée de protection par fusible. Ici aussi on peut ajouter des lampes à filaments comme moyen de protection et de limitation du courant. Même les résistances fusibles, qu'une certaine température rend défectueuses, devraient si nécessaire être remplacées par une lampe à filament plutôt que par une résistance normale.

On peut utiliser l'ampoule d'une lampe de poche de p. ex. 2,5 V / 150 à 200 mA lorsque les chutes de tension en jeu sont faibles ; si elles sont fortes, on peut employer une ampoule de cadran ou d'automobile. Rien n'empêche d'en câbler plusieurs en série ou en parallèle. Sur la **figure 4**, on peut les substituer à la résistance, ou les placer en A et/ou B.

Dans un chargeur de batterie au plomb, une lampe à filament peut stabiliser le courant de charge dans certaines limites (**fig. 5**).

Il va de soi qu'une lampe à filament offre un moyen simple d'abaisser une tension d'alimentation trop haute. Une lampe bien choisie peut p. ex. servir de résistance-talon à un appareil alimenté en 110 V à partir du réseau de 230 V.

Haut-parleurs

Rempart peu coûteux et facile à installer, une lampe à filament placée à la sortie d'un ampli-

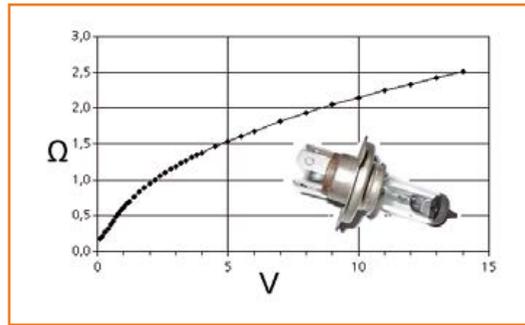


Figure 1. Caractéristique résistance-tension d'une ampoule halogène de phare d'automobile.

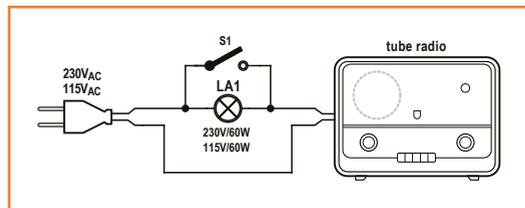


Figure 2. Une ampoule dans le cordon secteur évitait d'endommager les appareils sous test.

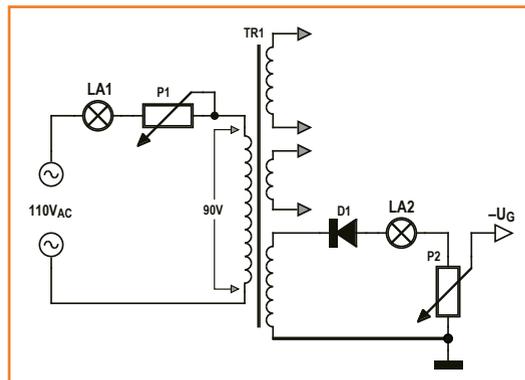


Figure 3. Utilisation type des lampes à filament dans les appareils à tubes US.

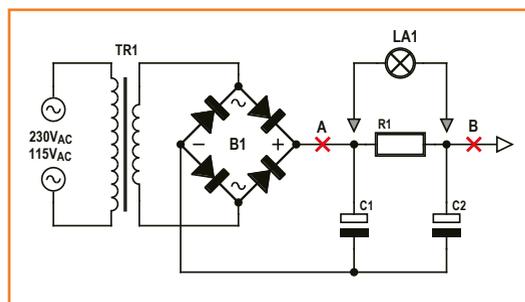


Figure 4. Des lampes pour limiter le courant dans un amplificateur.

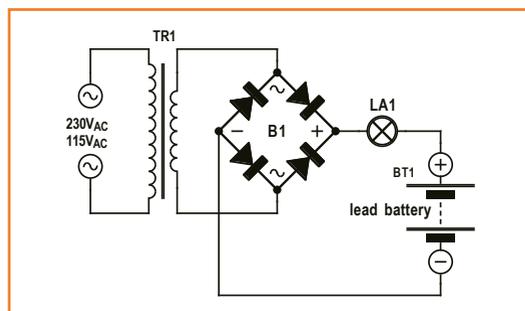


Figure 5. Une lampe pour limiter le courant lors de la charge d'une batterie.

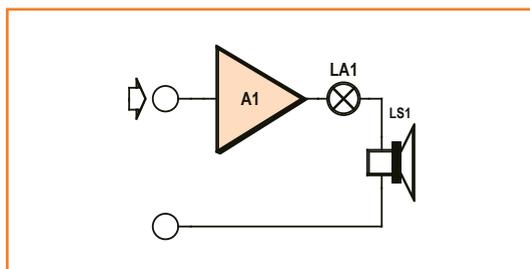


Figure 6.
La lampe à filament protège le haut-parleur.

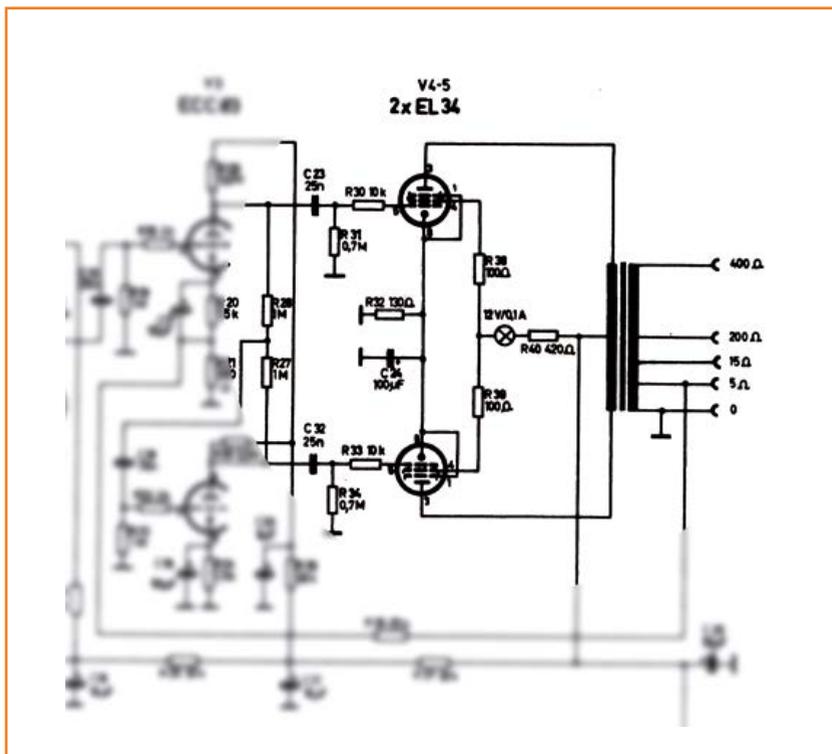


Figure 7.
Extrait du schéma d'origine d'un amplificateur à tubes (RIM Gigant). Ici aussi une lampe à filament sert à limiter la puissance.

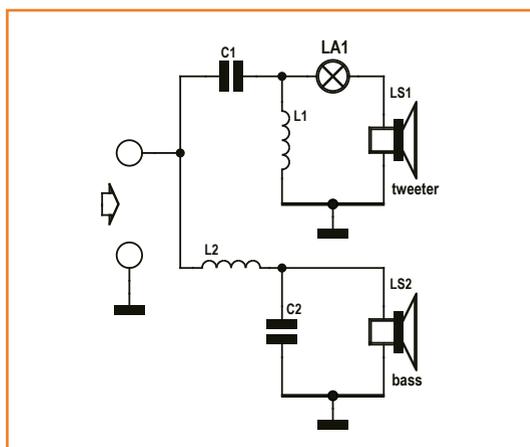


Figure 8.
Le sensible haut-parleur d'aigus protégé par une lampe.

ificateur protège efficacement un haut-parleur contre les surcharges (fig. 6). Et elle préservera en même temps la quiétude des foyers dans lesquels un rejeton de la génération numérique aime célébrer son groupe favori en amenant le bouton de volume sur la butée droite.

La résistance à froid d'une lampe à filament est négligeable à faible volume car les puissances en jeu restent alors cantonnées dans le domaine des milliwatts. Lorsque la tension de sortie augmente, la lampe absorbe de plus en plus de puissance et limite ainsi la montée du niveau sonore. En outre la résistance de charge de l'amplificateur augmente elle aussi, ce qui limite encore la puissance de sortie. Cette résistance prédomine d'autant plus que la puissance est grande, ce qui conduit à une certaine compression et limitation du signal lorsque le niveau sonore est élevé.

Les lampes placées sur le trajet du signal ne sont pas audibles. On ne les perçoit qu'aux niveaux sonores élevés. C'est une bonne chose, car c'est le signe que la protection agit. Perdre quelques notes en route coûte moins cher qu'acheter un nouveau haut-parleur.

Le schéma historique de la figure 7 montre que les lampes à filament servaient déjà (même en série) de limiteurs de puissance dans les anciens amplificateurs à tubes.

Dans les systèmes de sonorisation, le recours aux lampes pour protéger les sensibles haut-parleurs d'aigus a depuis longtemps fait ses preuves (fig. 8). Les haut-parleurs utilisés pour des tests sur une paille sont eux aussi protégés de cette façon. La règle approximative énoncée ci-dessus ($R_{chaud}/R_{froid} = 10/1$) peut aider à choisir la bonne lampe. Une lampe navette de 12 V / 5 W peut limiter de façon fiable le niveau sonore de divers haut-parleurs. Une ampoule de 6 V / 5 W sera sans doute un meilleur choix pour un système Hi-Fi.

À partir d'un certain volume, les guitares électriques et leurs amplis se mettent souvent à « chanter », effet qui ne déclenche que rarement des applaudissements nourris chez les voisins. Ici le guitariste aura intérêt à utiliser un atténuateur de puissance, un circuit dont les résistances de charge, inductances et condensateurs reproduisent l'impédance d'un haut-parleur et consomment l'essentiel de la puissance amplifiée. La puissance laissée au haut-parleur suffit à

renvoyer un son « haut et clair », mais doux. Le son qui subit l'effet de compression mentionné ci-dessus est meilleur et plus réaliste si les résistances de puissance de l'atténuateur sont remplacées par des lampes à filament correctement dimensionnées (LA2 dans la **fig. 9**).

L'effet tube

Si arguer que les lampes à filaments sont aussi creuses et vides que les tubes n'est pas une raison suffisante pour justifier leur emploi, il n'empêche qu'elles permettent d'obtenir facilement l'équivalent à tubes d'un amplificateur à transistors. On peut commencer par placer une lampe dans un circuit d'alimentation pour reproduire la tension plus douce d'un tube redresseur (fig. 4). Dans un amplificateur à tubes, la chute de tension peut par ailleurs être compensée par une lampe lorsque le redresseur à tubes est remplacé par un redresseur à semi-conducteurs.

Pour obtenir un effet tube dans un amplificateur à transistors, la lampe doit être placée entre la sortie de l'étage amplificateur et le haut-parleur. L'effet tube est accentué lorsque les résistances d'émetteur sont remplacées par des petites lampes (**fig. 10**). La limitation de l'amplificateur est plus douce, et le son est (presque) celui d'un appareil à tubes.

(140188 - version française : Hervé Moreau)

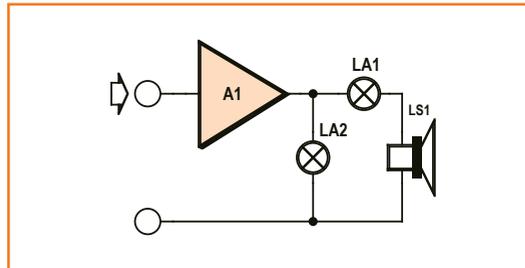


Figure 9. LA2 utilisée comme atténuateur de puissance.

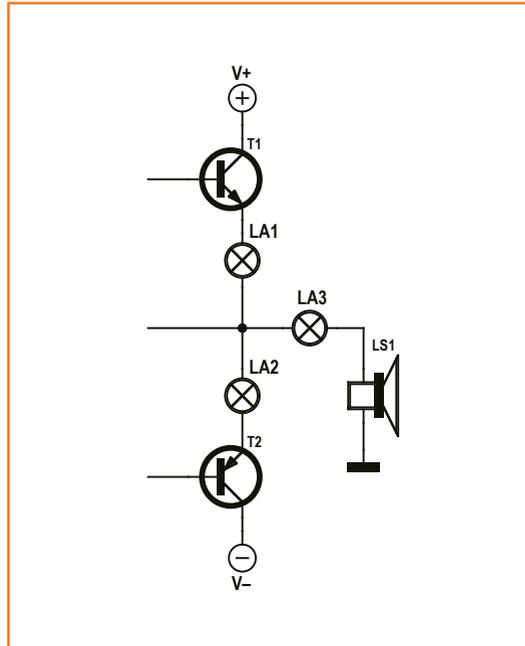


Figure 10. Amplificateur à transistors « tubifié ».

Publicité

Une qualité professionnelle à prix discount!

**Vous trouverez plus de
400 ampoules LED**

sur www.reichelt.fr

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

reicht.fr
elektronik

GB 30588
3,60

2 W / 170 lm
A+_{EEK}

gobay®

Plus de 45 ans d'expérience
Envoi en 24 heures
Plus de 50 000 produits
+49 (0)4422 955-360

générateur de sinus à filtre RII code compact pour petits microcontrôleurs

Rüdiger Knörig
(Allemagne)

Il est relativement facile de créer un générateur numérique de signaux sinusoïdaux sans employer ni table de sinus ni interpolations complexes. La rétroaction d'un filtre numérique suffit en effet à faire « osciller » comme il faut un générateur. Bien choisi, ce filtre récursif permet en outre de s'affranchir de quelques chinoïseries mathématiques et de nicher le code associé dans un petit microcontrôleur.

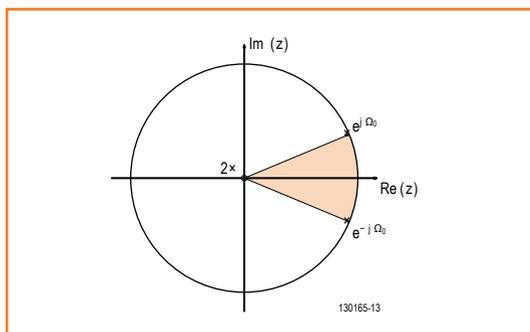


Figure 1.
Représentation graphique des pôles et zéros du filtre sinusoïdal.

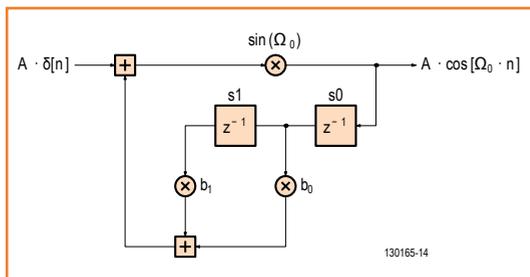


Figure 2.
Filtre RII sinusoïdal d'ordre 2.

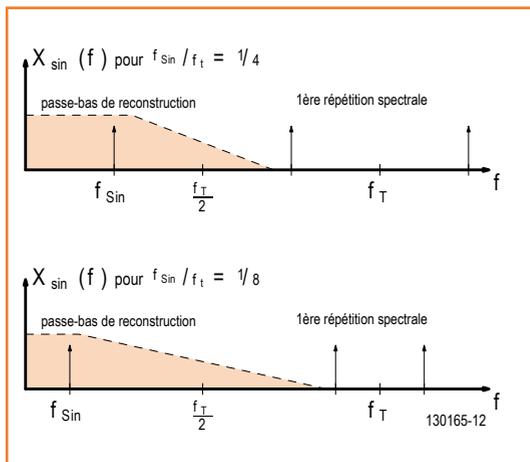


Figure 3.
Exigences pour le passe-bas de reconstruction en fonction de $\Omega_0 = 2\pi \cdot \frac{f_{sin}}{f_T}$.

La voie habituelle pour produire un signal sinusoïdal à l'aide d'un microcontrôleur est de passer par un tableau de valeurs représentant la fonction sinus. Avec cette méthode, dite synthèse numérique directe (DDS, pour **D**irect **D**igital **S**ynthesizer), on peut calculer des signaux de différentes fréquences par interpolation des valeurs stockées dans le tableau. Une telle table s'avère toutefois vite encombrante pour les 1024 octets d'un petit contrôleur, voire impose l'utilisation d'une mémoire plus lente.

Ma méthode n'utilise aucune table et, dans le cas le plus simple, ne nécessite que deux registres comme mémoire d'état. Comme la fréquence de la fonction sinus ne dépend que d'un seul coefficient, on peut même, pour peu que la multiplication (en virgule flottante) soit suffisamment précise, modifier la fréquence lors de l'exécution du code.

Le principe exposé en [1] concerne les processeurs de signaux mais peut être adapté aux petits microcontrôleurs. Vous pouvez également consulter les liens [2] et [3] si vous souhaitez approfondir le sujet puisqu'ici je ne ferai qu'effleurer la théorie.

Théorie et application

Les fondements mathématiques de la méthode sont exposés dans le fichier *IIRsinus.pdf* [4]. Tout y est détaillé, donc consultez-le si vous voulez savoir de quelles cornues mathématiques proviennent les résultats ci-dessous. Voici la version courte :

Un signal sinusoïdal continu est converti en signal (temporel) discret. Un filtre numérique qui possède ce signal discret comme réponse

impulsionnelle est ensuite cherché via plusieurs étapes de calcul. Le filtre obtenu possède un zéro et deux pôles. La présence d'un zéro supplémentaire à l'origine rend le filtre causal. La **figure 1** montre la position des zéros et des pôles déduite de l'équation résultante. Cette solution nécessite une grande précision numérique.

Le résultat est un filtre purement récursif (ou RII, à réponse impulsionnelle infinie) d'ordre 2 (**fig. 2**). La forme normalisée de l'équation du filtre permet de déduire les constantes $b_0 = 2\cos(\Omega_{sin})$ et $b_1 = (-1)$.

En négligeant l'atténuation, on peut calculer la réponse impulsionnelle du filtre à l'aide du pseudo-code suivant :

Require:

$z0 \leftarrow \alpha$

$z1 \leftarrow 0$

loop

$y \leftarrow z0 \cdot b0 - z1$

$z1 \leftarrow z0$

$z0 \leftarrow y$

end loop

Le calcul ne nécessite que deux registres, une multiplication, une soustraction, et deux registres pour les opérations de copie. L'utilisation d'un simple microcontrôleur à 8 bits demande un peu plus de réflexion. Deux solutions rapides sont possibles : effectuer la multiplication avec des opérateurs de décalage, ou bien la faire avec un tableau de 256 entrées, tableau qui permet par ailleurs d'économiser l'espace mémoire lorsque la période du sinus est plus grande que 256. Traduit en C++, cela donne :

```

1  int8_t y,z0 ,z1;
2  z0 = alpha ; z1 = 0;
3
4  for ( int n=0;n< Nvalues ;n ++)
5  {
6      y = (z0 << 1);
7      y -= (z0 >> Nshift );
8      y -= z1;
9      z1 = z0;
10     z0 = y;
11     result << ( int ) y << endl;
12 }
    
```

Pour identifier les combinaisons utilisables de la valeur initiale a et de la constante N_{shift} (shift =

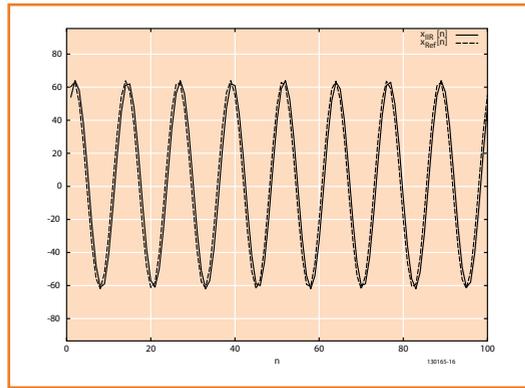


Figure 4.
 $N_{Shift} = 2, \alpha = 61 ; f_{sin}/f_T$
 attendu : 12,43,
 mesuré : 12,42.

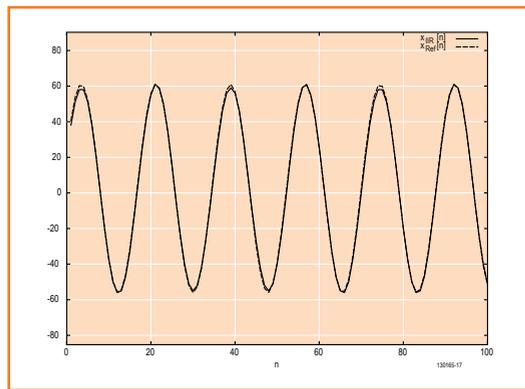


Figure 5.
 $N_{Shift} = 3, \alpha = 40 ; f_{sin}/f_T$
 attendu : 17,68,
 mesuré : 17,75.

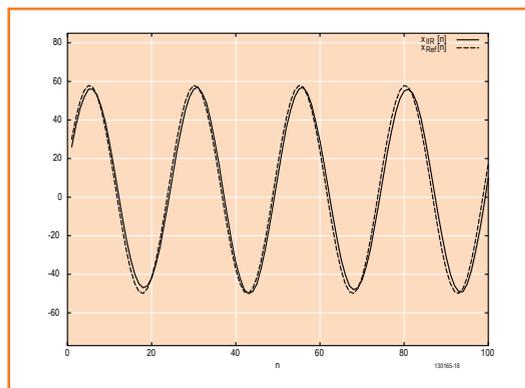


Figure 6.
 $N_{Shift} = 4, \alpha = 26 ; f_{sin}/f_T$
 attendu : 25,07,
 mesuré : 25.

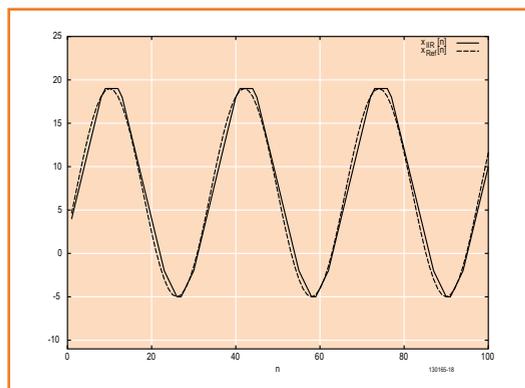


Figure 7.
 $N_{Shift} = 5, \alpha = 4 ; f_{sin}/f_T$
 attendu : 35,05,
 mesuré : 32.

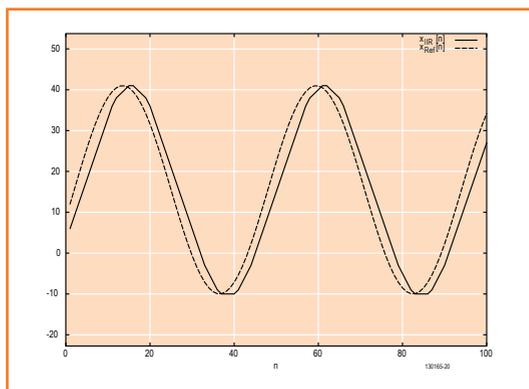


Figure 8.
 $N_{Shift} = 6, \alpha = 6 ; f_{sin}/f_T$
 attendu : 50,23,
 mesuré : 46.

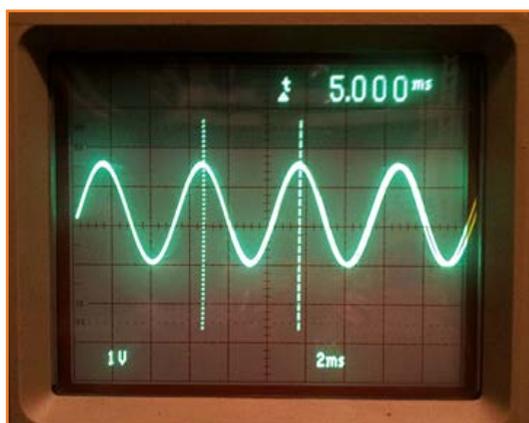


Figure 9.
 Oscillogramme du sinus.

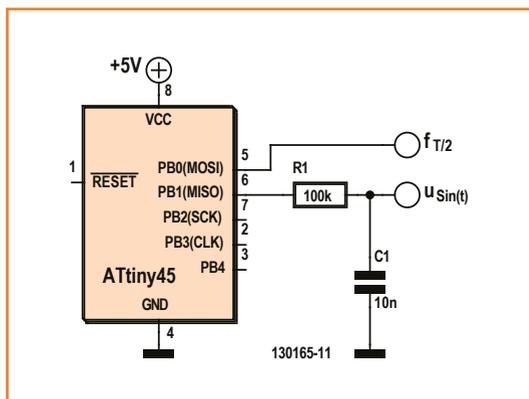


Figure 10.
 Schéma du générateur
 de sinus à filtre RII avec
 ATtiny45.

décalage) ainsi que les caractéristiques du signal produit, j’ai calculé avec ce code 102400 valeurs de sortie et les ai analysées avec un script Octave (un logiciel libre comparable à MATLAB). Les résultats possédant les amplitudes les plus grandes sont regroupés dans le **tableau 1**. Les signaux correspondants sont représentés sur les figures **4 à 8**.

Implantation dans un μC à 8 bits

Ces études préalables effectuées, j’ai implanté le code dans un ATtiny45 avec $N_{Shift} = 4$ et $a = 26$ (tableau 1). Le microcontrôleur est cadencé par l’horloge interne (8 MHz avec le fusible CKDIV8 activé, soit 1 MHz réel).

Le Timer0 est configuré en mode CTC (voir le code source [4]) pour qu’une nouvelle valeur de sinus soit produite à une fréquence $f_T = f_{CPU}/200$, puis envoyée sur la sortie OC1A (PB1) avec la MLI rapide de 64 MHz de Timer1. Pour contrôler la fréquence d’échantillonnage, le code bascule le niveau de la broche PB0 à chaque valeur de sinus créée.

J’ai mesuré 200 μs comme écart entre deux fronts, soit une fréquence f_T de 5 kHz. L’oscilloscope a confirmé la fréquence du sinus ainsi produit (**fig. 9**). D’après le tableau 1, $f_{Sin} = f_T/25 = 200$ Hz, soit une période de 5 ms. Pour supprimer les répétitions spectrales parasites ainsi que le signal porteur de la MLI, il m’a suffi d’improviser un filtre passe-bas avec une fréquence de coupure $f_c = 1$ kHz ($R1 = 100$ k Ω , $C1 = 10$ nF, cf. **fig. 10**).

(130165 – version française : Hervé Moreau)

Bibliographie et liens

- [1] N. S. Jayant, Peter Noll, *Digital Coding of Waveforms*, Prentice Hall, 1984.
- [2] G. Cardamone, L. Lo Presti, *A direct digital frequency synthesizer using an iir filter implemented with a dsp microprocessor*, vol. 6, p. 201, IEEE International Conference on Acoustics, Speech, and Signal Processing, 1994.
- [3] Thomas Sikora Peter Noll, *Signale und Systeme*, TU Berlin.
- [4] www.elektor-magazine.fr/130165

Tableau 1. Analyse numérique du sinus créé.									
N_{Shift}	Valeur initiale α	f_{sin}/f_T	att.	η_{Sinus}	A	f_{sin}/f_T	mes.	μ	Δ
2	61	1/12,43	0,98	63,00	1/12,42	1,00	10		
3	40	1/17,68	1,00	58,50	1/17,75	2,50	2		
4	26	1/25,07	0,99	54,00	1/25,00	4,00	2		
5	4	1/35,50	0,99	12,00	1/32,00	7,00	1		
6	6	1/50,23	0,98	25,50	1/46,00	15,50	1		

NIDays

Le 3 février 2015

CNIT Paris la Défense



Vous êtes ingénieur, scientifique ou enseignant et vous souhaitez découvrir les dernières tendances en matière de conception de systèmes, de test et de contrôle ?

Alors NIDays est le rendez-vous à ne pas manquer !

Venez nous rejoindre sur l'édition 2015 de NIDays pour partager les bonnes pratiques de programmation, échanger des informations avec les ingénieurs NI et découvrir les dernières innovations qui inspirent les clients NI et leur permettent de résoudre leurs problèmes d'ingénierie.

- 70 stands de démonstrations
- 2 conférences plénières, avec en invité d'honneur Christophe Galfard, Docteur en physique théorique, spécialiste des trous noirs et de l'origine de l'Univers
- 50 conférences techniques et métiers
- 18 sessions de TP pour s'initier aux matériels et aux logiciels
- 2 coupes de robotique pour l'enseignement

Programme et inscription gratuite sur www.nidays.fr

01 57 66 24 24

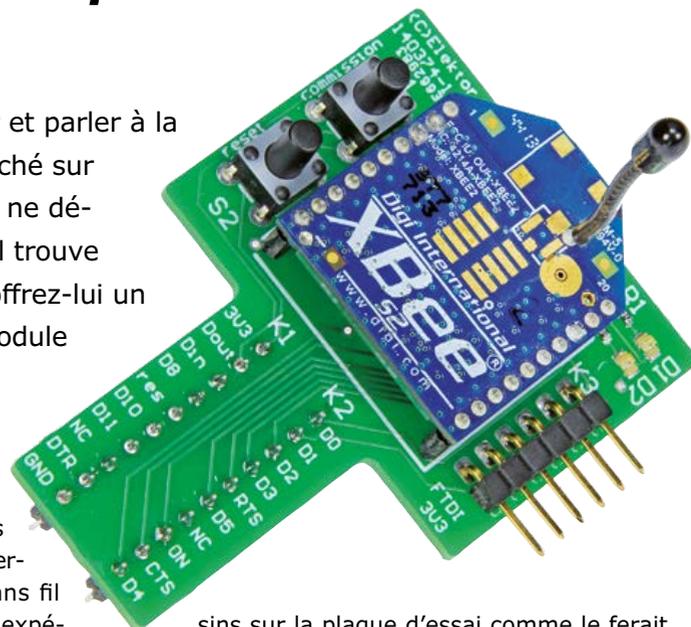


T-Board sans fil

XBee, Bluetooth, WiFi... écoutez *Radio Plaque-à-Trous*

Luc Lemmens
(Elektor.labs)

Le microcontrôleur aime écouter et parler à la radio, de plus en plus ! Même juché sur une plaque d'expérimentation, il ne dédaigne pas la liaison sans fil qu'il trouve bien pratique. Faites-lui plaisir, offrez-lui un moTule capable d'héberger le module sans fil de son choix !



Cette petite carte *Wireless* s'inscrit dans notre lignée de moTules T-Board. Elle permet d'utiliser nombre de modules sans fil populaires sur une plaque perforée ou expérimentale au pas standard de 2,54 mm. Parce que les modules radio ne marchent pas dans ces traces-là, ils comptent bien 2 x 10 broches de raccordement, mais distantes de... **2 mm**. Le présent moTule assure donc la compatibilité en interfaçant les deux réseaux. Il se relie à la plaque de base par deux barrettes de 10 picots au pas de 2,54 mm distantes l'une de l'autre de 7,62 mm (trois trous).

sins sur la plaque d'essai comme le ferait une carte rectangulaire. On fixe cette carte-ci de manière à ce que la large barre du T, avec le module radio, déborde à l'extérieur de la plaque et donc n'y occupe pas de place.

Figure 1.
Différents modules radio de Digi, Ciseco et Microchip. Ils s'adaptent tous au moTule T-Board Wireless.

La forme de la carte en T est propice à la disposition d'autres composants à proximité immédiate, car elle n'empêche pas l'accès aux contacts voi-

Au départ, ce moTule a été conçu pour le module XBee de Digi International dans le cadre d'un autre projet d'Elektor. Raison pour laquelle on retrouve, dans le schéma comme sur le circuit imprimé, les références du brochage de ce module-là. Son empreinte, désignée « *XBee form factor* », correspond exactement à celle d'autres modules WiFi et Bluetooth, comme les RN-171-XV, RN41x et RN42x de Microchip et les modules XRF de Ciseco [1]. Le **tableau 1** affiche la correspondance entre les brochages du module XBee de Digi, d'un Bluetooth, d'un WiFi de Microchip et du XRF de Ciseco (bande ISM). Il en ressort qu'on peut enficher tous ces modules sur le T-Board sans aucune modification. Toutes les broches essentielles, comme l'alimentation, l'UART et la mise à zéro, de Microchip et Ciseco correspondent exactement au brochage des produits Digi. Malgré tout, prenez la précaution de vérifier, si vous choisissez un autre type de module, qu'il s'adapte bien au T-Board, en



particulier les lignes d'alimentation, parce qu'il en existe des variantes.

Le circuit

À part le module radio, il y a peu de choses sur la carte, confirmation à la **figure 2**. Les connecteurs MOD1, K1 et K2 y jouent un rôle essentiel en raison de la différence d'écartement des broches sur les modules. On y trouve quand même un raccordement (K3) pour le câble d'interface USB/TTL de 3,3 V, un régulateur de tension de 3,3 V (IC1 et ses condensateurs de découplage C1 et C2), plus deux boutons-poussoirs (S1 et S2) et les LED D1 et D2 pour des fonctions spéciales. Le schéma montre aussi deux broches qui, sur le module XBee, sont intitulées NC, pour *Not Connected*, sans usage. Elles sont pourtant reliées au connecteur vers la plaque d'expérimentation, parce que certains modules d'autres fabricants leur attribuent une fonction.

Les deux boutons-poussoirs S1 (*Reset*, initialisation) et S2 (*Commissioning*, mise en service) ne s'utilisent généralement pas. S1 est parfois nécessaire lors du flashage d'un autre micrologiciel et lors du développement de nouvelles applications. En cas de malheureux plantage, ce n'est pas un luxe. On se sert de S2 avec le module XBee pour le diagnostic et la configuration de réseaux. Ce n'est pas non plus une manœuvre quotidienne, mais quand il faut... Vous trouverez davantage d'explications sur le sujet dans les notes et caractéristiques du produit sur Digi International [2].

La LED D1 (RSSI) peut servir d'indicateur de la force du signal lors de la réception du dernier paquet de données ; d'origine, elle est en activité, mais on peut l'éteindre et l'allumer avec le logiciel XCTU, par exemple. Digi appelle D2 une LED « associée », elle donne l'état du réseau et des informations de diagnostic en combinaison avec le bouton-poussoir S2. Sur le module XRF de Ciseco, D1 clignote chaque seconde quand le module fonctionne bien et D2 s'allume lors de l'envoi d'un paquet.

La **figure 3** reproduit le tracé des pistes du T-Board Wireless. Le régulateur de tension, les LED, condensateurs et résistances y sont en versions CMS, mais en raison de la grandeur choisie, ils sont encore raisonnablement faciles à souder à la main. Ce module est aussi disponible assemblé,

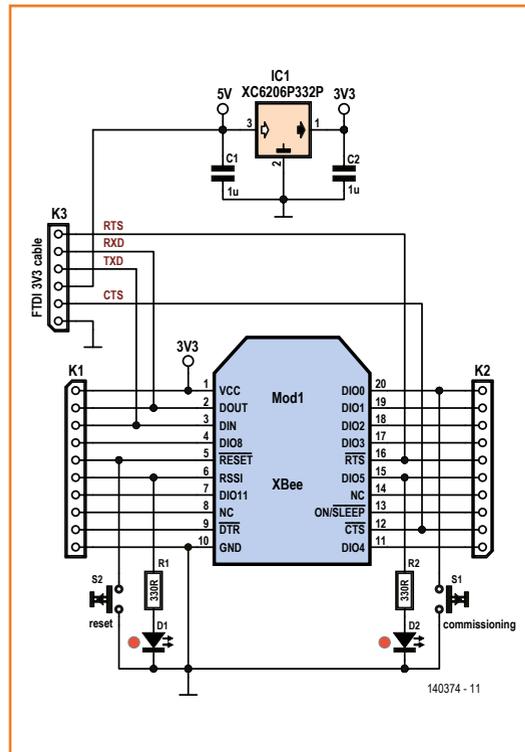


Figure 2. Un régulateur de tension et quelques composants passifs, il n'en faut pas plus à ce module sans fil.

Tableau 1. Brochage de différents modules sans fil

Broche n°	Digi XBee XBee	Ciseco XRF ISM (868 MHz)	RN171XV WiFi	RN42XV Bluetooth
1	3V3	3V3	3V3	3V3
2	Dout	Dout	Dout	Dout
3	Din	Din	Din	Din
4	DIO8	RTS	GPIO8	GPIO7
5	#reset	#reset	#reset	#reset
6	RSSI	P1_7	GPIO5	GPIO6
7	DIO11	P1_6	GPIO7	GPIO9
8	NC	P1_5	GPIO9	GPIO4
9	DTR	P1_4	GPIO1	GPIO11
10	GND	GND	GND	GND
11	DIO4	P0_1	GPIO14	GPIO8
12	CTS	CTS	RTS	RTS
13	on/#sleep	on/#sleep	GPIO4	GPIO2
14	NC	NC	NC	NC
15	DIO5	P0_7	GPIO6	GPIO5
16	RTS	P2_0	CTS	CTS
17	DIO3	P2_1	Sensor5	GPIO3
18	DIO2	P2_2	GPIO3	GPIO7
19	DIO1	P2_3	Sensor3	AIO0
20	DIO0	P0_5	Sensor2	AIO1

prêt à l'emploi dans l'e-choppe (140374-91). Le même circuit imprimé, dépouillé des composants CMS énumérés ici, sert également d'indispensable adaptateur pour installer le module radio sur une plaque d'expérimentation.

La bonne tension

Ce moTule dispose d'une barrette à 6 picots pour se connecter en UART à un PC par un câble d'interface USB/TTL de **3,3 V** (réf. Elektor 080213-72) pour la communication et la configuration d'un module sans fil XBee ou autre. Il faut savoir que la version 3,3 V du câble d'interface reçoit sur la ligne V_{DD} du 5 V, d'où la nécessité d'un régulateur de tension pour alimenter le module sans fil.

On trouve sur le marché différents modèles et versions de modules XBee et autres modules radio ainsi qu'au moins autant de discussions sur l'internet pour savoir si un type déterminé est capable ou non de supporter le 5 V. Il y a effectivement (eu) des exécutions qui le toléraient, mais soyons clairs sur la question, le 3,3 V est toujours sûr et satisfaisant, si vous y dérogez en vous écartant des spécifications, c'est sous votre propre responsabilité. Il ne faut jamais raccorder de la logique à 5 V aux entrées comme aux sorties. Utilisez toujours la version à 3,3 V du câble d'interface USB/TTL sur K3 et veillez aussi à intercaler un changeur de niveau 3,3 V/5 V quand vous utilisez l'un de ces modules en combinaison avec de la logique à 5 V !

Autres possibilités

Le moTule peut aussi servir à configurer des modules XBee ou à charger un nouveau micrologiciel. Le câble d'interface 3,3 V assure alors la liaison avec le PC sur lequel tourne le logiciel XCTU de Digi pour la configuration et la programmation nécessaire à rendre le module capable d'exécuter l'application souhaitée. Par ailleurs, on peut également adapter de nombreuses instructions avec des commandes AT en utilisant un programme de terminal, mais XCTU est d'emploi plus facile pour un module XBee.

Dans de nombreuses applications avec ce genre de modules radio, on n'utilise qu'un UART pour la transmission de données vers le sans fil et en sens inverse. Il suffit alors de raccorder le module par K3 sur un système à microcontrôleur ou un PC. Les autres broches d'entrée/sortie ne sont donc pas utilisées. Le cas échéant, on peut même découper la partie du circuit imprimé où se trouvent K1 et K2 pour réduire l'encombrement.

(140374 - version française : Robert Grignard)

Figure 3.
Tracé des pistes du T-Board Wireless. Dans l'e-choppe, il est déjà disponible assemblé et prêt à recevoir son module sans fil (140374-91).

Liens

- [1] www.elektor.fr/xrf-wireless-module-140403-91
- [2] www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/
- [3] www.elektor-magazine.fr/140374

Liste des composants

Résistances :

R1, R2 = 330 Ω 5 % 100 mW, CMS 0603

Condensateurs :

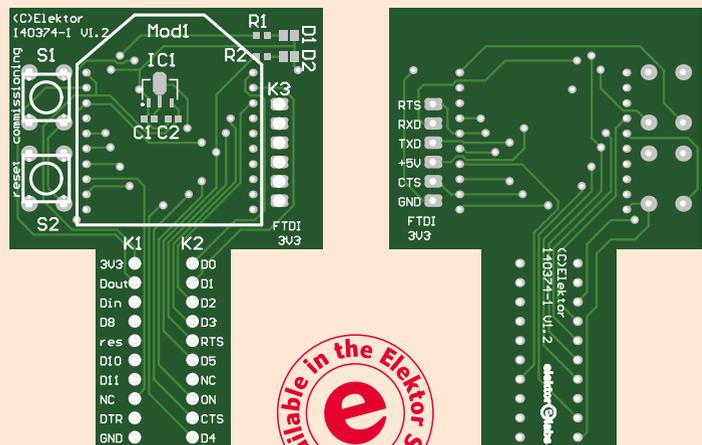
C1, C2 = 1 µF/10 V 10 %, CMS 0603

Semi-conducteurs :

D1, D2 = LED rouge 20 mA, CMS 0805
IC1 = XC6206P332PR, régulateur de tension 3,3 V, CMS SOT-89-3

Divers :

K1, K2 = barrette à 10 picots, au pas de 2,54 mm
K3 = barrette à 6 picots au pas de 2,54 mm
Mod1 = 2 barrettes à 10 voies femelles, au pas de 2 mm
S1, S2 = bouton-poussoir encartable, 6 x 6 mm module sans fil, p.ex. module XRF de Ciseco [1] ou module XBee ZB de Digi
circuit imprimé réf. 140374-1 ou moTule construit, mais sans module radio réf. 140374-91



adaptateur de niveau 3 états (*tristate*)

Certains utilisateurs d'Arduino aimeraient augmenter la quantité de RAM de cette carte. Ils retombent souvent sur la puce SRAM 23K256 à connexion SPI. Lors de la mise en œuvre de ce circuit, ils se trouvent confrontés au fait que la tension d'alimentation maximale tolérée est de 3,6 V, alors qu'Arduino travaille avec des niveaux logiques de 5 V.

Nicolas Boullis
(France)

On trouve, sur Internet, différentes solutions : la plupart ne sont guère fiables voire catastrophiques (telle la suggestion de connecter le 23K256 purement et simplement au 5 V, bien au-delà de la valeur maximale absolue).

Il y a heureusement, sur la carte Arduino, une tension régulée de 3,3 V que l'on pourra utiliser pour alimenter le 23K256. Il est facile d'adapter les signaux CS, SCK et MOSI à l'aide de diviseurs de tension à deux résistances, ces signaux allant d'Arduino au 23K256.

Ça se complique un peu avec le signal MISO (*serial data output*). Normalement, on utilise, pour les signaux I²C, un MOSFET à canal N avec deux résistances de rappel, mais dès lors il n'est plus possible de distinguer un niveau haut d'un état haute impédance. En outre, des capacités parasites rallongent les temps de montée des fronts de signaux. Il est préférable d'utiliser un TXB0104 (décaleur de niveau bidirectionnel), mais lui non plus ne permet pas un état de haute impédance.

Le présent circuit, simple et à base de composants courants, constitue une meilleure solution. Si la sortie SO du circuit de RAM sériel est flottante (= haute impédance), les transistors T1 et T2 sont juste passants. Comme la tension base-émetteur de T1 et T2 est alors de 0,65 V (diviseur de tension R1/R7/R8/R2), il circule environ 0,45 mA à travers chaque transistor (environ 0,15 V/330 Ω). Il en résulte sur R5 et R6 une chute de tension d'environ 0,45 V, de sorte que T3 et T4 bloquent tous deux. La sortie MI est alors flottante. En cas de niveau haut sur la ligne SO, la tension sur la base de T2 croît, tandis que T1 bloque totalement. Le courant à travers T2 dépasse alors 2 mA, de

sorte que la chute de tension sur R5 fait conduire T3 : ainsi la sortie IM passe au niveau haut (5 V). Dans le cas d'une entrée au niveau bas, c'est l'inverse : la conduction de T1 augmente ce qui rend T4 passant et se traduit par un niveau bas à la sortie de l'étage-tampon.

Le courant de base de T1 et T2 est limité par R7 et R8, tandis que C1 et C2 accélèrent la commutation. Les diodes Schottky D1 et D2 empêchent la saturation de T3 et T4 lesquels commutent donc rapidement. Monté sur une plaque d'expérimentation, ce circuit a fonctionné de façon fiable jusqu'à 3 MHz.

(140224 - version française : Guy Raedersdorf)

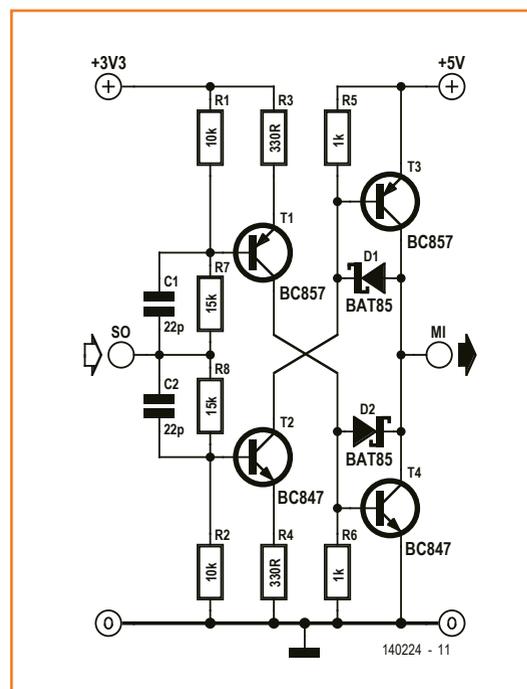


Figure 1.
Ce circuit adapte le niveau de 3,3 à 5 V et transfère l'état flottant (haute impédance).

Allons au son de la logique

Beep indique le niveau d'un signal au moyen d'un son, vous pourrez donc l'utiliser sans quitter des yeux votre circuit. Beep produit un son de basse fréquence lorsque le niveau mesuré est bas, et un son de haute fréquence lorsque le niveau est haut. Lorsque le niveau du SUT est indéfini parce qu'il s'agit d'une entrée non connectée, d'une sortie flottante (haute impédance) ou à collecteur ouvert sans résistance de rappel vers le haut, Beep reste silencieux. Nombreux sont les signaux qui changent trop fréquemment de niveaux pour que l'oreille puisse distinguer ces variations ; Beep a donc été conçu pour rendre audibles également de tels signaux.

Osciller et diviser

Beep n'utilise aucun microcontrôleur. Alors comment fait-il ? Facile, il utilise deux oscillateurs et un diviseur de fréquence. Un oscillateur (IC3B, R4, C5) se charge du son de basse fréquence, l'autre (IC3C, R5, C6) produit le son de haute fréquence. Les fréquences sont déterminées par les valeurs de R4/C5 (basses) et R5/C6 (hautes). Puisque le SUT ne peut être simultanément haut et bas, un seul des deux oscillateurs est actif à un moment donné, et leurs sorties peuvent donc être mises en parallèle pour attaquer le buzzer. Le choix de l'oscillateur dépend du signal appliqué sur la broche 1 de JP2. Lorsque son niveau est bas, le niveau de la broche Enable (8) d'IC3C est également bas ; la sortie d'IC3C prend alors un niveau haut stable (une porte NAND fonctionne

de cette façon, entrées à bascule de Schmitt ou pas) et l'oscillateur est bloqué. La sortie d'IC3A prend de même le niveau haut ; le niveau de sa broche d'entrée 2 ne sera pas exactement 0 V, mais sera suffisamment bas pour que la porte décide que la sortie doit être au niveau haut. Contrairement à IC3C, IC3B voit un niveau haut sur sa broche Enable (5) et cet oscillateur ainsi débloqué produit un son de basse fréquence. C'est l'inverse qui se produit lorsque le SUT présente un niveau haut. Les deux entrées d'IC3A se retrouvent au niveau haut, ce qui entraîne un niveau bas de sa sortie, qui donc bloque l'oscillateur basse fréquence IC3B. La broche Enable (8) d'IC3C est quant à elle au niveau haut et l'oscillateur produit un son de haute fréquence.

Lorsque Beep est soumis à une entrée non connectée, une sortie flottante (haute impédance) ou à collecteur ouvert sans résistance de rappel, le signal sous test n'est pas assez puissant pour forcer l'entrée à haute impédance de Beep (valeurs élevées de R1, R2 et R3) dans un état univoque. En raison des niveaux de tension créés par le diviseur R1, R2 et R3, les entrées d'IC3A seront toutes deux au niveau haut et la broche Enable d'IC3C sera estimée être au niveau bas. Les deux oscillateurs sont alors bloqués et Beep ne bip pas.

Lorsque le niveau du SUT alterne entre haut et bas de façon instable, les oscillateurs sont activés tour à tour et produisent un son à deux tons. Ce « double ton » varie de plus en plus

Principe de l'oscillateur

Considérez l'oscillateur IC3B, R4 et C5, et supposez que la broche 5 d'IC3B soit au niveau haut. C5 est déchargé au démarrage, ce qui maintient la broche d'entrée 6 d'IC3B au niveau bas. Comme IC3B est une porte NON-ET, sa sortie passe au niveau haut (**tableau 1**) et R4 charge donc C5. Lorsque la tension aux bornes de C5 dépasse un certain seuil, IC3B voit sa broche d'entrée 6 au niveau haut. Les deux entrées sont alors au niveau haut ; la sortie d'IC3B passe donc au niveau bas et R4 décharge C5. Lorsque la tension aux bornes de C5 chute sous un certain seuil, IC3B voit sa broche 6 au niveau bas, et nous revoici dans la situation de départ : le cycle reprend et l'oscillateur oscille. Sa fréquence

dépend des valeurs de R4 et C5, ainsi que des seuils haut et bas de la porte.

En raison de sa fonction NON-ET, le niveau de sortie d'IC3B ne peut pas être bas lorsque sa broche 5 a un niveau bas stable ; l'oscillateur est alors désactivé.

Tableau 1. Table de vérité d'une porte NON-ET (porte ET inversée).

entrée A (broche 5)	entrée B (broche 6)	sortie (broche 4)
0	0	1
0	1	1
1	0	1
1	1	0

Liens

- [1] Page du projet Beep : www.elektor.com/140410
- [2] Composants Passe-Partout : <https://github.com/ElektorLabs/PreferredParts>

vite à mesure qu'augmente la fréquence du SUT, jusqu'au point où les deux oscillateurs ne peuvent plus suivre et deviennent silencieux. Le diviseur de fréquence IC1 entre alors en jeu. Il divise la fréquence par 256, 1024 ou 2048 pour rendre audibles des fréquences allant jusqu'à 4 MHz (avec l'hypothèse optimiste que vous perceviez les fréquences jusqu'à 20 kHz). Le bouton SW1 permet de sélectionner la fréquence de sortie la plus adaptée à votre oreille.

Les sorties des oscillateurs sont trop faibles pour actionner directement le buzzer. Un coup de patate est donc envoyé par l'intermédiaire de deux tampons reliés en parallèle ; cet étage de sortie a également été ajouté au diviseur de fréquence afin que l'intensité sonore soit toujours égale.

Construction et utilisation

Beep est alimenté par le circuit sous essai. Plus de tracas de piles vides ! Connectez bien la broche 1 de JP1 à l'alimentation du circuit dont vous souhaitez mesurer les niveaux logiques. Si

vous utilisez des CMOS, l'alimentation peut aller jusqu'à 12 V ; avec des versions TTL comme le (non recommandé) 74HCT4040, l'alimentation ne doit pas excéder 5 V.

Pour construire Beep, c'est facile, vous pouvez commander son circuit imprimé depuis l'e-choppe. La carte a été conçue pour tenir dans un boîtier Hammond 1593D.

Les composants sont tous traversants (**fig. 2**) et la plupart font partie des **composants Passe-Partout d'elektor.labs** [2]. Leurs empreintes et dimensions sont donc bien définies et vous les trouverez facilement. Les CI logiques sont eux aussi standard et faciles à trouver. Ce sera peut-être plus délicat pour SW1, mais vous pouvez le remplacer par une barrette mâle et un cavalier.

(140410-I - version française : Hervé Moreau)

Figure 2. Le circuit imprimé de Beep a été conçu pour tenir dans un boîtier Hammond 1593D.

Liste des composants

Résistances

Défaut : 5 %, 0,25 W
 R1, R3 = 2,2 MΩ
 R2 = 1 MΩ
 R4, R6 = 10 kΩ
 R5 = 1 kΩ
 R7 = 0 Ω (ou morceau de fil)

Condensateurs

Défaut : au pas de 5,08 mm
 C1, C2, C3 = 4,7 μF, 50 V (au pas de 2 mm)
 C4 = 82 pF
 C5 = 470 nF
 C6 = 330 nF
 C7 = 100 nF

Semi-conducteurs

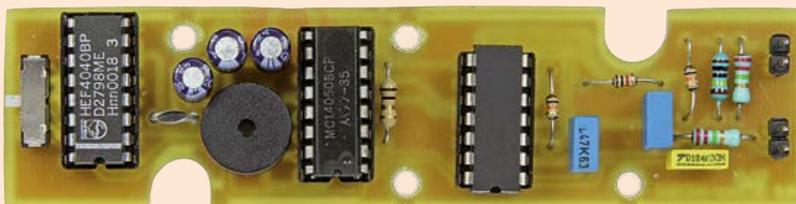
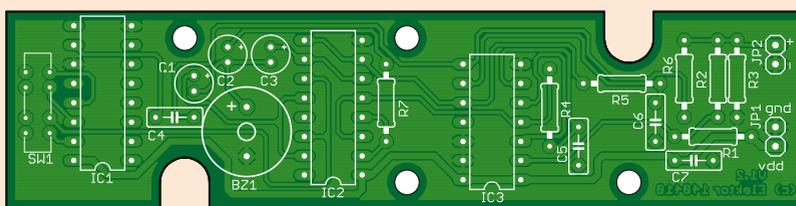
IC1 = HEF4040BP (ou équivalent)
 IC2 = CD4050BE (ou équivalent)
 IC3 = HEF4093BP (ou équivalent)

Divers

SW1 = interrupteur à glissière, 2 contacts

et 3 positions (RS 702-3568 ou équivalent)
 BZ1 = buzzer, Ø 12 mm
 JP1, JP2 = barrette mâle à 2 broches
 2 supports CI DIP-16 (IC1 et IC2)
 1 support CI DIP-14 (IC3)

boîtier, Hammond 1593D
 circuit imprimé réf. 140410-1



une capa et puis ça va

Il y a quelques jours, à la déchetterie où j'étais venu pour me défaire de vieilles imprimantes, mon attention a été attirée par un PC, intact en apparence. À la vue du macaron *Intel Core™ 2 Duo* et de l'autocollant avec une clé *OEM Windows Vista™ Édition Familiale Premium*, j'avais une idée approximative de l'âge de la machine. Mon ordinateur portable actuel est aussi de la génération Vista. La tour n'avait apparemment pas trop souffert et je l'ai donc embarquée... pour voir ! L'instinct de récupération, vous connaissez ?

De retour à la maison, je l'ai aussitôt ouverte, évidemment. Tous les composants étaient présents à l'exception du disque dur, que l'ancien propriétaire avait retiré proprement plutôt que de l'arracher comme un sauvage. Il ne restait qu'à essayer de ressusciter la bête : j'y relie un écran, un clavier et une souris, je branche ensuite le câble d'alimentation, appuie sur le bouton de mise sous tension en façade et... ça marche ! Un petit tour dans son BIOS m'a agréablement surpris quant à la configuration de la machine :

- CPU : Intel Core 2 Duo E4500 à 2,2 GHz
- RAM : 2 Go DDR2 à 667 MHz
- Carte mère : Fujitsu D2151-A21
- Carte graphique : NVidia Geforce 8400GS 256 Mo
- Lecteurs : lecteur DVD-ROM et graveur DVD-R/RW DL
- Alimentation : 350 W
- Lecteur de carte interne
- Boîtier Cooler Master

Dans 2ⁿ>3 mois, on rira de cette configuration, mais pour l'instant, c'est bien suffisant pour naviguer sur la Toile, suivre le courrier et même faire un peu de musique. Avant de retourner à la déchetterie pour en rapporter un disque dur pour mon bébé, il me fallait résoudre le problème (potentiel) qui avait attiré mon attention lors de l'inspection de la carte mère : les condensateurs près du connecteur ATX.

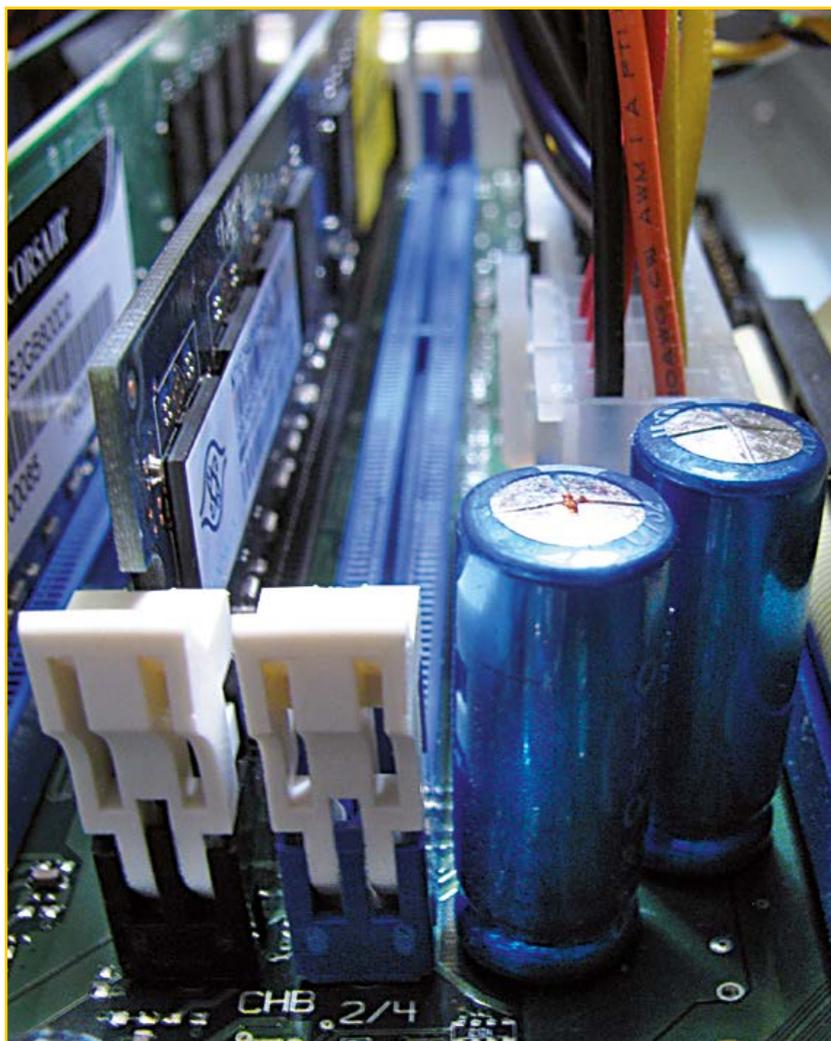
La carte mère fonctionne, mais mon intuition d'électronicien me dit que **ces condos sont sur le point d'exploser** — ce qui, vous en convien-

dre, serait navrant. S'il suffit de remplacer quelques condensateurs boursoufflés pour offrir une seconde vie à ce PC, c'est avec plaisir que je le ferai ! J'ai donc remplacé ces *bombes à retardement* par des condensateurs à faible ESR de bonne facture (Panasonic, série FM).

J'ai fini par installer un nouveau système d'exploitation sur un disque dur récupéré entretemps, et depuis mon PC de la déchetterie tourne comme un coucou suisse. Avec un accent californien... ah oui, au fait, j'en ai fait un Hackintosh, mais chut...

Thijs Beckers
(elektor.labs)

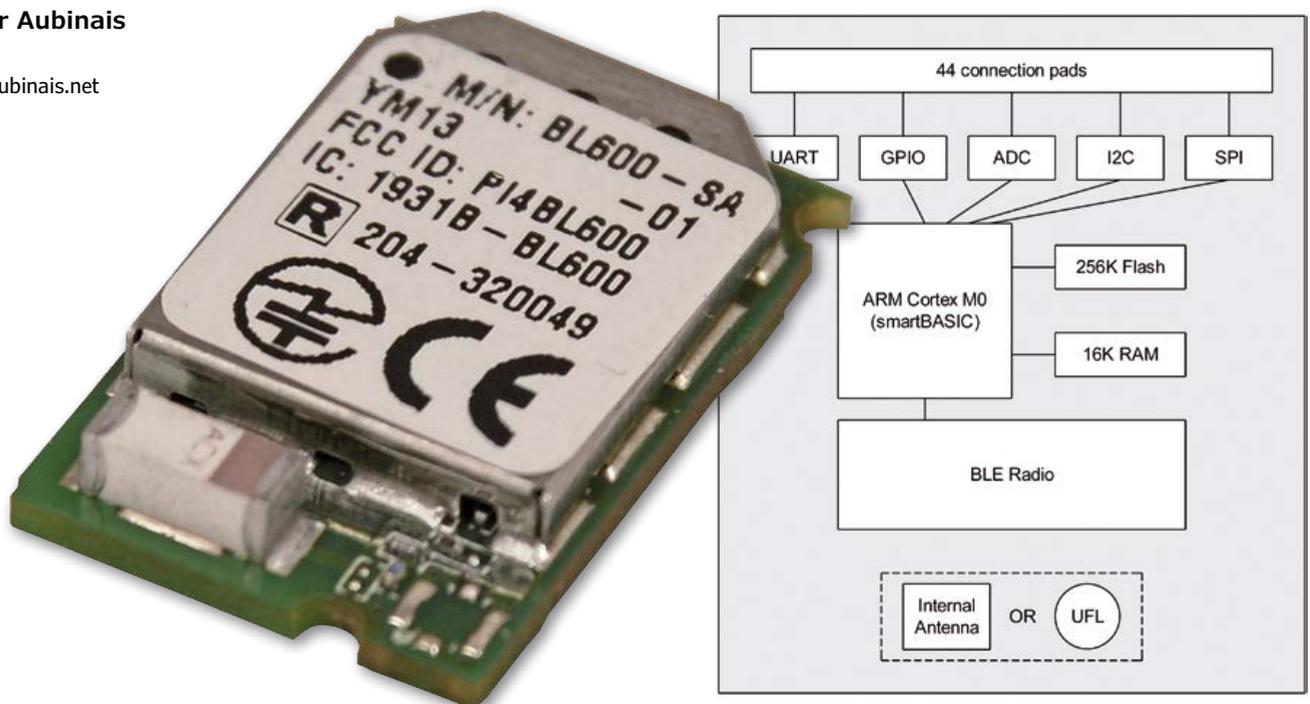
(140363 – version française : Kévin Petit)



thermomètre sans fil Bluetooth Low Energy

Affichage de la température à distance sur votre téléphone tactile

Jennifer Aubinais
(Paris)
elektor@aubinais.net



Ce thermomètre extérieur en boîtier étanche communique en *Bluetooth Low Energy 4.0* (BLE) avec un téléphone tactile récent : iPhone 4S (ou plus), Android 4.3 (ou plus) grâce au module autonome BL600-SA conçu par *Laird Technologies* pour offrir sur un plateau le mode de communication sans fil.

Il n'y a aucun (autre) composant actif, tout est dans le mini-module qui se programme en... BASIC !

J'adore le Bluetooth et ce thermomètre me fournit un bon prétexte pour l'utiliser dans sa version la plus récente, intéressante car l'intensité du courant requis est beaucoup plus faible qu'avec les normes Bluetooth antérieures (2.0 et 1.0) avec lesquelles le mode BLE n'est donc pas compatible.

Bluetooth en BASIC

Le langage de programmation *smartBASIC* (orienté événements) du BL6000-SA simplifie

l'intégration du Bluetooth en facilitant non seulement la gestion des capteurs connectés directement au module, mais aussi la transmission des valeurs mesurées à n'importe quel récepteur Bluetooth v4.0 (téléphone ou tablette tactiles, ordinateur, passerelle...). C'est donc presque un jeu (de grands enfants) désormais de communiquer par ondes radio avec de petits appareils portables, alimentés par piles AAA ou par piles bouton.

Remerciements :
Philippe et le support
Laird Technologies

Le A de la référence du module indique que son antenne y est intégrée.

Remarquable par sa faible consommation, le module BL600-SA est basé sur le jeu de puces nRF51822 de *Nordic Semiconductor* et réunit tout le matériel et le logiciel requis pour une communication radio avec un débit brut de 1 Mbps dans la bande 2,402 – 2,480 GHz. Ses caractéristiques principales :

- interfaces UART, I²C, SPI
- 28 entrées/sorties polyvalentes (GPIO)
- 6 entrées analogiques (CAN à 10 bits)
- consommation : 0,4 µA en sommeil profond
- 5 µA en veille
- 10 mA durant l'émission
- programmation aisée en langage *smartBASIC*
- faible encombrement : 19 x 12,5 x 3 mm
- portée en champ ouvert : jusqu'à 20 m

Impressionnant, hein ? Il est remarquable aussi par son prix, tout petit. La photo (**fig. 1**) montre le module sur la carte d'évaluation (SDK) proposée par *Laird Technologies*. Oui, il faut bien regarder pour le trouver... Le fabricant semble le destiner principalement à des applications de télé-métrie paramédicales (pression sanguine, rythme cardiaque, température...), mais il ne tient qu'à nous tous d'étendre le champ de ses applications.

Lors de la première utilisation, il ne m'a pas fallu une heure pour envoyer des 1 et des 0 au BL600 depuis mon iPhone et puis faire clignoter une LED. Après ce début encourageant, je n'ai plus compté les heures pour aboutir à mon petit thermomètre extérieur autonome. Quand on aime, on ne compte pas. Surtout pas le nombre de pages de la documentation très nourrie ni le temps passé sur le site du fabricant [4]. Et puis il y a la sempiternelle question de la miniaturisation : la densité d'intégration du module est si forte qu'il est difficile mais pas impossible de le souder à la main (c'est de l'horlogerie !). Heureusement, *Laird Technologies* a trouvé une astuce simple pour caler le module avec précision, j'y reviendrai.

Clignotement

Avec ce premier programme – très simple – de LED clignotante, j'ai pu vérifier que lorsque la LED est éteinte, le module ne consomme que



5 µA. On utilisera le même principe pour stopper la fonction Bluetooth du module tous les x temps afin d'économiser la pile.

Figure 1.
À en juger par la taille et la densité de la population de la carte d'évaluation du BL600, il doit s'en passer des choses dans ce discret module radio (le rectangle blanc au bord à droite).

```
//*****
// Jennifer AUBINAIS 2014
// Test sleep with led
//*****

//*****
' LONG TIME : 2 secondes => led off
//*****

FUNCTION Func0()
Dim rc
' led off
rc = GPIOSetFunc(17,2,0)
TIMERSTART(1,2000,0)
ENDFUNC 1

//*****
' SHORT TIME : 100 millisecondes => led on
//*****
FUNCTION Func1()
Dim rc
' led on
rc = GPIOSetFunc(17,2,1)
TIMERSTART(0,100,0)
ENDFUNC 1

//*****
' event
//*****
ONEVENT EVTMR0 CALL Func0
ONEVENT EVTMR1 CALL Func1
```

```
'*****
' main program
'*****
Dim rc
' output GPIO 17 and led on
rc = GPIOSetFunc(17,2,1)
TIMERSTART(0,100,0)
// SLEEP : close uart
uartclose()
rc = GPIOSetFunc(21,2,1) '// TX
rc = GPIOSetFunc(23,2,0) '// RTS
WAITEVENT
```

Schéma

Le tour du schéma sera vite fait. Pour mesurer la température, j'utilise le CAN du BL600-SA (MOD1 sur le schéma de la **fig. 2**) avec une thermistance CTN reliée à K3. Elle peut éventuellement être déportée, la longueur de fil n'est pas critique. La mesure est faite à l'aide d'un diviseur de tension avec une résistance connue : $R1 = 10\text{ k}\Omega$ 0,1 %.

L'équation de la température en fonction de la valeur de la CTN est une fonction logarithmique qui dépasse les capacités de calcul du BL600. De plus, ce calcul, qui dépend d'un tableau de données (températures, résistances, coefficient thermique alpha) fourni par le fabricant de la thermistance [10], sera donc effectué sur et par le téléphone tactile et une application iOS ou Android. J'y reviens dans un instant.

L'alimentation passe par la broche SIO_19 du BL600 pour réduire la consommation à 5 μA en mode de veille. En mode de sommeil profond, la consommation tombe à 0,4 μA , mais le réveil ne peut se faire que par une remise à zéro du module ou par un changement d'état d'une entrée. Alors qu'en mode de veille, des interruptions par le logiciel permettent de rétablir la communication Bluetooth brièvement, le temps de voir si quelqu'un veut se connecter. J'ai choisi arbitrairement un sommeil de 500 ms (paramètre dans la fonction TIMERSTART). Quand la connexion est établie, la consommation pendant l'émission est de 10 mA.

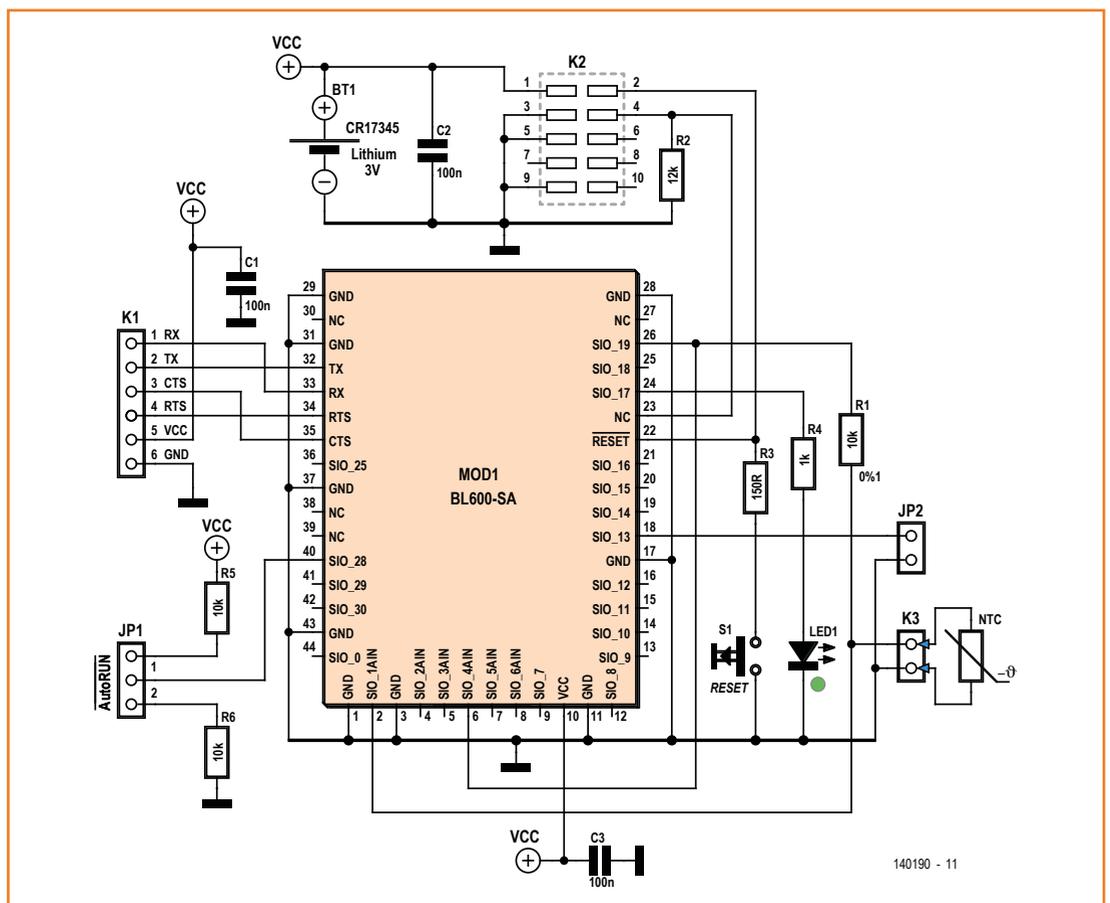


Figure 2. Schéma du thermomètre BT avec le BL600-SA. Dans la famille Laird, le BL600 ne connaît que le mode LE (*low energy*) tandis que le BT900 fait du BT classique et du BLE, mais il est encore plus petit et donc plus difficile à souder !

Le connecteur K2 sert pour une éventuelle mise à jour du progiciel du BL600 par une interface spécifique JTAG. En principe, cette fonction est inutile pour ce projet (... mais on ne sait jamais). Le connecteur K1 permet de programmer le BL600 via une interface série 3,3 V à partir d'un PC. Parfait pour cet usage, le BOB FT232R [6] proposé par Elektor dans son e-choppe.

La LED a deux rôles :

- *débugage* (avec cavalier) : elle clignote tout le temps et montre les modes veille de Bluetooth et connexion au module
- *normal* (sans cavalier) : elle clignote brièvement lors de l'initialisation pour signaler le bon démarrage du thermomètre

Lors de mes premiers essais, la transmission des données était tronquée : il m'a fallu choisir entre réveiller le module plus longtemps ou transmettre moins de données. Pour ménager la pile, j'ai retenu la deuxième solution.

Exemple de train de données communiquées par le module via Bluetooth :

PW3012V853C433

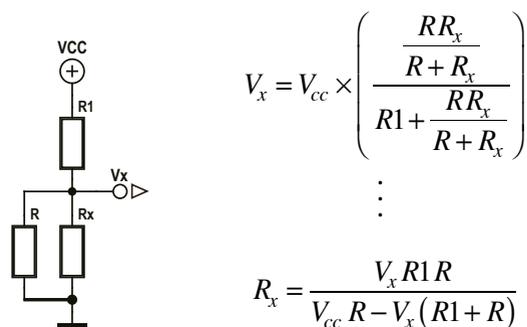
PW : tension d'alimentation (pile) à afficher sur l'application

V : tension d'alimentation du pont diviseur de tension

C : tension aux bornes du pont diviseur : R1 / CTN
Soit, après calcul, une température correspondante de 24,3 °C [voir le paragraphe **Calculs** ci-dessous]

Calculs

Sans entrer dans les détails, on retiendra que le calcul de la valeur de la thermistance passe par une mesure de tension sur un diviseur.



Pour la température, je ne me contente pas ici d'une équation simple entre thermistance et tem-

pérature en appliquant un coefficient béta comme on le fait ordinairement :

$$R_{ctn} = R_{25} \times e^{\beta \times \left(\frac{1}{T} - \frac{1}{298} \right)}$$

R₂₅ : valeur de la résistance à 25 °C

T : température de calcul en K

β : coefficient béta de la CTN

R_{ctn} : valeur de la CTN à la température de calcul

J'utilise un coefficient alpha α(%/K) qui varie selon des plages de température établies par le fabricant [1] (**tableau 1**), ce qui donne l'équation suivante :

$$R_T = R_{Tx} \cdot e^{\left[\frac{\alpha_x (Tx)^2}{100} \cdot \left(\frac{1}{T} - \frac{1}{Tx} \right) \right]}$$

R_T : résistance à la température T

R_{Tx} : résistance à la température Tx dans le tableau

T_x : température en K inférieure à la température de mesure trouvée dans le tableau

T : température de mesure en K (T_x < T < T_{x+1})

α_x : coefficient thermique à la température T_x

Tableau 1.		
R/T No	4901	
T (°C)	B _{25/100} = 3950 K	
	RT/R25	α (%/K)
-30,0	16,915	6,1
-25,0	12,555	5,9
-20,0	9,4143	5,7
-15,0	7,1172	5,5
-10,0	5,4308	5,4
-5,0	4,1505	5,2
0,0	3,2014	5,0
5,0	2,5011	4,9
10,0	1,9691	4,7
15,0	1,5618	4,6
20,0	1,2474	4,5
25,0	1,0000	4,3
30,0	0,808	4,2
35,0	0,6569	4,1
40,0	0,5372	4,0

À partir de cette même équation, le calcul de la température, en appliquant le coefficient thermique α_x , donnera ceci :

$$T = \frac{1}{\frac{100}{\alpha_x \cdot (T_x)^2} \cdot \ln\left(\frac{R_T}{R_{T_x}}\right) + \frac{1}{T_x}}$$

Et enfin un exemple de calcul à partir de cette dernière équation et des paramètres suivants :

$R_T = 10506,46 \Omega$ (R_{CTN})
 $\Rightarrow 12474 \Omega > R_T/T_{25} > 10000 \Omega$
 $\Rightarrow \alpha_x = 4,5$
 $\Rightarrow T_x = 20 \text{ °C} = 293,15 \text{ K}$
 \Rightarrow

$$T = \frac{1}{\frac{100}{4,5 \cdot (293,15)^2} \cdot \ln\left(\frac{10506,46}{12474}\right) + \frac{1}{293,15}}$$

$$T = 297,01 \text{ K}$$

$$T = 297,01 - 273,15 = 23,86 \text{ °C}$$

Avec une tolérance de 0,1% pour R1, la précision de ce thermomètre est de 2% ou mieux. Si on se contente de 3% de précision, R1 pourra être une résistance ordinaire (et meilleur marché) avec une tolérance de 1% (au lieu de 0,1%).

Réalisation

Après avoir trouvé un modèle convenable de boîtier étanche, je me suis lancé dans l'étude d'un circuit imprimé (**fig. 3**) pour réaliser une sonde de

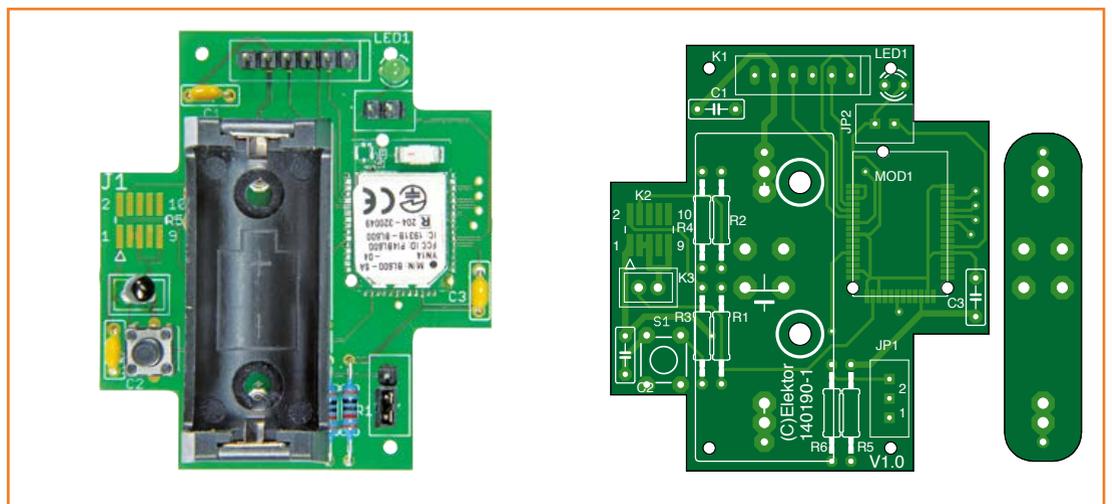
température à laquelle je n'aurais plus à toucher pendant... mettons 10 ans ! J'ai créé moi-même la bibliothèque Eagle du BL600 (disponible dans le téléchargement de cet article [2]). Mon boîtier est un modèle en ABS très robuste et étanche dont la configuration intérieure impose une découpe irrégulière du circuit imprimé.

J'étais sceptique au départ devant la consommation de ce nouveau module. C'est pourquoi je ne l'ai pas alimenté avec une pile bouton CR2032, mais une CR123. J'ai dessiné pour le support de pile un adaptateur (fig. 3) amovible par simple enfichage sur le circuit principal au moyen de picots tulipe. Il suffit de récupérer les 8 tulipes en les extrayant de l'enrobage plastique d'une rangée de picots, puis de les souder sur le circuit imprimé oblong, et enfin de souder, sur la même carte, le support de pile CR123 (attention à la polarité !).

Le thermomètre Bluetooth est proposé par Elektor sous forme de module partiellement assemblé, avec le BL600 **déjà soudé** [3]. Ceux qui sont tentés par le soudage au four du module BL600 se confectionneront un stencil pour appliquer la pâte. Trois vis de 1,6 mm dans chacun des trous prévus pour ça permettent de caler parfaitement le stencil avant d'appliquer la pâte à souder sur le circuit imprimé. Pour caler le module avec la même précision, engagez les trois vis de 1,6 mm autour de l'emplacement du BL600 par en dessous. Puis posez le module BL600 qui, guidé ses trois encoches de 1,6 mm glissant le long des vis, sera calé au dixième de millimètre près. Je vous propose un clip vidéo [5].

Figure 3.

Pour ceux qui ne se sentent pas une âme de dentellière, Elektor propose ce circuit imprimé avec le module BL600-SA monté. Il ne leur restera qu'à souder quelques composants traversants.



Une application exemplaire du module radio BL600-SA

Il ne reste ensuite qu'à enfourner le circuit imprimé.

Dégagez 8 autres picots d'une barrette tulipe de leur enrobage plastique et soudez-les sur la carte puis coupez la partie fine de ces picots pour n'en garder que les tulipes : elles recevront la carte oblongue sur laquelle est soudé le support de la pile.

Il n'y a rien à ajouter sur les autres composants, tous traversants, si ce n'est que R2 n'est utile que pour la mise à jour du progiciel et peut donc être omise, de même que le connecteur K2.

Selon l'utilisation que vous en ferez, vous pouvez adopter mon dispositif avec support de piles amovible, dans le même boîtier que moi, ou toute autre configuration adaptée aux besoins et au coffret de votre choix.

Le cavalier JP1 permet de choisir entre deux modes :

- *autorun* (position 1) qui permet de lancer le programme automatiquement lors du branchement de l'alimentation ou après une remise à zéro.
- *commande* (position 2) qui permet l'accès au module via la liaison série pour lancer des commandes AT comme p. ex. « AT&f 1 »



qui efface la mémoire du programme puis relance le BL600 et permet de charger le programme via l'interface série.

Le cavalier JP2 permet de se mettre en mode débogage et d'envoyer sur le port série les messages du programme. Il est absent pour une utilisation normale.

Programme du BL600

La programmation du module est simple grâce au *smartBASIC* de *Laird Technologies*. C'est du BASIC comme à l'école, avec des fonctions arithmétiques simples pour faciliter le traitement des données acquises. Il permet la création de sous-programmes et de fonctions et gère les fonctions E/S du BL600 ainsi que toutes les fonctions

Liste des composants

Résistances :

R1 = 10 kΩ 0,1%*
R2 = 12 kΩ*
R3 = 150 Ω
R4 = 1 kΩ
R5, R6 = 10 kΩ
thermistance CTN B57891S103F8 10 kΩ (2112816)

Condensateurs :

C1, C2, C3 = 100 nF (pas de 5,08 mm)

Semi-conducteurs :

LED1 = LED 3 mm
MOD1 = module BL600 - SA *Laird Technologies*

Divers :

JP1 = embase droite à 3 picots

JP2 = embase droite à 2 picots p. ex. *Molex* (9731148)

2 cavaliers

K1 = embase droite à 6 picots

S1 = micro bouton poussoir (1555985)

support pour pile CR123A* (1650670)

16 picots tulipes

coffret *Multicomp* G302 (1094697)

circuit imprimé nu 140190-1

**thermomètre Bluetooth 4.0 composé du circuit imprimé principal,
avec le module BL600-SA implanté réf.
140190-91**

* = voir texte

(références du catalogue Farnell entre parenthèses)



L'auteure par elle-même

Mon grand-père, que je voyais récupérer toutes sortes de choses, est à l'origine de mon engouement pour l'électronique. Grâce à lui, je n'ai pas eu peur de me brûler les doigts. Mon parcours commence au collège avec le TO7 et passe par un Brevet de Technicien Supérieur (avec un 20/20 en électronique), puis par l'obtention du diplôme d'ingénieur en informatique en 1992. En tant qu'ingénieur d'exploitation sur grands systèmes, ingénieur support serveurs Windows, et développeuse d'applications vba et vbnet, j'ai fait un peu de tout sauf Linux et les réseaux. Vers 2012, nostalgie des vapeurs d'étain : retour à l'électronique, avec une auto-entreprise alimentée surtout par mes projets avec Elektor. Je me spécialise dans les interfaces Bluetooth et Wifi avec les *smartphones* iOS et Android.

plus complexes comme I²C, SPI, CAN et UART. Ce qui le met donc aussi à la portée de débutants. Le programme peut être écrit dans n'importe quel éditeur de texte. Je recommande Notepad+, téléchargeable sur le site de Laird Technologies. Le programme est compilé et transféré par le programme gratuit *UWTerminal*.

Dans le téléchargement associé à cet article [2] se trouve le code source de mon programme JATEMP qui montre l'établissement de la liaison entre module et téléphone, l'initialisation des CA/N puis après une demi-seconde, l'envoi par le module des données au téléphone : tension de la pile, tension d'alimentation du pont diviseur de tension, tension aux bornes du diviseur de tension, sous la forme d'un télégramme comme celui-ci : PW3012V853C433. C'est l'app qui fait le calcul sur le téléphone.

Programmes iOS & Android

Laird Technologies propose au téléchargement le code source de l'application *BL600 Serial* pour iOS, mais, comme j'ai de l'expérience avec iOS, j'ai préféré écrire ma propre application [7] qui affiche deux informations : température et statut de la connexion. J'ai intégré alors les sources du programme *BL600 Serial* (*UARTPeripheral.c* et *DataClass.c*). Je n'ai pas créé de bouton de connexion et de déconnexion. J'ai remplacé cette action par une boucle (un *thread*) qui recherche le thermomètre JATEMP lorsqu'il est déconnecté (c'est le thermomètre qui se déconnecte) et se connecte.

Comme je n'avais aucune expérience de programmation Java, le programme sous Android m'a donné du fil à retordre. Merci au forum français *developpez.com* où j'ai trouvé de l'aide, surtout

pour créer une interface homogène à toutes les tailles de téléphones Android. J'ai créé ma propre application *BL600 Serial* (code et interface) mais sous un nouveau nom et sous une forme simplifiée [8], juste pour me connecter à mon thermomètre JATEMP. Le bouton *Scan and Connect* permettait de recevoir la chaîne de caractères des données brutes du thermomètre ; je l'ai remplacé par une boucle comme sous iOS.

Pour iOS et Android, le calcul de la température est conforme à la description faite ci-dessus, les détails en plus. Le fond d'écran (**fig. 4**) de l'application change selon la température mesurée.

Pour éviter de rester connectée indéfiniment au thermomètre, l'application iOS s'arrête après 5 min (ou si vous appuyez sur la touche *Home* du téléphone). Sous Android, l'application ne recherche plus le thermomètre après 5 min ou si vous appuyez sur la touche *Home* ; l'application n'est pas vraiment arrêtée.

Pour établir la communication avec le téléphone et connecter le thermomètre pour la première fois, il n'y a **rien** à faire. Tout est automatique.

Frigo = cage de Faraday

Vous pensez sans doute comme moi qu'un réfrigérateur est une cage de Faraday, c.-à-d. une enceinte étanche aux ondes radio. Lors de mes essais, début septembre 2014, c'était l'été indien à Paris : difficile d'obtenir durablement des températures inférieures à 20 °C. J'ai donc mis le thermomètre au frigo pendant un quart d'heure afin de faire un relevé à basse température au moment de l'en sortir. En attendant, je m'étais remise à figoler l'application Android en cours de mise point ; quelle ne fut pas ma surprise

de voir descendre rapidement la température affichée par le téléphone Android ! Pour vérifier, je suis allée faire le même test dans le frigo de ma voisine, où j'ai également vu la température descendre en direct jusqu'à 4 °C. J'en déduis que le joint autour de la porte laisse passer le signal Bluetooth (une caractéristique des réfrigérateurs que les fabricants d'électroménager gagneraient à mentionner dans leurs fiches techniques).

Conclusion

J'espère vous avoir convaincu de l'intérêt du BL600 et peut-être même de tenter l'aventure du four à refusion voire du soudage à la main si vous avez des doigts de fée. Pour la mise en œuvre d'une application basée sur le module BL600, l'inconvénient de la miniaturisation des contacts est largement compensé par les avantages d'une réalisation très facile et par la simplicité de son langage, le *smartBASIC*, et de la programmation par l'interface UART du module. Et ce n'est qu'un début. Entre-temps *Laird Technologies* propose le BL620, qui est le BL600 *master* : même matériel mais nouveau progiciel, capable de communiquer avec d'autres modules BL600. Cela ouvre de nouvelles perspectives qu'Elektor et moi ne manquerons pas de vous inviter à explorer prochainement.

[140190]

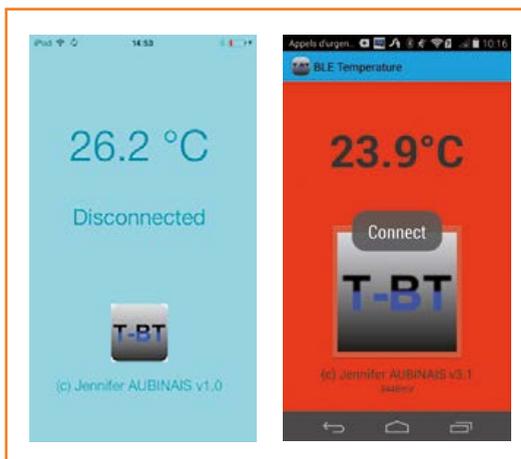


Figure 4. L'application du thermomètre sur l'écran de téléphones tactiles iOS et Android.



Figure 5. Le prototype de l'auteur, dans un boîtier étanche et muni d'un solide crochet vissé.

Liens

- [1] www.lairdtech.com/Products/Embedded-Solutions/Bluetooth-Radio-Modules/BL600-Series/
 - [2] www.elektor-magazine.fr/140190
 - [3] www.elektor.fr/bluetooth-thermometer
 - [4] https://laird-ews-support.desk.com/?b_id=1945
 - [5] www.youtube.com/watch?v=0YIKxtYwQiE
 - [6] www.elektor.fr/ft232r-usb-serial-bridge-bob-110553-91
 - [7] <https://appsto.re/fr/XTwnV.i>
 - [8] <https://play.google.com/store/apps/details?id=com.JA.bletemperature&hl=fr>
 - [9] www.aubinais.net
 - [10] CTN EPCOS www.epcos.com/inf/50/db/ntc_13/NTC_Leaded_disks_S891.pdf
www.epcos.com/blob/531152/download/2/pdf-standardizedrt.pdf
- NTC Thermistors / General technical information :
www.physics.queensu.ca/~robbie/ENPH354/NTC-Thermistors-Technote.pdf



générateur de fonctions avec Platino

sinus, dents de scie, rampes ... aux manettes



Sunil Malekar
(elektor.labs)

Un générateur de fonctions peut délivrer des signaux périodiques de différentes formes, fréquences et amplitudes. Allié à un multimètre, une alimentation et un oscilloscope, le générateur de fonctions est un outil indispensable à tout électronicien soucieux de concevoir et fabriquer ses circuits dans les règles de l'art, pour ne pas dire dans l'esprit Elektor. L'instrument à monter

Ce générateur de fonctions a été conçu pour satisfaire les applications aux besoins raisonnables. Ancré sur la carte à microcontrôleur Platino d'Elektor, il vous rendra service pour toute application impliquant du dépiage, de la synchronisation de circuits numériques, du filtrage audio basique, de la syntonisation ou autre essai de haut-parleur.

soi-même présenté ici repose sur la carte Platino d'Elektor [1] et a été inspiré par l'enrichissante série *radio logicielle avec AVR* [2].

Un jeu à deux cartes

Pas d'interminable liste de composants à dresser ici, ni de problèmes de conception à redouter, la carte Platino forme une base assez riche et solide pour que l'ajout de quelques extensions suffise à créer des hybrides analogiques/numériques. Comme le veut l'époque, notre générateur possède une face matérielle et une face logicielle, chacune chargée de garantir la stabilité et la précision des signaux de sortie. Le cœur du projet est un ATMEGA1284P qui exécute un micrologiciel taillé sur mesure. Le matériel comprend deux cartes : la Platino équipée d'un LCD, et une carte d'extension *Générateur de signal*, même si en toute rigueur les signaux sont produits par la Platino.

La carte Platino et le LCD

En conjonction avec la carte d'extension, la carte Platino contrôle et crée à la fois ses signaux et ceux produits par le générateur de fonctions. Le schéma de la **figure 1** montre les interconnexions entre les deux cartes. On voit clairement que la

Caractéristiques

- Microcontrôleur ATmega1284p sur carte Platino
- Afficheur LCD 4 x 20
- Alimentation : 18 à 20 V CC
- Bornes BNC standards pour les E/S
- Sorties :
 - Horloge, 10 MHz max., commutable entre 5 V et 3,3 V
 - Sinus, rectangle, triangle, dents de scie, dents de scie inversées, impulsions, formes arbitraires et aléatoires (bruit), 100 kHz max., environ 5 V max. pour 50 Ω
- Entrée : modulation de fréquence (125 kHz \pm 50 Hz pour radio logicielle)
- Commande : codeur rotatif avec poussoir, bouton *Retour*
- Mode *Setup* pour modes *Arbitrary* et *Clock*
- Mode *Normal* pour les formes d'onde définies
- Amplitude, fréquence et décalage modifiables en temps réel
- Un seul calibrage nécessaire

Platino et la carte du générateur communiquent par l'intermédiaire de connecteurs.

La carte Platino est équipée d'un LCD rétroéclairé de 4 lignes de 20 caractères pour l'affichage de menus, textes et autres données. Ici le codeur rotatif et son poussoir intégré servent en particulier à sélectionner les différentes formes d'onde. Un autre poussoir sert de touche *Retour* pour naviguer entre les menus du générateur de signal. Le rôle des broches formant l'interface entre la carte Platino et la carte analogique est indiqué dans le **tableau 1**.

La carte du générateur de signaux

La carte d'extension (**fig. 1**) comprend pour l'essentiel un convertisseur, un amplificateur, une alimentation, ainsi que des composants chargés de produire les différentes formes d'onde. Regardons un peu mieux les cinq sous-circuits qui composent cette carte.

CNA

Le CNA (convertisseur numérique/analogique) est constitué d'un réseau R-2R relié au PORTA de l'Atmega1284 de la Platino. Ce réseau (R1 à

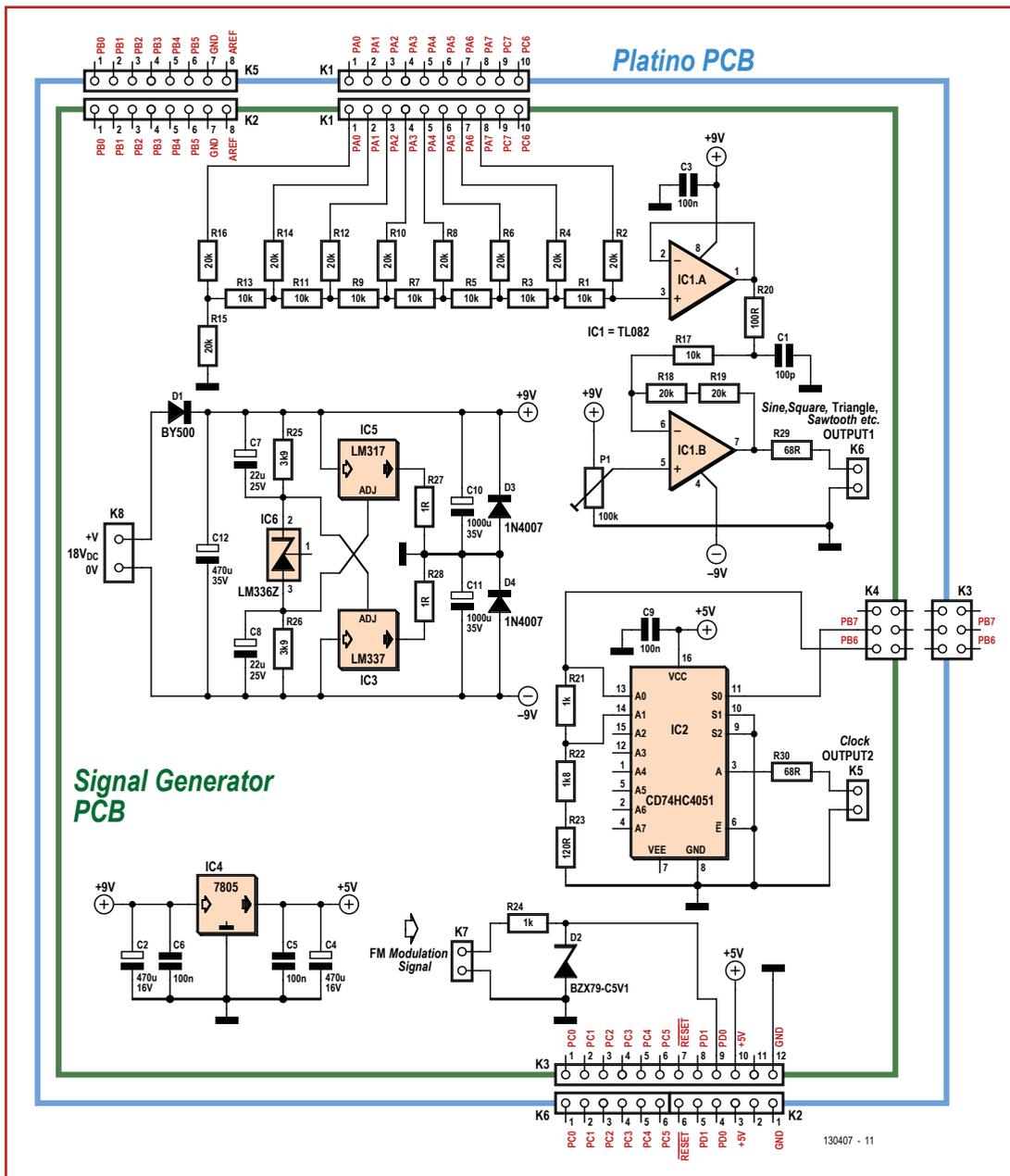


Figure 1. Schéma de la carte du générateur de signal et des connecteurs de la carte Platino.

**Tableau 1. Broches de l'interface Platino-Générateur**

port / broche	fonction
PORTA	relié au circuit R-2R du CNA
PB5	commande rétroéclairage LCD
PB3	bouton <i>Retour</i>
PB6	broche de sortie d'horloge
PB7	broche numérique pour sélection des 3,3 V / 5 V de l'horloge
PC0-PC2	codeur avec poussoir
PD0	entrée FM (5 V numérique)

R16) convertit en 255 étapes tout mot de 8 bits représentant une tension comprise entre 0 et 5 V en sa valeur analogique équivalente.

Amplificateur de ligne

La tension analogique du réseau R-2R alimente un circuit amplificateur séparateur formé de TL082

(IC1). L'AOP de ce circuit empêche toute charge excessive du réseau R-2R par la charge en K6. Grâce à R20 et C1, le signal de sortie d'IC1.A subit un filtrage RC qui supprime les composantes de haute fréquence. L'onde ainsi nettoyée est appliquée au second AOP, configuré avec un gain de 4 pour transposer les oscillations des signaux de 5 V dans l'intervalle -10 V à +10 V (max.). L'ajustable P1 relié à l'entrée « + » d'IC1.B sert à régler le niveau médian par rapport à la masse virtuelle. La forme d'onde programmable (sinus, rectangle, dents de scie, triangle) est disponible sur le connecteur de sortie K6.

Commutateur 3,3 V / 5 V de sortie d'horloge

C'est ce bon vieux démultiplexeur analogique 74HC4051 (IC2) qui s'occupe de sélectionner la tension pour la branche traitant la sortie d'horloge. L'entrée A0 est connectée à PB6, la ligne

Listage 1. Code Bascom (extrait)

```
Select Case Freq      'According to the frequency selects the prescaler value
  Case Is =< 80 : Config Timer1 = Timer , Prescale = 1024
    Pres = 1024
  Case 81 To 280 : Config Timer1 = Timer , Prescale = 256
    Pres = 256
  Case 281 To 2250 : Config Timer1 = Timer , Prescale = 64
    Pres = 64
  Case 2251 To 18000 : Config Timer1 = Timer , Prescale = 8
    Pres = 8
  Case Is => 18001 : Config Timer1 = Timer , Prescale = 1
    Pres = 1
End Select

Fraction = Freq
Fraction = Fraction / 1000000
Fraction = Fraction * 256
Fraction = Fraction * Pres
Fraction = Fraction / 20

Select Case Screen:
  Case 2 : For C = 0 To 1024
    C1 = C * Fraction
    C2 = C1
    If Signal = 5 Then
      C1 = DutyCycle / 100
      C1 = 256 * C1
      B = C1
    End If
    If C2 =< 255 Then
      Select Case Signal:
        Case 0 : Sig(c + 1) = Lookup(c2 , Sine )      'store sine values to signal variable
        Case 1 : Sig(c + 1) = Lookup(c2 , Square )   'store Square values to signal variable
        Case 2 : Sig(c + 1) = Lookup(c2 , Triangular) 'store Triangular values to signal variable
        Case 3 : Sig(c + 1) = Lookup(c2 , Sawtooth ) 'store Sawtooth values to signal variable
        Case 4 : Sig(c + 1) = Lookup(c2 , Inv_sawtooth) 'store Inv_sawtooth values to signal variable
        Case 5 : If B >= C2 Then
          Sig(c + 1) = &HFF      'store Pulse values to signal variable
        Else
```

de sortie de l'horloge Arduino. L'entrée associée A1 est reliée au diviseur de tension qui délivre 3,3 V lorsque 5 V sont appliqués sur la broche PB6. IC2 aiguille l'une ou l'autre de ces tensions vers la sortie en fonction d'une commande reçue d'Arduino via la broche PB7. La sortie d'horloge (onde carrée) est disponible en K5.

Entrée FM

Un signal externe peut être appliqué via le connecteur K7 pour moduler la fréquence. Le signal de modulation doit être numérique, avoir un niveau de 5 V, et sa fréquence doit être comprise en 0 et 20 kHz. La diode D2 protège l'ATmega contre les surtensions pouvant survenir sur la broche PD0.

Alimentation

Une tension pure ou brute de 18 V CC est appliquée via K8 à la branche traitant l'alimenta-

tion. Ces 18 V pourront p. ex. être délivrés par l'adaptateur d'un ordinateur portable, comme nous l'avions déjà suggéré pour l'alimentation de table régulée par Platino [3]. Grâce à un niveau de masse virtuel, la tension d'entrée est séparée en deux rails, un positif et un négatif.

Les résistances R25 et R26 et la référence de tension LM336 (IC6) divisent d'abord la tension d'entrée en deux « moitiés ».

Les régulateurs de tension IC3 (LM337) et IC5(LM317) forment le rail de masse virtuel pour la branche traitant l'alimentation. Ils permettent à la forme d'onde programmable d'osciller en théorie entre presque -9 V et +9 V (soit 18 V_{CC}) avec une alimentation bas de gamme de 18 V CC à un seul rail. Les diodes D3 et D4 ont un rôle protecteur. Le rail de masse virtuel pour les AOP est disponible à la jonction entre R27 et R28.

La tension de +9 V est aussi appliquée à un régu-

```

                Sig(c + 1) = &H00
            End If
        Case 6 : Sig(c + 1) = Arbitrary(c2 + 1) + 127          'store Arbitrary values to signal variable
        Case 7 : Sig(c + 1) = Rnd(255)                      'store Random values to signal variable for noise generation
        Case 8 : Enable Pcnt3                               'Enable pin change interrupt for FM
        Case Else : Sig(c + 1) = 127
    End Select

    C1 = Ampl / 10                                         'Amplitude calculation using user input
    C3 = 127 * C1
    C3 = 127 - C3
    C1 = Sig(c + 1) * C1
    C1 = C1 + C3

    C3 = Offset / 10                                       'Offset calculation using user input
    C3 = -1 * C3                                           'compensate 180 degree phase shift by OP-Amp
    C3 = C3 * 127
    C1 = C1 + C3                                           'Adding offset value to the signal

    If C1 > 255 Then                                       'Limit the signal value if overflow happens
        C1 = 255
    End If
    If C1 < 0 Then
        C1 = 0
    End If
    Sig(c + 1) = C1
    C21 = C - 1
    End If
Next
Case Else : Disable Pcnt3                                'Disable pin change interrupt at the FM input
    For C = 0 To 1030
        Sig(c + 1) = 127
    Next
    C21 = 1024
End Select

Flag = 0
Disable Timer0
End If
Return

```

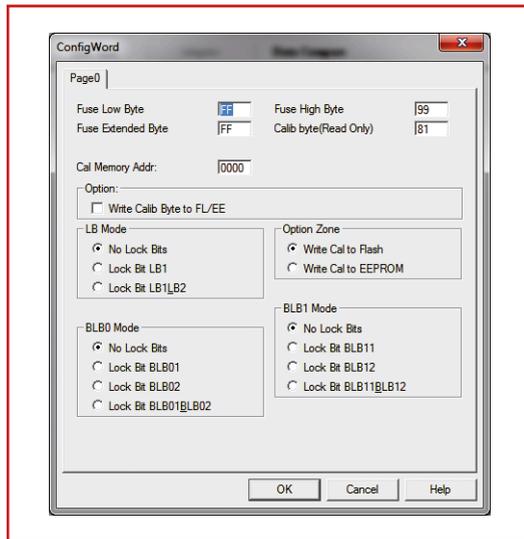


Figure 2. Oh tiens, la configuration des fusibles de l'Atmega1284P. Je vais pouvoir programmer mon micro sans avoir à passer par l'e-choppe.

lateur 7805 (IC4) chargé de délivrer une tension régulée au microcontrôleur et aux composants alimentés par le rail de +5 V.

Si une tension d'entrée de +20 V CC est appliquée, la tension divisée par rapport à la masse virtuelle oscillera en théorie entre -10 V et +10 V (max.).

Programme

J'ai utilisé une carte Platino pour réaliser l'ensemble du projet, de son idée à sa publication ici, et j'ai écrit le micrologiciel pour l'ATmega1284P en BASCOM AVR. Vous pouvez télécharger gratuitement le code source ainsi que tout le nécessaire logiciel depuis [4]. Le **listage 1** montre un extrait du code. La configuration des fusibles AVR,

que chacun redoute ou réclame, est reproduite sur la **figure 2**.

Le programme comprend deux modes, le mode *Normal* et le mode *Setup*.

Mode Normal

En mode *Normal*, vous pouvez sélectionner différentes formes d'onde, modifier leurs paramètres (amplitude, fréquence, etc.) et visualiser la sortie en temps réel en utilisant le codeur rotatif et le bouton Retour.

Le mode *Normal* comprend également une modulation de fréquence (FM). Elle produit une onde rectangulaire en fonction de la valeur entrée : si un niveau haut est lu sur la broche PD8 de l'ATmega, la sortie est 125 kHz +50 Hz, et 125 kHz -50 Hz si un niveau bas est détecté. Cette fonction permet de faire des essais avec une radio logicielle.

Mode Setup

Le mode de configuration *Setup* a deux sous-modes :

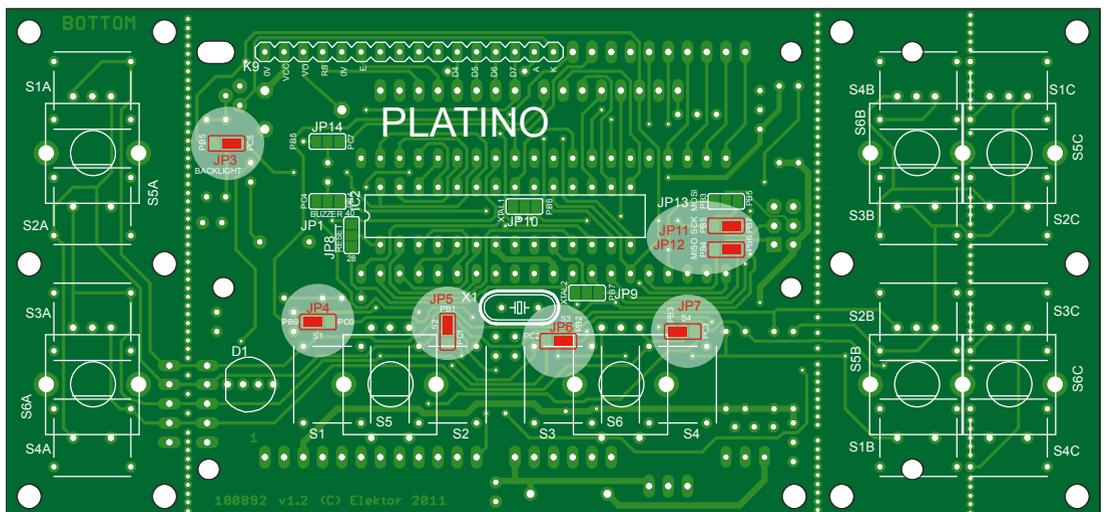
- Mode *Arbitrary* : l'utilisateur peut entrer n'importe quelles valeurs, puis sélectionner un signal arbitraire du mode *Normal* afin de produire un signal personnalisé.
- Mode *Clock* : l'utilisateur peut sélectionner la fréquence d'horloge produite parallèlement au principal signal de sortie.

La production de la fréquence repose sur un système d'indexation, autrement dit le temporisateur/compteur est paramétré en fonction de la

Tableau 2.
Configuration des cavaliers (ponts de soudure) Platino

cavalier	broche
JP4	PB0
JP5	PB1
JP6	PB2
JP7	PB3
JP11	PB7
JP12	PB6

Figure 3. La configuration de Platino pour cette application.



Liste des composants

Carte Générateur de signaux

Résistances

Défaut : 5 %, 0,25 W

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

R20 = 100 Ω

R21, R24 = 1 k Ω

R22 = 1,8 k Ω

R23 = 120 Ω

R25, R26 = 3,9 k Ω

R27, R28 = 1 Ω , 2 W

R29, R30 = 68 Ω , 1 %

P1 = ajustable 100 k Ω multitours, vertical

Condensateurs

C1 = 100 pF

C2, C4 = 470 μ F, 16 V, radial

C3, C5, C6, C9 = 100 nF, radial

C7, C8 = 22 μ F, 25 V, radial

C10, C11 = 1000 μ F, 35 V, radial

C12 = 470 μ F, 35 V, radial

Semi-conducteurs

IC1 = TL082ACP

IC2 = (CD)74HC4051

IC3 = LM337KCSE3

IC4 = MC7805

IC5 = LM317TG

IC6 = LM336BZ-5.0

D1 = BY500-800-E3/4

D2 = BZX79-C5V1

D3, D4 = 1N4007

Divers

K1 = barrette mâle à 10 contacts, SIL, droite

K2 = barrette mâle à 8 contacts, SIL, droite

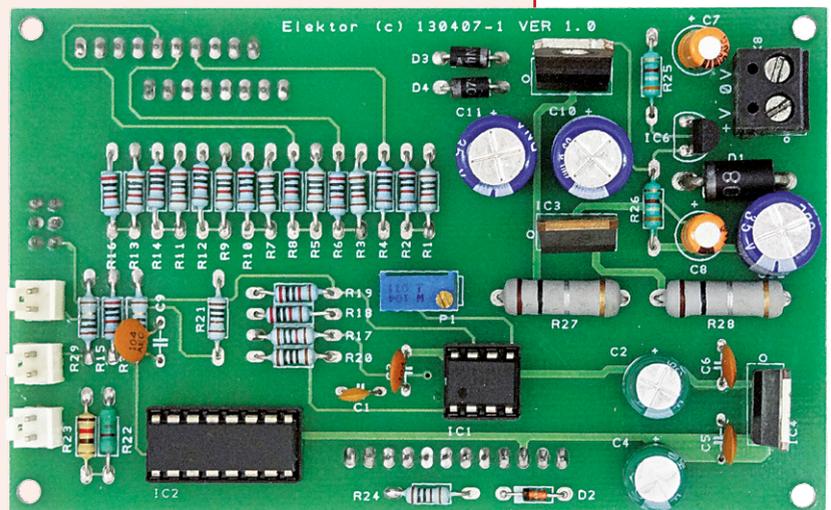
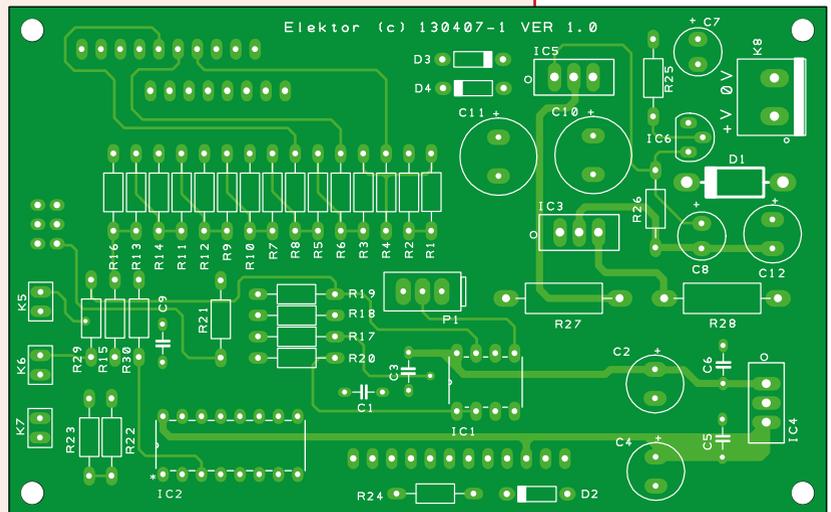
K3 = barrette mâle à 12 contacts, SIL, droite

K4 = barrette mâle à 6 contacts double rangée (2x3)

K5, K6, K7 = barrette mâle à 2 contacts, verticale, au pas de 2,54 mm

K8 = bornier à vis à 2 contacts pour circuit imprimé, au pas de 5,08 mm

support CI, DIP-16



support CI, DIP-8
 circuit imprimé 130407-1

Figure 4.
 Dessin de la carte
 d'extension. Que des
 composants traversants.

Configuration de la Platino*

Résistances

Défaut : 5 %, 0,25 W

R3 = 47 Ω

R4, R5, R6, R7, R10, R12 = 10 k Ω

R11 = 4,7 k Ω

P1 = aj. 10 k Ω hor.

Condensateurs

C1, C2 = 22 pF, 50 V, C0G/NP0, au pas de 2,54 mm

C5, C6 = 100 nF, 50 V, X7R, au pas de 5,08 mm

Semi-conducteurs

IC1 = ATMEGA1284P, programmé

T1 = BC547C

Inductances

L1 = 10 μ H

Divers

Support CI, DIP-40

LCD1 = LCD, 4x20, 5 V, rétroéclairé

S5A = codeur rotatif avec poussoir

X1 = quartz 20 MHz, C_L = 18 pF

K1, K2, K5 = barrette mâle à 40 contacts SIL, verticale

K4 = barrette mâle à 80 contacts à double rangée, verticale

K9 = embase femelle à 36 contacts, SIL, verticale

S4A = poussoir

* Cf. [3] pour une description détaillée de la carte Platino

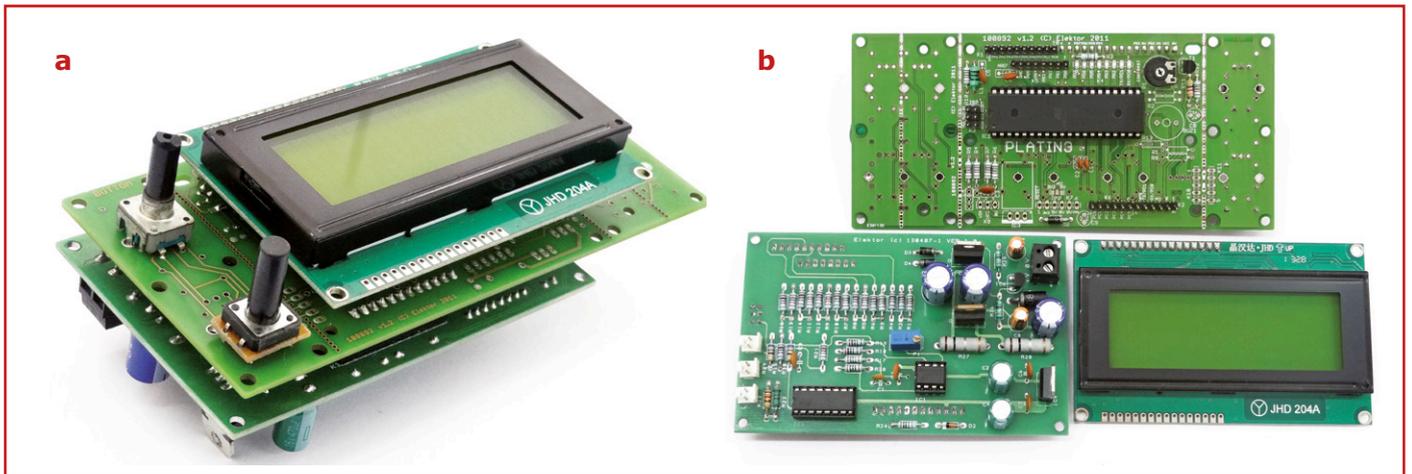


Figure 5.
(a) Les trois cartes empilées ; de haut en bas : LCD, Platino, Générateur de signal. (b) Les mêmes cartes, séparées.

fréquence sélectionnée : la valeur de ce paramètre sert d'indice pour extraire d'une table de correspondance la fréquence à produire. Cette valeur est également utilisée pour déterminer la vitesse et la période du signal.

Construction et essai

Vous devrez assembler sur la Platino : le LCD, le codeur rotatif et son poussoir intégré, un poussoir pour le bouton *Retour*, l'ATmega1284P, et enfin le reste des composants. Vous aurez également à réaliser des ponts de soudure pour créer les cavaliers du **tableau 2** (**fig. 3** et article [3]).

La construction de la carte d'extension devrait être facile puisqu'elle ne nécessite que des composants traversants (**fig. 4** pour le plan d'implantation). La carte est à double face et possède des trous métallisés. En cas de doute sur l'emplacement d'un composant, référez-vous aux différentes photographies de cet article. Notez qu'il y a aussi une liste des composants pour la carte Platino puisque nous avons dû la personnaliser pour réaliser ce projet.

Prenez le temps de bien vérifier vos soudures et le placement des composants avant de mettre

l'instrument sous tension via un adaptateur de 18 V CC relié à K8.

Avant de sélectionner une forme d'onde, calibrez une fois pour toutes le générateur en réglant l'ajustable P1 pour 0 V en K6.

La carte d'extension a été conçue et dimensionnée pour pouvoir être enfichée sur le dos de la Platino via les connecteurs K1, K2, K3 et K4. Le tout tient dans un boîtier *Bopla*. Le modèle de construction reprend celui déjà adopté pour l'alimentation de table Platino [3] : empilage du LCD, de la Platino et de la carte analogique (**fig. 5**), puis montage vertical de ces trois éléments derrière le panneau avant en aluminium (**fig. 6**). Ce panneau est équipé de 3 fiches BNC : sortie onde (F1 Out), sortie horloge (F2 Out), et entrée FM (MOD in).

Platino aux platines : ça gigote

Inutile d'expliquer comment sélectionner les formes d'ondes ou les fréquences, l'utilisation de l'instrument devrait être intuitive. En jouant avec le codeur rotatif et le bouton *Retour* vous découvrirez vite que vous pouvez même changer les paramètres en temps réel. Utilisez le bouton



Précédemment dans cette série

Pour vous convaincre de la flexibilité de la carte à microcontrôleur Platino conjuguée avec BASCOM, le laboratoire indien Elektor conçoit une panoplie d'appareils d'essai à monter soi-même. Le générateur de fonctions présenté ici est le deuxième instrument de la série ; le premier, *l'alimentation de table régulée par Platino*, a été décrit dans le numéro d'avril 2014. Restez à l'affût !

Retour pour revenir à un des écrans précédents du menu (fig. 7).

La tension de la sortie d'horloge peut être réglée sur 5 V ou 3,3 V depuis le mode *Clock* du mode *Setup*.

Pour finir en images, la **figure 8** montre quelques formes d'onde produites par l'instrument.

(130407 - version française : Hervé Moreau)

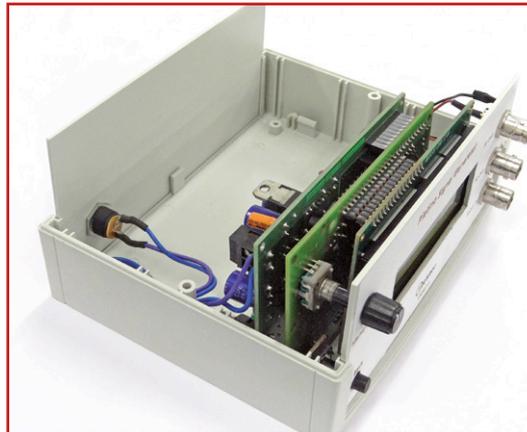


Figure 6.
Montage en boîtier *Bopla*
des trois cartes assemblées.

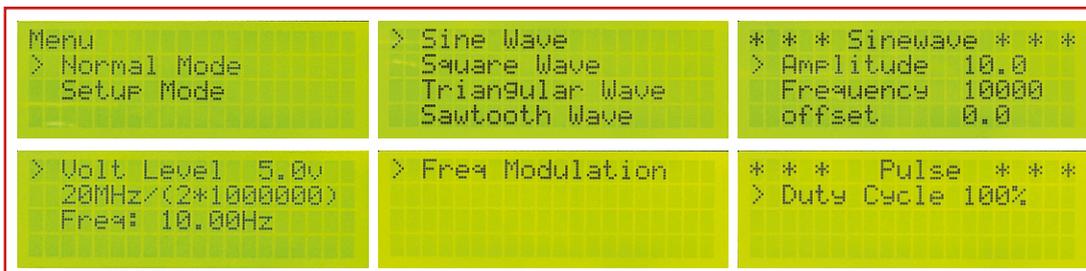


Figure 7.
Quelques menus messages
et menus menus.

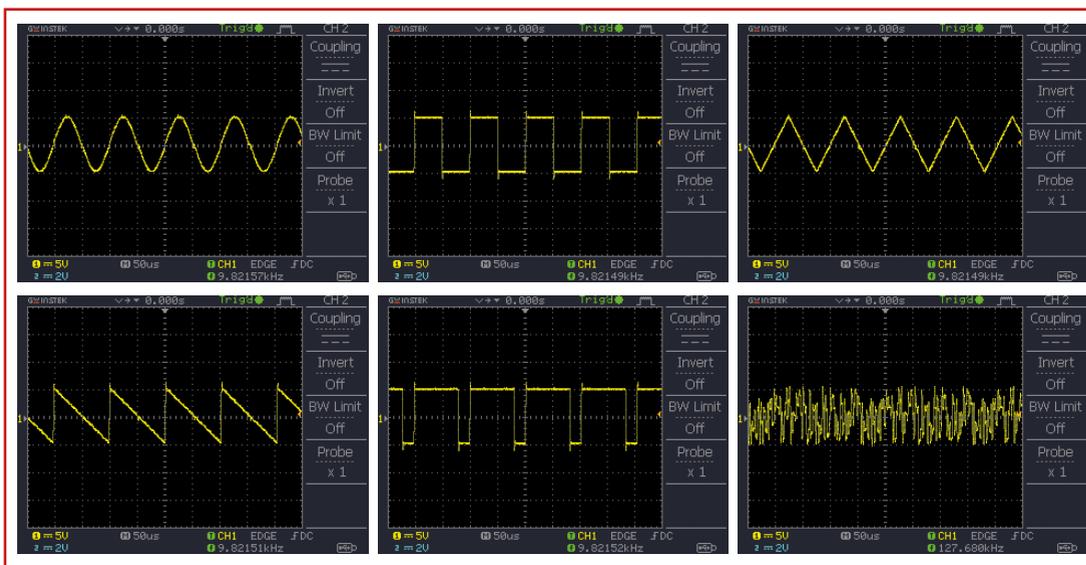


Figure 8.
Formes d'onde et bruit
produits par le générateur
de fonctions.

Liens

- [1] Platino : Elektor, octobre 2011, <http://www.elektor-magazine.com/fr/100892>
- [2] Radio logicielle avec AVR : Elektor, mars à septembre 2012 (premier article) www.elektor-magazine.com/fr/100180
- [3] Alimentation de table Platino : Elektor, avril 2014, www.elektor-magazine.com/fr/130406
- [4] Téléchargements : www.elektor-magazine.com/fr/130407

synthétiseur J²B

plateforme musicale numérique ouverte

Clemens Valens
(elektor.labs)



Ce projet est né de ma découverte du synthétiseur Atmegatron [1], un instrument construit par *Soulsby Synthesizers* autour d'un ATmega328 à 8 bits, le micro-contrôleur d'Atmel qui équipe la carte Arduino Uno. J'ai trouvé si intéressante la façon dont était conçu l'Atmegatron que ma curiosité s'est transformée en envie de construire le synthétiseur présenté ici.

Spécifications

- synthétiseur monophonique à 9 bits
- 32 formes d'ondes + ondes définies par l'utilisateur
- 15 types de filtres
- 2 générateurs d'enveloppe
- LFO avec 16 formes d'ondes
- arpégiateur de 15 arpèges
- 16 mémoires de sons
- 6 boutons de réglage
- MIDI
- sauvegarde/chargement de sons par MIDI
- microcontrôleur ARM Cortex-M3 à 32 bits (LPC1347 de NXP)
- 2 canaux de sortie
- code source et matériel ouverts

**Webinaire
gratuit sur le
synthétiseur
J2B**

Organisé par Elektor Academy et element14

Animateur : Clemens Valens

Date : 22 janvier, 16h00 HEC

Inscription : www.elektor.com/webinar



Avant toute chose, que je prévienne ceux d'entre vous qui seraient essentiellement intéressés par la façon dont le synthétiseur produit le son : je n'en dirai que quelques mots, je m'attacherai surtout à expliquer *comment j'ai adapté le code de l'Atmegatron à mon propre matériel*. Cela dit, si c'est le moteur de son qui vous intéresse, décortiquez la figure 2 et le code source [2].

L'Atmegatron

Dans l'Atmegatron, c'est le microcontrôleur (abrégé en μC ci-après) qui s'occupe de quasiment tout, de la synthèse du son aux interactions avec l'utilisateur. Il dialogue aussi avec d'autres périphériques via un port MIDI (*Musical Instrument Digital Interface*). Tout a été codé en C/C++ dans un croquis Arduino et tient dans la mémoire de programme de 32 Ko du μC . C'est une prouesse d'autant plus remarquable à mes yeux qu'Arduino peut créer pas mal d'octets supplémentaires lors de l'exécution du programme (*overhead*). Contrairement au code, ouvert et téléchargeable gratuitement, le matériel est hélas « fermé » (bien qu'il soit assez facile de le reconstruire en étudiant le logiciel).

Autre point intéressant, l'Atmegatron a été conçu pour pouvoir être utilisé en concert et est ainsi équipé de six potentiomètres permettant de modifier à la volée dix paramètres sonores. Ajoutez à cela deux boutons principaux de sélection des fonctions et des valeurs numériques, et vous avez 8 boutons pour jouer sur l'Atmegatron, qui plus est selon deux modes (« vert » ou « rouge »). L'instrument n'a aucun afficheur, les modes et les valeurs sont indiqués au moyen de LED. Contrairement aux synthétiseurs classiques qui n'offrent que des sons lisses et insipides à mon avis, l'Atmegatron dispose de plusieurs algorithmes capables de produire des sons distordus et râpeux. Qu'on l'aime ou pas, l'Atmegatron a un son bien à lui et ne saurait être comparé à un synthé ordinaire.

Le son est produit par modulation de largeur d'impulsion (MLI). Le rôle du convertisseur externe numérique/analogique (CNA) peut par conséquent se limiter à celui d'un filtre anti-repliement.

J'étais donc face à un moteur de son *open source*, commandé par huit boutons (six potentiomètres et deux codeurs rotatifs) et dont le code tenait dans 32 Ko de mémoire. Je me suis alors demandé si je ne pourrais pas embarquer le tout sur ma carte Elektor J2B créée il y a quelques années [3]. Je l'avais construite autour d'un ARM Cortex-M3, un μC de NXP doté lui aussi de 32 Ko de mémoire et de tous les périphériques utilisés par le synthétiseur, dont un excellent module pour la MLI. On peut en outre monter jusqu'à neuf codeurs rotatifs sur la carte, ou huit codeurs et un poussoir. Il manquait bien à J2B une EEPROM pour la sauvegarde des préréglages sonores, mais je m'étais dit que je trouverais une solution.

L'Atmegatron utilise des LED bicolores pour indiquer les modes, fonctions et valeurs sélectionnés par l'utilisateur. Puisque la carte J2B possédait un LCD, je me suis dit que j'allais remplacer ces LED par une interface sur écran.

Je décidai alors de procéder en trois étapes :

1. Transformer le croquis Arduino de l'ATmega328



Figure 1.
L'Atmegatron, le synthétiseur qui a inspiré ce projet.

- en projet Eclipse/LPCXpresso pour le LPC1343.
2. Remplacer les six potentiomètres analogiques par six codeurs rotatifs numériques.
 3. Substituer un LCD aux LED bicolores.

Étape 1

Adapter du code est plus facile lorsqu'il est bien structuré que s'il vous rappelle votre dernière assiette spaghetti-fromage. Le niveau d'abstraction du matériel est également déterminant. Un code qui appelle des fonctions pour modifier des registres et des périphériques est bien plus aisé à porter qu'un ensemble de lignes qui interagissent directement avec le matériel. Un fichier regroupant les fonctions utilisées est de même plus commode qu'un code dans lequel tout a été bourré à coups de pelle. Le code de l'Atmegatron est heureusement bien structuré, bien documenté et entièrement portable. D'ailleurs j'en ai traité une grande partie tout en regardant un film de Bruce Willis à la télévision. Une bonne part du travail a consisté à créer les fichiers d'en-tête nécessaires au projet Eclipse/LPCXpresso (les croquis Arduino peuvent être écrits sans ces fichiers).

Je suis tout de même tombé sur quelques écueils, p. ex. sur des types de données mal utilisés par le programme ou incom-

patibles entre les compilateurs GCC AVR et ARM. Mêmes bogues à chaque fois : débordement de données (variables qui ne tiennent plus dans l'espace mémoire alloué) et changements de signe impromptus (nombres positifs devenant négatifs et vice versa). Éviter ces problèmes est simple, il suffit d'utiliser des types de données bien définis, c.-à-d. qui indiquent clairement la taille en bits et la présence ou non d'un bit de signe. On déclarera p. ex. un entier non signé de 8 bits avec `uint8_t`, et non pas avec `unsigned char` ; on utilisera de même `int16_t` ou `int32_t` à la place de `int`, car la représentation interne d'un entier dépend de la plateforme. Mieux, on peut créer des types de données dont le nom indique clairement la nature de la variable, et s'en tenir à ce nom dans tout le code, p. ex. `sample_t` pour toutes les valeurs d'échantillonnage.

D'autres complications surgissent lorsque les matériels diffèrent. De ce point de vue les temporisateurs/compteurs (*timers*) sont relativement faciles à porter, car ils se paramètrent tous plus ou moins de la même façon, l'essentiel étant de bien comprendre leur fonctionnement. Mais que faire si la plateforme cible n'a pas une des fonctions exploitées par la plateforme source ?

Ou si deux périphériques se comportent de façon légèrement différente ? C'est justement ce que j'ai découvert avec les modules MLI du LPC1343 et de l'ATmega328. Résultat : un son distordu. Laissez-moi vous expliquer ça.

Sur l'AVR, le module MLI compare en permanence la valeur d'un compteur avec un seuil, et place le niveau de sa sortie en conséquence. C'est ce qui détermine le rapport cyclique du signal modulé. Supposons que la sortie MLI soit mise au niveau bas lorsque la valeur du compteur est supérieure ou égale au seuil, au niveau haut dans le cas contraire. Si vous amenez le seuil sous la valeur du compteur, la sortie MLI passe alors immé-

diatement au niveau bas. Voici le pseudo-code (PWM = MLI) :



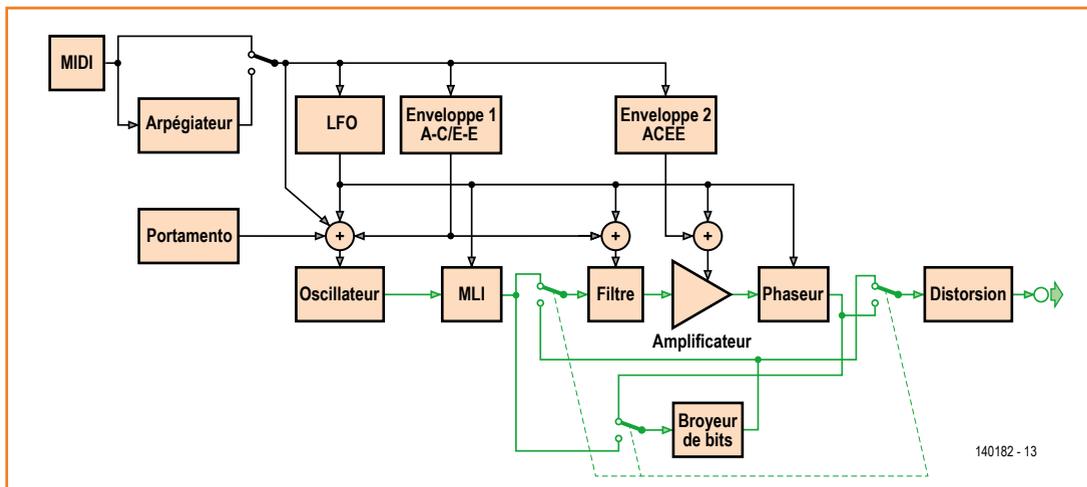


Figure 2. Schéma fonctionnel du moteur de son. Ce sont les blocs MLI, broyeur de bits (Bit crusher) et Distorsion qui donnent leur originalité aux sons produits.

```
counter = counter + 1
if (counter == max) then counter = 0
if (counter >= threshold) then PWM = 0
else PWM = 1
```

Le mécanisme est presque identique sur le LPC1343. *Presque*. Au lieu d'être commandée en permanence, la sortie MLI ne change de niveau que si le compteur et le seuil ont la même valeur. Une modification du seuil n'a donc pas d'effet immédiat, cet effet n'apparaît qu'au moment où le compteur atteint le nouveau seuil. Le pseudo-code est cette fois-ci :

```
counter = counter + 1
if (counter == max) then
{
  counter = 0
  PWM = 1
}
if (counter == threshold) then PWM = 0
```

Ouvrons donc le capot du moteur de son de l'Atmegatron pour comprendre les conséquences de cette différence.

Le moteur de son

Le moteur de son (**fig. 2**) calcule les sons par blocs de 32 échantillons. Un bloc correspond toujours à une période de la forme d'onde de sortie (ce qui signifie que le taux d'échantillonnage n'est pas constant). Les calculs se font à partir d'une table d'ondes contenant 32 échantillons d'une période de l'onde (32 prédéfinis, mais vous pouvez créer les vôtres). Ces échantillons sont filtrés,

traités, puis stockés dans un tampon de sortie. Ce tampon est mis à jour *aussi rapidement que possible* dans la boucle principale du programme, entendez par là qu'il s'agit d'une sorte de tâche de fond sans réelle priorité, exécutée quand le code en laisse la possibilité. Les paramètres modifiés dynamiquement, comme l'enveloppe sonore et les signaux de modulation, sont synchronisés avec un compteur de millisecondes de façon à être indépendants de la vitesse d'exécution plus élevée de la boucle principale.

La hauteur du son est commandée par un *timer* exécuté à une fréquence 32 fois plus élevée que celle du son, ceci pour compenser la taille de la table d'ondes. Lorsque le *timer* atteint sa valeur finale (fonction de la hauteur du son), l'échantillon suivant est extrait du tampon de sortie et le seuil MLI (rapport cyclique) est paramétré en fonction de la nouvelle valeur d'échantillonnage. Avec la façon particulière dont l'AVR effectue la MLI, le rapport cyclique suit les changements de cette valeur sans retard notable.

Sur le LPC1343, la mise à jour de la MLI se fait uniquement lorsque le *timer* MLI atteint le nouveau seuil, ce qui peut prendre jusqu'à une période si ce nouveau seuil est réglé (juste) sous la valeur actuelle du compteur MLI. Ce retard variable produit une interférence qui prend la forme sonore d'une suite de clics.

J'ai réussi à éliminer cette interférence en mettant à jour les échantillons à l'aide des interruptions du module MLI du LPC. La fréquence du signal MLI peut vous sembler élevée (140 625 Hz), mais le μC s'en sort sans problème. La routine d'in-

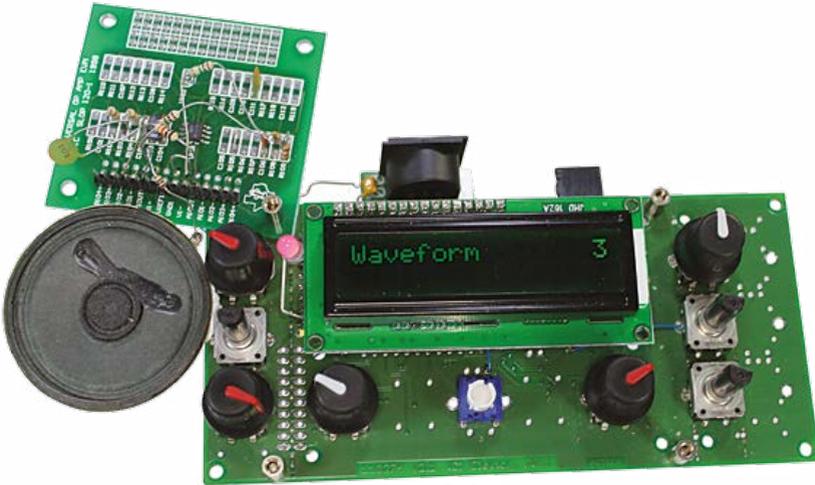


Figure 3. Le prototype de la carte J²B, ses 8 codeurs rotatifs et son LCD 2x16.

l'interruption du *timer* s'occupant de la hauteur du son parcourt encore le tampon d'échantillonnage, mais le seuil MLI n'est plus mis à jour directement avec la nouvelle valeur d'échantillonnage : cette valeur est stockée dans une variable intermédiaire que lit la routine d'interruption MLI pour mettre à jour le rapport cyclique. Les clics ont disparu grâce à cette mise à jour maintenant parfaitement synchronisée avec le signal MLI.

Il me manque une case

Alors que j'allais conclure le portage du code, je me suis aperçu que je n'avais pas assez de mémoire. Plutôt surprenant. L'AVR et le LPC n'ont-ils pas la même quantité de mémoire de programme (32 Ko) ? En plus le code LPC n'est-il pas compilé en code Thumb compact, censé économiser l'espace mémoire ? Le compilateur Arduino serait-il réellement bon ? Ou le code ARV vraiment efficace ? J'ai fini par comprendre que l'excès de

poinds de mon code provenait pour partie de la bibliothèque d'abstraction matérielle LPCXpresso, et pour partie de la bibliothèque *newLib* incluse par défaut dans les projets LPCXpresso. Outre qu'elle fait partie de la bibliothèque C standard, elle fournit également une prise en charge du débogage « à la printf », même en mode Release. Vous pouvez la supprimer si vous n'en avez pas besoin. L'option est bien cachée : allez dans les propriétés du projet, développez *C/C++ Build*, puis *Settings*. Ensuite sélectionnez *Managed Linker Script* sous *Tool Settings* (en déroulant d'abord éventuellement *MCU Linker*). Après quelques essais j'ai trouvé que la bibliothèque *Redlib* (*none*) donnait des résultats aussi bons que n'importe quelle autre bibliothèque *none*. J'ai ainsi pu libérer pas mal de mémoire, mais il me fallait encore implanter la prise en charge de l'EEPROM et une partie de l'interface utilisateur. L'étape 1 était tout de même franchie.

Étape 2

Un bête remplacement des potentiomètres par des codeurs rotatifs ne serait pas satisfaisant. Un potentiomètre peut être amené de 0 à sa position maximale d'un mouvement de doigt rapide, pas un codeur rotatif. Un codeur rotatif permet de contrôler très précisément la valeur d'un paramètre, mais il faut le tourner plusieurs fois pour parcourir sa plage de valeurs. J'ai donc songé à un mécanisme accélérateur qui lui donnerait la maniabilité du potentiomètre sans lui retirer sa précision.

Je suis d'ordinaire plus enclin à procéder par essais et erreurs, mais ici je m'y suis pris comme un ingénieur en blouse blanche : j'ai connecté un oscilloscope à mon codeur rotatif préféré (24 détentes/rotation) et mesuré les vitesses de rotation. J'ai ainsi découvert qu'en faisant tourner le codeur aussi vite que je le pouvais, la fréquence des impulsions s'élevait jusqu'à près de 100 Hz. Partant de là, j'ai écrit un algorithme qui mesurait cette fréquence et en déduisait un facteur d'accélération. Résultat, je pouvais parcourir l'intervalle allant de 0 à 255 en un seul tour rapide au lieu de dix.

Étape 3

Il me fallait maintenant remplacer les LED par un LCD. J'aurais pu les garder, mais pour cela il m'aurait fallu ajouter des duplicateurs de ports au μ C, et mon objectif était de m'éloigner aussi peu que possible de la conception de la carte J²B.



L'avantage d'un LCD est qu'il n'oblige pas l'utilisateur à compulsurer sans arrêt le manuel d'utilisation puisqu'il fournit déjà pas mal d'informations. Huit codeurs rotatifs impliquaient un LCD de 2x16, sinon je n'aurais pas eu assez de place sur la carte pour tout embarquer (**fig. 3**). Trouver les bons textes à afficher n'est pas facile, mais trouver la façon d'afficher les valeurs ou positions des six boutons l'est encore moins. J'aurais bien sûr pu me contenter d'afficher deux rangées de trois valeurs, mais la correspondance entre la position d'une valeur et son codeur n'aurait pas été très claire. Après avoir longuement cogité, j'ai trouvé une solution, plaisante me semble-t-il : des icônes représentant un curseur. Un LCD standard permet jusqu'à huit caractères personnalisés de 5x7, juste ce qu'il me fallait pour créer un curseur vertical à 7 positions. En plaçant deux de ces caractères l'un au-dessus de l'autre et en y ajoutant un caractère spécial pour les positions 0 et Maximum, j'ai obtenu un curseur à 16 positions. Un algorithme détecte si l'utilisateur manipule un bouton de réglage ou un codeur de fonctions/valeurs ; le programme peut alors commuter automatiquement entre deux pages et afficher l'information au bon moment. J'aurais préféré utiliser le LCD pour indiquer les modes rouge et vert de l'Atmegatron, mais j'ai finalement ajouté une LED bicolore.

Le Catatron

Quoique pas entièrement fonctionnel, mon prototype (fig. 3) prenait forme et je pouvais passer à la conception matérielle. Mon idée première était de placer les commandes du synthétiseur sur la carte J2B, mais je me suis vite aperçu que la trop grande proximité des encodeurs les uns avec les autres rendait leur manipulation malaisée. Comme une carte d'extension était prévue pour le filtre anti-repliement, l'interface MIDI, l'amplificateur et une EEPROM, je décidai finalement de concevoir une carte-mère et d'y placer aussi les codeurs. Cette carte-mère aurait pour carte-fille la J2B et serait assez grande pour y implanter des composants traversants de la bibliothèque de composants Passe-Partout d'Elektor Labs. Concevoir cette carte a été assez rapide. Grâce à l'outil gratuit FilterLab de Microchip, il ne m'a fallu que 15 s pour calculer que la fréquence de coupure du filtre de Tchebychev d'ordre 5, formant le filtre anti-repliement, valait 15 Hz (je n'entends pas les fréquences supérieures). Le reste m'a pris plus de temps, mais une dizaine

Comment sonne-t-il ?

L'Atmegatron ne prétend pas être un synthétiseur haut de gamme. On pourrait le qualifier de synthétiseur Casio des années 80 dopé aux stéroïdes. Il peut en effet produire toutes sortes de bip-bip d'ordinateur et sortir des sons râpeux, voire cacophoniques, grâce à des algorithmes spéciaux de distorsion. Cela dit il offre aussi des sons d'excellente qualité, et jouer avec est amusant. L'arpégiateur en particulier est une fonction appréciable. La qualité sonore du J2B est légèrement supérieure à celle de l'Atmegatron, car le LPC1347 permettait de faire de la MLI sur 16 bits, pour seulement 8 avec l'Atmegatron (le mode MLI sur 16 bits de l'AVR est trop lent pour cette application). J'ai ainsi pu ajouter un bit de profondeur MLI et faire du J2B un synthétiseur à 9 bits. J'ai aussi plus que doublé la fréquence MLI ; le filtrage anti-repliement est donc lui aussi meilleur (d'ordre 5, contre 3 pour l'Atmegatron), et la qualité sonore y gagne.

Cette qualité pourrait encore être améliorée de nombreuses façons. Les algorithmes du moteur de son ont tendance à tronquer leurs sorties à 8 bits. De ce point de vue c'est un peu faire honte au μC de 32 bits. Les tables d'ondes pourraient être plus grandes. Le filtrage actuel se fait en virgule flottante, ce qui est valable mais consomme temps et mémoire. Utiliser une arithmétique en virgule fixe libérerait de la capacité de calcul pour les autres algorithmes de traitement du son.

Autre idée d'extension, pouvoir utiliser un synthétiseur virtuel via le port USB, et/ou une liaison MIDI par USB.

Le synthétiseur J2B est une plateforme bon marché qui permet d'expérimenter des sons créés par ordinateur directement sur l'instrument. *Prog and Play* pourrait être sa devise.

de jours plus tard (soit le temps de fabrication du circuit imprimé) j'étais prêt à assembler mon nouveau prototype.

Je me suis alors rendu compte que ma conception était mauvaise : les connexions à réaliser entre la J2B et la carte-mère étaient trop nombreuses et, pire, la plupart d'entre elles étaient inaccessibles. Une cata... j'ai abandonné.

Je me retrouvais donc avec un circuit imprimé inutilisable et un problème de mémoire à résoudre. J'allais probablement devoir oublier la carte J2B et repenser entièrement la conception, alors pourquoi garder le LPC1343 ? Pourquoi ne pas utiliser le LPC1347 ? Il est doté de deux fois plus de mémoire (64 Ko), il possède une EEPROM de 4 Ko, et il utilise le même excellent chargeur d'amorçage USB. Alors, oui, pourquoi pas ?

MkII

Voilà, ma nouvelle conception (V2 ou *MakeII*) reposait maintenant sur un LPC1347 et comportait désormais des CMS (exception faite des connecteurs, codeurs et LED). En relisant le

Figure 4.
Schéma de la carte-mère du synthétiseur J2B.

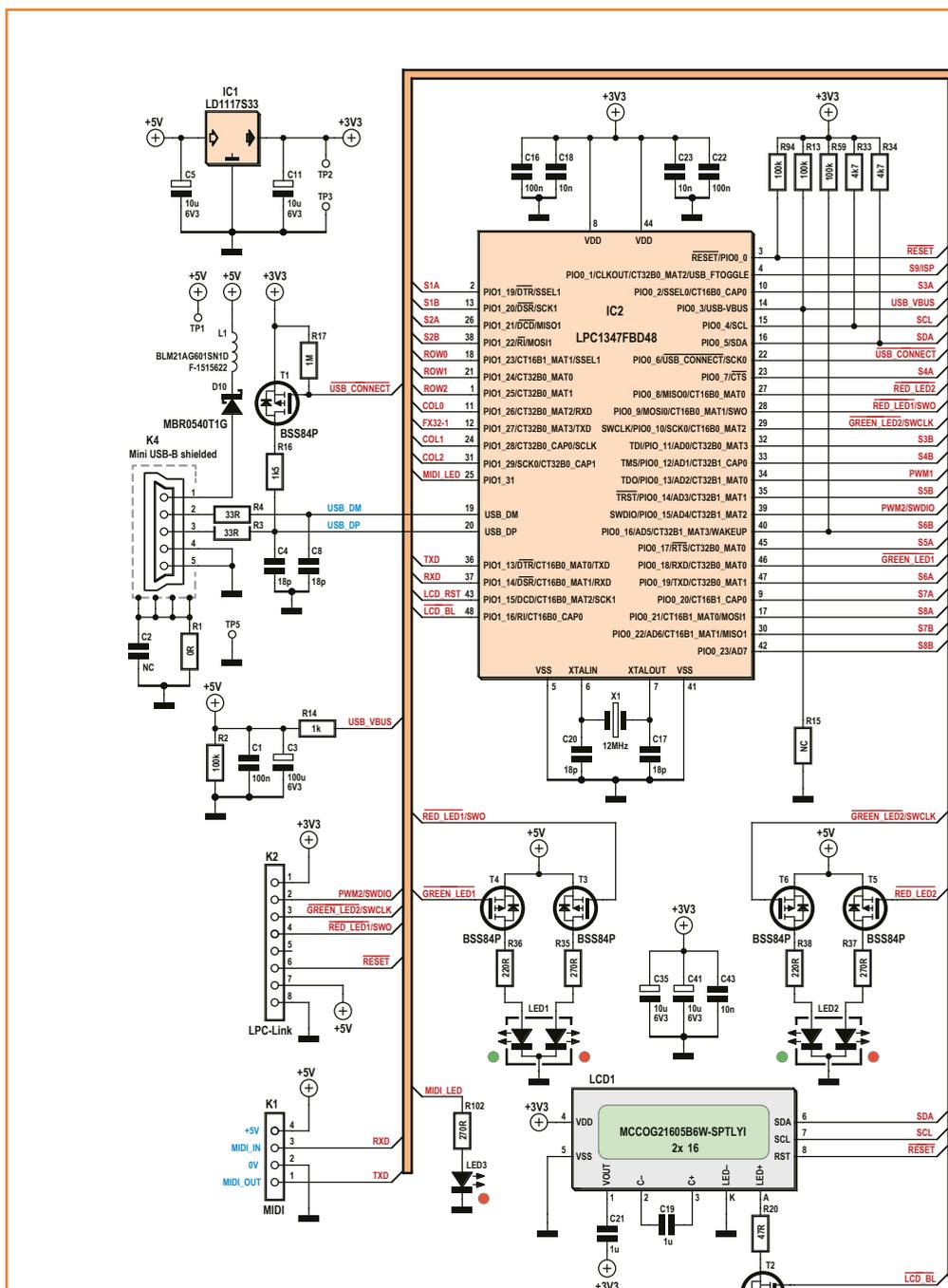
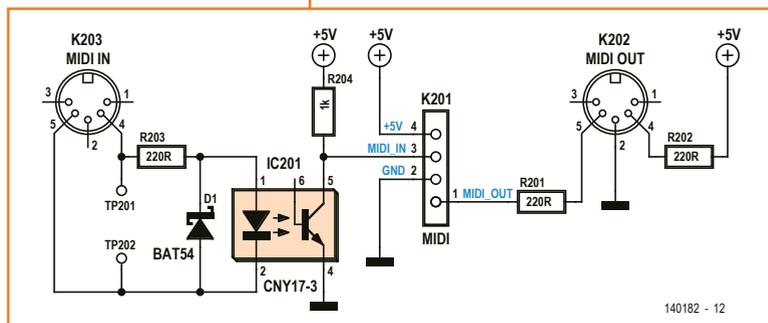
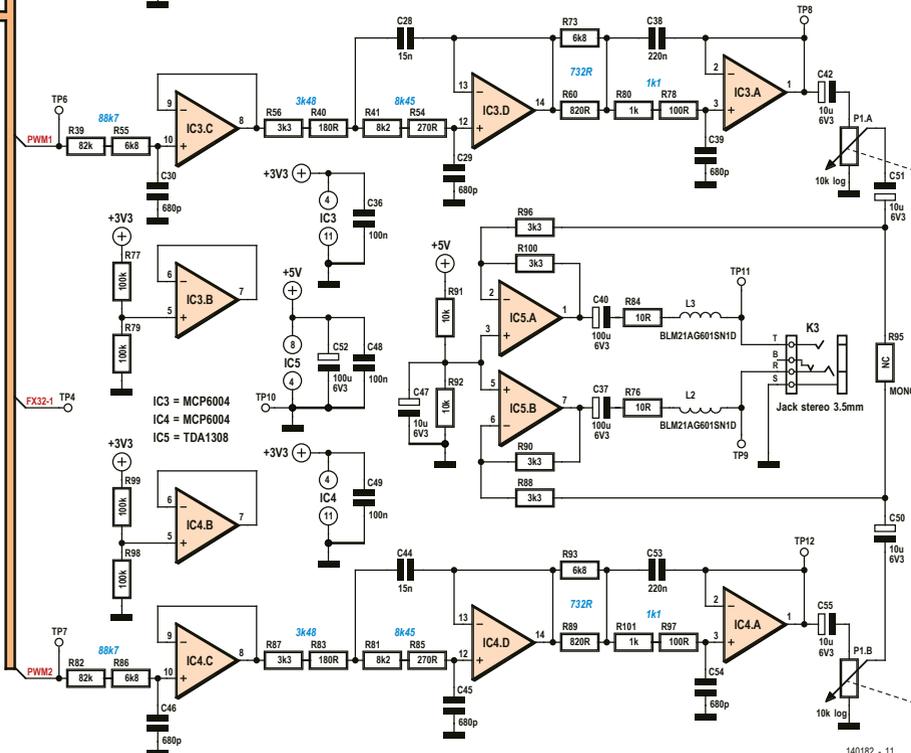
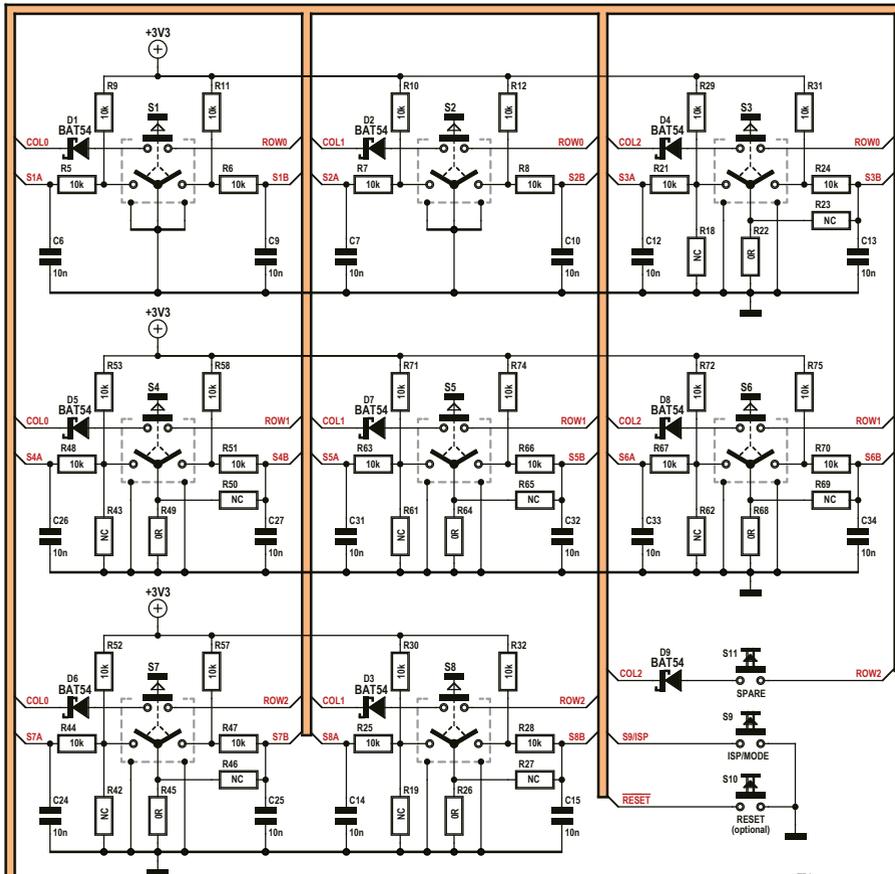


Figure 5.
Schéma de l'interface MIDI.
Elle possède sa propre carte en raison de la taille des connecteurs DIN. La flexibilité mécanique de la conception s'en trouve améliorée.



140182 - 12



140182 - 11



Figure 6.
Le synthétiseur J²B dans un boîtier en plexiglas découpé au laser.

manuel de l'Atmegatron je m'étais rendu compte que j'étais passé à côté de la LED indiquant l'activité de l'interface MIDI, donc j'en ai ajouté une. J'ai aussi ajouté un second filtre anti-repliement pour un fonctionnement sur deux canaux car

le LPC1347 possède plusieurs canaux par timer MLI. Après tout il aurait été dommage de ne pas chercher à exploiter toute la capacité de traitement d'un Cortex-M3 à 32 bits cadencé à 72 MHz. Le schéma du synthétiseur ne réserve aucune surprise (**fig. 4**). J'ai pu utiliser les 16 ports d'E/S disponibles, soit deux ports par codeur rotatif. Les résistances qui entourent les six boutons de réglage rendent le circuit flexible : leur en substituer d'autres permet de remplacer un codeur rotatif par un potentiomètre, car un côté des codeurs est relié à une entrée du convertisseur analogique/numérique du μ C. Considérez p. ex. le codeur S5, normalement monté avec R63, R64, R66, R71, R74, C31 et C32. Pour le remplacer par un potentiomètre, il faut monter R61, R65, R74 (0 ohm), et éventuellement C32. Le signal S5B (le curseur) est alors relié à l'entrée AD3 du convertisseur.

Les codeurs sont équipés de poussoirs, reliés entre eux de façon à former un clavier matriciel 3 x 3. La neuvième touche est une « touche de secours », sans fonction pour le moment.

Liste des composants

Carte principale

Résistances

Défaut : CMS 0805, 5 %, 0,1 W
 R1, R22, R26, R45, R49, R64, R68 = 0 Ω
 R2, R13, R59, R77, R79, R94, R98, R99 = 100 k Ω
 R3, R4 = 33 Ω
 R5-R12, R21, R24, R25, R28-R32, R44, R47, R48, R51, R52, R53,
 R57, R58, R63, R66, R67, R70-R75, R91, R92 = 10 k Ω
 R14, R80, R101 = 1 k Ω
 R16 = 1,5 k Ω
 R17 = 1 M Ω
 R20 = 47 Ω , 1206, 0,25 W
 R33, R34 = 4,7 k Ω
 R35, R37, R54, R85, R102 = 270 Ω
 R36, R38 = 220 Ω
 R39, R82 = 82 k Ω
 R40, R83 = 180 Ω
 R41, R81 = 8,2 k Ω
 R55, R73, R86, R93 = 6,8 k Ω
 R56, R87, R88, R90, R96, R100 = 3,3 k Ω
 R60, R89 = 820 Ω
 R78, R97 = 100 Ω
 R76, R84 = 10 Ω
 P1 = pot. 10 k Ω , stéréo, log.
 ! R15, R18, R19, R23, R27, R42, R43, R46, R50, R61, R62, R65, R69,
 R95 = non montées

Condensateurs

Défaut : CMS 0805
 C1, C16, C22, C36, C48, C49 = 100 nF
 C3, C37, C40, C52 = 100 μ F, 6,3 V, tant., boîtier B
 C4, C8, C17, C20 = 18 pF

C5, C11, C35, C41, C42, C47, C50, C51, C55 = 10 μ F, 6,6 V, tant.
 C6, C7, C9, C10, C12-C15, C18, C23-C27, C31-C34, C43 = 10 nF
 C19, C21 = 1 μ F
 C28, C44 = 15 nF
 C29, C30, C39, C45, C46, C54 = 680 pF
 C38, C53 = 220 nF
 ! C2 = non monté

Inductances

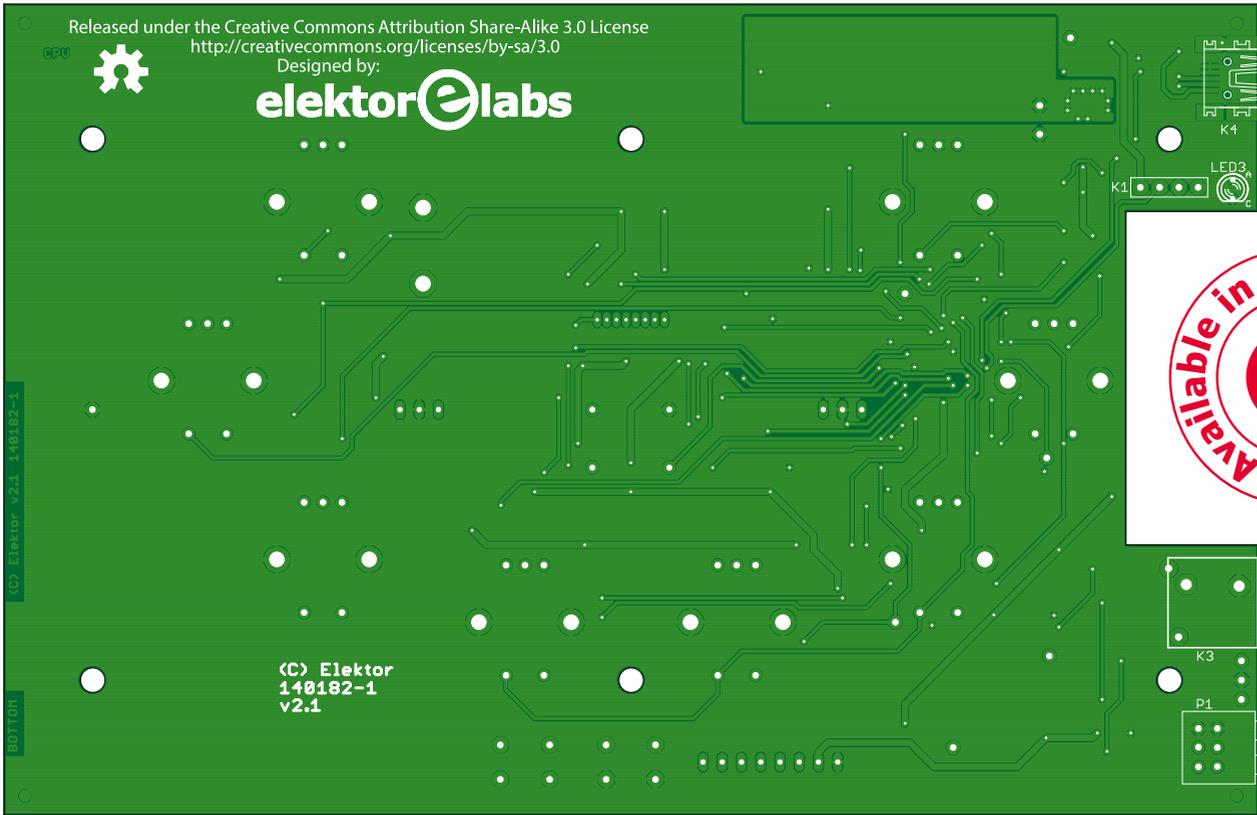
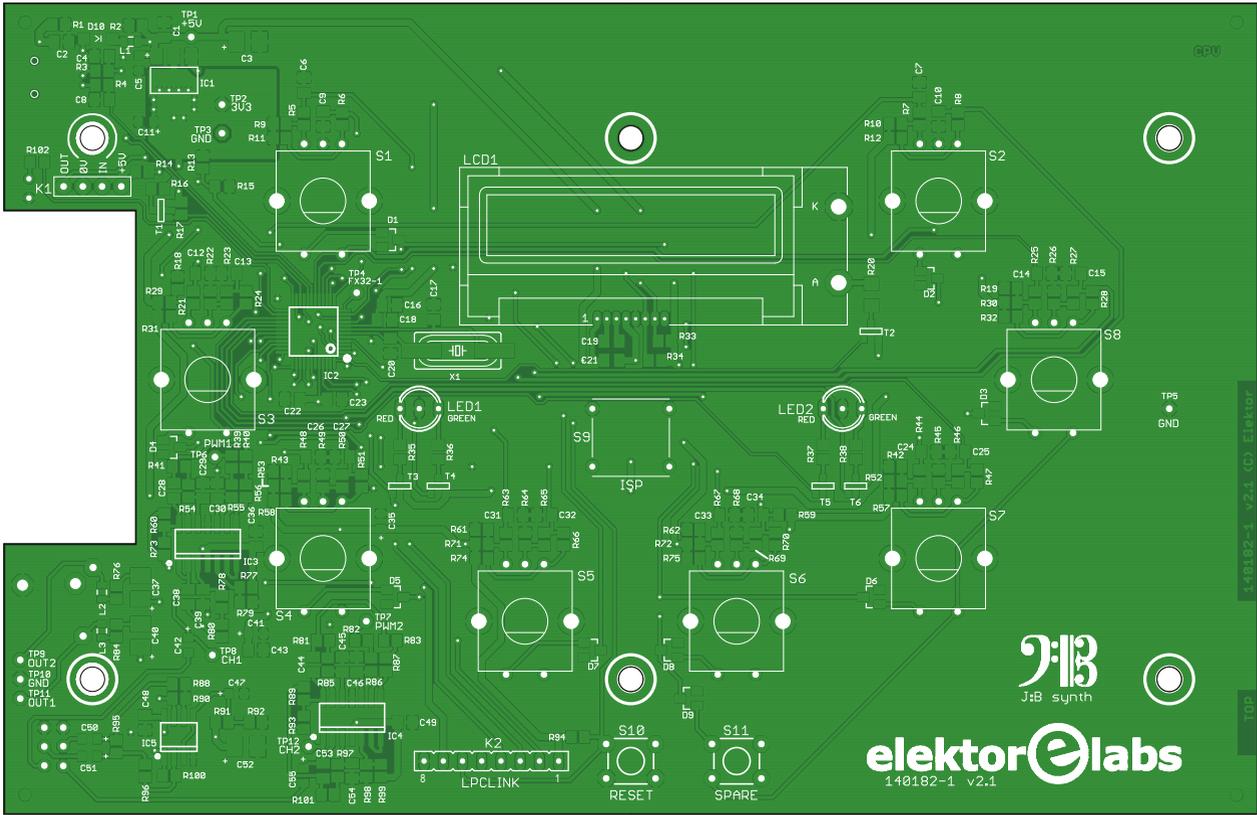
L1, L2, L3 = perle de ferrite, 0,21 Ω , 0,6 A, CMS 0805

Semi-conducteurs

D1-D9 = BAT54C (SOT-23)
 D10 = MBR0540T1G
 IC2 = LPC1347FBD48
 IC1 = LD1117S33CTR
 IC3, IC4 = MCP6004-I/SL
 IC5 = TDA1308T/N2
 LED1, LED2 = LED, bicolore rouge-verte, CC, 5 mm
 LED3 = LED, rouge, 3 mm
 T1-T6 = BSS84P (SOT-23)

Divers

K1 = barrette mâle à 4 broches, pas de 2,54 mm
 K2 = barrette mâle à 8 broches, pas de 2,54 mm
 K3 = jack 3,5 mm, stéréo
 K4 = connecteur mini USB-B, blindé
 S1-S8 = codeur rotatif
 S9 = poussoir, Multimec 3FTL6
 LCD1 = LCD 2x16, I²C, p. ex.. Midas MCCOG21605B6W-SPTLYI
 X1 = quartz de 12 MHz
 BOX1 = boîtier Hammond type 1597DGY ou découpé au laser
 carte principale 140182-1 (e-choppe)
 carte MIDI 140182-2 (e-choppe)



Un poussoir Reset est présent, de même qu'un poussoir pour les modes rouge/vert. Ce bouton sert aussi à mettre le μC en mode de mise à jour du micrologiciel. Les LED rouges/vertes sont commandées par des MOSFET pouvant drainer le courant des LED sans perturber le contrôleur. Le LCD est à interface I²C, une bonne chose selon moi puisqu'il est plus petit et moins cher qu'un LCD classique, nécessite moins de broches et a une hauteur parfaite pour utiliser les codeurs. Les filtres anti-repliement comprennent trois ampli-op et une grappe de résistances et de condensateurs. Les valeurs non-standards des résistances sont obtenues au moyen de deux résistances de la série E12. Deux ampli-op restent inutilisés, un par canal. Les potentiomètres de volume par lesquels passent les sorties des filtres sont isolés par deux condensateurs afin d'éviter les craquements dus à la tension CC. Certains audiophiles fronceront les sourcils en lisant ça, mais ici cette protection suffit. L'amplificateur pour casque stéréo devrait fournir assez de puissance à la plupart des applications. L'interface MIDI (**fig. 5**) a été placée sur une carte séparée car les connecteurs DIN étaient

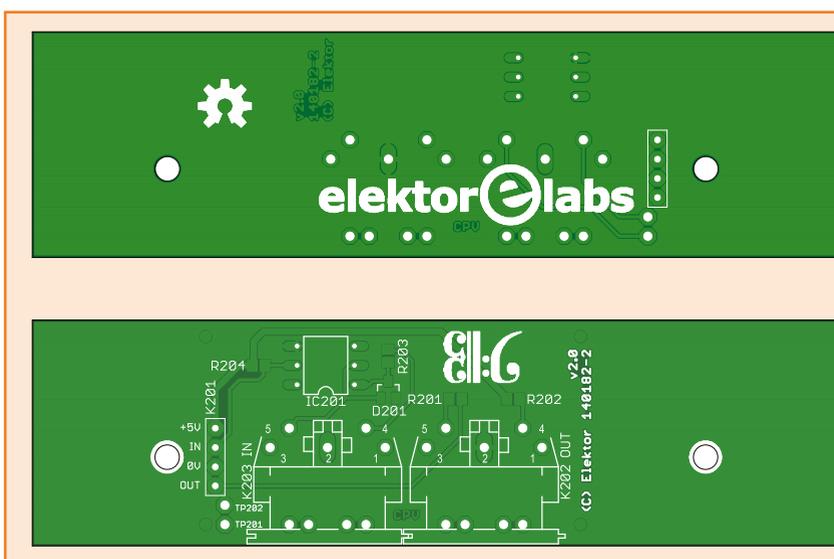
trop gros pour le boîtier choisi. Cette fois-ci ma conception mécanique était correcte, j'ai même dessiné et testé un gabarit de perçage [2]. Je me suis ensuite lancé dans une découpe par laser, ce qui a donné une seconde conception mécanique (**fig. 6**) dont vous pouvez aussi télécharger les plans [2].

Retour au code

Nouvelle conception, nouveau microcontrôleur : une nouvelle adaptation du code était devenue inévitable. Je n'ai pas expédié ça devant un autre Bruce Willis, ça a été plus compliqué que prévu. Je m'attendais en effet à n'avoir que quelques signaux à recâbler puisque le LPC1347 et le LPC1343 sont de la même famille. C'était sans compter sur une drôle de surprise : NXP a en effet décidé de réécrire la bibliothèque pour le LPC1347, qui possède en fait une architecture plus proche de celle des LPC11Cxx que de celle des LPC134x. En deux mots, les broches du LPC1347 ne sont pas compatibles avec celles du LPC1343, et leur code n'est pas non plus compatible à 100 %. Comme Bruce Willis, j'ai serré les dents et passé le code au crible pour régler ces problèmes de compatibilité.

Faites le vous-même

Le projet LPCXpresso/Eclipse du synthétiseur comprend trois sous-projets : un pour la bibliothèque de la puce, un pour la bibliothèque de la carte, et un pour le programme du synthétiseur. C'est un peu plus compliqué que nécessaire mais... c'est ainsi. Vous pouvez importer le paquetage



Liste des composants

Carte MIDI

Résistances

Défaut : CMS 0805, 5 %, 0,1 W
 R201, R202, R203 = 220 Ω
 R204 = 1 k Ω

Semi-conducteurs

D201 = BAT54C (SOT-23)
 IC201 = CNY17-3 (DIP-6)

Divers

K201 = embase femelle à 6 contacts, pas de 2,54 mm, verticale
 K202, K203 = embase DIN à 5 contacts, pour montage sur CI, 180°
 Carte MIDI : circuit imprimé 140182-2 (e-choppe)



en tant que fichier zip (c.-à-d. sans le dézipper d'abord). Après la compilation, vous trouverez un fichier BIN dans le dossier *Release* du projet. Pour programmer le µC avec ce fichier, reliez simplement le synthétiseur à un port USB libre tout en appuyant sur le poussoir du mode rouge/vert (le synthétiseur doit être hors-tension, le PC l'alimentera). Si tout se déroule normalement, Windows détectera une clé USB de 64 Ko contenant un fichier. Supprimez ce fichier et copiez le fichier BIN. Appuyez sur le bouton *Reset* si vous pouvez y accéder, sinon mettez l'instrument hors tension puis alimentez-le de nouveau. L'écran devrait vous accueillir avec un large « J2B Synthesizer ». Le synthétiseur est prêt lorsque ce message disparaît. Il ne vous reste plus qu'à brancher un clavier MIDI et un casque.

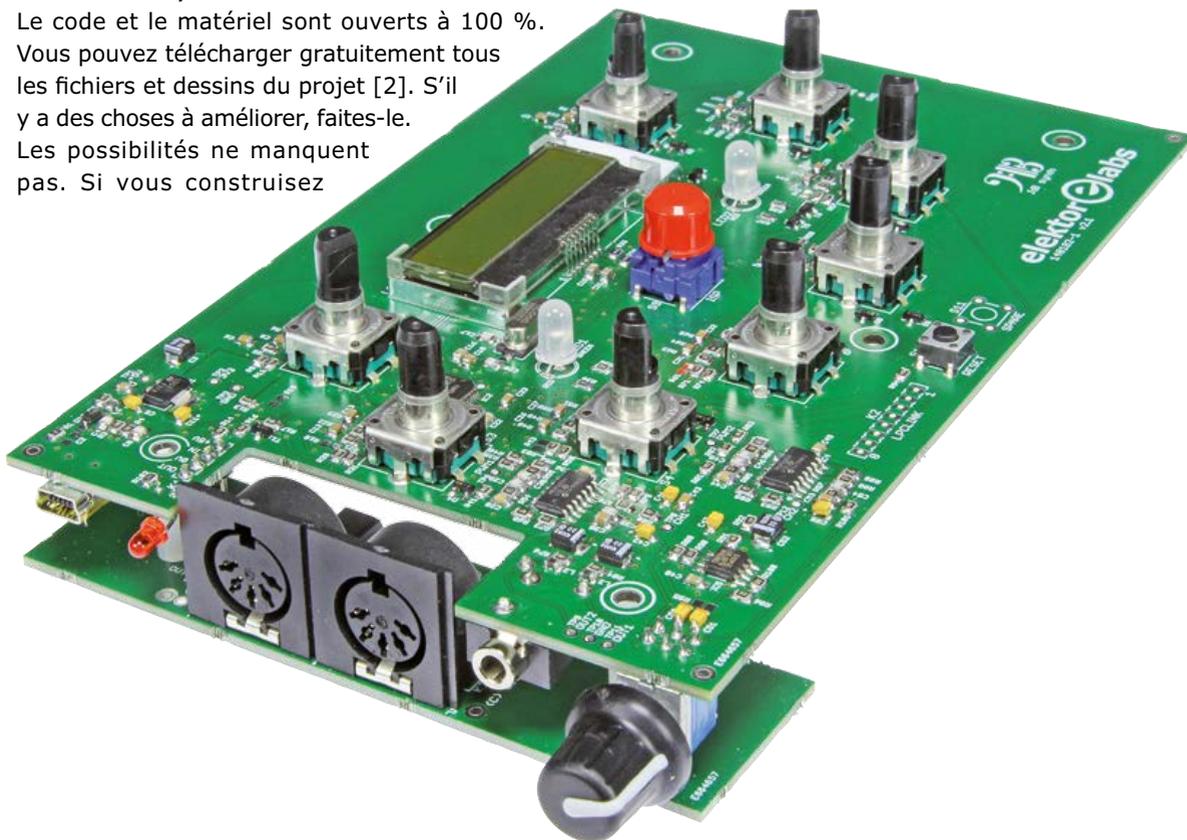
votre propre instrument, merci de m'en envoyer une photo ou ajoutez-la sur la page *elektor.labs* associée au projet [2].

Adapter le code de l'Atmegatron m'a permis de profiter de son *User Manual* et de l'utilitaire appelé *Librarian*. Disposer du guide de l'utilisateur m'a vraiment fait gagner du temps. Téléchargez-le [1] et lisez-le attentivement. Il est complet. Le synthétiseur J2B possède tous les boutons de réglage de l'Atmegatron, à l'exception du réglage de la tonalité. Les six poussoirs absents de l'Atmegatron permettent d'initialiser rapidement la valeur du paramètre correspondant.

(140182 - version française : Hervé Moreau)

C'est ouvert, faites comme chez vous

Le code et le matériel sont ouverts à 100 %. Vous pouvez télécharger gratuitement tous les fichiers et dessins du projet [2]. S'il y a des choses à améliorer, faites-le. Les possibilités ne manquent pas. Si vous construisez



Liens

- [1] Atmegatron : <http://soulsbysynths.com/>
- [2] Téléchargements : www.elektor-magazine.com/140182
- [3] J2B : module IHM universel à ARM Cortex-M3. Elektor sept. 2011
www.elektor-magazine.com/110274
www.elektor-labs.com/node/3832

VariLab 402 bis

la carte microcontrôleur/affichage



Ton Giesberts (elektor.labs)

Les fonctions de service de l'alimentation VariLab 402 sont l'apanage d'une carte à microcontrôleur qui veille au réglage des tensions et courants demandés tout en surveillant les paramètres essentiels aux endroits stratégiques du circuit. La communication est assurée par un afficheur LCD à quatre lignes, mais elle est aussi possible via un bus USB, avec un PC.

Pour être complète, l'alimentation à tout faire du numéro de décembre dernier [1] a besoin d'une unité spéciale de commande et d'affichage de la tension, du courant et autres données. C'est un μC ATxmega128A4U-AU qui préside à toutes ces fonctions. Des convertisseurs A/N et N/A embarqués permettent de réduire la taille du circuit complet.

Description du schéma

C'est bien cet ATxmega, IC1 dans le schéma (fig. 1), qui gouverne la carte μC /affichage dont l'alimentation symétrique de $\pm 5\text{ V}$ provient du circuit imprimé de l'alimentation VariLab. Comme le μC travaille sous $3,3\text{ V}$, le régulateur de tension fixe IC3 y pourvoit.

Toutes les tensions de mesure et de réglage se trouvent sur le connecteur K3, ainsi que les tensions d'alimentation nécessaires au circuit. Le convertisseur interne ADC3 reçoit le signal représentatif de la tension de sortie de l'alimentation V_{out} et ADC4 le signal du courant de sortie I_{out} , tamponné par IC2D. La résistance R5 adapte la

mesure de la tension de sortie à la pleine échelle de l'ADC3.

Le μC est aussi informé des tensions V_{smps} et $48V_{\text{in}}$ numérisées par ADC5 et ADC6. La résistance R6 fait en sorte que l'ADC5 obtienne $1/20^{\text{e}}$ de V_{smps} pour rester dans les limites de ses possibilités d'entrée. En outre, les diodes D2 à D5 protègent les entrées des convertisseurs d'éventuelles tensions trop hautes ou négatives.

Le réglage de la tension et du courant maximum de sortie s'opère avec deux tensions, de 0 à 4 V pour la tension, de 0 à 2 V pour le courant. Les convertisseurs N/A disposent d'une source de référence externe de $2,5\text{ V}$ prodiguée par un bon vieux LM336 (IC6), parce que la référence interne du μC n'est pas assez précise. Au moyen de P1, il est encore possible de fixer exactement cette référence, mais provisoirement, on le met au maximum en attente de l'étalonnage à l'aide du logiciel. Les deux amplificateurs opérationnels IC2A et IC2B multiplient par deux les tensions de sortie des CN/A pour qu'elles atteignent un maximum théorique de 5 V.

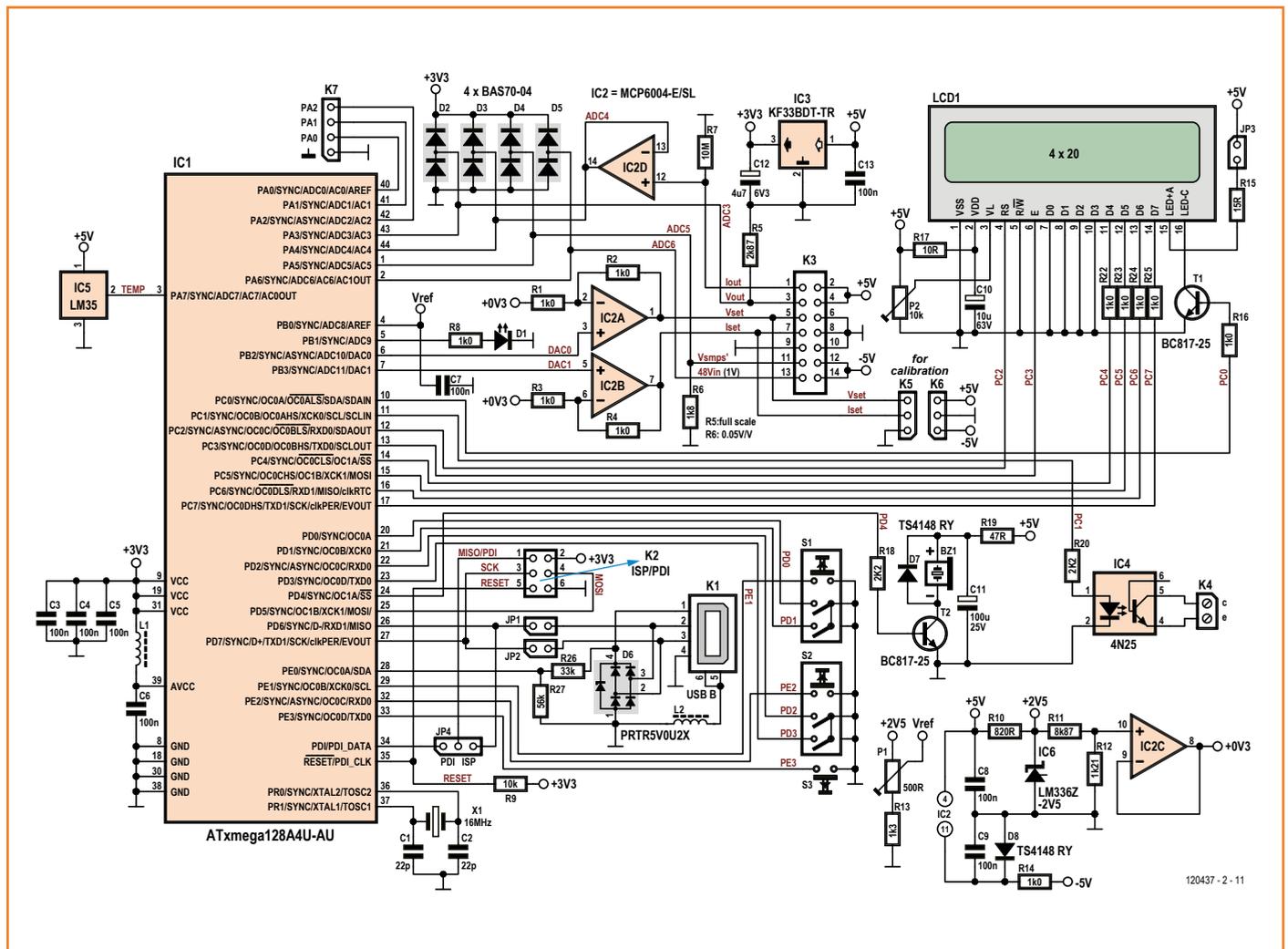
Un inconvénient des CN/A internes est le résidu de tension de sortie qui n'arrive pas jusqu'au zéro réel, les caractéristiques du fabricant révèlent un minimum de 0,15 V. Or, nous voulons absolument un réglage jusqu'à 0 V. Aussi avons-nous prévu une tension de décalage de 0,3 V sur les deux amplificateurs opérationnels pour garantir que leurs sorties descendront bien à 0 V. Cette tension de 0,3 V est prélevée sur la référence de 2,5 V par le diviseur R11/R12, puis tamponnée par IC2C.

Le choix s'est porté sur une puce quadruple ampli op MCP6004-E/SL de Microchip dont les entrées aussi bien que les sorties peuvent rejoindre les tensions d'alimentation (*rail-to-rail*), lesquelles s'élèvent à +5 V/-0,6 V (celle-ci via D8), vu que leurs alimentations ne peuvent pas excéder 6 V. Une manière de s'assurer encore d'un réglage jusqu'au zéro volt.

L'ATxmega pilote l'affichage LCD à quatre lignes en mode à 4 bits, sous 5 V. Les niveaux logiques fournis par le µC ne font bien sûr que 3,3 V, mais ils suffisent à garantir un fonctionnement fiable de l'afficheur. Le contraste se règle par P2, alors que T1 commute l'éclairage d'arrière-plan (si JP3 est en place). Comme il n'y a pas de radiateur prévu pour le stabilisateur de 5 V, il n'est pas souhaitable de laisser l'éclairage consommer en permanence. Avec la version actuelle du logiciel, il ne s'allume que quand la limitation de courant entre en action (si JP3 est inséré).

Deux codeurs rotatifs avec poussoir (S1 et S2) servent à procéder aux différents réglages. Il y a en outre le poussoir S3, entre autres pour la mise sous et hors tension de la sortie. S1 est affecté au réglage de la tension de sortie et S2 à la limitation de courant.

Figure 1. Schéma de la carte µC/affichage. Un ATxmega128 s'y affaire à toutes les tâches de mesure et de réglage.



Une indication acoustique par le vibreur BZ1, commandé par le μC assisté de T2, permet de signaler, entre autres, une limitation de courant. La LED D1 s'éclaire quand la sortie de l'alimentation est activée. C'est IC5 qui assure la thermométrie dans le boîtier et l'on peut en afficher la valeur sur le LCD. Ce LM35 est implanté directement sur le circuit imprimé, quand il sera dans le boîtier, il jouxtera le radiateur de l'alimentation sur l'autre circuit imprimé. Si vous préférez le voir ailleurs, allongez ses broches par des fils. Il y a encore une sortie à isolation galvanique (IC4) disponible en prévision d'une commutation supplémentaire, fonction actuellement en friche. Une liaison avec un PC par bus USB est prévue de manière à envoyer des données de mesure au PC ou lui permettre de régler la tension et le courant à partir d'un programme de terminal. Le connecteur USB est relié à l'ATxmega par l'intermédiaire des cavaliers JP1 et JP2, donc, si nécessaire, vous pouvez l'en isoler complètement. D6 protège le μC de tensions trop élevées sur le connecteur USB. Le diviseur R26/R27 est là pour empêcher le programme du μC de se planter en cas d'interruption inopinée de la liaison USB. De toute façon, la programmation du μC s'opère via K2. Le cavalier JP4 choisit le mode de programmation : ISP ou PDI. Il est préférable d'utiliser le mode PDI avec le programmeur AVRISP mkII d'Atmel.

Les tensions Vset, Iset et la masse arrivent sur K5 pour en faciliter la mesure pendant l'étalonnage. En cas d'utilisation de la carte d'affichage sans Varilab, K6 permet de l'alimenter en symétrique $\pm 5\text{ V}$ (consommation : $+50/-5\text{ mA}$, avec l'éclairage LCD : $+105/-5\text{ mA}$). Et K7 donne accès à quelques lignes de port vacantes du μC : PA0, PA1 et PA2.

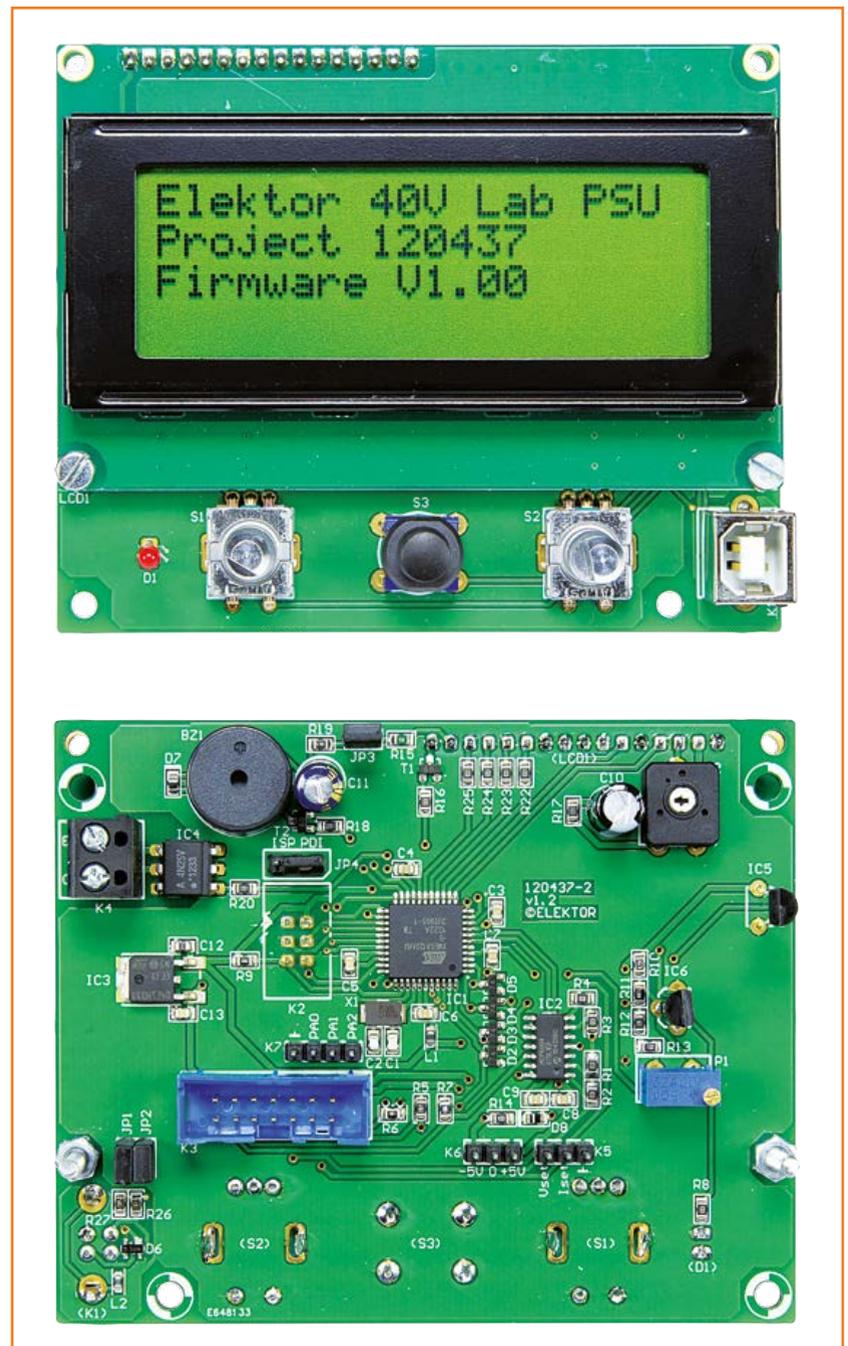
Construction

Les dimensions du circuit imprimé (fig. 2) sont calculées au plus juste pour soutenir l'affichage à quatre lignes et les organes de commande (fig. 3). Il faut dès lors implanter des composants sur les deux faces, la plupart du côté sérigraphié. On y trouve tous les CMS, les connecteurs et les composants traversants. Viennent de l'autre côté tous les boutons-poussoirs, les deux codeurs rotatifs, la prise USB montée verticalement, le témoin à LED et bien sûr l'afficheur lui-même. Suivez la liste des composants, il serait risqué de s'en écarter. Pour l'installation des CMS, la même mise en garde que pour le grand circuit

imprimé s'impose : il faut une certaine expérience pour réussir à les souder à la main, sinon, mieux vaut s'abstenir ou faire appel à un électronicien confirmé comme tuteur.

La face des organes de commande est plus accommodante, les composants y sont tous passibles d'un soudage normal. On monte l'afficheur LCD sur une embase à 16 broches ou à l'aide de petits fils

Figure 3. Les deux faces du circuit imprimé de la carte μC /affichage, montage terminé. L'afficheur se situe logiquement à proximité des organes de commande.



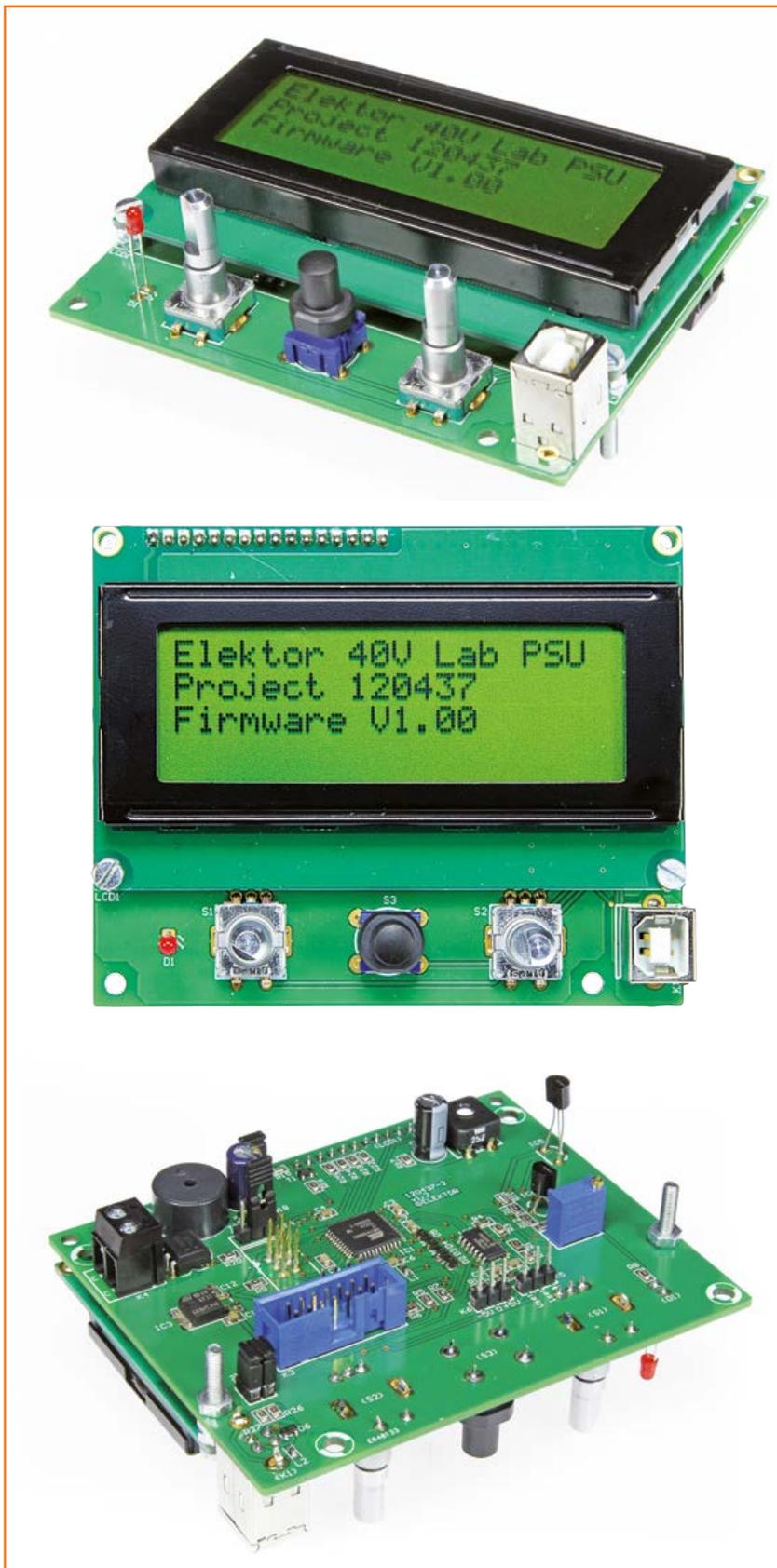


Figure 4.

Voyez ici comment fixer le LCD sur la carte au moyen de boulons M3 et d'écrous supplémentaires.

soudés, quelques millimètres au-dessus du circuit imprimé, en veillant à ce qu'il n'entre en contact avec aucun des conducteurs de la carte. Le bas du LCD sera solidement fixé au moyen de boulons M3 sur lesquels on intercalera deux écrous pour atteindre la bonne distance avec la carte (**fig. 4**). Pour la partie supérieure, ce n'est pas nécessaire, les fils de liaison ou les broches suffiront au maintien stable et à l'écartement voulu.

Quand le montage est terminé, il faut encore installer les cavaliers JP1 et JP2 pour le bus USB, JP3 pour l'éclairage arrière et JP4 pour choisir le mode PDI de programmation. La carte est alors prête à fonctionner en combinaison avec le circuit imprimé d'alimentation.

C'est un câble plat à 14 conducteurs qui assure la liaison entre le connecteur K3 de la carte et son homologue et homonyme du circuit imprimé d'alimentation. Pour éviter l'induction de parasites, mieux vaut qu'il soit aussi court que possible. Il n'y a pas d'autre connexion à prévoir entre les deux circuits.

Il y a deux possibilités de se procurer un μC programmé. Le commander à l'e-choppe sous la référence 120437-41 sur [2] ou en programmer un soi-même en circuit au moyen de Studio d'Atmel et d'un programmeur adapté comme AVRISP mkII d'Atmel branché sur K2. Le fichier hexadécimal et le fichier source (120437-11) sont téléchargeables gratuitement [2]. Vous pourrez en lire davantage sur le logiciel, la mise en boîtier et le réglage dans la prochaine édition.

(140373 - version française : Robert Grignard)

Liens

- [1] www.elektor.fr/120437
- [2] www.elektor.fr/140373
- [3] www.elektor-labs.com/120437

tension de sortie précise

Depuis des décennies, l'électronique emploie des régulateurs de tension, de toutes formes et tailles, à tension de sortie fixe ou réglable. L'un des plus anciens encore en production, le LM317, de National Semiconductor, date de 1971. Si les versions actuelles de la puce du LM317 diffèrent sans doute de l'original, ses propriétés, brochage et dimensions n'ont pas changé.

Les deux résistances sur la broche *Adjust* (**fig. 1**) permettent un réglage facile de la tension de sortie. La tension d'entrée maximale admissible atteint 40 V, le régulateur peut fournir plus de 2 A si la différence entre l'entrée et la sortie reste inférieure à 15 V.

Il est possible de régler la tension de sortie de façon très précise si l'on prend la peine de commencer par mesurer la tension de référence interne du composant. Elle vaut entre 1,2 et 1,3 V selon le fabricant. Pour mesurer avec exactitude la tension de référence du LM317 utilisé, il faudra le monter sur une platine d'expérimentation et le connecter selon le schéma de la **figure 2**. R1 pourra valoir entre 240 et 470 Ω. Appliquer à l'entrée une tension comprise entre 3 et 10 V (en tous cas plus de 3 V, la différence de potentiel minimale entrée et sortie pour que le LM317 fonctionne correctement). À l'aide d'un multimètre, mesurez la tension en sortie du régulateur : c'est la tension de référence interne. L'auteur a mesuré les valeurs suivantes sur des exemplaires de différentes sources : ST317 : 1,249 V, UA317 : 1,275 V, SSS317 : 1,231 V.

Il faut tenir compte d'un courant de 50 µA en sortie de la broche *Adjust* ; celui-ci traverse donc aussi la résistance R2 du diviseur de tension de la figure 1. Cette valeur aussi peut varier d'un fabricant à l'autre – cf. la fiche technique.

En tenant compte de ces éléments, on pourra calculer la tension de sortie requise :

$$R2_{\text{théorique}} = (U_{\text{sor}}/U_{\text{réf}} - 1) \cdot R1$$

$$R2_{\text{adjust}} = (U_{\text{sor}} - U_{\text{réf}}) / I_{\text{adjust}}$$

$$R2_{\text{tot}} = R2_{\text{théorique}} \cdot R2_{\text{adjust}} / (R2_{\text{théorique}} + R2_{\text{adjust}})$$

Autre option : utiliser un ajustable pour R2 et le tourner jusqu'à ce que la tension de sortie désirée soit atteinte ; nos calculs permettent cependant de monter immédiatement la résistance fixe requise. On pourra utiliser les mêmes calculs pour la version négative, le LM337.

Lors de ces calculs, nous avons supposé que la température de fonctionnement du circuit intégré restait relativement constante. Tant $U_{\text{réf}}$ que I_{adjust} changent lors de fortes variations de température, tandis que la tension de sortie varie également quelque peu en fonction de la charge.

(140341 – version française : Guy Raedersdorf)

D'après une idée de
Peter Krueger
(Allemagne)

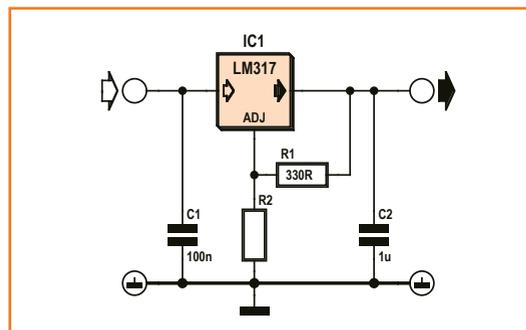


Figure 1.
Schéma standard de mise en œuvre du régulateur de tension LM317.

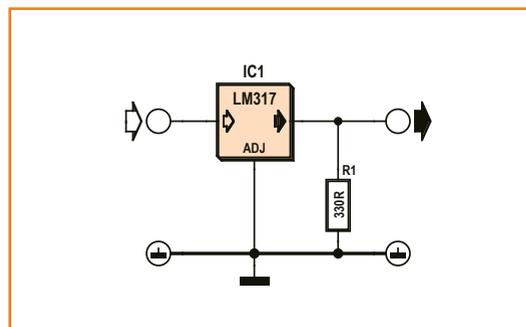


Figure 2.
En forçant la broche *Ajust* à la masse nous pouvons mesurer, en sortie, la tension de référence interne.

USBprog 5.0

programmateur *open source* + interface web

Benedikt Sauter &
Jens Nickel



Depuis 2007, le programmeur USBprog *open source* est apprécié des lecteurs d'Elektor qui s'en servent comme outil multifonctionnel pour divers contrôleurs. En voici une nouvelle version, dotée d'un serveur internet. Son interface HTML permet à ce programmeur de fonctionner avec tous les systèmes d'exploitation et toutes les tablettes. Aucune installation d'un logiciel propre ! Autres atouts à bord, une interface d'automatisation, la mémorisation des fichiers hexadécimaux et l'intégration du débogueur GDB pour ARM.

De plus en plus d'appareils sont dotés d'un serveur web. Avec les pages HTML fournies, on peut alors les configurer et les piloter à partir d'un ordinateur sur lequel on ne doit installer aucun logiciel, le navigateur habituel suffit. Le tout travaille sous n'importe quel système d'exploitation et aussi sur tablette.

Cette riche idée, Embedded Projects GmbH l'a étendue au monde des programmeurs. Flasher et lire le micrologiciel, placer et lire les fusibles de bit et plein d'autres choses, USBprog 5.0 peut le faire au départ d'une simple interface utilisateur HTML. Évidemment, l'équipe de Benedikt Sauter a voulu maintenir la nouvelle version de l'USBprog dans le concept open source, comme nous allons le voir.

Caractéristiques techniques

- Programmeur AVR (avrdupe [5])
- Débogueur et programmeur ARM-JTAG (openocd [4])
- Adaptabilité à d'autres contrôleurs en préparation
- Changement de niveau réglable sur 1,8 V, 3,3 V et 5,0 V
- Interface navigateur pour la simplicité d'emploi
- Emploi automatique par outil à ligne de commande
- Utilisable sur Atmel Studio et autres programmes
- Mémorisation des fichiers hexadécimaux utilisés fréquemment

Raccordement au PC

Un USBprog 5.0 est une simple petite carte Linux dans un élégant boîtier à relier par USB au PC qui lui fournit le courant. Il se comporte comme une interface réseau par USB ; Mac et Linux reconnaissent automatiquement l'appareil et le connectent. Un serveur DHCP tourne dans l'USBprog et lui attribue une adresse IP. Ensuite, à partir d'un PC, on peut accéder au programmeur avec un navigateur par l'adresse 10.0.0.1. Lors de la première connexion sous Windows, l'habituelle boîte de sélection du pilote s'affiche. On lui dit qu'on veut le choisir soi-même : comme appareil, adaptateur réseau, ensuite carte réseau. Une série de fabricants apparaît alors. On choisit *Microsoft Corporation* et *Remote NDIS Compatible Device*.

Pour ARM et AVR

La connexion réseau une fois établie, un navigateur (<http://10.0.001>) donne accès au programmeur ; vient alors à l'écran une fenêtre telle qu'à la **figure 1**. Elle permet de relier l'USBprog au processeur cible qui se trouve aussi alimenté du même coup. Pour le moment, seuls sont concernés les processeurs ARM et les micro-contrôleurs Atmel, mais bientôt les contrôleurs PIC s'y ajouteront, une mise à niveau en ce sens du micrologiciel est en préparation.

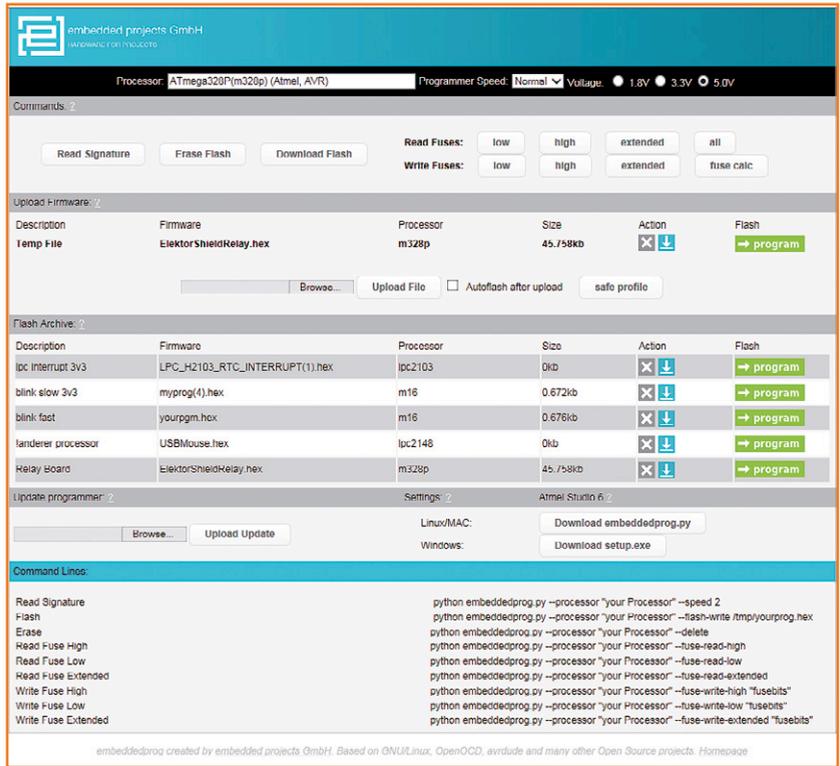
Pour le contrôleur AVR, le programmeur porte l'habituelle embase à rebords à 10 picots pour l'ISP (**figure 2**) que l'on relie par câble plat au connecteur correspondant sur la carte cible. Pour permettre d'établir aussi la connexion avec une fiche ISP à 6 contacts, l'USBprog est accompagné d'un petit circuit imprimé d'adaptation. Il faut souder soi-même l'embase à rebords, mais ce n'est pas bien difficile.

Pour flasher et déboguer les contrôleurs ARM, on a besoin d'un adaptateur sur une embase JTAG à 20 picots qu'il faut aussi souder soi-même. Elle fait encore partie de la livraison.

Le programmeur actualisable est déjà doté d'un connecteur Gnuclin/EEC à 14 contacts pour commander ultérieurement la carte d'extension Gnuclin, également par interface internet, évidemment.

Commande manuelle

Dans la fenêtre du navigateur, on commence par choisir le processeur : la boîte de sélection



se trouve tout en haut. Pour choisir un ATmega328P (Arduino Uno [1]) ou un ATmega328 (moTule 28 [2]), il suffit chaque fois de quelques lettres. On indique la tension du processeur cible à l'aide d'un bouton radio à droite.

Pour un premier essai, lisons la signature du contrôleur ou ses fusibles. Après quelques secondes s'ouvre une fenêtre noire textuelle

Figure 1. Une page HTML dans n'importe quel navigateur pour piloter le programmeur USBprog 5.0.

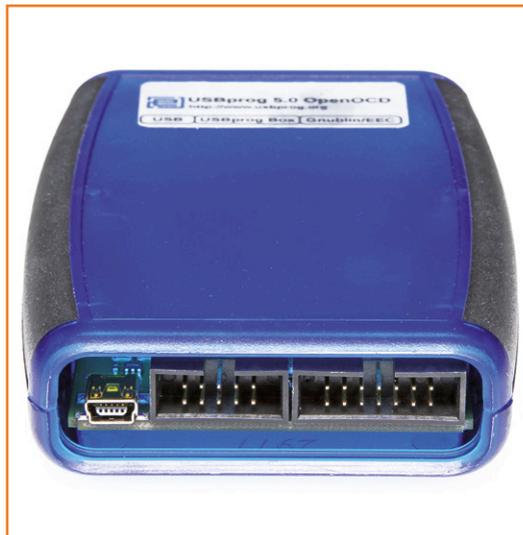


Figure 2. Les connexions du programmeur. À droite, le connecteur à 14 broches Gnuclin/EEC. Après une prochaine actualisation du micrologiciel, il pourra aussi se faire piloter par la carte d'extension Gnuclin.

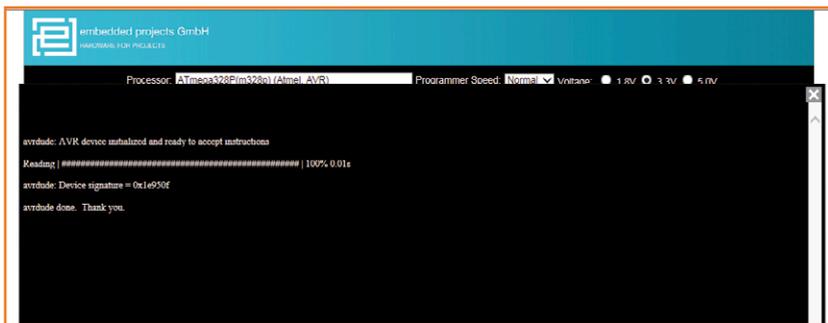


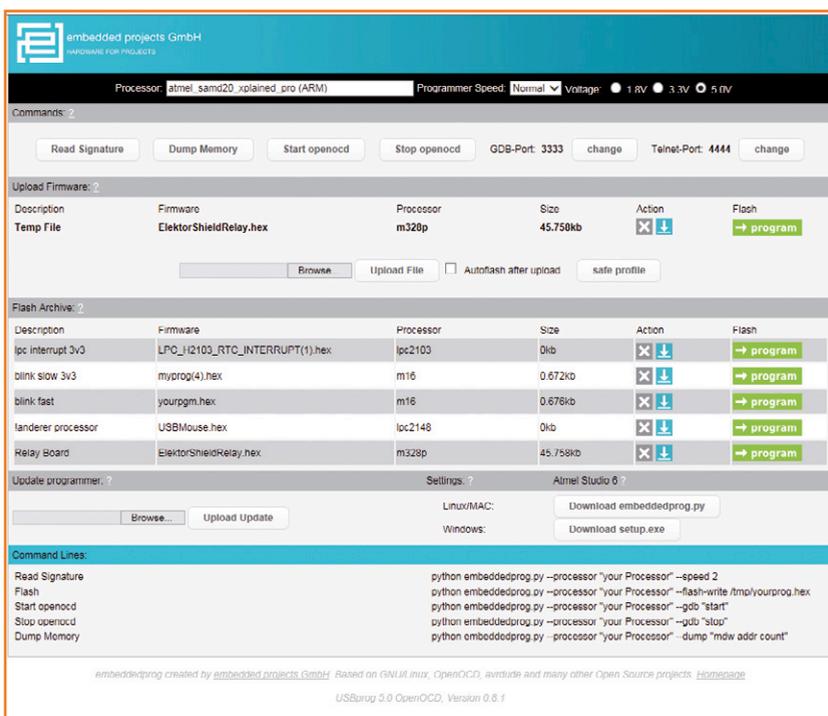
Figure 3. L'interface de programmation open source **avrdude** utilisée pour USBprog.

(**figure 3**) avec la réponse. Voulons-nous programmer un fichier hexadécimal, il y a pour cela la partie *Upload Firmware* de la fenêtre. On sélectionne un fichier avec *Browse* ; alors avec *Upload File*, on l'envoie sur le programmeur, puis finalement, le bouton vert « -> *program* » le transfère dans le processeur cible. On peut joindre au fichier hexadécimal un titre court dans la mémoire du programmeur à condition de cliquer sur le bouton *safe profile*. Les fichiers mémorisés apparaissent dans la partie *Flash Archive*. Une fonction qui permet de charger très rapidement chaque micrologiciel dans plusieurs petites séries de produits.

Figure 4. La page d'interface après le choix d'un contrôleur ARM.

Commande par script

Si le maniement manuel du programmeur est



confortable, il ne couvre pourtant pas tous les scénarios d'application. Le développement du micrologiciel se fait le plus souvent dans l'environnement du Studio d'Atmel ; on épargne ainsi du temps quand on peut flasher le contrôleur avec un programme compilé depuis IDE. Bien entendu, le nécessaire de programmation permet aussi le flashage automatique avec un fichier de traitement par lot ou avec un logiciel pour PC.

C'est pourquoi l'équipe de Benedikt Sauter a rédigé un petit outil en Python que l'ordinateur de développement appelle par ligne de commande. Python offre l'avantage qu'il existe des interpréteurs pour tous les systèmes d'exploitation actuels. Sous Linux ou Mac, on peut appeler directement *embeddedprog.py* sans autre installation. Python fait partie des programmes standard de ces systèmes d'exploitation. Il faut seulement faire tourner soi-même l'outil Python sur l'ordinateur, ce que l'on fait dans la fenêtre du navigateur en poussant sur le bouton *Download embeddedprog.py*.

Par ligne de commande, on appelle l'interpréteur Python qui exécute le programme Python. Le processeur et la tâche que le programmeur doit accomplir, on les passe comme paramètres. Par exemple pour lire la signature d'un ATmega328P, l'appel à partir du dossier où l'outil est stocké s'écrit :

```
python embeddedprog.py --processor m328p
--speed 2
```

Suivies de leurs paramètres, les instructions affichées dans la fenêtre internet sont alors exécutées.

Important : c'est lors du premier appel de l'outil Python qu'il faut donner l'adresse IP du programmeur. Elle est alors inscrite dans le fichier de configuration et sera ensuite automatiquement répétée.

```
python embeddedprog.py --eeprog-
ip 10.0.0.1 --eeprog-port 8888
--processor m328p --speed 2
```

Avec Windows, il faut commencer par installer Python [3]. On télécharge l'outil Python *embeddedprog.py* et on le place à l'endroit adéquat ; dans la ligne de commande, il faut indiquer le chemin

vers l'interpréteur Python et vers l'outil Python. Le plus simple est de le faire avec un fichier .exe à installer sur le PC Windows, le fichier *Setup* adapté est à charger avec le bouton *Download setup.exe* en bas. Lors de l'installation, on communique aussi à Windows les chemins ; en outre, on indique l'adresse IP dans le fichier de configuration. Dans la ligne de commande Windows (que l'on atteint dans le menu Démarrer avec « cmd ») on appelle alors simplement de fichier .exe :

```
embeddedprog.exe --processor m328p
--speed 2
```

Avec beaucoup de langages de programmation, on peut introduire des appels de ligne de commande (et utiliser les lignes de sortie), de manière à se lancer dans la rédaction de son propre logiciel pour le programmeur. Dans un prochain article, nous présenterons une extension du configurateur EFL.

Il est aussi possible de commander directement l'outil Python à partir de Studio 6 d'Atmel (cf. encadré).

Débogage des processeurs ARM

Le nouvel USBprog sert aussi à déboguer les processeurs ARM. On utilise alors le débogueur open source openocd [4] qui tourne sur le programmeur. Il faut pour cela transférer au préalable le micrologiciel dans le processeur. Le démarrage du débogueur se fait par le navigateur (*Start openocd* comme à la **figure 4**) ou par une ligne de commande.

On peut piloter le débogueur par une interface Telnet simple ou par une interface GDB. Les plus grands environnements de développement proposent souvent une interface appropriée. Sur internet, on trouve de nombreuses explications quand on cherche sur ARM Eclipse GDB. Comme *Remote Address*, on doit fournir *localhost* qui indique l'adresse IP du débogueur : 10.0.0.1.

Radioscopie

La **figure 5** montre les entrailles du programmeur : le processeur maître est un LPC3131, la mémoire de travail, du type A43L4616 (8 Mo) est juste à côté de lui. Le changement de niveau est confié au GTL2010PW qui permet aisément

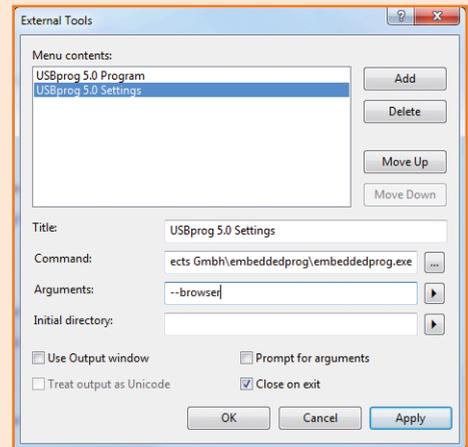
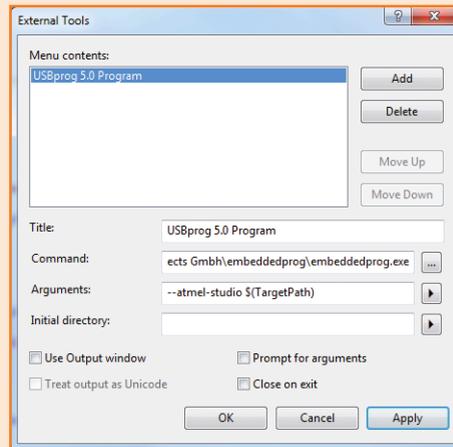
Au travail avec Atmel Studio

Que serait un programmeur pour contrôleur AVR s'il ne pouvait pas profiter de l'environnement de développement gratuit Studio d'Atmel ? Pareil pour l'USBprog grâce à l'outil à ligne de commande (cf. texte). Ses principaux atouts : flasher des fichiers hexadécimaux dans le contrôleur AVR et appeler la page du Studio sur le navigateur, ne fût-ce que pour régler les paramètres d'initialisation.

Pour chacune de ces actions, on vise un point du menu par *Tools -> External Tools*. D'un clic sur le bouton *Add*, une entrée se présente dans chacune des zones de texte *Title*, *Command* et *Arguments*, puis on clique sur *Apply*. Commençons par *USBprog 5.0 Program* pour télécharger le micrologiciel (cf. copie d'écran). En dessous de *Command*, il y a toujours le chemin complet vers l'outil *embeddedprog.exe* (cf. texte).

Si vous avez déposé une entrée pour chacune des deux actions comme dans les écrans ci-joints, elles se retrouvent dans le menu principal de Studio sous *Tools*.

La première fois que vous sélectionnez *USBprog 5.0 Program*, le navigateur se rend à la page ad hoc. Il faut commencer par le choix du processeur, puis de sa tension d'alimentation. Dès ce moment, vous pouvez, par ce point du menu du Studio d'Atmel, flasher directement les fichiers hexadécimaux dans le contrôleur.



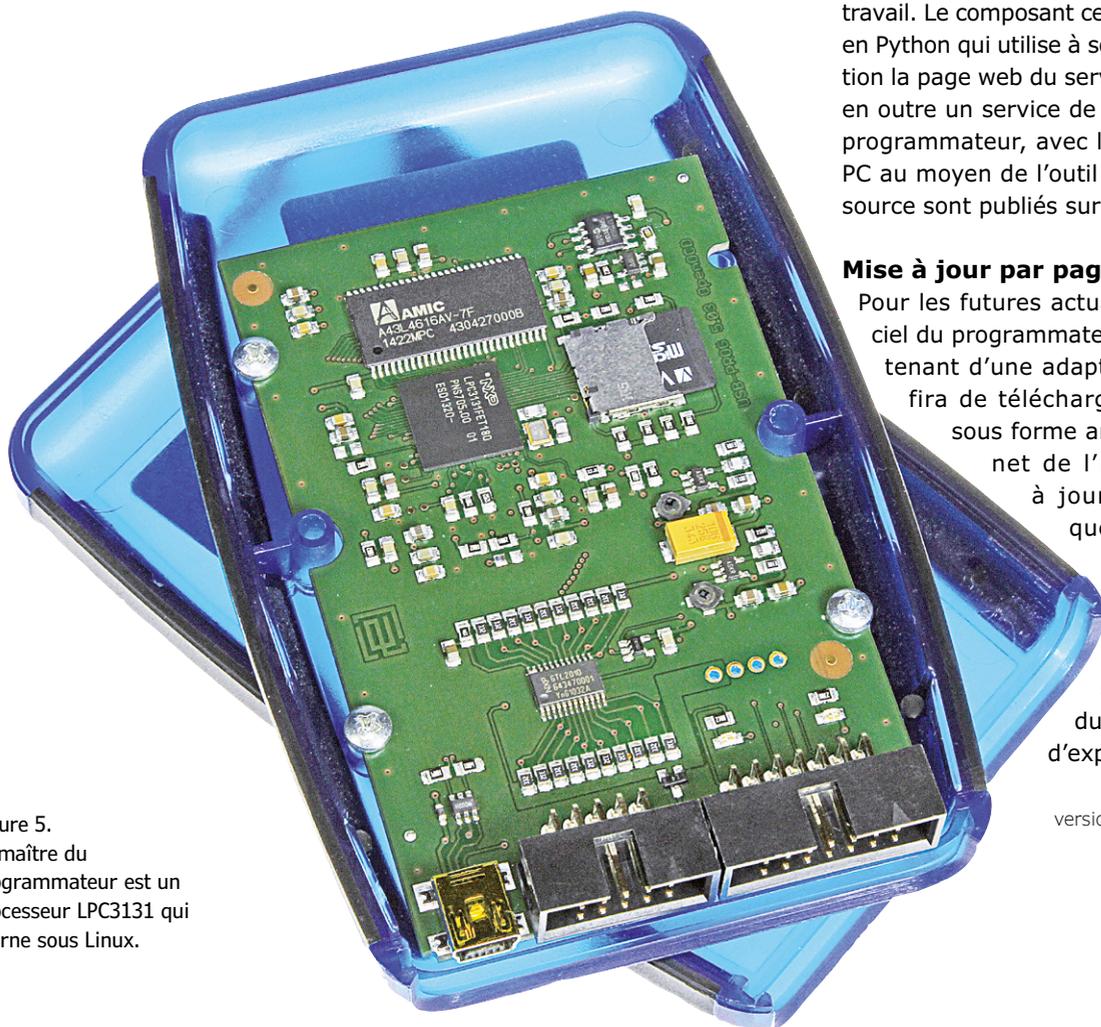


Figure 5.
Le maître du programmeur est un processeur LPC3131 qui tourne sous Linux.

la conversion bidirectionnelle des signaux. Pour pouvoir choisir par logiciel la tension à appliquer au matériel cible (1,8 V, 3,3 V et 5,0 V), c'est un LDO AP2127K-ADJ qui est utilisé, commandé par un potentiomètre numérique MCP4131. Le courant de sortie peut atteindre jusqu'à 300 mA.

Sur le processeur géré par un Linux embarqué, un logiciel standard comme avrdude, pour la programmation des processeurs Atmel [5], fait son

travail. Le composant central est un serveur écrit en Python qui utilise à son tour pour la présentation la page web du serveur Lighttpd. Il propose en outre un service de soutien en ligne pour le programmeur, avec la communication sur le PC au moyen de l'outil Python. Tous les textes source sont publiés sur GitHub [6].

Mise à jour par page web

Pour les futures actualisations du micrologiciel du programmeur, il est doté dès maintenant d'une adaptabilité intégrée. Il suffira de télécharger la nouvelle version sous forme archivée sur le site internet de l'USBprog [7]. La mise à jour sera transmise telle quelle par l'intermédiaire de la page du navigateur pour être ensuite extraite sur l'adaptateur du programmeur et copiée en bonne et due place dans le système d'exploitation.

(140285 -
version française : Robert Grignard)

Liens

- [1] www.elektor.fr/arduino-uno
- [2] www.elektor.fr/t-board-28-130581-93
- [3] <https://www.python.org>
- [4] www.openocd.org
- [5] www.nongnu.org/avrdude
- [6] <https://github.com/embeddedprojects/usbprog5>
- [7] <http://usbprog5.embedded-projects.net>



USBprog 5.0

À l'e-choppe, les USBprog 5.0 sont disponibles en deux versions. Il y a le circuit imprimé simplement assemblé ou le même en boîtier robuste avec le câble plat et les adaptateurs qu'il faut souder soi-même.

Toutes les infos sur elektor.fr/usbprog5.

transistors tétrodes

drôle de composant n°12

Nous avons tous utilisé des transistors bipolaires NPN et PNP à trois broches. J'ai découvert qu'il existait des transistors tétrodes à quatre broches, de plusieurs sortes. Ils comportent deux éléments de commande, c'est-à-dire deux bases pour les modèles bipolaires et deux grilles pour les MOSFET, comme l'indiquent leurs symboles (**fig. 1**). Qu'est-ce qui les rend si spéciaux ?

En électronique une tétrode est un composant avec quatre électrodes actives même si le terme est habituellement réservé aux tubes à vide du même nom qui possèdent deux grilles au lieu d'une. La grille supplémentaire, appelée grille-écran, diminue la capacité entre les plaques par rapport à une triode conventionnelle, ce qui augmente la gamme de fréquence du tube. Les transistors tétrodes ne peuvent pas remplacer directement les tubes tétrodes, mais leur atout est également une capacité parasite réduite et donc une plus vaste gamme de fréquence. Un transistor tétrode bipolaire est fabriqué en ajoutant une deuxième connexion de base, de l'autre côté du silicium (**fig. 2**, en haut). La deuxième partie de la figure montre la construction d'un MOSFET à deux grilles.

Pourquoi la capacité est-elle si importante ? La conception d'amplificateurs à transistors discrets est toujours une bataille où l'on cherche à maximiser la bande passante, le gain et l'isolement entre l'entrée et la sortie. L'une des raisons qui rend cela difficile est l'effet Miller et la façon dont il peut affecter un amplificateur. L'effet Miller se produit lorsque le gain d'un amplificateur amplifie la capacité entre l'entrée et la sortie, ce qui augmente la capacité d'entrée équivalente. Dans les amplificateurs à transistors, c'est habituellement la capacité parasite inhérente au transistor qu'il faut vaincre pour maximiser la bande passante de l'amplificateur.

Une manière de minimiser l'effet Miller est d'utiliser deux transistors pour réaliser un amplificateur cascode (**fig. 3**). L'un des transistors est utilisé comme amplificateur de transconductance, l'autre sert de tampon de courant. C'est en isolant la capacité d'entrée du transistor du bas de l'étage tampon que l'amplificateur à deux étages permet de minimiser l'effet Miller. Les transis-

tors tétrodes sont donc un excellent choix pour les amplificateurs cascode à petits signaux qui nécessitent également une large bande passante en fréquence.

Le transistor tétrode le plus courant est le MOSFET à double grille qui possède d'autres applications intéressantes dans les circuits RF. Des modèles fameux, tout droit venus des années

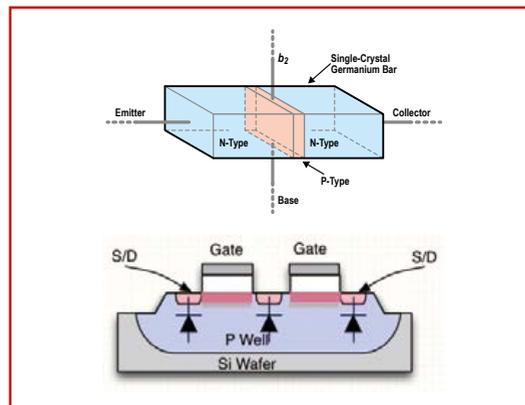


Figure 2. Construction des transistors tétrodes.

80 et bien placés au palmarès des modèles utilisés en conception RF sont : en Europe les séries **BF96x/BF98x**, et aux États-Unis le **40673** et le **3N211**. Par exemple, un simple mélangeur à FET reliera la sortie du mélangeur local et la sortie RF à la grille d'un FET à l'aide de composants supplémentaires pour isoler le mélangeur de la RF. Un MOSFET à double grille permet d'éliminer ces composants d'isolement : ses grilles sont déjà isolées l'une de l'autre. Les MOSFET à double grille fonctionnent aussi bien dans les circuits de commande de gain automatique (AGC) : il est possible de polariser une grille et d'amener le signal sur l'autre. La tension de polarisation est alors l'entrée qui permet de commander le gain global du transistor.

Drôles de composants, ces transistors tétrodes, et intéressants ! Pourtant, je n'en avais jamais entendu parler avant d'écrire cet article.

(140418 - version française : Kévin Petit)

Neil Gruending
(Canada)

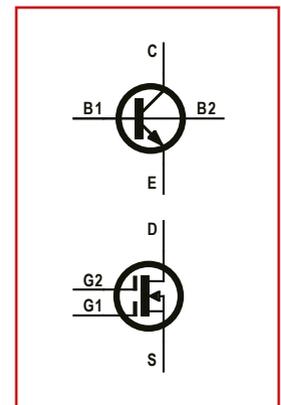


Figure 1. Symboles de schéma de transistors tétrodes.

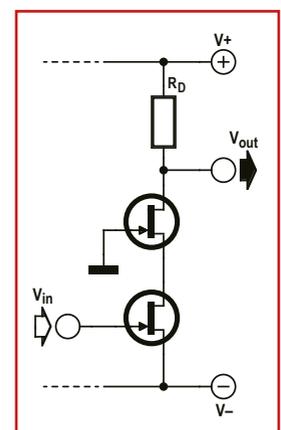


Figure 3.
Un amplificateur cascode.

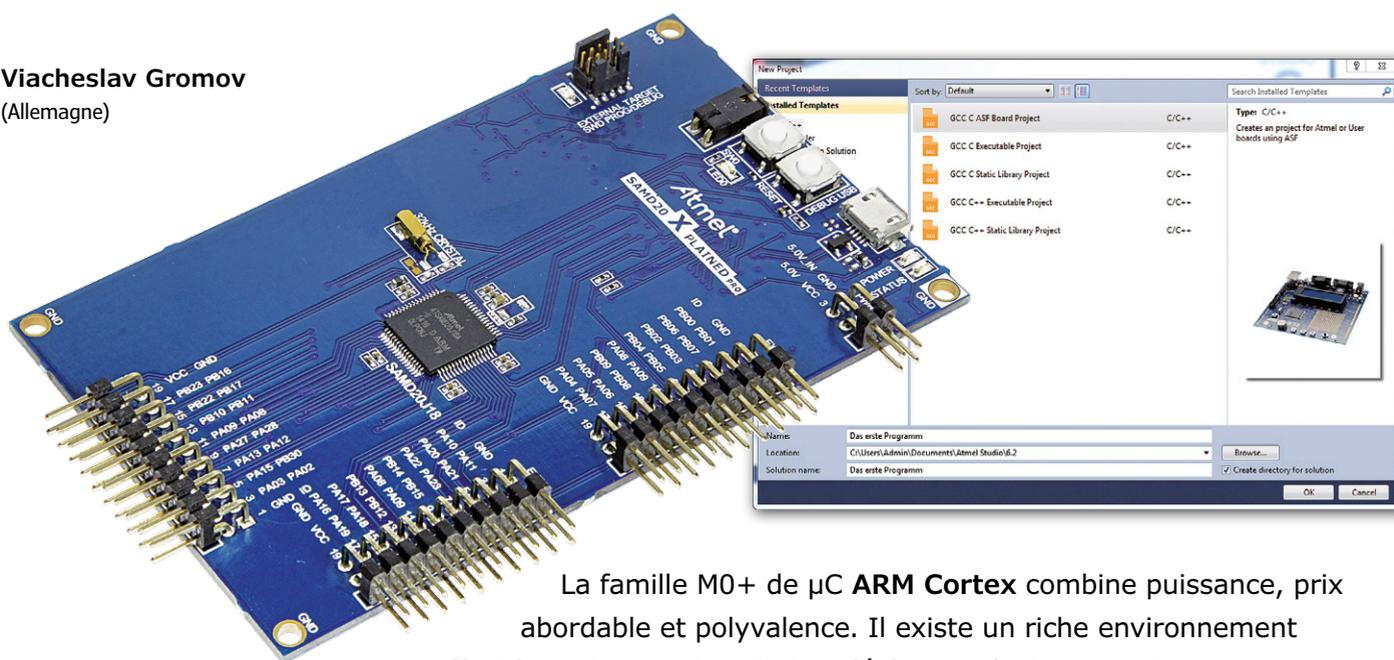
microcontrôleur ARM pour néophytes

1^{ère} étape

pour passer de 8 bits à 32 bits

une carte, un logiciel et un premier programme

Viacheslav Gromov
(Allemagne)



La famille M0+ de μ C **ARM Cortex** combine puissance, prix abordable et polyvalence. Il existe un riche environnement d'outils pratiques et gratuits. Idéal pour s'initier, non ? Voici un cours pour apprendre à utiliser ces microcontrôleurs à 32 bits. Tout ce qu'il faut pour se lancer, c'est de la curiosité et une certaine familiarité avec les μ C à 8 bits !

Ce nouveau cours de programmation Elektor vous ouvre le monde des μ C Cortex-M0+ d'ARM en mettant l'accent sur la pratique. Il existe de nombreux environnements de développement gratuits et une pléthore de cartes abordables. Le pilier de notre

cours est la carte *SAMD20 Xplained Pro*, avec un μ C à faible consommation, le SAM D20. Grâce au soutien d'Atmel, son fabricant, nous sommes en mesure de proposer aux lecteurs d'Elektor intéressés **1 000 cartes à prix réduit**. Après avoir examiné la carte et le μ C, nous passe-

1 000 cartes au meilleur prix dans l'e-shoppe Elektor !

Grâce au soutien d'Atmel, Elektor propose **1 000 cartes SAMD20-Xplained-Pro** au prix plancher de **19,95 €** (TVA incluse, frais de port en sus).

Premier arrivé, premier servi ! www.elektor.fr/samd20-board

rons à l'installation de l'environnement de développement Atmel Studio 6.2 et attaquerons un premier projet. Nous lorgnerons, à l'occasion, vers la famille AVR à 8 bits, réputée s'il en est, en raison de nombreuses similitudes. Dans les articles suivants, nous présenterons, sous forme de petits projets, les périphériques majeurs et leurs fonctions.

La carte

À première vue, rien de spectaculaire sur la carte (**fig. 1**). Bon nombre des 64 broches du μC sont amenées vers l'extérieur sur des embases mâles. Des tableaux donnent le brochage des connecteurs et les dénominations d'autres points de connexion.

L'alimentation de la carte peut se faire soit par l'embase *USB* soit par application de +5 V au connecteur *PWR*. En cas d'alimentation via le port USB, on dispose sur *PWR* respectivement de 5 V et de 3,3 V pour l'alimentation des circuits connexes. Si la carte est alimentée via l'embase *PWR*, le débogueur embarqué EDBG (Cf. encadré) se désactive automatiquement pour économiser de l'énergie. L'alimentation ou le port USB d'alimentation doivent pouvoir fournir un courant de 500 mA minimum.

Les embases *EXT1* à *EXT3* mettent chacune à disposition une tension de 3,3 V pour l'alimentation de cartes d'extension. Sur chacune de ces embases, la broche 1 baptisée ID est réservée pour la connexion d'une puce ID présente sur la carte d'extension. Cela permet à EDBG de reconnaître de quelle carte d'extension il s'agit. Ladite information sera transmise à l'environnement de développement sur le PC.

Autres éléments sur la carte, outre le quartz 32 kHz (une des sources d'horloge du μC principal) : l'embase *DEBUG USB* pour un débogueur externe, un bouton *RESET*, le bouton *SW0* et une LED jaune, *LED0*. *SW0* et *LED0* sont reliés respectivement à PA15 et PA14, et sont à disposition de l'utilisateur. Le cavalier proche de *SW0* relie la tension de sortie du régulateur de tension présent sur la carte au μC . Si, au lieu du cavalier, on branche à cet endroit un instrument de mesure, on pourra mesurer la consommation électrique du μC .

Les LED d'alimentation et d'état (non représentées) voisines du connecteur USB sont connectées

à l'EDBG. La LED d'alimentation s'allume lorsque la carte est alimentée. La LED d'état clignote lorsque l'EDBG a accès au μC principal SAM D20 en cours du débogage ou lors d'autres opérations. En cas de réactualisation du progiciel (*firmware*) du débogueur, les deux LED clignotent. Pour le manuel d'utilisation de cette carte, voir [1].

Le microcontrôleur

Le SAM D20J18, membre intéressant de la famille ARM-Cortex-M0+, est *armé* de 64 broches, il possède 256 Ko de mémoire Flash, 32 Ko de SRAM et peut être cadencé à 48 MHz maximum. Économe en énergie, rapide, il comporte nombre de périphériques : sa polyvalence est remar-

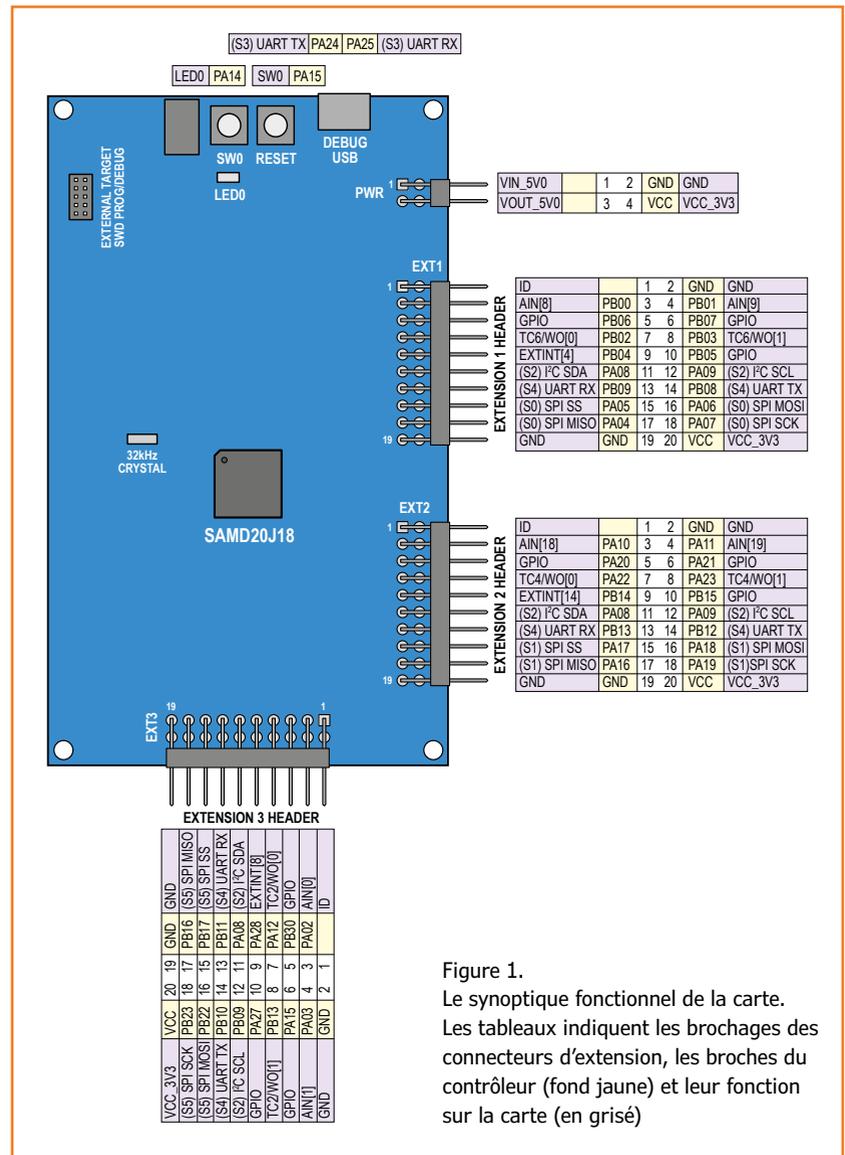


Figure 1. Le synoptique fonctionnel de la carte. Les tableaux indiquent les brochages des connecteurs d'extension, les broches du contrôleur (fond jaune) et leur fonction sur la carte (en grisé)

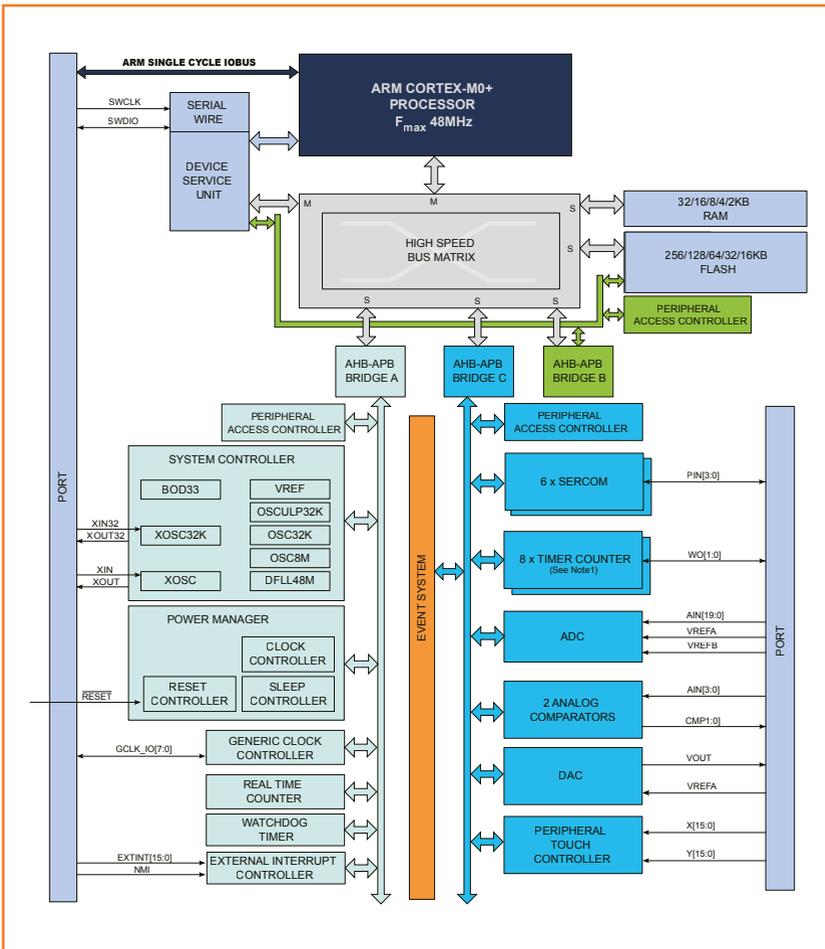


Figure 2. Synoptique du contrôleur SAM-D20 (Source : Atmel).

quable. Il ne consomme que 70 $\mu\text{A}/\text{MHz}$ sous 1,62 à 3,63 V. Au nombre de ses spécificités : l'*ES* (*Event-System* = système d'événement) et le *PTC* (*Peripheral Touch Controller* = contrôleur tactile périphérique). Nous reviendrons en détail sur le PTC quand nous le mettrons à contribution. En fonction de la configuration, l'*Event-System* peut par exemple, à l'image des μC ATxmega, réveiller l'unité centrale quand certains modules périphériques tel que l'ADC déclenchent un événement. Le μC connaît deux modes de sommeil : *attente* (*Idle*), seule la CPU est désactivée ; *veille* (*Standby*), l'ensemble du système d'horloge et tous les éléments (hormis ceux configurés différemment dans le programme) roupillent.

Sur le synoptique fonctionnel (**fig. 2**) de cette famille, tout en haut à gauche, on découvre le bus *ARM Single Cycle I/O*, qui permet au processeur un accès ultrarapide aux GPIO (*General Purpose I/O*). On retrouve en dessous l'interface de débogage (*Debug*) série avec accès direct au processeur. Sous le bloc *High Speed Matrix*, auquel, sur la droite, sont connectées les mémoires avec leurs contrôleurs respectifs fonctionnant en esclaves, on peut identifier (contrairement à ce qui est le cas avec un μC à 8 bits conventionnel) divers bus de données et *Peripheral Access Controller*, qui peuvent, le cas échéant, empêcher l'écriture dans des registres de périphériques. Le Bus *APB-C* est le « cortex » des périphériques les plus importants, dont les plus intéressants sont les 6 blocs *SERCOM*. Ils sont utilisés pour la communication série et peuvent être configurés en interfaces série (USART, I²C, SPI). Les éléments périphériques restants (exception faite du PTC) existent sur la plupart des μC à 8 bits, mais avec moins de puissance et en nombre moindre. Les 8 *TIMER COUNTER* peuvent être utilisés en compteurs à 2x8 bits, 1x16 bits ou (par paire) en compteur à 32 bits. L'autre côté du synoptique est moins spectaculaire, concernant en fait l'alimentation et l'horloge. Nous reviendrons, étape par étape

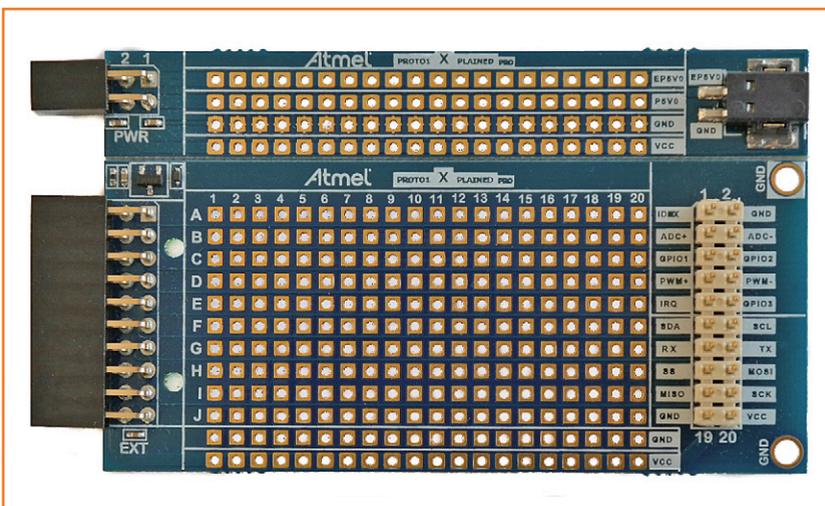


Figure 3. La carte d'extension la plus simple que propose Atmel est une carte de prototypage à îlots de soudure.

et par la pratique, sur les possibilités, le paramétrage et l'application de tous ces périphériques. La fiche de caractéristiques compte 700 pages [3].

Les cartes d'extension

Pour la *Xplained Pro-Board*, Atmel a développé différentes cartes d'extension afin de faciliter la prise en main, d'accélérer le prototypage, d'arriver rapidement au but et d'apprendre à mieux connaître le μC . Aspect pratique : les cartes d'extension viennent directement s'enficher sur les embases mâles de la carte mère. Chaque carte d'extension comporte une puce ATSHA204 (*CryptoAuthentication*) à 3 pattes qui renseignent la puce EDBG de la carte-mère *plained-Pro-Board* sur l'identité de l'extension connectée (p. ex. sa plage de tension et son courant de service maximal). La puce EDBG relaie ces informations à l'environnement Atmel Studio. Studio ouvre une fenêtre proposant des liens vers des fiches de caractéristiques, des bibliothèques ou des programmes didactiques.

Si vous avez l'intention de connecter une carte de votre propre cru à une carte *Xplained-Pro-Board* nous vous recommandons d'utiliser la carte **PROTO1 Xplained Pro** [4] (fig. 3). Elle comporte 200 îlots de soudure et vient s'enficher sur les connecteurs *EXT1* et *PWR*. Sur le côté droit, on trouve une connexion pour les nommés *Xplained-Top-Module*, caractérisée par un routage différent. La partie supérieure de la platine, l'alimentation, peut être facilement séparée du reste.

La carte d'extension (fig. 4), la bien nommée **IO 1 Xplained Pro** [5] permet de se familiariser avec les périphériques les plus importants du μC . Elle comporte une LED, un photocapteur, un passe-bas pour s'essayer à la MLI (PWM) et au CAN (ADC), un capteur de température à 12 bits avec 8 Ko d'EEPROM et connexion au bus I²C ainsi qu'un connecteur pour carte microSD, ce dernier étant relié à la SPI. La carte microSD accompagne l'envoi. On y découvre en outre un certain nombre de broches relayées vers l'extérieur.

Il vous faudra, si vous souhaitez afficher des données sur un écran, la carte d'extension **OLED1**

Figure 6.

Dans une partie de cours à venir, nous utiliserons les deux cartes QTouch.

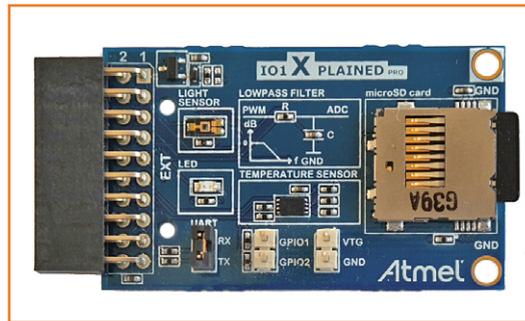


Figure 4.
Carte d'extension universelle à utiliser à des fins d'essais et de formation.

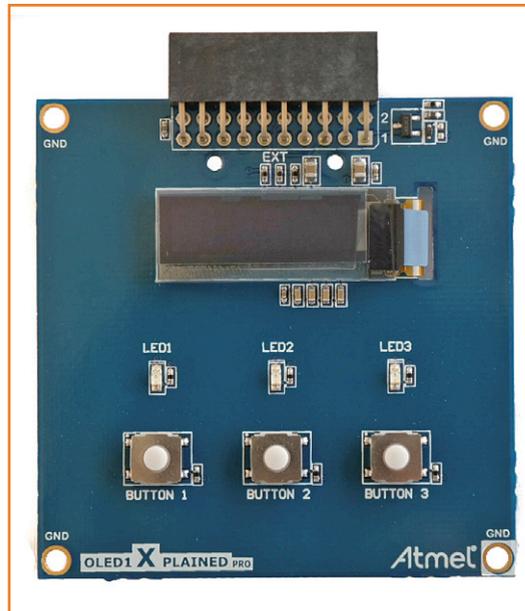
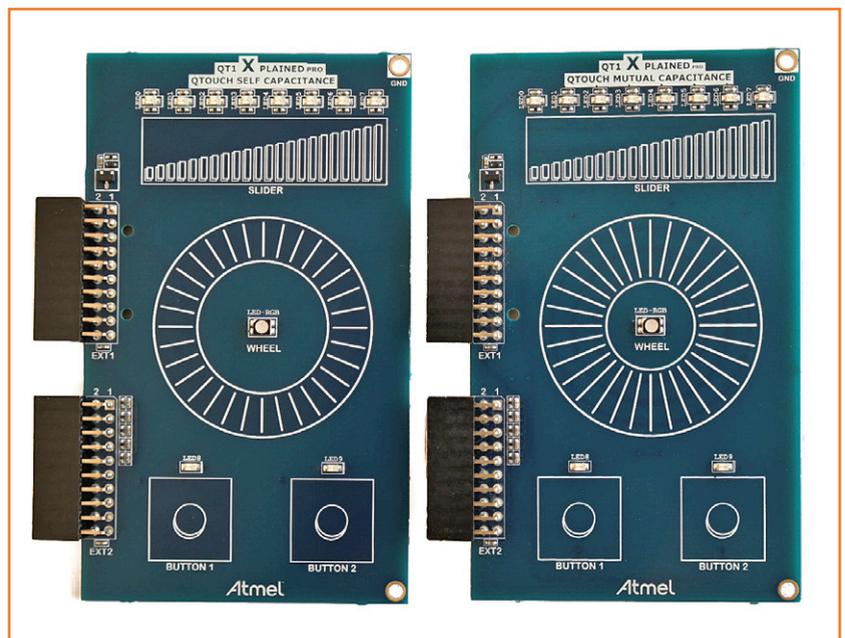


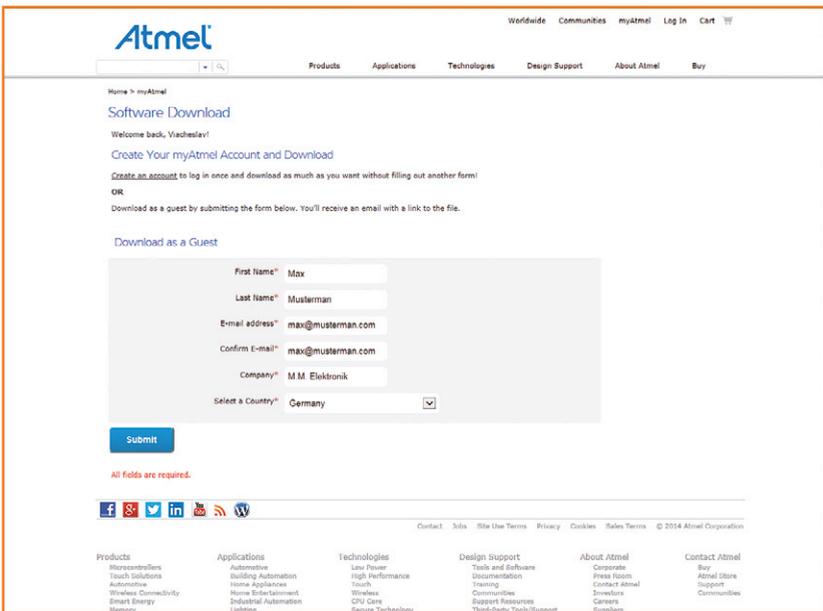
Figure 5.
Carte d'extension OLED pour projets élaborés.



Software	Description
	<p>Atmel Studio 6.2 sp1 (build 1502) Installer – with .NET (721 MB, updated October 2014)</p> <p>This installer contains Atmel Studio 6.2 service pack 1 with Atmel Software Framework 3.19 and Atmel Toolchain. This installer also contains MS Visual Studio Shell and .NET 4.0. Select this installer if you need to install Atmel Studio on a computer not connected to the internet.</p>
	<p>Atmel Studio 6.2 sp1 (build 1502) Installer (506 MB, updated October 2014)</p> <p>This installer contains Atmel Studio 6.2 service pack 1 with Atmel Software Framework 3.19 and Atmel Toolchain. Install this if you have previously installed Atmel Studio or have access to internet when installing.</p>

Figure 7.

Si vous voulez installer Atmel Studio sur un ordinateur sans accès permanent à Internet, cliquez sur l'icône tout en haut (captures d'écran : Atmel).



Xplained Pro [6], avec un écran OLED SPI (128 x 32 pixels) (fig. 5). Cette même carte met à disposition trois autres LED et trois boutons. Elle se connecte sur l'embase EXT3.

Comme d'autres µC d'ATMEL, le SAM D20 possède un PTC ; Atmel fournit le kit (fig. 6) **QT1 Xplained Pro** [7] réunissant 2 cartes d'extension. Identiques à première vue, ces cartes font cependant appel à des technologies tactiles différentes : *QTouch Self Capacitance* et *QTouch Mutual Capacitance*. Nous découvrirons les avantages et les inconvénients de ces technologies tactiles dans l'un des derniers articles de cette série. Sur chacune de ces cartes, on trouve, outre les éléments tactiles molette (*Wheel*) et glissière (*Slider*) et une paire de boutons, 10 LED jaunes et 1 LED RGB.

Figure 8.

Soit vous cliquez sur *Create an account* soit vous saisissez vos données.

Débogage et EDBG

Vous êtes peut-être familier des µC à 8 bits, mais pas nécessairement de la notion de débogage, car cette technique n'est que rarement utilisée avec les petits µC. Le débogage permet de déceler des erreurs dans le logiciel (*bug*) et de les corriger.

Pour cela, il faut deux composants : un outil dans l'environnement de développement et un débogueur, qui sert de passerelle entre l'ordinateur et le système (µC). Durant l'exécution du programme, le débogueur permet de visualiser et de manipuler les variables, la mémoire de travail, voire les registres du µC. Il est possible de placer des points d'arrêt pour suspendre l'exécution du programme et interroger l'état du matériel au point d'arrêt. On débusque ainsi les erreurs, rapidement et à moindre coût, même si le µC est déjà monté dans le prototype (en principe) fonctionnel. Pour cela, le débogueur doit rester connecté au µC [9].

L'EDBG (**E**Embedded **D**e**Bu**Gger) n'équipe pas seulement

toutes les cartes *Xplained-Pro-ARM* d'Atmel, mais est aussi souvent utilisé avec les cartes AVR. Il s'agit d'un débogueur intégré aux cartes ; Atmel l'a conçu spécialement pour les kits de développement. Outre la possibilité de programmer ou de déboguer un µC, il offre une fonction spéciale dont l'utilité apparaîtra plus tard : la *Data Gateway Interface* établit une passerelle entre plusieurs interfaces différentes ou les GPIO et l'ordinateur. On peut donc, pendant que le µC travaille, voir les états de GPIO sélectionnés et également recevoir des données via différentes interfaces, ce qui facilite énormément l'étude. Sur la carte *SAMD20-Xplained-Pro* cette interface est reliée aux broches SPI de SERCOM5, aux broches I²C de SERCOM2 et aux broches GPIO PA27, PA28, PA20 et PA21.

La puce de débogage pilote les LED d'état et lit les ID des cartes d'extension. L'EDBG peut aussi émuler un port COM via USB, sachant qu'il est relié aux broches UART (SERCOM3) du SAM D20 [2].

L'environnement de développement

Pour programmer et déboguer un μC Atmel, il suffit d'un programmeur/débogueur et d'*Atmel Studio*. Cet IDE est gratuit, provient directement des laboratoires du fabricant et propose de nombreuses fonctions intéressantes pour les produits Atmel. C'est donc l'IDE idéal pour le néophyte. Vous connaissez peut-être déjà Atmel Studio.

Au moment d'écrire ces lignes (2014-12), la version à jour est la 6.2 [8]. Cliquer sur l'icône de CD à côté de l'entrée *Atmel Studio 6.2 sp1 (build 1502) Installer* (fig. 7). Vous verrez apparaître une fenêtre (fig. 8), par le biais de laquelle vous devez soit vous inscrire soit vous connecter en tant qu'invité (*Guest*), pour pouvoir télécharger *Atmel Studio* du site. Il est recommandé de vous inscrire sur le site, car vous aurez l'occasion de vous y connecter pendant la suite de ce cours. Si vous avez déjà un compte chez Atmel, il suffit de vous connecter en haut à droite (*Log In*). Un lien sur vous permettra de télécharger le programme. Si votre navigateur vous laisse le choix entre télécharger le programme ou l'installer directement, optez pour *Install*. Dans l'avertissement de sécurité sous Windows, cochez de préférence la case « *Always trust software from Atmel Norway* » et cliquez sur *Install*. Si Microsoft .NET Framework et/ou Visual Studio n'est/ne sont pas encore installé(s) sur votre ordinateur, *Atmel Studio* le signale immédiatement et vous demandera de l'installer. Suivez les instructions. Lorsque vous installez les deux programmes, il faudra accepter les conditions de licence et toujours opter pour la version complète. Si vous n'avez pas d'objection, acceptez le chemin d'installation proposé. Un peu plus tard, l'*InstallShield Wizard* vous demandera d'installer les pilotes USB (fig. 9).

Une fois confirmée la licence (fig. 10), l'installation des pilotes démarre (cela peut durer quelques minutes). Un message final confirmera l'installation des pilotes.

L'installation d'Atmel Studio proprement dit démarre alors. Dans la première fenêtre (fig. 11), cliquez sur *Next>*, dans la seconde acceptez la licence et quittez cette fois encore la fenêtre avec *Next>* (fig. 12). Lors de l'étape suivante (fig. 13), choisissez le chemin d'installation de Studio. Si le chemin du fichier par défaut ne vous convient pas, appuyez sur *Browse...*, recherchez le chemin voulu puis cliquez sur *Next>*.

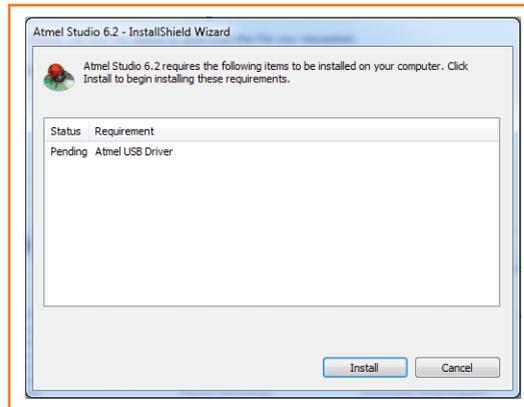


Figure 9. Studio nécessite l'installation des pilotes USB.



Figure 10. Lisez attentivement les termes de la licence avant de vous engager.



Figure 11. L'installation principale est sur le point de démarrer.

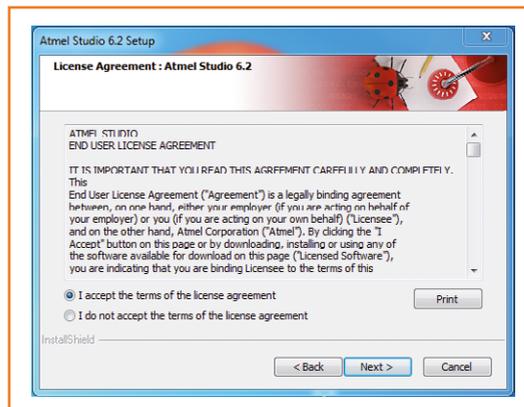


Figure 12. Si vous n'acceptez pas la licence, le processus en restera là.

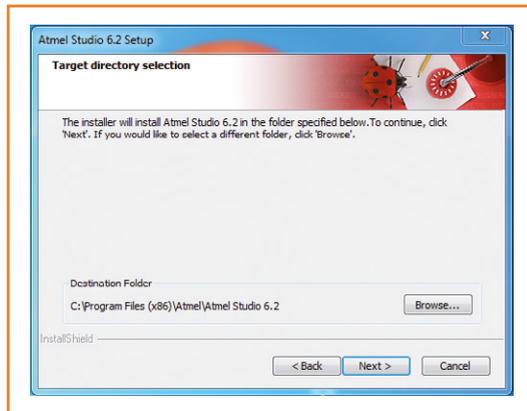


Figure 13.
Le chemin par défaut du fichier est généralement choisi à bon escient.



Figure 14.
Installation réussie !



Figure 15.
L'icône d'Atmel Studio 6.2.

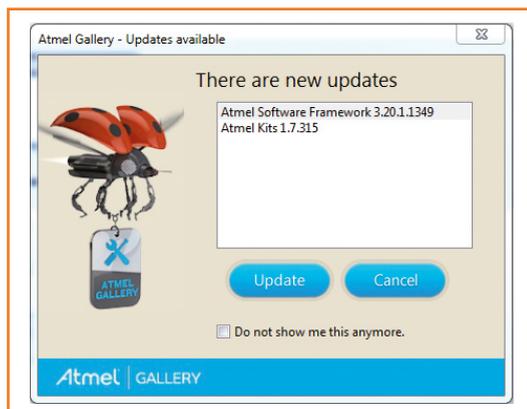


Figure 16.
Il faudra impérativement effectuer les mises à jour.

La dernière étape avant le processus d'installation proprement dit est une récapitulation de tout ce qu'il y a à installer. Cliquez sur *Next>* et attendez *un certain temps* (Cf. Fernand Raynaud) la fin de l'installation (**fig. 14**). Avant d'appuyer sur *Finish*, si sur votre ordinateur vous n'avez pas installé d'autre environnement qui utiliserait aussi les types de fichiers répertoriés, vous pouvez cocher l'option correspondante. L'*InstallShield-Wizard* disparaît. Sur le bureau apparaît une icône comme celle de la **figure 15**. Redémarrez l'ordinateur.

Mon premier programme

La carte de développement est reliée au PC à l'aide d'un câble à connecteur USB A mâle – connecteur micro-USB B. Le Gestionnaire de périphériques se manifeste et installe les pilotes que vous avez téléchargés avec Studio. Si votre Gestionnaire de périphériques ne le trouve pas, saisissez à la main le chemin à suivre vers le pilote. Il est temps de lancer *Atmel Studio* en double cliquant sur l'icône correspondante du Bureau. Vous verrez sans doute s'afficher un message (**fig. 16**) de mise à jour de différents outils.

Au nombre de ces derniers, *Atmel Software Framework* (ASF), que vous ne manquerez pas d'utiliser souvent dans les prochains temps. Cliquez donc sur *Update* de façon à être redirigé vers l'*Extension Manager* (**fig. 17**), qui exécutera successivement les mises à jour nécessaires. Au départ, vous serez invité à vous connecter afin de pouvoir installer les mises à jour. Or, les procédures d'installation diffèrent de l'une à l'autre, nous n'entrerons donc pas dans les détails ici. Selon le cas, vous verrez des messages de sécurité, de licences, de liaison des mises à jour avec *Studio*... Il suffit de suivre les instructions du programme d'installation. Une fois les mises à jour terminées, fermez l'*Extension Manager* et redémarrez *Studio*.

Nous pouvons enfin nous lancer dans notre premier projet ! Commençons par découvrir l'IDE. Après un nouveau démarrage d'*Atmel Studio*, sélectionnez la page *Start* et cliquez sur *New Project...* pour créer un nouveau projet (**fig. 18**). Vous pouvez également opter pour *Project...* sous *File/New*.

Vous verrez alors apparaître dans Studio la

À propos de l'auteur

Du haut de ses 15 ans, **Viacheslav Gromov** est le benjamin des auteurs d'Elektor ; pourtant, son enthousiasme pour l'électronique analogique et numérique ne date pas d'hier : son atelier est bien équipé, Elektor a déjà publié plusieurs de ses articles, il est aussi l'auteur de livres qui traitent, entre autres, des μ C ARM Cortex.

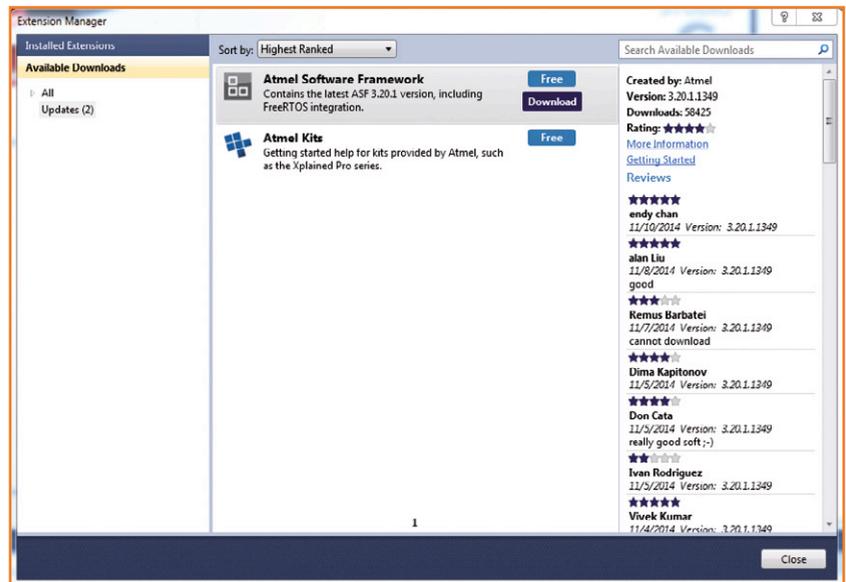
Pour leur vigoureux soutien, l'auteur remercie sa famille ainsi qu'Andreas Riedenauer de la société Ineltek Mitte GmbH.



Figure 17.

Dans *Extension-Manager* vous trouverez de nombreux outils logiciels supplémentaires à installer si besoin était.

fenêtre *New Project* qui demande de saisir le type, le nom et le chemin du programme à créer. Sélectionnez (**fig. 19**) *GCC C ASF Board Project*, qui, comme l'indique son nom, fait appel au langage de programmation C et repose sur *Atmel Software Framework*. Le mois prochain, nous nous intéresserons à ASF de plus près. Dans la fenêtre qui suit (**fig. 20**), on sélectionne la carte concernée en s'aidant de la fonction de recherche. Il faudra ensuite un certain temps jusqu'à ce que le projet soit créé et que vous puissiez ouvrir le fichier `main.c` dans le dossier `src`. Vous y trouverez le code source de la miniapplication créée (**listage 1**).



On commence par inclure le fichier d'en-tête ASF, puis, dans la routine principale, on initialise le système. Pour finir on procède, dans la boucle sans fin, à une interrogation *if*, d'action ou non sur le bouton SW0 ; dans le premier cas, la LED0 s'allume. Notre projet est donc parfaitement fonctionnel dans ce fichier d'exemple et ses fonctions faciles à saisir.

Passons brièvement aux autres fichiers. Le fichier `asf.h` ne constitue qu'une déviation vers d'autres fichiers d'en-tête. Très utile, car il faudra sinon inclure individuellement dans le fichier principal chacun des fichiers nécessaires. Le fichier `samd20_xplained_pro.h` de trouve sous `src\ASF\sam0\boards\samd20_xplained_pro` ; il définit les noms de certaines broches-clés sur la

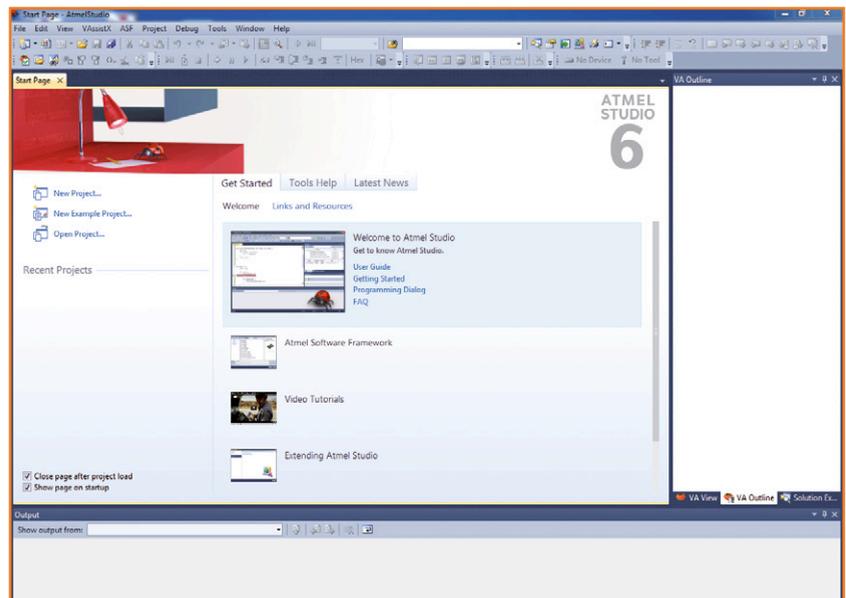


Figure 18.

Voilà à quoi ressemble la page d'accueil d'Atmel Studio 6.2.

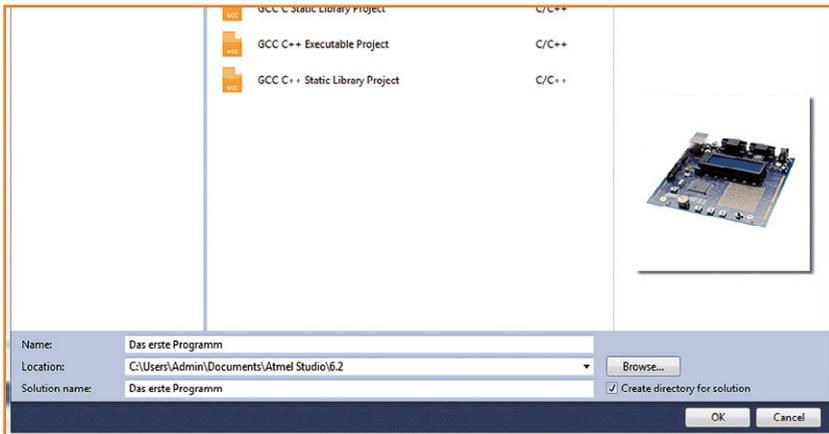


Figure 19.
Il faudra donner un nom à votre projet.

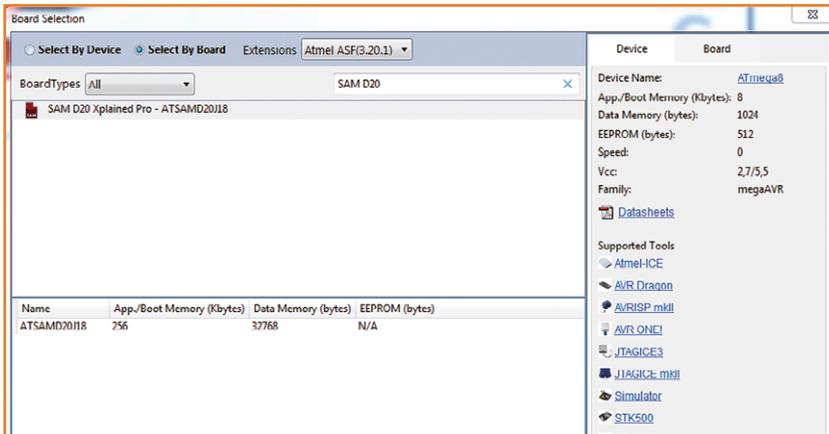


Figure 20.
Avant de lancer la recherche, cochez l'option *Select By Board*.

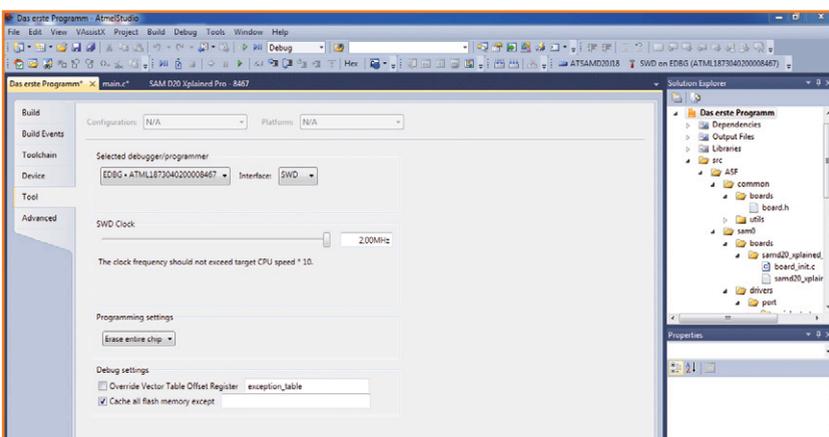


Figure 21.
Studio devrait reconnaître le débogueur sur la carte.

carte (**listage 2**). *Studio* a créé ce fichier automatiquement vu qu'il s'agit d'un projet « carte ». Autres fichiers importants de ce projet : `system.c` dans lequel est définie la fonction de `system_init ()`; et `port.h` dans lequel sont déclarées les fonctions GPIO. Jetez-y un coup d'œil.

Dans les articles à suivre, nous verrons le détail des fonctions, mais nous voulons, maintenant, compiler le projet et le transférer vers la carte. Pour ce faire, appuyez sur la flèche verte *Start Without Debugging* pour limiter le processus à la compilation et au transfert, sans débogage. Auparavant il faudra sélectionner le débogueur (**fig. 21**). Les autres paramètres peuvent rester inchangés. Après la compilation, il se peut qu'il apparaisse un message (**fig. 22**) qui propose d'actualiser le progiciel du débogueur. Appuyez dans ce cas-là sur *Upgrade* et attendez que le transfert du progiciel vers la carte soit effectué.

Appuyez ensuite à nouveau sur la flèche verte, pour réaliser, enfin, le transfert de notre programme. Vous devriez alors, dans la fenêtre *Output*, voir s'afficher un message de succès de ce genre (abrégé) :

```
Program Memory Usage : 1628 bytes
    0,6 % Full
Data Memory Usage      : 8256 bytes
    25,2 % Full
```

```
Build succeeded.
===== Build: 1 succeeded or up-to-
date, 0 failed, 0 skipped =====
```

Appuyons sur le bouton de réinitialisation (*Reset*) pour essayer notre premier programme pour un μC ARM 32 bits.

De belles perspectives

Les présentations sont faites. Nous espérons vous avoir mis l'eau à la bouche, le pied à l'étrier et la puce à l'oreille. Envoyez-moi vos commentaires et vos questions (en allemand ou en anglais) sur le forum d'Elektor [10].

Dans le prochain article, j'aborderai la commande des GPIO et l'interface sur U(S)ART. D'ici là, familiarisez-vous avec la carte, le μC et Atmel Studio en essayant quelques-uns des nombreux exemples de programmes proposés. Vous les trouverez sous l'écran d'accueil soit sous *File/*

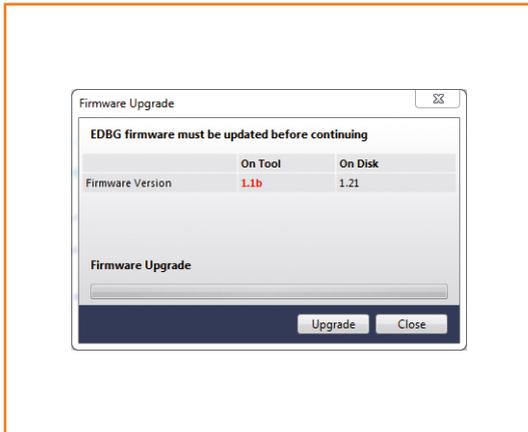


Figure 22.
Lors d'une mise à niveau du progiciel (*firmware*) du débogueur les deux LED d'état devraient clignoter.

New/Example Project... soit sous *New Example Project...* Amusez-vous bien et rendez-vous au prochain article !

(140037 - version française : Guy Raedersdorf)

Liens

- [1] www.atmel.com/Images/Atmel-42102-SAMD20-Xplained-Pro_User-Guide.pdf
- [2] www.atmel.com/Images/Atmel-42096-Micro-controllers-Embedded-Debugger_User-Guide.pdf
- [3] www.atmel.com/images/Atmel-42129-SAM-D20_Datasheet.pdf
- [4] www.atmel.com/tools/atproto1-xpro.aspx
- [5] www.atmel.com/tools/atio1-xpro.aspx
- [6] www.atmel.com/tools/atoled1-xpro.aspx
- [7] www.atmel.com/tools/ATQT1-XPRO.aspx
- [8] www.atmel.com/tools/atmelstudio.aspx
- [9] <http://fr.wikipedia.org/wiki/D%C3%A9bogueur>
- [10] www.elektor.de/forum

Listage 1. Notre premier programme (extrait).

```
#include <asf.h>

int main (void)
{
    system_init();

    // Insérer ici le code de l'application, après
    // initialisation de la carte.

    // Cette structure de code rudimentaire met simplement la
    // LED à l'état du bouton.
    while (1) {
        // Y a-t-il eu action sur le bouton ?
        if (port_pin_get_input_level(BUTTON_0_PIN) == BUTTON_0_
ACTIVE) {
            // Oui, allumer la LED dans ce cas-là.
            port_pin_set_output_level(LED_0_PIN, LED_0_ACTIVE);
        } else {
            // Non, éteindre la LED.
            port_pin_set_output_level(LED_0_PIN, !LED_0_ACTIVE);
        }
    }
}
```

Listage 2. Définitions pour un accès facile des LED et du bouton.

```
#define LED0_PIN                PIN_PA14
#define LED0_ACTIVE             false
#define LED0_INACTIVE           !LED0_ACTIVE

#define SW0_PIN                 PIN_PA15
#define SW0_ACTIVE              false
#define SW0_INACTIVE            !SW0_ACTIVE
#define SW0_EIC_PIN             PIN_PA15A_EIC_EXTINT15
#define SW0_EIC_MUX              MUX_PA15A_EIC_EXTINT15
#define SW0_EIC_PINMUX           PINMUX_PA15A_EIC_EXTINT15
#define SW0_EIC_LINE             15

#define LED_0_NAME               "LED0 (yellow)"
#define LED_0_PIN                 LED0_PIN
#define LED_0_ACTIVE              LED0_ACTIVE
#define LED_0_INACTIVE            LED0_INACTIVE
#define LED0_GPIO                 LED0_PIN

#define BUTTON_0_NAME            "SW0"
#define BUTTON_0_PIN              SW0_PIN
#define BUTTON_0_ACTIVE           SW0_ACTIVE
#define BUTTON_0_INACTIVE         SW0_INACTIVE
#define BUTTON_0_EIC_PIN          SW0_EIC_PIN
#define BUTTON_0_EIC_MUX           SW0_EIC_MUX
#define BUTTON_0_EIC_PINMUX       SW0_EIC_PINMUX
#define BUTTON_0_EIC_LINE         SW0_EIC_LINE
```



kit Hillstar de Microchip

une puce de commande tactile et gestuelle

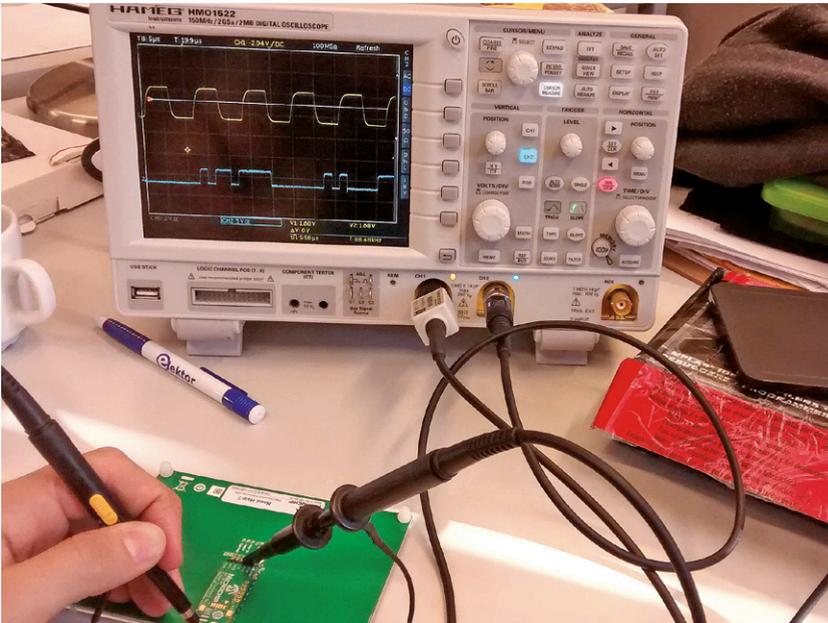


Figure 1. Oscillogramme, en haut : porteuse sur Tx ; en bas : flux de données SDA. À gauche : la main « conductrice » ; à droite, sur l'emballage de la plaque : le TouchPad 3D qui fait partie du lot de produits Elektor/Microchip.

Jaime González Arintero Berciano
(elektor.labs)

Le kit de développement *Hillstar* de Microchip est le point de départ idéal pour un système à μC ou PC commandé par gestes, sans contact. Elektor vous a sûrement déjà mis l'eau à la bouche dans ses numéros de novembre [1] et décembre [2]. J'ai moi-même fait connaissance avec le kit Microchip/Elektor pour un parcours initiatique. La plaque sensible est un circuit imprimé à 4 couches qui contient les électrodes : un émetteur et cinq récepteurs au total. Une petite carte avec la puce MGC3130 se branche directement sur la plaque au moyen d'une barrette à 7 picots, pour les six électrodes et la masse. C'est la puce qui gère toute la magie, elle produit la porteuse pour l'électrode émettrice, elle conditionne, numérise et traite les signaux issus des cinq récepteurs et les rassemble en série par I²C pour votre usage. Pour communiquer avec le PC, le kit offre aussi une passerelle I²C vers USB avec convertisseur

Un chef d'orchestre dirige un orchestre entier sans bouton ni clavier, pas même une souris ; rien qu'une baguette qui, d'ordinaire, ne frappe personne. Vous n'avez pas d'orchestre, mais plein d'appareils à commander sans les toucher ? Microchip propose le contrôleur GestIC MGC3130, prêt à se faire embarquer, avec son kit de développement pour construire, enfin, ces commandes sans contact dont vous rêviez. C'est encore un peu de la science-fiction, mais ça marche déjà en réalité.

de 5 V en 3,3 V, la tension de service de la puce MGC3130. Le kit contient aussi quatre blocs de mousse et un morceau de feuille de cuivre pour un étalonnage poussé. Grâce à un arrangement particulier avec Microchip, vous trouverez dans notre e-choppe [3] le kit d'évaluation *Hillstar* accompagné d'une plaque tactile prête à l'emploi, moins cher que par Microchip Direct.

Lire avant de s'en servir ? Bof !

Lire la notice d'un nouveau produit avant de jouer avec, c'est pas pour moi. Généralement, c'est en me plantant que j'apprends le plus. Mais est-ce encore de l'électronique que tout cela ?

J'avais bien quelques notions (élémentaires) de physique au sujet de ce genre de phénomènes et j'ai tout de suite attaqué, sans logiciel ni pilote, c'est l'oscilloscope qui m'explique. Que se passe-t-il dans le module MGC3130 ? Je suppose que TX veut dire émetteur. *iOie!* Voilà un magnifique signal à 88 kHz. Il était supposé être dans la gamme de 100 kHz, tout va bien. Au retour d'une pause-café, je m'aperçois que le signal

a sauté sur 103 kHz... *¿Qué pasa?* Il monte à 115 kHz puis retombe à 88 kHz, je m'interroge, je consulte la documentation. Le MGC3130 change automatiquement de fréquence en fonction des bruits extérieurs. Comment l'aurais-je deviné ? Ce qui arrive avant que le signal ne soit traité n'est pas sorcier. Les électrodes de l'émetteur engendrent un champ électrique dont les variations, s'il est affecté par votre main ou un objet, sont détectées par les électrodes réceptrices. Il y a dès lors 10 condensateurs à considérer : cinq entre les électrodes Rx et la masse et cinq entre les Rx et l'électrode Tx. À partir de ces informations, la puce calcule la position du corps étranger et envoie les données en paquet dans un flux à travers le bus I²C pour être observées convenablement sur l'oscilloscope.

Enfin, après une bonne lecture, j'ai installé le logiciel joint Aurea, une suite bien structurée et complète de test et de conception, de quoi remettre à plat le système dans son ensemble, modifier tous les paramètres, recommencer l'étalonnage et flasher un nouveau fichier de bibliothèque dans la puce. En fait, Aurea donne des explications probantes sur l'affaire des sauts de fréquence d'émission. Ce logiciel, avec toute la documentation et les références, est disponible ici [4]. La quantité d'information sur le MGC3130 GestIC est énorme et bien organisée.

Touche ou pas touche ?

Il est agréable de jouer avec l'interface graphique Aurea par l'intermédiaire de ses différents assistants (*wizard*) de configuration et beaucoup d'options intuitives. Mais il y a des aspects qui méritent qu'on en parle, spécialement à ceux qui n'aiment pas lire le manuel *in extenso*.

La plaque est très sensible, vérifiez bien qu'il n'y a pas de perturbateur dans les parages qui affecterait l'opération. Le fond de la plaque est blindé et pourtant, une fois posée sur votre bureau, tout ce qu'il y a en dessous peut déranger le champ, que ce soit le chien ou même vos jambes.

En plus d'être un périphérique d'interface utilisateur sans contact, donc sensible aux gestes, il sert aussi de pavé tactile ordinaire, capable de détecter jusqu'à cinq doigts en même temps. Il n'est pas certain que ce soit déjà possible, mais comme on peut aussi toucher chacune des électrodes à l'entour (N, S, E, W), on peut envisager d'en faire une commande à glissière dans de prochaines applications.

L'interface graphique d'Aurea comporte un assis-

tant pour le paramétrage. C'est ici qu'entrent en scène les étonnants blocs de mousse. Vous emballez le plus grand dans la feuille de cuivre fournie et vous y soudez un fin fil pour le relier à la masse. Vous pourriez aussi tenir le fil en main si vos pieds touchent le sol. Toute la procédure est décrite dans le manuel et c'est marrant à faire, sachant qu'on joue là sur des fractions de picofarad. L'évaluation de l'électrode et la linéarisation du champ électrique sont réalisées au moyen de ce bloc « cuivré » et des autres blocs. Je l'ai compris maintenant, mais pas du premier coup !

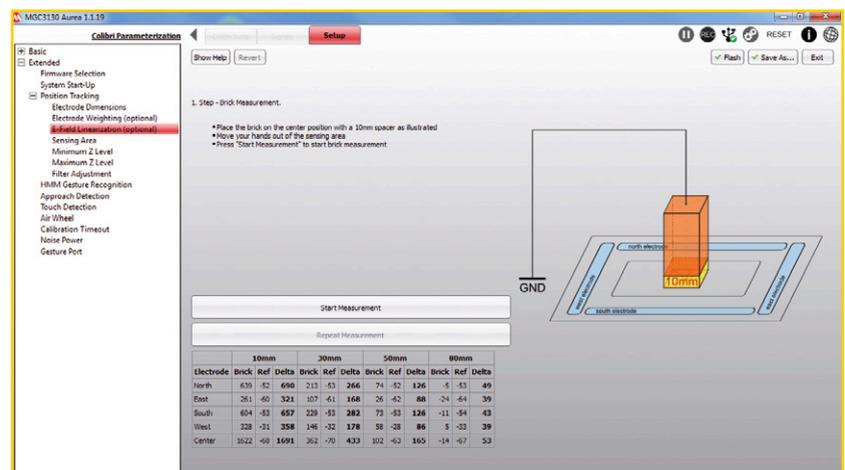


Figure 2. Étalonnage de précision et paramétrage du système en utilisant les blocs de mousse de polystyrène présents dans le kit d'évaluation Hillstar.

La plaque *Hillstar* est faite d'un circuit imprimé nu recouvert d'une couche de plastique transparent et, avec tous ses outils logiciels, se destine aux concepteurs de systèmes enfouis. L'offre spéciale d'Elektor [3] contient en plus une plaque 3D prête à l'emploi, dans un coffret commercial, pour tous ceux qui veulent développer des applications sur PC ou tablette. La documentation recommande de ne pas utiliser de plastique conducteur, sous prétexte que le plastique noir peut contenir du carbone conducteur. Or, la plaque tactile est... noire. Exempte de carbone peut-être ? Je me le demande.

(140434 - version française : Robert Grignard)

Liens

- [1] Commande par gestes 3D pour µC et PC
Elektor, n° 437, nov. 2014, p. 75
- [2] GestIC & 3D TouchPad : manip. n°1
Elektor, n° 438, déc. 2014, p. 8
- [3] www.elektor.fr/microchip-dm160218-hillstar-development-kit-and-dm160225-3d-touchpad
- [4] <http://j.mp/MGC3130>



17^e jour : plus sur les composants

Neil Gruending
(Canada)

Après avoir exploré la création de composants simples, aujourd’hui nous créerons des composants multiples, comme les portes logiques.

Les composants créés jusqu’ici ne comportaient qu’une seule porte et une seule empreinte pour circuit imprimé, ce qui couvre la plupart des composants courants. Aujourd’hui nous explorerons la capacité de *DesignSpark* à gérer les portes et empreintes multiples.

Composants à portes multiples

Chaque composant doit, dans DS, posséder au moins un symbole pour le représenter sur un schéma. La plupart du temps un symbole suffit mais certains composants, comme les circuits logiques et amplis op comportent plusieurs sous-composants, que DS appelle des portes, dans un même boîtier. DS permet dans ce cas d’associer plusieurs symboles à un même composant. Habituellement il s’agit de plusieurs symboles identiques mais, tant que vous utilisez toutes les broches de l’empreinte du circuit intégré, rien de vous

empêche de combiner différents symboles à votre guise.

Nous allons créer un composant pour portes NAND 74HC00 comportant plusieurs symboles de schéma. Je n’aimais pas trop le symbole pour portes NAND qui se trouvait dans les bibliothèques de DS et en ai donc rapidement dessiné un. J’ai ensuite créé une empreinte pour boîtier SO à 14 broches afin d’avoir tout sous la main pour l’assistant de création de composant. Démarrez l’assistant puis, sur la page *Component Details*, saisissez ‘4’ pour le nombre de portes (**fig. 1**) afin d’informer l’assistant que nous voulons utiliser quatre symboles de schéma pour notre composant NAND. Nous nous soucierons des alimentations plus tard. Les prochaines étapes consistent à sélectionner le symbole de schéma et l’empreinte que vous voulons utiliser. L’assistant vous demandera aussi d’associer les connexions des symboles à l’empreinte. Je préfère en général utiliser le bouton *Assign 1-to-1* puis changer le brochage plus tard dans l’éditeur de composant.

Vous pourrez ensuite éditer le composant une fois sorti de l’assistant (**fig. 2**). La partie gauche de la fenêtre montre chacune des portes dans une grille qui rendra facile l’éditition des associations de broches. Chaque porte est nommée avec une lettre, à partir de *a* et chaque broche est représentée par une ligne dans la grille. Les numéros de broches de l’empreinte sont listés dans leur propre colonne ; je préfère cela pour configurer les associations de broches. Je trouve confuse la présentation de l’assistant. La fenêtre d’éditition de composant permet de vérifier en un coup d’œil tous les symboles des portes et l’empreinte. Une fois terminé, vous placerez notre composant 74HC00 sur un schéma, comme n’importe quel autre composant (**fig. 3**). Les portes NAND en noir ont déjà été placées sur le schéma contrairement à celles en rouge : elles sont dessinées près du pointeur de souris pour vous montrer le nombre de portes qu’il reste à placer dans le

Figure 1. Saisie des détails dans l’assistant de création de composant.

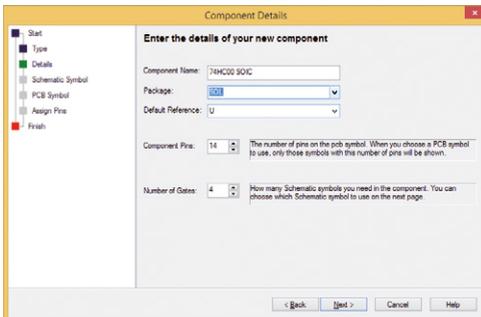
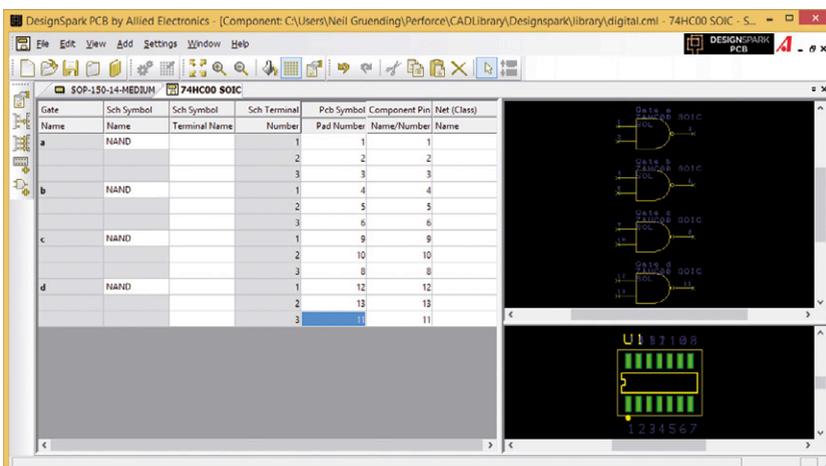


Figure 2. L’éditeur de composant.



composant. Ici, il reste quatre portes à placer. DS se souviendra du nombre de portes déjà placées depuis le composant et vous n'aurez donc pas à vous en soucier. Par exemple, si vous placez une porte NAND puis quelques résistances, DS se souviendra que seule la porte **a** du 74HC00 a été utilisée. La prochaine fois que vous placerez une porte NAND, DS commencera automatiquement à la porte **b** et vous montrera qu'il reste trois portes disponibles.

DS s'assure également que vous n'utilisiez pas accidentellement deux fois la même porte d'un même composant physique. Par exemple, la figure 3 montre que j'ai déjà placé les quatre portes de U1, nommés U1a à U1d. Pour échanger deux portes après que vous les avez câblées, il suffit de double-cliquer sur les portes et choisir la lettre que vous souhaitez utiliser. DS vous avertira que la porte est déjà utilisée dans le projet et vous proposera d'échanger les portes. Par exemple, si je voulais échanger les portes **a** et **c**, je pourrais manuellement changer la porte **a** en **c** et dans la fenêtre *Gate Properties* et DS me proposerait de renommer la porte **c** en **a**.

Broches d'alimentation

Notre 74HC00 comporte maintenant le bon nombre de portes NAND mais il ne fonctionnera pas correctement : notre composant ne comporte aucune broche d'alimentation. Il y a plusieurs façons d'y remédier : utiliser les broches d'alimentation cachées, en placer sur chacune des portes ou créer un symbole séparé pour les alimentations. Les deux dernières options sont prises en charge par DS, jetons-y un coup d'œil. Ajouter les broches d'alimentation à chacune des portes n'a de sens que si elles sont utilisées dans un cadre limité et seulement à l'intérieur de la porte. DS prend cela en charge en vous permettant d'associer plus d'une broche de schéma à chacune des broches de l'empreinte. Dans notre 74HC00, chacune des portes pourrait avoir des broches d'alimentation associées aux mêmes broches d'alimentation de l'empreinte. En ce qui me concerne, je préfère que chaque broche de l'empreinte n'ait qu'une connexion et j'utilise donc une porte séparée pour représenter les alimentations. L'assistant de création de composant ne permet pas de choisir différents symboles de portes et il vous faudra donc le faire depuis l'éditeur de composant avec l'entrée *Add → Gate* du menu. Cela ouvrira la fenêtre *Add Gate* qui vous permettra de choisir le symbole à ajouter. La **figure 4** montre

ma porte d'alimentation une fois les associations de broches effectuées.

Empreinte multiple d'un composant

Les composants peuvent également avoir plusieurs empreintes, c'est une fonction très pratique. DS ne requiert pas que toutes les empreintes aient le même nombre de pattes. Il est cependant plus facile de s'y astreindre : toutes les empreintes peuvent utiliser la même association de broches. Vous pouvez ajouter des empreintes à un composant en utilisant l'entrée *Add → Package* du menu qui ouvrira la fenêtre *Add Package*.

Cette fenêtre vous permettra de sélectionner l'empreinte à ajouter et d'y associer un nom de boîtier. Des noms prédéfinis existent mais vous pourrez aussi saisir celui de votre choix. Les noms de boîtier sont affichés lorsque vous placez le composant.

La **figure 5** montre la fenêtre *Add Component* lorsque vous placez votre 74HC00 et cliquez sur *Package*. Pour cet exemple, j'ai choisi que SOL corresponde à une empreinte SOIC étroite et SOIC WIDE à une empreinte SOIC large. Une fois l'empreinte choisie, vous pourrez la placer comme d'habitude. Si vous voulez changer l'empreinte d'un composant déjà présent dans le schéma, il faudra ouvrir la fenêtre de propriétés du composant et cliquer sur le bouton *Change*. La fenêtre *Change Component* s'ouvrira et vous permettra de choisir la nouvelle empreinte.

La prochaine fois, nous jetterons un œil aux outils de disposition de circuits intégrés de *DesignSpark*.

(140419 - version française :
Kévin Petit)

Figure 3.
Placer un 74HC00.

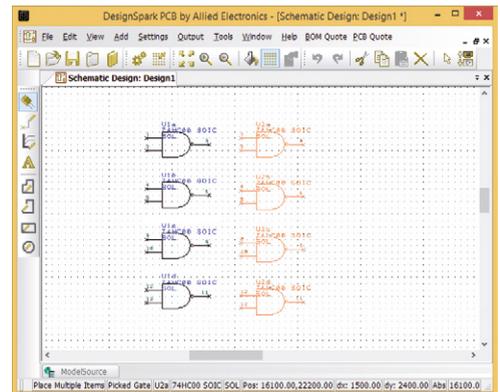


Figure 4.
Symbole final de notre 74HC00.

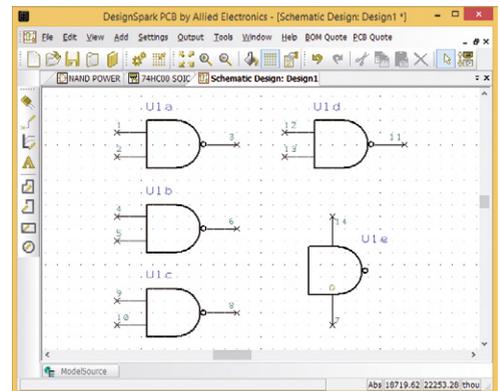
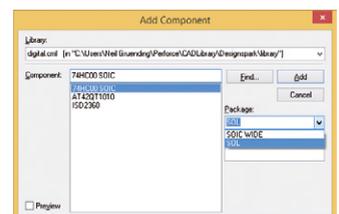


Figure 5.
Placer un composant à empreintes multiples.



Digi-Disco 1978

jusqu'au bout du 7406

Dan Koellen
(États-Unis)

Printemps 1978, "Burn Baby Burn, Disco Inferno" emplit un gymnase plein de lycéens dansant au son des tubes disco. Assis sur la scène, près du DJ, je vérifie les niveaux sonores et agite frénétiquement des interrupteurs pour tenter de faire correspondre l'éclairage au rythme de la musique.

J'en ai vite eu marre d'animer l'éclairage à la main, c'est pourquoi j'ai démarré la conception du Digi-Disco, un système de commande d'éclairage automatique réagissant à la musique. Une fois installé, il ne nécessite aucune interaction, à part si l'on souhaite changer le mode de fonctionnement. Comme j'étais étudiant à plein temps, il me fallait attendre les vacances d'été pour commencer la construction.

Conception

Comme je voulais un système le plus automatique possible, j'ai choisi d'utiliser la musique pour changer les motifs d'éclairage. Aussi, la mode du disco avait rendu les chenillards très populaires. Il fallait donc que le système en soit capable. De plus, l'éclairage devait être suffisamment puissant

pour illuminer complètement un gymnase. Chacune de nos soirées se déroulait dans un endroit différent et il fallait donc que le tout soit facile à installer, démonter et stocker.

Les microcontrôleurs n'étaient pas aussi répandus qu'aujourd'hui et j'ai donc utilisé des circuits logiques discrets et des composants, à l'époque (1978) très courants : C.I. TTL de la série 74XX pour la partie numérique, C.I. bipolaires pour l'analogique et temporisateur 555. Sans les eBay, Amazon ou autres sites de déstockage d'aujourd'hui, il me fallait utiliser des composants qui se trouvaient déjà dans mes tiroirs ou étaient disponibles au magasin d'électronique du coin ou chez Radio Shack. Les achats par correspondance étaient populaires mais peu pratiques et lents. Le Radio Shack du coin a fourni les comparateurs de tension, boîtiers et triacs. Les circuits logiques CMOS ou de la série TTL 74LS existaient mais n'étaient ni accessibles, ni dans mes moyens.

Le Digi-Disco a été construit en deux parties. La première (**fig. 1 et 2**), était une console avec la majeure partie du circuit : sélection du mode de fonctionnement et de l'échantillonnage du son. Un boîtier déporté (**fig. 3**) contenait les triacs utilisés pour commander les lampes et alimentait la console. La console et le boîtier déporté étaient reliés par un câble à 11 conducteurs. La version finale alimentait quatre barres de six spots couleur de 75 W. Chacun des six spots était commandé séparément mais de la même façon sur les quatre barres. Deux des barres étaient accrochées au-dessus de la scène du DJ, les deux autres étaient réparties sur les côtés de la scène, reposant sur les grosses enceintes *Voice of the Theater* que nous utilisions. Le boîtier déporté était suspendu entre les deux barres, au dessus de la scène du DJ ; des câbles à huit conducteurs



La toute première vidéo Rétronique a été enregistrée avec des appareils numériques et postée sur www.elektor.tv.

Regardez les visiteurs du salon *electronica 2014* à Munich confrontés à l'ElektorScope de 1976 et au générateur de signal à tubes Rohde & Schwarz des années 60 ouvert pour l'occasion. Plus de vidéos Rétronique et rares seront prochainement mises en ligne sur elektor.tv.

Rétronique envahit elektor.tv



étaient utilisés pour y relier les barres. Le tout consommait 1800 W en pointe (16 A_p). Nous pouvions illuminer complètement un gymnase avec des éclairages animés mais il fallait un circuit d'alimentation séparé juste pour l'éclairage. Pour que le montage et démontage soient faciles, toutes les connexions étaient réalisées avec des câbles qui se branchaient dans la console, le boîtier déporté et les barres. Les câbles comportaient un nombre de broches différent et ne pouvaient donc pas être branchés dans la mauvaise prise. Chacune des barres était munie d'un code de couleur afin de pouvoir être facilement installée et orientée. Le plus difficile dans l'installation était de s'assurer que la séquence des couleurs des ampoules était correcte.

Circuit

Le schéma original dessiné à la main se trouve en **figure 4** ; les étudiants n'avaient pas accès aux programmes de CAO à l'époque. Tout le circuit est dans la console, excepté la partie en bas à droite qui comprenait les triacs et le transformateur d'alimentation.

L'audio était échantillonné directement depuis une enceinte et attaquait le potentiomètre et la diode de la partie *audio in* à la gauche du schéma. J'ai utilisé une diode au germanium 1N34A pour sa tension de chute en polarisation directe plus faible que celle d'une diode au silicium ; le potentiomètre de 10 kΩ permet d'ajuster le niveau du signal audio. La diode fournit un signal audio redressé, A.R. sur le schéma. Le signal rectifié n'était pas filtré mais cela fonctionnait très bien pour l'effet que nous cherchions. J'avais pensé à utiliser un redresseur actif et du filtrage mais ne l'avais pas considéré nécessaire.

Le signal audio redressé est utilisé par deux parties du circuit : celle qui commute les lampes suivant l'amplitude de l'audio et une autre qui fait évoluer un compteur lié à la musique. Au milieu du schéma se trouvent les comparateurs de tension LM339 IC3 et IC4 (la version de Radio Shack : RS339) dont la tension de référence provenait d'un diviseur à résistances à plusieurs étages : leurs sorties passent à l'état bas lorsque l'amplitude du signal audio redressé dépasse la valeur de référence correspondante. Le nombre de lampes allumées augmentait avec le niveau du signal audio, comme un gros vu-mètre. L'échelle de résistances était conçue pour bien correspondre à l'audio, pas pour faire des mesures ou garantir une charge constante.

Pour l'effet chenillard, un 555 configuré en multivibrateur astable est utilisé pour cadencer le (dé)compteur à décade 74190. La sortie BCD de ce dernier est décodée en décimal par un 7442, comme le montre le haut du schéma. Le compteur à décade peut être cadencé directement par le 555, par les impulsions issues d'un



Figure 1. La console gère le fonctionnement des effets lumineux ; l'usure visible sur la photo montre que la console a souvent servi. Le « PHASE » que l'on voit en bas à droite était le nom d'animateur son et lumière que j'utilisais dans les soirées.

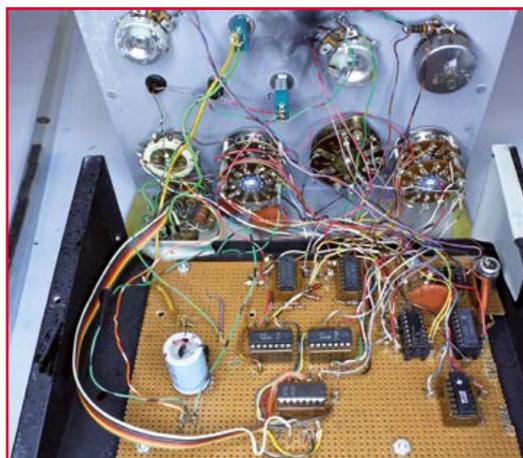


Figure 2. Le circuit contenu dans la console était construit sur une plaque d'essais. La suite que l'on peut voir en haut, au centre, provient des 7406 qui avaient cramé. Sur la gauche on peut voir le tube de circuits de remplacement que j'apportais dans les soirées.



Figure 3. Le boîtier déporté contenait les triacs pour commuter les lampes. En bas se trouvent les prises pour les connexions aux barres de lampes ; le câble en provenance de la console se branchait sur la prise en bas à gauche. Le transformateur fournissait une tension alternative abaissée à la console.

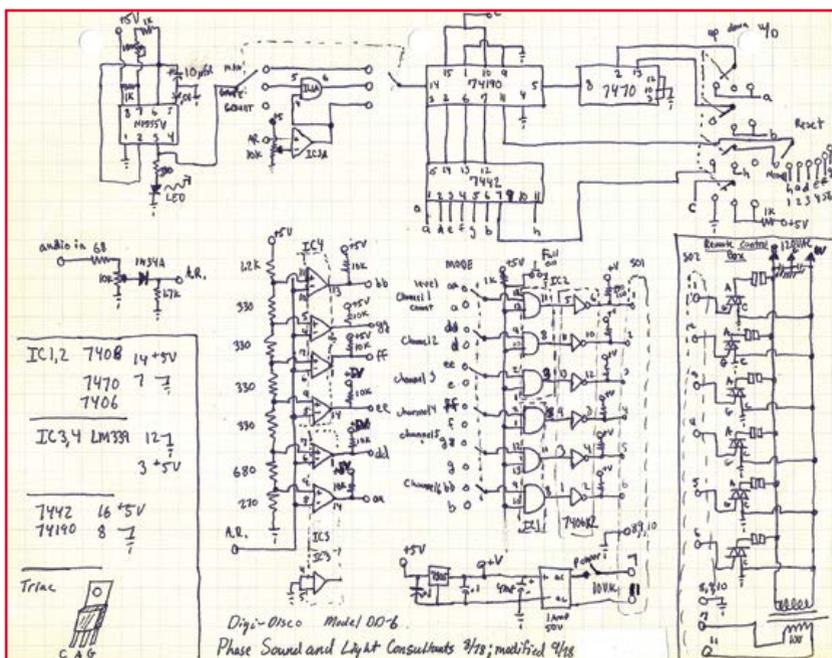


Figure 4. Schéma dessiné à la main du Digi-Disco. « Phase Sound and Light Consultants » était le nom que j'utilisais à l'époque

ET logique, réalisé par une porte 7408, entre les impulsions du 555 et la sortie du comparateur de tension ou enfin directement par la sortie du comparateur. L'entrée dédiée de réglage du sens de comptage et la possibilité de préchargement font du 74190 le circuit idéal pour cette application ; deux caractéristiques que le 7490, alors très utilisé, ne possédait pas. J'étais capable de produire trois motifs de chenillard : progression de la voie 1 à la voie 6 avec remise à zéro sur la voie 2 ; la même progression en sens inverse ; et une progression en aller-retours entre les voies 1 et 6. Le 7470 est une bascule JK avec preset et clear, que j'utilisais pour régler le sens de comptage du 74190 et charger 0 ou 5 en fonction du sens. Pour les aller-retours, le préchargement n'était pas utilisé et je me contentais d'inverser le sens du comptage arrivé à 0 ou 5.

Comme le signal audio redressé est toujours relié au circuit analogique de détection de niveau et au compteur, le mode de fonctionnement est sélectionné en reliant les sorties adéquates au 7408 (IC1 et IC2) et au pilote de sortie 7406 à l'aide du commutateur de mode représenté au milieu du schéma. Cela permettait de changer le mode de fonctionnement très rapidement. Le 7406 est un inverseur à collecteur ouvert « haute-tension » pouvant absorber 40 mA. Les triacs utilisés pour commuter les lampes demandaient un courant de gâchette d'au moins 50 mA pour basculer. Le

pilote devait pouvoir absorber ce courant quand les triacs étaient bloqués. C'est pourquoi j'avais soudé deux 7406 l'un sur l'autre (le schéma n'en montre qu'un). Lorsqu'une sortie du 7406 était inactive, le courant de gâchette était envoyé sur le câble à onze conducteurs pour déclencher le triac correspondant dans le boîtier déporté.

Vous aurez peut-être remarqué que les triacs n'étaient pas isolés électriquement. Pour ma défense, les optocoupleurs étaient, contrairement aux circuits TTL, très onéreux en 1978, surtout pour un étudiant. J'ai décidé de construire le montage tel quel mais je savais bien qu'il me faudrait avoir des C.I. de remplacement sous la main. Ne faites pas comme moi : ce n'est ni sûr, ni fiable. Les 7406 ont cramé de manière catastrophique durant le premier événement où j'avais utilisé le Digi-Disco. J'avais remarqué qu'il valait mieux éviter les chansons avec trop de distorsion et isoler l'alimentation des lampes du système de sonorisation pour éviter les pannes.

Utilisation

Le Digi-Disco fonctionnait très bien, surtout avec les basses régulières du Disco pour lesquelles il avait été conçu. Il s'est avéré facile à monter et démonter. La plupart du temps j'utilisais le mode en aller-retours avec le compteur cadencé par le signal issu de la porte ET ; le 555 était réglé pour une vitesse modérée d'environ 2 Hz. Cela donnait l'apparence que les lampes dansaient avec un motif aléatoire qui changeait avec le rythme de la musique. Le mode vu-mètre rendait plutôt bien lorsque le DJ parlait.

Les jeunes qui venaient aux soirées adoraient les effets lumineux, et le Digi-Disco avait vite fait parler de lui. Grâce à ça nous avons pu augmenter à la fois le nombre de soirées que nous faisons et nos tarifs. Et, cerise sur le gâteau, plus besoin de tripoter des interrupteurs mécaniques : le Digi-Disco fonctionnait très bien avec peu d'interventions de l'opérateur.

Comparé au Digi-Disco, le *Programmable Disco Array* paru dans Elektor en 1984 était beaucoup plus avancé avec ses 32 motifs lumineux stockés en mémoire, ses circuits CMOS, son boîtier rack 19 pouces et ses afficheurs à sept segments. Ce projet a été commémoré dans Rétronique en février 2008.

(140420 - version française : Kévin Petit)

EST^D 2004

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



distorsion à OTA pour guitare

La guitare électrique, jouée (même sommairement) avec les deux mains, est indissociable des effets dont on joue avec les pieds. Un des (mal)traitements du son de l'instrument consiste à en saturer le signal pour enrichir le timbre. Cette forme de distorsion est obtenue avec un bon vieux circuit analogique, avec des diodes au germanium (triées) et des amplificateurs opérationnels à transconductance (OTA).



Platino : testeur de transistors

Dans le numéro que vous avez entre les mains, nous présentons un générateur de fonction, le 2^e d'une série d'appareils de mesure basée sur la carte Platino d'Elektor. Bientôt ce sera le tour d'un testeur de transistors de types variés : bipolaires, JFET et MOSFET. Selon le type du transistor testé, son brochage et divers paramètres sont affichés par un LCD sur 4 lignes.



e-BoB BL600

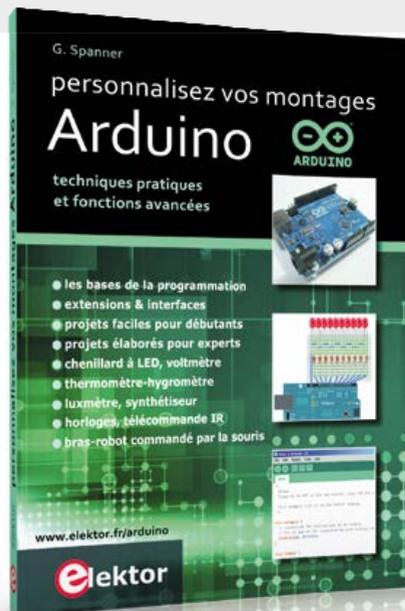
Le mot « *BoB* » apparaît dans pas moins de 3 titres d'articles de ce numéro. C'est le signe d'une tendance : Elektor étendra encore sa collection de *Break out Boards*, pour permettre à tous les électroniciens d'utiliser facilement des modules dont la forte miniaturisation compromet la maniabilité. Le prochain sera e-BoB BL600, avec le module Bluetooth BL600-SA de *Laird Technologies*, déjà utilisé ce mois-ci pour le thermomètre de la page 72.

Sous réserve de modification

Le numéro de mars paraîtra le 24 février.

Personnalisez vos montages **Arduino**

techniques **pratiques** et fonctions **avancées**



NOUVEAU

L'objectif de ce livre est de vous emmener à pas guidés vers la maîtrise d'Arduino. Les projets, regroupés par thème, accompagnés de bases théoriques, sont des applications concrètes : chenillard à LED, voltmètre, thermomètre numérique, horloges sous différentes formes, ou encore bras de robot commandé par la souris.

Vous apprendrez ainsi à exploiter des techniques essentielles comme la conversion analogique-numérique, la modulation de largeur d'impulsion, ou encore les interruptions. Après avoir mené à bien tous ces projets vous maîtriserez les fondamentaux de la technique des microcontrôleurs.

272 pages | ISBN 978-2-86661-191-0 | 34,50 €

elektor

www.elektor.fr/arduino

hexadoku casse-tête pour elektorniciens

Vous tenez encore sur vos cannes après les excès de fin d'année... ? Un peu de gymnastique*, peut-être ?

« Idées écartées | frissons légèrement en dehors | la vie derrière les idées | Élever les angoisses tendues | au-dessus de la tête | Marquer un léger temps d'arrêt | et ramener la vie à son point de départ | Ne pas baisser les frissons | et conserver le vide très en arrière. » Remplissez la grille selon les règles, envoyez-nous votre solution, et vous gagnerez peut-être. Vous serez en tout cas d'aplomb pour reprendre ensuite vos cogitations.

* extrait de *Quart d'heure de culture métaphysique*, Ghérasim Luca, Le chant de la carpe, Librairie José Corti

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

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

Participez et gagnez !

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

Où envoyer ?

Envoyez votre réponse (les chiffres sur fond grisé) avec vos coordonnées par courriel, avant le **1^{er} mars 2015** à l'adresse **hexadoku@elektor.fr**

Les gagnants

La solution de la grille du numéro de novembre (437) est : **63D95**

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

Andrej Marn (Slovénie), **Gerrit van Leeuwen** (Pays-Bas), **Jozef Bouwen** (Belgique), **François Jongbloet** (France) et **Larry Burns** (Canada).

Bravo à tous les participants et félicitations aux gagnants !

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

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

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.

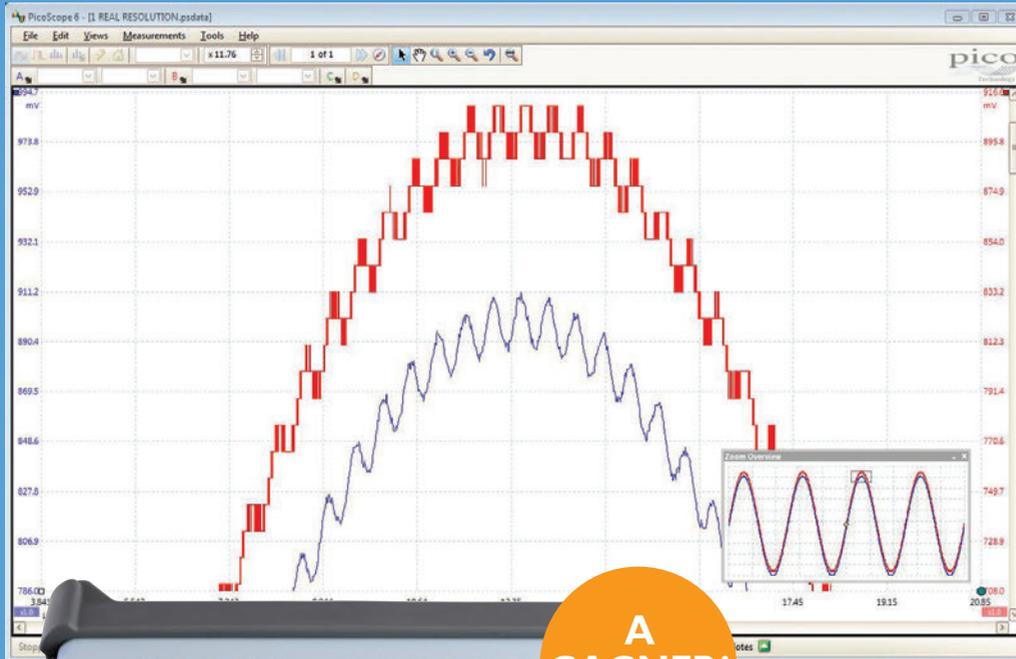
GAGNEZ UN



PicoScope® Série 5000

OSCILLOSCOPE A RESOLUTION FLEXIBLE

OSCILLOSCOPES SERIE PICOSCOPE 5000 AVEC UNE RESOLUTION FLEXIBLE REGLABLE DE 8 A 16 BITS JUSQU'A 1 GÉCH/S



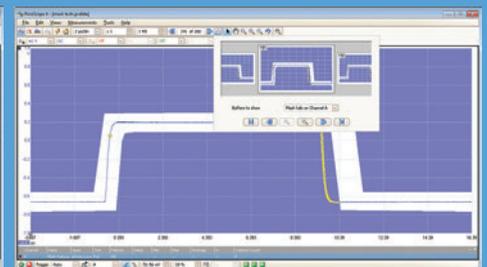
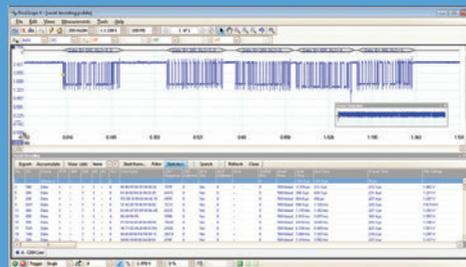
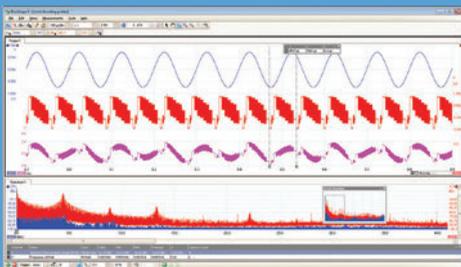
L'électronique moderne fait appel à une variété de signaux très rapides et à haute résolution. Etre capable de détecter et de caractériser de petits signaux en présence de plus grands est essentiel pour la vérification de conceptions électroniques la prochaine génération. La précision d'un oscilloscope est déterminée par sa résolution et son exactitude. Voici des caractéristiques d'oscilloscopes avec différentes résolutions:

La courbe supérieure de l'écran capturée avec 8 bits de résolution et zoomée en 64x montre les limites de la résolution 8 bits. Le même signal capturé avec un Picoscope réglé en résolution 12 bits montre des caractéristiques du signal invisibles en mode 8-bits.

A GAGNER!



Résolution de l'oscilloscope	Niveaux de quantification	Plus petite variation pouvant être détectée (ou pleine échelle)	Gamme Dynamique Maximale
8 Bits	256	0.39% (4,000 ppm)	48 dB
10 Bits	1024	0.097% (976 ppm)	60 dB
12 Bits	4096	0.024% (244 ppm)	72 dB
14 Bits	16384	0.0061% (610 ppm)	84 dB
16 Bits	65536	0.0015% (15 ppm)	96 dB



TOUS LES MODÈLES INCLUENT LE LOGICIEL ET UNE GARANTIE DE 5 ANS. LE LOGICIEL INCLUT LES MESURES, L'ANALYSEUR DE SPECTRE, LE SDK, LES DÉCLENCHEMENTS AVANCÉS, LE MODE PERSISTANCE COULEUR, LES DÉCODAGES SÉRIE (CAN, LIN, RS232, I²C, I²S, FLEXRAY, SPI), MASQUES, VOIES CALCULÉES, TOUT EST EN STANDARD, MISES À JOUR GRATUITES.

TAPEZ: www.picotech.com/PS303

Besoin de combiner le multipoint 2D et la reconnaissance de gestes 3D sur un seul et unique périphérique PC ?

Le 3DTouchPad de Microchip vous offre la toute première plateforme de développement tactile à entrée 2D/3D



Microchip présente le 3DTouchPad, un kit de développement prêt à l'emploi et un système de référence qui associe la reconnaissance de gestes 2D multipoint (jusqu'à 10 points de contact simultanés) et de gestes 3D effectués dans l'espace, pour le développement rapide d'interfaces d'entrée pour périphériques PC et autres applications.

Basée sur la technologie GestIC® de Microchip, la reconnaissance des gestes 3D, robuste et innovante, du 3DTouchPad offre une distance de perception maximale de 10 cm, tandis que l'interface de détection tactile 2D capacitive projetée « multi-touch », extrêmement sensible, supporte jusqu'à 10 points de contact simultanés ainsi que les gestes en surface à plusieurs doigts.

Grâce au nouveau driver pour lignes d'écran tactile capacitif à haute tension de Microchip, le MTCH65X, il affiche des performances tactiles capacitives projetées robustes, intègre des capteurs plus gros et un revêtement plus épais en augmentant le rapport signal sur bruit (SNR).

En tant que périphérique PC plug-and-play, le 3DTouchPad se connecte à un ordinateur grâce à un simple câble USB et inclut une interface graphique gratuite (GUI), un kit de développement logiciel (SDK) et une interface de programmation d'application (API). Il offre également des fonctionnalités clé en main qui ne nécessitent pas de pilote et améliorent l'expérience utilisateur sous Windows® 7/8.X et MacOS®.

POINTS CLÉS :

- Kit de développement pour interface d'entrée tactile 2D/3D 3DTouchPad : DM160225
- Driver pour lignes d'écran tactile capacitif projeté 2D MTCH65X
- Technologie GestIC® pour la reconnaissance avancée de gestes 3D

Pour de plus amples informations : www.microchip.com/get/eu3DTouchPad



Microcontrollers • Digital Signal Controllers • Analog • Memory • Wireless

Le nom et le logo de Microchip, GestIC et mTOUCH sont des marques déposées de Microchip Technology Incorporated aux États-Unis et dans d'autres pays. Toutes les autres marques commerciales ci-dessus mentionnées sont respectivement la propriété des sociétés concernées. ©2014 Microchip Technology Inc. All rights reserved. DS30010086A.ME1119Fre12.14