



elektor

DÉCOUVRIR

CRÉER

PARTAGER

radio logicielle réinventée

**Shield
Arduino
comme
récepteur
SDR**

Dans cette édition :
6 projets du labo
7 projets de lecteur
3 articles sur du logiciel
1 banc d'essai

iBaxandall - correcteur de tonalité • nouvelle vie pour une
balise NFC (1) • **pluviomètre avec MPXV5004D et Arduino**

NAIT • cours intensif d'assembleur 2.1 • **programmeur JTAG**

universel / adaptateur de débogage • hors-circuits avec R. Lacoste : ABC des

CA/N • **sonde de courant pour oscilloscopes** • twinBot, téléprésence mobile •

indicateur de charge faible • ScanaQuad, le plus petit analyseur logique • **pondre**

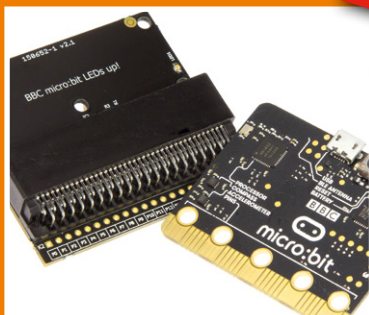
un projet en C • accordeur de guitare (basse) de scène • **MAXQ à la rescousse**



**indicateur
de niveau d'eau**
pour sac d'hydratation



**passerelle
Bluetooth 2**
vers Bluetooth LE



station météo
pour micro:bit



enova

PARIS

Le salon des technologies et des services
ÉLECTRONIQUE | EMBARQUÉ | IoT
MESURE | VISION | OPTIQUE

14-15
SEPTEMBRE
2016
PARIS EXPO
Porte de Versailles

Donnez vie à vos projets

.....AÉRONAUTIQUE.....MILITAIRE.....

.....AGROALIMENTAIRE/AGRICOLE.....

.....AUTOMOBILE/TRANSPORT.....

.....SMART CITIES.....SMART BUILDING.....

.....MÉDICAL.....INDUSTRIE 4.0.....

.....RECHERCHE ACADEMIQUE.....

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

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

Fax : (+33) 01.49.19.22.37

www.elektor.fr | www.elektormagazine.fr

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

Publicité :

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

DROITS D'AUTEUR :

© 2016 Elektor International Media B.V.

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



Chat alors !*

Le centre industriel de stockage géologique (CIGÉO) doit recevoir à Bure (Meuse) des milliers de mètres cubes de déchets nucléaires à haute activité. Eh oui, depuis bientôt quarante ans, les sites de stockage souterrain de déchets radioactifs se multiplient. Parmi les nombreuses interrogations suscitées par cette prolifération se pose la question de savoir comment informer nos descendants de la présence de nos poubelles radioactives et des dangers qu'elles constitueront encore pendant des milliers d'années.

Bienvenue dans le monde de la *sémiotique nucléaire*, c'est-à-dire l'étude et la conception de signes qui, en restant compréhensibles à travers les siècles, alerteront sur la dangerosité de ces sites.

Partant du fait que le chat été déjà vénéré dans l'Égypte antique et de la conviction qu'il demeurerait longtemps encore un animal à forte charge symbolique, deux philosophes, Françoise Bastide et Paolo Fabbri, ont proposé un système *félidé* : des chats qui changeraient de couleur lorsque le niveau de radioactivité augmente*. Cette idée farfelue pourrait devenir une voie de recherche sérieuse.

L'ANDRA, à la recherche d'un moyen de transmission de la mémoire du CIGÉO au moins cinq siècles après sa fermeture, a choisi de mettre les archives du centre sur un support censé rester stable entre 600 et 1000 ans. Cette agence pour la gestion des déchets radioactifs entend aussi reproduire ses informations sur des disques en saphir synthétique dont la durée de vie est estimée à plusieurs millions d'années. Magnifique, mais dans quelle langue ? Les langues évoluent et meurent.

Ne vaudrait-il pas mieux faire comme les sémioticiens de la variante Ikea dont les pictogrammes, pourtant à la limite de l'incompréhensible, n'en font pas moins le bonheur dans les chaumières du monde entier ?

Mariline Thiebaut-Brodier

* Si vous voulez voir chat : www.dailymotion.com/video/x38bf2c_la-solution-radiochats_tv

Notre équipe

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

- 5 bientôt dans Elektor
- 6 Elektor : guide de connexion(s)
- 25 agenda juillet 2016
- 38 ElektorBusiness
- 40 batteries avant-gardistes
l'avenir est en charge
- 44 labo d'Elektor
- 108 l'e-choppe d'Elektor
- 128 des nouvelles du monde d'Elektor
- 130 hexadoku casse-tête pour elektorniciens

DÉCOUVRIR CRÉER PARTAGER

- 8 bienvenue dans la section Découvrir
- 9 descente d'antenne symétrique
drôle de composant n°25
- 10 nouvelle vie pour une balise NFC (1)
accès aux entrailles
- 14 cours intensif d'assembleur 2.1 (2)
adressage indirect
- 18 CircuitMaker
trucs & astuces (2)
- 20 hors-circuits avec R. Lacoste
ABC des CA/N
- 26 quand CE devient UE
une directive chasse l'autre
- 29 twinBot
ou la téléprésence mobile
- 35 Flowcode 7
le logiciel qui fait avancer les choses

DÉCOUVRIR CRÉER PARTAGER

- 46 bienvenue dans la section CRÉER
- 47 station météo micro:bit
et aussi carte d'extension générique !
- 50 SDR d'Elektor réinventé
sur *shield* Arduino

SDR D'ELEKTOR RÉINVENTÉ

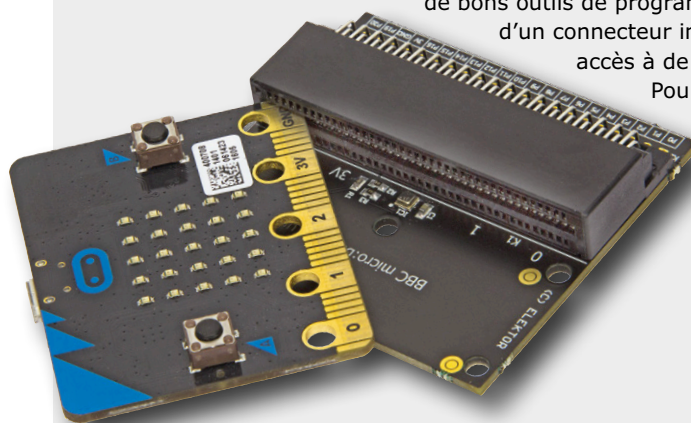
sur *shield* Arduino

50

Une radio logicielle, SDR pour *Software Defined Radio*, est un outil universel de travail en HF, qui permet aussi d'effectuer des mesures. Les caractéristiques du récepteur, c'est le logiciel qui les définit. Pour l'interface de communication, nous disposons à présent de la plateforme Arduino sur laquelle monter un nouveau *shield*

station météo micro:bit et aussi carte d'extension générique !

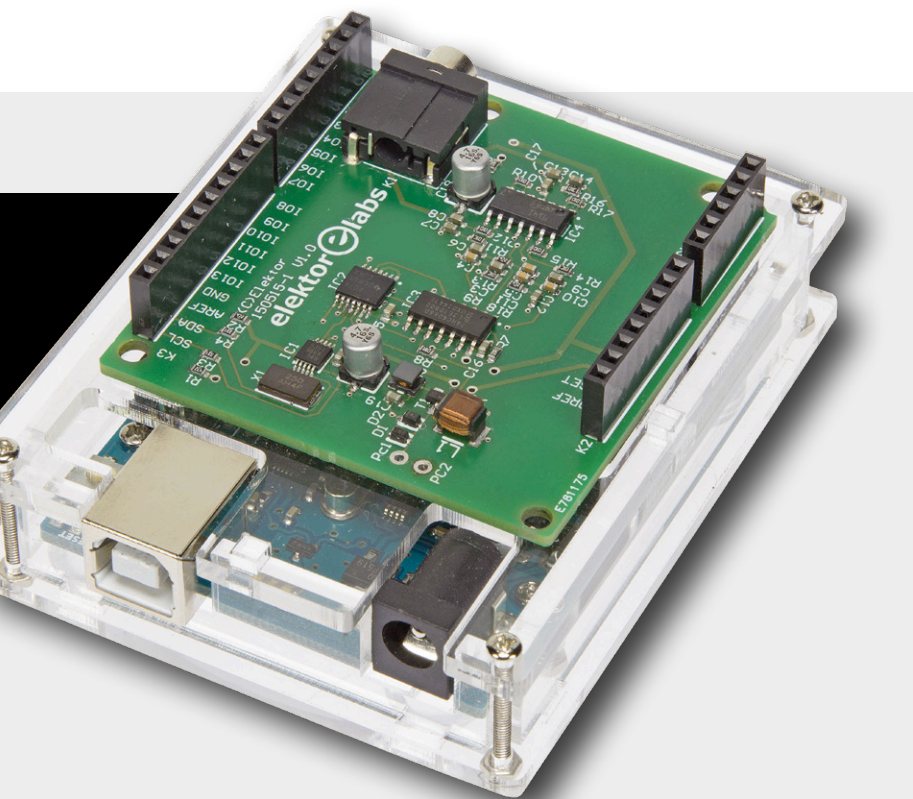
La carte micro:bit de la BBC est une petite centrale à tout faire qui bénéficie d'excellentes bibliothèques de fonctions, de bons outils de programmation et d'un connecteur imprimé donnant accès à de nombreux ports. Pour illustrer son utilisation, nous avons construit une station météo.



47

- 58 indicateur de niveau d'eau
pour sac d'hydratation
sûrement approuvé par Lonely Planet
- 62 iBaxandall
variations sur le correcteur de tonalité
- 70 passerelle Bluetooth 2 vers Bluetooth LE
ou comment consulter le thermomètre BLE
avec un ordiphone Android un peu âgé
- 74 pluviomètre
avec MPXV5004D et Arduino NAIT
- 81 programmeur JTAG universel /
adaptateur de débogage
avec connexions pour divers microcontrôleurs





DÉCOUVRIR | CRÉER | PARTAGER

- 112 bienvenue dans la section PARTAGER**
avionique
- 113 bruits de labo...**
au labo d'Elektor, nous adorons les surprises
- 114 projet 2.0**
corrections, mises à jour et
courrier des lecteurs
- 116 Rétronique**
téléphone Telefunken pour voiture de 1965
« Schätzchen*, je serai en retard pour le dîner »
- 121 implantation des circuits intégrés**
- 122 NIDays**
coupe de robotique NXT 2016
- 124 Kodi : lecteur multimédia**
pour PC, Raspberry Pi, tablette et bien
d'autres
- 126 ScanaQuad, le plus petit analyseur logique**
décode de nombreux protocoles série

iBaxandall

variations sur le correcteur de tonalité

Vous devez construire en vitesse un correcteur de tonalité, trois possibilités s'offrent à vous : chercher sur l'internet un circuit classique, en trouver un qui vous séduit ou vouloir faire mieux et concevoir une nouvelle variante du circuit. Et pourquoi ne pas partir d'une page blanche pour créer du neuf, sortir des sentiers battus ? Alors, on s'amuse ?

62

bientôt sur ces pages

Le numéro de rentrée aura fait le plein de projets, cours, trucs et astuces... pour les électroniciens.

Extrait du sommaire :

- OBD avec Raspberry Pi
- Solutions de domotique
- Relais Bluetooth
- RPi s'attaque aux LEGO
- Alimentation
- Cours d'assembleur : SPI et I2C
- Filtrage sur Red Pitaya : filtres IIR
- Questions & Réponses : MQTT

Cette liste n'est pas exhaustive

Sous réserve de modifications.

Le numéro de septembre 2016 paraîtra le 17 août 2016.



- 86 sonde de courant pour oscilloscopes**
mesure à potentiel flottant de la chute de
tension aux bornes de shunts
- 89 indicateur de charge faible**
pour pile de 9 V
- 90 pondre un projet en C d'un seul clic**
configurateur scripté pour logiciel modulaire
sous Atmel Studio
- 98 accordeur de guitare (basse) de scène**
rapide et précis
- 105 MAXQ à la rescousse**
un multimètre sauvé du dépotoir

la communauté d'Elektor

DÉCOUVRIR

CRÉER

PARTAGER

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



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



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



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



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



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



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



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



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

connectez-vous à notre communauté

formule GREEN

92,50 € par an

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

connectez-vous

www.elektor.fr/formule-greencard

formule GOLD

127,50 € par an

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

connectez-vous

www.elektor.fr/formule-goldcard

formule GRATUITE

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

connectez-vous

www.elektor.fr/inscription

79

pays

247031

membres actifs

1034

experts & auteurs

485

publications

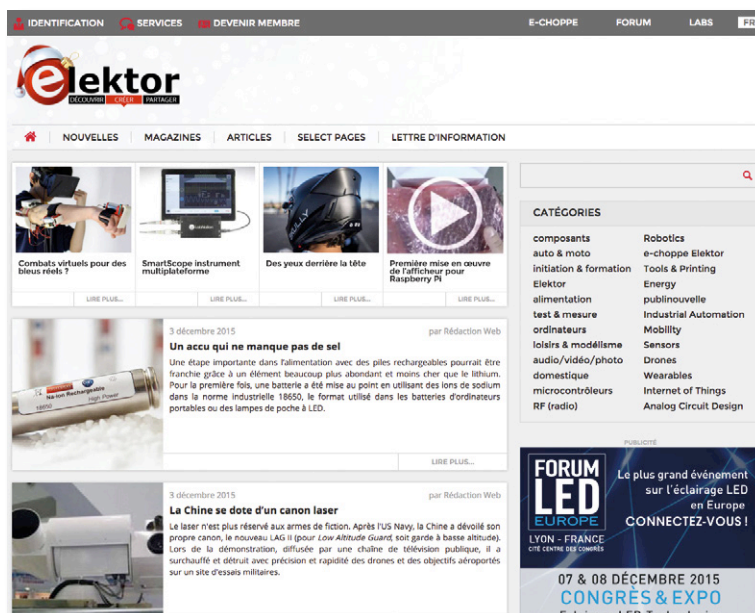
235102

visiteurs (par mois)

www.elektormagazine.fr

Explorez le monde de l'électronique

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



Elektor : gratuit & sans papiers

Restez informé, restez passionné !

Comme plus de 120 000 autres électroniciens, grâce à notre lettre d'information gratuite Elektor, vous êtes au courant des dernières tendances et des récents développements de l'électronique.

En outre, les abonnés à Elektor reçoivent régulièrement des offres et remises spéciales pour la boutique en ligne.

Pour vous abonner :
www.elektor.fr/inscription



bienvenue dans la section **DÉCOUVRIR**

Jens Nickel (Elektor)

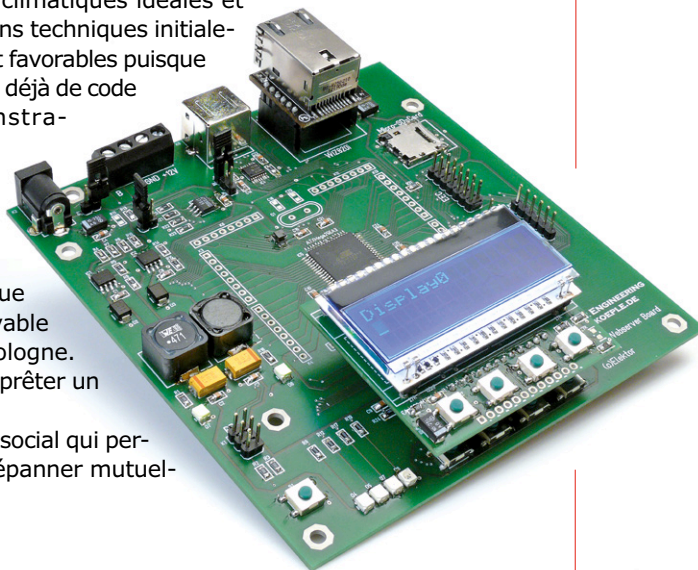
Un réseau de partage de fonds de tiroir

Je pensais revenir ici sur mon voyage dans l'internet des objets. Les nuages abondants du lundi de Pentecôte m'ont inspiré l'idée d'utiliser la célèbre carte de serveur web Xmega comme plateforme pilote pour connecter un circuit directement au réseau, sans passer par le relais d'un PC. En dépit de conditions climatiques idéales et de conditions techniques initialement plutôt favorables puisque je disposais déjà de code de démonstration et de

bibliothèques, notamment pour le module réseau WIZ820io, j'ai fait chou blanc.

Le routeur et ma table de travail se trouvent en effet aux deux coins opposés de mon logement, impossibles à relier avec le seul câble de 10 m que j'avais. Il en aurait fallu un de 15 m, hélas introuvable un lundi de Pentecôte dans ma bonne ville de Cologne. Aucun des quelques copains qui auraient pu m'en prêter un n'était chez lui ce jour-là.

Cela m'a donné l'idée de créer un nouveau réseau social qui permettrait, aussi et surtout les jours fériés, de se dépanner mutuellement entre bricoleurs passionnés. Chiche ?



C'est nouveau et dans le nuage

Dans l'internet des objets, le temps passe encore plus vite qu'ailleurs. Pour suivre, il faudrait rester en veille technologique nuit et jour. Témoin de cette accélération est le service en nuage créé sous son nom propre par Conrad. Le grand distributeur allemand invite les internautes à regrouper leurs données et leurs commandes *IoT* sur <https://conradconnect.de>, une belle initiative (hélas en allemand seulement pour l'instant – on vous avait pourtant dit que les langues vous seraient utiles dans la vie !). D'autre part, Jaime, qui a quitté Elektor l'an dernier pour rejoindre la jeune pousse berlinoise Relayr, me signale qu'il s'apprête – je cite – « à lancer une application avec laquelle ton téléphone se transforme en batterie de capteurs reliés à l'internet – je suis sûr que ça t'intéressera et tes lecteurs aussi. Le code source sera ouvert, tout sera sur Github... »

Pour sûr que ça m'intéresse ! Et j'y reviendrai bientôt dans mes nouvelles en ligne sur **www.elektormagazine.fr**



Arduino télécommandé

J'ai encore appris quelque chose : dans un article en préparation, l'auteur et spécialiste de logiciel Tam Hanna télécommande *Arduino* avec un ordiphone ou une tablette *Windows*, par USB ou sans fil par Bluetooth. Vous avez déjà entendu parler du protocole *Firmata* ? Moi pas. Pourtant même Microsoft propose sa propre bibliothèque .NET ! La suite bientôt dans ces pages. ◀

(160012 – version française : Denis Meyer)

descente d'antenne symétrique

drôle de composant n°25

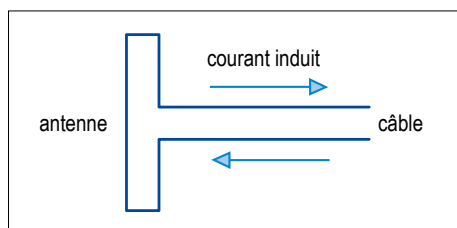


Figure 1. Le principe de fonctionnement d'une antenne en dipôle replié.

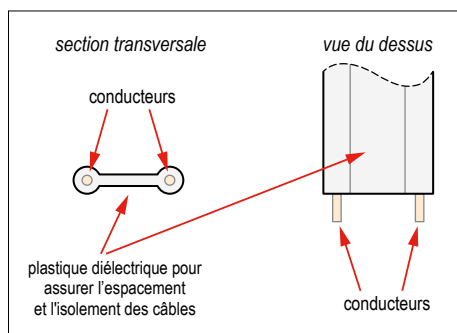


Figure 2. Câble pour antenne symétrique [2].

Neil Gruending (Canada)

Aux premiers temps de la télévision et de la radio FM, on utilisait couramment des antennes dipôles sur le toit, plutôt que le câble de télédistribution. Une antenne dipôle fournit un très faible courant en présence du champ électromagnétique d'un émetteur à RF et pour l'amener au récepteur, on a recours à une ligne de transmission symétrique (*twin-lead*) [1] à deux conducteurs parallèles, jumelés. Voyons ce que ce câble a de si particulier. Le défi, c'est que les signaux sont à très bas niveau et donc que le risque d'interférence est considérable. Heureusement, une antenne dipôle a une sortie symétrique, ce qui veut dire qu'elle fonctionne comme une boucle de courant au bout des deux fils, comme à la **figure 1**. Le signal utile est la différence entre les deux conducteurs, donc tant que les interférences l'influencent en mode commun, elles sont éliminées par le récepteur. Une autre caractéristique des lignes symétriques est que leur impédance doit être constante pour travailler correctement.

Tout défaut d'adaptation perturbe l'équilibre des courants et se manifeste par des interférences sur le récepteur.

Pour ce genre de transmission, nous utilisons généralement le câble coaxial, mais il ne convient pas dans ce cas-ci, parce que sa gaine tressée doit être mise à la terre pour servir de blindage contre les parasites et protéger le signal. Cette mise à la terre disloque l'antenne dipôle, cause un déséquilibre total de la symétrie en créant pour le courant un autre chemin à travers le blindage. Un autre défi pour le câble coaxial serait de l'adapter sur 300 Ω , l'impédance caractéristique du dipôle replié.

En revanche, le câble symétrique comme celui de la **figure 2** répond exactement à ces exigences par une transmission sur deux conducteurs. L'écartement entre eux détermine l'impédance de la ligne. Le câble est revêtu de plastique, qui assure aussi l'espacement constant et lui donne cette apparence qui nous rappelle l'époque de la profusion d'antennes TV sur les toits. Cette construction réduit fortement la capacité différentielle en comparaison de celle du coaxial, elle réduit ainsi les pertes, à fréquence et longueur de câble identiques.

En bas, côté récepteur, il est difficile de recueillir un signal de bonne qualité si le câble est torsadé ou s'il y a du métal à proximité, ce qui entraîne les interférences que nous avons évoquées. La solution consiste à utiliser un symétriseur à l'envers (*balun*) pour fournir une sortie asymétrique qui divise par quatre l'impédance en la faisant correspondre aux 75 Ω du câble coaxial ordinaire. Le récepteur en met à la terre le blindage et protège ainsi le signal des parasites. Maintenant que la télédiffusion terrestre est moins répandue, construire une antenne VHF pour la radio à modulation de fréquence est une voie royale pour expérimenter sur le câble symétrique. La **figure 3** montre une antenne dipôle [3] du commerce réalisée avec du câble symétrique à 300 Ω . Quant à la **figure 4**, elle représente une application inattendue

de ce câble comme injecteur en échelle (*ladderfeed*) fait de *twin* ajouré, sur un oscilloscope des années 1960 de la collection Rétronique de Jan. ■

(150704 – version française : Robert Grignard)

Liens

- [1] <https://en.wikipedia.org/wiki/Twin-lead>
- [2] www.radio-electronics.com/info/antennas/balanced_feeder/balanced_feeder.php
- [3] www.retrokabel.de



Figure 3. Une antenne intérieure de réception de la FM en VHF. Elle est faite de câble symétrique *twin* de 300 Ω .



Figure 4. L'oscilloscope Philips PM3110 utilise du câble d'antenne symétrique pour attaquer les plaques de déviation verticale du tube cathodique. Ces courts raccords conduisent du 10 MHz sous 360 V_{Y1-Y2} max. à capacité constante.



Envoyez-moi vos idées en quelques lignes ou sous forme d'un article déjà rédigé : neil@gruending.net

nouvelle vie pour une balise NFC (1)

accès aux entrailles

Patrick Gueulle (France)

Maintenant que le jeu-concours du numéro de janvier-février 2016 est derrière nous, il est temps de chercher de nouvelles applications à la balise NFC ST25TA02K qui vous a été offerte en tant que clef de participation ! Par exemple (mais ce ne sera sûrement qu'un point de départ...), changer l'URL désormais inutile qui y avait été préprogrammée. C'est une bonne occasion de se familiariser avec l'art et la manière de lire et écrire dans ces intéressantes cartes à puce sans contact que sont les balises NFC de « Type 4 ».



Du Type 1 au Type 4

Comme la communication en champ proche (NFC) n'est qu'une façon particulière (et prometteuse) d'associer intelligemment des technologies sans contact (RFID) préexistantes, pratiquement toute carte à puce sans contact pourrait théoriquement faire office de balise NFC. Mais afin de promouvoir une interopérabilité aussi large que possible (notamment avec les ordi-

phones), le NFC Forum a défini des « Types » standardisés auxquels il est vivement recommandé d'adhérer.

Cela n'a aucunement empêché certains fabricants de semi-conducteurs de faire cavalier seul, afin de « parachuter » leurs propres produits (voir notre compte-rendu du salon CARTES 2015, Elektor de mars 2016 [1]) en les numérotant parfois d'une façon fantaisiste.



Figure 1. Carte Topaz d'Innovation conforme aux spécifications « Type 1 » du NFC Forum.



Figure 2. Mifare Ultralight de NXP, facile à configurer en balise NFC de « Type 2 ».

L'une des premières balises NFC « natives » à avoir été commercialisée fut la puce Topaz d'Innovision (puis Broadcom), répondant strictement aux spécifications « Type 1 » du NFC Forum. Avec seulement 96 octets de mémoire utile, elle ne convenait cependant qu'à des applications très simples (**fig. 1**). Mais nous en sommes maintenant à 512...

De son côté, NXP a apporté sa puce Mifare Ultralight, facile à configurer en balise NFC Forum « Type 2 ». Mais avec une capacité de tout juste 64 octets, c'était encore plus étriqué (**fig. 2**) ! Aujourd'hui, il faut plutôt miser sur les familles Ultralight C et NTAG, mais reconnaissons à nos deux pionniers un avantage décisif : leur prix dérisoire !

Faisons l'impasse sur le « Type 3 » (Felica) qui ne se rencontre pratiquement qu'en Asie, pour nous intéresser au très populaire Mifare Classic (ou « Standard ») spécialement formaté en balise NFC (parfois surnommé « Type 7 »).

Malgré un espace mémoire bien plus vaste (déjà 1024 octets pour le modèle 1K), l'adopter pour un projet NFC « ouvert » n'est pas forcément une bonne idée (**fig. 3**)...

En effet, son protocole de communication est en partie propriétaire, et par conséquent incompatible avec des quantités d'ordiphones équipés de jeux de puces NFC concurrents de ceux de NXP !

Le « Type 4 » relève, lui, d'une approche complètement différente et, nous semble-t-il, infiniment plus séduisante. Au lieu de dépendre d'une hypothétique compatibilité « de bas niveau » avec un choix limité de plates-formes de lecture, il reconnaît de véritables commandes ISO7816-4 (T=CL, *contactless*) comme toute carte à microprocesseur bien élevée.



Figure 3. Balise de type Mifare Classic : 1024 octets de mémoire, mais compatibilité limitée.

Quelques rappels utiles

« ISO 14443 » regroupe une série de normes qui régissent la communication radio en champ proche à 13,56 MHz des cartes d'identification, cartes à circuit(s) intégré(s) sans contact et cartes de proximité :

Partie 1 : caractéristiques physiques

Partie 2 : interface radiofréquence et des signaux de communication

Partie 3 : initialisation et anticollision

Partie 4 : protocole de transmission

Le « A » de la référence ST25TA02K de notre balise par ex. indique qu'elle est conforme au protocole de type A de la norme 14443-4.

Comme nous l'avons déjà dit dans le texte, les fabricants s'écartent des normes. Voici quelques cartes dérivées de la norme ISO 14443 :

Mifare Classic : protocole propriétaire de Philips

(aujourd'hui NXP) qui respecte 14443-1,2,3 (type A), mais pas 14443-4 ; avec ou sans algorithme CRYPTO1.

Mifare Ultralight : variante de Mifare Classic, sans CRYPTO1.

LEGIC RF : norme propriétaire, mais avec des similitudes avec ISO 14443. Elles utilisent toutes les deux la fréquence de 13,56 MHz par exemple.

FeliCa : protocole propriétaire, développé par Sony.

Nous avons également parlé de commandes APDU. Cet acronyme signifie « **A**pplication **P**rotocol **D**ata **U**nit », c'est le message échangé entre une carte à puce et un lecteur de carte à puce. Il est normalisé et décrit dans la norme ISO 7816-4 (Organisation, sécurité et commandes pour les échanges).

Si nous reprenons la commande de sélection de l'application, elle se décompose ainsi :

C-APDU of the NDEF Tag Application Select Command

Name	CLA	INS	P1	P2	Lc	Data	Le
-	0x00	0xA4	0x04	0x00	0x07	0xD27600 00850101	0x00
Class byte							
Select instruction code							
P1 field							
P2 field							
Number of bytes of data							
Application ID							
Le field							

Tableau extrait de la documentation de STMicroelectronics.

La signification des champs est la suivante :

CLA : classe d'instruction, soit le type de la commande, par ex.

interindustry ou *proprietary*

INS : code de l'instruction (lecture, écriture...)

P1 et P2 : paramètres de l'instruction, par ex. position dans un fichier

Lc : nombre (Nc) d'octets envoyés par la commande

Data : données proprement dites

Le : nombre (Ne) maximum d'octets attendus dans la réponse, s'il y en a une



Figure 4. La carte MIFARE DESFire de NXP est une des premières balises de « Type 4 ».

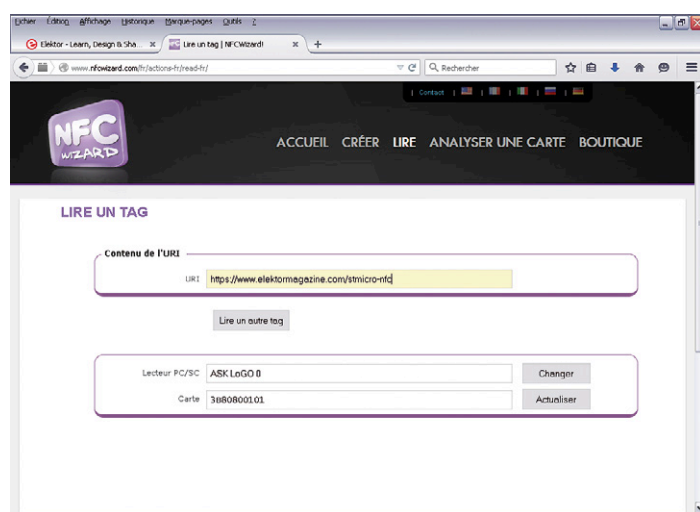


Figure 5. Lecture du contenu de la balise avec un lecteur PC/SC sans contact et un logiciel en ligne.

Il est vrai que les premières balises de « Type 4 » ont été développées autour de cartes DESFire ou JCOP, dont l'inconvénient majeur est toutefois le prix bien plus élevé (fig. 4). C'est pleinement justifié pour des balises à « contenu actif », embarquant un micrologiciel capable de faire subir des traitements complexes (voire cryptographiques) aux données qu'elles hébergent. Et cela permet même de doter une BasicCard sans contact (ZC7.5) d'authentiques fonctions NFC (voir notre code source expérimental dans Elektor de janvier-février 2015 [2]), bénéficiant d'ailleurs d'une énorme capacité mémoire. Tout en étant strictement compatible « Type 4 », la balise ST25TA02K de STMicroelectronics ne contient pas de microprocesseur et reste donc fort peu coûteuse (tout en étant dotée de 256 octets de mémoire EEPROM). Toutes proportions gardées, ce concept rappelle celui de la CryptoMemory d'Atmel, une très originale carte à puce « à contact » de type synchrone (sans microprocesseur et donc à logique câblée), mais acceptant les commandes (APDU) T=0 des cartes asynchrones. À l'origine, c'est une brillante idée de Gemplus (aujourd'hui Gemalto) qui l'avait d'ailleurs brevetée avant le lancement (sous nos yeux...) de sa carte « GemClub-Memo » lors du salon CARTES 1998.

Nouons le dialogue

Nous savons déjà que la balise ST25TA02K est supportée par des applications pour (certains) ordiphones, qui permettent de faire pas mal de choses... mais dans des limites infranchissables. Une autre approche, encore plus attractive, consiste à se servir d'un lecteur PC/SC de carte sans contact relié à un ordinateur. Peu important la marque et le modèle, pourvu qu'il soit installé avec le pilote qui convient (ou même avec un pilote générique s'il est également compatible CCID comme le très populaire ACR122).

Tout ce qu'il y a d'intéressant à faire passe par un fort petit nombre de commandes très courantes, au premier rang desquelles figure *Select*.

Les commandes sont détaillées dans la documentation d'une cinquantaine de pages (DM00179392.PDF) que STMicroelectronics met à notre disposition sur son site web [3].

Avant toute chose (autrement dit juste après la présentation de la balise devant le lecteur), il faut en effet sélectionner l'application dite « NDEF » (*NFC Data Exchange Format*) : 00A4040007D2760000850101.

Une variante (pas vraiment documentée) de cette commande (00A4040007D2760000850100) rendrait la balise temporairement compatible avec l'organisation de la mémoire (*mapping*) version 1.0 et non plus 2.0, la plus communément utilisée de nos jours.

Cela fait, plusieurs fichiers sont disponibles à la sélection :

- un « Capability Container » (CC) qui renseigne sur les caractéristiques de la balise,
- un fichier système spécifique à STMicroelectronics,
- et surtout le fichier (NDEF) qui contient la « charge utile » de la balise.

Brûlons un peu les étapes, et jetons directement un coup d'œil sur celle-ci, car c'est principalement là que nous allons pouvoir intervenir.

Bricolons l'URL !

La commande de sélection est 00A4000C020001 puisque l'identifiant de ce fichier est 0001h (ce serait plutôt E103h pour le CC et E101h pour le fichier système).

Deux commandes de lecture (*ReadBinary* = B0) sont reconnues par la balise, qui se distinguent l'une de l'autre par leur octet de « classe ISO » : 00h pour la lecture ordinaire, et A2h pour la lecture dite « étendue ». La différence est qu'une commande de lecture 00 B0 renverra un compte-rendu d'erreur (6282h) si l'on tente de lire au-delà de la fin du « message NDEF », tandis que A2 B0 permet de lire le fichier jusqu'au bout (dans la limite de 255 octets).

Fort heureusement, les deux premiers octets du fichier NDEF indiquent la longueur de son contenu, et on peut en prendre connaissance par la commande 00B0000002.

Dans le cas de notre balise-concours, le résultat est 00 24, ce qui indique qu'il y a 36 octets (24h) à lire derrière.

Lisons-les donc (avec un décalage ou « offset » de deux octets, soit 02h) au moyen d'une commande 00B0000224, ou alors lisons le fichier jusqu'à la fin avec la commande A2B00002FE pour visualiser l'espace mémoire inutilisé (et naturellement récupérable).

Dans un cas comme dans l'autre, on verra apparaître le message NDEF contenant l'URL ayant servi à participer au concours ([4]) codée de la façon suivante :

D1 01 20 55 02 65 6C 65 6B 74 6F 72 6D 61 67 61 7A 69
6E 65 2E 63 6F 6D 2F 73 74 6D 69 63 72 6F 2D 6E 66 63

À partir du sixième octet (de 65h jusqu'à 63h), on retrouve le texte *elektormagazine.com/stmicro-nfc* exprimé en ASCII. Juste avant, l'octet 02h est une abréviation normalisée de *https://www*. 01h signifierait par contre *http://www*. Et il existe bien d'autres variantes, dont certaines potentiellement fort utiles ! L'octet 55h indique pour sa part (lettre U majuscule) que ce qui le suit est une URL (ou plus exactement une URI), tandis que 20h précise sa longueur, abréviation comprise (à vrai dire, 4 de moins que l'octet de longueur 24h lu précédemment, soit ici 32 octets).

Il n'y a pas besoin d'en savoir davantage pour modifier « proprement » l'URL, sachant que toute anomalie entraînerait le rejet de la balise lors de son utilisation avec un ordiphone.

On peut même se livrer à une petite expérience : la transformer en *http://www.elektormagazine.com* (toujours d'actualité, elle !) par bricolage judicieux de... trois octets seulement. Deux commandes *UpdateBinary* (00 D6) suffisent : la première (00D600060101) change l'abréviation *https* en *http*, tandis que la seconde (00D60000050018D10114) vient trafiquer les deux octets de longueur.

La fin du fichier se trouvant ainsi décalée juste après .com, le reste de l'URL (/stmicro-nfc) ne sera plus pris en compte. Cependant, il restera lisible par une commande *ReadBinary* « étendue » (A2B00002FE), portant sur la totalité de l'espace mémoire. C'est tout à fait analogue à la façon dont des SMS effacés dans une carte SIM peuvent être récupérés par les experts de la Police Scientifique !

Et il serait même possible de faire carrément machine arrière en enchaînant les deux commandes 00D600060102 et 00D60000050024D10120 !

Jusqu'à récemment, on pouvait vérifier le succès de ces opérations en lisant la balise, en ligne, sur *http://www.nfcwizard.com/fr/actions-fr/read-fr/* à l'aide de n'importe quel lecteur PC/SC sans contact (fig. 5). Le site est maintenant fermé parce que les applets JAVA regorgent de plus en plus de restrictions qui empêchent les navigateurs Internet de dialoguer avec les lecteurs de cartes. Vous pouvez vous tourner vers *gototags.com/products/windows-app*. Quand on vous le dit, qu'on peut se passer d'Android ! En attendant mieux dans un second épisode...

(150805-I)

Liens

- [1] Article « salon CARTES 2015 : le dernier du nom ? », Elektor janvier-février 2016 : www.elektormagazine.fr/150640
- [2] Article « salon CARTES SECURE CONNEXIONS 2014 », Elektor janvier-février 2015 : www.elektormagazine.fr/140428
- [3] Documentation des commandes, STMicroelectronics : www.st.com/web/en/resource/technical/document/datasheet/DM00179392.pdf
- [4] Concours NFC STMicroelectronics : www.elektormagazine.com/stmicro-nfc
- [5] Exécutable du programme en ZCBasic : www.elektormagazine.fr/150805

mouser.fr

Les dernières nouveautés pour
vos conceptions les plus récentes™

La plus vaste sélection de produits les plus récents.

Plus de **4 millions** de produits
de plus de **600 fabricants**.



Distributeur agréé de semi-conducteurs
et de composants électroniques

cours intensif d'assembleur 2.1 (2) adressage indirect

Dans l'article précédent de notre cours intensif d'assembleur [1], nous avons traité de la commande multiplexée d'un afficheur à LED. L'utilisation d'un microcontrôleur (µC) récent aux modes d'adressage sophistiqués a permis de simplifier sensiblement la programmation. Dans ce second article, vous découvrirez l'adressage indirect.

Miroslav Cina (miroslav.cina@t-online.de)

La famille plus moderne des µC PIC1xF18xx met à disposition une fonction très pratique : l'adressage indirect. Son principal avantage est que l'on peut accéder à la mémoire de façon totalement linéaire, ce qui permet de contourner le traitement parfois complexe inhérent aux blocs segmentés. Ainsi, ces µC redeviennent faciles à manipuler, à l'image de ce que permettaient les « vrais » microprocesseurs à 8 bits tel le populaire Z80 (Zilog) des années 80.

Registres INDFx

Dans l'article précédent, nous avons vu l'adressage direct via le registre *BSR*. Tous les µC de la série PIC1xF18xx comportent

deux autres registres spéciaux qui permettent un accès linéaire à la mémoire de travail : *INDF0* et *INDF1*. Mieux, ils autorisent même un accès linéaire à la mémoire de programme. Avec cette méthode d'accès, toute la mémoire, RAM + Flash (mais hors EEPROM), du µC se présente sous la forme d'un seul bloc de 64 Ko.

Qui dit espace d'adressage de 64 Ko, dit adresses de 16 bits. C'est ce type d'adresses que contiennent les paires de registres **FSR0** (*FSR0H* | *FSR0L*) et **FSR1** (*FSR1H* | *FSR1L*), dont nous avons déjà parlé comme membre des registres noyaux (*Core*). Il nous faut maintenant, pour l'adressage indirect, deux instructions additionnelles de l'assembleur. Nous verrons alors à quoi peuvent servir les registres **INDF0** et **INDF1**.

La RAM se retrouve en double dans l'espace d'adressage de 64 Ko. En premier lieu, dans la plage 0000h à 0FFFh, comme mémoire de travail conventionnelle. L'accès à ces adresses par le biais de **FSRx** correspond en grande partie à l'adressage direct via **BSR**. La RAM est à nouveau disponible dans la plage d'adresses 2000h à 29AFh – mais en linéaire cette fois. Ceci est pertinent si l'on veut, par exemple, accéder élégamment à une matrice de données, un tableau par exemple. Mais pourquoi donc la plage d'adresses s'arrête-t-elle à 29AF et pas à 2FFF ? Dans cet espace de mémoire, les blocs sont, à raison de 80 octets, un miroir des banques de mémoire individuelles. La banque *BANK 0* sise dans la plage d'adresses 020h à 06Fh se trouve ici dans la plage 2000h à 204Fh, *BANK 1* entre 2050h et 20AFh, etc. La dernière banque en miroir (*BANK 30*) se termine donc à l'adresse 29AFh.

L'accès à la mémoire de programme Flash se fait par les adresses 8000h à FFFFh. Il est évident qu'il n'y a pas occupation effective de tous les emplacements de mémoire. Par exemple, la Flash de 8 Kmots du PIC16F1829 se trouve dans la plage 8000h à 9FFFh. Les 2 Kmots du PIC12F1822 se situent eux dans la plage 8000h à 87FFh. Le reste inutilisé est bien adressable, mais inutile.

Caractéristique typique de ces µC PIC, des cellules de RAM de 8 bits de large, bien que la mémoire de programme soit

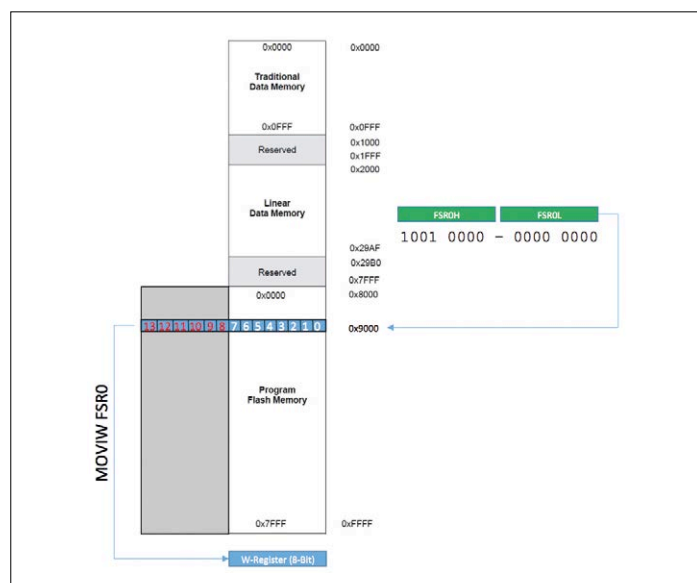


Figure 1. Cartographie de la mémoire des µC de la série PIC12F18xx.

organisée en mots de 14 bits. La question se pose : comment ces mots de 14 bits ont-ils été compactés dans l'espace d'adressage. La solution est simple : on ne peut accéder directement à l'octet qu'avec les huit LSB (bits de poids faible). Avant que nous passions à la description de l'instruction **MOVIW**, examinons l'organisation de la mémoire à la **figure 1**. On y voit que la mémoire de programme a une largeur de 14 bits ; on n'a pas accès aux six bits de poids fort, c'est pourquoi cette zone a été grisée. La commande **MOVIW** recopie le contenu de la cellule de mémoire dont l'adresse se trouve dans la paire de registres **FSR0** ou **FSR1** dans le registre **W**. Seuls les huit bits de poids faible sont alors copiés.

MOVIW

Cette instruction est principalement destinée au transfert du contenu du registre **INDFn** dans le registre **W** (**MOVE IndFn to W**). **INDFn** peut être soit **INDF0** soit **INDF1**. Maintenant, l'astuce : le registre **INDFn** n'existe pas. Lors de l'accès à **INDF0**, on accède toujours à l'emplacement de mémoire dont l'adresse se trouve dans **FSR0** (pour **INDF1** ce sera évidemment l'adresse de **FSR1**).

Il existe deux variantes d'utilisation : pour l'une, on a ajustement du pointeur (incrémenté ou décrémenté) et pour l'autre pas d'ajustement. Il s'agit ici de la première variante, car souvent utilisée lors de l'accès à des champs de données.

La première variante connaît en outre quatre sous-types cf. le **tableau 1**, qui se différencie par le moment de changement du pointeur, avant ou après l'accès, et la fonction, incrémentation ou décrémentation ; « n » peut en outre avoir la valeur « 0 » ou « 1 » (registre **FSR0** ou **FSR1**).

Parallèlement à l'instruction **MOVIW** décrite ici, il existe, avec **MOVWI**, son inverse : cette instruction écrit le contenu du registre **W** dans la cellule de mémoire adressée correspondante. Ceci ne fonctionne bien entendu que dans la zone où il existe de la RAM. Il faudra, pour modifier le contenu de la mémoire Flash, opter pour une autre technique. Tout cela est à utiliser avec prudence : du code qui peut se modifier lui-même présente des pièges.

Affichage de la mémoire de programme

Vu que l'on dispose déjà, avec l'afficheur à LED de l'article précédent, du matériel requis pour afficher des valeurs, il nous servira à afficher en boucle le contenu de la mémoire Flash à partir de l'adresse 1000h. Il suffit de modifier le programme principal du logiciel du µC. Hormis cela, on réutilise le pilote d'affichage de l'article précédent légèrement modifié. Il n'y a pas cette fois de compteur dans la routine principale, mais le code qui sert à lire les données de la mémoire de programme. Le principe : une boucle lit les données de la mémoire de programme et les affiche. Pour cela, le programme extrait les données octet par octet à partir de l'adresse 1000h. Chaque octet lu est affiché un court instant, on passe ensuite au sui-

Tableau 1.

syntaxe	signification
<code>moviw ++FSRn</code>	pré-incrémentation incrémentation avant accès en lecture
<code>moviw --FSRn</code>	pré-décrémentation décrémentation avant accès en lecture
<code>moviw FSRn++</code>	post-incrémentation incrémentation après accès en lecture
<code>moviw FSRn--</code>	post-décrémentation décrémentation après accès en lecture



Figure 2. Affichage d'un octet sous forme hexadécimale sur l'afficheur à LED du premier article.

vant. La lecture de la valeur de FFh de la mémoire signale la fin de la boucle. Tout repart alors à l'adresse 1000h.

Comme il s'agit d'octets, la représentation requiert des caractères numériques ordinaires, mais aussi hexadécimaux. L'affichage d'un octet de format hexadécimal ne nécessite que deux positions d'afficheur. Nous utilisons pour cela les deux positions gauches de l'affichage, LD0 et LD1. LD2 reste vide et la quatrième position, LD3, affiche un « h » en permanence. La valeur décimale 122 s'affiche alors comme « 7A h », cf. la **figure 2**.

Modification du pilote d'affichage

Nous le disions, l'affichage doit maintenant pouvoir représenter des caractères hexadécimaux. Contrairement au code du 1^{er} article, il ne suffit plus de pouvoir afficher uniquement les chiffres « 0 » à « 9 » et le signe « - », il nous faut maintenant les lettres « A » à « F », l'espace et le « h » minuscule, d'où modification du pilote d'affichage. Il devient possible alors de représenter en hexadécimal toutes les valeurs décimales 0 à 255. Le générateur de police de la routine « `wm_char_set` » va gonfler un peu. Les différences entre l'ancienne version décimale du **listage 1** et la nouvelle version hexadécimale du **listage 2** restent limitées. Cet exemple montre combien il est facile, si besoin était, d'étendre le pilote d'affichage. Un caractère est défini dans une ligne d'assembleur. Ainsi les modifications nécessaires pour l'affichage sont simples et rapides.

Programme principal

Passons au programme principal ; il est composé de trois parties :

- Lecture des données de la mémoire Flash par le sous-programme `f_get_data`.

Listage 1.

```

;-----
; ancien jeu de caractères décimaux
;-----
wm_char_set brw

```

```

retlw B'00111111' ;00h -> 0
retlw B'00000110' ;01h -> 1
retlw B'01011011' ;02h -> 2
retlw B'01001111' ;03h -> 3
retlw B'01100110' ;04h -> 4
retlw B'01101101' ;05h -> 5

```

```

retlw B'01111101' ;06h -> 6
retlw B'00100111' ;07h -> 7
retlw B'01111111' ;08h -> 8
retlw B'01101111' ;09h -> 9
retlw B'01000000' ;0Ah -> moins

```

- Conversion des octets au format hexadécimal à deux chiffres avec le sous-programme *calc_hex_disp*.
- Structure de la boucle principale, sans fin.

Lecture de la Flash

Dans cette partie, nous découvrons l'utilisation de l'adressage indirect. La déclaration complète des constantes comme données pour la mémoire Flash et l'accès en lecture correspondant sont faciles à implémenter, cf. le **listage 3**.

Les données à lire doivent être stockées dans la mémoire de programme à partir de l'adresse 1000h. Ceci est obtenu par la directive **ORG 1000** (cf. *step 3*). En outre, le début des données est indiqué sous la forme de « *c_data1* ». En principe, l'utilisation d'autres zones de mémoire est possible, mais avec deux restrictions : la zone de mémoire doit, pour le µC utilisé, correspondre à de la mémoire Flash physique et la déclaration ne doit pas entrer en conflit « accidentel » avec le code de programme proprement dit. Le µC PIC16F1829 utilisé ici possède 8 K mots de mémoire de programme ; la cellule de la mémoire la plus haute se situe donc à l'adresse 1FFFh. Si l'on utilisait à tort l'adresse 2000h, l'accès se ferait à l'adresse 0000h, ce qui n'est pas très malin.

Avec la directive ORG, si vous déclarez la zone de mémoire des données à une adresse trop basse, l'exécution de l'assembleur produira des messages d'erreur. Cela dépendra de l'endroit où l'on « débarque » dans la mémoire de programme. Pour chaque octet écrasé par erreur, deux messages d'erreur sont affichés, cf. l'exemple ci-après :

```
Error[118]  C:\...\05_TEST.ASM 339 :
            Overwriting previous address contents (0051)
Error[118]  C:\...\05_TEST.ASM 339 :
            Overwriting previous address contents (0051)
```

Pourquoi deux messages par octet ? Parce que les mots de

mémoire de programme peuvent contenir jusqu'à 14 bits. Ainsi, une déclaration *H'AB'* est identique à *H'00AB'*. De ce fait, une telle déclaration écrase donc deux octets – d'où deux messages. La première partie du mot ne peut prendre que des valeurs allant de 00h à 3Fh, mais comme, avec cette méthode, il est de toute façon impossible d'accéder aux six premiers bits, on peut ici faire une déclaration abrégée.

On aurait pu totalement oublier la directive ORG 1000, les données auraient cependant été stockées dans la mémoire Flash juste après le sous-programme *f_get_data*. Il peut être important, à l'occasion, de connaître l'emplacement de mémoire exact, car en l'absence de directive ORG la position change avec le code.

La routine *f_get_data* (*step 2*) lit un octet de la mémoire Flash ; elle ne requiert que quatre lignes de code. Comme déjà dit, l'instruction *MOVIW* copie la valeur dans le registre *W*. Ensuite la valeur est sauvegardée dans la variable-tampon *v_value* pour une utilisation ultérieure ; et voilà. Comme ici on utilise la variante *post-incrémentation* de l'instruction, le pointeur correspondant est **incrémenté après** lecture. Lors du prochain appel du sous-programme, il y a automatiquement lecture de l'octet suivant.

Un mot au sujet de l'initialisation du pointeur – dans ce cas, la paire de registres *FSR1* : pour lire les données à partir de l'endroit correct, il faut qu'il s'y trouve l'adresse correspondante. C'est la fonction du sous-programme *f_get_add* (*step 1*). L'adresse est divisée par les directives « *LOW* » et « *HIGH* » et écrite dans les moitiés de registre appropriées. La ligne

```
movlw    LOW c_data1
```

commence par copier l'octet de poids faible de l'adresse *c_data1* vers le registre *W* – ici, la valeur 00h (de 1000h).

Ensuite, la ligne

Listage 2.

```
;-----
; nouveau jeu
; de caractères hexadécimaux
;-----
wm_char_set brw
    retlw B'00111111' ;00h -> 0
    retlw B'00000110' ;01h -> 1
```

```
retlw B'01011011' ;02h -> 2
retlw B'01001111' ;03h -> 3
retlw B'01100110' ;04h -> 4
retlw B'01101101' ;05h -> 5
retlw B'01111101' ;06h -> 6
retlw B'00100111' ;07h -> 7
retlw B'01111111' ;08h -> 8
retlw B'01101111' ;09h -> 9
retlw B'01110111' ;0Ah -> A
```

```
retlw B'01111100' ;0Bh -> b
retlw B'00111001' ;0Ch -> C
retlw B'01011110' ;0Dh -> d
retlw B'01111001' ;0Eh -> E
retlw B'01110001' ;0Fh -> F
retlw B'01000000' ;10h -> moins
retlw B'00000000' ;11h -> espace
retlw B'01110100' ;12h -> h
```

Listage 3.

```
;-----
; déclaration et lecture des données
;-----
;step 1
f_get_add    clrf    BSR
    movlw    LOW c_data1
    movwf    FSR1L
    movlw    HIGH c_data1
    movwf    FSR1H
    return
```

```
;
;step 2
f_get_data    clrf    BSR
    moviw    FSR1++
    movwf    v_value
    return
;
;step 3
    ORG 1000
c_data1    da    H'01', H'02', H'03', H'04'
            da    H'AB', H'CD', H'EF'
            da    H'FF'
```



```
movlw HIGH c_data1
```

copie l'octet de poids fort 10h (de 1000h) dans le registre *W*. En fait, ce n'est pas tout à fait exact ; en raison de l'adressage indirect, la mémoire Flash ne se trouve maintenant qu'à partir de l'adresse 8000h. L'emplacement 0000h se trouve lui dans l'espace d'adressage *FSR* à l'adresse 8000h. *c_data1* pointe donc sur l'adresse 9000h. C'est pourquoi pour la ligne de code précédente, c'est la valeur 90h qui atterrit dans le registre *W* et non pas 10h.

1 octet dans 2 quartets

Cette routine n'a rien à voir avec l'adressage indirect, mais est nécessaire si l'on veut avoir sortie d'un octet en hexadécimal. Pour mieux comprendre, nous allons décrire une nouvelle instruction : **LSRF** (**L**ogical **S**hift **R**ight). Elle appartient à la catégorie « *Byte-Oriented File Register Operations* » et se contente de décaler tous les bits du registre vers la droite. La syntaxe est :

```
lsrf f
```

où « *f* » identifie le registre traité. Le bit tout à droite, de poids le plus faible, finit dans les limbes ; il est ici représenté par le bit de retenue (*Carry Bit*) du registre d'état, puisque supprimé. Tout à gauche c'est un « 0 » binaire qui entre dans le registre comme bit de poids fort (MSB). La **figure 3** montre ce qui se passe. Si, initialement, on a la valeur 7Ah dans le registre *F*, elle devient, par l'application de *LSRF*, 3Dh (cf. **fig. 4**). Notons qu'il existe une instruction produisant l'effet inverse, **LSLF**. Le **listage 4** donne le code de division d'un octet en deux quartets de poids fort et faible (4 bits = 1 quartet = 1/2 octet). L'octet lu commence par être copié dans le registre *W* (*step 1*) ; l'instruction *AND* avec 00001111b met à 0 les quatre bits de poids forts (MSB). Le résultat est écrit dans une variable *v_lsb*. L'octet lu est à nouveau copié dans le registre *W* (*step 2*) ; une quadruple exécution de l'instruction de décalage *LSRF* permet de supprimer les quatre bits de poids faible (LSB). Les quatre bits LSB qui en découlent contiennent alors la valeur des quatre bits MSB précédents (variable *v_msb*). La **figure 5** traduit le comportement de ce morceau de code. Les deux variables *v_lsb* et *v_msb* contiennent alors les valeurs utilisables telles quelles pour l'affichage des octets sous forme hexadécimale.

Perspectives

La lecture d'un octet et la préparation de son affichage hexadécimal terminent ce second article de la deuxième mouture du cours intensif d'assembleur. Le code complet est disponible au téléchargement sur la page Web de cet article [2]. Dans le prochain article, nous aborderons la communication série via I²C ou SPI, qui prend une importance particulière dans le cas des µC et de leurs puces périphériques. Ensuite, nous verrons comment, par le biais d'une telle interface série, commander très simplement un expenseur de port. En outre, nous utiliserons l'affichage à LED pour réaliser une horloge ou un multimètre U/I, applications qui nécessitent chacune une puce externe et un rien de micrologiciel.

Pour toute question concernant le code et cet exemple, voici mon courriel : miroslav.cina@t-online.de. ◀

(160018 – version française : Guy Raedersdorf)



Figure 3. Les opérations lors du décalage de bits par l'instruction assembleur LSRF.

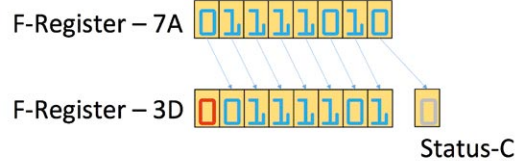


Figure 4. Décalage vers la droite des bits (*Bit-Shifting*) avec l'instruction LSRF. Le bit de poids faible précédent fini dans le registre d'état et à gauche on ajoute un « 0 » comme bit de poids fort.

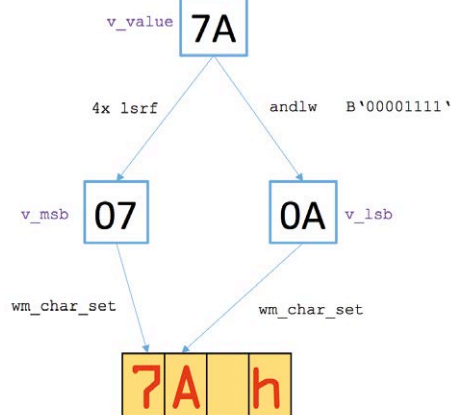


Figure 5. Division d'un octet en deux quartets, ensuite affichés au format hexadécimal.

Liens

- [1] *Cours intensif d'assembleur 2.1 (1)*, Elektor 06/2016 : www.elektormagazine.fr/150518
- [2] Page de cet article : www.elektormagazine.fr/160018

Listage 4.

```

;-----
; octet -> 2 quartets
;-----
;step 1
calc_hex_disp
movf v_value,0
andlw B'00001111'
movwf v_lsb

;step 2
movf v_value,0
movwf v_msb
lsrf v_msb,1
lsrf v_msb,1
lsrf v_msb,1
lsrf v_msb,1
return
```

CIRCUITMAKER

trucs & astuces (2)

Neil Gruending (Canada)

La dernière fois je vous ai présenté CircuitMaker et quelques-unes de ses spécificités. Nous verrons ici comment créer un projet et ajouter des composants à un schéma.

Création d'un projet

Un projet CircuitMaker est un dépôt dans lequel sont regroupés tous les fichiers nécessaires à la création d'un circuit. Ce

dépôt est à la fois stocké dans la cave communautaire (*Community Vault*) et dans un cache de votre ordinateur. Ce double stockage permet de partager un projet entre membres d'une même équipe.

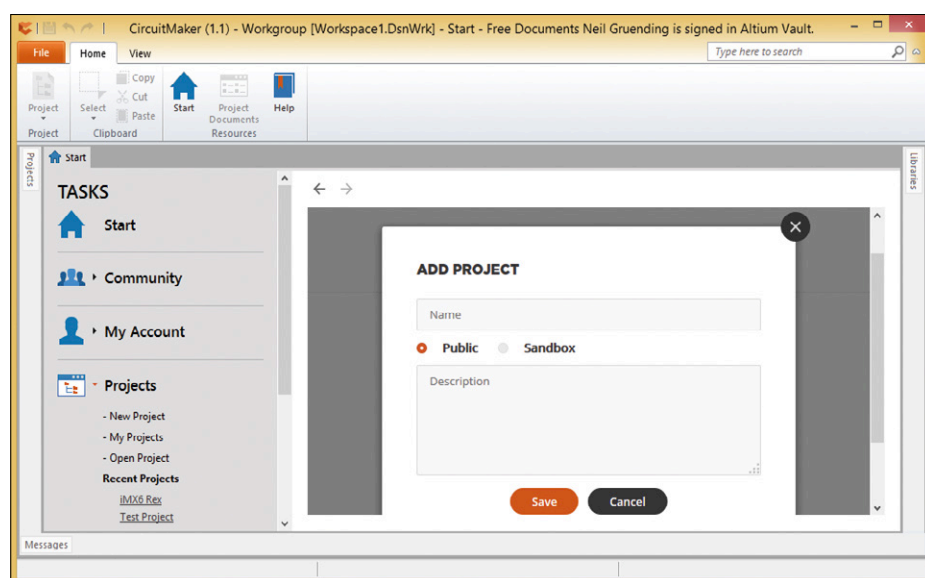


Figure 1. Ajout d'un projet.

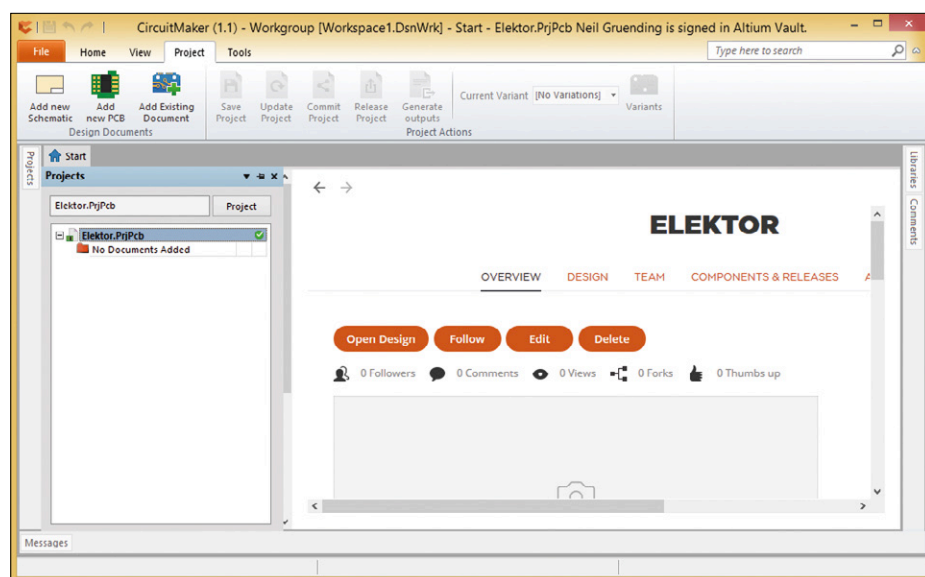


Figure 2. Ouverture d'un projet.

Pour créer un projet, connectez-vous d'abord à votre compte CircuitMaker. Vous accéderez à la page reproduite sur la **figure 1**. L'onglet *Start* permet de démarrer certaines tâches. Celle qui nous intéresse ici est la tâche *Projects*. Lorsqu'on clique sur *New Project*, la fenêtre *Add Project* s'ouvre et demande un nom et une description du projet créé. On peut aussi choisir s'il s'agira d'un projet public ou privé. S'il est privé, le projet est stocké dans une zone privée du système Vault appelée *Sandbox* (bac à sable). La décision n'est pas irréversible, un projet privé peut ensuite devenir public.

Le menu *My Projects* liste tous vos projets et leurs contenus. Le menu *Open Project* affiche et permet d'ouvrir tous les projets publics de la cave. Les projets que vous avez récemment ouverts sont listés sous *Recent projects*. Là aussi on peut cliquer sur l'un d'eux pour y accéder directement. Lorsque vous ouvrez un projet, une fenêtre semblable à celle de la **figure 2** s'ouvre (Elektor est le nom du projet que j'ai créé pour cet article).

Création d'un schéma

Nous pouvons maintenant ajouter un schéma à notre projet en cliquant sur le bouton *Add new Schematic*. CircuitMaker nous demande un nom de fichier, puis ajoute un dessin vierge au projet. Notez que, contrairement à d'autres programmes, CircuitMaker stocke chaque page de schéma dans un fichier différent. Pour faciliter la navigation dans un projet, j'associe donc aux noms de ces fichiers la fonction des schémas correspondants

(alimentation, contrôleur, etc.) Vous pouvez ajouter autant de pages que vous le souhaitez, CircuitMaker en garde la trace dans le fichier du projet.

Vous passerez le plus clair de votre temps dans l'onglet *Home* du ruban (**fig. 3**) car la plupart des outils utiles à la création d'un schéma y sont regroupés. Le groupe *Circuit Elements* contient les outils de connexion électrique, et le groupe *Graphical Elements* les éléments graphiques qui n'affectent pas les connexions. Les outils du groupe *Collaboration Elements* permettent quant à eux d'ajouter des commentaires à un schéma lorsqu'on travaille en équipe.

Ajoutons donc un composant à notre schéma vierge en cliquant sur l'icône *Component* du groupe *Circuit Elements*. La fenêtre *Place Component* qui s'ouvre contient le dernier composant utilisé. Cliquez sur le bouton *Choose* pour ouvrir la fenêtre *Browse Libraries* (**fig. 4**). Vous pouvez y sélectionner une bibliothèque et entrer le nom du composant cherché dans la zone *Mask*. Ici je voulais un transistor PNP MMBT3906. Lorsqu'on clique sur un des modèles proposés, CircuitMaker affiche son symbole et son modèle 3D (dans la fenêtre *Footprints*). Pour l'insérer dans le schéma, cliquez simplement sur le bouton *OK*.

C'est rapide, mais comment faire si le composant souhaité n'est pas listé ou n'est pas exactement le modèle recherché ? L'outil *Component* ne permet pas de modifier un composant, mais dans ce cas on peut appeler la fenêtre *Librairies* (**fig. 5**) en cliquant sur le bouton *Librairies* placé à l'extrémité droite de l'écran. Au bas de cette fenêtre est proposée l'option *Build your own*. En cliquant dessus vous accéderez à l'éditeur de composant et pourrez, une fois votre composant construit, l'enregistrer dans le système Vault et le rendre public.

On peut également modifier un composant existant, soit en cliquant sur le bouton *Build*, soit en choisissant *Edit* dans le menu qui apparaît après un clic droit sur le composant. Dans le premier mode d'édition, seules sont chargées les informations de base du composant (pas de symbole, pas de modèle 3D, etc.) L'autre mode permet de changer toutes les données liées au composant. Quel que soit le mode choisi, CircuitMaker crée une nouvelle version du composant lorsque vous l'enregistrez à nouveau dans Vault. Aucune information précédemment enregistrée n'est donc perdue, et on peut choisir entre l'une ou l'autre des versions stockées.

Une fois placés les composants sur un schéma, nous disposons des outils *Wire* et *Bus* pour les connecter. CircuitMaker offre des raccourcis pour la plupart des fonctions. Vous les trouverez sous *View* → *Help group* → *Shortcuts*. Et nous voici déjà prêts à passer à l'étape du routage, ce que nous ferons la prochaine fois. ◀

(150813 – version française : Hervé Moreau)

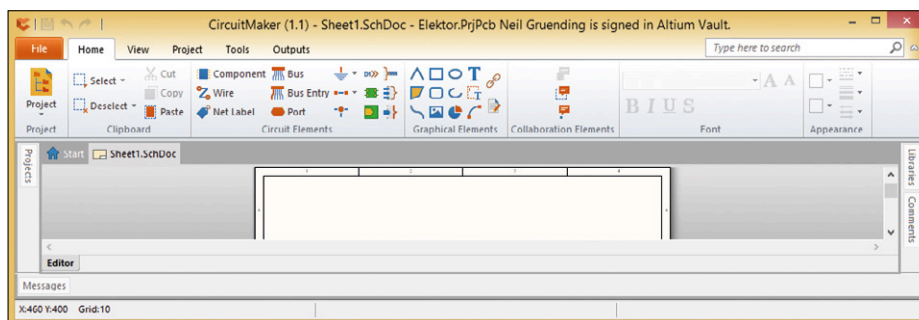


Figure 3. Les groupes de l'onglet *Home*.

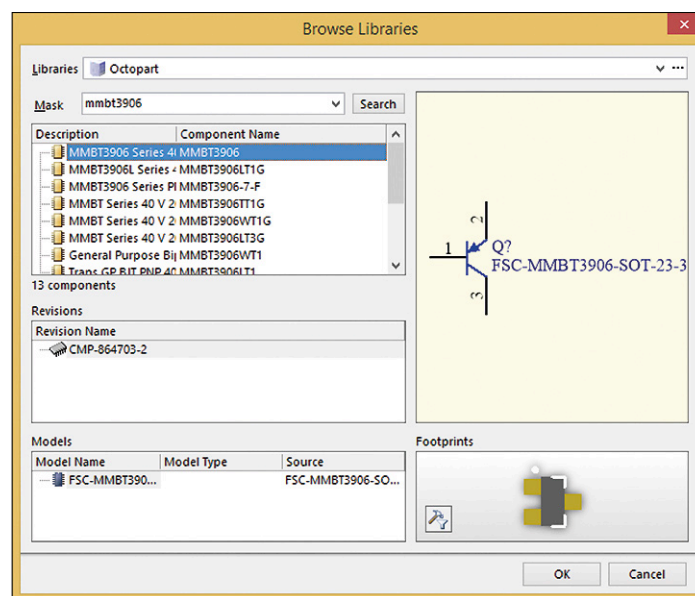


Figure 4. La fenêtre *Browse Libraries*.

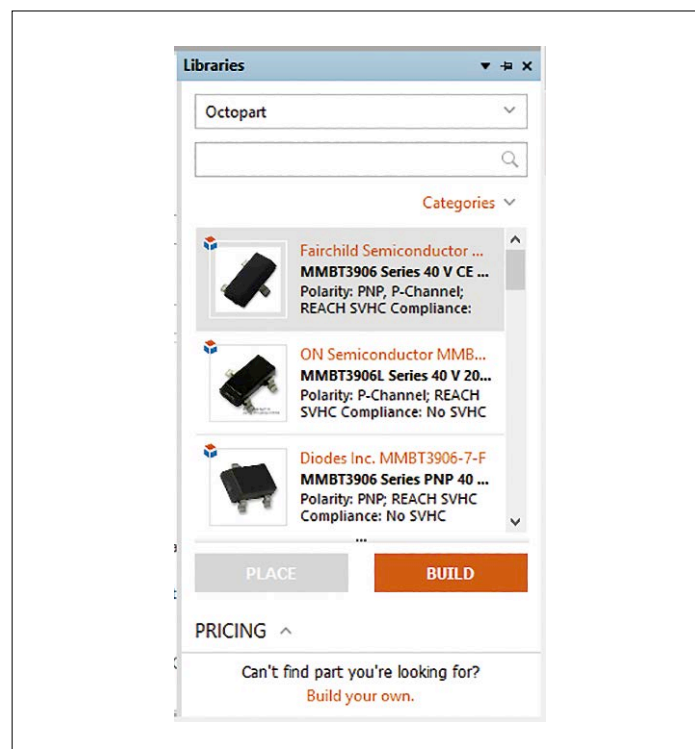


Figure 5. La fenêtre *Libraries*.

hors circuits

L'ABC des CA/N

DNL, INL, SNR, SINAD, ENOB, SFDR et consorts

Robert Lacoste (Chaville)

Ce mois-ci, je vous propose une petite balade dans le monde des convertisseurs analogiques-numériques (CA/N, ou *ADC* pour les anglophiles), et en particulier les modèles les plus performants. Comme vous le savez sûrement, un CA/N permet de transformer une tension analogique en sa représentation numérique, comme le fait un multimètre. Les CA/N ne sont pas une invention récente. Le premier, baptisé DATRAC, a été développé par Bernard Gordon pour la société EPSCO en 1954. Évidemment il fonctionnait avec des tubes à vide et il pesait 70 kg, mais ses caractéristiques étaient impressionnantes pour l'époque : résolution de 11 bits, conversion en 20 μ s, et une consommation d'à peine 500 W (!), le tout pour la modique somme de 8500 \$ de 1954, soit environ 100 000 € aujourd'hui.

On trouve maintenant des CA/N qui présentent des caractéristiques incroyables pour une poignée d'euros : des résolutions jusqu'à 32 bits, des fréquences d'acquisition en centaines de MHz, voire plus, etc. Avec un microcontrôleur, vous disposez de CA/N quasi gratuits. Est-ce à dire qu'il est inutile de se creuser la tête pour les mettre en œuvre efficacement ? Bien évidem-

ment non, et ce pour deux raisons. Tout d'abord parce que s'il y a des milliers de CA/N différents, c'est justement parce tous ne sont pas adaptés à votre application. Ensuite parce qu'un CA/N nécessite un certain nombre de circuits auxiliaires (**figure 1**), qui doivent être judicieusement conçus si l'on veut de bonnes performances.

Mémo des acronymes

- MSPS, **M**ega**S**amples **P**er **S**econd : méga-échantillons par seconde
- FSR, **F**ull-**S**cale **R**ange : plage de la pleine échelle
- DNL, **D**ifferential **N**on**L**inearity : non-linéarité différentielle
- INL, **I**ntegral **N**on**L**inearity : non-linéarité intégrale
- SNR, **S**ignal-to-**N**oise **R**atio : rapport signal sur bruit
- SINAD, **S**ignal-to-**n**oise and **d**istortion ratio : rapport signal sur bruit avec distorsion
- ENOB, **E**ffective **N**umber of **B**its : nombre de bits effectif
- SFDR, **S**purious-**F**ree **D**ynamic **R**ange : plage dynamique sans parasites

Fiche de caractéristiques d'un CA/N...

Commençons par le choix du CA/N lui-même. Bien sûr vous pouvez commencer par fureter dans les catalogues des fabricants et sélectionner les composants a priori adaptés à votre besoin en termes de précision, vitesse et coût. Un butinage sur l'internet plus tard, vous aurez la fiche caractéristique du composant sur votre écran. Toutefois sa lecture risque de ne pas être aussi évidente que ça. À titre d'exemple, la **figure 2** vous montre un extrait des caractéristiques du CA/N *de compétition* que j'ai choisi pour illustrer cet article, un modèle à 16 bits/125 Méch/s (millions de conversions par seconde ; en anglais **MSPS**, **MegaSamples Per Second**) de la société Analog Devices, l'AD9265 [1]. Que veulent donc dire tous ces chiffres ? Tout d'abord remarquez qu'il y a deux groupes de spécifications : DC et AC.

Coté DC, le fournisseur vous dit d'abord « no missing code ». Cela veut simplement dire que chacune des $2^{16} = 65536$ valeurs possibles peuvent apparaître en sortie du convertisseur si l'on applique une rampe de tension en entrée. Cela peut paraître évident, mais beaucoup de CA/N ne le garantissent pas... Ensuite viennent les caractéristiques de précision de la conversion elle-même. L'erreur sur le zéro est de $\pm 0,05\%$ de la pleine échelle (FSR, **F**ull-**S**cale **R**ange), ce qui fait quand même une trentaine de bits de poids faible. Cela peut être critique, ou pas, tout dépend de votre application. Pire, son erreur en ce qui concerne le gain est de $+0,4\%$ de la pleine échelle. Réfléchissez, $\pm 0,4\%$ c'est une précision inférieure à 10 bits effectifs (car la résolution d'un convertisseur à 10 bits est $1/1024 = 0,1\%$), alors que ce convertisseur est un modèle à 16 bits. Qu'est-ce qui cloche ? Rien, simplement ce CA/N, comme la plupart des modèles rapides, a une bonne résolution (nombre de bits), mais une faible précision absolue. Il n'est simplement pas optimisé pour une grande précision, parce qu'il n'est pas conçu pour équiper un multimètre, mais pour des acquisitions rapides !

Les deux spécifications DC suivantes sont baptisées DNL et INL, et méritent une explication (**figure 3**). DNL veut dire « non-linéarité différentielle » (**D**ifferential **N**onlinearit**y**). Ici elle est de $\pm 0,7$ LSB (bit de poids faible), ce qui veut dire que l'écart de tension d'entrée pour deux codes de sortie successifs est toujours compris entre $1 - 0,7 = 30\%$ et $1 + 0,7 = 170\%$ de la valeur théorique.

La seconde valeur, INL (**I**ntegral **n**onlinearit**y**), est l'intégrale de cette erreur, c'est-à-dire l'écart maximal entre la courbe « code=f(tension d'entrée) » et la droite théorique. Jetez de nouveau un œil à la figure 3 pour comprendre.

La dernière valeur importante des caractéristiques DC est le bruit du CA/N, ici il est égal à « $2,17\text{LSB rms}$ ». Cela veut dire que, même avec une tension d'entrée parfaitement stable et propre, la sortie du CA/N aura des variations aléatoires de cet ordre de grandeur autour de la « bonne » valeur. Attention, il s'agit toujours d'une spécification en valeur efficace : le bruit en valeur crête à crête est bien plus élevé, typiquement 6 à 8 fois supérieur au bruit en valeur efficace. De là on peut s'attendre sur un tel convertisseur à avoir un bruit crête à crête de l'ordre de 16 LSB. Autrement dit avec un tel CA/N à 16 bits, seuls les 12 bits de poids forts vont être stables, les 4 bits de

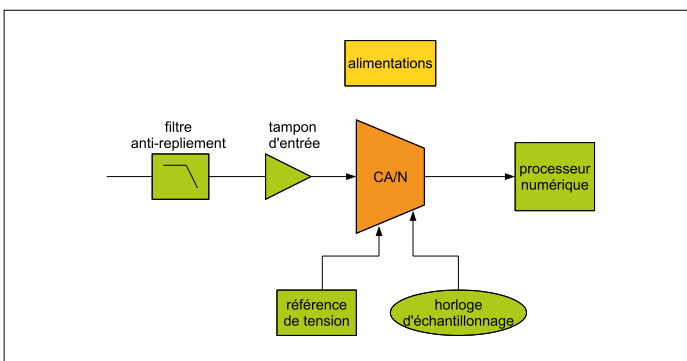


Figure 1. Le CA/N n'est qu'un élément d'un système d'acquisition de signal, les autres blocs sont aussi importants que le CA/N pour obtenir un circuit de qualité.

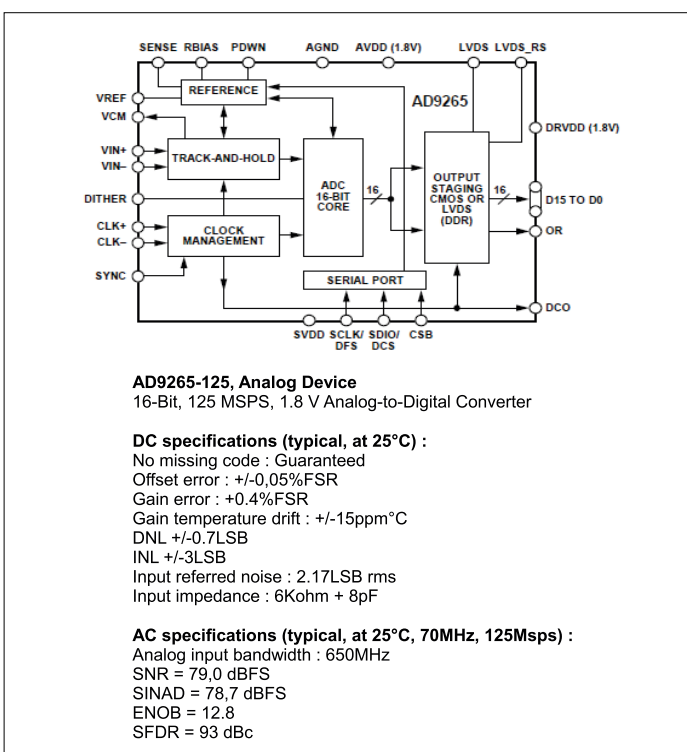


Figure 2. Un exemple d'excellentissime CA/N, il est utilisé pour cet article : l'AD9265-125 d'Analog Devices

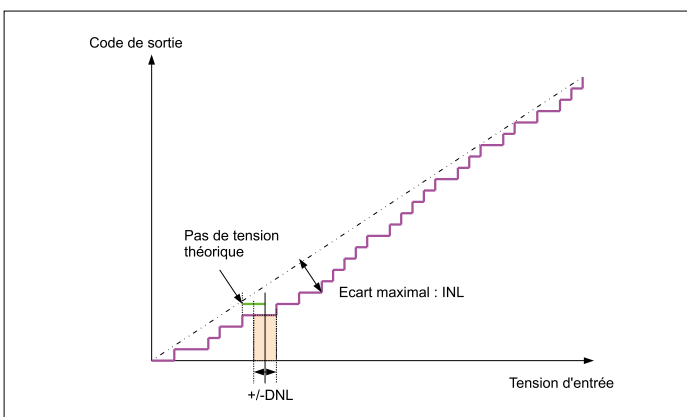


Figure 3. Deux des caractéristiques DC souvent mal comprises : la non-linéarité différentielle (DNL) et la non-linéarité intégrale (INL).

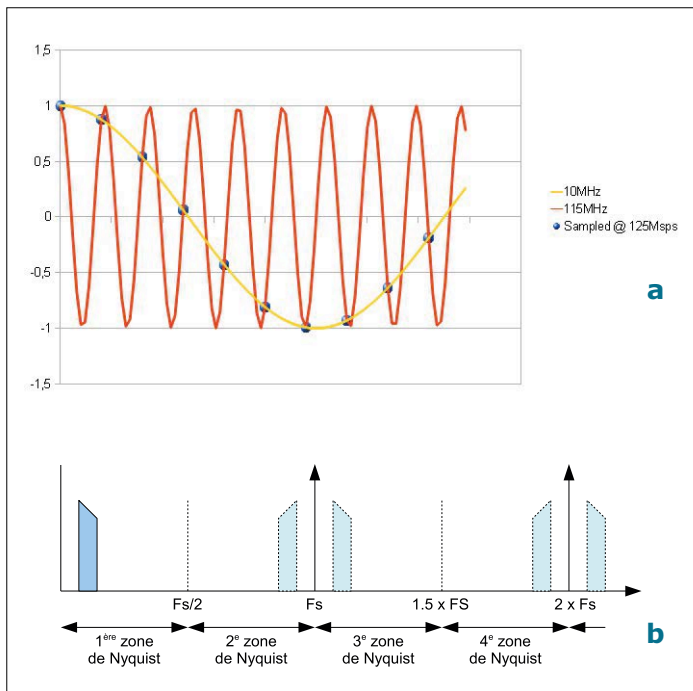


Figure 4. a) Une sinusoïde à 10 MHz (jaune) et une à 115 MHz (orange), toutes deux échantillonnées à 125 Méch/s : leurs valeurs sont exactement les mêmes aux points d'échantillonnage, c'est le repliement spectral. b) La première zone de Nyquist s'étant du continu à la moitié de la fréquence d'échantillonnage.

pois faibles sont très probablement bruités. Est-ce que cela veut dire qu'il n'est pas meilleur qu'un CA/N à 12 bits ? Non, même s'il y a du bruit, les bits de poids fort donnent de l'information sur le signal : un filtrage numérique ou une analyse de Fourier permet par ex. d'exploiter ces bits.

Enfin l'impédance d'entrée du composant est spécifiée ; elle est égale à 6 k Ω plus 8 pF. Ne négligez pas ces 8 pF : si la fréquence d'entrée est par ex. de 50 MHz alors ces 8 pF réduiront l'impédance d'entrée de 6 k Ω à 400 Ω ($1/(2 \cdot \pi \cdot f \cdot C)$), d'où des contraintes pour les étages en amont que nous verrons dans quelques minutes.

Caractéristiques AC

C'était la partie facile, regardons maintenant les spécifications AC. Tout d'abord la bande passante d'entrée de ce CA/N est

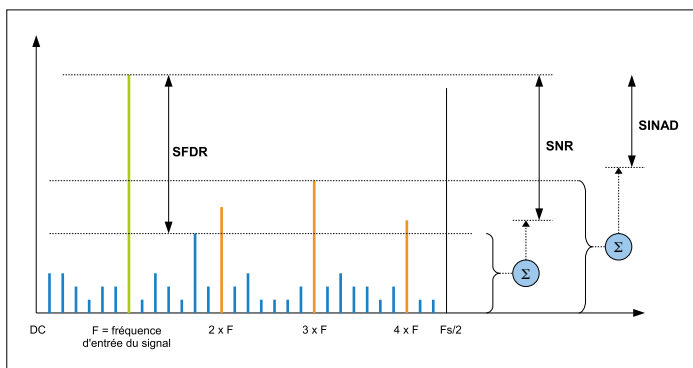


Figure 5. Une illustration des principales caractéristiques AC d'un CA/N, dans le domaine fréquentiel.

spécifiée : 650 MHz. Oh là, stop. Comment un CA/N d'une vitesse maximale de conversion de 125 Méch/s peut-il accepter un signal d'entrée de 650 MHz ? Monsieur Nyquist n'a-t-il pas dit que la fréquence de conversion doit être égale au moins au double de la fréquence d'entrée maximale ? Et bien pas exactement. Plus précisément, il a montré qu'en échantillonnant un signal à 125 Méch/s, il fallait que la bande utile du signal soit inférieure à $125/2 = 62,5$ MHz, sinon il y a des soucis, à savoir repliement spectral.

Oublions les formules, prenons un exemple. Regardez la **figure 4a**, qui illustre ce que « voit » notre CA/N qui fonctionne à 125 Méch/s dans le cas d'un signal d'entrée de respectivement 10 MHz et $125 - 10 = 115$ MHz. Dans les deux cas, les valeurs de sortie du CA/N sont exactement les mêmes ! Il faut donc impérativement que le signal d'entrée ne fasse pas plus de 62,5 MHz de largeur de bande, mais cette bande n'est pas nécessairement entre 0 et 62,5 MHz, appelée première bande de Nyquist (**figure 4b**). Dans le cas de notre AD9265, elle peut en fait être choisie n'importe où jusqu'à 650 MHz, c'est ce qu'on appelle le sous-échantillonnage. L'important est d'être certain de la bande dans laquelle on est, par une bonne connaissance du signal d'entrée ou par l'utilisation d'un filtre anti-repliement adapté, ce que nous verrons plus bas.

Revenons à nos spécifications AC. Les quatre autres valeurs portent les doux noms de SNR, SINAD, ENOB et SFDR. Il s'agit des caractéristiques dans le domaine spectral de notre CA/N, illustrées sur la **figure 5**. « SNR=79dBFS » veut dire que le bruit total en valeur efficace sur le signal numérique de sortie est 79 dB plus faible qu'un signal sinusoïdal de pleine échelle à une fréquence donnée (ici 70 MHz). Attention, on parle en puissance, et 79 dB correspondent à un ratio de puissance de $10^{79/10}$, donc à un ratio de tension de $10^{79/20} = 8900$. D'ailleurs cette valeur peut être comparée au SNR théorique d'un CA/N parfait de résolution N bits, qui vaut $6,02 N + 1,76$ dB, soit 98 dB pour un CA/N à 16 bits parfait. Comme ce SNR est mesuré avec un signal de très forte amplitude en entrée, il est bien sûr bien plus grand que le « bruit efficace » donné dans la rubrique DC, qui considère que l'entrée ne bouge pas.

Le second paramètre, SINAD, correspond au « rapport entre signal et (bruit + distortion) ». C'est comme le SNR, mais ici on tient compte non seulement du bruit, mais aussi des harmoniques de la fréquence d'entrée, qui sont inévitablement présents sur les codes de sortie du CA/N. Le SINAD est donc toujours moins bon que le SNR. L'écart est ici très faible ce qui implique que l'on a un CA/N particulièrement linéaire à forts niveaux. Ce SINAD, en dB, peut être converti en nombre de bits et on obtient ce qu'on appelle ENOB (« nombre de bits effectifs équivalents ») via la formule : $ENOB = (SINAD - 1,76 \text{ dB}) / 6,02$.

Enfin la dernière valeur, SFDR, veut dire « dynamique exempte de parasites ». Il s'agit grosso modo de la différence, sur une transformée de Fourier, entre l'amplitude du signal voulu et l'amplitude du parasite le plus fort créé par le CA/N lui-même. Comme le SFDR ne prend en compte qu'une raie parasite et non pas tout le bruit, il est plus optimiste que le SNR ou le SINAD.

Que de chiffres et de notions ! Seuls certains sont pertinents pour une application donnée. Par contre il est fondamental de

tous les connaître afin de savoir lequel est à examiner de près, ou pas, pour votre prochain projet...

Anti-repliement ?

Avec ces informations, vous devriez pouvoir sélectionner le CA/N idéal pour votre application. Reste maintenant à faire le schéma du circuit autour de celui-ci (figure 1). Commençons par le sacro-saint filtre anti-repliement. Comme nous l'avons vu, l'utilisation d'un CA/N nécessite que le spectre du signal d'entrée ait une largeur de bande inférieure à la moitié de la fréquence d'échantillonnage, soit ici $125 \text{ Méch/s} / 2 = 62,5 \text{ MHz}$. Le rôle du filtre anti-repliement est de s'assurer que c'est toujours le cas. Il s'agit en général d'un filtre passe-bas (qui sélectionne la première bande de Nyquist), mais un filtre passe-bande convient aussi pour sélectionner n'importe quelle autre bande compatible avec le CA/N choisi. Restons simples, intéressons-nous au premier cas.

Comment concevoir ce filtre pour utiliser au mieux les capacités du CA/N ? Supposons que l'on ignore tout du signal d'entrée. Le CA/N a un SINAD de 78,7 dBFS. Si on veut conserver cette qualité, il faut que le filtre anti-repliement atténue d'au moins 78,7 dB tout signal parasite au-delà de 62,5 MHz. Mais, de même, si vous ne voulez pas dégrader la précision du gain du CA/N dans la bande utile (qui est de 0,4%), alors le filtre anti-repliement doit avoir une réponse extrêmement plate dans cette bande, meilleure que 0,4 %, c'est-à-dire 0,01 dB. Il faut donc concevoir un filtre très plat dans la bande utile, mais qui procure une atténuation de plus de 78 dB à partir de 62,5 MHz (**figure 6**). Sa fréquence de coupure à 3 dB doit donc être bien plus basse que 62,5 MHz.

Est-ce qu'un tel filtre est facile à concevoir ? Pour mémoire un filtre dit d'ordre 1 atténue de 6 dB par octave (c'est-à-dire à chaque fois que la fréquence double). Il faut donc un filtre d'ordre 10 pour avoir une atténuation de 60 dB par octave. Comme il nous faut 78 dB d'atténuation, un tel filtre ne serait utilisable ici que si sa fréquence de coupure était $78/60 = 1,3$ octave plus basse que 62,5 MHz, c'est-à-dire à $62,5/2^{1,3} = 25,3 \text{ MHz}$. En résumé : même avec un filtre aussi complexe qu'un filtre du 10^e ordre, il est impossible de concevoir un filtre anti-repliement pour le CA/N à 125 Méch/s choisi en exemple qui ne dégrade pas ses performances, à moins d'avoir une fréquence de coupure de 25 MHz. Et la bande réellement utilisable sera encore plus faible pour garantir un gain à $\pm 0,01 \text{ dB}$ près. On est très loin des 62,5 MHz espérés ! Bien sûr en pratique, on connaît à peu près le signal d'entrée, et les atténuations nécessaires du filtre peuvent donc être assouplies, mais vous avez compris l'esprit. Notez aussi que c'est justement pour simplifier les filtres anti-repliement qu'on utilise souvent des techniques dites de sur-échantillonnage, sujet que j'aborderai dans mon prochain article.

Ampli d'entrée

Nous avons le filtre anti-repliement. Comment relier sa sortie à l'entrée du CA/N ? Via un autre élément critique du circuit : l'amplificateur d'entrée. Il a pour rôle de présenter une impédance de sortie compatible avec l'impédance d'entrée du CA/N. Et ce n'est pas toujours une partie de plaisir, car celle-ci peut être très basse en hautes fréquences, et surtout dépendante

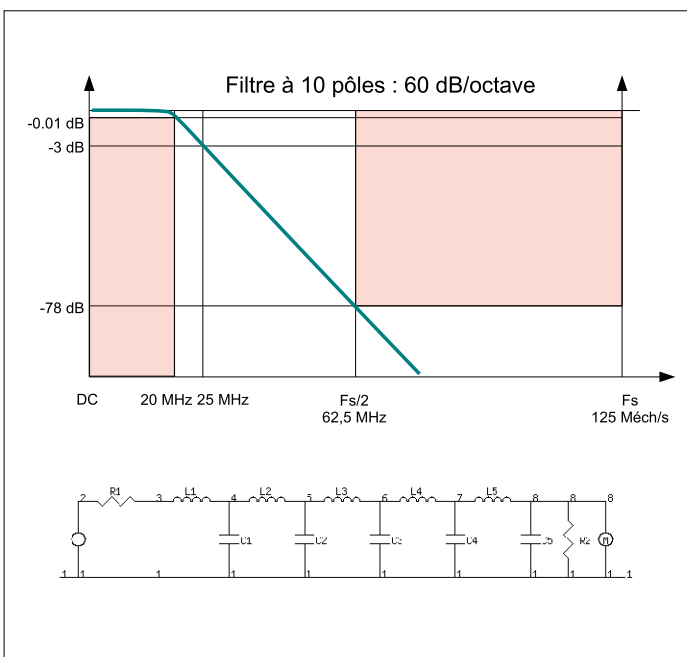


Figure 6. Dans l'exemple présenté, le filtre anti-repliement doit atténuer d'au moins 78 dB les signaux hors de la première bande de Nyquist. Pas facile, même avec un filtre d'ordre 10 !

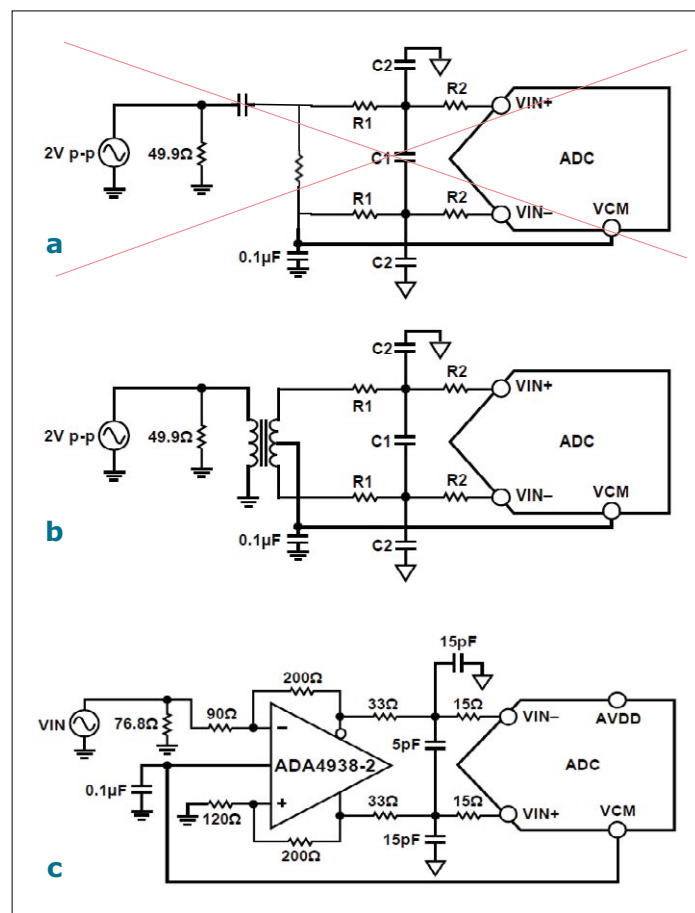


Figure 7. Trois solutions pour convertir un signal unipolaire en différentiel : une mauvaise (a) et deux bonnes (b et c, extraites de la notice de l'AD9265).

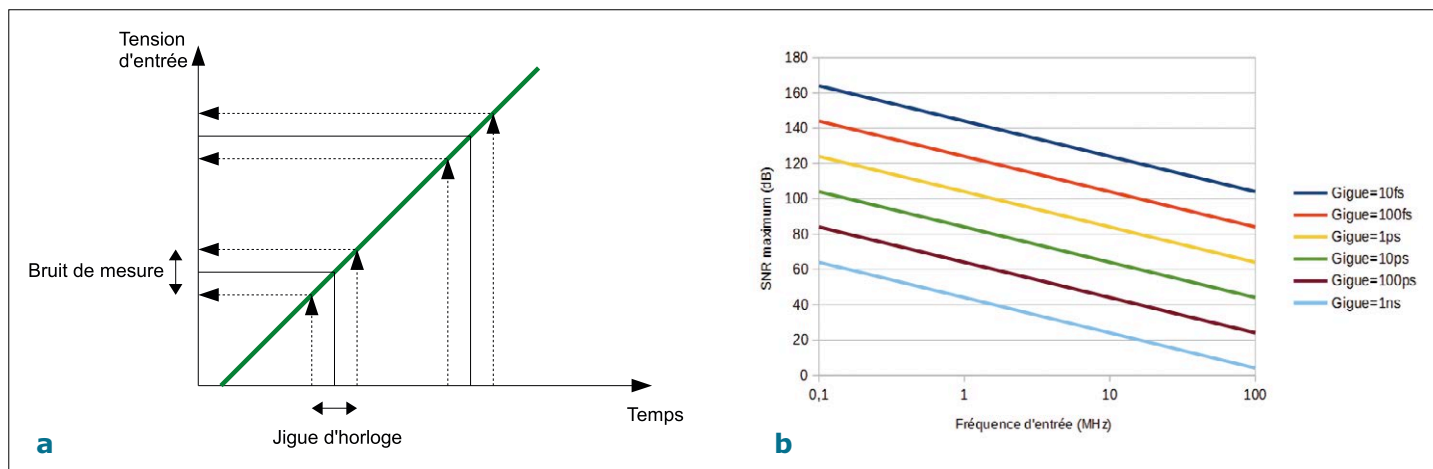


Figure 8. a) Une gigue temporelle du signal d'horloge est équivalente à un bruit sur la tension mesurée.

b) Cette abaque fournit le SNR maximal possible avec une gigue d'horloge donnée, en fonction de la fréquence du signal d'entrée.

de la fréquence. Ici encore tout est lié aux objectifs visés ; en règle générale, plus le CA/N est haut de gamme, plus cette section est délicate... Dans ce cas, le mieux est de se fier aux recommandations du fabricant du CA/N ! Une autre contrainte est que l'entrée d'un CA/N rapide est souvent différentielle, alors que votre signal l'est rarement. Dans un tel cas, il y a trois solutions, une mauvaise et deux bonnes. La mauvaise, illustrée sur la **figure 7a**, est de n'utiliser qu'une des deux entrées différentielles du CA/N en mettant l'autre à la masse ou à un potentiel fixe. Ça marche, mais au prix de performances très dégradées. Quel dommage pour un composant si cher ! En HF, la meilleure solution est souvent d'utiliser un petit transformateur, appelé dans ce cas « balun », pour convertir un signal unipolaire en signal différentiel (**figure 7b**). Le seul inconvénient de cette approche est qu'un transformateur ne laisse pas passer de signal continu. Si c'est un souci alors la troisième solution est d'utiliser un circuit intégré dédié qui se charge de la conversion (**figure 7c**), mais il faut bien le choisir pour ne pas perdre des capacités du CA/N.

Horloge d'échantillonnage

Un autre sujet difficile et souvent mal compris : l'horloge d'échantillonnage. Par nature un CA/N permet de prendre des mesures de la tension d'entrée à des instants précis. Imaginez que le signal d'entrée est une jolie rampe linéaire, mais que l'horloge n'est pas très précise, c'est-à-dire que les instants

d'acquisition ne sont pas exactement les instants voulus (ce qu'on appelle de la gigue d'horloge). Comme illustré sur la **figure 8a**, cette erreur temporelle est strictement la même chose qu'une erreur en amplitude.

Dans un cas plus proche de la réalité, imaginons que le signal d'entrée est une sinusoïde. Alors une gigue sur l'horloge d'échantillonnage va impliquer un bruit additionnel sur la valeur de sortie du CA/N, à savoir une réduction du rapport signal-bruit que l'on pourra atteindre si tout le reste du circuit est parfait. La formule qui donne cette limite est très simple :

$$SNR_{\max} = 20 \log_{10} \left(\frac{1}{2\pi f t_{\text{gigue}}} \right)$$

Dans cette formule, f est la fréquence du signal d'entrée et t_{gigue} est la gigue (valeur efficace) du signal d'horloge en seconde. Faites le calcul, ou regardez l'abaque correspondant (**figure 8b**). Vous constaterez que si l'on ne veut pas dégrader le SNR de 79 dB de notre CA/N et que si la fréquence du signal d'entrée est de 20 MHz, alors l'horloge doit avoir une gigue maximale de 0,9 ps. Oui, picosecondes. Et ça n'est pas beaucoup...

Utiliser une simple porte logique CMOS comme une 74HCT00 par ex. produit déjà une gigue de 2,2 ps ! La conclusion de

Liens et références

- [1] Feuille de caractéristiques du CA/N AD9642 à 16 bits, 125 Méc/s, Analog Device : www.analog.com/static/imported-files/data_sheets/AD9642.pdf
- [2] Feuille de caractéristiques de la référence de tension ultra-précise MAX6126, Maxim : www.maxim-ic.com/datasheet/index.mvp/id/3623
- [3] *The Data Conversion Handbook*, Walt Kester (Analog Device), mars 2004 : www.analog.com/library/analogDialogue/archives/39-06/data_conversion_handbook.html
- [4] *Understanding AC behaviors of high-speed ADCs*, David Kress, Analog Devices : www.analog.com/static/imported-files/tech_articles/MS-2124.pdf
- [5] *Understanding the effect of clock jitter on high-speed ADCs*, Derek Redmayne & Alison Steer, Linear Technologies : <http://cds.linear.com/docs/Design%20Note/dn1013f.pdf>

tout ceci est que l'horloge d'un CA/N rapide et précis doit être aux petits oignons, au risque de n'obtenir que des résultats décevants. On utilise donc des oscillateurs à très faible gigue comme des TCXO (*Temperature Compensated Crystal Oscillator*) ou des OCXO (*Oven Controlled X-tal(Crystal) Oscillator*), et des transistors en régime non saturé pour éviter tout ajout de gigue, ou des circuits d'horloge dédiés. Malheureusement, faible gigue rime avec forte consommation énergétique (car les courants faibles sont plus perturbés par du bruit), mais c'est la vie.

Référence de tension

Encore une tracasserie ? Allez, une dernière pour la route : la référence de tension. Un CA/N compare la tension d'entrée avec une tension de référence (interne ou externe) pour calculer le code de sortie. La précision absolue de la conversion dépend donc directement de la précision de cette référence. Si l'on échantillonne un signal rapide alors en général la valeur absolue n'est pas réellement critique, le plus important est de réduire le bruit sur la référence de tension (ferrites, condensateurs bien choisis, etc.). Par contre dans d'autres applications, plus lentes, on recherche une précision absolue. C'est par ex. le cas d'un multimètre. En réfléchissant, vous allez vite vous rendre compte que dans de tels cas c'est la précision de la référence de tension qui est décisive.

Un exemple ? Imaginez que vous avez un CA/N à 16 bits, qui ne coûte qu'un ou deux euros, piloté par une tension de référence de 2 V (qu'elle soit interne ou externe au CA/N, c'est la même chose). La résolution d'un tel CA/N est de $2 \text{ V} / 2^{16} = 15 \mu\text{V}$. Quelle référence de tension utiliser si l'on veut une précision absolue du même ordre ? C'est quasiment impossible ! En effet les meilleures références de tension disponibles sur le marché, comme par ex. le circuit MAX6126 [2] de Maxim (qui coûte trois ou quatre fois plus cher que le CA/N), a une précision de $\pm 0,02 \%$ et ce sans même compter sa dérive en température. $0,02 \%$ de 2 V donnent $400 \mu\text{V}$, soit 30 fois moins bien que la résolution du CA/N !

Cela vous explique pourquoi les multimètres de grande précision sont hors de prix : ce n'est pas le nombre de digits qui fait le prix, donc la résolution du CA/N, mais la précision absolue qui est celle de sa référence de tension.

Pour conclure

L'objet de cet article n'était pas de vous effrayer, mais de vous expliquer pourquoi l'utilisation d'un CA/N doit s'accompagner d'un peu de rigueur dans sa mise en œuvre dès lors qu'on veut atteindre des caractéristiques remarquables en termes de vitesse ou précision. Les fabricants ont développé des composants dotés de capacités ahurissantes, mais on ne conduit pas une Ferrari comme une 2 CV ! Ceci dit, et comme d'habitude, expérimenter est souvent la meilleure solution pour réellement comprendre ce qui se passe. Je vous encourage donc vivement à mettre tout cela en pratique : regardez en détail les caractéristiques de votre CA/N, et tâchez de ne pas les gâcher avec les circuits connexes.

Mon prochain article traitera d'une technique de conversion analogique/numérique un peu spécifique : la conversion sigma-delta. À bientôt donc !

(160099)

Cet article a été publié dans la revue Circuit Cellar (n°259, février 2012).

Können Sie Deutsch? Verstaat u Nederlands?

Vous comprenez l'allemand ou le néerlandais et l'électronique, rejoignez l'équipe de traducteurs qui prépare chaque mois la version française du magazine Elektor.

Nous recherchons des traducteurs techniques néerlandais/français et allemand/français pour un travail régulier.

Contactez-nous : redaction@elektor.fr



Le Groupement de Service *eco-info* du CNRS organise une conférence sur l'éco-conception, le 27 septembre 2016, à Grenoble. Les



organisateur est à la recherche d'intervenants et de spécialistes impliqués dans l'éco-conception des cartes électroniques à toutes les étapes de leur cycle de vie (substrats, composants, techniques de réalisation des circuits imprimés, conception de programmes économes en énergie, filières de recyclage et économie circulaire, etc.).

Si vous souhaitez apporter votre contribution à cette conférence, contactez-nous : redaction@elektor.fr

L'objectif du GDS *eco-info* d'agir pour réduire les impacts environnementaux et sociétaux des TIC (Technologies de l'Information et de la Communication).

AGENDA

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

juillet 2016

♦ **Japan Expo Paris** Festival des loisirs japonais
07 au 10/07 – Paris
www.japan-expo-paris.com/fr/



♦ **Maker Faire Nantes**
08 au 10/07 – Nantes
www.makerfairenantes.com



♦ **Salon du véhicule électrique et hybride**
14 au 17/07 - Val d'Isère
www.salon-vehicule-electrique.com

Un événement oublié ?

Vous organisez une conférence, un salon... ou bien vous participez à un séminaire ou tout autre événement qui aurait sa place ici, partagez cette information avec tous les lecteurs.

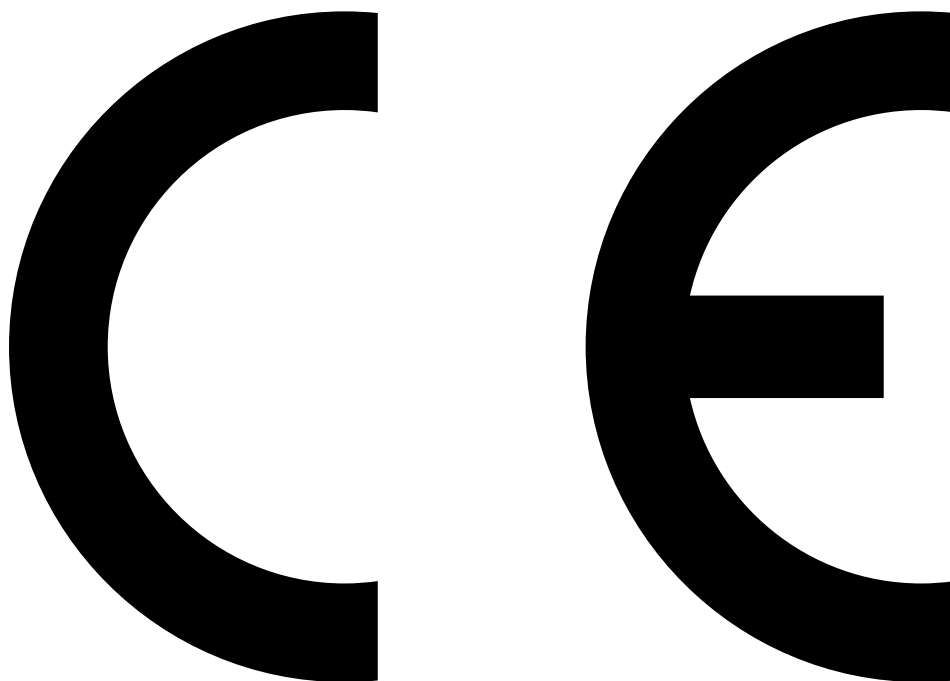
Envoyez-nous tous les détails à redaction@elektor.fr.

quand CE devient UE

une directive chasse l'autre

Mariline Thiebaut-Brodier (Elektor)

Les produits électroniques et leurs modes d'utilisation évoluent. Aujourd'hui ces produits sont non seulement connectés, pilotés à distance, mais ils sont aussi émetteurs et récepteurs de données... C'est pourquoi la législation est contrainte de s'adapter. En 2016, les directives Basse tension (BT), Compatibilité électromagnétique (CEM) et Radio (R&TTE) changent également.



Déclaration « UE » de conformité

Pour mémoire, si vous souhaitez vendre du matériel sur le marché européen, vous êtes responsable de l'évaluation de la conformité du produit et soumis à une série d'obligations, y compris à des exigences en matière de traçabilité, que vous soyez établi en dehors de l'Union européenne ou dans un État membre.

Avant de distribuer un produit, son fabricant doit :

1. réaliser ou faire réaliser l'évaluation de la conformité ; selon le cas soumettre le produit à un organisme tiers (généralement un organisme notifié). En tout état de cause, le fabricant assume la pleine responsabilité quant à la conformité du produit.
2. établir la documentation technique
3. établir la déclaration « UE » de conformité
4. veiller à ce que le produit soit accompagné d'instructions et d'informations de sécurité, dans une langue aisément compréhensible par les consommateurs et les autres utilisateurs finaux
5. satisfaire aux exigences en matière de traçabilité (documentation technique

et déclaration « UE » conservées pendant dix ans ; apposition d'un numéro de série/type/lot ; nom/raison sociale/marque déposée et adresse apposés sur le produit)

6. apposer le marquage de conformité (« CE », *Conforme aux Exigences*)
7. s'assurer que la production en série reste conforme
8. le cas échéant, procéder à la certification du produit et/ou du système de qualité.

La déclaration « UE » de conformité est le document qui atteste qu'un produit est conforme aux exigences applicables d'une – ou plusieurs – directive(s) européenne(s). Ces directives définissent des exigences essentielles en matière de sécurité, de santé et d'environnement, ainsi que des procédures d'évaluation. Même si un produit est soumis à plusieurs directives, il n'y a qu'une seule déclaration à établir. Généralement, celle-ci est jointe à la notice du produit. L'Union Européenne publie régulièrement une liste des normes applicables pour répondre aux exigences de telle ou telle directive. Notez que la déclaration « CE » de confor-

mité a été rebaptisée « UE » parce que l'abréviation CE a fait l'objet d'un détournement. En effet des produits sont entrés en Europe avec une déclaration *China Export*.

Le marquage « CE » est un indicateur (mais pas une preuve) de la conformité d'un produit à la législation de l'UE et permet sa libre circulation au sein du marché européen, qu'il soit fabriqué au sein de l'Espace économique européen, en Turquie ou dans un autre pays.

Pour des informations plus détaillées, reportez-vous au **guide bleu** relatif à la mise en œuvre de la réglementation de l'UE sur les produits [1].

Basse Tension

Les directives Basse tension (BT), Compatibilité électromagnétique (CEM) et Radio (R&TTE) sont les principales directives applicables aux produits électroniques. Le **tableau** donne une vue d'ensemble des principaux changements intervenus depuis le début de l'année. Les directives BT et CEM ont été modifiées ; la directive R&TTE a été révisée et renommée RED. La version révisée de la directive Basse

Tension contient quelques paragraphes dont les conséquences ne sont pas anodines :

- (4) La présente directive s'applique au matériel électrique destiné à être employé dans certaines limites de tension qui est **nouveau** pour le marché de l'Union lors de sa mise sur le marché ; en d'autres termes, il s'agit soit d'un matériel électrique neuf dont le fabricant est établi dans l'Union, soit d'un matériel **neuf ou d'occasion** importé d'un pays tiers.
- (5) La présente directive devrait s'appliquer à toutes les formes de fourniture, y compris la **vente à distance**.

La distinction entre neuf et d'occasion n'apparaissait pas auparavant, ni la VPC. Autrement dit, si vous achetez un appareil électronique usagé sur un site hébergé à l'autre bout de la planète, le vendeur doit fournir une déclaration de conformité UE. Cela va être compliqué !

- (9) Le fabricant, en raison de la connaissance détaillée qu'il a de la conception et du processus de production, est le mieux placé pour mettre en œuvre la procédure d'évaluation de la conformité. L'évaluation de la conformité devrait, par conséquent, incomber au seul fabricant. **Cette directive ne prévoit aucune procédure d'évaluation de la confor-**

mité qui requiert l'intervention d'un organisme notifié.

Les organismes notifiés (ON) tendent à disparaître de la chaîne d'évaluation de la conformité des produits. Le risque est bien évidemment la mise sur le marché de produits non conformes. *(Un organisme (ou laboratoire) notifié est un organisme qui est désigné, à sa demande, par les pouvoirs publics d'un État membre auprès de la Commission européenne pour réaliser des procédures d'évaluation de la conformité de produits.)*

- (18) Les normes harmonisées pertinentes pour la présente directive devraient aussi tenir compte de la

Directive			
	Basse Tension (BT)	Compatibilité électromagnétique (CEM)	Radio (R&TTE)
ancienne directive	2006/95/CE abrogée le 19/04/2016	2004/108/CE abrogée le 19/04/2016	1999/5/CE (R&TTE) seule applicable jusqu'au 13/06/2016
directive révisée	2014/35/UE obligatoire à partir du 20/04/2016	2014/30/UE obligatoire à partir du 20/04/2016	2014/53/UE (RED) applicable à partir du 13/06/2016
principales nouveautés	La déclaration CE de conformité devient une déclaration UE de conformité		
	Les importateurs doivent mentionner leurs noms et coordonnées sur les produits (ou à défaut sur l'emballage).		
	Les responsabilités relatives à la mise sur le marché des produits (exigences de traçabilité et de surveillance notamment) sont précisées et renforcées pour les fabricants, les importateurs et les distributeurs.		
	Les procédures d'évaluation de la conformité respectent la « Nouvelle Approche ».		
			Les produits Terminaux de Télécommunication (TTE) sortent du champ d'application de la directive RED
			Les produits de type récepteur TV et FM entrent dans le champ d'application de la directive RED.
ce qui ne change pas	Marquage CE		
	Champ d'application et exigences essentielles de santé et de sécurité		
conséquences	Comme les déclarations UE de conformité suivant la directive révisée remplacent les déclarations CE de conformité, chez les fabricants, il a fallu les remplacer dans les emballages de tous les produits en stock avant le 20/04/2016.		Pendant un an les deux directives seront applicables pour les produits qui étaient couverts par la directive R&TTE et qui restent couverts par la directive RED. Depuis le 13/06/2016, le matériel en stock doit aussi répondre aux exigences RED.
	Le matériel stocké chez un distributeur ou un utilisateur est considéré comme déjà mis sur le marché : la directive révisée ne s'applique pas.		
période de transition	Il n'existe pas de période de recouvrement pendant laquelle les directives anciennes et révisées sont applicables. La date de basculement a eu lieu 20/04/2016.		Il y a une période de recouvrement pendant laquelle les directives R&TTE et RED sont applicables. La période transitoire prendra fin le 13/06/2017.

convention des Nations unies relative aux [droits des personnes handicapées](#).

Annexe sur le contrôle interne de la fabrication : la documentation technique inclut [une analyse et une évaluation adéquates du ou des risques](#).

C'est nouveau et le travail se complique ! Est-on capable de penser à tous les cas de mauvaise utilisation pour tous les types d'utilisateurs (handicapés ou non) ? Va-t-on arriver à des notices comme celles des médicaments avec une interminable liste d'effets indésirables ?

- **Article 6 - Obligations des fabricants**
§ 8. Les fabricants qui considèrent ou ont des raisons de croire qu'un matériel électrique qu'ils ont mis sur le marché n'est pas conforme à la présente directive prennent immédiatement les mesures correctives nécessaires pour le mettre en conformité, le retirer ou le rappeler, si nécessaire.

La responsabilité des fabricants est renforcée, ils doivent pouvoir démontrer qu'ils ont pris en compte un problème signalé par leur SAV. L'article qui suit celui-ci indique qu'ils doivent coopérer avec les autorités pour démontrer la conformité de leur produit.

Dans le même ordre d'idée, il y a une autre grande nouveauté : les importateurs et plus généralement tous les opérateurs économiques (mandataire, distributeur...) ont les mêmes obligations et responsabilités en matière de conformité que le fabricant. Même si cela peut paraître évident, ce n'était pas clairement mentionné dans la version précédente de la directive.

Un importateur doit par ex. conserver pendant dix ans la documentation technique du produit, apposer son nom et adresse sur le produit, s'assurer que le produit est accompagné d'un manuel d'utilisation et d'instructions de sécurité, coopérer avec les autorités... Il doit également veiller à ce que les conditions de stockage soient correctes, comme ne

pas laisser des piles au lithium derrière des baies vitrées pendant de longs mois.

Enfin, certains matériels sont exclus du champ d'application de la directive Basse Tension, en particulier les *kits d'évaluation fabriqués sur mesure à destination des professionnels et destinés à être utilisés uniquement dans des installations*

de recherche et de développement à de telles fins, en clair les prototypes.

Maintenant vous savez pourquoi cette année vous trouverez dans l'emballage de vos appareils électriques neufs soit une déclaration UE, soit une CE. ◀

(160094-I)

EU Declaration of Conformity (DoC)

We

Company name:
Postal address:
Postcode and City:
Telephone number:
E-Mail address:

declare that the DoC is issued under our sole responsibility and belongs to the following product:

Apparatus model/Product:
Type:
Batch:
Serial number:

Object of the declaration (identification of apparatus allowing traceability; it may include a colour image of sufficient clarity where necessary for the identification of the apparatus):

The object of the declaration described above is in conformity with the relevant Union harmonisation legislation:
EMC Directive 2014/30/EU

The following harmonised standards and technical specifications have been applied:

Title, Date of standard/specification:

Notified body (where applicable):

4 digit notified body number:

Additional information:

Signed for and on behalf of:

Place of issue

Date of issue

Name, function, signature

Lien

[1] Guide bleu, version 2014 :

<http://docplayer.fr/377927-Le-guide-bleu-relatif-a-la-mise-en-oeuvre-de-la-reglementation-de-l-ue-sur-les-produits.html>

twinBot

ou la téléprésence mobile

Chris Krohne (Allemagne)

Participer à une réunion alors que vous n'êtes pas sur place ? C'est déjà possible par téléconférence avec Skype, par exemple, mais twinBot va plus loin. Grâce à la mobilité de twinBot, vous prenez part à la conversation en téléprésence, vous allez voir ailleurs, vous suivez le guide... À l'instar d'ElektorWheelie, ce robot autostabilisé peut rouler pour vous et il est disponible en kit dans l'e-choppe.

Il n'est fait ni de chair, ni de sang, mais de tôle et d'électronique, cet avatar qui ressortit au genre technique singulièrement aride de « téléprésence ». Il s'agit d'une sorte de robot télécommandé, équipé d'une caméra pour voir, d'un haut-parleur pour s'exprimer, de roues pour se mouvoir et surtout d'un écran pour être vu. C'est une solution de haute technologie pour rencontrer des collègues dans différents pays ou même d'autres continents. De récentes expériences ont étonné par la vitesse à laquelle les collègues biologiques peuvent s'habituer à la présence artificielle de collègues distants. Il est même arrivé que des doubles métalliques offrent aux personnes présentes une tournée au champagne...

La téléprésence

L'idée d'une présence mobile virtuelle n'est pas vraiment neuve. Elle a été mise en œuvre par le personnage de Sheldon Cooper dans la série américaine *The Big Bang Theory*. Ce super « nerd » s'est fabriqué un robot mobile improvisé avec une caméra et un écran, à télécommander par l'internet. À condition d'avoir une connexion rapide, le système se déploie n'importe où dans le monde. Caméra, écran et moyen de locomotion sont les ingrédients indispensables d'un robot de téléprésence. Ce double moderne est manifestement plus souple qu'une simple liaison vidéo, puisqu'il permet de communiquer



Figure 1. Le twinBot sous toutes les coutures : un robot de téléprésence mince, mobile sur rouleaux et autostabilisé.

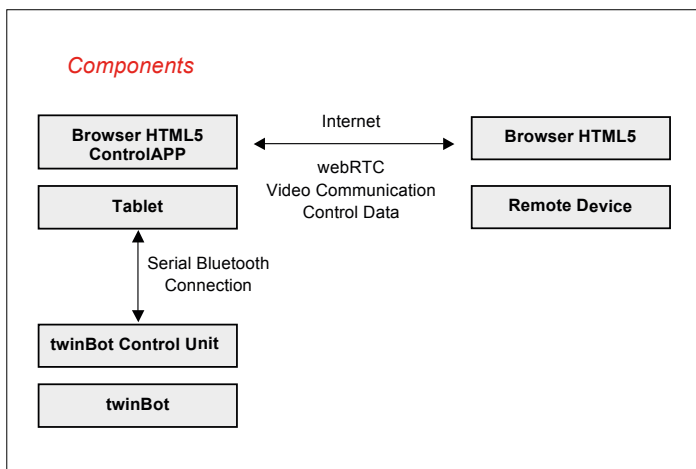


Figure 2. Un twinBot, c'est essentiellement une unité de commande et une tablette Android. Le reste passe par l'internet.

rapidement et d'interagir comme si l'on était présent physiquement sur place.

Les rôles envisageables à confier à un tel robot sont innombrables. On pense à une salle de conférence très lointaine ou à la surveillance d'un immeuble. Il peut aussi participer utilement à la protection, l'accompagnement et l'éducation. Il deviendra possible par ex. de louer depuis l'Europe un de ces robots pour visiter des endroits lointains, de parcourir des salles d'exposition comme celles du MoMA (musée d'Art moderne de New York). On dirige alors soi-même le robot à travers le musée pendant qu'il transmet par l'internet les images stéréo HD de sa caméra 3D directement sur votre télévision 3D en haute définition. Des études en ce sens prédisent un énorme succès de la téléprésence robotique dans les cinq prochaines années, sans compter tous les domaines d'application auxquels on n'a pas encore pensé.

Je considère le twinBot comme une plateforme de développement pour la construction et la mise au point d'un robot de téléprésence autostabilisé, comme un petit gyropode Segway. D'autres robots de téléprésence en revanche exigent pour une stabilité convenable un assez grand plateau. Le twinBot est beaucoup plus svelte, ses mouvements en sont plus proches de ceux de l'homme, autant que son aspect. L'algorithme d'équilibrage rend la chute impossible en principe. Comme les deux moteurs sont commandés séparément, le rayon de braquage est minimal. Le twinBot emporte une tablette Android compacte et bon marché au format de huit pouces. La tablette communique avec l'extérieur et pilote l'unité de commande du twinBot par une liaison série Bluetooth. Les figures 1a, 1b et 1c montrent toute l'esthétique du robot terminé.

Composition

Un twinBot se compose d'une partie électromécanique de propulsion, d'une unité de commande et d'une tablette Android comme à la figure 2. L'unité de commande est équipée d'un module Bluetooth par lequel il reçoit de la tablette les commandes de conduite. Il est aussi prévu de se relier directement à l'unité de commande et ainsi de piloter soi-même le twinBot. Il y a pour cela un protocole simple qui permet, par exemple, de le faire sortir du mode **Park** par une séquence d'instructions et de le mettre en marche par l'instruction *forward*. Ces ins-

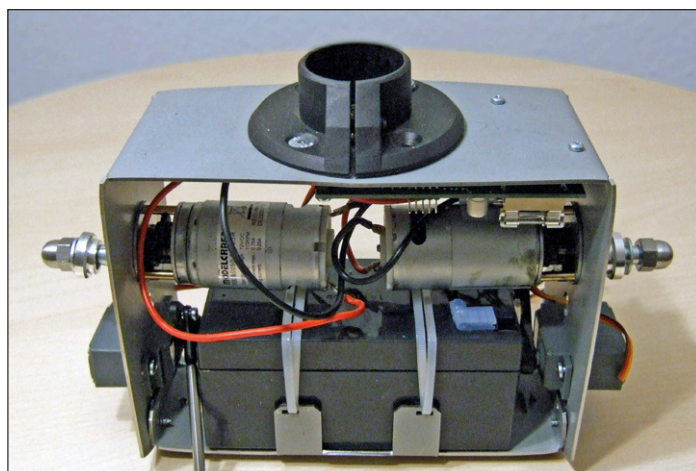


Figure 3. Châssis de montage pour les moteurs et l'électronique.

tructions doivent lui parvenir continuellement, sinon il passe en mode **Stop** par sécurité. L'unité de commande envoie à la tablette les données de charge de l'accumulateur, de l'angle d'inclinaison et de la vitesse.

Quand la liaison entre l'unité de commande et la tablette est établie, l'appli passerelle sur la tablette s'occupe de la connexion à l'internet.

La mécanique

La construction mécanique du twinBot est très simple. Tous les composants nécessaires sont casés dans un châssis de montage profilé (figure 3). Les roues en aluminium ont un diamètre de 20 cm et sont garnies d'une ceinture en caoutchouc (figure 4) pour former une bande de roulement à friction. Chaque roue est accouplée directement à l'arbre du moteur par un cône de serrage. Les moteurs d'entraînement à courant continu ne réclament pas plus de 20 W de puissance chacun. L'alimentation est assurée par une batterie de 12 V. Le twinBot est doté de deux béquilles de stationnement actionnées par servomoteurs de modélisme. Les forces sont ainsi partagées pour qu'un très faible courant de maintien du servo soit suffisant. Comme on le voit à la figure 3, c'est l'habituel accu au plomb gélifié (AGM) de 12 V qui prend place dans le profilé principal à côté des moteurs et de l'unité de commande. L'ensemble disparaît alors dans un boîtier tubulaire en plastique de belle facture (cf. figure 1).

Au sommet du châssis se trouve un collier de soutien de la colonne verticale d'un diamètre de 35 mm. Le boîtier plastique comporte déjà toutes les ouvertures pour la colonne, l'interrupteur principal, la LED, la prise de recharge et les béquilles de stationnement. Il ne faut que deux vis sous le boîtier pour fixer le châssis profilé. La colonne verticale abrite le câble qui relie le servomoteur d'inclinaison capable de modifier l'orientation de la tablette d'un angle de $\pm 45^\circ$. Le câble à trois conducteurs pour le servomoteur se branche directement à l'unité de commande située immédiatement sous le collier de soutien. La colonne verticale est en deux parties télescopiques pour l'ajuster à la hauteur désirée et la conserver au moyen d'une bague de serrage synthétique. Il est très facile, de la sorte, d'adapter la hauteur du robot à toute nécessité. Le bout supérieur de la colonne porte un cadre rectangulaire

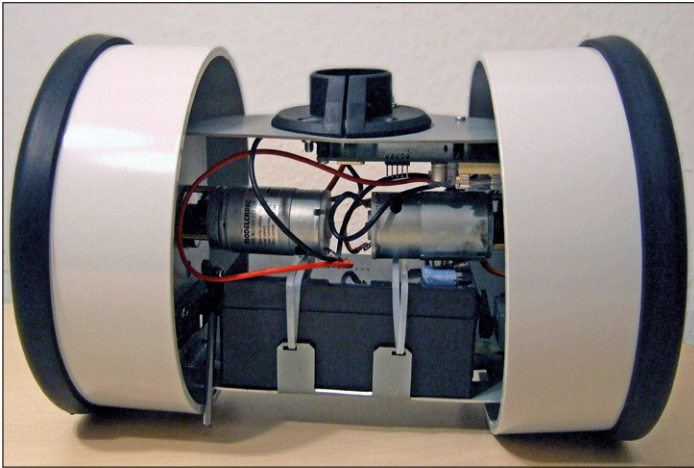


Figure 4. Le châssis équipé des roues.

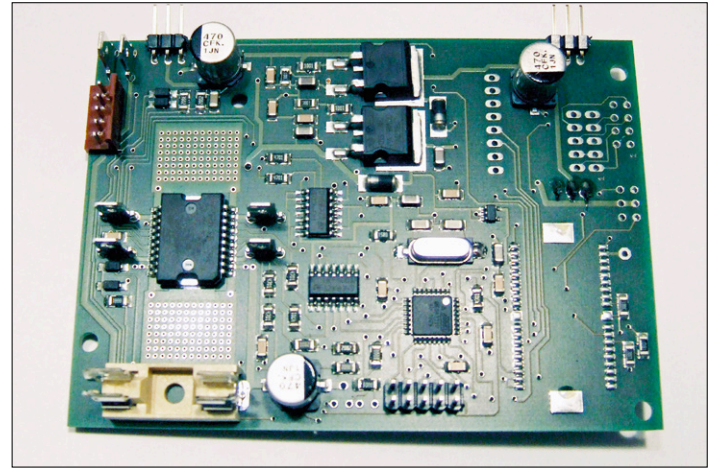


Figure 5. Les constituants électroniques du twinBot.

auquel se fixe le servomoteur de basculement. Reste à coller sur ce servomoteur un support universel pour tablette à écran de huit pouces.

L'électronique

La pièce maîtresse de l'unité de commande, à la **figure 5**, est un microcontrôleur AVR ATmega cadencé à 20 MHz, le schéma en est détaillé à la **figure 6**. Sa tâche principale consiste à consulter le module MPU6050 équipé d'un gyroscope et d'un accéléromètre, tous deux à trois axes, de traiter les données des capteurs et de commander les moteurs pour que le twinBot puisse s'équilibrer sur place et se déplacer lentement à la demande. Le module MPU6050 communique par I²C avec l'ATmega. Il y a dans l'unité de commande un double pont en H, un L6205P de ST (IC6), pour actionner les deux moteurs de propulsion à courant continu. Cette puce offre en outre des protections internes contre la surcharge et l'excès de température. L'entraînement des moteurs s'effectue par deux signaux MLI et deux lignes de commande de leur sens de rotation. Ces signaux passent par des portes NON ET qui commutent en MLI les bornes correspondantes des moteurs en fonction du sens de rotation voulu. La même puce dispose encore de deux entrées de validation (*enable*) qui permettent au microcontrôleur de désactiver chacun des ponts en H.

Les lignes RX et TX de l'UART de l'ATmega sont raccordées à IC3, un module Bluetooth BTM222 (classe 1) dont la portée atteint 100 m. Il en résulte une liaison Bluetooth très stable.

Deux autres entrées de l'ATmega sont reliées à une embase d'E/S à usage général ; on peut y brancher d'autres capteurs et des extensions personnelles. Un exemple serait d'y mettre des capteurs à effet Hall pour déterminer la vitesse de rotation des roues.

La tension de batterie est surveillée par une entrée de C/A/N de l'ATmega. Une autre entrée du C/A/N sert à connaître la vitesse du robot grâce à deux diodes reliées chacune à l'un des moteurs et un condensateur. Dans ce but, les ponts en H sont désactivés brièvement à une fréquence de 2 Hz, le temps de mesurer la force contre-électromotrice sur chaque moteur, qui fonctionne alors en génératrice. Cette f. c. e. m. est proportionnelle à la vitesse. Le signal est aplani par le condensa-

teur et échantillonné plusieurs fois pendant la phase de pause pour en calculer la moyenne. L'interrupteur principal, le témoin à LED et la prise de recharge se branchent à une embase à cinq broches.

L'unité de commande dispose de deux régulateurs de tension. Un LM317 alimente uniquement les trois servomoteurs. L'ATmega fournit pour cela sur trois lignes de port les signaux MLI habituels en modélisme à une périodicité de 20 ms. Comme ils occasionnent manifestement de grosses pointes de courant, quelques gros condensateurs électrolytiques servent de réservoir d'énergie. Raison pour laquelle l'ATmega, lui aussi, profite seul d'un autre régulateur de tension.

Le micrologiciel

Le micrologiciel a été rédigé en BASCOM AVR, il est centré sur la boucle principale dont la récurrence est de 100 Hz (cf. **figure 7**). Elle traite les données issues des capteurs du MPU6050 à travers le bus I²C et du filtre PID (proportionnelle, intégrale, dérivée) pour assurer l'équilibre. Le micrologiciel retient du signal de sortie du filtre PID les valeurs nécessaires à la commande des moteurs et les inscrit directement dans le registre de réglage de la MLI pour l'adapter à l'entraînement des moteurs. À l'arrêt (mode [Stop](#)), les paramètres PID sont modifiés pour maintenir le twinBot immobile. Il faut au préalable atteindre un état bien défini de l'angle et du décalage, pour lesquels l'ensemble du système est stabilisé. Cet angle dépend de différents facteurs, la masse de la tablette, son point de fixation, l'obliquité du sol, etc. On peut ajuster manuellement le décalage de l'angle. S'il est mal choisi, le twinBot en mode [Stop](#) aura tendance à réaliser de lents mouvements pendulaires d'équilibrage. Un outil d'étalonnage permet d'adapter le décalage de manière à ce que le twinBot s'équilibre sur place.

La commande des servomoteurs est une fonction auxiliaire de la boucle principale. Elle s'effectue au moyen d'impulsions traditionnelles d'une durée proportionnelle au débattement du servomoteur, entre 1 et 2 ms. La routine s'exécute une fois sur deux parcours de la boucle principale, ce qui fait environ toutes les 20 ms. Une autre fonction auxiliaire concerne la mesure de tension de batterie pour l'ajouter à une moyenne glissante. Après 50 exécutions de la boucle principale, donc à 2 Hz, on débranche momentanément l'alimentation des deux

ponts en H des moteurs. Durant cette courte pause, après disparition des composantes inductives, on prélève la valeur de la force contre-électromotrice suscitée par l'inertie des moteurs en rotation. Cette tension sert à déterminer leur vitesse instantanée. Malgré un calcul de la moyenne sur plusieurs mesures, la vitesse ainsi obtenue n'est pas très précise, mais largement suffisante pour empêcher le twinBot de dépasser le régime limite.

Vient alors la transmission par UART vers le module Bluetooth de la mesure de la tension de batterie. On vérifie ensuite si le tampon d'instructions a reçu une commande de conduite. Ce tampon a une capacité de 50 caractères, il est contrôlé par une petite routine d'interruption. Dès réception d'une commande de conduite par Bluetooth et UART, les caractères sont inscrits dans un tampon circulaire où ils attendent leur traitement. La

routine Parse Command s'assure alors que l'instruction suivante à exécuter est dans le tampon et la convertit en commande de conduite concrète ou modifie le mode de fonctionnement. Il existe trois modes de fonctionnement : *Park*, *Stop* et *Drive*. En mode *Park*, les béquilles sont sorties et le moteur désactivé. Il faut d'abord une commande *Stop* pour que le twinBot retrouve son équilibre. S'il reçoit alors une commande de mouvement, *forward*, *backward*, *left* ou *right*, les paramètres du régulateur PID sont adaptés et pendant un moment, l'angle de décalage est modifié dans le sens demandé. Le robot s'incline vers l'avant pour *forward*, en arrière pour *backward* et pour retrouver son équilibre, il démarre. L'accélération qui en résulte dépend de trois facteurs : primo, la hauteur et le poids de la tablette ; secundo, la différence d'angle et tertio, la friction entre les roues et le sol. La modification du décalage d'angle peut influencer directement les possibilités d'accélération du robot. On peut aussi régler ces paramètres avec l'outil d'étalonnage. Il est ainsi possible de prévoir un réglage plus doux de l'accélération dans le PID, principalement en mode *Stop*. Comme le twinBot ne peut pas accélérer sans cesse, ce qui entraînerait inévitablement sa chute, on s'arrange pour ramener à la normale le décalage d'angle après un certain nombre d'itérations de la boucle. Grâce à cela et à une modification du régulateur PID, le twinBot se retrouve quasi en roue libre aussi longtemps qu'il ne reçoit plus de nouvelle commande. Tant que le mode

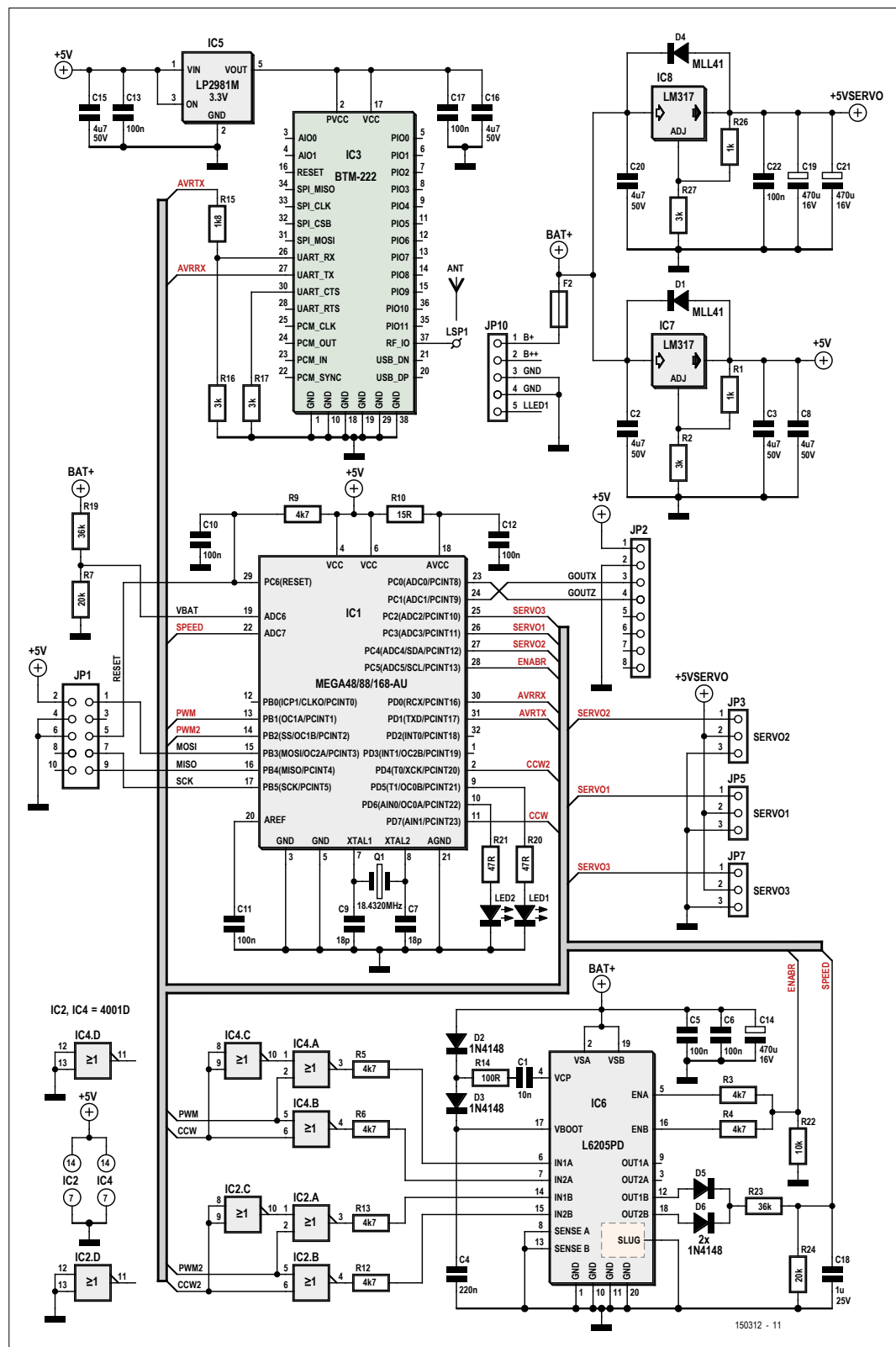


Figure 6. Le schéma du circuit de l'unité de commande du twinBot.

Drive est actif et que le robot reçoit régulièrement des commandes de conduite, il est maintenu dans un état entre l'accélération et le freinage. Un réglage doux des paramètres PID produit une allure sans à-coups.

En l'absence de toute nouvelle commande de conduite pendant une seconde, le twinBot passe automatiquement en mode **Stop**, dans lequel le régulateur PID est plus ferme. Il est alors possible de revenir en mode **Park**.

La radio

Pour permettre la conduite par l'internet, l'unité de commande du twinBot est en liaison par Bluetooth avec la tablette qui, elle, dispose d'une connexion par WLAN et exécute une appli passerelle. Elle extrait les commandes de conduite qui sont envoyées selon le standard « WebRTC » depuis l'appareil de commande vers le navigateur Firefox sur la tablette. Ces informations de conduite, l'appli les transmet alors à l'unité de commande du robot. Celle-ci communique de nouveau l'état de charge de la batterie et les données d'état à la tablette qui les répercute à l'appareil de commande au moyen de l'appli passerelle, sous forme de message WebRTC. Mais ce n'est pas tout, dans le navigateur de la tablette s'exécute une transmission vidéo et audio basée sur WebRTC. En voilà comment nous sommes arrivés à une téléprésence robotique complète, avec commande à distance !

WebRTC, communication en temps réel par web

Le socle WebRTC, basé sur HTML5 et JavaScript, permet assez facilement l'échange direct de données entre deux appareils connectés à l'internet. Avec lui, un navigateur peut non seulement récupérer des données sur un serveur, mais échanger des informations en temps réel avec d'autres navigateurs. WebRTC permet, avec un navigateur, de réaliser des vidéoconférences, de partager un bureau, de chatter et de transférer des fichiers. Les flux vidéo et audio peuvent ainsi être transmis directement entre navigateurs, sans passer par des serveurs de flux intermédiaires. Il ne faut pour cela qu'une collection de bibliothèques de programmes en JavaScript sur un site internet. Pour commander le twinBot, il faut installer le canevas EasyRTC (*framework*). Il se compose d'une API client, incorporée dans un site internet par JavaScript, qui met à disposition toutes les fonctions nécessaires pour se connecter à un serveur d'accès. Ce serveur met en relation les deux clients par un simple échange entre eux de numéros de port et d'adresses IP. Dès que cette liaison est établie, les deux clients peuvent échanger directement des données et des flux vidéo et audio en temps réel.

Le site internet *ad hoc* a été créé pour le twinBot, il permet de se connecter automatiquement à une banque de données et à un serveur d'accès. D'abord la tablette du twinBot se connecte au serveur qui lui fournit un identifiant de connexion (*socket ID*) unique, composé d'un numéro de canal et d'une adresse IP, indispensable à la poursuite de la communication. Il y a ensuite la connexion à la banque de données. Après vérification de la validité des données renseignées, l'identifiant attribué par le serveur est transmis à la banque de données. Si vous voulez commander le twinBot à partir d'un autre appareil terminal, il faut l'indiquer à la banque de données en passant par le site *twinrobotics* [1] pour obtenir l'identifiant actuel par lequel joindre le twinBot à ce moment-là. Une fois que l'appareil terminal s'est présenté et est reconnu par le serveur, il devient facile de contacter le twinBot directement au

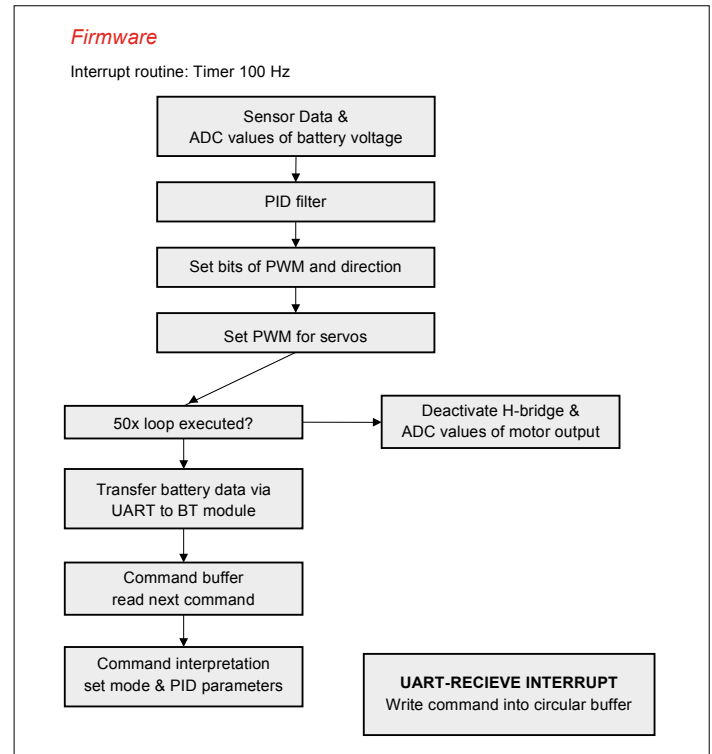


Figure 7. Le diagramme fonctionnel du micrologiciel.

moyen d'une fonction *call*. Après aboutissement du *call*, les deux clients, twinBot et appareil terminal, pourront échanger des données audio et vidéo en temps réel. La page web de l'appareil terminal contient une fonction JavaScript qui envoie des événements comme l'action sur une touche (*Keydown*) par la fonction WebRTC *SendMessage* directement à la tablette du twinBot. Donc quand on utilise le clavier de l'appareil terminal (par ex. les curseurs) pour naviguer, un message texte est immédiatement envoyé à la tablette. L'arrivée de ce message appelle une fonction JavaScript de la page web de la tablette. Les données de conduite sont alors transférées au moyen de l'appli passerelle à l'unité de commande du twinBot, reliée par Bluetooth à la tablette. L'échange de données va dans les deux sens, donc les informations d'état du twinBot, comme la tension de batterie ou l'angle d'inclinaison, peuvent parvenir à l'appareil qui le gouverne.

Le kit prêt à monter

Si vous avez des connaissances suffisantes en robotique, micro-contrôleurs et mécanique, les informations présentées dans cet article devraient vous aider à réaliser une version personnelle du robot mobile de téléprésence. À tous les autres qui ont moins d'appétence pour forer, scier et limer, Elektor propose sur la page de cet article [2] un kit avec tous les composants mécaniques et électroniques ainsi qu'une unité de commande montée et testée. Vous y trouverez également le logiciel si vous souhaitez y apporter des modifications personnelles. ◀

(150312 – version française : Robert Grignard)

Liens

[1] www.twinrobotics.eu

[2] www.elektormagazine.fr/150312

DÉCOUVRIR

CRÉER

PARTAGER

elektor•PCB Service

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



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

➡ **le prix !**

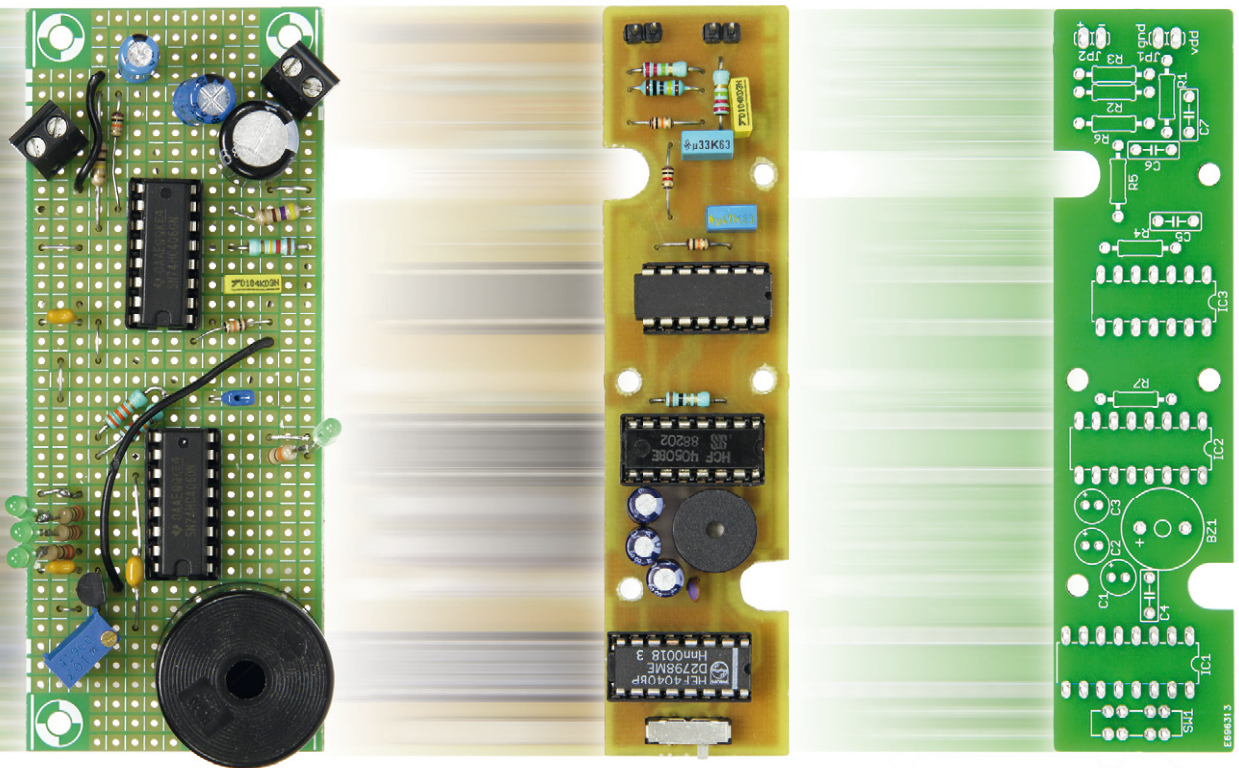
irrésistible...

➡ **la qualité !**

imbattable...

➡ **la fiabilité !**

inégalée...



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

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

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



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

www.elektorPCBservice.com

FLOWCODE⁷

Smarter programming

le logiciel qui fait avancer les choses

Liam Walton, Matrix Technology Solutions (Royaume-Uni)

L'activité principale de Matrix est la conception et la distribution d'outils de développement à base de microcontrôleurs. Comme toute entreprise, nous devons faire face aux évolutions rapides du marché actuel. Nous expliquons dans cet article la façon dont ces changements guident ou influencent le développement de notre produit phare : Flowcode.



Si les microcontrôleurs monopuce sont utilisés dans les produits industriels et grand public depuis une quarantaine d'années, leur volume de production est devenu si gros aujourd'hui qu'il est difficile de comprendre leur rapport caractéristiques/coût. On trouve ainsi des microcontrôleurs à 8 broches pour quelques centimes d'euros, et certains contrôleurs à 32 bits et 40 broches sont vendus à un prix à peine plus élevé que leurs homologues à 8 broches. Les coûts d'encapsulation des puces comptent plus ici que ceux du silicium depuis que le nombre de transistors par cm^2 s'est envolé.

Le prix de la gratuité

Conscients que les prochaines innovations viendront probablement du mouvement des *makers* et des amateurs, les fabricants de semi-conducteurs proposent désormais des kits de développement à 15 \$ ou moins. Chez Matrix, nous avons récemment acheté pour seulement 45 \$ un kit de développement ARM de ST équipé d'un LCD couleur de six pouces ; un module entièrement Bluetooth directement importé de Chine ne nous a coûté que 3 \$. Avec un marché intérieur en contraction et des frais de douanes nuls, les entreprises chinoises ont en effet mis

en place via eBay des canaux de vente directe aux consommateurs européens. En un certain sens, le matériel de conception ne coûte plus rien. (Comment la fondation RPi fait-elle pour produire et vendre une carte 5 \$?)

Côté outils de programmation gratuits, Arduino et RPi ont de la même façon placé la barre très haut. L'EDI Arduino par exemple est simple et efficace.

Comment cet état de fait nous a-t-il affectés ? Cet été, nous lançons la version 7 de Flowcode, un environnement de pro-

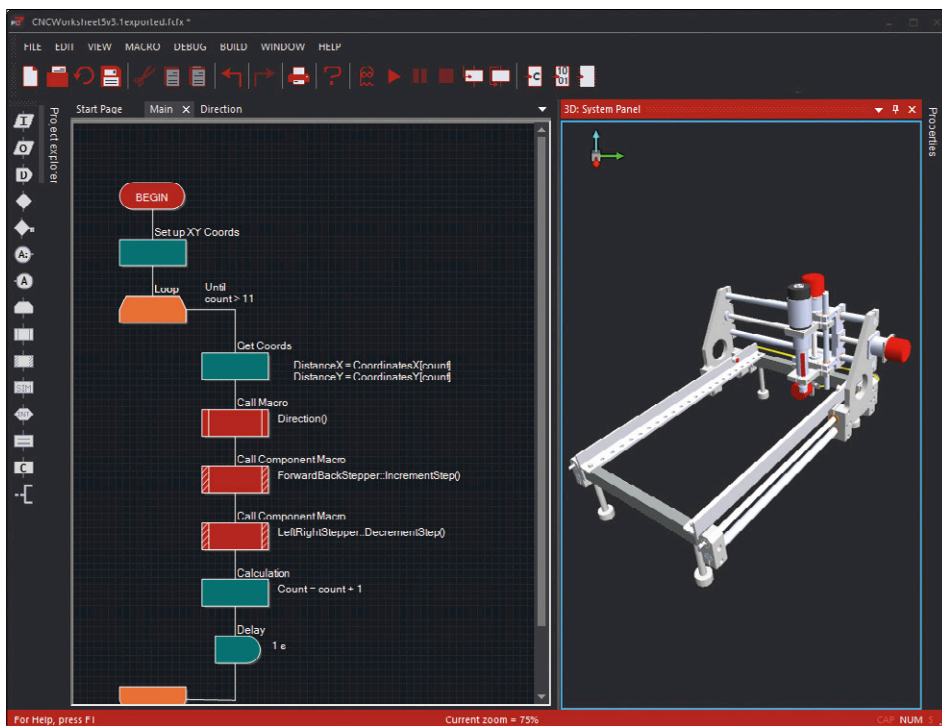


Figure 1. Bienvenue sous Flowcode 7 !

grammation graphique pour systèmes à microcontrôleurs (**fig. 1**). Notre logiciel ne s'est jamais aussi bien vendu, notamment parce que sa simplicité attire les utilisateurs des mouvements Arduino et Maker confrontés à la difficulté du langage C. Flowcode est compatible avec de nombreux microcontrôleurs auxquels la

v.7 ajoutera la famille des PIC32 à 32 bits de Microchip. L'utilisateur bénéficiera alors d'une incroyable puissance de traitement pour un coût supplémentaire modique. Tous les programmes écrits en Flowcode pourront être transférés directement sur cette plateforme et exploiter la puissance de traitement du processeur à 32 bits.

Nous continuerons bien sûr à vendre notre propre matériel, certes plus cher que celui des grands fabricants de semi-conducteurs puisque nos volumes sont moindres et que nous devons faire des bénéfices, mais notre technologie « Ghost », qui comprend un enregistreur de données, un oscilloscope, un analyseur logique, un décodeur de paquets et une console, accélère le débogage des systèmes : Flowcode et le matériel Matrix font gagner du temps. Matériel et outils de développement peuvent être gratuits, le temps a toujours un coût.

Ni électronique, ni mécanique : mécatronique et robotique

La robotique n'est plus réservée à l'assemblage des pièces automobiles et s'immisce petit à petit dans nos vies. On la retrouve par exemple dans un objet aussi anodin qu'une barrière de parking ou, pour prendre un exemple plus récent, dans les imprimantes 3D. Nous sommes (pour l'instant) loin des prouesses humanoïdes chères au cinéma, mais les progrès des imprimantes 3D affranchiront un jour les robots des problèmes mécaniques qui freinent leur évolution.

Sous l'influence peut-être du courant observé dans l'industrie, la grande tendance de ces dernières années en matière d'enseignement technologique aura été la fusion de deux disciplines majeures, le génie électrique et le génie mécanique. L'industrie a en effet besoin d'ingénieurs multidisciplinaires, l'ingénieur d'aujourd'hui est plus susceptible de passer son temps de travail avec une tablette entre les mains qu'avec un fer à souder ou une clé à molette. Un ingénieur apprend donc très tôt à programmer, en particulier des microcontrôleurs puisque ces puces seront au cœur de chaque système technique qu'il rencontrera.

Cette tendance a-t-elle guidé nos choix ? Outre la 2D, la version 6 de Flowcode incluait la simulation de systèmes en 3D, ce qui a troublé bon nombre d'utilisateurs qui se sont demandé : « Pourquoi aurais-je besoin de modéliser en 3D le comportement d'un contrôleur ? » La réponse est qu'un ingénieur ne conçoit pas un microcontrôleur, mais un système comportant un microcontrôleur. En donnant à un concepteur la possibilité de simuler à la fois la mécanique et l'électronique d'un système, nous facilitons

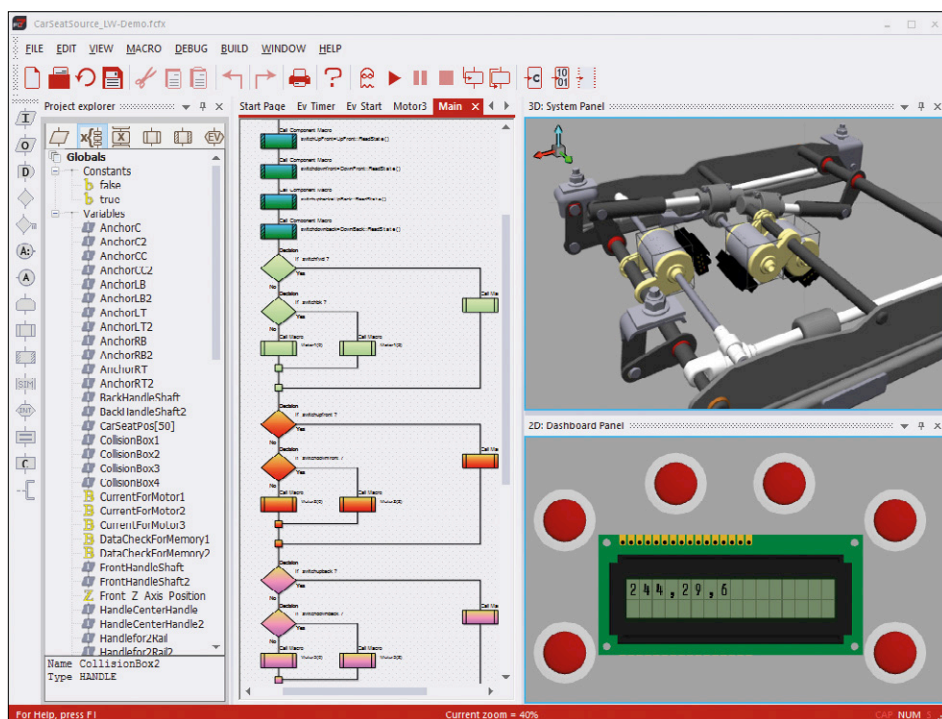


Figure 2. L'approche « Flowcode » pour la conception de l'électronique d'un siège automobile.

son travail et raccourcissons le cycle de conception de son produit – une économie de temps et d'argent. De la même façon, un étudiant qui peut simuler un système électromécanique ou chimique comprend mieux et assimile plus vite ce qu'il fait. Forts de ce constat, nous avons amélioré la simulation 3D de Flowcode 7 et ajouté une intégration à des programmes de CAO tiers comme Solidworks et DesignSpark Mechanical. La **figure 2** illustre la façon dont un ingénieur en automobile peut utiliser les applications et les exemples de Flowcode pour concevoir l'électronique d'un siège, et aussi apprendre le fonctionnement d'un microcontrôleur.

Technologie automobile

La prochaine mutation technologique à laquelle nous serons confrontés sera celle des systèmes d'aide à la conduite automobile. À nos yeux il s'agira même d'une révolution de la robotique. Les analystes de ce secteur prévoient que des véhicules autonomes circuleront sur les routes vers 2025 (au plus tard). De fait, les constructeurs automobiles disposent déjà – comme Google et d'autres – de prototypes opérationnels. Pour nous, la question intéressante est de savoir combien de clients utilisent déjà Flowcode pour concevoir des technologies propres à l'automobile. Non pas l'UCE (unité de commande électronique), mais des systèmes comme les afficheurs/écrans des véhicules électriques, les habitacles des camping-cars, les instrumentations de mesure, etc. Autrement dit : les systèmes constitués de microcontrôleurs en réseau – qui utilisent un bus CAN comme moyen de communication.

Comme précédemment, le point clé ici est qu'il s'agit de systèmes à plusieurs microcontrôleurs, par opposition à ce que seraient de simples circuits à microcontrôleur. Flowcode 7 offre de très nombreuses fonctions pour la conception de tels systèmes. Il est ainsi possible de simuler plus d'un microcontrôleur à la fois – jusqu'à dix instances de Flowcode exécutées sur un PC ou un réseau de PC peuvent communiquer entre elles pour simuler un système. Les données des protocoles peuvent en outre provenir d'un tableau : vous pouvez écrire un programme, avoir de nombreux microcontrôleurs, mais les données dictant le comportement de chacun d'entre eux peuvent être externes. Cela facilite grandement la conception

de ces systèmes. Le composant « Injecteur » de Flowcode 7 rend lui aussi très faciles les tests et débogages : on peut simuler les signaux reçus par un système et s'assurer que le programme y répond correctement.

Conclusion

Nous n'avons examiné ici que trois des changements technologiques auxquels nous avons répondu. Nous aurions aussi pu citer :

- L'expansion croissante de l'Internet des Objets : Flowcode 7 prend en charge les systèmes de communication Bluetooth et Wi-Fi/Internet.
- L'utilisation généralisée des tablettes/PC à écran tactile : les icônes de Flowcode sont plus grandes, donc aussi adaptées aux « gros » doigts.
- L'exploitation de l'espace d'affichage disponible : Flowcode 7 minimise la place que prend un système à l'écran.

- Les modèles Microchip : l'utilisateur peut aisément programmer les modèles préconçus des kits de développement de Microchip, dont les cartes *PICkit 3 Low Pin Count Demo Board* et *PICkit 44-Pin Demo Board*.

Bien d'autres nouveautés faciliteront le travail du concepteur : de nouveaux compilateurs rapides XC de Microchip (versions PIC), un profilage de code (**fig. 3**) qui affiche graphiquement la fréquence d'accès aux icônes durant la simulation et le débogage, une aide hors-ligne, une nouvelle interface utilisateur graphique, et plus encore.

Depuis fin juin 2016, Flowcode 7 est disponible dans l'e-choppe (www.elektor.fr/matrix). ◀

(160050-I – version française : Hervé Moreau)

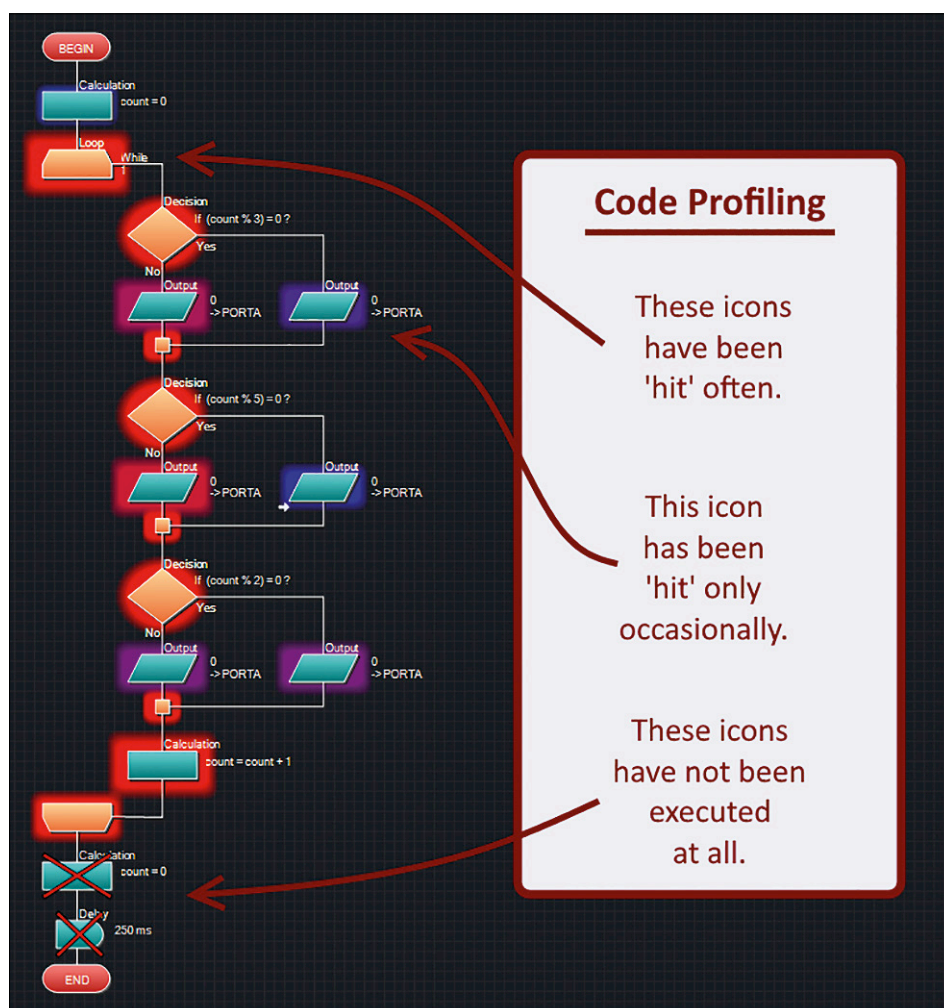


Figure 3. Nouveauté Flowcode 7 : le profilage de code.



electronica

FAST FORWARD START-UP AWARD

LES IDÉES NOVATRICES, VOUS LES AVEZ

powered by



VOICI NOS CONTACTS PROFESSIONNELS !

Nous sommes à la recherche des idées, des projets ainsi que des nouveaux entrepreneurs les plus prometteurs, et ce dans le monde entier. Pour la première fois, le concours « electronica fast forward Start-up Award powered by Elektor » tisse des liens entre le marché mondial de l'électronique, les technologies novatrices et une présence médiatique internationale ; c'est la garantie que votre idée révolutionnaire marquera les esprits des experts.

Les jeunes entreprises et les concepteurs créatifs du monde entier sont appelés à soumettre leur idée sur notre site web www.elektor.com/electronica-startup-award, avant le 15 juillet 2016.

Un jury composé de rédacteurs du magazine Elektor et d'ingénieurs du laboratoire d'Elektor examinera et évaluera les envois. Les meilleurs des idées, projets et jeunes entreprises devront participer à la finale qui aura lieu au salon *electronica 2016* – le salon international des composants, systèmes et applications électroniques qui a lieu à Munich (Allemagne).

Si vous concourez dans l'une des catégories « Idée », « Prototype » et « Start-up », vous aurez une occasion unique d'être membre de la plateforme « electronica fast forward Start-up powered by Elektor » et de nouer des contacts dans le monde entier.

Sur le salon *electronica*, vous bénéficierez de conseils personnalisés et de mises en relation, et vous participerez à de nombreux événements. Vous disposerez d'un stand pour présenter votre idée aux visiteurs de

tous horizons.

Parmi les finalistes présents sur le salon, un jury international choisira trois gagnants. Soyez un des trois lauréats du prix « electronica fast forward Start-up Award » qui sera remis le 11 novembre 2016 ! Les prix prendront la forme de services internationaux en matière de relations publiques, consulting et marketing, pour une valeur totale de 150 000 € ainsi que d'un accès au réseau international d'Elektor, qui compte plus de 250 000 membres.

Le vainqueur toutes catégories confondues sera récompensé par une campagne de marketing transversale et internationale, d'une valeur de 75 000 €, à laquelle s'ajoute un stand sur le salon *electronica 2018*. Sur la deuxième marche du podium, on trouve une campagne médiatique (Elektor) d'une valeur de 50 000 € et enfin sur la dernière marche une présence médiatique (Elektor) d'une valeur de 25 000 €.

Profitez du salon *electronica* pour nous soumettre vos projets, nous pourrions vous conseiller pour votre plan de développement et vous permettre de présenter votre idée à des partenaires stratégiques internationaux ainsi qu'à de futurs clients.

Avons-nous éveillé votre curiosité ? Pour trouver des informations sur les conditions de participation et le dossier de candidature, rendez-vous sur

www.elektor.com/electronica-startup-award

Kit de démarrage Intelligent IoT Gateway

ADLINK a conçu une solution de connexion inter-machines (M2M) en temps réel, basée sur MXE-202i, une passerelle intelligente qui repose sur la technologie *IoT Gateway* d'Intel® et sur la plateforme Vortex DDS (*Data Distribution Services*). La capture d'un mouvement active un capteur relié au MXE-202i, qui traite alors les données et les distribue aux dispositifs robotiques via Vortex DDS. Le MXE-200i est un système robuste et intelligent, facile à implanter et adapté à un large spectre d'applications, dont les automatismes industriels ; la plateforme



Vortex de partage intelligent des données permet de distribuer facilement des données entre machines, et entre machines et nuage.

Le kit associé *IoT Gateway Starter* fournit pour chaque environnement d'application une solution complète de connexion qui réduit les temps de conception et de déploiement. Il combine la passerelle MXE-202i pour l'IdO, l'application *Edge-Pro* de gestion des capteurs & dispositifs IdO, un capteur optique, une sortie sirène, un module TCP Modbus, et divers accessoires. L'application *EdgePro* et le kit *IoT Gateway Starter* bénéficient en outre d'une interface d'administration simple et de tableaux de bord de télésurveillance.

www.adlinktech.com (150026-1)



Le chapeau blanc du prince Philip

Le Musée national de l'informatique de Milton Keynes (Angleterre) a ajouté à ses archives l'histoire du piratage de la boîte aux lettres du prince Philip qui conduisit à l'adoption du *Computer Misuse Act 1990* (Loi de 1990 sur l'usage abusif de l'informatique).

Robert Schifreen faisait partie de ces « chapeaux blancs » (hackers dénués d'intentions malveillantes) qui dans les années 1980 collectionnaient les noms et mots de passe d'utilisateurs en s'introduisant dans des bases de données privées. C'est ainsi qu'en 1985 il parvint à pénétrer les services en ligne de Prestel (le service pré-internet de British Telecom) et à accéder aux courriels du prince Philip. Il avertit cette société, mais elle le fit arrêter.

Comme en ce temps-là il n'existait aucune loi relative au piratage informatique, Schifreen fut jugé pour tromperie envers un non-humain.

Au terme d'une longue bataille juridique étalée sur deux ans et trois procès, il fut finalement acquitté par la Chambre des Lords et l'affaire fit jurisprudence.

Lors de la présentation de l'archive au grand public, Schifreen a rappelé qu'Internet n'existait pas en 1985 et que l'informatique domestique commençait seulement à se répandre. Le fournisseur de services en ligne Prestel avait quelques années d'existence, mais la sécurité informatique n'était pas encore une priorité.

Comme il le raconte : « À cette époque, les hackers ne commençaient jamais avant 18h, car avant cela la connexion coûtait trop cher. Mais surtout, après 18h, l'équipe de sécurité de Prestel n'était plus là pour traiter les messages automatiques qui les auraient avertis qu'il y avait eu trois tentatives infructueuses d'accès à leurs services. Je pouvais donc lire ces mes-



sages et les supprimer pour effacer mes traces. J'avais les mains aussi libres qu'un administrateur système de Prestel. Je suis même parvenu à accéder aux courriels du prince Philip. J'ai téléphoné à Prestel pour leur dire ce que je pouvais faire en croyant qu'ils me proposeraient un poste, mais ils ne l'ont pas fait. À la place, ils ont appelé Scotland Yard. »

www.tnmoc.org (160026-4)

Jusqu'à 5 A pour ce module de puissance ultraplat

Placé dans un fin boîtier de 10 x 6 x 2 mm, le MIC45404 de Microchip est un convertisseur abaisseur de type point de charge conçu pour les applications de conversion de puissance à fortes contraintes d'espace (serveurs, routeurs, disques SSD...)

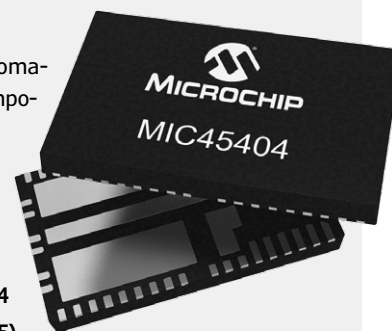
Il comprend un contrôleur, des MOSFET, un circuit de rétroaction et un régulateur à découpage MLI. Le MIC45404 contient aussi une inductance très mince, un condensateur autoélévateur et un condensateur d'entrée haute fréquence montés sur une grille de connexion aux propriétés thermiques améliorées. Ceci simplifie la conception des cartes

et réduit les perturbations électromagnétiques imprévues dues aux composants passifs externes.

Pour la conception d'applications avec le MIC45404, Microchip propose le kit d'évaluation MIC45404YMP-EV.

www.microchip.com/EUMIC45404

(160026-5)



PMIC universel à trois canaux pour FPGA et SoC basse consommation

Le XR77103 d'Exar est un PMIC universel intégrant trois étages de puissance à MOSFET synchrones. Ce circuit intégré en boîtier de 4 x 4 mm fournit une solution de gestion de puissance facile à utiliser avec de nombreux FPGA, SoC, processeurs vidéo et processeurs de traitement du signal.

L'interface I²C permet de commander la tension de sortie (0,8 à 6 V), la fréquence de commutation (300 kHz à 2,2 MHz), la mise sous/hors tension ainsi que la limite de courant. Le XR77103 est pris en charge par une nouvelle

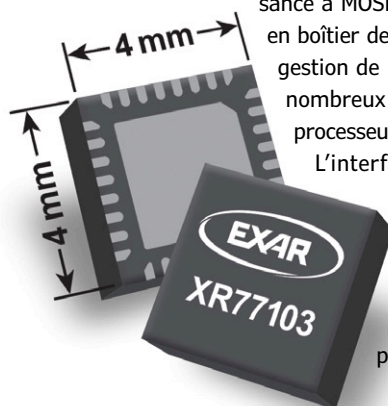
version du logiciel de conception et configuration PowerArchitect™ 4.

Le XR77103 peut être alimenté entre 4,5 V et 14 V. Ses trois sorties acceptent des charges de 2 A avec des valeurs de crête jusqu'à 3 A. Puisqu'il se commande en mode courant, ses sorties peuvent être mises en parallèle et ainsi délivrer jusqu'à 5 A ; le XR77103 peut donc alimenter de nombreux processeurs à basse consommation. Le mode *Pulse Skipping* (PSM) réduit par ailleurs la puissance absorbée par le système lorsque le processeur hôte est en veille.

Les XR77103ELB, XR77103ELB-A0R5 et XR77103ELB-A1R0 sont conformes à la directive RoHS, sans halogène et en boîtiers QFN de 4 x 4 mm.

www.exar.com/products/power-management/universal-pmics

(160026-3)



batteries avant-gardistes

l'avenir est en charge

Universités et instituts de recherche du monde entier enchaînent à qui mieux mieux les communiqués de presse révélant quelque percée dans le domaine des batteries. De leur côté, entreprises de technologie et constructeurs automobiles y consacrent de gros budgets. Alors, pourquoi encore et toujours les lithium-ion ?

Compilation de **Jan Buiting**, rédaction en chef anglaise



Ne vous inquiétez pas, un jour innovation finira bien par rimer avec révolution, peut-être même plus tôt que vous ne le pensez. D'où viendra la percée ? Peut-

être d'une des technologies présentées ici, peut-être d'une équipe que personne n'attendait. Gardez cependant à l'esprit que certaines des batteries décrites ici

sont pure prospective. Quelques-unes ont déjà été mises en vedette dans notre e-zine.



Écologique : Bioo, chargeur à plante verte

Le chargeur Bioo (**fig. 1**) pour ordiphones et tablettes est déjà commercialisé [1]. Il se présente sous la forme d'une plante en pot dont les produits de la photosynthèse sont exploités dans une cellule hybride électrochimique/biologique pour produire un courant permettant jusqu'à trois charges par jour sous 3,5 V et 0,5 A.

Le port USB est camouflé dans un caillou. Rêvons un peu : une version « Bioo Forest » pourrait alimenter des villes entières avec de l'énergie 100 % verte, et fournirait *de facto* la meilleure des raisons pour protéger nos arbres et nos plantes !

Figure 1. Bioo, le chargeur qui exploite la photosynthèse.

Dorées : batteries à nanofils

Conçues à l'université de Californie d'Irvine, les batteries à nanofils pourraient bien être les premières batteries inusables.

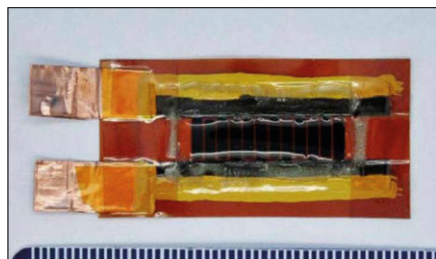
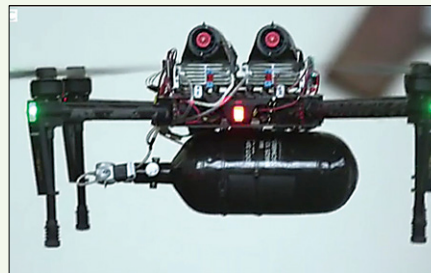
Un millier de fois plus fins qu'un cheveu humain, les nanofils ont longtemps semblé une voie prometteuse pour les batteries, mais se rompaient inévitablement à force de cycles de charge. Des essais ont montré qu'ils restaient intègres après plus de 200 000 charges lorsqu'ils étaient enrobés d'un gel électrolytique. Des batteries idéales pour les futurs véhicules électriques, engins spatiaux et ordiphones.

Légères : piles à combustible

Plus résistante et durable qu'une lithium-ion, une pile à combustible légère donnerait une autonomie d'une semaine à un ordiphone et de plus d'une heure à un drone (**fig. 2**). Elle est constituée d'électrodes à faible capacité calorifique et d'inox poreux recouvert d'un film d'électrolyte.

Ordiphones, drones, voire véhicules électriques, devraient bientôt en bénéficier. La forte implication des Sud-Coréens laisse même supposer son emploi dans le prochain Samsung Galaxy S8.

Figure 2. Drone expérimental alimenté par une pile à combustible. (BBC)



Fabriquées par laser : batteries à micro-supercondensateurs

La fabrication des micro-supercondensateurs reste coûteuse, mais cela pourrait changer à tout moment. Des chercheurs de l'université de Rice sont en effet parvenus à graver des motifs d'électrode dans des feuilles de plastique au moyen d'un laser, une méthode de fabrication peu chère et facile à mettre en œuvre (**fig. 3**). Mécaniquement très résistante, la batterie obtenue se charge 50 fois plus vite que les batteries actuelles et se décharge plus lentement que les supercondensateurs classiques.

Figure 3. Prototype du micro-supercondensateur. (Rice University)

Mousseuses : batteries à mousse de cuivre

La société *Prieto Battery* a été la première à concevoir une batterie « 3D » utilisant un substrat à base de mousse de cuivre (**fig. 5**).

Plus petites et plus sûres que les Li-ion classiques puisque l'électrolyte est ininflammable, ces batteries ont aussi une durée de vie plus longue, un temps de charge plus rapide, une densité cinq fois plus élevée et un coût de fabrication plus faible. Prieto vise le créneau de l'informatique vestimentaire avant le possible marché des ordiphones et de l'automobile.

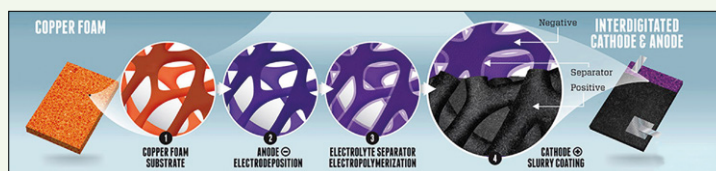


Figure 5. Constitution de l'anode et de la cathode de la batterie Prieto. (Prieto)

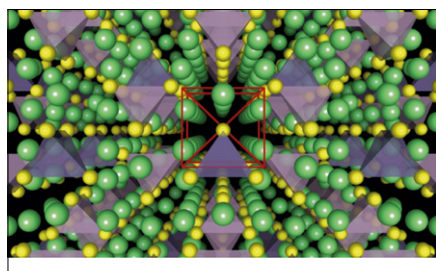


Figure 6. Aucun risque de fuite avec les batteries à électrolyte solide. (MIT)

Ininflammables : batteries à électrolyte solide

En collaboration avec Samsung, des chercheurs du MIT ont mis au point une batterie à électrolyte à état solide plus sûre, plus puissante et de plus longue durée de vie que les Li-ion utilisant un électrolyte liquide pour le transfert des charges d'une électrode à l'autre. C'est ce liquide qui est potentiellement inflammable et limite la durée de vie de la batterie (**fig. 6**).

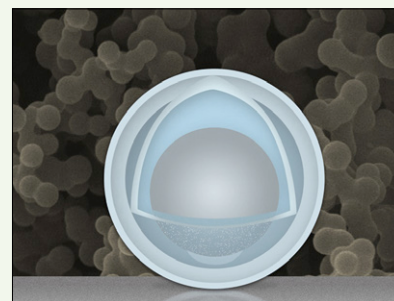
Selon le MIT, une batterie à électrolyte solide pourrait subir des centaines de milliers de cycles de charge avant de se dégrader. Sa densité d'énergie est de 20 à 30 % supérieure à celle des Li-ion, autrement dit elle fournit plus de charge à encombrement équivalent. Et puisqu'elle est ininflammable, elle est idéale pour les véhicules électriques.

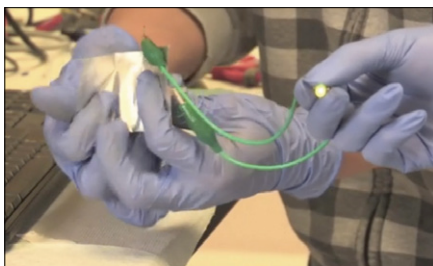
Inspirée de l'œuf mollet : prête en six minutes

C'est à nouveau au MIT que l'on doit cette batterie d'une capacité de 1,2 Ah/g qui offre un temps de charge de seulement 6 min, à peu près la durée de concentration maximale d'un ado. L'électrode est constituée de nanoparticules à coquille de dioxyde de titane renfermant un « jaune d'œuf » (dixit le MIT, **fig. 7**) en aluminium.

Parce qu'elles peuvent se dilater et se contracter, ces nanoparticules améliorent la durée de vie de la batterie. Elles sont par ailleurs faciles et peu coûteuses à fabriquer à échelle industrielle.

Figure 7. Le « jaune » (sphère) et la « coquille » (en bleu clair) des nanoparticules. (MIT)





Ultra-rapide : batteries à aluminium-graphite

Sans doute par rivalité avec le MIT, des scientifiques de l'université Stanford ont mis au point une batterie à aluminium (anode) et graphite (cathode) capable de charger un ordiphone en 1 min. Elle est contenue dans une poche en polymère souple (**fig. 8**), est endurante et se recharge en un temps record.

Son défaut est de ne produire que 2 V, moitié moins qu'une Li-ion actuelle. Mais est-ce vraiment un problème quand on peut recharger un téléphone en 1 min ?

Figure 8. Seulement deux de tension, mais véloce pour ce qui est de la charge. (Stanford University)

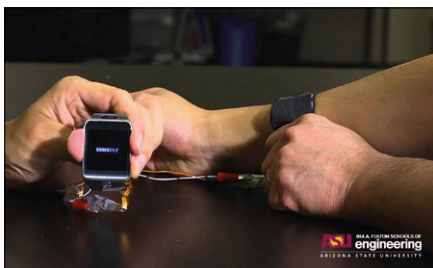
À l'eau pendant 14 jours : la batterie Alfa

La batterie aluminium-air Alfa conçue par Fuji Pigment se recharge par simple remplissage d'eau, salée ou ordinaire. Sa capacité est 40 fois supérieure à celle des Li-ion et l'accumulateur tient 14 jours sans recharge.

Fuji espère voir sa technologie adoptée en premier lieu par l'industrie automobile. Avant de l'être dans nos appareils portables ?

La capacité de 8,1 kWh/kg de cette Al-air surpasse les 0,12 à 0,2 kWh/kg des Li-ion, mais on est encore en deçà des 11,4 kWh/kg des Li-air.

Figure 9. La batterie Alfa exploite la technologie aluminium-air. (Fuji Pigment)



Élastiques : batteries souples

Une équipe de l'université d'État de l'Arizona s'est inspirée des principes du *kirigami* (l'art japonais du pliage et découpage du papier) pour concevoir un bracelet-chargeur alimentant une montre Samsung Gear 2 (**fig. 10**).

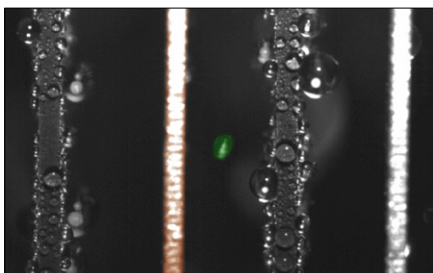
Des batteries souples permettraient de construire des montres connectées plus petites, plus fines et de plus grande autonomie, mais on peut aussi s'attendre à les retrouver bientôt au service de l'informatique vestimentaire.

Figure 10. Le prototype du bracelet-chargeur.

Froissables comme du papier : batteries J.Flex

La batterie J.Flex du coréen Jenax peut être pliée, froissée ou roulée sur elle-même (**fig. 11**). Ajoutez à cela sa résistance à l'eau, et vous avez une candidate idéale pour l'informatique vestimentaire. Et pourquoi pas l'imaginer dans des écrans souples que nous pourrions rouler et mettre dans nos poches ! La batterie a subi divers tests de sécurité, de résistance et de fiabilité. Elle a notamment été pliée plus de 200 000 fois sans subir de baisse de performance.

Figure 11. La J.Flex peut être utilisée dans toutes sortes de gadgets pliables.



Poétiques : batteries exploitant la rosée

Créé par le MIT, futuriste, expérimental, ce dispositif constitué de plaques super-hydrophobes produit de l'énergie à partir de la condensation de l'humidité de l'air (**fig. 12**). Il ne délivre pour l'instant que quinze picowatts, mais sera modifié pour produire au moins un microwatt. C'est encore peu, mais si le temps de charge n'est pas un critère essentiel, et avec des conditions d'humidité favorables, il peut charger un téléphone en 12 h si on lui donne les dimensions d'une boîte réfrigérante.

Figure 12. Un téléphone alimenté par l'humidité pourrait un jour devenir réalité.

Prodigues : batteries aluminium-air (autonomie de 1 800 km)

La société Phinergy a prouvé qu'une petite voiture (**fig. 13**) équipée de sa batterie aluminium-air pouvait rouler 1 800 km sans recharge. Ce type de batterie utilise l'oxygène de l'air pour la réaction de réduction à la cathode, ce qui la rend beaucoup plus légère qu'une Li-ion à électrolyte liquide.

« Petit inconvénient » : les batteries Al-air doivent être changées régulièrement puisqu'elles sont non-rechargeables. En contrepartie elles sont plus légères et moins coûteuses que les Li-ion actuelles. L'alumine qui recouvre l'anode usée est par ailleurs recyclable.



Figure 13. Fin d'un parcours de 1 800 km sur une seule charge. Un record. Le hic : maintenant, pour rentrer, il va falloir changer la batterie.



Fidèles à la loi de Lavoisier : batteries à l'urine

Financée par le généreux canal de la Fondation Bill Gates et tirant son énergie du canal de l'urètre, la batterie *Urine-tricity* du *Bristol Robotics Laboratory* fonctionne à l'urine pure (*sic*) et délivre suffisamment de courant pour charger un ordiphone (**fig. 14**). Il s'agit en fait d'une pile à combustible microbienne. Une version améliorée, *Urine-tricity++*, produit également de l'eau potable et un fertilisant.

Figure 14. Sale boulot pour les micro-organismes de cette pile à combustible : transformer l'urine en électricité.

À l'odeur de rhubarbe : batteries organiques

En utilisant des quinones (des composés aromatiques que l'on trouve dans certaines plantes comme la rhubarbe), des chercheurs de l'université d'Harvard ont mis au point une batterie à flux (c.-à-d. à circulation d'électrolytes) « organique » (**fig. 15**). Elle ne coûte que 27 \$ par kWh, à comparer aux 700 \$ par kWh d'une batterie utilisant des ions métalliques. Elle permettrait un stockage d'énergie à grande échelle, et peut donc être ajoutée à liste des sources d'énergie *réellement* durables.

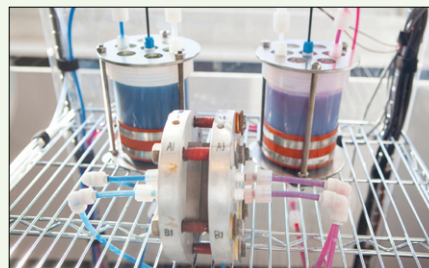


Figure 15. Les quinones utilisées ici sont extraites de la rhubarbe ; elles servent aussi de laxatif et d'agent antibactérien.



Repassez-moi le sel svp : batteries sodium-ion (2)

La recherche d'une batterie au sodium-ion efficace remonte aux années 1980, époque à laquelle les constructeurs cherchaient un substitut au lithium. Contrairement à ce dernier, élément rare et cher, le chlorure de sodium est l'un des minéraux les plus abondants de notre planète et ne subit pas les aléas d'approvisionnement du lithium. Certains e-prophètes prédisent d'ailleurs que la forte demande en lithium fera grimper son prix au point qu'il deviendra économiquement impossible de l'utiliser dans les batteries des véhicules électriques. La commercialisation des batteries Na-ion devrait décoller d'ici 5 à 10 ans. Résultat prometteur, des chercheurs japonais ont mis au point une batterie Na-ion jusqu'à sept fois plus efficace que les batteries conventionnelles (**fig. 16**).

Figure 16. Le vert de la croissance pour cette vue d'artiste d'une batterie Na-ion.

Conclusion

Les batteries présentées ici ne représentent qu'un petit échantillon d'une recherche très active. Si certains produits ont été jetés dans le grand bain du commerce, cela ne veut pas dire pour autant qu'ils aient été massivement adoptés. Il est par exemple peu probable que vous trouviez un char-

geur Bioo public dans un aéroport. Secteur et batteries Li-ion règneront probablement encore un certain temps.

Les nouveautés en matière de batteries intéressent aussi le grand public, ce qui explique sans doute que l'on tombe souvent sur des articles décrivant des sources d'énergie parfois inattendues comme les

pommes de terre, la peau humaine, les muscles, le bruit ambiant, les oignons, les peptides, l'activité cérébrale...

(160013 - version française : Hervé Moreau)

Lien

[1] Bioo : <http://www.bioo.tech/>

IDENTIFICATION SERVICES DEVENIR MEMBRE

eLektor DÉCOUVRIRE CÉLÈBRE PARTAGER

DEPUIS 30 ANS NOUS SOMMES LÀ POUR VOUS ACCOMPAGNER
Découvrez ce qui fait de RS un partenaire unique.

En savoir plus » fr.rs-online.com

NOUVELLES MAGAZINES ELEKTOR LABS ARTICLES SELECT PAGES LETTRE D'INFORMATION

Home > Labs

Chercher un projet ou tag

Créer un projet

Sujets populaires : Raspberry Pi Arduino LED T-Board AVR USB temps PIC horloge

GPS DISTANCING - A GREAT AID FOR THE PIPE LINE PATROLLERS
par barto
GPS Distancing is the project which took birth out of compulsion. For taking long readings of our pipe lines this device is just unparallel !
38 Vues | 0 Commentaires

GPS ASSISTED 10 MHZ FREQUENCY REFERENCE
par brend
A GPS assisted 10 MHz frequency reference with 0.002Hz accuracy controlled by an Arduino Nano
188 Vues | 0 Commentaires

RF CIRCUIT
par Kiole
This radio listening device project is an example of what is not allowed on Elektor Labs.
1558 Vues | 4 Commentaires

OTA-OVERDRIVE-2 (NONE-OPAMPS)
par jockazo
Next step on Guitar-Effects and really vintage - again quite experimental and guaranteed boombastic - I finally call it the "Dark Flame Distortion" !
46 Vues | 0 Commentaires

SIMULATOR PT100
par michael
This is an simulator for a resistive platinum probe PT100 for temperature.
253 Vues | 0 Commentaires

TEMPERATURE CONVERTER 0 - 100°C TO 0 - 10V
par michael
Modular board for a temperature converter from a resistive platinum probe PT100 from 0 to 100°C to a tension from 0 to 10V
278 Vues | 0 Commentaires

RUDIMENTARY ELECTRONIC MUSICAL INSTRUMENT AND...
par MJB
Introducing the "REM" - a musical wind instrument with touch sensitive pads. A novel fingering scheme makes it easy to play.
679 Vues | 8 Commentaires

BATHROOM SMART DOOR
par ando
A nice way to protect your and your guests' privacy with a red led that flashes when the door is closed!
479 Vues | 3 Commentaires

SCROLLING LED MARQUEE
par ClementVatons
180 LEDs controlled by a single PIC 18F46K22 clocked at 64 MHz. The program is written in C. Open source, open hardware: Use and improve.
822 Vues | 0 Commentaires

T-BOARD ADAPTER
par Iain Gano
A T-Board for training students. I designed this T-Board because of the lack of useful terminals on the breadboard...
2145 Vues | 2 Commentaires

MODULARES BOARD MIT ATMEGA1284P IN C
par schin S.
Modulares Board mit vielen Erweiterungen und einheitlichen Verbindungen und Grossen Modular Board with many extensions and.
388 Vues | 0 Commentaires

MONITOR AND DATA LOGGER FOR A HEATING BOILER
par walter
Measure and log 6 parameters of a heating boiler. Works as client/server, communicates via WLAN. ESP8266 plus 429 Dinco board.
485 Vues | 0 Commentaires

ETAT DU PROJET
☒ TOUT
☐ PROPOSITION (508)
☐ EN COURS (45)
☐ TERMINÉ (224)

DATE DE PUBLICATION
☒ TOUT
☐ CETTE SEMAINE (1)
☐ CE MOIS-CI (10)
☐ CETTE ANNÉE (149)
☐ PLUS D'UN AN (777)

1 2 3 4 5 6 7 8 ... 64 65 »

SERVICES
Conditions générales
Copyright
Contactez-nous
Publicité

ELEKTOR LINKS
Elektor MAGAZINE
Elektor LABS
Elektor STORE

INFOS
f g+ t u

PAIEMENT SÉCURISÉ
VISA MasterCard PayPal

INSCRIPTION À NOTRE LETTRE D'INFORMATION
Adresse e-mail
Votre prénom
Votre nom
INSCRIVEZ-VOUS



Ouvert à tous et à toute heure

N'importe quel visiteur du labo d'Elektor en ligne (Elektor.Labs) peut non seulement lire les descriptions des projets, mais également y participer, que ce soit pour les commenter ou en télécharger les fichiers ; le visiteur peut même créer un nouveau projet. Il lui suffit de se connecter avec son ID Elektor (gratuit). Ce compte donne accès au labo, mais pas aux archives du magazine et autres services réservés aux membres Green et Gold.

Un seul point d'entrée, un seul identifiant

L'ID Elektor (gratuit), une combinaison unique de courriel et de mot de passe, établit le lien entre votre compte *Elektor.Labs* et vos autres comptes Elektor. Une fois identifié, vous avez accès à tous les services liés à votre ID. Cliquez sur l'onglet **Mes projets LABS** pour afficher la liste des projets que vous avez créés et/ou que vous suivez. De là vous pouvez ouvrir un projet, que ce soit pour le voir ou le modifier.

Modification d'un projet

Chaque mise à jour doit maintenant être accompagnée d'un commentaire pour que tous les participants à un projet soient au courant. La page principale d'un projet (**Description**) ne peut être modifiée qu'en mode brouillon et n'est plus modifiable une



fois le projet publié. Lorsque le propriétaire d'un projet ajoute des commentaires, ceux-ci sont placés par ordre chronologique, directement sous la description du projet et au-dessus des commentaires des visiteurs et suiveurs.

Création d'un nouveau projet

... c'est très simple. Pour créer un projet, cliquez sur le bouton rouge **Créer un projet** (en haut de la page d'accueil d'*Elektor.Labs*). Attention : votre profil de membre doit être complet (avec pictogramme et pseudo). Entrez ensuite un titre (explique !), un commentaire et une description du projet. Le commentaire est votre « bande-annonce », celle qui apparaîtra sur les pages montrant une vue d'ensemble des projets, donc soyez clair et concis (pas plus de 150 caractères). L'ajout d'une photo du projet est fortement recommandé, mais si vous n'en avez pas vous pourrez l'insérer plus tard. Le bouton **Pièces jointes** permet de téléverser des fichiers, photos, codes, etc. Vous pouvez leur attribuer un type de façon à ce qu'ils s'affichent dans certaines zones de la page.

Brouillons

Les projets en mode *brouillon* ne sont pas visibles du reste du monde et peuvent être modifiés indéfiniment ou supprimés. Le mode *brouillon* est utile si vous souhaitez présenter un projet terminé en un seul « post ». Relisez et polissez votre description jusqu'à en être satisfait, puis publiez-la. Attention, il n'y a pas de retour en arrière possible ; un projet publié est immédiatement visible du monde entier.

À la recherche du projet perdu

La richesse d'*Elektor.Labs* est formidable, mais le site perdrait tout son intérêt sans une bonne boussole pour s'y diriger. C'est pourquoi nous avons ajouté plusieurs outils et filtres de recherche. Des mots-clés bien choisis et plusieurs options de tri vous permettront de trouver rapidement ce que vous cherchez. ◀

(150818)

www.elektormagazine.fr/labs

- DÉTAILS DE MON COMPTE
- MES COORDONNÉES
- MES ABONNEMENTS
- MES PROJETS LABS
- MES PRÉFÉRENCES NEWSLETTER
- DÉCONNEXION

MES PROJETS LABS

Nom du projet	Etat	Rôle	Commentaires	Suiveurs	Actions
Elektor mbed interface...	En cours	Chef de projet	0	2	Ouvrir Modifier
Elektorino Uno R4 [150790]	En cours	Chef de projet	0	13	Ouvrir Modifier
Analog circuit design with...	Terminé	Chef de projet	0	2	Ouvrir Modifier
Webinar 22/10: IoT Security ...	Terminé	Chef de projet	0	1	Ouvrir Modifier
Elektor Intel Edison Challenge	Terminé	Chef de projet	0	2	Ouvrir Modifier
Guitar BoB [150382]	En cours	Chef de projet	0	2	Ouvrir Modifier
CoCo-ri-Co! Guitar Tuner...	En cours	Chef de projet	0	1	Ouvrir Modifier
Network Connected Signal...	Terminé	Chef de projet	0	19	Ouvrir Modifier
MI X90393 experiments	Terminé	Chef de projet	5	2	Ouvrir Modifier

bienvenue dans la section **CRÉER**

Clemens Valens, labo d'Elektro



André-Marie Ampère (1775 – 1836)

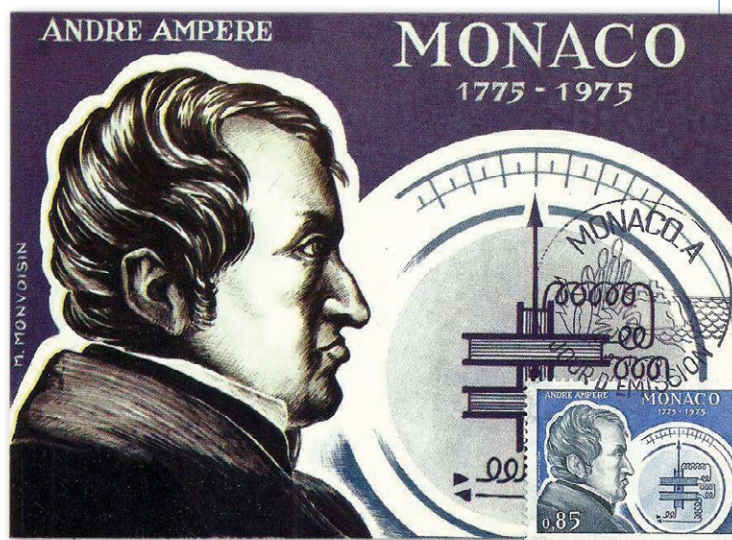
Né à Lyon en 1775, le jeune André-Marie Ampère n'est pas envoyé à l'école et s'instruit lui-même en puisant dans une bibliothèque familiale bien garnie. Il s'intéresse aux mathématiques à l'âge de 12 ans et publie son premier article un an plus tard. Durant la Révolution française, alors qu'il n'a encore que 16 ans, il propose l'adoption d'un nouveau système de mesure décimal. Son père est guillotiné deux ans plus tard et la fortune familiale saisie. Ampère s'intéresse alors à la biologie, invente des instruments scientifiques, observe les étoiles, apprend le grec et l'italien, ébauche une langue universelle, lit les poètes latins, et écrit même une tragédie dans laquelle il critique Christophe Colomb. Malgré cet apparent éparpillement intellectuel, Ampère ne perd pas son intérêt pour la chimie et mène en particulier des recherches sur l'oxyde de carbone. Lorsqu'à 25 ans il se rend

enfin à l'école, c'est en qualité de professeur et non d'élève.

En 1820, tandis qu'Ampère enseigne la philosophie à Paris, le physicien danois Ørsted remarque qu'un courant électrique dévie l'aiguille aimantée d'une boussole. L'absence de Twitter n'empêche pas la découverte de devenir virale. Aussitôt enthousiasmé, Ampère laisse tout tomber et mène ses propres expériences. Il réalise rapidement qu'un fil transportant un courant se comporte comme un aimant. De là il conclut que deux fils parallèles doivent se repousser ou s'attirer selon qu'un courant de même sens ou opposé les traverse.

Il en déduit également qu'à l'instar d'une aiguille de boussole un fil parcouru par un courant devrait subir l'influence du champ magnétique terrestre. Une semaine plus tard, Ampère présente ses premières découvertes devant l'Académie des sciences. Il y retourne après une semaine de travail acharné pour y lire un deuxième mémoire dans lequel il crée et impose le vocabulaire de l'électricité, dont les termes *courant* et *tension*. Poursuivant ses travaux, il invente des techniques de mesure du courant ainsi qu'une méthode de magnétisation de l'acier. Ampère résume ses recherches en décembre 1820 dans un article signant la naissance de l'électrodynamique. Poursuivant son étude des rapports entre électricité et magnétisme, il formule quelques années plus tard l'hypothèse selon laquelle le courant électrique est dû à la circulation à travers un conducteur d'une sorte de molécule électrodynamique. C'est en 1827 qu'il publie son *magnum opus* : *Mémoire sur la théorie mathématique des phénomènes électrodynamiques uniquement déduite de l'expérience*.

L'ampère étant devenu l'unité de base SI de l'intensité électrique, le nom d'Ampère est généralement associé à l'électricité. Son œuvre scientifique s'étend toutefois bien au-delà de ce domaine, puisqu'il aura aussi contribué de façon notable aux mathématiques et à la physique, jouant par exemple un rôle actif dans la découverte du chlore, du fluor et de l'iode. Ampère fait partie de ces grands esprits qui ont fait progresser la science à grands pas. ◀



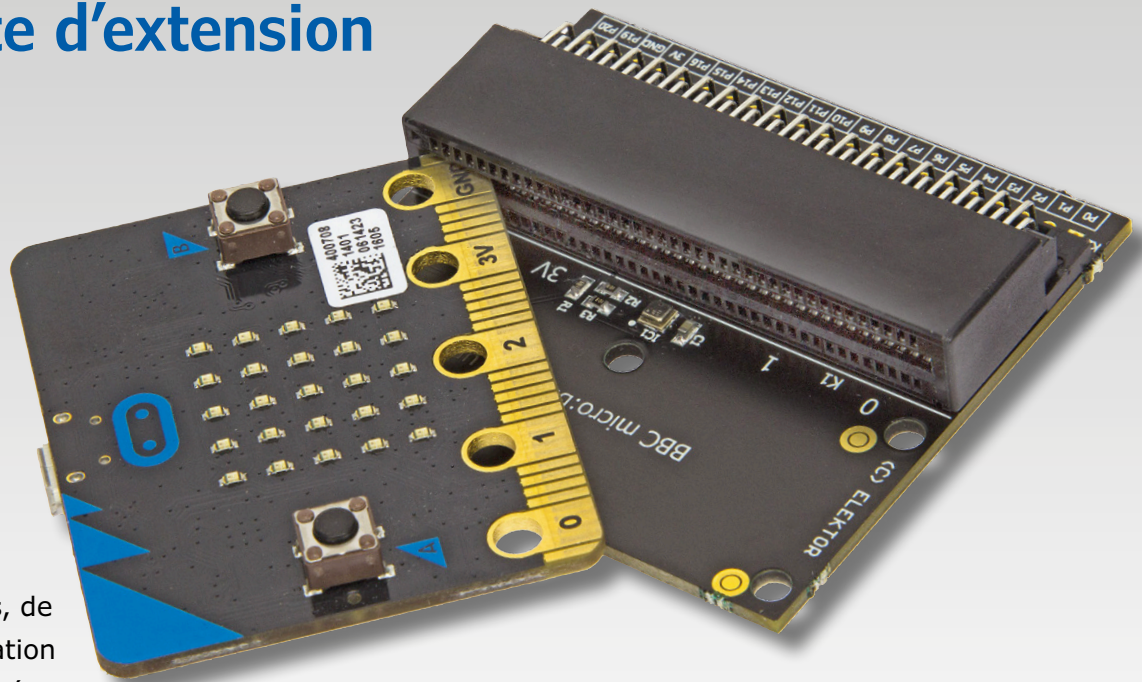
(160015 – version française : Hervé Moreau)

station météo micro:bit

et aussi carte d'extension générique !

Clemens Valens
(labo d'Elektor)

La carte micro:bit de la BBC est une petite centrale à tout faire qui bénéficie d'excellentes bibliothèques de fonctions, de bons outils de programmation et d'un connecteur imprimé donnant accès à de nombreux ports. Pour illustrer son utilisation, nous avons construit une station météo.



L'homme adore mesurer le monde. Mettez un mètre ruban dans les mains d'un enfant, il se mettra aussitôt à mesurer tout ce qu'il trouve. Les enfants devraient donc aimer la carte micro:bit [1]. D'abord parce qu'elle a été conçue pour eux, ensuite parce qu'elle est équipée de deux capteurs de mesure : un accéléromètre 3D (à trois axes) MMA8653, et un magnétomètre 3D MAG3110, capable également de retourner la température de sa puce. Ces deux capteurs ont été choisis pour leurs capacités de mouvement, d'orientation et de reconnaissance gestuelle. Nous les avons complétés par un capteur BME280 de pression, température et humidité de Bosch SensoTech. Ces trois grandeurs sont mesurées avec une grande précision et accessibles par bus I²C ou SPI. Munis de ces trois capteurs, nous pouvons transformer la micro:bit en centrale inertielle précise, ou en station météo sans fil si nous exploitons aussi son module Bluetooth.

Matériel

Le schéma de notre carte d'extension (**fig. 1**) est particulièrement simple : un capteur (IC1, **fig. 2**), une LED, trois résistances, un condensateur et K1, un connecteur dans lequel s'encarte la micro:bit, mais difficile à trouver.

Nous avons relié le signal de CSB d'IC1 (broche 2) à V_{IO} (reliée à V_{CC}) car c'est ainsi que s'active l'interface I²C du BME280. Pour sélectionner l'interface SPI, il faut placer la broche CSB

au niveau bas, la puce reste alors en mode SPI. Si, pour une raison ou une autre, vous souhaitez sélectionner l'interface I²C au moyen d'un signal externe, ce signal doit être au niveau de V_{IO} avant que la puce ne soit (ré)initialisée.

L'interface SPI est compatible avec les modes SPI « 00 » (CPOL = CPHA = 0) et « 11 » (CPOL = CPHA = 1). La sélection entre ces deux modes est automatiquement déterminée par la valeur de SCK après le front descendant de CSB. L'interface SPI a deux modes : 4 fils (par défaut) et 3 fils. Pour sélectionner le mode 3 fils, on met à 1 le bit `spi3W_en` du registre `config`. SDI est la ligne de données du mode 3 fils ; SDO n'est pas utilisé dans cette configuration.

Comme nous utilisons le capteur en mode I²C, nous avons placé pour le bus l'empreinte de deux résistances de rappel vers le haut (R2 et R3). Vous n'en aurez pas besoin si vous n'utilisez notre carte qu'avec la micro:bit, car celle-ci en est déjà pourvue.

Lorsque le BME280 est en mode I²C, sa broche SDO (n°5) détermine son adresse d'esclave : 0x76 si elle est reliée à GND, 0x77 si comme ici elle est reliée à V_{IO}. La broche SDO ne peut pas être laissée flottante (niveau indéfini).

Nous avons ajouté une LED témoin de marche/arrêt (LED1) car celle de la micro:bit est masquée lorsque la carte est insérée dans K1.

L'embase mâle K2 facilite la connexion des contacts du connecteur.

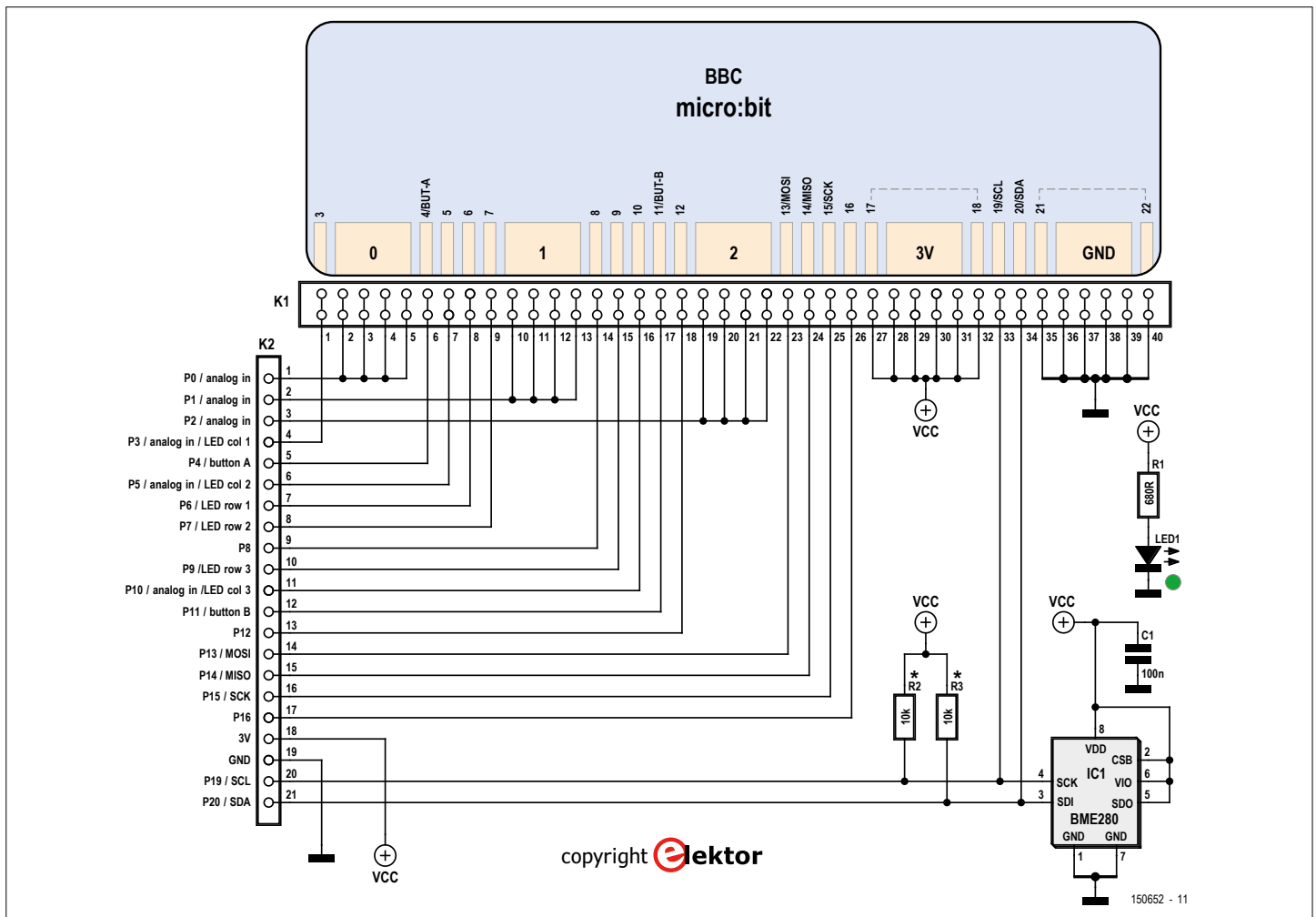


Figure 1. Schéma de la carte d'extension pour station météo.

teur imprimé de la micro:bit avec p. ex. une plaque d'essai. L'ordre de ses broches est légèrement différent de celui de la micro:bit car nous avons opté pour un ordre croissant de 0 à 20. Sur la micro:bit, les broches 0, 1 et 2 correspondent aux trois premiers gros trous de connexion ; chacun reçoit plusieurs contacts de K1. Les autres contacts sont numérotés de 3 à 21 de la gauche vers la droite (connecteur imprimé vers le bas, face « matrice de LED » visible).

Logiciel

Lorsque nous avons démarré ce projet, il n'existait encore aucune bibliothèque micro:bit qui nous eût permis d'écrire le

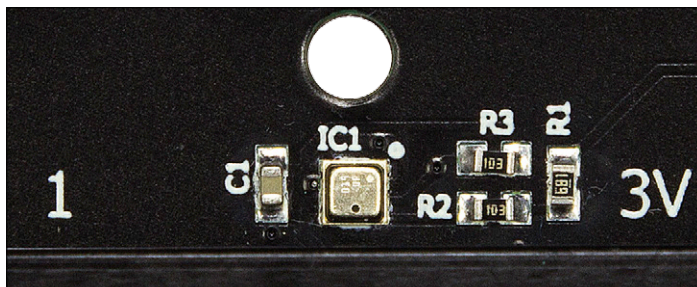


Figure 2. IC1, le petit cube placé au milieu de la carte, embarque trois capteurs de haute précision.

pilote nécessaire à la communication entre le microcontrôleur et le BME280. Nous avons donc utilisé le compilateur en ligne mbed [1], qui reconnaissait déjà la micro:bit [2]. Nous avons aussi écrit un croquis Arduino afin que vous puissiez utiliser le capteur avec une carte compatible Arduino.

Bosch Sensortech fournit bien un pilote pour le BME280 (sur GitHub), mais il est compliqué ; d'abord parce qu'il se veut compatible avec de nombreuses plateformes (64/32/16 bits, avec ou sans unité à virgule flottante), ensuite parce qu'il gère chaque détail de la puce. Pas besoin de tout ça pour notre simple station météo. Et puis, écrire son propre *driver* est toujours instructif. Bref, pour en revenir à notre pilote, lui aussi vous donne la main sur chaque bit de la puce, mais il vous faudra ouvrir la fiche technique pour savoir quoi lire/écrire et dans quel registre. Nous avons choisi le langage C++ car il permettait d'écrire une API façon Arduino. Vous n'aurez aucun mal à utiliser notre pilote [3]. Il crée d'abord un objet BME280 puis, comme le veut l'usage depuis qu'Arduino domine le monde, appelle la méthode (fonction) `begin` :

```
BME280 bme280;
bme280.begin(0x77);
```

La méthode `begin` reçoit comme argument l'adresse I²C de la puce (0x77). Un bus SPI est supposé être utilisé si aucune

adresse n'est fournie. La méthode retourne 0 si le capteur a bien été trouvé sur le bus et si ses données de calibrage ont pu être lues. On configure ensuite le capteur :

```
bme280.writeConfigRegister(BME280_STANDBY_500_
US,BME280_FILTER_OFF,0);
bme280.writeControlRegisters(BME280_
OVERSAMPLING_1X,BME280_OVERSAMPLING_1X,
BME280_OVERSAMPLING_1X,BME280_MODE_NORMAL);
```

Ici que des réglages essentiels, pas de haut débit, pas de filtrage, pas de paramètre sophistiqué. Le pilote est maintenant prêt à collecter les données. Nous appelons d'abord la méthode `read` de lecture des mesures, puis une ou plusieurs des méthodes membres `temperature`, `pressure` et `humidity`. Notez l'absence de la boucle habituellement utilisée ici.

```
bme280.read();
printf("T=%d degrees C, ", bme280.temperature()/100);
printf("P=%d mbar, ", bme280.pressure()/100);
printf("RH=%d%%\n", bme280.humidity()/10);
```

La température est divisée par 100 pour être convertie en degrés Celsius, la pression en pascals l'est par 100 pour être convertie en millibars, et l'humidité est divisée par 1024 pour être convertie en pourcentage ($>>10$ = dix décalages binaires vers la droite = division par 2^{10}). On peut activer l'arithmétique en virgule flottante en attribuant à la constante `BME280_ALLOW_FLOAT` une valeur différente de 0 (au début du fichier `bme280.h`) :

```
#define BME280_ALLOW_FLOAT (1)
```

Cela grossira et ralentira probablement l'exécutible. Que ce soit un problème ou non dépendra de votre application. En arithmétique flottante, il est inutile de convertir la température et l'humidité, et la pression sera exprimée en pascals. Notre pilote prend en charge les bus SPI et I²C (bien que nous n'ayons pas testé SPI), à l'utilisateur de fournir les fonctions correspondantes. Selon la configuration choisie, le pilote appellera les fonctions appropriées :

`i2cWrite(...)` & `i2cRead(...)`
`spiWrite(...)` & `spiRead(...)`

Ces quatre fonctions doivent être fournies mais peuvent être laissées vides. Si vous utilisez I²C, laissez vides les pseudo-codes SPI ; si vous utilisez SPI, laissez vide la partie I²C. Notez qu'en mbed les adresses I²C sont représentées sur 8 bits car le bit de lecture/écriture est inclus dans le bit 0 ; autrement dit 0x77 devient 0xee. En Arduino, l'adresse est 0x77.

Ceci n'est pas une annonce de la BBC

K2 est supposé être une barrette mâle, mais libre à vous d'utiliser un autre connecteur. Mâle ou femelle, K2 permet de relier facilement la carte à une plaque d'essai, en particulier avec ces fils de connexion prédécoupés si pratiques. Si vous montez une barrette mâle, horizontale ou verticale, vous serez prêt pour notre prochain projet : une station d'accueil pour la micro:bit. ◀

(150652 – version française : Hervé Moreau)

Liens

- [1] <https://developer.mbed.org/platforms/Microbit/>
- [2] <https://github.com/ElektorLabs/bme280-driver>
- [3] <https://www.elektormagazine.fr/150652/>



Liste des composants

Résistances (5 %, 50 V, 0,1 W, 0603)

R1 = 680 Ω
 R2, R3 = 10 kΩ

Condensateur (0603)

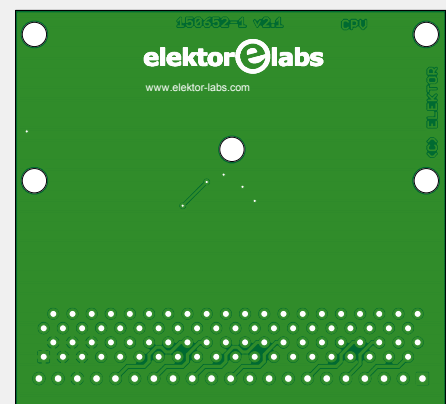
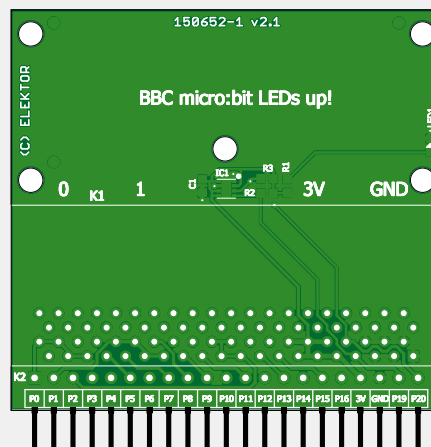
C1 = 100 nF

Semi-conducteur

IC1 = BME280
 LED1 = LED, verte, 0603

Divers

K1 = embase femelle circuit imp.
 vers circuit imp., 2 x 40, pas de 1,27 mm
 K2 = barrette 1 x 21 broches, pas de 2,54 mm
 Circuit imprimé réf. 150652 (www.elektor.fr)



SDR d'Elektor réinventé

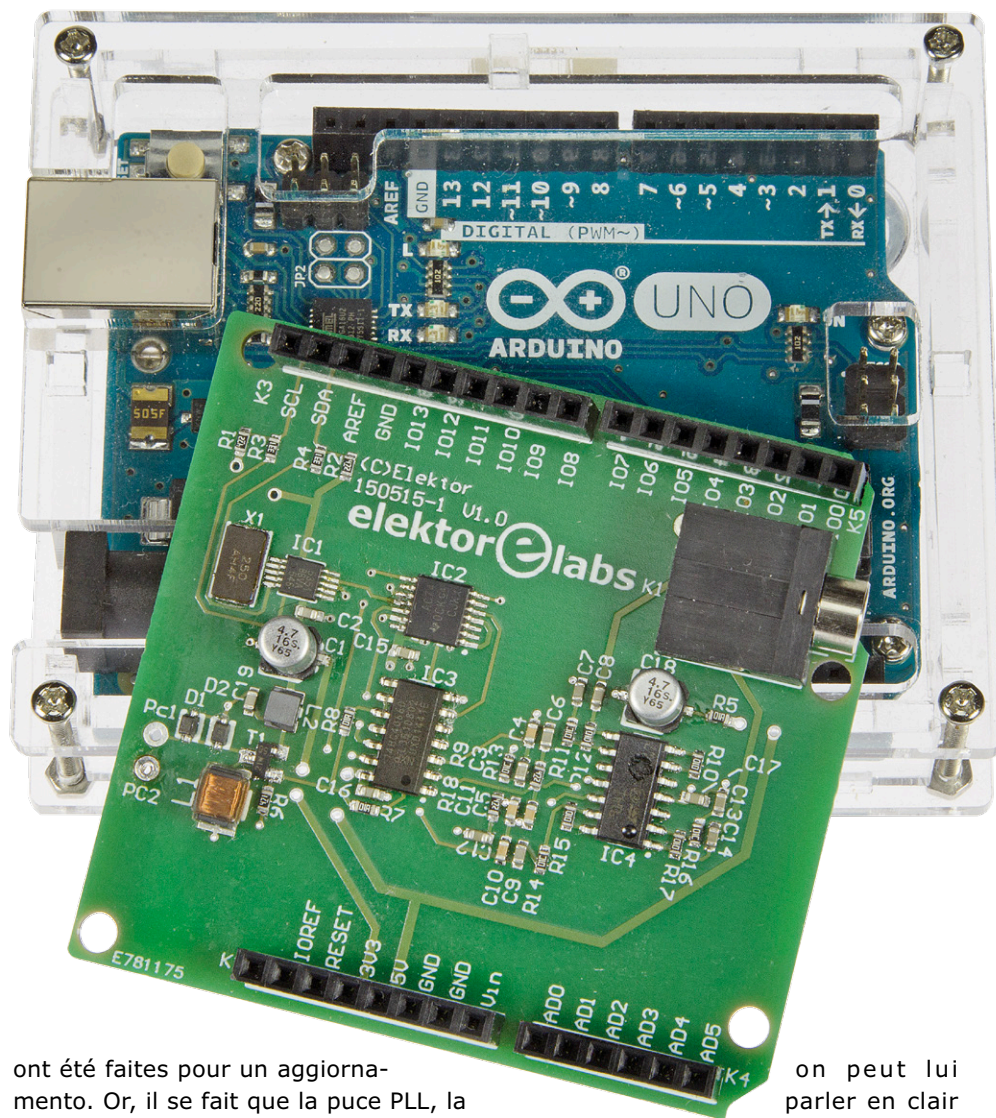
sur *shield* Arduino

Burkhard Kainka (Allemagne)

Une radio logicielle, SDR pour *Software Defined Radio*, est un outil universel de travail en HF, qui permet aussi d'effectuer des mesures. Les caractéristiques du récepteur, c'est le logiciel qui les définit. Pour l'interface de communication, nous disposons à présent de la plateforme Arduino sur laquelle monter un nouveau *shield*.

Caractéristiques

- Tension d'alimentation : 5 V et 3,3 V d'Arduino
- Gamme de fréquences : 150 kHz à 30 MHz
- Sensibilité : 1 μ V
- Gain global : 40 dB
- Niveau maximal d'antenne : 10 mV
- Plage de dynamique : 80 dB



De plus en plus de services de radiodiffusion quittent les ondes longues, moyennes et courtes qui offrent alors un magnifique terrain de jeu pour se lancer dans la chasse aux ondes. Maintenant plus que jamais, on y découvre de nombreuses stations lointaines, enfin débarrassées des signaux puissants qui les noyaient. En ondes courtes, il m'arrive souvent de douter de mon récepteur, tellement le silence est assourdissant. Dans de nombreuses bandes, les signaux les plus forts proviennent de radioamateurs. Vous y trouvez toujours du nouveau, depuis les stations pirates en radiotéléphonie sur SSB jusqu'aux récents développements numériques. De quoi exciter la curiosité ! Elektor a présenté et construit de nombreuses radios et en mai 2007 [1] sortait déjà la radio logicielle à interface USB. Depuis lors, de fréquentes suggestions

ont été faites pour un aggrégement. Or, il se fait que la puce PLL, la boucle à phase asservie, utilisée alors n'est plus fabriquée. Il fallait trouver une solution de rechange. Elle est venue sous la forme de la puce en CMOS Si5351 de Silicon Labs, un générateur d'horloge de 8 kHz à 160 MHz avec bus I²C.

Les premières recherches ont eu lieu avec un BoB d'Adafruit. Les logiciels d'exemple actuels ont été rédigés pour Arduino et ont donc servi aux essais préliminaires. Le nouvel oscillateur réglé en tension (VFO) a été simplement relié à l'ancienne carte SDR pour prouver son efficacité (**figure 1**).

Ainsi est née l'idée de construire tout le récepteur sur un *shield* Arduino. La question de l'alimentation est réglée et l'interface USB du PC, à portée de main. Arduino pilote le VFO et pour ainsi dire,

on peut lui parler en clair (à 6 030 kHz tout de même). Et sans doute le plus important, arriver à construire un récepteur complètement indépendant. Il devrait être relativement simple de faire migrer l'interface utilisateur du PC vers Arduino. Et qui sait, décoder les signaux IQ un de ces jours ?

Comment ça marche ?

Reprenons d'abord depuis le début. C'est quoi, au juste une radio logicielle ? Malgré le développement du numérique en électronique, la radio est longtemps restée à la traîne. Quand les ordinateurs domestiques se sont répandus, la plupart des radios étaient encore analogiques. Commence alors un timide développement, du moins avec la numérisation de l'accord.

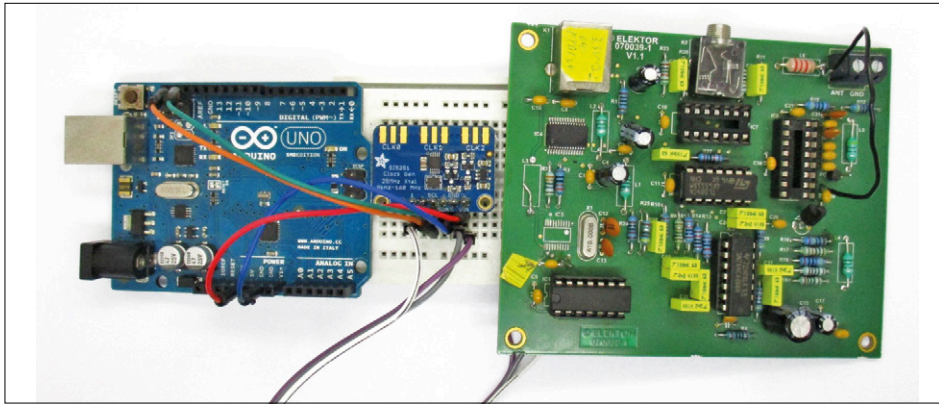


Figure 1. Les essais préliminaires pour le SDR « nouvelle génération » : une puce PLL Si5351 raccordée à une Arduino Uno et à l'ancienne carte de récepteur SDR.

Les postes sont généralement équipés d'un synthétiseur à PLL, ce qui simplifie l'accord et garantit le respect de la grille des canaux. Mais le reste des circuits est encore largement analogique.

Pendant ce temps-là, l'électronique numérique a débarqué dans la technique HF et chez les radioamateurs. Sans cesse, plus de fonctions analogiques des appareils sont remplacées par du logiciel. Le plus souvent, un processeur de signal numérique (*DSP*) travaille avec son logiciel à l'insu de l'utilisateur et s'occupe des caractéristiques optimales des filtres, de faire varier la largeur de bande, de décoder des signaux, de déparasiter et plein d'autres choses. Les appareils sont ainsi de meilleure qualité avec moins de dépense en matériel. D'autres exemples de ces développements sont les ordinateurs et les terminaux mobiles. Dans cette cour-là, l'électronicien amateur ne peut plus jouer.

Mais en réalité, il n'est pas nécessaire de chercher des complications. On utilise un convertisseur A/N rapide auquel on branche directement l'antenne. Tout le spectre est numérisé pour être traité ensuite. Le système peut fonctionner dans tout le domaine de 0 à 30 MHz. Le logiciel épure certaines fréquences et démodule le signal voulu. Peu importe qu'il s'agisse d'émetteurs radiophoniques AM ou DRM, de signaux SSB, d'émetteurs en morse, de télécopieurs météo ou d'autres réceptions, tout est possible et pour tous, il existe un logiciel adapté. Toutefois, le matériel nécessaire pour une aussi grande largeur de bande est vraiment cher et le traitement ultérieur d'un spectre aussi étendu impose des exigences sévères.

La carte son d'un PC moderne offre déjà une possibilité. Avec un taux d'échantillonnage de 96 kHz courant aujourd'hui, on peut déjà recevoir toute la gamme jusqu'à 48 kHz. Sur l'entrée pour microphone, on branche une grosse bobine comme antenne pour recevoir la gamme TBF. Il y a là beaucoup de signaux intéressants, jusqu'aux émetteurs des sous-marins.

Pour utiliser la carte son à de plus hautes fréquences, il faut d'abord effectuer une conversion vers le bas par mélangeur. Le système correspond à celui d'un superhétérodyne avec fréquence intermédiaire (FI) plus basse. Les étages FI, le filtrage, le réglage automatique

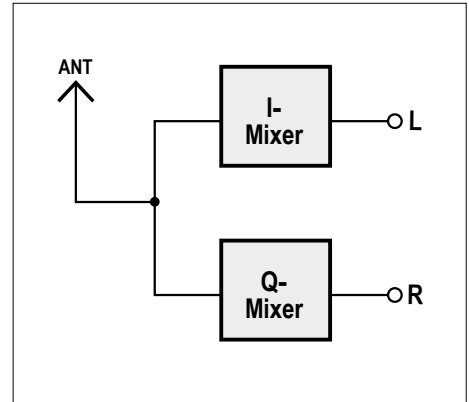


Figure 2. L'idée du circuit : l'interface se résume à un double mélangeur direct sur des signaux déphasés de 90°.

de gain et la démodulation, le PC s'en charge. En principe, il suffit d'un simple mélangeur direct à diodes ou le célèbre NE612. Il ne faut y ajouter qu'un VFO stable. Pour des applications spéciales, on utilise un oscillateur à quartz. Mais si l'on veut pouvoir s'accorder sur toute la gamme, il faut un générateur DDS ou une puce PLL.

Avec un mélangeur IQ, en bref, on a affaire à un double mélangeur direct avec deux signaux déphasés de 90° l'un par rapport à l'autre. Le signal de l'oscillateur se trouve toujours à proximité de la fréquence de réception. Les signaux de sortie se situent dans le domaine BF généralement entre 0 et 24 kHz. Les deux signaux s'appellent I et Q comme à la

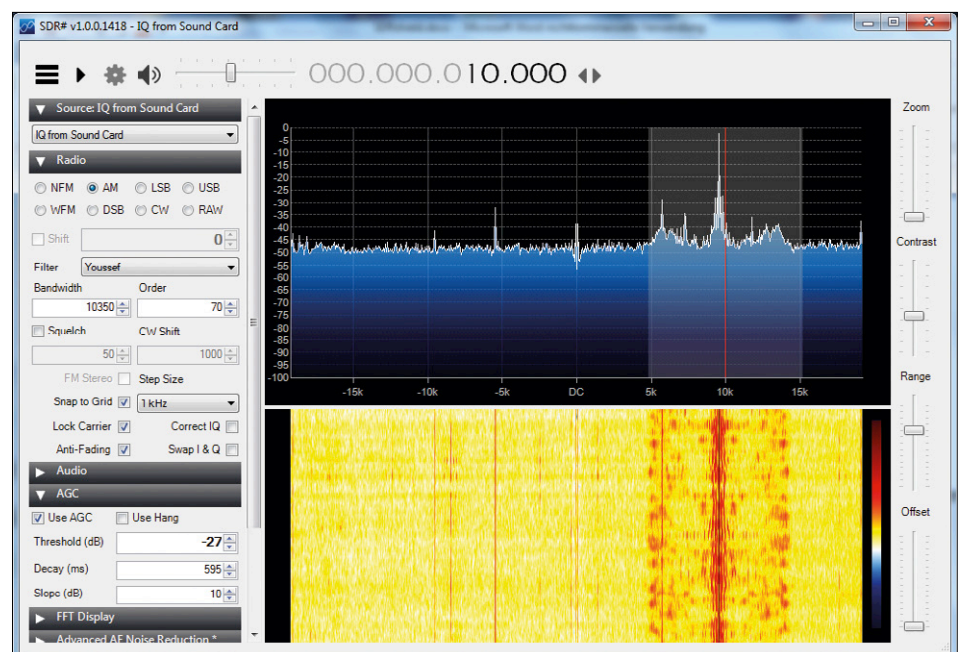


Figure 3. Le programme SDR# en réception d'un signal AM.

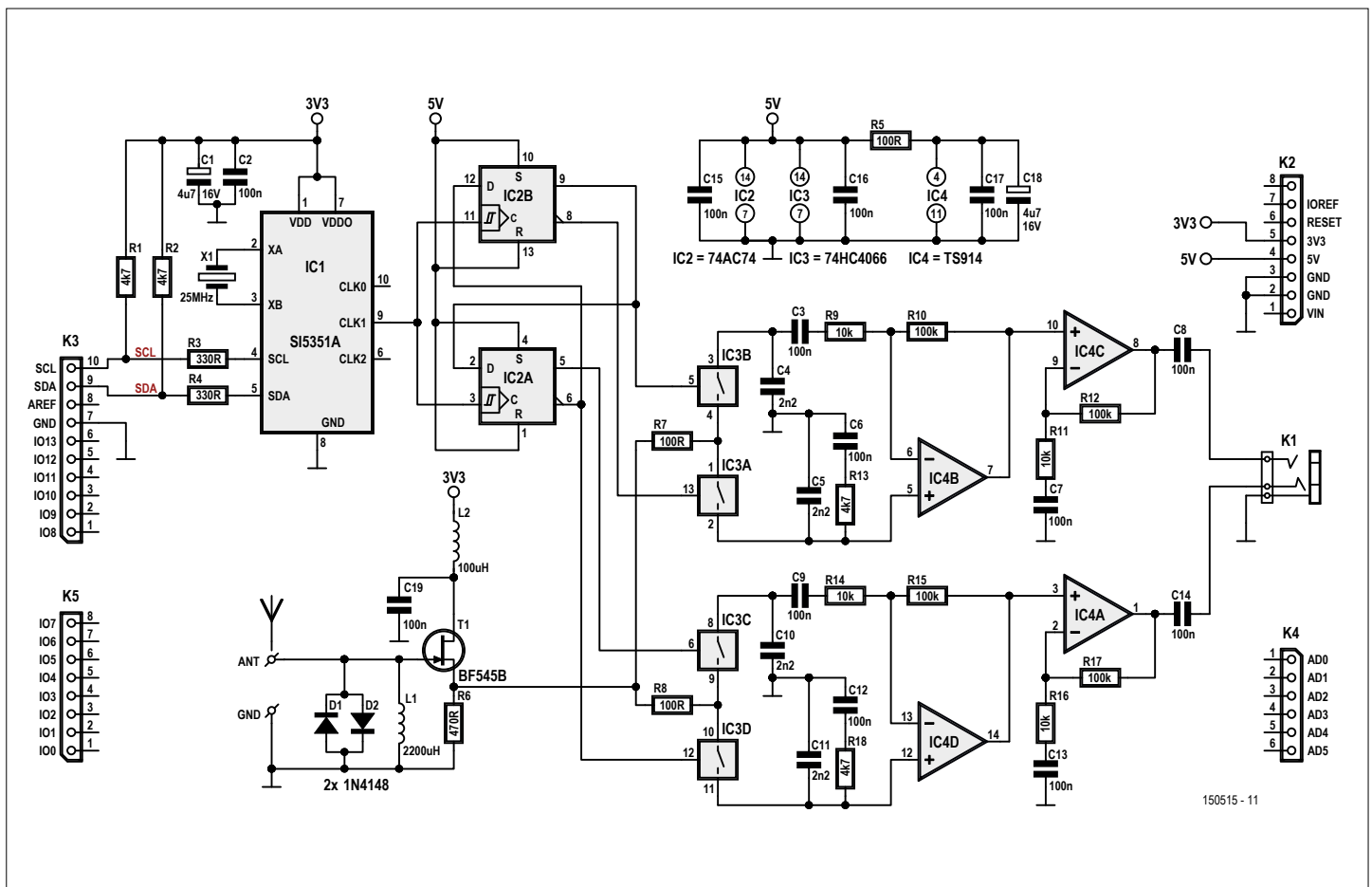


Figure 4. Le schéma de la nouvelle radio logicielle.

Liste des composants

Résistances :

R1, R2, R13, R18 = 4,7 k Ω , 1 %/100 mW, CMS 0603
 R3, R4 = 330 Ω , 1 %/100 mW, CMS 0603
 R5, R7, R8 = 100 Ω , 1 %/100 mW, CMS 0603
 R6 = 470 Ω , 1 %/100 mW, CMS 0603
 R9, R11, R14, R16 = 10 k Ω , 1 %/100 mW, CMS 0603
 R10, R12, R15, R17 = 100 k Ω , 100 mW, CMS 0603

Condensateurs :

C1, C18 = 4,7 μ F/16 V, CMS boîtier B
 C2, C3, C6, C7, C8, C9, C12, C13, C14, C15, C16, C17, C19 = 100 nF/50 V, X7R, CMS 0603
 C4, C5, C10, C11 = 2,2 nF/50 V, X7R, CMS 0603

Inductances :

L1 = 2 200 μ H (Fastron L-1812AF)
 L2 = 100 μ H (Murata LQH32CN101K23L)

Semi-conducteurs :

D1, D2 = 1N4148WS, SOD-323
 T1 = BF545B, SOT-23
 IC1 = SI5351A-B-GT, MSOP-10
 IC2 = SN74AC74PW, TSSOP-14
 IC3 = 74HC4066, SOIC-14

IC4 = TI914IDT, SOIC-14

Divers :

K1 = prise pour jack stéréo 3,5 mm encartable
 K2, K3, K4, K5 = ensemble de connecteurs compatibles Arduino (1x6 broches, 2x8 broches, 1x10 broches)

X1 = quartz 25 MHz (Abracon ABM7) circuit imprimé, réf. 150515-1 ou circuit imprimé avec les CMS installés, réf. 150515-91 (www.elektor.fr)

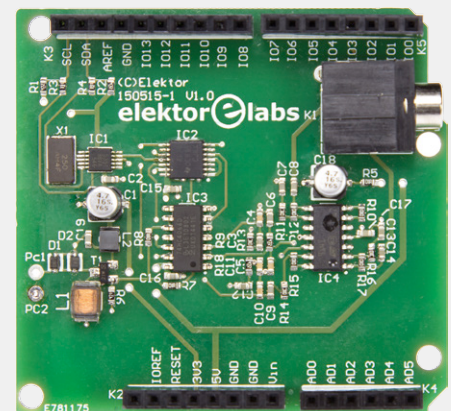
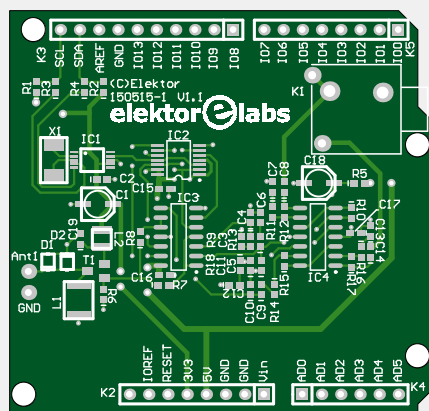
Figure 5. Le circuit imprimé à double face pour le nouveau récepteur SDR prend la forme d'un *shield* Arduino.

figure 2. On les envoie directement aux canaux gauche et droit de l'entrée de la carte son. La suite appartient au logiciel. Un mélangeur simple mélangerait la bande en bas et celle en haut de la fréquence de l'oscillateur dans la même gamme. Il s'ensuivrait la production de fréquence image. Avec le mélange double et le déphasage, le logiciel est en mesure de déduire la fréquence image. On reçoit ainsi un domaine entre -24 kHz et $+24$ kHz si la carte son dispose d'un taux d'échantillonnage de 48 kHz. La **figure 3** montre comment s'y prend un programme comme SDR# (cf. aussi l'encadré Le logiciel pour SDR).

Le schéma

Voyez dans le schéma de la **figure 4** les groupes individuellement. Le générateur PLL Si5351 (IC1) fournit le signal d'oscillateur, quadruple de la fréquence de réception, au diviseur 74AC74 (IC2B), lequel la divise par quatre pour appliquer le signal déphasé de 90° au mélangeur 74HC4066 (IC3). Ce commutateur analogique est câblé en inverseur et envoie le signal HF alternativement aux entrées inverseuse et non inverseuse de l'amplificateur opérationnel TS914 (IC4B/IC4D). C'est ce qui effectue la conversion vers le bas du signal dans le domaine BF. Après un léger filtrage et une amplification (IC4C/IC4A), le signal aboutit à la sortie audio. L'étage de sortie HF est un tampon à source suiveuse, formé par le JFET BF545B (T1), l'équivalent CMS du BF245B.

En comparaison avec l'ancienne version de radio logicielle d'Elektor, on remarque une sérieuse simplification dans le trajet du signal. Les multiples filtres passe-bas commutables ont laissé place à une entrée à large bande avec deux diodes anti-surtension. L'expérience du passé ayant conclu à un risque de dommages sur l'étage d'entrée en cas d'orage, il est ainsi bien couvert pour la réception des ondes courtes avec un fil d'antenne. Pour des besoins particuliers, il reste possible d'y ajouter un filtre extérieur ou un préamplificateur. Le gain en BF, réglable sur trois niveaux sur l'ancien modèle, reste maintenant moyen, valeur révélée d'usage normal. Avec ces simplifications, il est plus aisé de tout installer sur un *shield*.

Pour les premiers essais, il faut juste raccorder une antenne. L'idéal est un fil d'an-

tenne libre haubané d'une longueur d'au moins trois mètres. Si ce n'est pas possible, on peut utiliser un fil plus long n'importe où dans la pièce, mais une antenne intérieure est souvent plus parasitée. La construction optimisée d'antennes fera encore l'objet d'un thème d'article pour Elektor.

La construction

Le circuit imprimé est donc exécuté au format de *shield* pour Arduino, de manière à se brancher directement sur Arduino Uno. Comme le Si5351 n'est livrable qu'en CMS à dix contacts, il a été décidé de construire l'ensemble en CMS et de proposer dans l'e-choppe la carte assemblée [3], sur laquelle il reste à souder le connecteur compatible Arduino. Si vous préférez fabriquer complètement le circuit, vous pouvez télécharger en [3] le tracé des pistes ou acheter dans l'e-choppe le circuit imprimé nu.

Le réglage de fréquence

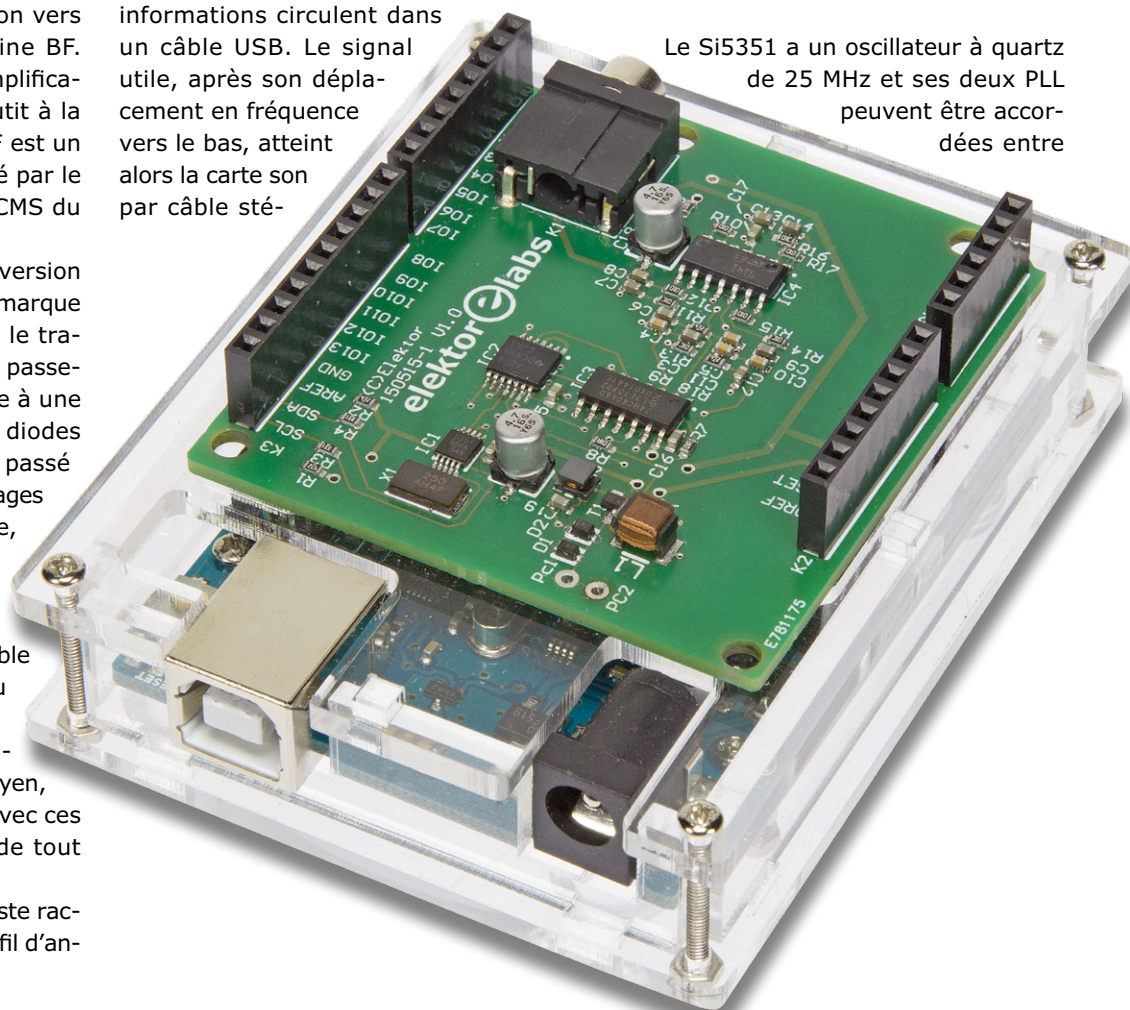
Arduino, de concert avec le *shield* SDR, sert d'interface entre l'antenne et le PC. Sa tâche se limite au réglage du VFO. Et en plus, il indique au PC quelle fréquence on veut avoir. Entre le PC et Arduino, les informations circulent dans un câble USB. Le signal utile, après son déplacement en fréquence vers le bas, atteint alors la carte son par câble stéréo

pour le traitement ultérieur. Bien entendu, on aurait pu essayer de déplacer l'interface vers Arduino et peut-être y ajouter un traitement simplifié du signal, mais ce serait beaucoup demander à un petit système. Ce qui compte d'abord, c'est qu'Arduino reçoive ses commandes du PC et règle le VFO.

Je ne vais pas décrire maintenant ce qui se passe avec l'EDI d'Arduino, supposons que tout se passe bien. Il faut commencer par charger un programme Arduino adapté. Ce que ce logiciel fait exactement vous sera expliqué dans la suite. Vous préférez sans doute arriver tout de suite à un premier captage, alors laissez de côté ces informations et chargez simplement le logiciel [3].

Sa mission décisive consiste à persuader le Si5351 de délivrer une fréquence appropriée. Cette puce compte deux PLL internes et trois sorties, son diagramme fonctionnel est à la **figure 6**. Nous n'utilisons ici que la PLL A et la sortie CLK1. Le croquis utilise la bibliothèque d'Adafruit, ce qui rend la chose agréablement simple. Mais avant d'en profiter, il faut que la bibliothèque soit chargée depuis [2] et incluse.

Le Si5351 a un oscillateur à quartz de 25 MHz et ses deux PLL peuvent être accordées entre



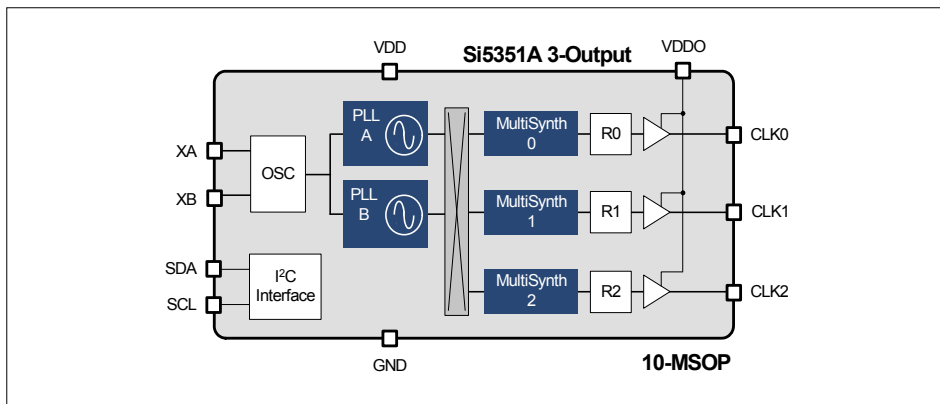


Figure 6. Diagramme fonctionnel dans la fiche technique du Si5351A.

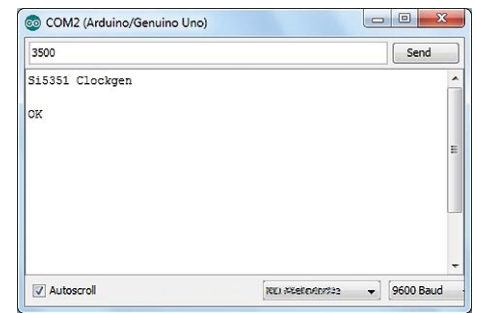


Figure 7. Pilotage du générateur d'horloge avec le terminal Arduino.

600 MHz et 900 MHz. Comme les diviseurs des PLL travaillent avec des nombres fractionnaires, on peut obtenir pratiquement n'importe quelle résolution. Les diviseurs MultiSynth qui suivent utilisent également des fractionnaires. On dispose ainsi de deux possibilités (A ou B) pour produire

la fréquence voulue :

- régler la PLL sur la fréquence fixe de 900 MHz, par exemple, et ensuite diviser par un nombre fractionnaire ;
- régler la PLL par tâtonnements, puis diviser par un nombre entier pour

arriver à la fréquence voulue.

Voyons la méthode A. La fréquence du VFO se situe à quatre fois celle du mélangeur, qui se trouve 12 kHz en dessous de la fréquence de réception. Le programme doit recevoir cette fréquence de réception exprimée en kHz au format texte et la

Listage 1. Programme pour un réglage fixe de PLL

```
//SI5351_vfo PLL fixed at 900 MHz (si5351vfo2.zip)

#include <Adafruit_Sensor.h>
#include <Wire.h>
#include <Adafruit_SI5351.h>

Adafruit_SI5351 clockgen = Adafruit_SI5351();
void setup(void)
{
  Serial.begin(9600);
  Serial.println("Si5351 Clockgen"); Serial.println("");

  /* Initialise the sensor */
  if (clockgen.begin() != ERROR_NONE)
  {
    Serial.print("Error");
    while(1);
  }
  Serial.println("OK");
  clockgen.enableOutputs(true);
  clockgen.setupPLL(SI5351_PLL_A, 36, 0, 1000); //900
  MHz
  setfreq (6000);
}

void setfreq (unsigned long freq)
{
  unsigned long f2;
  unsigned long f3;
  unsigned long f4;
  unsigned long f5;
  unsigned long div2;
```

```
unsigned int Divider2;
unsigned int rdiv;

if (freq > 0) {
  f2=(freq-12)*4;
  if (f2<1000) {
    rdiv = 16;
    f2 = f2 * 16;
  }
  else {
    rdiv = 1;
  }
  div2 = 900000000/f2;
  f4 = div2/1000;
  f5=div2-(f4*1000);
  clockgen.setupMultisynth(1, SI5351_PLL_A, f4, f5,
1000);
  if (rdiv == 16) {
    clockgen.setupRdiv(1, SI5351_R_DIV_16);
  }
  if (rdiv == 1) {
    clockgen.setupRdiv(1, SI5351_R_DIV_1);
  }
}

void loop(void)
{
  unsigned long freq;
  if (Serial.available()) {
    freq = Serial.parseInt();
    setfreq (freq);
  }
}
```


convertir. Pour recevoir sur 3 500 kHz, le Si5351 doit produire une fréquence de $4 \times (3\,500 - 12) \text{ [kHz]} = 13\,952 \text{ kHz}$ sur la sortie 1. Le diviseur de la PLL est mis sur 36 ($25 \text{ [MHz]} \times 36 = 900 \text{ MHz}$), le diviseur Multisynth sur $900\,000/13\,952 = 64,506$. De cette manière, on descend jusqu'à 1 MHz. Pour aboutir à des fréquences plus basses, on met en œuvre le diviseur R_DIV réglé sur 16. Le **lis-tage 1** montre le logiciel correspondant et la **figure 7**, la conduite avec le terminal Arduino.

La méthode A présente l'avantage de pouvoir mettre au point à tout moment le VFO, sans causer d'interruption pendant le changement de fréquence.

La méthode B, en revanche, garantit une meilleure pureté de phase, ce qui suffit aussi pour la DRM. Toutefois, chaque changement de fréquence s'accompagne d'une courte interruption d'environ 1 ms qui se manifeste comme parasite en radio logicielle. Cette méthode demande un calcul du diviseur optimal suivant (**lis-tage 2**) pour que la PLL reste toujours dans le domaine de 600 à 900 MHz.

Tout terminal peut diriger chacun des deux programmes. Pour votre confort, on a écrit un programme VB en Visual Studio 2015. Téléchargement en [3] : SDRshield.zip. Il envoie à Arduino la fréquence souhaitée à 9600 bauds au format texte, par ex. 3500. La glissière de réglage (cf. **figure 8**) va par pas de 9 kHz jusqu'à 1,6 MHz, puis s'en tient au standard de 5 kHz. Comme moyen subsidiaire, on peut donner directement la fréquence voulue ou cliquer sur les boutons à côté de chaque bande de radiodiffusion ou de radioamateurs.

N'oubliez pas, lors du premier accès, de choisir le bon port COM.

Le logiciel SDR

Voici un aperçu du logiciel SDR utilisable. Pratiquement tous les programmes utilisés avec l'ancienne radio logicielle d'Elektor peuvent servir, ils fonctionnent encore maintenant.

- SDRadio reste encore un bon choix
- SoDoRa peut aussi décoder la DRM
- DREAM fonctionne encore, mais n'utilise pas le signal IQ et nécessite un récepteur comme un mélangeur direct
- HDSDR est un logiciel actuel et très puissant
- SDRSharp (SDR#) se signale par un maniement simple et une bonne

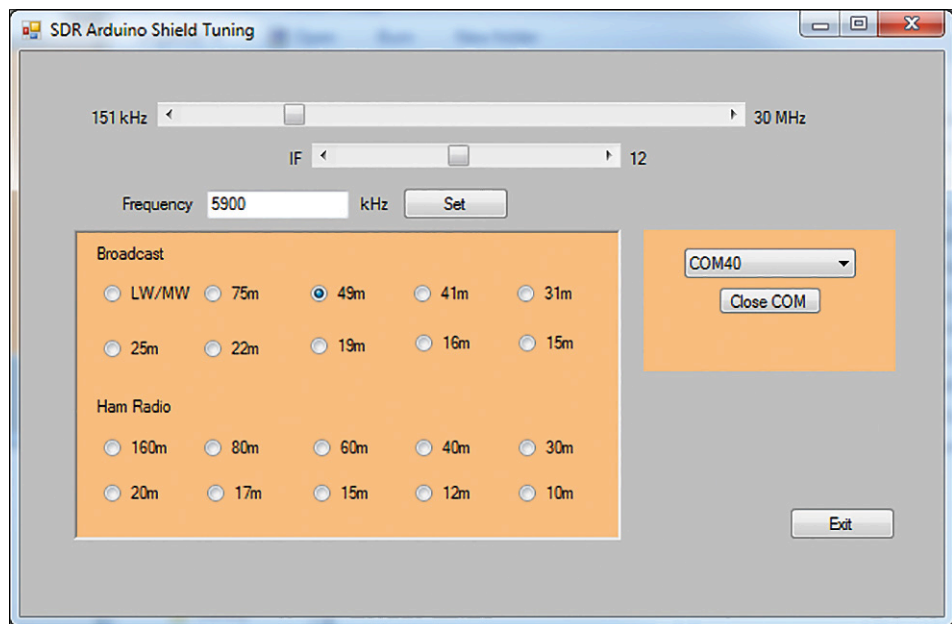


Figure 8. C'est un petit programme VB qui rend le maniement confortable.

présentation.

Dans un prochain article, nous examinerons en détail chaque programme et ses possibilités.

Premières expériences de réception

À défaut d'une meilleure antenne, on peut faire ses premiers essais avec un fil d'un à trois mètres de long à brancher à l'entrée d'antenne pour recevoir sans difficulté des émetteurs de radiodiffusion dans toutes les bandes d'ondes courtes. L'expérience

montre que le soir, on capte moins que le jour. Et le phénomène se déplace le soir vers les bandes plus basses de 75 jusqu'à 41 m. On peut aussi recevoir les stations de radioamateurs avec une antenne filaire plus courte. Il s'en trouve plus dans la bande de 40 m et avec un peu de chance, on écoute quelques stations CW (morse) et SSB (BLU). On choisit le mode approprié dans le logiciel SDR, réglage de volume, bande passante, contrôle automatique de niveau et bien d'autres paramètres. Avec les bons réglages, on récolte

Le logiciel pour SDR

Les étoiles qui brillent actuellement au firmament du logiciel SDR s'appellent SDR# [4] et HDSDR [5]. Tous deux suivent la nouvelle tendance vers des fréquences toujours plus hautes et il est possible de les utiliser avec une simple clé électronique (*dongle*) comme récepteurs DVB-T. C'est un bon choix quand on veut écouter les gammes VHF et UHF. Des essais ont cours pour leur permettre de descendre sous les 30 MHz. On pourrait mettre en jeu un mélangeur élévateur qui remonte tout le spectre de 50 MHz. On aurait alors un superhétérodyne multiple accompagné des nombreux soucis bien connus de signaux fantômes et d'interférences, ainsi qu'une perte de dynamique. Une radio logicielle dédiée au spectre sous les 30 MHz ne demande qu'une seule conversion de fréquence et procure ainsi une réception très propre et sans sifflements.

Sur le PC, il y a deux programmes qui tournent, celui qui réalise l'accord et le logiciel SDR. Chaque programme SDR a sa propre interface de commande. Les premiers pas sont toujours semblables. D'abord, s'assurer que c'est la bonne entrée qui est utilisée. Puis sélectionner la carte son et activer cette entrée-là (*Line In*). Le logiciel SDR démarre. On voit qu'on a choisi la bonne entrée par une augmentation significative du bruit de fond, qui grimpe encore avec une antenne. Le plus souvent, il faut régler la carte son sur un niveau plus faible, du fait que le récepteur peut fournir jusqu'à 1 V d'amplitude du signal.

Listage 2. Programme pour PLL variable

```
//SI5351_vfo, variable PLL (si5351vfo3.zip)

#include <Adafruit_Sensor.h>

#include <Wire.h>
#include <Adafruit_SI5351.h>

Adafruit_SI5351 clockgen = Adafruit_SI5351();
void setup(void)
{
  Serial.begin(9600);
  Serial.println("Si5351 VFO"); Serial.println("");

  if (clockgen.begin() != ERROR_NONE)
  {
    Serial.print("Error");
    while(1);
  }
  Serial.println("OK");
  clockgen.enableOutputs(true);
  setfreq (6000);
}

void setfreq (unsigned long freq)
{
  unsigned long f2;
  unsigned long f3;
  unsigned long f4;
  unsigned long f5;
  unsigned int Divider2;
  unsigned int rdiv;

  if (freq > 0)
  {
    f2=(freq-12)*4;
    // f2=freq;
    if (f2>120000) {
      f2=120000;
    }
    if (f2<800) {
      rdiv = 16;
      f2 = f2 * 16;
    }
    else {
      clockgen.setupRdiv(1, SI5351_R_DIV_1);
      rdiv = 1;
    }
    if (f2 >= 100000) {
      Divider2 = 6;
    }
    if (f2 < 90000) {
      Divider2 = 10;
    }
    if (f2 < 60000) {
      Divider2 = 15;
    }
    if (f2 < 50000) {
```

```
      Divider2 = 18;
    }
    if (f2 < 45000) {
      Divider2 = 20;
    }
    if (f2 < 30000) {
      Divider2 = 30;
    }
    if (f2 < 20000) {
      Divider2 = 45;
    }
    if (f2 < 15000) {
      Divider2 = 60;
    }
    if (f2 < 10000) {
      Divider2 = 90;
    }
    if (f2 < 6000) {
      Divider2 = 150;
    }
    if (f2 < 4000) {
      Divider2 = 220;
    }
    if (f2 < 2700) {
      Divider2 = 330;
    }
    if (f2 < 1800) {
      Divider2 = 500;
    }
    if (f2 < 1500) {
      Divider2 = 600;
    }
    if (f2 < 1000) {
      Divider2 = 900;
    }
    f2=f2*Divider2;
    f2=f2*1000/25;
    f3=f2 /1000;
    f4 = f3/1000;
    f5=f3-(f4*1000);
    clockgen.setupPLL(SI5351_PLL_A, f4, f5, 1000);
    clockgen.setupMultisynth(1, SI5351_PLL_A, Divider2,
0, 2);
    if (rdiv == 16) {
      clockgen.setupRdiv(1, SI5351_R_DIV_16);
    }
  }
}

void loop(void)
{
  unsigned long freq;
  if (Serial.available()) {
    freq = Serial.parseInt();
    setfreq (freq);
  }
}
```

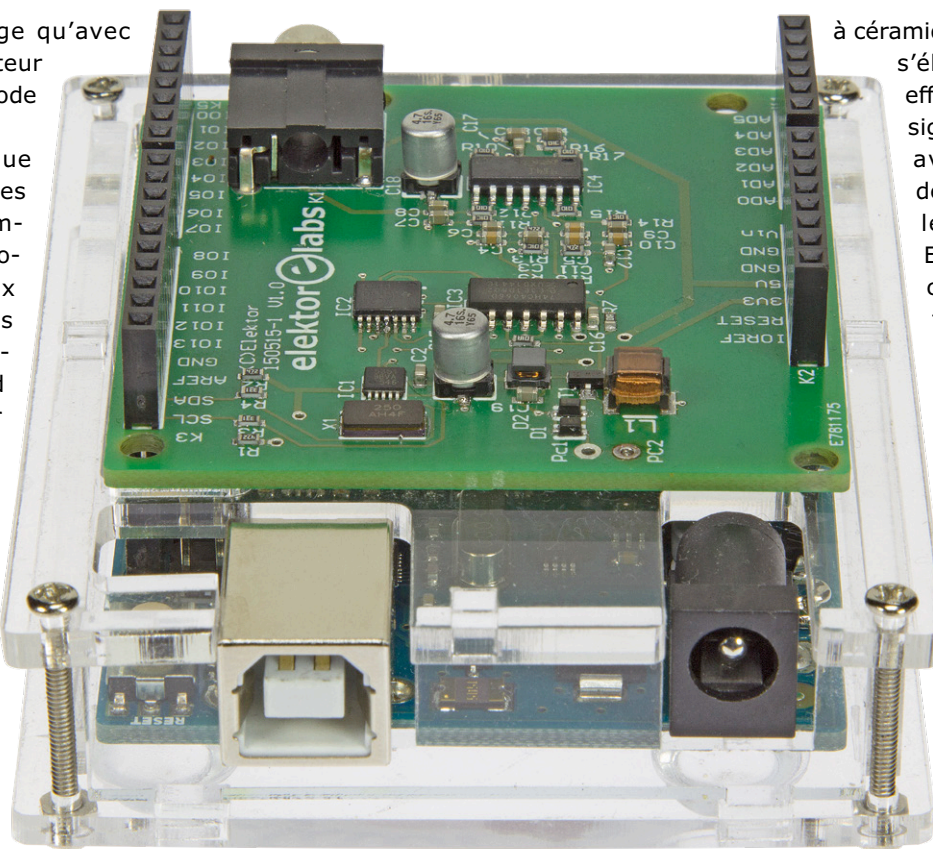
souvent davantage qu'avec un coûteux récepteur analogique à la mode ancienne.

Une caractéristique capitale de tous les mélangeurs à commutation est la production de signaux à des multiples impairs de la fondamentale. Quand on veut recevoir un signal à 1 MHz, des harmoniques à 3 MHz, 5 MHz, 7 MHz, etc. peuvent perturber la réception. Aussi, utilise-t-on souvent des filtres passe-bas commutables. Le *shield* SDR n'y a pas recours, donc il est logique d'utiliser des

antennes sélectives, mais il fonctionne étonnamment bien avec une antenne filaire à large bande. La raison en est qu'en cours de journée, des signaux puissants dominent sur différentes bandes et pourtant n'occasionnent pas de soucis. La réception des grandes ondes et des ondes moyennes fait exception, parce qu'elle peut subir l'influence de signaux provenant des gammes d'ondes courtes. Mais avec une antenne ferrite et un condensateur variable, l'embarras s'en va.

Il faudra encore approfondir le sujet des antennes, filtres et préamplificateurs, pas juste pour en tirer le plus fort signal possible, mais surtout se prémunir des parasites. Il ne s'agit pas d'une possible longue antenne filaire tendue à distance de la maison, ce n'est pas réalisable partout, alors il faut chercher des compromis. Et en tête de liste, on trouve l'antenne-cadre magnétique. Héritière de l'antenne-cadre à air des pionniers de la TSF, ce genre d'antenne petite et discrète fonctionne à l'intérieur. Nous y reviendrons.

Lors des premiers essais avec ce récepteur, notre attention a été attirée par l'influence d'Arduino : cette carte, si proche de l'entrée, peut-elle produire des parasites dans la réception ? D'origine, elle a été l'objet d'une grande attention au sujet



du découplage. Elle bénéficie d'un plan de masse étendu à la face inférieure et ses tensions d'alimentation de 5 V et 3,3 V sont découplées par filtres LC. Dans les faits, ces mesures se révèlent fort efficaces. Normalement, il n'y a pas lieu de soupçonner Arduino.

Arduino à l'écoute

À tout le moins, on devrait pouvoir capter l'oscillateur d'horloge à 16 MHz. Sans aucune antenne, c'est bien le cas. Alors, le *shield* peut nous servir d'appareil de mesure. En fait, il y a deux oscillateurs. L'un est à quartz de 16 MHz sur la puce USB de l'Uno avec une déviation de moins de 1 kHz. Quand on touche l'endroit où le quartz est soudé sous la carte, on a une petite réaction. On sait donc qu'il s'agit bien de ce signal-là. Avec un petit bout de fil comme antenne, le signal est plus fort, mais le bruit de fond aussi. Et à cette occasion, on remarque encore une bizarrerie. Les signaux qui arrivent par l'entrée d'antenne subissent une bonne atténuation de la fréquence image. Mais pour ceux qui se fauillent par la tension d'alimentation sur le chemin du signal, c'est tout différent. Ils apparaissent en double, quoique nettement plus faibles.

Il faut aller chercher le signal d'horloge du Mega328. Il travaille lui, avec oscillateur

à céramique dont la déviation peut s'élever jusqu'à 50 kHz. Et effectivement, il y a bien un signal faible à 15 950 kHz, avec même des signaux de bande latérale auxquels le contrôleur contribue. Et quand on approche la carte Arduino du résonateur céramique, on obtient encore une large modulation de fréquence et une autre contrariété qui montre que le résonateur présente une certaine dépendance à la température. Là, il nous faut d'urgence un SDR pour aller tâter le pouls d'Arduino !

Quand il reste sans antenne, le SDR augmente normalement son amplification pour rendre visibles les signaux les plus faibles. On voit

alors autour de la fréquence moyenne de faibles parasites produits par l'USB d'Arduino. Pour faire la distinction entre les signaux qui proviennent d'Arduino et ceux de l'USB, branchons un bloc secteur sur Arduino et, après syntonisation sur la fréquence voulue, retirons le câble USB pendant le fonctionnement.

Tous les parasites internes sont très faibles. Dès qu'on branche une antenne, le bruit de fond s'élève au point de couvrir totalement ces parasites. Cela montre la haute sensibilité de la radio logicielle. On peut même recevoir des signaux d'à peine 1 μ V. La plupart du temps, pareille sensibilité est inutile, puisque le niveau de bruit sur l'antenne est bien supérieur. Avec une longue antenne, on peut même sursaturer le récepteur. Il faut déjà songer à un atténuateur d'entrée. ◀

(150515 - version française : Robert Grignard)

Liens

- [1] www.elektormagazine.fr/070039
- [2] https://github.com/adafruit/Adafruit_Si5351_Library
- [3] www.elektormagazine.fr/150515
- [4] <http://airspy.com/download>
- [5] <http://www.hdsdr.de>

indicateur de niveau d'eau pour sac d'hydratation sûrement approuvé par Lonely Planet

Laurent Labbe (France, Chine)

Pour survivre dans la jungle, le désert, New York ou l'Outback, un randonneur a impérativement besoin d'eau potable, si en plus il a un GPS, une carte de crédit et le beau temps, c'est parfait. Le randonneur accompli est donc toujours équipé d'un sac d'hydratation, à savoir un sac conçu pour recevoir une poche d'eau souple avec tube et embout buccal. Malheureusement, avec le réservoir dans le dos, il est difficile de voir combien il reste d'eau. Notre indicateur de niveau à énergie solaire vous permettra d'affronter la *Nature*, qu'il pleuve ou qu'il neige.

Ce projet – baptisé à l'origine *Camel Bag Water Indicator* – a passé les Fourches Caudines d'Elektor Labs, c.-à-d. qu'il a été mis dans les *Pro-*

posals, puis découvert, discuté et évalué par les membres. Il a déménagé ensuite vers *In Progress*, et atterri enfin dans *Finished*, c'est pourquoi vous le lisez maintenant. Lors de l'étape *In Progress*, Luc Lemmens du labo l'a adopté, habillé d'une platine Elektor et fait subir les tests requis pour le rendre reproductible par les lecteurs. Luc a travaillé en étroite collabo-

ration avec l'auteur du projet, Laurent Labbe ; voici les différentes étapes de ce projet.

Éléments déclencheurs et préoccupations

Lors de mes excursions en VTT, j'ai l'habitude de porter un sac d'hydratation (de marque : Camelbak®) pour boire tout en roulant quand il fait chaud.

J'ai toujours été frustré, lors de ces expéditions, de ne pouvoir qu'estimer la quantité d'eau restante.

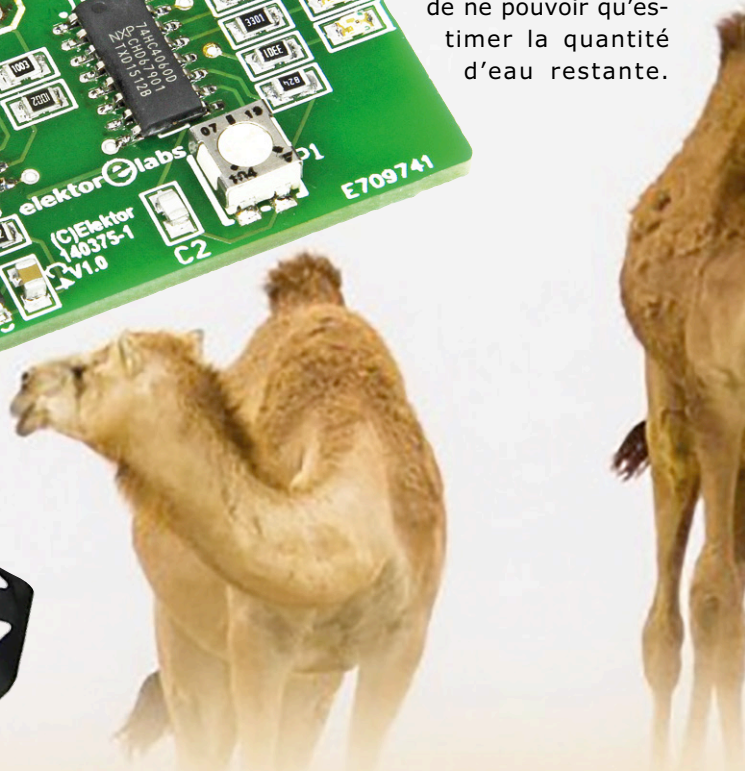


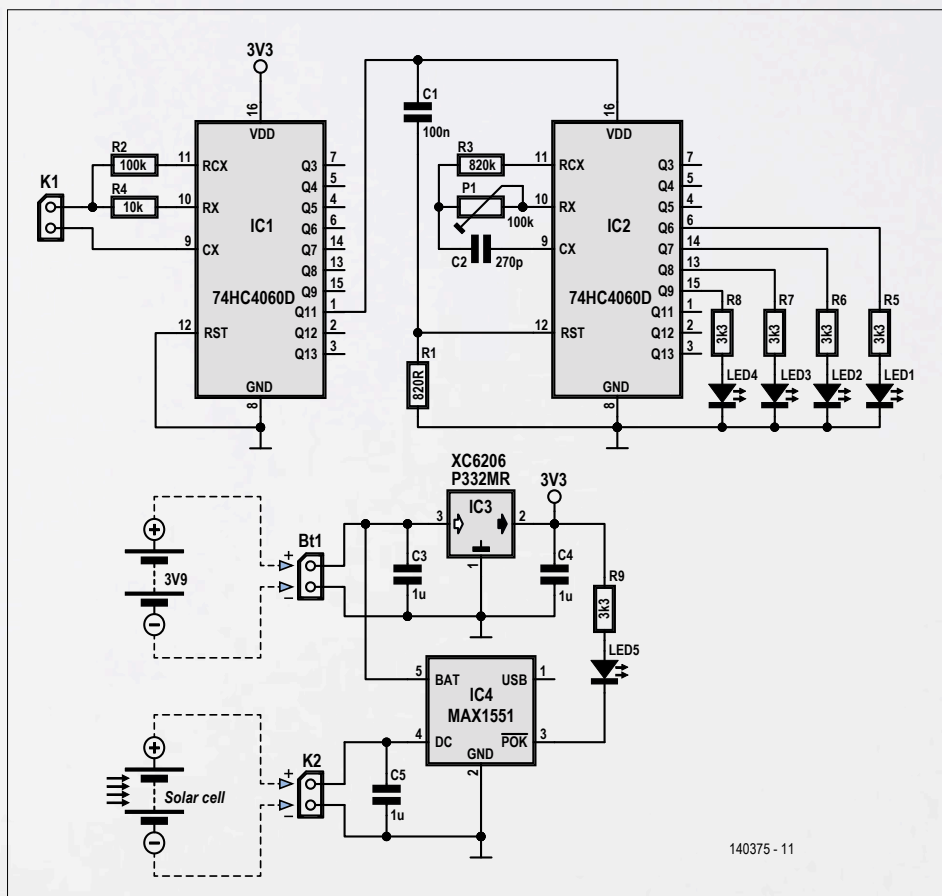
Figure 1. Schéma de l'indicateur de niveau d'eau pour Camelbak®.

Notez comment pour économiser l'énergie, le premier 4060 alimente le second.

Ces sorties m'ont laissé le temps de réfléchir à une solution. Je pensais pouvoir mettre un capteur à l'intérieur de la poche d'eau, voire utiliser un capteur de débit d'eau, mais rien n'était satisfait. Jusqu'à ce que je découvre le détecteur de liquide capacitif de Paul Cordonnier dans son article *Capteur de pluviomètre à siphon* (Elektor, 03/2014 [1]). L'idée de Paul de placer le capteur à l'extérieur de la poche d'eau plutôt qu'à l'intérieur m'a bien plu. Superbe idée, merci Paul.

Pas de fata morgana

J'ai fait des essais avec un oscillateur/diviseur, un 74HC4060 sous 3,3 V (alimentation par pile au lithium) et avec une poche



140375 - 11



Liste des composants

Résistances :

Par défaut : 5%, 0,125 W, 0805

R1 = 820 Ω

R2 = 100 k Ω

R3 = 820 k Ω

R4 = 10 k Ω

R5,R6,R7,R8,R9 = 3,3 k Ω

P1 = pot. ajust., 100 k Ω , CMS

Condensateurs :

C1 = 100 nF 5%, 50 V, 0805

C2 = 270 pF 5%, 50 V, 0805

C3,C4,C5 = 1 μ F 10%, 16 V, 0805

Semi-conducteurs :

LED1,LED2,LED3,LED4,LED5 = LED, verte, faible courant, 0805

IC1,IC2 = 74HC4060 (SOIC-16)

IC3 = XC6206P332MR (SOT23-3)

IC4 = MAX1551

Divers :

Panneau solaire 5 V, 150 mW

Batterie Li+ 4,2 V de téléphone portable

Circuit imprimé 140375-1*

Deux bandelettes de cuivre adhésives, de 15 à 20 cm de long

* Circuits imprimés, modules assemblés et composants programmés disponibles en ligne (www.elektor.fr)

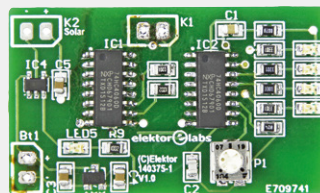
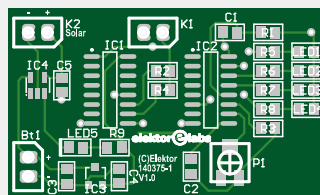


Figure 2. Comme le montage doit être portable, chaque gramme/millimètre compte. Le circuit imprimé a donc été conçu pour accueillir des CMS.

en plastique remplie d'eau puis vide. Ça a marché ! La fréquence changeait. J'ai pris la sortie Q11 du 4060 (broche 1) ; poche vide, la fréquence était de 150 Hz, poche remplie de 60 Hz.

Voyons le schéma de la **figure 1**. Comment afficher les « données » fournies par le capteur capacitif-maison relié à K1 ? Comme la fréquence sur Q11 est plus faible poche pleine, les « 1 » logiques sont plus longs. J'ai eu l'idée d'utiliser ce signal pour lancer un autre 4060 en compteur : les sorties du diviseur 128/256/512/1024 donneraient une mesure du niveau d'eau avec des LED.

L'entrée RST (CLR) du 4060 est active au niveau « haut » ; pour économiser une énergie précieuse, le 4060 lecteur du capteur (IC1) alimente le 4060 de l'affichage (IC2). Avec une impulsion à la mise sous tension pour initialiser les compteurs, cela marche à condition de choisir une fréquence d'horloge proche de 30 kHz.

Les LED à haut rendement avec les résistances de 3,3 k Ω (R5-R8) permettent une lecture même en plein soleil. Le cap-

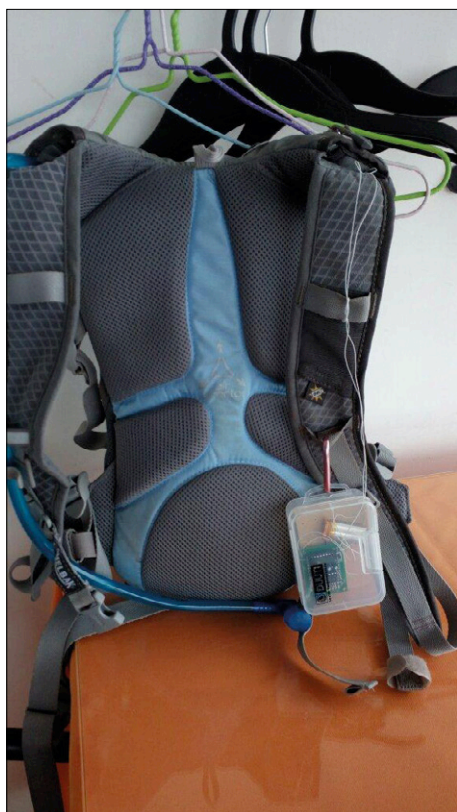


Figure 3. Le prototype de l'indicateur de niveau d'eau de l'auteur, suspendu sur le devant de son sac à dos.



Figure 4. La poche à eau sur laquelle est fixé le capteur capacitif. Vous trouverez ce type de poche dans les magasins pour matériel de randonnée et de plein air.

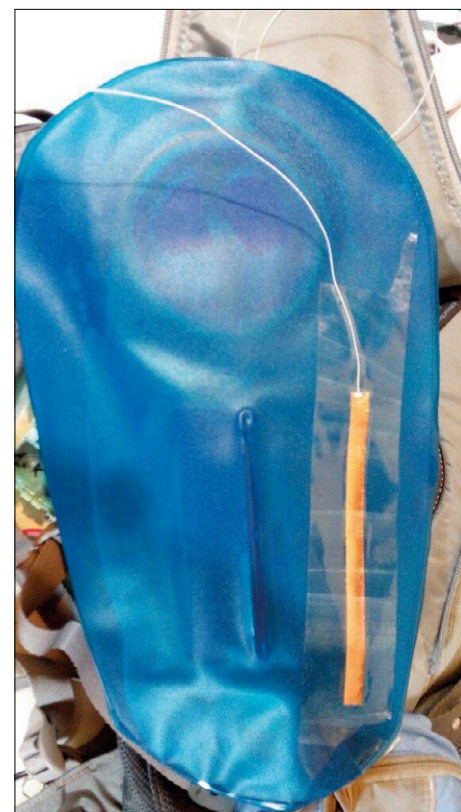


Figure 5. Le capteur de fabrication maison fonctionne tout aussi bien monté sur l'autre côté du réservoir.

teur est constitué de deux fines bandes de cuivre adhésives, disposées parallèlement et fixées verticalement, sur l'extérieur du réservoir d'eau.

Pour l'étalonnage, il est utile que l'horloge du second 4060 soit réglable. En effet il est délicat de trouver le bon réglage de P1 pour les détections « vide » et « plein ». Le prototype a pourtant bien fonctionné, avec une précision suffisante pour mes besoins. J'ai glissé le tout dans un coffret transparent sur mon Camelbak® – ultra-léger et indispensable.

Plus tard, pendant la mise au point du projet chez Elektor, j'ai pensé à ajouter un panneau solaire pour recharger la batterie LiPo, puis à utiliser un microcontrôleur pour un étalonnage automatique des points « vide » et « plein ». La première de ces idées a été mise en œuvre, la seconde est toujours à l'étude.

Le circuit est alimenté par une petite batterie LiPo (150 mAh), rechargée par une cellule solaire 5 V/50 mA via un circuit intégré chargeur dédié, un MAX1551.

Sur mon prototype, j'ai constaté que la batterie s'était chargée de 3,3 V à 3,7 V après une exposition de quelques heures

au soleil. Ajouter une autre cellule augmenterait le courant. Après une petite sortie (deux heures de vélo), en profitant du soleil en fin d'après-midi, j'ai mesuré 3,72 V ; la batterie avait donc le même niveau de charge qu'au départ. Ainsi, bien que les conditions n'aient pas été favorables, aucune décharge de la batterie. La consommation globale est de l'ordre de 3 mA ; l'autonomie de fonctionnement à long terme ne doit donc pas poser de problème. Pour signifier, on pourrait ajouter une prise externe pour un chargeur USB.

Réalisation

Vu les tailles de la platine et des composants, le montage au fer à souder classique sur le circuit imprimé (**fig. 2**) semble un défi. Luc Lemmens a prouvé, à force de patience et grâce à sa dextérité, qu'il est possible de placer correctement ces minuscules composants pour les souder ensuite. La mise en coffret de l'indicateur de niveau d'eau pour Camelbak® et la réalisation du capteur demandent de l'attention ; consultez les photos du prototype de l'auteur (**figures 3, 4 et 5**).

Assurez-vous que tout est bien fixé dans et sur le coffret ; vous pouvez y aller. Le premier randonneur ou pédaleur (H/F) à arriver au siège d'Elektor [2] avec un *Camel Bag Water Indicator* fonctionnel et ce à une température extérieure supérieure à 30 °C (86 °F) se verra offrir la visite du labo d'Elektor et celle du puits dans le sous-sol du château, ainsi qu'une règle Elektor Labs. Date de clôture de la compétition : le 1^{er} octobre 2016. Sont hors compétition les employés d'Elektor International Media, ses filiales, ses titulaires de licence, et/ou les maisons d'édition associées. ❏

(140375 – version française : Guy Raedersdorf)

Liens

- [1] Capteur de pluviomètre à siphon, Elektor, mars 2014, www.elektormagazine.fr/120554
- [2] @ 51.016224, 5.838898; lundi-jeudi 08h – 17h, vendredi 08h – 13h. Sonnez à la porte !

EXPÉRIENCE, COMPÉTENCE, IDÉES, TALENT & ENVIES

Vous avez de l'expérience, des compétences, des idées, du talent et des envies d'aller de l'avant dans la jungle des réseaux et des protocoles ?

Les fauves et les moustiques appelés *LoRa*, *SigFox*, *Neul*, *Qowisio*, *ThingSpeak*, *Brillo*, *Nest Weave*, *Thread*... ne vous effrayent pas ? Ce ne sont pas ces quelques réseaux à bas débit sur longue distance qui vous font perdre le nord ?

Pour vous orienter dans le monde de l'Internet des Objets, vous avez déjà pratiqué une ou plusieurs cartes d'expérimentation : Elektor vous invite à partager vos connaissances maintenant en publiant **votre livre !**

Forts de notre expérience d'éditeur, nous vous accompagnerons dans l'écriture de votre ouvrage et nous chargerons de la mise en page, de l'impression, de la commercialisation en librairie et bien sûr sur l'internet (notamment chez Amazon).

La relation éditeur-auteur est encadrée par un contrat d'édition qui stipule vos droits d'auteur sous la forme d'un pourcentage déterminé du prix de vente.

Présentez-nous votre projet en nous envoyant une description et un projet de sommaire par courriel à **redaction@elektor.fr**.

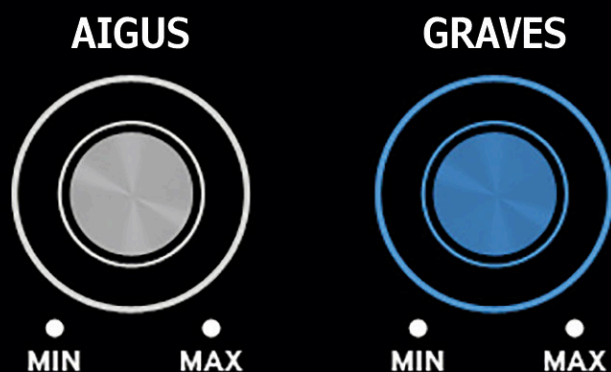


iBaxandall

variations sur le correcteur de tonalité

Thomas Scherer (Allemagne)

Vous devez construire en vitesse un correcteur de tonalité, trois possibilités s'offrent à vous : chercher sur l'internet un circuit classique, en trouver un qui vous séduit ou vouloir faire mieux et concevoir une nouvelle variante du circuit. Et pourquoi ne pas partir d'une page blanche pour créer du neuf, sortir des sentiers battus ? Alors, on s'amuse ?





En deux mots, je voulais ajouter à un simple ampli de puissance numérique – je sais, c’est diabolique – un réglage de tonalité tout aussi simple. Aujourd’hui on peut le faire aussi en numérique, mais la solution est à la fois chère et plus lourde, d’autant que je ne disposais d’aucune puce adéquate dans mes réserves. Alors, ce sera analogique. Même pour un ampli stéréo, il n’y faut qu’un double ampli op et quelques composants passifs, plus les potentiomètres. J’ai bien en tête le schéma traditionnel, mais par sécurité, je vais voir sur la Toile. Ce qu’il en est ressorti est assez surprenant, parce que je n’aurais jamais imaginé trouver autant d’homonymes de ce brave ingénieur britannique Peter J. Baxandall qui a publié ce filtre en 1952 et pour lequel l’*Audio Engineering Society* lui avait déjà en 1950 attribué comme prix une montre à 25 \$. Voyez vous-même la liste sur votre moteur de recherche préféré.

Donc, on est sur du velours ? Une fois de plus, Radio Erevan vous répondra : en principe oui, mais...

Le classique réglage de tonalité

Interrogez Google sur « *tone control* », vous aurez près de quatre millions de résultats. Le circuit passif est très prisé, mais vu son grand affaiblissement, je préfère ne pas en entendre parler. Passons au filtre actif. En comparant les nombreux circuits actifs, on voit qu’ils ne se distinguent que par des différences marginales par rapport à celui du haut de la **figure 1**. Par canal, il ne faut qu’un amplificateur opérationnel, deux potentiomètres, un pour le grave et un pour l’aigu, deux condensateurs et cinq résistances. En version stéréo, un double ampli op suffit.

Des amplis op aux caractéristiques appréciées des audiophiles, par ex. la puce double NE5532 [1] ou la quadruple LM837 [2], sont de nos jours très intéressants. Quand on envisage d’en obtenir le niveau de bruit minimal, il faut les maintenir à basse impédance de sortie, mais pour la plus faible distorsion, c’est une haute impédance qui leur convient, il y a donc un compromis à trouver qui se situe entre 2 k Ω et 10 k Ω . Les valeurs de composants indiquées à la figure 1 fournissent une plage de réglage autour de ± 12 dB et des fréquences de coupure à -6 dB de 100 Hz pour les basses et 5 kHz pour les aigus. Tout est dit.

Il y a comme un défaut

Aussi, l’histoire aurait-elle pu s’arrêter ici, avant même d’avoir commencé. Tous les composants calculés et choisis, et avec un quadruple ampli op, on a déjà un correcteur de tonalité stéréo à faible distorsion et bas bruit plus un réglage de volume. Mais avant la réalisation d’un circuit avec des composants réels, l’électronicien moderne s’offre un passage intermédiaire virtuel : j’ai pratiqué une simulation du circuit traditionnel avec différentes valeurs de résistances pour les deux potentiomètres. Pour un projet aussi simple, j’ai choisi le simulateur en ligne PartSim, proposé par le distributeur Digikey, d’emploi très aisé [3]. Il comprend un éditeur de schéma et un simulateur à configurer.

La moitié basse de la figure 1 montre la courbe de fréquence et de phase du circuit standard. Ici comme dans les autres graphes, l’amplitude du signal est de 1 V. Les simulations sur différents réglages des potentiomètres de grave et d’aigu révèlent qu’ils s’influencent mutuellement. Rien d’exceptionnel, puisque l’entrée inverseuse de l’ampli op est reliée à une cel-

lule RC commune aux deux branches, elle ne peut faire aucune distinction précise entre grave et aigu. Cette dépendance n’est pas vraiment grave, mais pour un usage domestique, elle peut influencer un autre paramètre, plus important : l’impédance de charge de l’amplificateur opérationnel ne reste pas dans la fourchette de 2 à 10 k Ω . Ces potentiomètres peuvent faire varier la charge nettement au-delà de part et d’autre. En dessous de 2 k Ω , l’entrée aussi en souffre, à moins d’y adjoindre un ampli op comme tampon ou un réglage de niveau. Les valeurs indiquées sont donc un compromis acceptable, mais pas idéal. La question est : peut-on faire mieux ?

Améliorations

Dès qu’on envisage des réformes, les idées s’enchaînent. Lorsqu’on conçoit un logiciel, on décide parfois de tout réécrire depuis le début. Ici c’est pareil, il faut créer un schéma radicalement différent. Les embarras à propos de l’impédance et de l’interaction d’un réglage sur l’autre, on peut les éliminer en séparant les deux voies. Ce qui donne la variante préférable

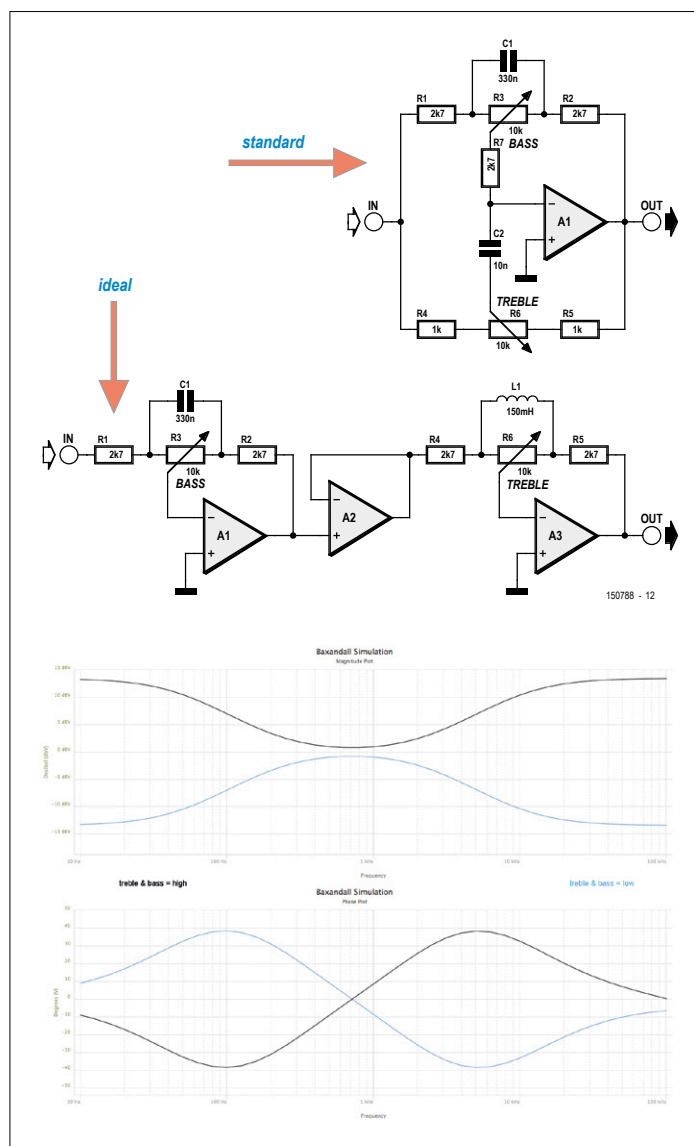


Figure 1. Le schéma standard (en haut) et la variante idéalisée (en bas) d’un correcteur de tonalité avec leurs caractéristiques de fréquence et de phase.

(ideal) du bas de la figure 1, avec l'ajout du tampon A2 pour décharger la sortie d'A1.

Si vous poussez cette méthode radicale encore plus loin, alors, spécialement dans l'aigu, débarrassez-vous de cette béquille en forme de filtre passe-haut CR et, pour atteindre le fin du fin, rendez le tout symétrique en fréquence : installez la bobine L1 là où se trouve le condensateur C1 pour le grave. Au point de vue technique, c'est très élégant. Malheureusement, la réalité barre cette idée géniale d'un trait de plume : la valeur de 150 mH donnée à L1 est en pratique illusoire au point de bousiller de nouveau la belle symétrie. Imaginez la bobine qui aurait cette inductance : une dimension monstrueuse. Et qui plus est une bobine ! Ces composants électromagnétiques sont vraiment détestables. Non seulement ils sont obèses, mais sont affublés d'une méchante résistance série, se font parasiter par n'importe quel champ magnétique et si vous en choisissez avec noyau, leur distorsion les excluent du casting pour une application audio. Ils n'arrivent même pas à la cheville des condensateurs modernes.

Le gyrateur

Que faire ? Enterrer le projet ? Pas si vite, l'électronique a une solution à (presque) toutes les situations. Elle porte ici le joli nom de gyrateur. Il s'agit d'un convertisseur d'impédance [4] particulier, capable de transformer le comportement d'un condensateur en celui d'une bobine. Le plus beau de l'histoire, c'est qu'il ne réclame qu'un seul amplificateur opérationnel supplémentaire.

À gauche dans la **figure 2**, un gyrateur est construit autour d'A1 qui, avec R3 et un réseau RL, forme un filtre passe-haut. Le circuit est tout simple et son calcul l'est tout autant. Avec les valeurs indiquées pour C1, R1 et R2, on arrive à une inductance de 100 mH, exactement dans la gamme souhaitée. Les impédances de C1 et R2 tombent aussi à pic. La faible valeur de R1 ne dérange pas, puisque la charge de sortie de l'ampli op s'établit à $R1 + R3 \parallel (C1 + R2)$. Avec 2,7 kΩ pour R3, on obtient un beau passe-haut dont la fréquence limite vaut $f_g = R / (2 \pi L) = 4,3$ kHz.

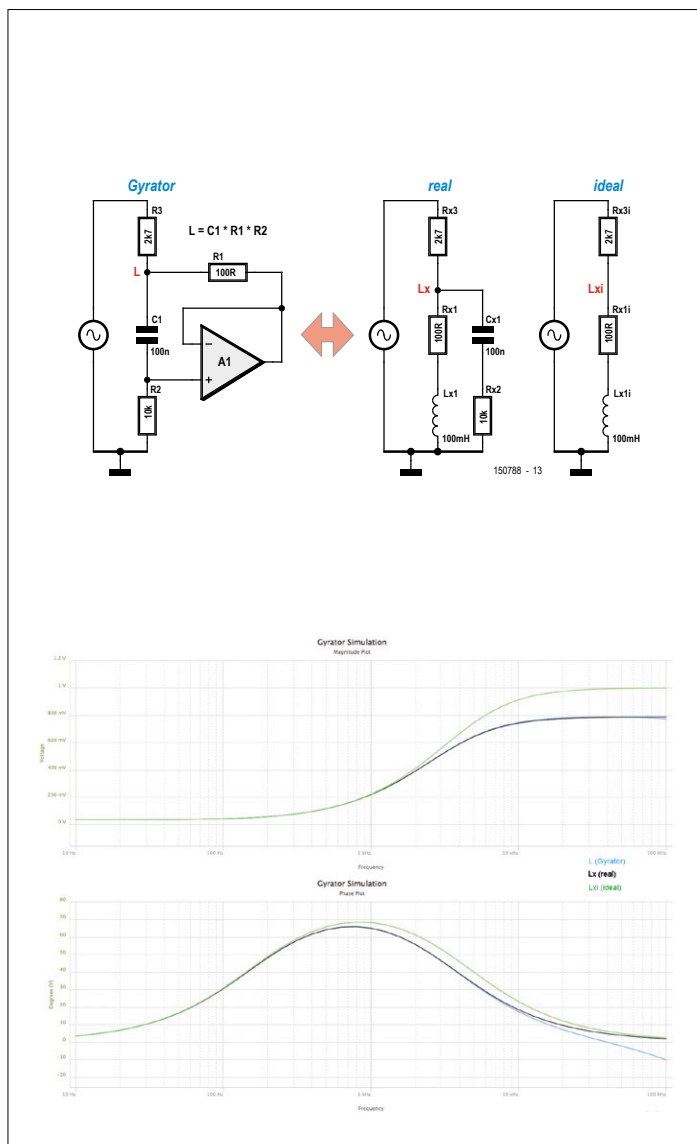


Figure 2. Le gyrateur classique comme filtre passe-haut RL (à gauche), à côté des schémas équivalents réel (au milieu) et idéalisé (à droite) ainsi que leurs caractéristiques de fréquence et de phase.

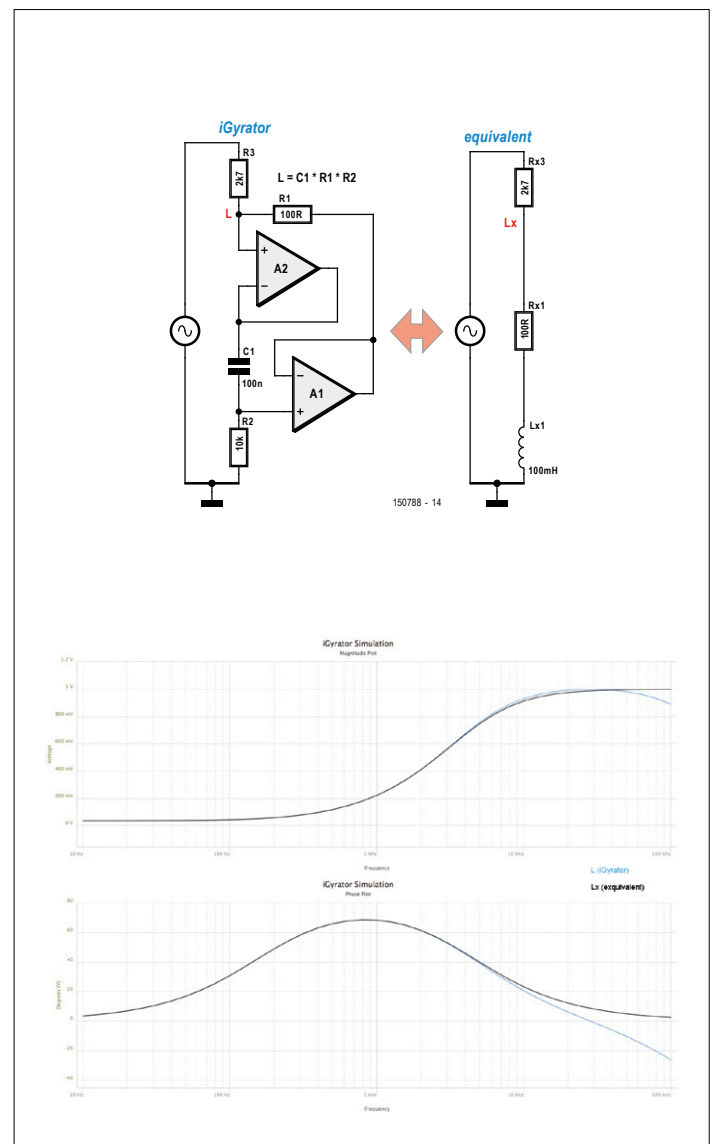


Figure 3. Un iGyrator est déjà meilleur, ici comme filtre passe-haut RL (à gauche) avec son schéma équivalent et ses caractéristiques de fréquence et de phase.

Mais l'inquiétude ne s'arrête pas là : l'inductance réalisée par le gyrateur présente une résistance série d'exactement $R1$. Ce n'est pas encore grave, puisque comparé à la grande valeur obtenue pour la bobine, $100\ \Omega$ est une quantité négligeable ; pour ce qui est de l'impédance du circuit et de la valeur de $R3$, cela n'occasionne qu'une erreur minime dans la caractéristique de fréquence. Ce qui est beaucoup plus grave, c'est cette mise en parallèle sur le circuit idéal de la bobine entre le point L et la masse, de la cellule RC composée de $C1$ et $R2$. Et cela influe très nettement sur la réponse en fréquence et le reste.

Pour clarifier la situation, j'ai utilisé PartSim pour tracer la réponse en fréquence et en phase de la cellule RL avec le gyrateur, mais aussi les courbes de son fonctionnement réel (au milieu) et le circuit équivalent idéalisé sans la cellule RL (à droite). On voit bien que le circuit du gyrateur et le circuit réel de remplacement sont pratiquement identiques. La déviation de la phase en haute fréquence est simplement due à la réponse en fréquence limitée de l'ampli op utilisé dans la simulation et

pas vraiment importante au point de vue audio. On voit aussi que seul le circuit équivalent idéalisé se comporte comme il faut. Confirmation également que l'on peut négliger $R1$.

iGyrateur, fGyrateur et fGyrateur2

Pour contrecarrer le nouveau défaut découvert, la perturbation du réseau RC avec $C1$ et $R2$, l'électronique dispose aussi d'une solution. Un autre ampli op monté en tampon peut simplement isoler le réseau RC du point L. Le résultat est à la **figure 3**. Comme il s'agit d'un gyrateur « amélioré », je l'ai appelé **iGyrator**. Comme on peut le voir, le circuit équivalent correspond maintenant exactement à celui idéalisé de la figure 2. Les courbes de fréquence et de phase de l'iGyrateur et du circuit équivalent coïncident largement dans le domaine audio et sont encore plus proches avec des amplis op rapides. Et pourtant, tout n'est pas encore résolu, du fait que le gyrateur ne peut hélas fonctionner qu'avec le point L tributaire de la masse. Comparé au circuit idéalisé de la figure 1, on voit que l'inductance $L1$ doit être à potentiel flottant si l'on veut réel-

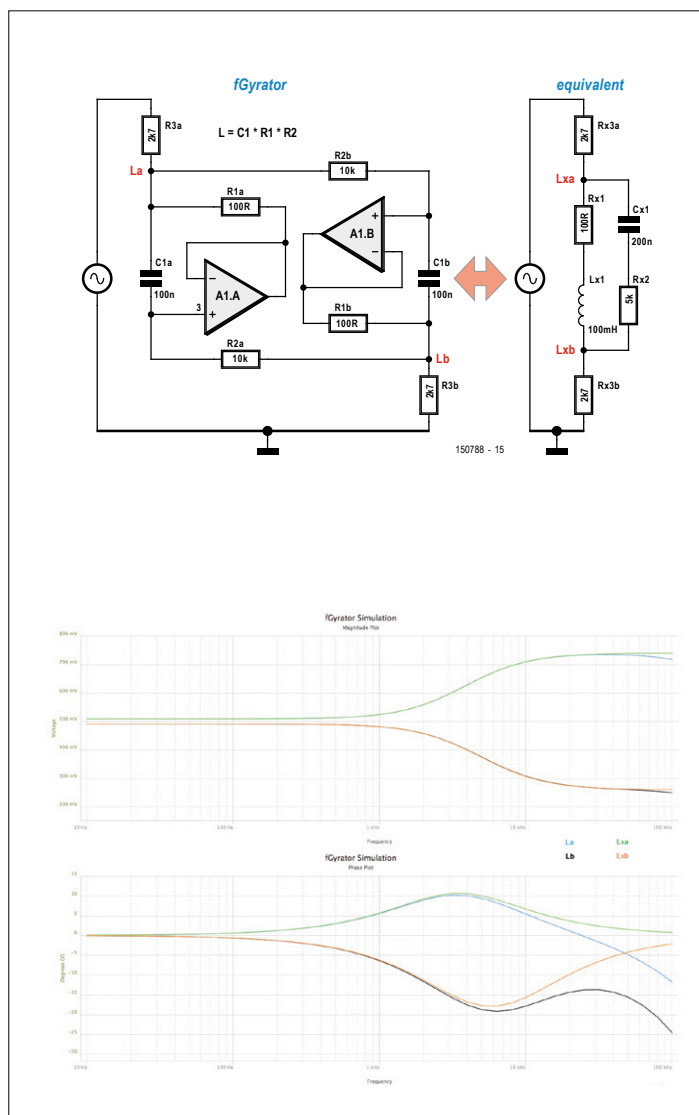


Figure 4. Avec deux gyrateur antiparallèles, on obtient le fGyrator flottant que l'on peut installer dans un circuit à la place d'une vraie bobine. L'ennui, c'est qu'il augmente la charge de la cellule RC parasitaire par rapport au gyrateur simple.

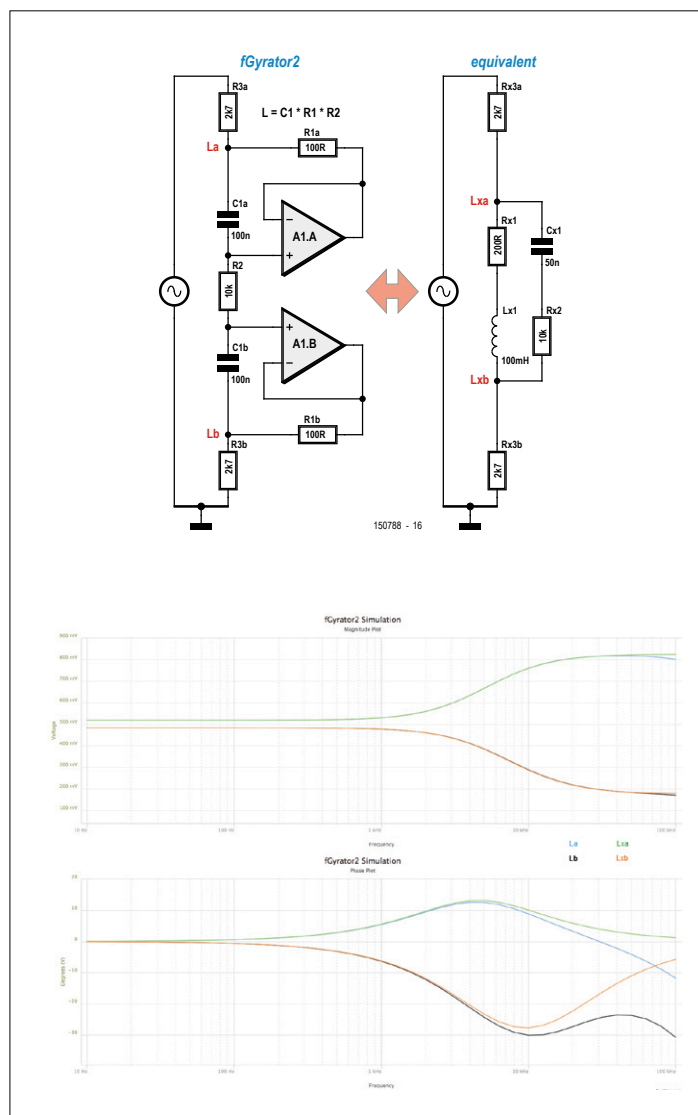


Figure 5. Au lieu de monter les gyrateurs en antiparallèle, les opposer en série donne le fGyrator2. La charge de la cellule RC baisse encore, mais la résistance série de l'inductance virtuelle augmente.

lement la remplacer par un gyrateur. Pour y parvenir, il existe une solution très simple, monter deux gyrateurs identiques tête-bêche (antiparallèle). Le circuit **fGyrator** de la **figure 4** est un modèle flottant et fonctionne réellement comme la simulation le montre. Les deux cellules RC formées de C1a et R2a ainsi que C1b et R2b se trouvent entre les terminaisons des inductances virtuelles La et Lb. Il en résulte aussi que les deux cellules RC sont en parallèle sur les inductances virtuelles. Le schéma équivalent en tient compte : Cx1 affiche le double et Rx2 la moitié de la valeur. En regard d'un iGyrator, le fGyrator présente bien l'avantage d'avoir l'inductance à potentiel flottant, mais il faut aussi considérer l'inconvénient de l'influence des deux cellules RC ensemble qui est le double de celle du premier gyrateur.

Ne pourrait-on pas obtenir cet état de potentiel flottant par une autre voie ? Il est évidemment possible de remplacer le montage tête-bêche par une mise en série inversée (antisériel). La **figure 5** montre la variante qui en résulte, le **fGyrateur2**, avec même un composant de moins, puisque R2 sert

à la fois dans les deux gyrateurs. Ce qui est intéressant, c'est que de ce fait-là, la capacité effective de Cx1 dans le schéma équivalent en est quadruplée par rapport au fGyrator, et que Rx1 reste doublée. Du coup, la résistance série de l'inductance Rx1 est aussi doublée, mais ce n'est pas grave. En somme, le fGyrateur2 apporte une amélioration, mais on ne s'est toujours pas débarrassé des réseaux RC en parallèle.

ifGyrateur, fGyrateur2 et iTreble

Pour conserver un gyrateur flottant sans le souci des cellules RC dérangeantes, il faut marier les deux principes et on découvre ainsi le circuit du **ifGyrator** (*improved floating Gyrator*) de la **figure 6**. Par rapport à son prédécesseur, l'influence des cellules RC est éliminée et la caractéristique de fréquence a aussi glissé vers la droite en direction de la valeur souhaitée. Le prix à payer pour ce gyrateur presque idéal et à potentiel flottant avec deux autres tampons est qu'il faut maintenant quatre amplificateurs opérationnels, mais n'oublions pas qu'ils tiennent tous les quatre dans le même boîtier et qu'en CMS, la

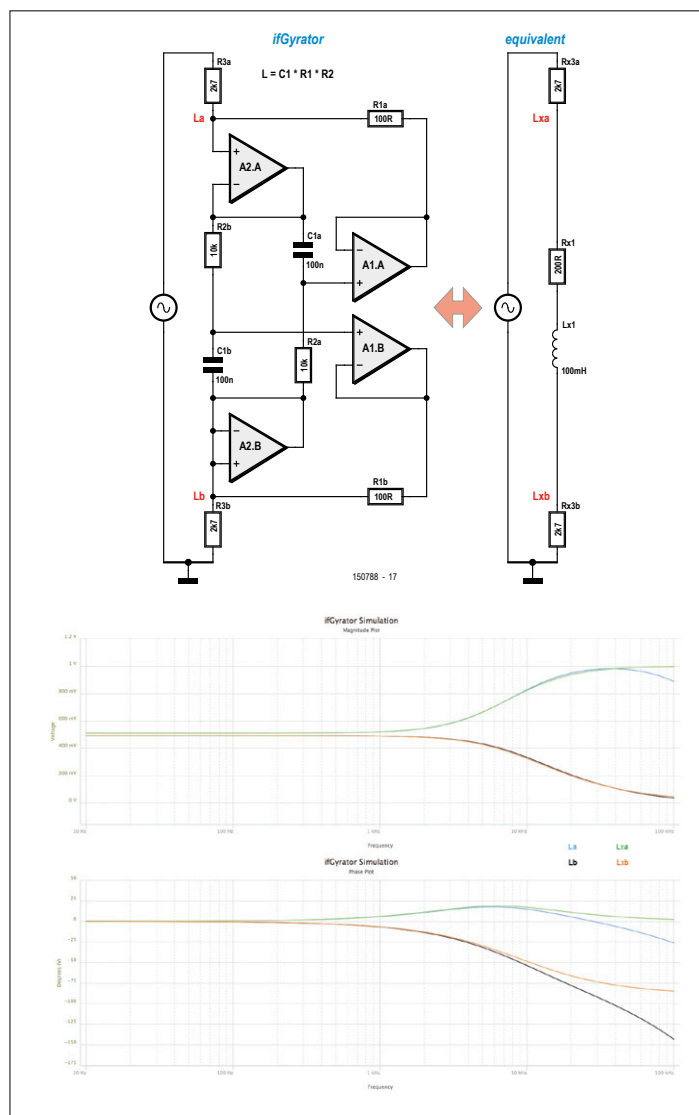


Figure 6. Combinons maintenant un iGyrator avec un fGyrator, nous obtiendrons un ifGyrator, un circuit qui fonctionne comme gyrateur à potentiel flottant et sans la charge parasitaire de la cellule RC.

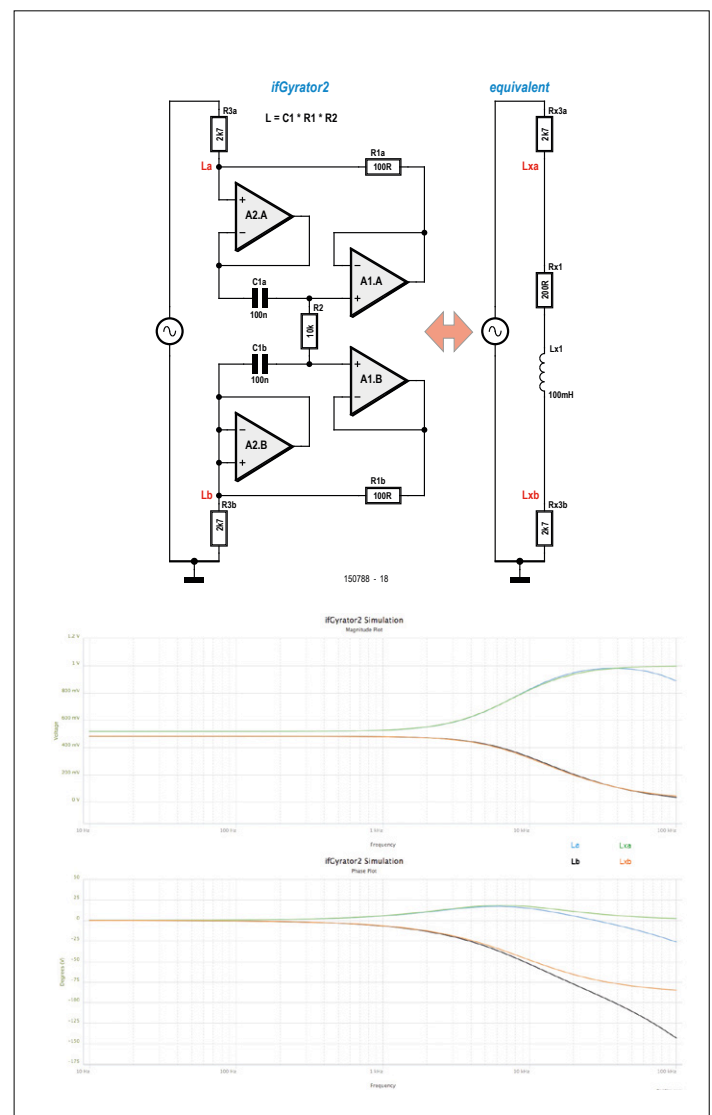


Figure 7. La combinaison d'un fGyrator2 à série inversée et d'un iGyrator donne un ifGyrator2. Une résistance de moins que l'ifGyrator, mais une résistance série plus grande.

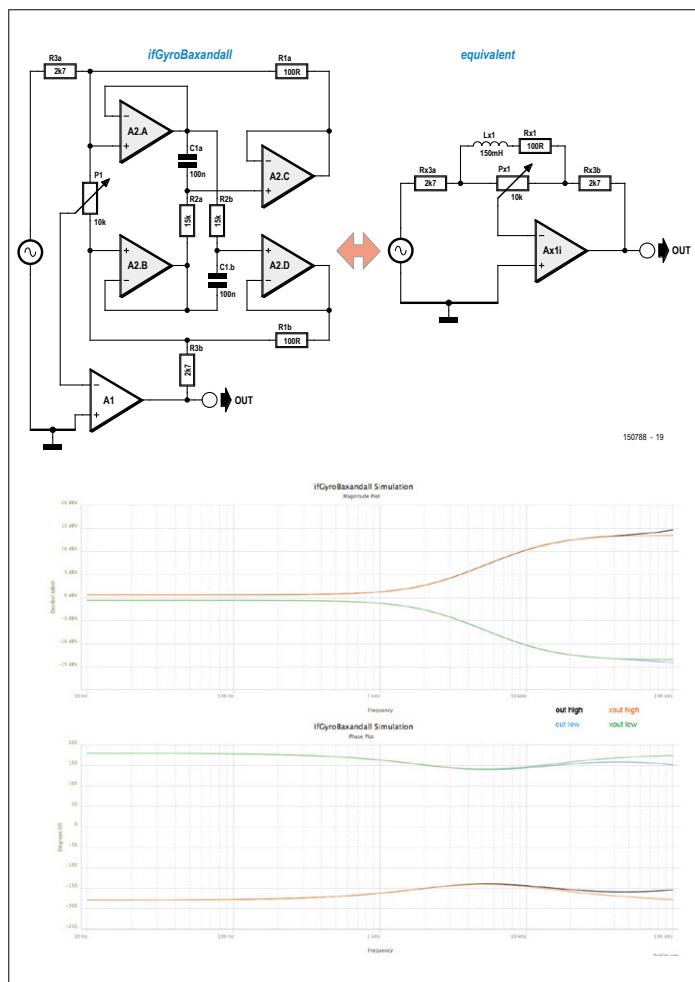


Figure 8. Dans le correcteur d'aigu idéalisé, vous remplacez l'inductance par un ifGyrator et vous avez le circuit de l'*improved Treble*. Petite dépense, grands effets.

superficie de pareil gyrateur est relativement minime.

Vous l'aviez déjà deviné, il y a aussi une deuxième variante avec le schéma de base antisériel, la **figure 7** vous présente l'**ifGyrator2**, dont le principal avantage en comparaison du

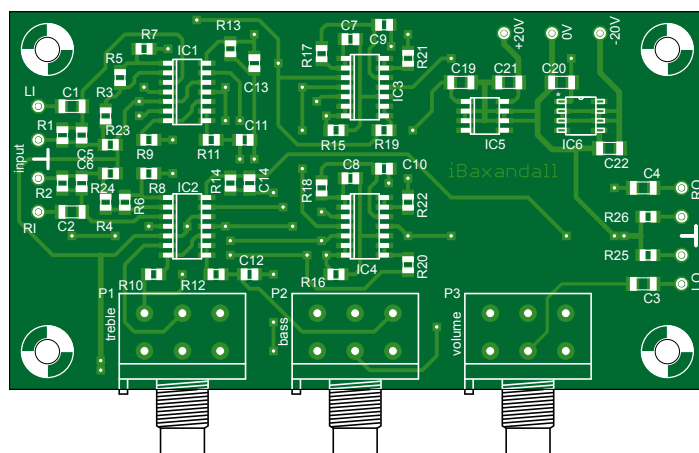


Figure 10. Le circuit imprimé pour cet iBaxandall offre de nombreuses possibilités d'expériences théoriques et pratiques.

premier ifGyrator est qu'il ne lui faut qu'une seule R2. En revanche, la résistance série en est doublée, l'économie n'est pas très payante. Si l'on voulait ainsi remplacer la bobine L1 du schéma idéal de la figure 1 par un gyrateur, on se tournerait vers le fin du fin, l'ifGyrator, parce qu'on n'en est plus à une résistance près.

Si maintenant nous utilisons l'ifGyrator pour remplacer le réglage d'aigu, on arrive alors au **iTreble**, le schéma amélioré de la **figure 8**.

On voit que les courbes de fréquence lors de l'accentuation et de l'affaiblissement des aigus couvrent largement le même domaine que le circuit équivalent avec une vraie bobine. C'est la preuve qu'il est possible de réaliser un correcteur de tonalité Baxandall bien symétrique en pure électronique et absolument sans inductance réelle. Même dans l'allure de la phase, la différence dans les hautes fréquences reste dans les marges, elle s'améliore encore avec les amplis op plus rapides, parce que la simulation a été réalisée avec des exemplaires standard d'une largeur de bande de 10 MHz.

iBaxandall

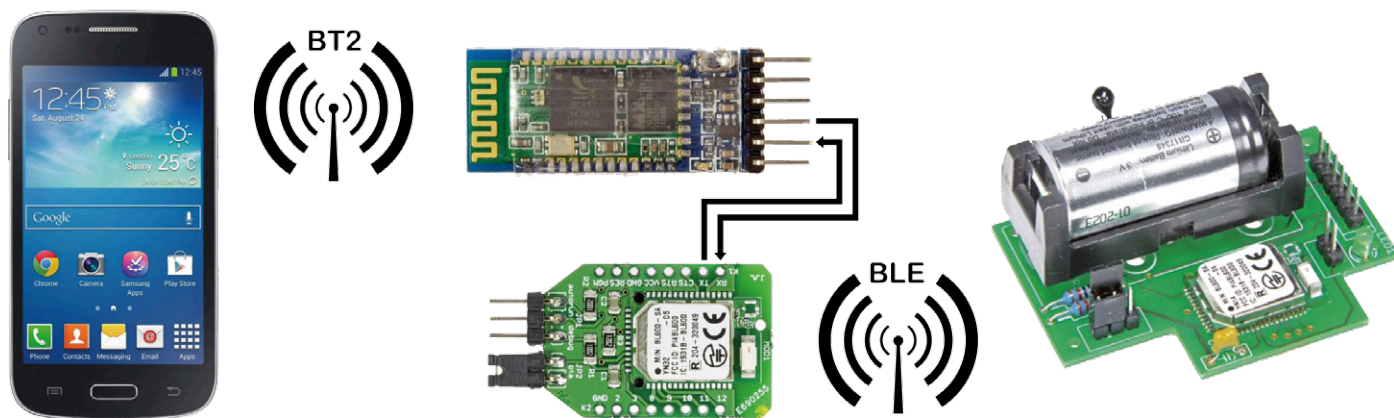
Résumons les connaissances techniques des circuits engrangés jusqu'ici pour faire la synthèse du correcteur de tonalité amélioré **iBaxandall** de la **figure 9**, dans lequel les réglages de grave et d'aigu ne s'influencent pas l'un l'autre. Le circuit en stéréo totalise, avec la commande de volume et les amplis tampons IC1A, IC1C, IC2A et IC2C, le nombre impressionnant de seize amplificateurs opérationnels ! C'est loin d'être un record, surtout quand on considère qu'avec quatre puces (régulateurs de tension mis à part) d'amplis quadruples on en a fait le tour. Le projet vaut-il la dépense ? À chacun de voir. En tout cas, la preuve est faite que, même avec des circuits analogiques, on peut encore faire des découvertes. J'en ai été moi-même étonné, au cours de toutes ces pérégrinations, en cherchant à améliorer un vieux circuit basique. Voulez-vous l'essayer ? Tout ce projet tient sur le circuit imprimé de la **figure 10** dont les mensurations ne font que 92 x 50 mm, potentiomètres compris.

Avec les valeurs de composants indiquées, le réglage s'étend sur ± 12 dB à 40 Hz d'un côté, 10 kHz de l'autre. Le gain d'ensemble est de deux, vous pouvez le modifier en adaptant les valeurs de $(R3+R5)/R3$ et de $(R4+R6)/R4$. Avec ce gain de deux et les commandes d'aigu et de grave complètement hautes, pour un niveau de signal normal (0 dB = 775 mV), une tension d'alimentation de ± 15 V laisse suffisamment de marge avant l'écroulement, la distorsion reste faible, pour autant que les étages suivants et les haut-parleurs soient sur la même longueur d'onde. Donner aux résistances R15 à R18 la valeur de 15 k Ω maintient encore une configuration à faible bruit, il y a là les condensateurs C7 à C10 en parallèle pour conduire l'alternatif. Ces valeurs produisent une inductance des gyrateurs de 150 mH, que demander de plus ?

Soit dit en passant, j'ai volontairement omis un réglage de la balance. Quand on veut écouter de la musique en haute qualité, on a pris la précaution d'orienter les enceintes et les sièges convenablement. En définitive, on ne regarde pas la télévision sous un angle de 45° ! J'attire votre attention sur les condensateurs qui déterminent les fréquences, il s'agit de C7 à C12 qui doivent être de bonne facture. Les conden-

passerelle Bluetooth 2 vers Bluetooth LE

ou comment consulter le thermomètre BLE avec un ordiphone Android un peu âgé



Qui n'a jamais rencontré un problème de température au bureau ? Il y fait trop chaud, trop froid... J'y ai donc installé mon *thermomètre Bluetooth Low Energy* (BLE) [1] que mes collègues peuvent consulter avec leurs ordiphones compatibles BLE sous Android ou iOS (mais pas mon chef avec son Windows Phone ;-). Ainsi il n'y a plus de débat sur la température et on règle la climatisation dans un sens ou l'autre. Bien sûr, il a fallu qu'un collègue possède un téléphone un peu vieux, incompatible avec le BLE... Alors, que faire ?

Jennifer Aubinais (Paris, France) - elektor@aubinais.net

Cohabitation de BT2 et BT4

Nous vous proposons dans cet article de réaliser une passerelle Bluetooth 2.0 (BT2) vers Bluetooth Low Energy (BLE). Cela a l'air facile aux premiers abords, mais il ne suffit pas de connecter un module BT2 à un module BLE (via une liaison série par ex.). Un véritable protocole de communication, une sorte de « tunnel », est nécessaire pour faire dialoguer les deux normes. Bien sûr, il aurait été plus simple de redessiner le thermomètre en version BT2, mais j'ai choisi une autre voie. Pourquoi ? La principale caractéristique du BLE (version 4.0 et suivantes) est sa faible consommation par rapport au Bluetooth classique, ou au Wi-Fi également, ce qui permet de réaliser un thermomètre sans fil simplement alimenté par une pile. Comme le BLE est peu gourmand en énergie, on le trouve aujourd'hui dans les objets connectés (traqueurs d'activité physique, montres connectées, etc.). Le BLE est une évolution majeure de la norme par rapport au BT2 et les deux ne sont pas compatibles. Cependant le Bluetooth classique reste présent dans nos périphériques en version 2.1+, car il possède un débit supérieur au BLE, ce qui est fort utile pour la transmission de musique par exemple.

Le BLE est surtout employé pour l'envoi de messages courts comme la valeur d'une température, un rythme cardiaque, etc. Les ordiphones récents sont capables de gérer les deux normes, c'est ce qu'on appelle le *Dual Mode*.

L'électronique

La partie matérielle de la passerelle est simple, il s'agit de l'association d'un module BT2, ici un HC-06, et d'un module BLE BL620. Le BL620 est l'e-BoB BL600 longuement décrit en 2015 [2], mais reprogrammé avec un micrologiciel spécifique pour lui donner le rôle de maître (voir encadré pour la procédure). Un tel module peut se connecter à quatre modules esclaves, comme le thermomètre BLE. Comme le module BLE doit être alimenté en 3,3 V, un petit régulateur LP2950-33 est placé sur la carte, avec ses deux condensateurs de 100 nF. De plus, nous avons ajouté une LED qui indique l'état de la liaison entre le module BLE et le thermomètre. Reportez-vous au schéma (**figure 1**) pour les détails.

Le protocole de communication

Le BT2 ne peut pas communiquer directement avec le BLE, c'est pourquoi une passerelle est nécessaire. Dans notre cas nous voulons lire le thermomètre BLE avec un téléphone qui

ne « parle » que le BT2. La passerelle doit donc traduire les commandes BT2 envoyées par le téléphone en commandes BLE et inversement.

La **figure 2** montre les couches du protocole BLE : c'est plutôt complexe ! Les commandes BT2 doivent traverser toutes les couches de bas en haut pour arriver au thermomètre. Plusieurs chemins semblent possibles, mais le plus commode est de creuser un tunnel à droite sur la figure, en passant par HCI et GATT. (Pour ce qui suit nous avons simplifié la description pour garder uniquement l'essentiel. Pour plonger au cœur du BLE et de ses couches, consultez l'internet.)

La lecture de la température mesurée par le thermomètre BLE est décomposée en plusieurs étapes :

1. Recherche du thermomètre
2. Connexion
3. Lecture des données
4. Déconnexion

La recherche du thermomètre et l'établissement de la liaison Bluetooth ont lieu dans la couche HCI ; la traduction des commandes BT2 en BLE, et inversement, est réalisée par la couche GATT.

Le fabricant de notre module BLE met à disposition un petit logiciel nommé *smartZ* ; il s'agit d'une sorte de micro-système d'exploitation qui, une fois chargé dans le module, offre les commandes dont nous avons besoin, en particulier les commandes *hcitool* (accès à la couche HCI) et *gatttool* (accès à la couche GATT). Pour exécuter ces commandes, il suffit d'envoyer des trames sur le port série, comme si on exécutait le programme sur un ordinateur depuis un terminal.

Le programme *smartZ* est écrit en *smartBASIC*. Puisque son code source est disponible, nous en avons profité pour ajouter quelques lignes pour commander la LED (GPIOFUNC) qui indique l'état de la connexion de notre passerelle et arrêter ce programme lors de la déconnexion (exitfunc 0). La version modifiée de *smartZ* est disponible sur la page de cet article [3].

Il n'est pas nécessaire de modifier le programme du thermomètre BLE, il sera donc toujours possible de l'interroger avec un ordiphone compatible BLE.

Pour charger *smartZ* dans le module BL620 de la passerelle, il faut passer par le port série du module, nous utiliserons le module BT2 pour cette opération. Magique, notre montage est aussi un programmeur BL620/BL600 sans fil !

Le programme Android

Pour lire la température, il faut commencer par connecter le module BLE de la passerelle au thermomètre, ce qui nécessite de connaître l'adresse BLE de ce dernier. Un scan avec la commande suivante permet de trouver tous les dispositifs BLE visibles depuis la passerelle :

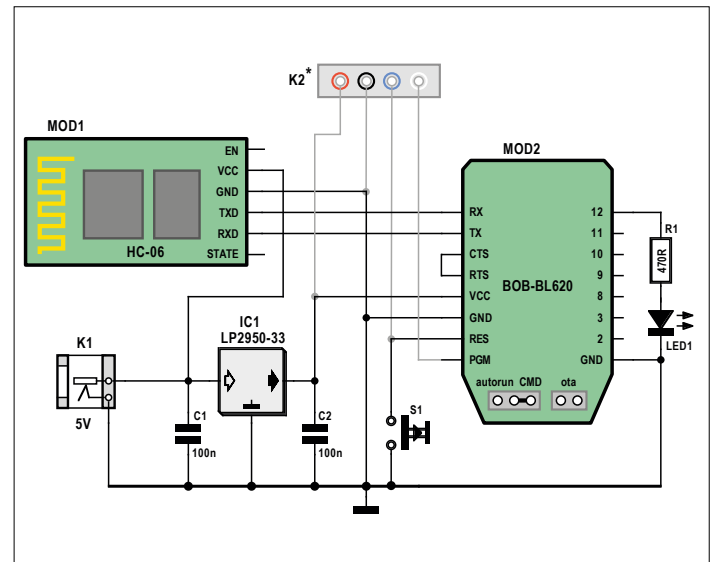


Figure 1 – Le circuit électronique de la passerelle est assez simple pour être réalisé sur une carte de prototypage.

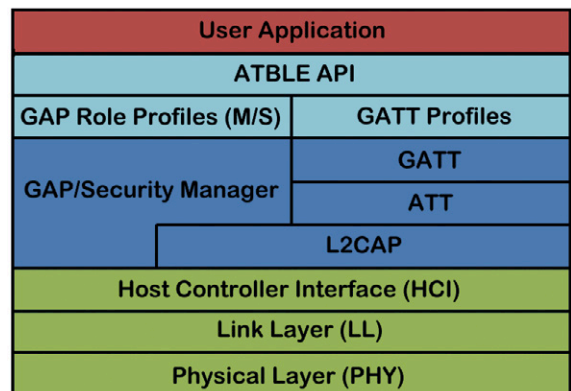
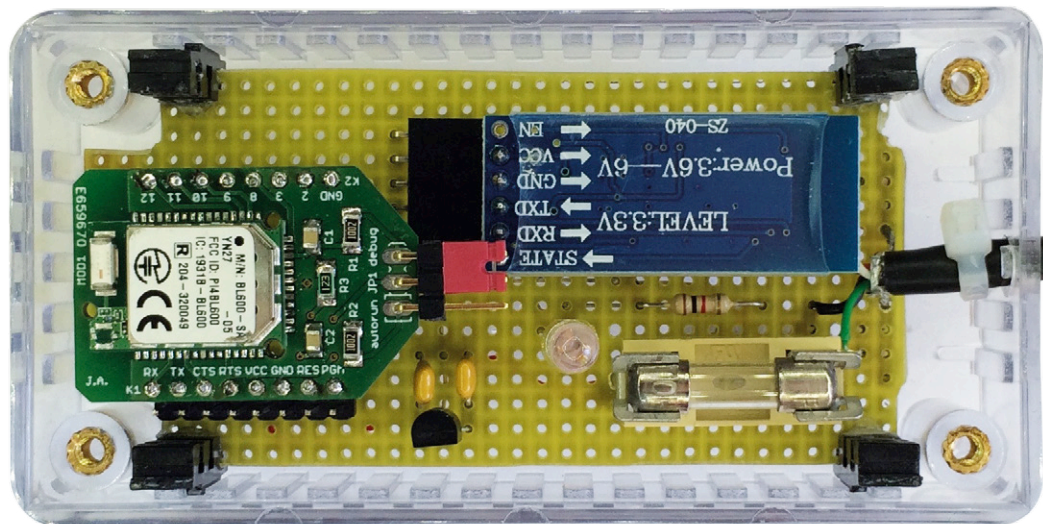


Figure 2 – Les différentes couches de la pile Bluetooth Low Energy (BLE).

```

----->hcitool lescan
LE Scan ...
----->
01E1A3CD4EA220 JA_SWITCH
01E601129AE8F6 LT_UPASS
01EBF5044B6C13 JATEMP

```

Figure 3 – Les résultats d'une recherche de dispositifs BLE avec *hcitool* à partir d'une application terminal.

`hcitool lescan`

Le résultat est une liste contenant les noms et adresses de tous les dispositifs BLE qui ont répondu présents à l'appel (**figure 3**). Dans cette liste, nous cherchons celui qui nous intéresse et nous notons son adresse. L'ordiphone gardera en mémoire cette adresse pour éviter de refaire le scan qui est long. Ensuite nous pouvons nous connecter à cet esclave avec la commande :

`hcitool lecc 01E601129AE8F6`

```

----->hcitool lecc 01E601129AE8F6
Connecting ...
----->
Connection handle 130816
----->

```

Figure 4 – Connexion réussie !

Le numéro à 14 caractères hexadécimal est l'adresse du module auquel nous voulons nous connecter (voir la figure 3). Si la connexion se passe bien, nous obtenons un *handle*, comme le montre la **figure 4**. Dans notre application nous ne nous servons pas de ce *handle* (littéralement une « poignée »). Nous avons franchi la couche HCI. Pour continuer notre chemin, à savoir traverser les autres couches, nous avons besoin de la commande *gatttool* (oui, avec trois « t ») et de sa panoplie de paramètres.

La commande pour lire la valeur de la température est :

`gatttool -char-write -handle=0x001C -value=0100 -listen`

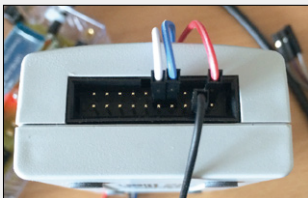
Cette commande accepte de nombreux paramètres et nous vous renvoyons à sa documentation si vous voulez l'explorer plus avant. Ici il suffit de comprendre que *handle* (pas celui de toute à l'heure) correspond à une fonction à exécuter par *gatttool*. Retenez que 0x001C correspond à la lecture de données, 0x001E à l'écriture de données et que 0x000E permet de récupérer le nom du fabricant du module BLE. Le paramètre *value* contient les paramètres de la fonction à exécuter (vous suivez ?). Une valeur de 0100 indique que le maître doit informer l'esclave que le maître est prêt à recevoir des données. Le paramètre *listen* met le maître en mode « attente de données » après avoir envoyé sa demande à l'esclave.

La réponse à cette commande complexe est une trame de caractères qui contient la température. Comme décrit dans

Transformer un BL600 en BL620

Voici comment reprogrammer un e-BoB BL600 (esclave) pour en faire un e-BoB BL620 (maître). Tout d'abord il faut une sonde J-Link de Segger, la version *Edu* est suffisante. Téléchargez en [5] le micrologiciel [BL620 Firmware v12.4.10.0 Revision 1](#) (connectez-vous et créez un compte avec Elektor comme *company*). Téléchargez en [4] le pilote [Software for Windows](#) de la sonde J-Link, puis installez-le. Reliez le connecteur JTAG de la sonde J-Link à un e-BoB BL600, comme ceci :

e-BoB BL600	J-Link (Edu)	couleur de fil sur les photos
PGM	TCK	blanc
RES	TMS	bleu
GND	GND	noir
VCC	VCC	rouge (ce n'est pas une alimentation)



Assurez-vous que le BL600 est doté d'une alimentation externe, la sonde J-LINK ne fournit pas d'alimentation. Lancez la commande [_DownloadFirmwarev12_4_14_0.bat](#) contenue dans le téléchargement de l'étape 1. Félicitations ! Votre module BL600 est devenu un module BL620 (notez-le sur le module).

l'article du thermomètre, la trame reçue est sous la forme PWxxxxVyyyCzzz où xxxx est la tension de la batterie du thermomètre, yyy la tension qui alimente le pont diviseur dont fait partie la thermistance CTN et zzz la tension aux bornes de la thermistance CTN. Ces deux dernières valeurs permettent de calculer la température selon la procédure donnée dans l'article sur le thermomètre. Vous n'avez pas à la calculer vous-même, c'est l'application sur le téléphone qui le fera à votre place. Une fois la température récupérée, on peut couper la liaison entre la passerelle et le thermomètre. Il n'y a rien à faire pour cela, car le thermomètre le fait automatiquement afin de réduire sa consommation.

Puisque toutes les commandes et les réponses sont sous forme de trames ASCII sur le port série du module BLE de la passerelle, tout ce que son module BT2 a à faire est de les relayer : les réponses du module BLE vers le téléphone BT2 et les commandes du téléphone BT2 vers le module BLE. Même si le téléphone est un modèle vieillot, il est tout à fait capable de coder les commandes et de décoder les réponses pour afficher la température, ce qui évite de rajouter un microcontrôleur à la passerelle.

Aller plus loin

Voilà, maintenant vous savez comment connecter un téléphone BT2 à un dispositif BLE à travers une petite passerelle. C'est plus compliqué qu'avec un téléphone BLE, mais ce n'est pas impossible. Grâce au surpuissant module BL620, nul besoin de rajouter des microcontrôleurs ou d'autres composants complexes, un peu de programmation en smartBASIC et le BLE obéit à tous vos ordres.

Si vous souhaitez aller plus loin dans la compréhension de la pile Bluetooth Low Energy et ses couches, nous vous conseillons vivement l'internet. En ce qui concerne le module BL620, c'est le site de son fabricant qui peut vous renseigner.

Si vous remplacez le module BT2 par un microcontrôleur et un afficheur, vous pouvez réaliser un afficheur déporté. Et pendant que vous y êtes, avec un capteur de température supplémentaire et un peu de programmation, vous pouvez réaliser un réseau sans fil de thermomètres capable de mesurer la température à cinq endroits différents. Intéressant non ?
Merci à Mustafa pour l'idée du projet. ◀

(150753)

Liens

- [1] *Thermomètre sans fil Bluetooth Low Energy, affichage de la température à distance sur votre téléphone tactile*, Elektor 01/2015 : www.elektormagazine.fr/140190
- [2] Compilation des huit articles sur le BL600, parus dans Elektor : www.elektor.fr/elektor-select-compilation-bl600
- [3] Version modifiée de smartZ : www.elektormagazine.fr/150753
- [4] Segger : www.segger.com/jlink-software.html
- [5] Laird Technologies : www.lairdtech.com/products/bl620, voir *software downloads*

Charger smartZ sur l'e-BoB BL620

Sur le site de Laird Technologies [5], téléchargez le programme [UWTerminal](#) et le micrologiciel [BL620 Firmware v12.4.10.0 Revision 1](#) (connectez-vous et créez un compte avec Elektor comme *company*).

Jumelez votre ordinateur avec le module HC-06. La procédure exacte dépend de la version de Windows.

Connectez-vous au module HC-06 avec le code PIN 1234.

Identifiez le numéro de port de communication dans le gestionnaire de périphériques de Windows.

Lancez [UWTerminal.exe](#) en choisissant 9600 bauds et le bon port de communication.

Faites un clic droit puis, dans le menu, choisissez *Download* → *BASIC* → *Load Precompiled BASIC*

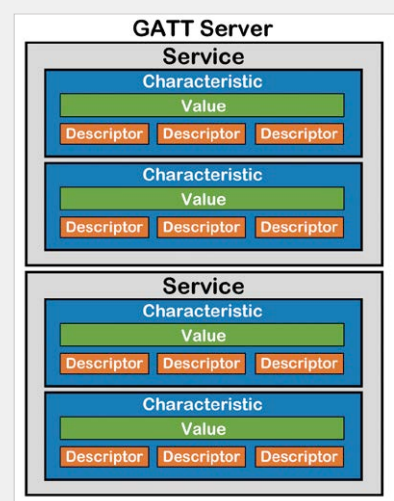
Choisissez le fichier [smartZ.bridge.uwc](#) et le chargement du programme commence.

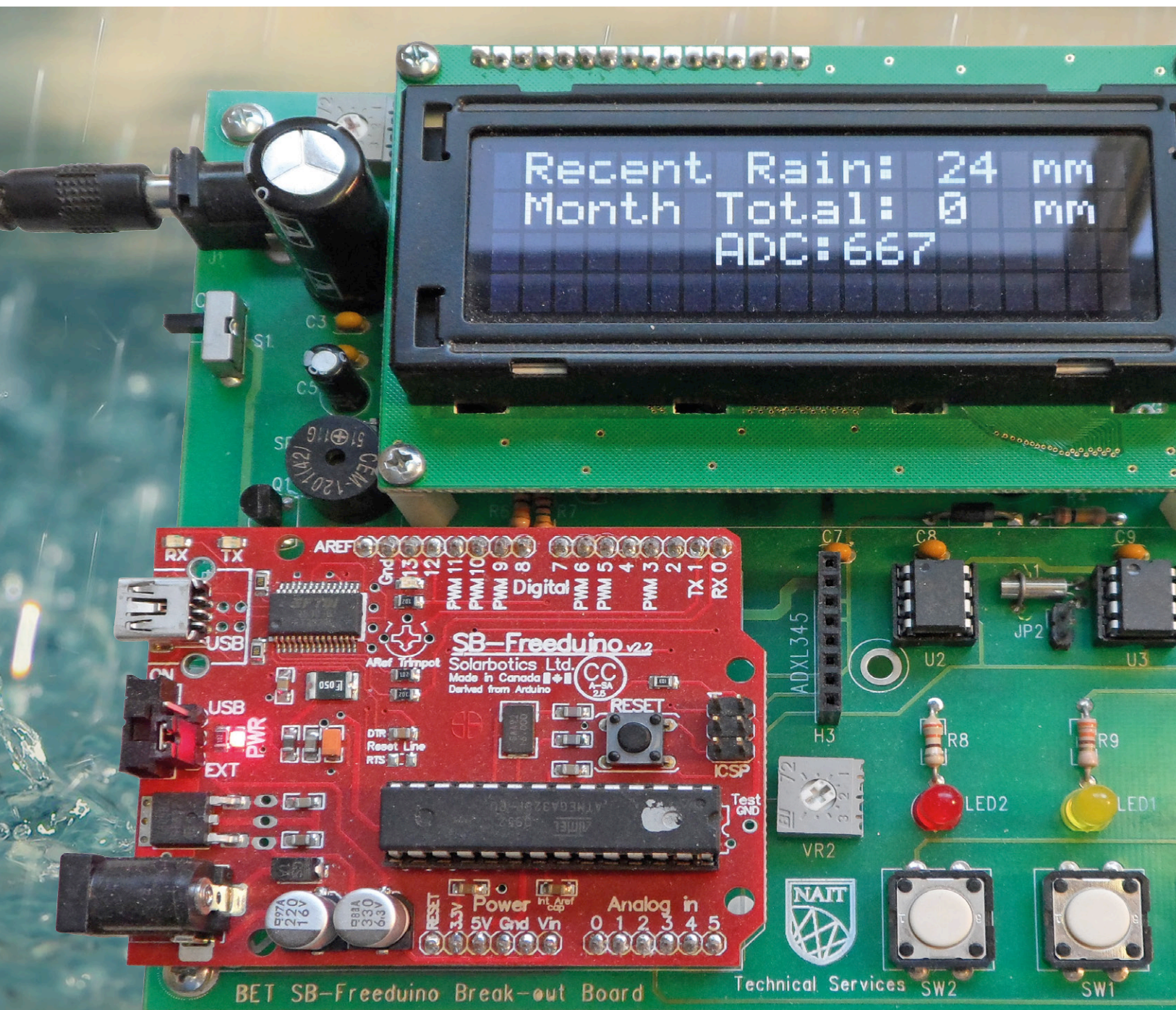
Le profil GATT

Le Bluetooth Low Energy, aussi appelé Bluetooth Smart ou BLE, est destiné aux objets connectés dont les débits de données sont faibles et dont l'autonomie est cruciale. Les dispositifs BLE offrent des fonctions standardisées (ou profils) qui couvrent la domotique, la santé, le sport, l'internet, la téléphonie, etc.

Pour les applications où les profils prédéfinis ne conviennent pas, il y a le profil générique appelé GATT (de *Generic ATtribute*). Le GATT permet de créer ses propres profils en assurant l'interopérabilité entre différents dispositifs BLE. Il comporte trois niveaux : *Profile*, *Service* et *Characteristics*. Un *Profile* est un ensemble de *Services* qu'un dispositif BLE met à disposition. Un *Service* est un ensemble de *Characteristics* qui permet d'effectuer certaines actions. Chaque *Service* a un identifiant unique, ce qui permet ainsi le dialogue entre les dispositifs BLE de différents fabricants. Par exemple, le service *internet protocol support* possède l'identifiant 0x1820.

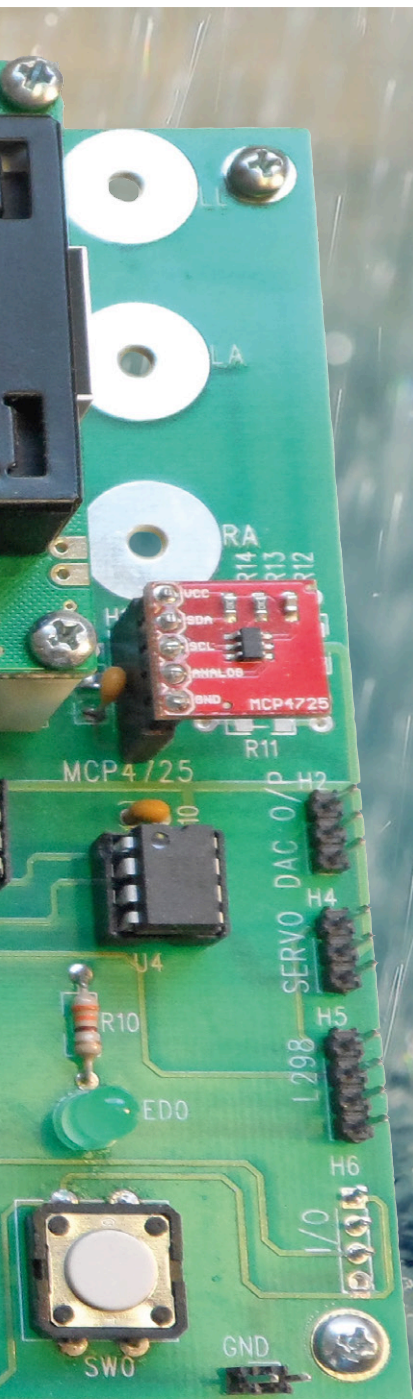
Les *Characteristics* sont les attributs d'un *Service*. Par exemple, le service *internet protocol support* peut posséder un attribut *URI*. En plus de sa valeur, une *Characteristic* peut posséder un certain nombre de *Descriptors* qui donne plus d'information sur la valeur. Exemple : l'attribut *URI* peut avoir un *Descriptor* qui indique s'il s'agit d'une adresse IP sous forme d'un nombre ou d'une trame de caractères. Ceci est une description de GATT plus que succincte. Pour de plus amples informations, reportez-vous à www.bluetooth.com/specifications/GATT





Gordon D. Dick (Canada)

Pour construire un pluviomètre, on utilise typiquement un condensateur variable ou un auget à bascule. Pour moi, ça n'a pas du tout marché ; il me fallait autre chose. Il y a environ un an, Freescale présente le capteur de pression MPXV5004D. La documentation indique qu'il peut servir de jauge de niveau dans les machines à laver domestiques. Pourquoi pas dans un pluviomètre ? Voici comment faire.



pluviomètre

avec MPXV5004D
et Arduino Nait

Ceux qui ont étudié un peu la physique savent que la pression P à la base d'un récipient est donnée par :

$$P = \rho g h \quad [\text{Pa}]$$

avec

ρ = densité en kg/m^3

g = accélération due à la gravité = $9,8 \text{ m/s}^2$

h = hauteur du fluide dans le récipient en m

La feuille de caractéristiques du capteur MPXV5004D de Freescale [1] spécifie qu'une pression de 0 à 3,92 kPa (soit 0 à 400 mm de hauteur d'eau) produit 1,0 à 4,9 V. La **figure 1** montre la fonction de transfert correspondante. Afin de pouvoir utiliser ce transducteur dans un pluviomètre, il faut une formule mathématique pour la fonction de transfert. Comme le transducteur est linéaire, il peut être modélisé par une droite avec un décalage à l'origine (fonction affine pour les matheux) :

$$V_o = G L_i + V_{\text{offset}}$$

avec

G = gain du transducteur en V/mm

L_i = niveau de liquide en mm

$V_{\text{offset}} = 1 \text{ V}$

Le gain du transducteur correspond au coefficient directeur de la droite (sa « pente ») de la fonction de transfert, donné par :

$$\text{pente} = \frac{\Delta V_o}{\Delta L_i} = \frac{4,9 - 1,0}{400 - 0} = 9,75 \frac{\text{mV}}{\text{mm}}$$

La fonction de transfert est donc :

$$V_o = (9,75 \times 10^{-3}) \times L_i + 1$$

Elle permet de prévoir la tension en sortie du transducteur pour un niveau de liquide donné.

Conception mécanique

Ce pluviomètre est conçu pour que la plupart d'entre vous puissent le construire à l'aide de pièces faciles à trouver : tuyaux d'évacuation d'eau et raccords correspondants. J'utilise des tuyaux et raccords aux dimensions américaines ici, il devrait être aisé de dénicher un équivalent européen. Le prototype du pluviomètre est visible en **figure 2**. Un entonnoir se trouve au

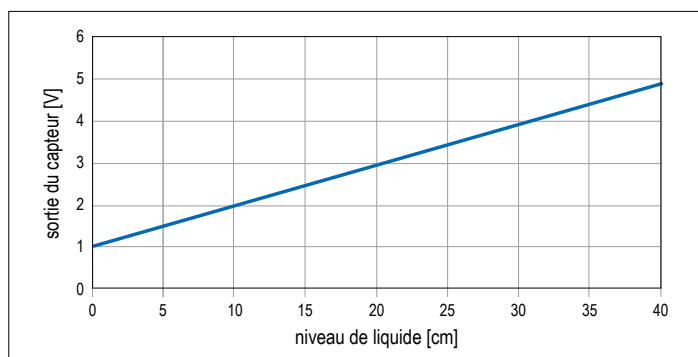


Figure 1. Fonction de transfert du MPXV5004x

sommet, un simple adaptateur 3,5" (8,9 cm) vers 1,5" (3,8 cm). Le tronçon de tube au-dessous fait un peu plus de 400 mm de long. À son extrémité il y a un coude qui permet d'accéder au fond du récipient de mesure de la pression et accueille le circuit imprimé. Enfin, le court tronçon de tube qui se trouve au-dessous est prévu pour être inséré dans un tube vertical qui sert de pied au pluviomètre. Une paire de vis en laiton montées en travers (peu visibles sur la photo) font office de butée et limitent l'enfoncement dans le pied.

Il y a plusieurs façons de réaliser le bouchon qui constitue le fond du récipient. Je me suis servi de tôle d'aluminium de 3/8" découpée avec une défonceuse à commande numérique. Vous pouvez utiliser d'autres matériaux et méthodes comme par ex. une scie à chantourner et une ponceuse ou bien une lime.

J'ai découvert à mes dépens que certaines méthodes d'installation du bouchon sont à éviter. La première fois, j'ai essayé de le coller avec du ciment pour ABS. Ça fuyait. J'ai ensuite employé un ciment qui se targuait d'être bon pour assembler différents matériaux. Je ne vais pas nommer le produit, qui est sans doute très bon par ailleurs, mais ce qui est sûr c'est qu'il ne convient pas pour le tube en ABS et l'aluminium. Après deux essais infructueux, j'ai enfin réussi avec le bon vieux silicone. Le bouchon n'a pas besoin d'avoir exactement la bonne dimension : le silicone remplira les trous.

Un court tube en laiton de 3/32", facile à trouver dans les magasins de bricolage, sert de prise de pression. Je l'ai collé à l'époxy dans un trou réalisé au fond du bouchon. Attention : réalisez le trou afin qu'il soit facile, en passant par le coude, d'y glisser le bout de tube de 3/32" provenant du transducteur. Si vous avez de gros doigts, cela peut être un défi ! Le petit circuit imprimé avec le transducteur et le bornier est inséré dans le tube par le bas après avoir fixé les fils sur une entretoise installée auparavant. Le bouchon visible en figure 2 sera installé après l'insertion de la prise de pression, pour protéger des intempéries ; il ne sera pas collé.

Test du transducteur

Il est prudent de vérifier que le transducteur fonctionne comme prévu, ce qui était le cas sur mon prototype. Commençons par un test simple : souffler dans un tube plastique relié au transducteur. Il faut vraiment souffler fort

Figure 2. Les composants du prototype de pluviomètre.



pour atteindre la limite de pression du transducteur. Dès que j'ai pu, j'ai retesté le transducteur avec un récipient, tout fonctionnait. La tension de décalage est donnée dans la feuille de caractéristiques avec une tolérance de $\pm 0,25$ V. Aucune tolérance n'est indiquée pour l'étendue.

Les schémas électriques et d'implantation du circuit imprimé du prototype sont visibles respectivement sur les **figures 3** et **4**. Vous remarquerez sans doute qu'il y a plus de composants sur la carte que sur le schéma. C'est parce que le prototype est alimenté depuis une source de 12 V et pas 5 V. Le boîtier TO-92 est juste un 78L05 pour réduire la tension pour le MPXV5004D. Il y a aussi un électrolytique de découplage sur le 12 V.

Interface avec Arduino

Habituellement, il faut conditionner le signal d'un transducteur avant de l'appliquer à l'entrée du CA/N d'un microcontrôleur. Freescale le fait déjà dans le capteur et l'on peut donc utiliser sa sortie directement, pour peu que l'on accepte de perdre la zone de 0 à 1 V de la gamme du CA/N (voir plus bas).

Supposons que nous utilisons la sortie du transducteur telle quelle (sans conditionnement du signal), nous devons déterminer la fonction de transfert inverse pour que le microcontrôleur puisse calculer le niveau de liquide à partir de la tension de sortie du transducteur. La fonction de transfert inverse est obtenue en manipulant l'équation de la fonction de transfert afin que la quantité en entrée devienne le membre de gauche de l'équation :

$$L_i = \frac{V_o - 1}{G}$$

Pour pouvoir calculer le niveau de liquide, il faut mesurer la tension de sortie du transducteur avec le CA/N. Je vais supposer que le CA/N a une résolution de 10 bits. Comme V_o , la tension en sortie du transducteur est maintenant une entrée pour le CA/N, et L_i , le niveau de liquide en entrée du transducteur, est le résultat du calcul effectué par le microcontrôleur, il sera plus clair de changer la notation comme suit :

$$L_o = \frac{V_i - 1}{G}$$

L'équation suivante (pour une sortie en binaire) décrit le comportement du CA/N :

$$N = \frac{V_i \times (2^n - 1)}{V_{ref}}$$

avec

V_i = tension en entrée du CA/N

n = nombre de bits du CA/N

V_{ref} = tension de référence du CA/N

N = valeur numérique en sortie du CA/N

Si l'on transforme l'équation pour en tirer le terme V_i et qu'on le substitue dans l'équation pour L_o , on obtient après quelques transformations :

$$L_o = \frac{N \times V_{ref} - (2^n - 1)}{G \times (2^n - 1)}$$

En prenant $n = 10$, $2^{10} = 1024$ et si $V_{ref} = 5,0$ V, l'équation

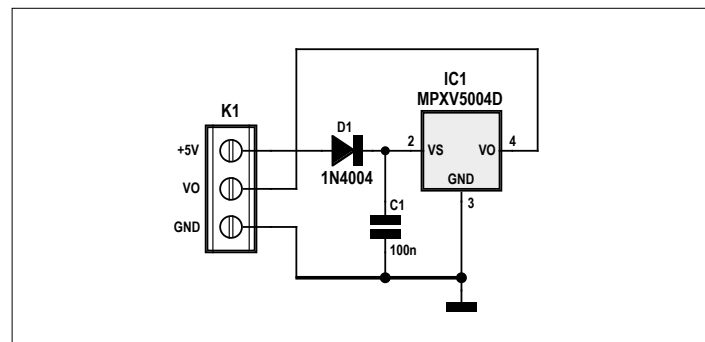


Figure 3. Schéma de la carte du transducteur avec une diode de protection et un condensateur de découplage.

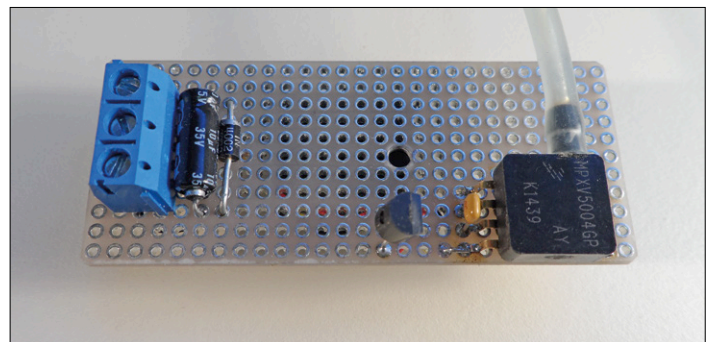


Figure 4. Le circuit du transducteur assemblé sur une plaque d'essais.

devient :

$$L_o = \frac{N \times 5,0 - 1023}{9,974}$$

Nous utiliserons cette équation pour calculer la hauteur de pluie à partir de la tension issue du capteur de pression et mesurée par le CA/N. La programmation en C est ici très utile : une ligne de code suffit. Ce serait nettement plus compliqué en assembleur.

Ce calcul suppose que la pluie arrive directement dans le récipient. Cependant, il est courant, en pratique, d'utiliser un entonnoir pour élargir l'entrée du récipient et collecter plus d'eau. Cela améliore la résolution de la mesure.

Avant d'avoir des données valides pour l'Arduino, il va falloir pas mal de traitement du signal (voir encadré).

Développement de l'application Arduino

Si tout ce que vous attendez de votre pluviomètre c'est qu'il affiche la quantité de pluie à chaque fois qu'il pleut, alors le code sera plutôt simple. En supposant que vous avez choisi de ne pas conditionner le signal en sortie du capteur, il suffit d'évaluer la fonction de transfert inverse et d'afficher le résultat en boucle. Cependant, si vous avez utilisé un entonnoir, il faut modifier la fonction de transfert inverse pour tenir compte de la plus grande surface de collecte. Pour le pluviomètre de la figure 2, le diamètre du récipient est 1,602" (4,069 cm), celui de l'entonnoir 3,508" (8,910 cm). Le rapport des surfaces est :

$$\frac{(3,508)^2}{(1,602)^2} = 4,796$$

L'ajout de l'entonnoir permet au récipient de collecter 4,796 fois plus d'eau. Le remplissage du récipient de 40 cm de la figure 2 correspond donc à 8,35 cm de pluie ; cela convient très bien pour la météo de la province de l'Alberta. En insérant le coefficient de modification dans l'équation, on obtient la fonction de transfert suivante :

$$\text{Pluie (mm)} = L_o = \frac{N \times 5,0 - 1023}{4,796 \times 9,974} = \frac{N \times 5,0 - 1023}{47,776}$$

Notre pluviomètre est équipé d'un microcontrôleur, il est facile d'y ajouter des fonctions intéressantes. Enregistrer la pluviométrie totale serait utile. Il suffirait de presser un bouton avant

de vider le pluviomètre pour ajouter chaque chute de pluie au total. Il faudrait donc un autre bouton pour remettre à zéro le total. On pourrait aussi stocker les données en EEPROM. Il y a vraiment de quoi laisser cours à sa créativité.

La carte de type Arduino utilisée dans ce projet est celle de l'illustration en tête d'article. Cette carte est employée par le **Northern Alberta Institute of Technology** (NAIT) au sein du *Biomedical Electronics program*. Plusieurs des périphériques qu'elle comporte ne seront pas utilisés ici. Il est cependant agréable d'avoir un afficheur à cristaux liquides et des boutons sous la main. Cette carte est une variante Arduino fabriquée par Solarbotics, AB, Canada, appelée Freeduino.

Conditionnement du signal

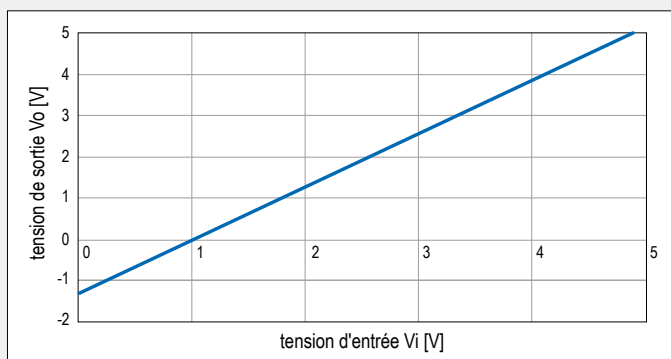


Figure 5. Fonction de transfert du circuit de conditionnement du signal.

Je vous propose d'étudier comment conditionner le signal en sortie du transducteur (1,0 à 4,9 V) pour obtenir un signal de 0 à 5 V. Le graphique de la **figure 5** montre le fonctionnement désiré. La sortie du transducteur attaque l'entrée du conditionneur, c'est l'axe des abscisses sur le graphique, et la sortie du conditionneur attaque l'entrée du CA/N, l'axe des ordonnées. Mathématiquement, la fonction de transfert du conditionneur est décrite par l'équation suivante :

$$V_o = G \times V_i + V_{offset}$$

avec G = gain en tension du conditionneur.

Le gain du conditionneur est la pente de la droite de la fonction de transfert, soit

$$\text{pente} = \frac{\Delta V_o}{\Delta V_i} = \frac{5,0 - 0}{4,9 - 1} = \frac{5}{3,9}$$

On obtient :

$$V_o = \frac{5}{3,9} \times V_i + V_{offset}$$

En remplaçant $V_i = 1,0$, $V_o = 0$, on peut trouver la tension de décalage :

$$0 = \frac{5}{3,9} \times 1,0 + V_{offset}$$

La fonction de transfert est donc :

$$V_o = \frac{5}{3,9} \times V_i - \frac{5}{3,9}$$

Beaucoup de circuits de conditionnement pourraient être utilisés ici. Celui que j'ai choisi (**fig. 6**) est simple, mais n'en est pas moins intéressant. Ce conditionneur peut être utilisé avec une alimentation unipolaire. La sortie du capteur serait reliée à l'entrée V_1 . Un signal continu serait appliqué à l'entrée V_2 pour produire le décalage de la fonction de transfert. Pour comprendre le montage, supposons que $V_2 = 1$ V. En utilisant, le théorème de superposition, avec $V_1 = 0$ V, R_2 ne fait plus partie du circuit, à cause de la masse virtuelle. Avec $R_1 = 39$ k Ω et $R_3 = 50$ k Ω , on obtient le circuit de la **figure 7**. La tension de décalage introduite est :

$$-(50/39) \times 1 = -5/3,9 \text{ [V]}$$

Redessignons maintenant le circuit avec $V_2 = 0$ V (**fig. 8**).

Si l'on calcule la valeur de R_2 permettant d'avoir le gain nécessaire pour les signaux du transducteur, on obtient une valeur négative. Pas facile de trouver une telle résistance dans nos tiroirs.

Effectuons maintenant une analyse plus générale du circuit de la **figure 6** pour expliquer le pourquoi de cette résistance négative. En utilisant le théorème de superposition, et une bonne dose d'algèbre, on obtient :

$$V_o = \frac{(R_1 R_2 + R_1 R_3 + R_2 R_3) V_1}{R_2 R_3} - \frac{R_1}{R_3} V_2$$

Si l'on procède comme précédemment pour établir R_1 et R_3 pour produire la tension de décalage, on obtient ($V_2 = 0$ V) :

$$V_o = \frac{(R_1 R_2 + R_1 R_3 + R_2 R_3) V_1}{R_2 R_3}$$

ou

$$\frac{V_o}{V_1} = A_v = \frac{(R_1 R_2 + R_1 R_3 + R_2 R_3)}{R_2 R_3}$$

Même si cette carte possède un afficheur à quatre lignes, je n'en ai utilisé que deux : les afficheurs à deux lignes sont bien plus courants. La troisième ligne que l'on peut voir sur la photo n'est pas utilisée dans la version finale. Pour ajouter la valeur courante au total, il suffit d'appuyer sur SW2, le bouton sous la LED rouge. Pour remettre à zéro le total mensuel, il faut utiliser SW0, le bouton sous la LED verte. Il n'y a pas de bouton pour remettre à zéro la valeur courante : cela se fait en vidant le pluviomètre.

Figure 10. Une des deux bagues de fixation pour le tube qui supporte le pluviomètre.

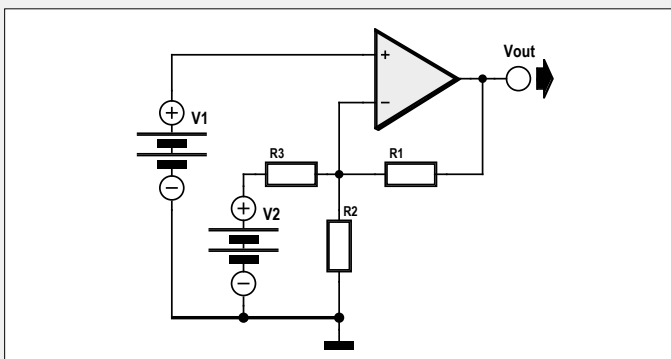
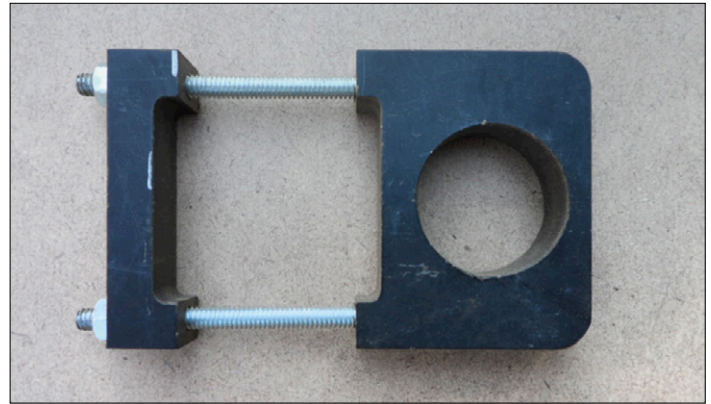


Figure 6. Le conditionneur sans les valeurs des composants.

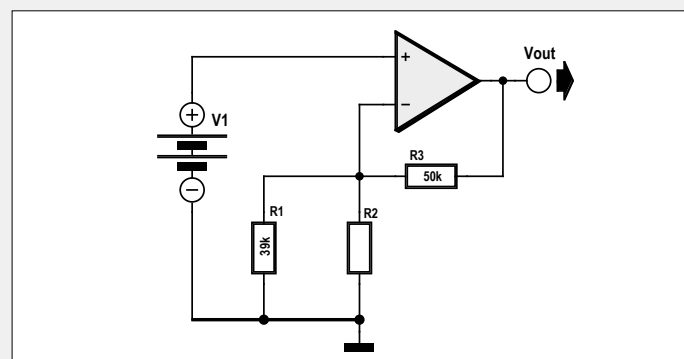


Figure 8. Le conditionneur avec V2 = 0 V.

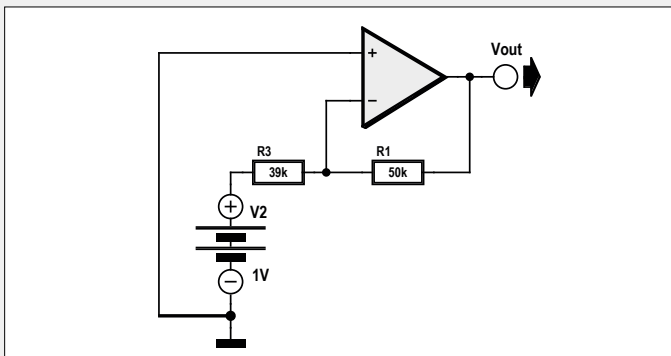


Figure 7. Le conditionneur avec V1 = 0 V.

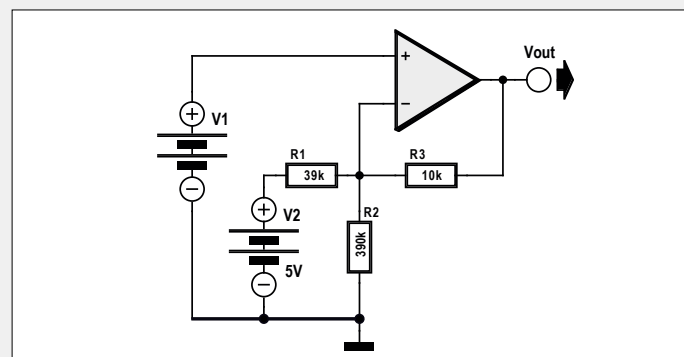


Figure 9. Version finale du conditionneur avec les valeurs des composants.

Étant donné que R_1 , R_3 et A_v sont connus, on peut extraire R_2 de l'équation. Après un peu plus d'algèbre :

$$R_2 = \frac{R_1 R_3}{R_3 A_v - (R_1 + R_3)}$$

Cette équation montre que R_2 peut être négative si l'expression du dénominateur est négative. Pour éviter cela, il faut faire en sorte que :

$$R_3 A_v > (R_1 + R_3)$$

Essayons à nouveau d'attribuer des valeurs, mais en gardant cette condition en tête. Si, comme précédemment, on commence par choisir R_2 et R_3 , mais cette fois-ci avec $V_2 = 5.0$ V (utilisez une puce référence de tension), R_1 doit valoir 10 k Ω pour produire la tension de décalage de $-5/39$ V. Calculez maintenant R_2 . Vous remarquerez que l'on obtient une valeur positive ; c'est parce que :

$$39k \left(\frac{5}{3.9} \right) > (10k + 39k)$$

Le schéma final du conditionneur de signal est en **figure 9**.



Figure 11. Le pluviomètre terminé. Il n'attend plus que la pluie.

Emplacement et montage

Ce pluviomètre est conçu pour être inséré dans un tube à la verticale qui servira de support. J'ai choisi de le monter sur un tube en aluminium fixé au coin de ma terrasse en bois. L'accessoire que j'ai utilisé pour fixer le tube est visible en **figure 10**. Une fois encore, la défonceuse à commande numérique a été utile pour fabriquer cette fixation à partir d'un *mystérieux* matériau de récupération, très résistant aux intempéries.

Bien sûr, le pluviomètre doit être monté à un endroit où rien ne fait obstruction à la pluie. Un autre problème à résoudre est la gestion de l'alimentation et du signal de sortie. Vous voulez enterrer des câbles pour atteindre l'endroit idéal ? Mon pluviomètre fini et installé est visible en **figure 11**.

Problèmes inattendus

Initialement, je m'attendais à ce que l'eau, une fois arrivée au petit tube qui mène au capteur, s'arrête dès que la pression hydrostatique serait la même dans le tube et au-dessus

de l'eau. J'imaginai que l'air dans le tube se comporterait comme une sorte de bouchon invisible. Cela ne s'est avéré que partiellement. Ce que je n'avais pas prévu, c'est que le bouchon d'eau aller se déplacer vers le bas du tube, jusqu'à ce qu'il soit au niveau du transducteur. Une fois dans le tube, l'eau est difficile à extraire à cause de la tension superficielle sur les bords du petit tube. Certaines sessions de test se sont déroulées comme suit :

1. Le récipient est sec ; signal en sortie du capteur = 1,0 V
2. Je verse de l'eau dans le récipient, le signal augmente avec la hauteur d'eau
3. Vidage du récipient ; signal en sortie du capteur = 1,23 V
4. Je verse de l'eau dans le récipient, le signal augmente avec la hauteur d'eau
5. Vidage du récipient ; signal en sortie du capteur = 1,23 V
6. Et ainsi de suite

Impossible d'obtenir à nouveau le signal que l'on mesure lorsque le pluviomètre est sec sans démonter les tubes et chasser l'eau du petit tube en laiton. Si l'on attend plusieurs minutes entre le remplissage et le vidage du récipient, les gouttes d'eau des parois du récipient s'accumulent au fond et élèvent légèrement la tension en sortie du capteur.

L'entrée du capteur se trouve env. 23,8 mm au-dessous du fond du récipient. Si l'on calcule la tension en sortie du capteur pour une colonne de 23,8 mm, on obtient 1,23 V. Tout est clair maintenant. Cela explique la tension observée après avoir vidé le récipient : le tube au-dessous contenait toujours de l'eau. Malheureusement, cela signifie qu'il y a un décalage de 23,8 mm dont il faut tenir compte dans le code. Cela signifie également que la quantité de pluie mesurable ne sera que d'environ 60 mm avant que la pression maximale tolérée par le capteur (400 mm d'eau) ne soit atteinte. J'imagine qu'il y a tout un tas de petits changements que l'on pourrait faire pour améliorer ce pluviomètre ; les suggestions sont les bienvenues. Arrivé là, je me disais qu'il serait sans doute prudent de chercher des notes d'applications de Freescale ou des exemples sur le net. Je n'ai pas trouvé grand-chose à part un fil de discussion très utile [2].

J'y ai appris que le capteur est prévu pour de l'air sec et que l'humidité réduit sa durée de vie. Bon à savoir ! La solution proposée était d'utiliser du silicone à haute viscosité dans le tube entre le capteur et le récipient. Tous les silicones que j'avais étaient trop visqueux et se comportaient comme un bouchon, empêchant les changements de pression d'arriver au capteur, sans doute à cause du faible diamètre du tube que j'ai utilisé. Ma solution a été d'ajouter une petite quantité d'huile moteur dans le tube. J'ai amené l'huile en bas d'une boucle dans le tube ; elle ne peut s'échapper et n'atteindra jamais le capteur. Voici ce qui conclut la conception et la construction du prototype de pluviomètre.

(150471 – version française : Kévin Petit)

Liens

[1] Feuille de caractéristiques du MPXV5004 : www.datasheetlib.com/datasheet/467524/mpxv5004_freescale-semiconductor.html

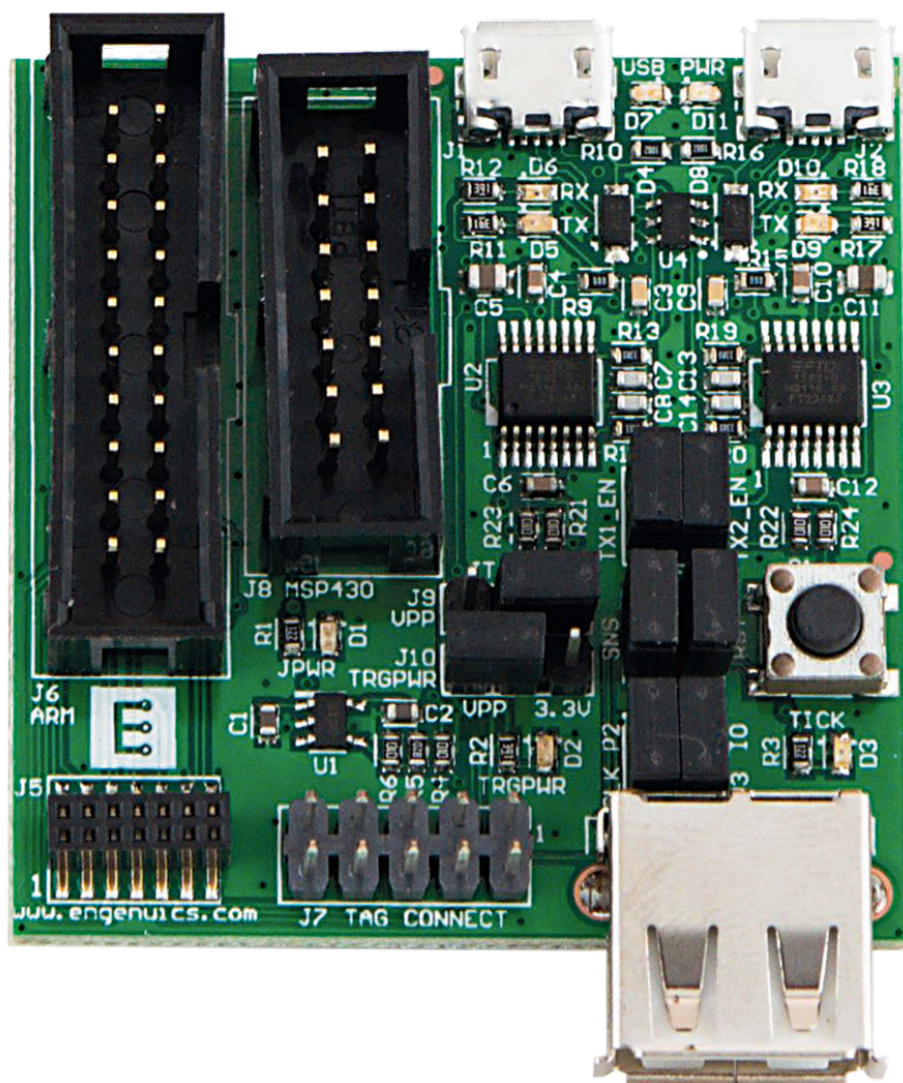
[2] Notes d'application : <https://community.freescale.com/thread/357642>

programmeur JTAG universel / adaptateur de débogage

avec connexions pour divers microcontrôleurs

Jason Long,
Engenuics Technologies (Canada)

Ce petit appareil pratique est un programmeur et adaptateur de débogage multimodes aux nombreuses fonctions, pour systèmes embarqués avec ARM ou MSP430. Utilisé avec un programmeur standard J-Link ou MSP-FET, il offre trois options de connexion avec le système cible, sans qu'il y ait besoin d'une interface spécifique sur celui-ci. Les deux convertisseurs USB-série peuvent être utilisés avec le programmeur ou seuls.



De nos jours, il n'est plus nécessaire de se torturer les méninges pour relier son ordinateur au microcontrôleur d'un système embarqué : la plupart de ces systèmes possèdent une interface de programmation et de débogage. Par contre, si vous voulez développer un nouveau circuit, vous aurez à coup sûr besoin d'un programmeur externe. Les familles de processeurs ARM et MSP430 utilisent des appareils à la norme JTAG pour la programmation et le débogage (la **figure 1** montre deux exemples).

Ces appareils sont reliés à l'aide de gros connecteurs, dont le brochage est illustré en **figure 2**. Tous deux fournissent une

tension (en général 5 V) qui peut alimenter la cible, ainsi que tous les signaux nécessaires à la programmation. Les processeurs ARM Cortex ont un mode deux fils (SWD, *Serial Wire Debug*), ce qui épargne pas mal de fils.

Les connecteurs sur la cible sont eux aussi volumineux – ce n'est pas vraiment souhaitable sur un produit fini – et on est surpris de ce que l'on doit prendre en compte pour les connexions. Cet article examine les critères essentiels pour le choix d'une connexion, et introduit un bel outil qui vous sera très utile, le programmeur/adaptateur de débogage JTAG (JTAGD).

Dimensions et coût

Les dimensions du connecteur sont aisées à imaginer, elles dépendent de la place disponible. Il existe des connecteurs vraiment minuscules et au pas très petit, mais ils sont chers, même en grosse quantité. Attention à ceux qui sont difficiles à trouver ou proposés par un seul fabricant, vous risquez les maux de tête lors du réapprovisionnement. Le prix est **toujours** un critère lors d'une production de masse ; pendant vos recherches, réfléchissez donc à une éventuelle utilisation du connecteur dans d'autres applications. Une autre possibilité est de mettre des plots de connexion sur le circuit imprimé



Figure 1. Le J-Link de Segger pour les microcontrôleurs ARM [1] et le MSP-FET pour la famille MSP430 [2]. Source : Segger et Texas Instruments

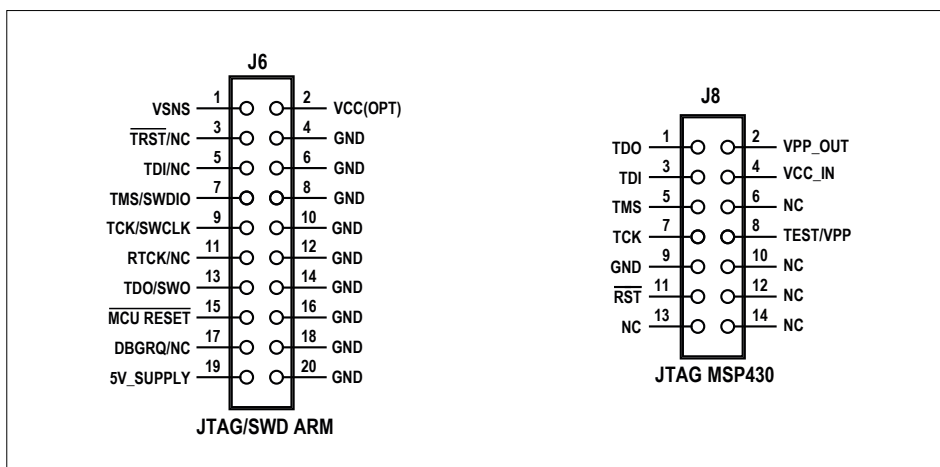


Figure 2. Brochage des connecteurs de programmation pour processeurs ARM et MSP.

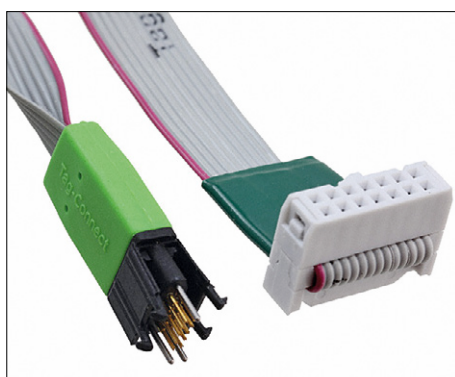


Figure 3. Un Tag Connect TC2050-IDC utilise un certain nombre de plots sur le circuit imprimé pour établir les connexions pour la programmation. Source : Tag-Connect LLC

et d'y « coller » votre connecteur maison. Mais prudence : il est pratiquement impossible de maintenir un connecteur rigide en place pendant la programmation sans rompre le contact. Idéalement il faut des contacts à ressort (*spring-loaded pogo pins*) ; s'il ne faut que quelques secondes pour « flasher » le processeur, il est possible de maintenir les contacts à ressort en place. Pour le débogage, oubliez cette solution ! Vous finiriez par essayer de fixer le connecteur au circuit avec un élastique ou du ruban adhésif. En résumé, si vous utilisez des plots de connexion, utilisez des contacts à ressort sur le connecteur, quel

que soit son type : indépendant, à fente ou à lit de clous.

Alimentation

Alimenter la cible à partir du programmeur est bien pratique, mais certains circuits y sont « allergiques » et doivent avoir leur propre alimentation. Beaucoup de circuits liés à l'internet des objets – comme des capteurs ou des balises – utilisent une pile bouton de 3 V ou sont alimentés via un port USB, même s'ils n'utilisent pas l'accès aux données USB. Ils ont donc un port USB disponible, il faut en tenir compte. L'idéal est d'avoir trois options – 3 V, 5 V ou rien – et c'est encore mieux de pouvoir changer facilement de tension sur le connecteur même.

Autres signaux

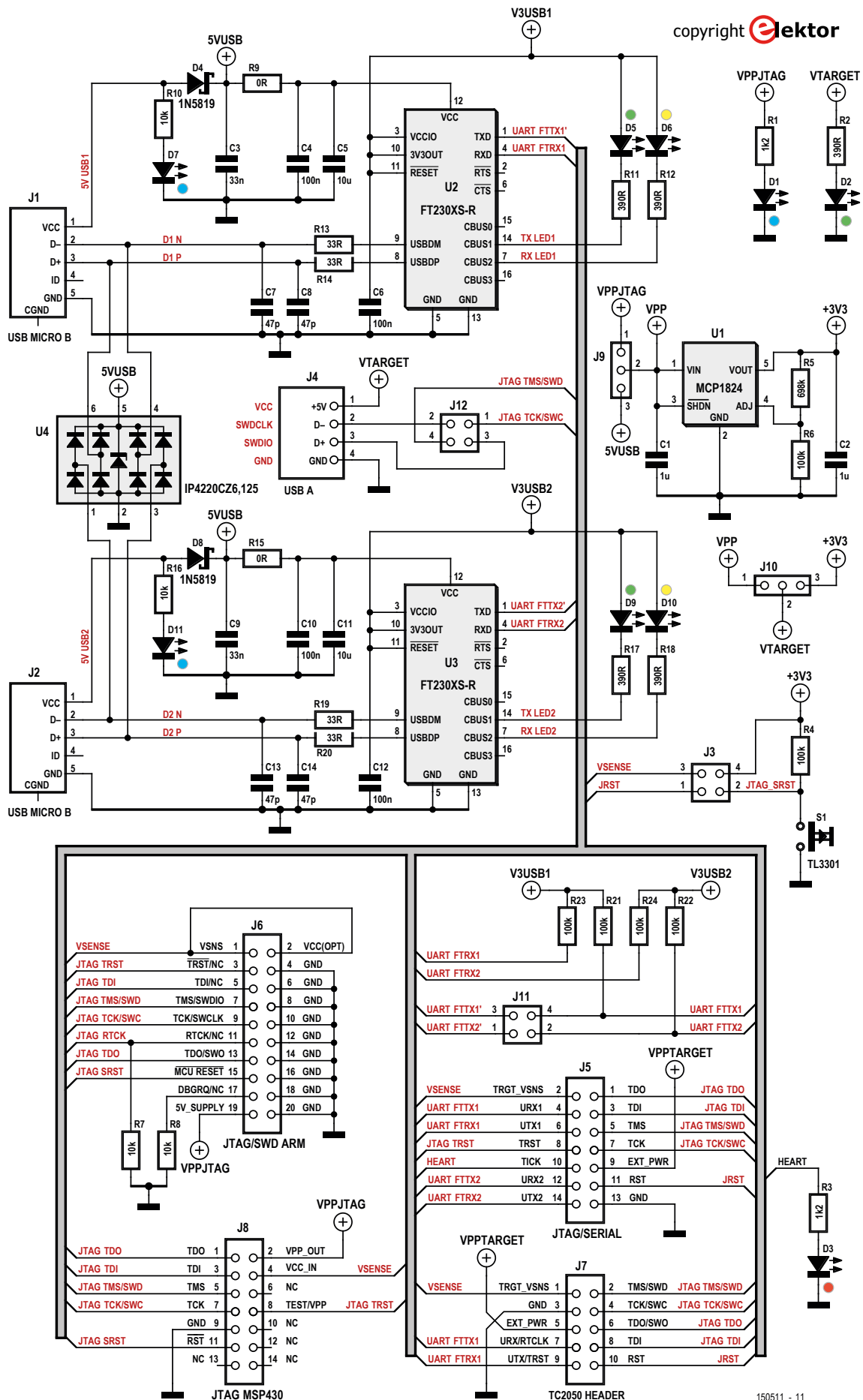
La plupart des systèmes embarqués ont probablement un port général de type UART pour le débogage. Il est essentiel d'y avoir accès, car il est souvent utilisé pour charger la configuration du circuit ou envoyer les résultats d'un auto-diagnostic. L'optimisation d'un processus de production en vue d'un usage facile est très importante : chaque étape supplémentaire coûte du temps et de l'argent, et accroît le risque d'erreur. Il est aussi utile d'avoir accès à la ligne de remise à zéro (*reset*) du processeur, ainsi qu'à quelques autres signaux d'E/S.

Choix

Notre JTAGD possède trois connecteurs ; ce choix est le fruit de l'expérience, acquise au fil des ans par l'auteur, dans le développement de produits :

- **Pas de 1,28 mm, 2x7 broches (J5).** Ce composant relativement petit est à montage en surface, disponible auprès de plusieurs sources, pas trop cher, et dispose de suffisamment de lignes pour tous les scénarios rencontrés. La cible peut être alimentée, et cinq broches sont utilisées pour deux paires UART et un signal d'entrée/sortie (*a priori* pour l'horloge ou la synchronisation).
- **Tag Connect TC2050-IDC (J7).** Si vous ne connaissez pas le connecteur,

Figure 4. Schéma du programmeur/adaptateur de débogage JTAG.



150511 - 11

Tableau 1. Cavaliers JTAG

Cavalier	Position 1	Position 2	Circuit ouvert
VPP (J9)	Position JTG : tension VPPJTAG de J6 ou J8 -> VPP	Position USB : tension USB la plus élevée de J1 ou J2 (5 V USB moins tension de diode) -> VPP	Pas de tension sur VPP
TRGPWR (J10)	Position VPP : VPP -> cible	Position 3.3 V : 3,3 V régulés -> cible	Pas d'alimentation de la cible
SNS (J3)	3,3 V -> JTAG VSNS	Pas d'application	3,3 V pas sur VSNS (Vcc de la cible -> VSNS)
RST (J3)	Remise à zéro de la cible déconnectée ; bouton-poussoir actif	Pas d'application	Remise à zéro pas connectée
TX1_EN (J11)	Pilote et rappel Tx d'USB1 reliés à la cible	Pas d'application	UTX1 -> cible flottant
TX2_EN (J11)	Pilote et rappel Tx d'USB2 reliés à la cible	Pas d'application	UTX2 -> cible flottant
CLK (J12)	CLK-P2 : SWDCLK connecté à la broche 2 du connecteur USB détourné (J4)	CLK-P3 : SWDCLK connecté à la broche 3 du connecteur USB détourné (J4)	Non valide si J4 est utilisé
IO (J12)	IO-P3 : SWDIO connecté à la broche 3 du connecteur USB détourné (J4)	IO-P2 : SWDIO connecté à la broche 2 du connecteur USB détourné (J4)	Non valide si J4 est utilisé

teur Tag-Connect, en route pour une visite guidée [3]. Il est appelé Tag-of-Nails™ (littéralement, prise à clous), et vous permet d'utiliser la méthode des plots de connexion

à « zéro coût par circuit imprimé », tout en fournissant une connexion robuste tant pour le développement que pour la production (**figure 3**). Les câbles s'enfichent sur une

empreinte du circuit imprimé, ce qui donne effectivement une connexion gratuite. Sur la version à dix broches, tous les signaux JTAG sont présents et il y a encore de la place pour une paire UART.

Tableau 2. Connexions USB -> Série du JTAG

Connecteur	Réf. du circuit	Cavalier pilote Tx	Connexions à la cible
J1	USB1	TX1_EN	Broche 4 J5 (Tx circuit -> Rx cible) Broche 6 J5 (Rx circuit <- Tx cible) Broche 7 J7 (Tx circuit -> Rx cible) Broche 9 J7 (Rx circuit <- Tx cible)
J2	USB2	TX2_EN	Broche 12 J5 (Tx circuit -> Rx cible) Broche 14 J5 (Rx circuit <- Tx cible)

- **USB détourné (J4)**. Bien que l'utilisation d'un connecteur standard à une autre fin soit un peu déroutante pour l'utilisateur, et pas vraiment conseillée, elle reste cependant une option intéressante pour le développeur. Il y a tant d'appareils qui n'utilisent leur connecteur micro-USB **que pour l'alimentation**, qu'il est pratiquement impossible de résister à la tentation d'utiliser les broches D+ et D- pour la programmation à deux fils (SWD) des processeurs ARM acceptant ce mode.

Le circuit

La **figure 4** montre le circuit du JTAGD. L'essentiel consiste en plusieurs connecteurs, un régulateur de tension et quelques borniers pour la sélection de divers réglages, dont la configuration est reprise dans le **tableau 1**.

Le circuit reçoit la tension de 5 V du connecteur JTAG ou USB, et peut fournir 3,3 V via le régulateur à faible tension

Tableau 3. Indicateurs à LED du JTAG

Indicateur	Marquage	Couleur	Indication
D1	JPWR	BLEUE	Alimentation JTAG
D2	TRGPWR	VERTE	Alimentation CIBLE
D3	TICK	ROUGE	Horloge / Synchronisation
D5	TX	VERTE	USB1 TX
D6	RX	JAUNE	USB1 RX
D7	USB PWR	BLEUE	Alimentation USB1
D9	TX	VERTE	USB2 TX
D10	RX	JAUNE	USB2 RX
D11	USB PWR	BLEUE	Alimentation USB2

de déchet U1. Le cavalier J9 sélectionne la source de tension 5 V, et J10 permet de sortir 5 V, 3,3 V ou rien. On laisse les borniers en circuit ouvert si on ne souhaite pas alimenter la cible.

Le connecteur USB détourné J4 ne peut accepter que deux signaux, il n'y a plus d'autre ligne disponible. En fait il manque une ligne pour les programmeurs qui doivent détecter la tension de la cible pour la programmation. Le JTAGD pallie à ce manque avec le cavalier SNS de J3, qui renvoie la tension de 3,3 V au programmeur pour émuler ce signal. J12 permet de diriger les données SWD et le signal d'horloge vers D+ ou D- en fonction de la configuration de la cible.

Les connecteurs J5 et J7 autorisent un accès aisé à la remise à zéro de la cible, grâce au bouton-poussoir S1. La broche RST de J3 permet de déconnecter le bouton-poussoir et la résistance de rappel de la cible. J5 possède une autre ligne, qui est en principe utilisée pour l'horloge ou un signal de synchronisation ; la cible peut cependant reconfigurer cette ligne pour n'importe quelle autre sortie.

Il y a deux convertisseurs série-USB sur la platine (U2 et U3), ce qui permet de connecter directement un ordinateur, sans ports USB-série externes. Les deux paires UART sont accessibles via J5, une seule via J7. Les pilotes de transmission de chaque UART peuvent être déconnectés avec J11, pour ne pas surcharger les lignes correspondantes de la cible. On trouvera plus d'informations à ce sujet dans le **tableau 2** et dans la fiche technique du JTAGD, disponible sur le site du magazine [4].

JTAGD s'illumine aussi avec de nombreux indicateurs visuels, qui illustrent ce qui se passe. Le **tableau 3** reprend les fonctions de toutes les LED de JTAGD. Les LED bleues D1, D7 et D11 indiquent quelles sources de 5 V sont sous tension, et la LED verte D2 vous rappelle si la cible est alimentée. Des LED vertes et jaunes montrent l'activité de transmission et réception des deux UART. Une LED rouge montre l'état du signal d'horloge/synchronisation.

Circuit imprimé

Le circuit imprimé à double face montré en **figure 5** accueille tous les composants sur seulement 44x46 mm². Comme la plupart des composants sont des CMS,

Liste des composants

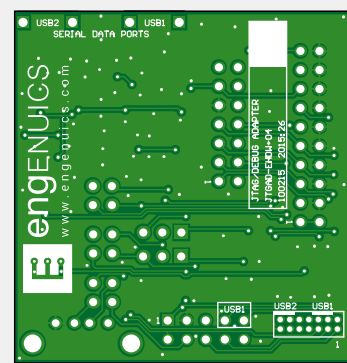
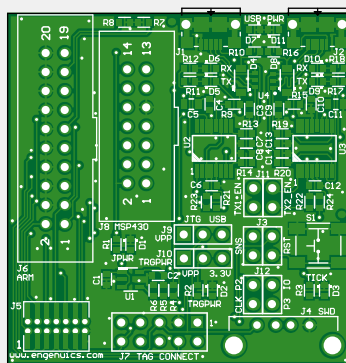


Figure 5. Implantation des composants du circuit JTAGD. Le circuit est disponible complètement assemblé et testé chez Elektor.

Résistances

Toutes CMS 0603, 1%, 0,1 W
 R1, R3 = 1,2 kΩ
 R2, R11, R12, R17, R18 = 390 Ω
 R4, R6, R21, R22, R23, R24 = 100 kΩ
 R5 = 698 kΩ
 R7, R8, R10, R16 = 10 kΩ
 R13, R14, R19, R20 = 33 Ω
 R9, R15 = 0 Ω

Condensateurs

C1, C2 = 1 μF 16 V, X5R, céramique, CMS 0603
 C3, C9 = 33 nF 25V, X7R, CMS 0603
 C4, C6, C10, C12 = 100 nF, 25V, X5R, CMS 0603
 C5, C11 = 10 μF, 10 V, X5R, céramique, CMS 0805
 C7, C8, C13, C14 = 47 pF, 50 V, C0G, CMS 0603

Semi-conducteurs

D1, D7, D11 = LED, bleue, CMS 0603
 D2, D5, D9 = LED, verte, CMS 0603
 D3 = LED, rouge, CMS 0603
 D4, D8 = 1N5819HW-7-F, SOD-123
 D6, D10 = LED, jaune, CMS 0603

U1 = MCP1824T-ADJE/OT, SOT-23 (Microchip)
 U2, U3 = FT230XS-R, SSOP-16
 U4 = IP4220CZ6,125, TSOP-6

Divers

J1, J2 = connecteur micro-USB 2.0 type AB
 R/A pour circuit imprimé, CMS avec pattes
 J3, J11, J12 = embase à 4 broches (2x2), pas de 2,54 mm
 J4 = connecteur USB 2.0 type A R/A pour circuit imprimé
 J5 = connecteur à 14 broches femelle (7x2), CMS, pas de 1,28 mm
 J6 = connecteur à 20 broches en boîtier (10x2), pas de 2,54 mm
 J7 = bornier mâle à 10 broches (5x2), pas de 2,54 mm
 J8 = bornier mâle à 14 broches (7x2), pas de 2,54 mm
 J9, J10 = bornier mâle à 3 broches, pas de 2,54 mm
 8 cavaliers, pas de 2,54 mm
 S1 = bouton-poussoir tactile de 6 mm, un pôle normalement ouvert, CMS
 Module assemblé : 150511-91 sur www.elektor.fr

les lecteurs moins expérimentés auront sans doute quelque difficulté pour le soudage. Le module complètement assemblé et testé est disponible chez Elektor [5], prêt à l'emploi ! Le dessin du circuit imprimé est bien entendu disponible gratuitement [4], si vous avez l'équipement, l'expérience et l'envie de construire votre propre circuit.

Conclusion

La combinaison de divers connecteurs et d'un bon choix d'options fait de cet adaptateur un outil parfait pour les développeurs. Nous sommes heureux de partager notre expérience avec la communauté, et nous espérons vivement qu'il rendra service à quelques-uns.

Amusez-vous bien ! ◀

(150511 - version française : Jean-Louis Mehren)

Liens

- [1] www.segger.com/jlink-debug-probes.html
- [2] www.ti.com/tool/MSP-FET
- [3] www.tag-connect.com
- [4] www.elektormagazine.fr/150511
- [5] www.elektor.fr

sonde de courant pour oscilloscopes

mesure à potentiel flottant de la chute de tension aux bornes de shunts

Alfred Rosenkränzer (Allemagne)

L'oscilloscope prend indéniablement, après le multimètre, la seconde place sur le podium des instruments essentiels en électronique. Universel et d'emploi flexible, il impressionne, car il permet d'afficher des courbes de tension ou de représenter d'autres signaux dynamiques. Toutefois, contrairement au multimètre, un oscilloscope n'est pas conçu pour la mesure de courants. La sonde présentée ici va le permettre !

Si l'on utilise un multimètre pour mesurer un courant, il n'y a pas de mesure directe du courant ; la mesure est celle de la chute de tension aux bornes d'une résistance (d'impédance faible), intégrée dans le multimètre et dont la valeur dépend de la plage de mesure. Cela marche parfaitement, car le multimètre est presque toujours alimenté par piles ; ainsi, il est non seulement à potentiel flottant, mais également bien isolé par son boîtier en plastique. Il existe une variante encore mieux isolée : la pince ampèremétrique ; avec elle, on place une pince annulaire métallique articulée autour d'un conducteur et on mesure, avec un capteur à effet Hall (ou une bobine pour des courants alternatifs purs), le champ magnétique créé.

Les deux méthodes ont leurs inconvénients : elles ne détectent pas l'aspect dynamique des courants variables, et les pinces ampèremétriques conviennent mieux pour des courants intenses dans des câbles et des tresses – de plus, leur plage de fréquences utile est relativement étroite. La capture de signaux dynamiques dans des circuits est le domaine des oscilloscopes – bien qu'ils ne soient pas conçus d'origine pour cela. Nous allons les y aider...

Shunt & oscilloscope

Les oscilloscopes rechignent donc à mesurer des courants. Pour des courants dynamiques, il s'agit souvent de courants de quelques ampères, voire de moins d'un ampère ; l'intéressant est surtout leur

évolution dans un circuit. Souvent, on aimerait connaître le comportement du courant qui traverse un composant en fonction, par exemple, d'une tension à un autre endroit – ou vice versa. Et, pour cela, les pinces ampèremétriques sont plutôt inconfortables. Souvent, on utilise comme shunt, une résistance déjà présente et on mesure simplement la chute de tension à ses bornes, facteur qui permet de déterminer le courant, voire on dessoude l'une des extrémités d'un composant ou on déconnecte la liaison vers une partie du circuit et on place à cet endroit un shunt de valeur de résistance appropriée et constituant une charge suffisante. Simplissime, non ?

Les oscilloscopes ont malheureusement certaines limites. Certains d'entre eux ne sont pas entièrement à potentiel flottant, ce qui ne permet pas d'utiliser l'astuce qui consiste à mesurer le courant sur une résistance en tant que différence

des signaux fournis par deux sondes d'oscilloscope normales. Les choses se compliquent encore si l'on veut, à l'aide d'un oscilloscope ne possédant « que » deux canaux, mesurer simultanément une tension relative à la masse du circuit testé, car il faudrait alors commencer par mettre aussi le shunt ou une extrémité de la résistance traversée par le courant à ladite masse. Il faudra donc inévitablement passer par une mesure à potentiel flottant et différentielle de petites tensions et c'est exactement la fonction que permet d'ajouter l'électronique décrite ici.

Différentiel & potentiel flottant

L'électronicien attentif n'aura pas manqué d'identifier le principe d'une solution « potentielle » : utiliser ce que l'on appelle un amplificateur d'instrumentation. Il mesure (à haute impédance généralement) une différence de tension indépendamment du potentiel de masse. Cepen-

Caractéristiques techniques :

- | | |
|--|--|
| <ul style="list-style-type: none"> • Sonde de courant pour oscilloscope • Tension de mode commun : <ul style="list-style-type: none"> - min. ± 200 V, IC1 sous ± 5 V - jusqu'à ± 600 V, IC1 sous ± 15 V • Réjection du mode commun : <ul style="list-style-type: none"> min. 90 dB • Tensions de mesure : <ul style="list-style-type: none"> - jusqu'à $\pm 4,7$ V, IC1 sous ± 5 V - jusqu'à $\pm 14,7$ V, IC1 sous ± 15 V • Impédance d'entrée : <ul style="list-style-type: none"> - 2 MΩ pour tension via shunt - 500 kΩ pour tension en mode | <p>commun</p> <ul style="list-style-type: none"> • Bande passante : 0 à 130 kHz (-3 dB) • Facteur d'amplification (gain) : $\times 1$ • Amplification : <ul style="list-style-type: none"> - erreur max. 0,001% - max. 5 ppm de non-linéarité - max. 5 ppm de dérive • Tension de décalage (offset) : <ul style="list-style-type: none"> max. 1 mV • Bruit <ul style="list-style-type: none"> - max. 35 μVCC pour 0,01 à 10 Hz - typ. 1,6 μV/$\sqrt{\text{Hz}}$ pour ≥ 100 Hz |
|--|--|



tant les schémas classiques se caractérisent par une plage de tension de mode commun limitée. Mais comme on a, lors de mesures sur des circuits électroniques, affaire à des tensions d'alimentation plus élevées que le classique ± 15 V des amplis op, il va falloir imaginer autre chose.

Analog Devices, fabricant réputé de circuits intégrés analogiques, propose à cet effet une solution parfaite sous la forme d'un ampli op spécial, l'AD8479. Ce type spécial intègre des résistances à la valeur ajustée au laser. La **figure 1** en montre la structure interne. En voici le principe de fonctionnement : une paire de résistances de $1\text{ M}\Omega$ prises respectivement aux entrées négative et positive du circuit attaquent respectivement, en interne, les

entrées inverseuse et non inverseuse de l'ampli op. À la différence des amplificateurs d'instrumentation ordinaires, la résistance de l'entrée non-inverseuse est, par rapport à la masse (broche 5), très faible, de sorte qu'elle divise par 60 la tension d'entrée et ainsi aussi le signal de mode commun. La résistance entre l'entrée inverseuse et la masse (broche 1) est également faible et ajustée de façon à induire une très bonne réjection du signal de mode commun – à en croire la fiche de caractéristiques [1] de pas moins de 90 dB. Résultat : la tension différentielle subit un gain de 1 très exactement et la plage de mode commun voit, théoriquement, sa valeur multipliée par 60. De ce fait, l'étendue de la plage de mode

commun devient plus large, comme le montre le graphique de la **figure 2** : une alimentation de ± 5 V permet une plage de mode commun de plus de ± 200 V (zone rouge). Cela devrait, dans la pratique, être plus que suffisant vu que les sondes et les oscilloscopes sont normalement conçus pour de telles tensions. Dans le cas d'une alimentation de ± 15 V, la plage de mode commun admissible s'étend jusqu'à ± 600 V (zone bleue) et se trouve limitée à cette valeur non pas par le diviseur de tension dans le circuit intégré, mais par l'écartement entre les broches et le boîtier du circuit intégré.

Schéma de la sonde

L'intégration, en principe, de quasiment tout dans l'AD8479, simplifie énormément le schéma de la **figure 3**. La cour autour d'IC1 se limite aux condensateurs nécessaires au découplage (C1 à C4) et à une résistance-série de $50\ \Omega$ prise à la sortie, R1, pour la connexion à un câble coaxial de $50\ \Omega$ – c'est tout.

Il faudra, pour faire en sorte que les excellentes caractéristiques de l'ampli op puissent se confirmer, implanter les régulateurs de tension dans la sonde, c.-à-d. au plus près de l'AD8479 pour obtenir une tension d'alimentation propre et sans parasites. Les deux régulateurs de tension permettent alors d'alimenter le

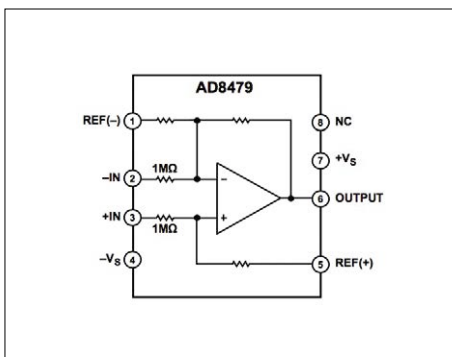


Figure 1. Synoptique de l'AD8479 à résistances intégrées.

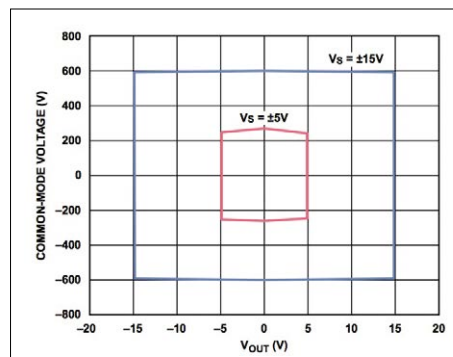


Figure 2. Plage de mode commun en cas d'alimentation sous ± 5 V (rouge) et ± 15 V (bleu).

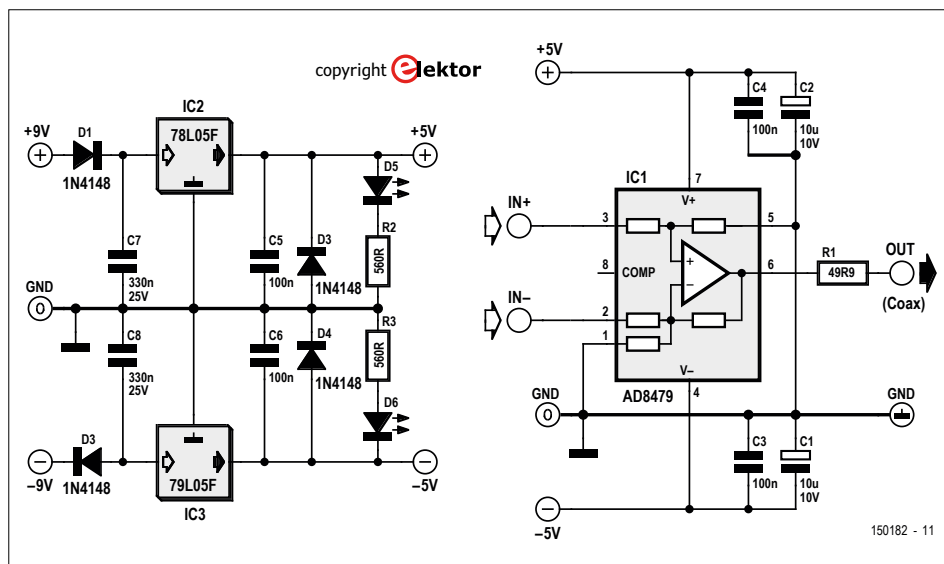


Figure 3. Le circuit de la sonde de courant, régulation de tension comprise.

circuit à l'aide d'une tension non régulée de ± 9 à 25 V. La consommation d'IC1 est minimale ; de ce fait, on peut utiliser des versions CMS pour IC2 et IC3. D1/D2 font office de protection contre une inversion de polarité et D3/D4 empêchent ce que l'on appelle un verrouillage (*latch-up*) des

régulateurs de tension, lorsque les deux tensions chutent de façon inégale, avec un risque alors d'inversion de la polarité d'une sortie.

On pourra, si la plage de mode commun de ± 200 V par rapport à la terre devait s'avérer insuffisante, alimenter IC1 à

des tensions plus élevées. Dès ± 9 V, on atteint une plage de mode commun de ± 400 V et avec ± 15 V, on arrive à la plage maximale de ± 600 V. Il faut alors impérativement veiller à ce que la tension d'alimentation non régulée soit de 4 V au-dessus et au-dessous respectivement de la tension de sortie du régulateur de tension. En outre, C1 et C2 devront alors devenir des types ayant une tension nominale de 25 V. Pour des tensions non régulées supérieures à 25 V, il faudra faire passer à 35 V la valeur de rigidité diélectrique (teneur en tension) de C7 et C8.

Réalisation

Lorsqu'on utilise des CMS, la réalisation sur un morceau de platine à trous est délicate, même si la simplicité du circuit engageait à le faire. C'est pourquoi l'auteur a conçu une platine (cf. **figure 4**) ; des fichiers pour produire le circuit imprimé sont disponibles sur la page Elektor de cet article [2]. Et si déjà nous utilisons un ampli op de type CMS, autant faire de même pour le reste des composants, ce qui est 100% bénéfice pour la miniaturisation de la platine et donc son confort d'utilisation en tant que sonde de shunt. Bien que le circuit soit simple et qu'il n'y ait que peu de composants à implanter, il faudra, lors de la soudure, prendre le temps et pourquoi ne pas utiliser une loupe ; certains des condensateurs et les résistances sont minuscules (type 0603).

Sur la platine, on voit deux cercles (*fixing holes*) en haut à droite. Ils désignent des trous à percer pour fixer, à l'aide d'un serre-câble, la sortie du câble coaxial. S'il est vrai que la sonde de courant n'a, de par l'AD8479 qu'une bande passante limitée de 130 kHz, il n'en reste pas moins qu'un câble coaxial sera utile pour une bonne qualité de signal et la durabilité. En raison de la bande passante limitée, l'entrée de l'oscilloscope ne devra pas être terminée sous 50 Ω , mais peut rester à la haute impédance classique de 1 M Ω , de sorte que le signal n'est pas alors divisé par deux.

Voici comment procéder pour le soudage du câble coaxial directement sur la platine : du côté de l'extrémité libre, on commence par dénuder 11 mm de l'isolant extérieur avant de raccourcir à 6 mm le blindage tressé devenu visible et de l'étamer. L'isolation intérieure est enlevée, ce qu'il faut pour que le conducteur

Liste des composants

Résistances :

Toutes CMS 0603

R1 = 50 Ω

R2, R3 = 560 Ω

Condensateurs :

Tous céramique

C1, C2 = 10 μ / 10 V*, CMS 1206

C3-C6 = 100 n / 50 V, CMS 0603

C7, C8 = 330 n / 25 V*, CMS 1206

Semi-conducteurs :

D1-D4 = 1N4148, DO214AA

D5 = LED, rouge, CMS 0805

D6 = LED, bleue, CMS 0805

IC1 = AD8479, SOIC-8

IC2 = 78L05F*, SOT89

IC3 = 79L05F*, SOT89

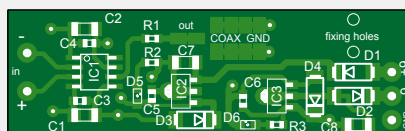


Figure 4. Circuit imprimé et implantation. La paire de trous à droite sert à fixer le câble de raccordement par serre-câble.

Divers :

Câble coaxial*

Câble de microphone*, à deux conducteurs + blindage

Pointes de sonde*

Circuit imprimé

* cf. texte

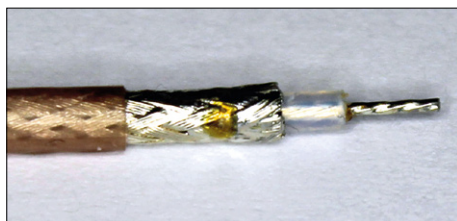


Figure 5. Voici un câble coaxial correctement dépouillé de son isolation et étamé.

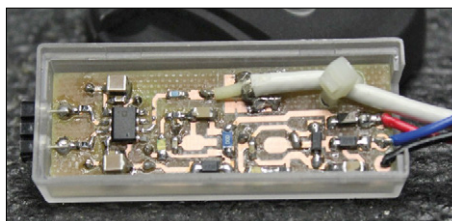


Figure 6. Un premier prototype de la sonde de courant, dotée d'un petit boîtier en plastique. Ici, seul le câble coaxial est fixé à l'aide d'un serre-câble.

intérieur dénudé dépasse de 3 mm ; on voit en **figure 5** ce dont il retourne. On place ensuite l'extrémité du câble préparé au bon endroit de la platine et on soude d'abord le conducteur interne. Ensuite, le blindage richement étamé est soudé aux deux plots de masse allongés.

Pour l'alimentation, on pourra utiliser un câble de microphone flexible à deux conducteurs + blindage. Il est soudé aux trois plots en bas à droite. À l'entrée, on connecte des pointes de sonde standard qui, pour de courtes longueurs, ne doivent pas nécessairement être blindées, vu que les impédances des sources sont en général assez faibles. On peut aussi rendre les sondes enfichables à l'aide, par ex., d'un bout d'embase au pas de 5,08 mm. Une fois la carte montée dans un petit boîtier en matière plastique approprié et les câbles passés dans les trous, il reste à doter l'extrémité libre du câble coaxial d'un connecteur BNC. La connexion enfichable de l'alimentation dépend de la source de tension prévue. Celle-ci n'aura pas à fournir une forte puissance, quelques mA suffisent amplement.

Et le reste

On voit en **figure 6** l'un des premiers prototypes de la sonde de courant. Une fois la sonde « mise en boîtier », elle est immédiatement utilisable. Il n'y a rien à régler, ni quoi que ce soit à configurer. Attention : si l'alimentation d'IC1 est de ± 5 V, et le gain unitaire, on ne peut mesurer que des tensions sur shunts de $\pm 4,7$ V, vu que la plage de tension de sortie a exactement cette taille. Si l'on a besoin, pour les cas particuliers, d'une plage de tension plus grande (ce

qui n'arrive pratiquement jamais lors de mesures de courant), il faudra, comme décrit plus haut, augmenter la tension d'alimentation d'IC1 sans perdre de vue la tenue en tension de C1 et C2.

La sonde est, tous comptes faits, un accessoire pour tout oscilloscope ; elle le dote d'une option de mesure de courant sans risque de problèmes, et peut se targuer d'un rapport utilité/prix hors pair. ◀

(150182 - version française : Guy Raedersdorf)

À propos de l'auteur

Cela fait 32 ans qu'Alfred Rosenkränzer opère en tant qu'ingénieur de développement, au départ dans le monde de la TV professionnelle. Il développe, depuis la fin des années 90, des circuits numériques à grande vitesse et analogiques pour des testeurs de circuits intégrés.

Liens

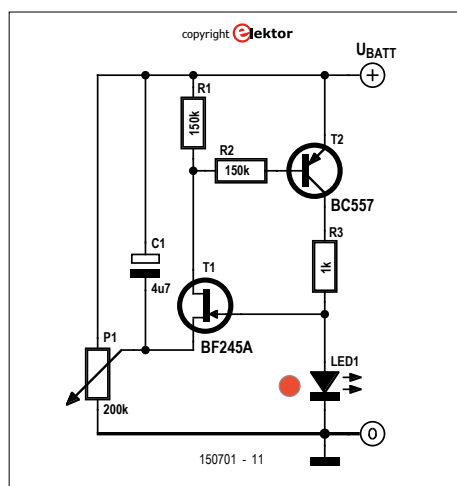
- [1] Fiche de caractéristiques AD8479 : www.analog.com/static/imported-files/data_sheets/AD8479.PDF
- [2] Page de cet article : www.elektormagazine.fr/150182

indicateur de charge faible pour pile de 9 V

Rainer Giedigkeit

Depuis des années, j'utilise, pour mon système de mesure d'enceintes, l'amplificateur pour microphone à électret (publié dans Elektor en 1977), alimenté par une pile de 9 V. Pour ne pas avoir à déterminer le niveau de charge de la pile à l'aide de mesures, je l'ai doté d'un indicateur de pile faible. Dans ce circuit (**figure 1**), il n'y a ni microcontrôleur, ni même d'amplificateur opérationnel.

La LED rouge remplit une double fonction : signaler que la pile approche de sa tension minimale, et servir en outre de référence de tension. T1, un JFET, compare cette tension de référence sur LED1 avec la tension de pile réduite par l'ajustable P1. Si la tension de la pile tombe en



dessous du seuil défini par P1, T1 devient conducteur. Le courant de drain à travers R1 abaisse la tension de base de T2, un transistor PNP au courant de base limité par R2. Le courant de collecteur à tra-

vers T2 résultant, limité d'ailleurs par R3 à 7 mA environ, fait s'allumer la LED1. Si la tension tombe sous la tension de seuil de la pile, C1 provoque le clignotement de la LED. Plus la tension de la pile faiblit, plus le clignotement s'accélère. Lorsque la tension atteint de l'ordre de 1 V en dessous du seuil défini, la LED s'allume fixe. Ce n'est que lorsque la tension de pile tombe en deçà de 3 V que la LED va s'éteindre progressivement.

Si l'on fait passer la valeur de P1 de 200 kΩ à 100 kΩ, un niveau de pile faible n'est plus signalé par un clignotement, mais, dès le départ, par un allumage fixe. Cependant, la valeur de seuil présente alors une hystérésis, due à l'augmentation de la tension aux bornes de LED1, lorsque cette dernière s'allume. ◀

(150701 - version française : Guy Raedersdorf)

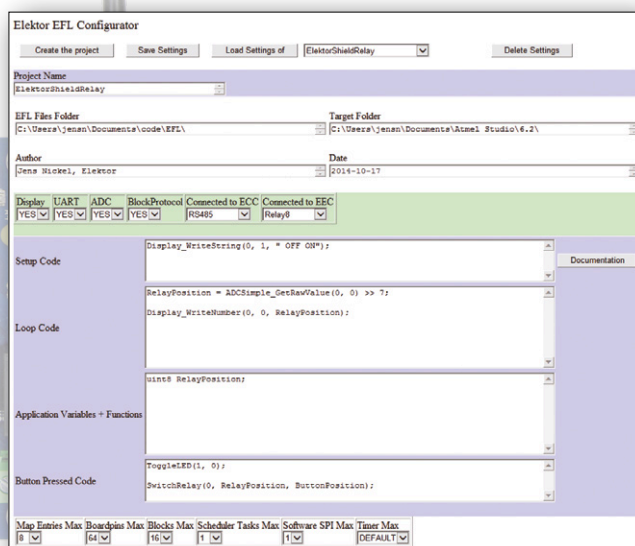
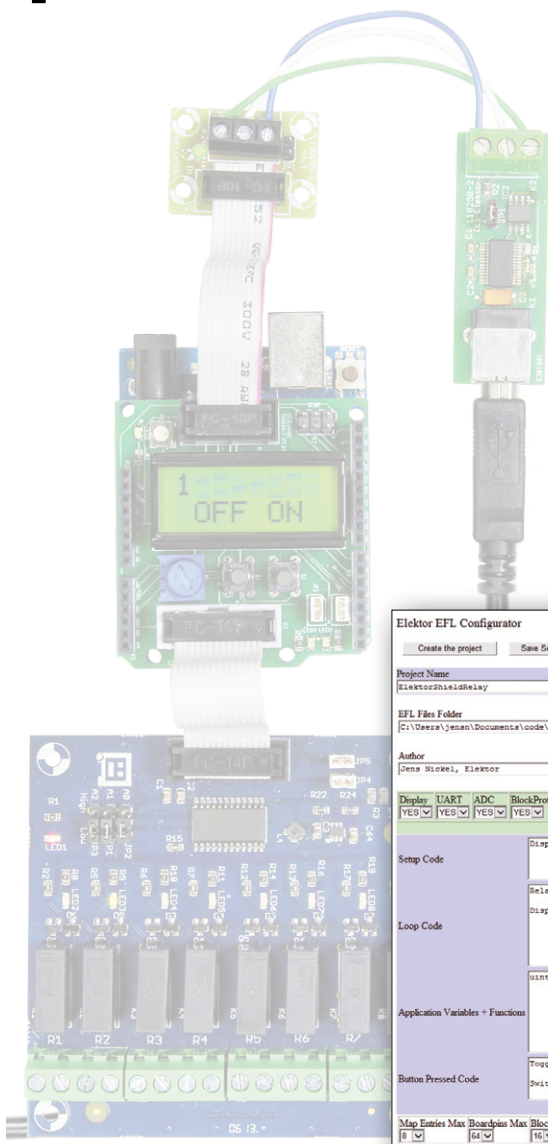
pondre un projet en C d'un seul clic

configurateur scripté pour logiciel modulaire sous Atmel Studio

Jens Nickel (Elektor)

Les bibliothèques logicielles modulaires telles qu'EFL ont de nombreux avantages. Certaines étapes de la création d'un projet dans un environnement de développement peuvent cependant décourager le débutant. D'un clic de souris, le

logiciel à base de script décrit ici produit un nouveau projet EFL. Dans la foulée, il pond le code d'application et toutes les commandes de configuration. Il est doté d'organes pour réduire la consommation de mémoire. En prenant l'exemple d'une commande de relais avec affichage, nous optimiserons un projet pour de (plus) petits microcontrôleurs.



Il y a quelque temps, nous avons découvert les avantages de bibliothèques logicielles modulaires telles qu'EFL [1]. Chaque module de logiciel est déterminé par les propriétés du microcontrôleur (μ C) ou de la carte auxquels il correspond, mais il est indépendant du reste du matériel du projet. Un projet logiciel personnel peut ainsi combiner les modules sans avoir à en modifier le code ; c'est un peu comme l'interconnexion de modules matériels, prêts à l'emploi, indépendants les uns des autres, qui permet de réaliser, en quelques minutes, un prototype fonctionnel. Malheureusement cette indépendance par rapport au matériel se paye par un nombre élevé de fichiers individuels à

intégrer dans le projet (EFL) à réaliser. Il nous faut, à chaque fois, une paire de fichiers (.h/.c) pour le μ C, la carte du μ C et chacune des cartes d'extension, auxquels s'ajoutent de nombreux fichiers de bloc, qui, pour les périphériques tels qu'un afficheur, mettent à disposition des fonctions de bas niveau (qui nous affranchissent p. ex. de nous préoccuper du câblage entre μ C et l'afficheur).

Nous avons besoin, en outre, de bibliothèques de plus haut niveau (*Libraries*) qui facilitent la programmation, car sur toutes les cartes tout périphérique du même type se laisse adresser par les mêmes fonctions. Et il nous faut aussi,

bien entendu, les fichiers *Common*, la base d'EFL.

Pas à pas

Malheureusement, dans des environnements de développement modernes tels qu'Atmel Studio, il ne suffit pas de traîner les fichiers requis vers un dossier Windows. Au contraire, tous les fichiers de code doivent être liés au cœur même du projet par des commandes Atmel Studio. Si nous démarrons Atmel Studio avec un projet « nu », il faut commencer par créer les trois dossiers de projet EFL, *Common*, *Hardware* et *Libraries*. Nous devons ensuite informer le compilateur des chemins d'accès à ces dossiers.

Mais, ce n'est pas tout, bien sûr. Dans le fichier principal du progiciel (*firmware*), dans lequel se trouve en fait le code de l'application proprement dit, il faut que tous les fichiers d'en-tête (*Header*) de la carte, des extensions et de la bibliothèque, soient inclus par une commande *Include*. Et, pour finir, il ne faut pas non plus oublier les fonctions d'initialisation et de configuration (*Init* et *Setup*) pour les modules matériels (*Hardware*) et de bibliothèque (*Library*).

Il est bien sûr possible, à l'occasion, de créer un projet EFL adapté (ou d'utiliser l'une de nos démos), et, par simple copie de l'ensemble du dossier de projet d'Atmel Studio, de produire de nouveaux projets modifiables ensuite selon les besoins. Malheureusement, le nom du projet initial reste ce qu'il est. Renommer manuellement le projet dans *Solution Explorer* ne résout le problème qu'en partie, car il restera des résidus du nom du projet initial dans le projet.

Pour produire des projets de démo propres pour les articles EFL précédents, moi aussi j'ai dû patiemment passer par toutes les étapes décrites ci-dessus.

Trois fichiers pour un projet

À la recherche d'une solution, j'ai commencé dans Atmel Studio par créer un projet en C 100% inédit ; je pris ensuite le temps d'examiner les dossiers et les fichiers créés. Tous les fichiers se trouvent

dans un dossier principal portant le nom du projet (choisi par l'utilisateur avant la création du projet, *GccApplication1* étant le nom par défaut). On découvre, dans le dossier principal, un fichier baptisé du nom du projet et doté de l'extension *.atsln* et un autre dossier portant le nom du projet (**fig. 1**). Ce dernier contient deux fichiers, qui portent eux aussi le nom du projet, l'un à l'extension *.cproj* et l'autre ayant celle de *.c*. On découvre en outre un dossier vide appelé *Debug*.

Les fichiers *.atsln* et *.cproj* peuvent être ouverts à l'aide d'un éditeur de texte ; c'est du format XML pur sang. Tout ceci constitue déjà un grand pas sur le chemin de la création automatique d'un tel projet (hors Atmel Studio). Tous les langages de programmation pour PC du commerce permettent en effet de créer des dossiers et des fichiers avec un certain nom et aux contenus spécifiques. Un examen plus approfondi du fichier *.atsln* montre que seul importe ici le nom du projet, le reste des informations ne se réfèrent pas directement au projet. Nous pouvons donc, pour créer un fichier *.atsln* sur mesure, utiliser comme modèle un fichier *.atsln* produit par Atmel Studio. Nous commençons par en lire le texte à l'aide d'un logiciel maison. Ensuite, nous remplaçons le nom du projet qu'il comporte dans le texte (XML) par le nom de projet souhaité et enregistrons le fichier avec le nouveau contenu et le nouveau nom (donc nom de projet + extension

.atsln). Le fichier *.atsln* n'existe d'ailleurs que parce qu'Atmel Studio tire ses racines de Visual Studio de Microsoft. On utilise cet IDE pour, dans le domaine de la programmation PC, entre autres, VB.NET et C#. Visual Studio fait une distinction entre une *solution* et un *projet*, une solution pouvant contenir plusieurs projets. Dans notre cas, une solution ne comporte, fondamentalement, toujours qu'un seul projet ; ils portent donc tous les deux le même nom.

Un coup d'œil au fichier *.cproj* nous apprend que, dans ce fichier, se trouvent stockées toutes les informations pertinentes ayant trait au projet Atmel Studio proprement dit. À titre de comparaison, j'ai examiné un fichier *.cproj* d'un projet EFL produit manuellement (on en voit, en **fig. 2**, une partie dans l'éditeur Notepad++). Tout ce dont nous avons besoin s'y trouve sous la forme d'éléments XML :

- Les noms des sous-dossiers du projet EFL (*Hardware*, *Libraries*, *Common*).
- Les chemins d'accès correspondants pour le compilateur.
- Tous les fichiers de code source faisant partie du projet.
- Les symboles définis pour le préprocesseur. Jusqu'à présent, nous avons déjà utilisé, dans l'EFL, la définition `F_CPU = 16000000UL` (fréquence d'horloge de l'ATmega328 de l'Arduino Uno). Par le biais de ce symbole, il est possible par exemple

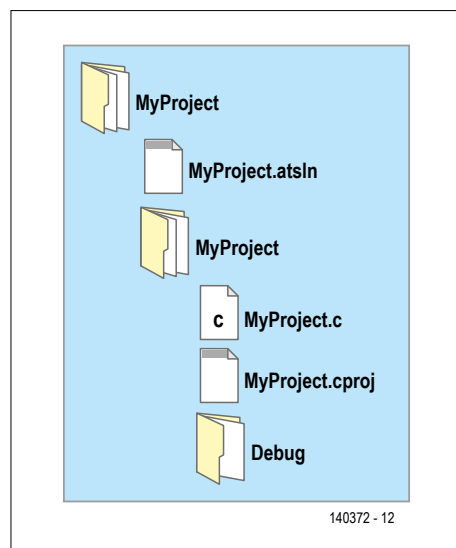


Figure 1. Voici l'arborescence (de fichiers) que fournit Atmel Studio lors de la création d'un nouveau projet en C.

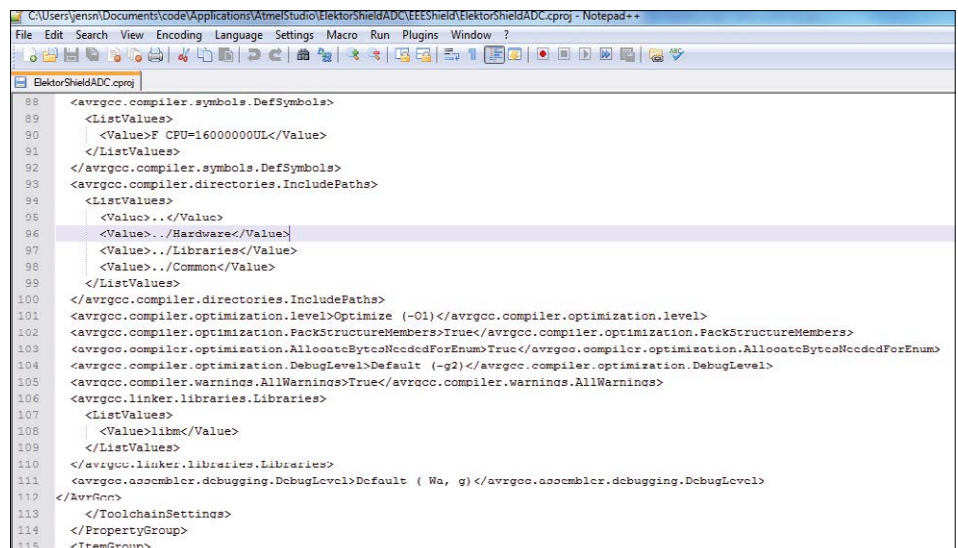


Figure 2. Le fichier *.cproj* contient toutes les informations ayant trait à un projet Atmel Studio, au format XML.

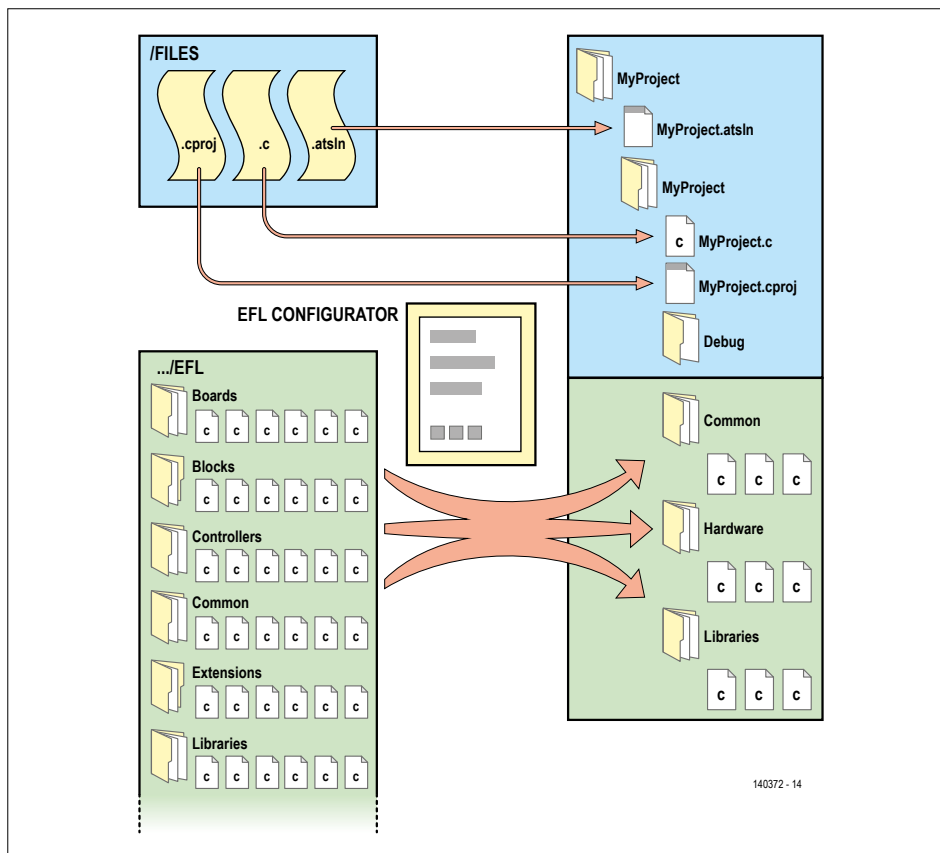


Figure 3. À partir de trois fichiers-modèles et du dossier comportant tous les fichiers EFL, notre configurateur crée un projet EFL pour Atmel Studio complet.

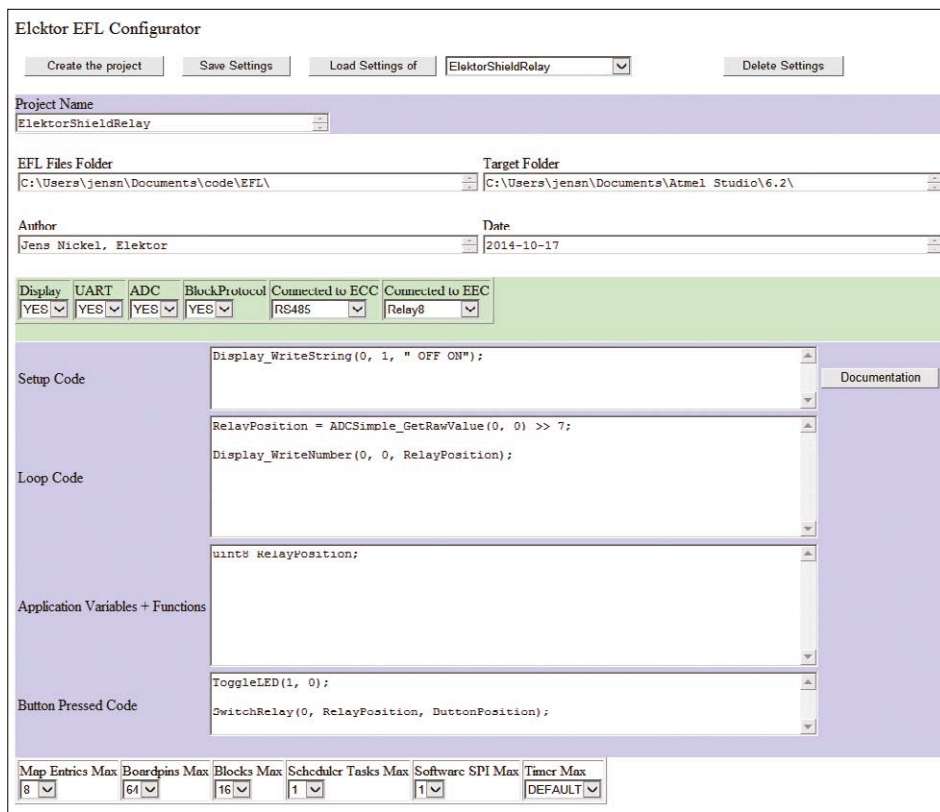


Figure 4. Si EFL Configurator ne mérite pas un prix de beauté, il ne se charge pas moins de bien des tâches. Quelques clics suffisent pour produire des versions du progiciel différentes pour voir quel est l'effet sur les besoins en code et en RAM.

de calculer dans le code du µC le contenu du registre de paramétrage pour régler le taux de transmission de l'UART.

Il nous reste encore à jeter un coup d'œil au fichier .c. Il s'agit ici du fichier principal du code source avec la fonction *main*, déjà créé par Atmel Studio. Ce fichier est, bien entendu, un fichier texte lui aussi, au contenu facile à manipuler.

La **figure 3** montre, sur la partie de droite, à quoi doit ressembler, dans Atmel Studio, la structure de dossiers d'un projet EFL.

On pourra, pour la création d'un tel projet (ou de tout autre projet en C modulaire), dans Atmel Studio, opter maintenant pour l'approche suivante. Dans un dossier spécifique, nous déposons trois fichiers-modèles à utiliser comme base pour les fichiers ...atsin, ...cproj et ...c. Notre logiciel PC commence par créer les dossiers requis pour le projet Atmel Studio. Suivent la lecture successive des fichiers-modèles, la manipulation appropriée de leur contenu et enfin l'enregistrement sous un nouveau nom au bon endroit au sein de l'arborescence du projet. Pour nous faciliter la vie lors de l'édition automatique des fichiers-modèles, nous intégrons dans le texte des expressions faciles à retrouver telles que AAAAA ou BBBBB. Elles seront ensuite remplacées par les contenus requis, tels que p. e. les éléments XML pour l'intégration (liaison) des fichiers de code source dans le projet.

Nous créons également, pour le projet EFL, les sous-dossiers *Common*, *Hardware* et *Libraries* (à côté du sous-dossier *Debug*) et copions à cet endroit tous les fichiers EFL nécessaires. Le logiciel PC sait où se trouvent les fichiers concernés dans l'arborescence des fichiers EFL (*Controllers* (contrôleurs), *Boards* (cartes), *Blocks* (blocs), etc.).

Script

Maintenant, nous étions tout près d'un logiciel PC approprié. En principe, on pourrait utiliser n'importe quel langage de programmation, dès lors qu'il dispose de commandes pour la création de dossier, la lecture et l'enregistrement de fichiers de texte et la manipulation de textes et de chaînes. J'ai opté ici pour un *Framework* à base de scripts, de mon cru, que nous

utilisons, chez Elektor, pour, entre autres, créer des listes et des fichiers à partir des données de notre système de planification rédactionnel.

Pour une application locale, comme dans le cas du logiciel ci-après, les scripts sont enregistrés sous forme de fichiers texte. Ils sont lus et exécutés par une application .exe (l'interpréteur voire le « Processeur de script »). Le premier script lu crée une interface utilisateur de type HTML, que l'application .exe visualise dans un élément de commande Web. En cas d'action sur un bouton de l'interface HTML, un rien de Javascript transmet l'événement « vers l'extérieur » à l'application exécutable qui peut alors appeler un autre script. Cela semble compliqué, mais présente un avantage majeur. Si nous voulons doter notre logiciel, d'éléments de l'interface utilisateur et/ou d'autres fonctions additionnelles, il nous suffit de modifier le texte des fichiers script, le fichier .exe ne change pas. En principe, les scripts sont aussi indépendants de la plateforme. Le processeur de script (créé sous C# pour .NET) n'existe, jusqu'à présent, que pour Windows.

Le fichier .exe, de même que les fichiers .dll et scripts additionnels requis, se trouvent tous dans un dossier commun, que l'on pourra télécharger à partir de la page du projet ayant trait à cet article [2]. Les scripts sont regroupés, de pair avec les fichiers de configuration et d'autres fichiers appartenant à l'application de script proprement dite, dans le sous-dossier « APP ». Nous pouvons aussi, dans ce « dossier APP », placer nos trois fichiers-modèles (dans le sous-dossier APP\FILES\).

Les programmeurs PC invétérés ne manqueront pas de jeter un coup d'œil aux fichiers script (dans le sous-dossier APP\SCRIPTS\) et aux fichiers de données (APP\DATA\). Dans les fichiers de données, on enregistre, par exemple, des paramètres personnalisés. L'encadré donne un aperçu du langage de script utilisé.

Création de projet ...

Un clic sur le fichier .exe lance l'application appelée EFL Configurator. Première étape : création, puis affichage de l'interface utilisateur HTML (fig. 4).

Ne voyons pas trop grand : en raison de contraintes de temps, il m'a fallu limiter le configurateur à une configuration matérielle spécifique. On pourrait imagi-

ner de nombreuses extensions : commencer par le choix des µC et carte utilisés, passer ensuite à une nouvelle interface où on pourrait choisir parmi les cartes d'extension appropriées. Pour le moment, EFL Configurator est adapté à la carte *shield* d'Elektor présentée dans le numéro double d'été 2014 [3] et installée sur un Arduino Uno. Il est possible, au choix, d'y connecter :

- un module ECC-RS485 et/ou
- un module EEC à huit relais ou
- un module EEC à CAN 16 bits externe

Les trois modules et des projets de démonstration correspondants ont été décrits dans l'article précédent [1].

Tout en haut de l'interface utilisateur on attribue un nom au projet. Dans les champs sous-jacents, on saisit les chemins donnant accès, sur son propre ordinateur, à la version la plus récente du fichier-code EFL enregistrée (téléchargeable sous [2]). Le chemin doit amener au dossier *EFL* et doit donc se terminer par « \EFL\ ». Il faut, en outre, spécifier le cheminement au bout duquel sont enregistrés les projets créés par Atmel Studio. On trouve ensuite les champs de texte pour l'auteur du projet et la date, dont on pourra bien entendu adapter le format à son goût. On retrouvera ces deux informations sous forme de commentaires dans le code source.

En dessous on trouve des champs pour définir les bibliothèques à utiliser dans le projet. Le projet inclut toujours la bibliothèque *LEDButtonEFL*. Lorsque l'on veut utiliser un écran, l'UART pour la communication ou une entrée analogique, on y ajoute les bibliothèques *DisplayEFL*, *UARTInterfaceEFL* et *ADCSimpleEFL*. Le réglage par défaut est ici trois fois YES, pour pouvoir utiliser toutes les fonctions de la carte *shield* d'Elektor et de l'Arduino (afficheur, potentiomètre, UART) de façon simple. *BlockProtocol* [4] permet, à partir d'un programme de terminal, de commander à distance les unités périphériques sur la carte et les modules d'extension. Dans les champs déroulants voisins, on paramètre les modules d'extension à utiliser.

... et gestion de projet

Dans les champs de texte se trouvant en dessous, on peut saisir le code source

spécifique à l'application du projet EFL. Dans le champ de texte tout en haut, on saisit toutes les fonctions à appeler au lancement de l'application. On trouve ensuite le code à exécuter en permanence. Les variables et fonctions (globales) spécifiques à l'application sont mises dans le champ de texte suivant. Dans le dernier champ de texte, on saisit le code à exécuter en cas d'action sur une touche quelconque (du premier bloc de touches) (la touche enfoncée, se retrouvant dans la variable *ButtonPosition*). Une action sur le bouton *Documentation* se traduit par l'ouverture de la documentation *Doxygen* dans *Internet Explorer*. On peut y retrouver la syntaxe des fonctions de la bibliothèque.

Il est bien entendu également possible de ne rien saisir dans les champs de texte et de produire un projet EFL ne comportant que les initialisations et instructions d'installation requises pour les bibliothèques. Il est en effet possible de saisir directement dans Atmel Studio Editor le code propre à l'application.

Le fait de programmer l'application dès le départ dans EFL Configurator a cependant un avantage. Un clic sur *Save Settings* permet d'enregistrer tous les paramètres, le code y compris, dans une petite base de données texte (Cf. APP\DATA\tblProjects.txt). Le jeu de données est toujours enregistré sous le nom attribué au projet. Le menu déroulant tout en haut permet de sélectionner tous les projets déjà enregistrés, et *Load Settings of a* pour effet de charger le projet dans EFL Configurator, paramètres et code inclus.

On dispose ainsi d'une petite gestion de projet EFL qui – et cet aspect pourrait avoir son importance dans le futur – est indépendante de l'environnement de développement utilisé. Sur la base des paramètres et du code de l'application saisis et enregistrés ici avec *Atmel Studio* comme cible, un script similaire pourrait, plus tard, produire p. ex. un projet Eclipse. Ainsi, pour créer des applications intégrées, les programmeurs peuvent non seulement aller au-delà des frontières de contrôleurs et de cartes, mais aussi sortir des limites des environnements de développement.

Premier essai

Pas d'article EFL sans exemple : en tant que base pour une gestion personnelle

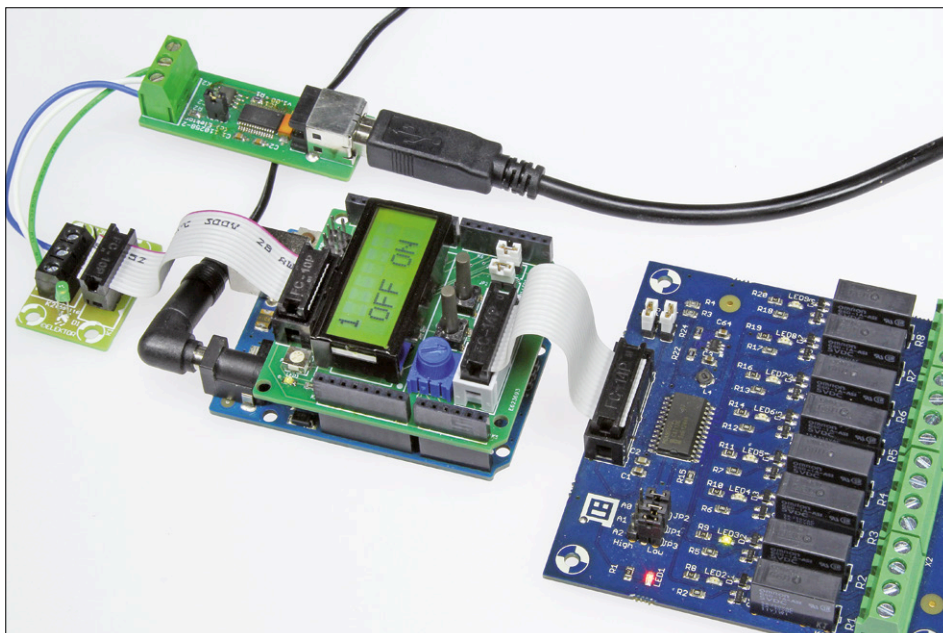


Figure 5. Nous allons, à titre de premier exemple, produire le code pour la commande du relais, décrite dans l'article précédent [1].

de projets, nous avons commencé par préparer les trois applications de l'article précédent [1]. Comme premier exemple, nous allons prédéfinir l'application *ElektorShieldRelay* (fig. 4). On découvre, en figure 5, le matériel du projet, décrit dans l'article précédent. Il est possible,

par le biais des boutons, du potentiomètre et de l'affichage de la carte *shield* d'Elektor, de commander huit relais, mais aussi, depuis un PC et via un programme terminal ; le protocole utilisé est le tout simple BlockProtocol.

Si nous appuyons maintenant sur le bou-

ton *Create Project*, le projet EFL complet devrait être pondu à l'emplacement voulu. Un clic sur le fichier .atsln nouvellement créé entraîne l'ouverture du projet sous Atmel Studio (fig. 6). Dans Solution Explorer, on voit dans les trois dossiers EFL tous les fichiers nécessaires. Le fichier principal du code source contient toutes les commandes *#include*, *Init* et *Setup* pour les bibliothèques, ainsi que le code spécifique à l'application.

Le projet devrait alors se compiler sans problème et se laisser charger dans le µC. Pendant la phase de développement, gardez ouverts en parallèle EFL Configurator et Atmel Studio. Essayez de modifier le code de l'un des champs de texte d'EFL Configurator et appuyez à nouveau sur le bouton *Create Project*, sans avoir modifié le nom du projet. Dans le projet Atmel Studio produit auparavant, le fichier principal du code est alors remplacé par un nouveau fichier qui s'affiche, modifié (après confirmation dans une boîte de dialogue) dans Atmel Studio. Nous pouvons alors compiler et essayer le nouveau code.

Moins de mémoire

Nous pouvons maintenant, à l'aide de la gestion de projet, concevoir aisément différentes versions d'une application. Nous allons l'utiliser pour optimiser un tant soi peu notre projet EFL quant à ses besoins en mémoire. Tout au bas du configurateur, nous trouvons quelques champs de sélection, devant nous permettre d'influer sur les besoins en RAM du projet. Premier pas, la taille des tableaux EFL [5], à savoir la cartographie de fonctions du µC telles que CAN, UART et Co., le tableau Boardpin pour la cartographie du câblage et le tableau Block, qui décrit tous les blocs périphériques et connecteurs d'extension.

Si nous avons programmé le progiciel décrit plus haut dans le µC, nous pouvons, par le biais de BlockProtocol, non seulement commander les relais, mais aussi afficher, dans un programme de terminal, le contenu des tableaux EFL utilisés à cet instant (commande `x <CR>`) [4]. Nous constatons qu'il nous faut 4 entrées dans le tableau de cartographie, de l'ordre de 60 entrées Boardpin et 12 entrées dans le tableau Block (fig. 7). Pour le moment il est réservé de la mémoire pour 8 entrées de cartographie (à 4 octets chacune), 64 entrées Boardpin (2 octets) et

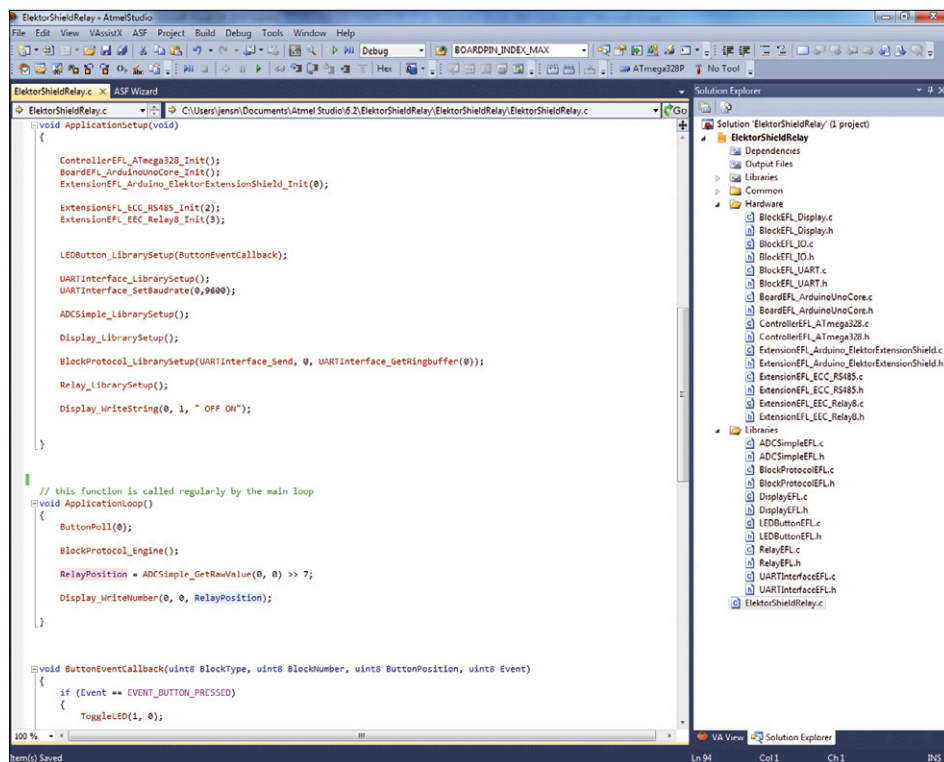


Figure 6. Hourra, cela a fonctionné : le projet créé automatiquement sous Atmel Studio.

16 entrées (à 5 octets chacune) dans le tableau Block. Nous pouvons maintenant optimiser tout cela en paramétrant, dans EFL Configurator, la taille du tableau de cartographie à 4 et à 12 celle du tableau Block. De plus, le code du µC attribue, à une interface SPI logicielle, une mémoire tampon en anneau de pas moins de 64 octets. Les trois temporisateurs (*timer*) se voient, eux aussi, attribuer une structure relativement gourmande en mémoire. Il en va de même pour le *Scheduler* (planificateur), qui peut appeler des fonctions pilotées par temporisation (pas encore décrit). Nous allons, par conséquent, optimiser le tout, comme le montre la **figure 8**, sachant que nous n'avons besoin ni du temporisateur ni du SPI.

Si nous créons maintenant un projet modifié, il y aura, dans le fichier .cproj, enregistrement des symboles correspondants (les symboles que l'on peut demander à Atmel Studio d'afficher (clic droit sur le nom du projet, et sélection de *Properties*, *Toolchain* et *Symbols*). Lors de la compilation du projet, ce sont maintenant les tailles optimisées des tableaux EFL qui sont prises en compte. Le résultat (dans la fenêtre *Out*) mérite le détour : il ne nous faut plus, au lieu des 1091 octets à l'origine, que 923 octets de RAM. Et cela sans avoir perdu la moindre fonctionnalité du programme. Comme les besoins en mémoire Flash sont, à chaque fois, de 16 260 octets. Le projet peut d'ores et déjà trouver place dans un ATmega168.

Nous pouvons dégraisser davantage en nous passant de la télécommande des relais. Nous oublions le module RS-485 et paramétrons en outre à NO les champs déroulants au niveau des BlockProtocol et UART.

Après compilation du projet, le code est réduit à 9072 octets et les besoins en mémoire RAM à 389 octets. Nous n'en avons pas moins conservé une superbe commande de relais par potentiomètre et boutons, sans oublier les retours d'infos sur l'affichage. Malheureusement, les besoins en mémoire Flash dépassent encore les 8 K magiques, qui permettraient à notre projet de trouver place dans un ATmega88 (ou un ATtiny85). On a cependant vite fait d'identifier, dans le code d'application, un gros consommateur de mémoire : la fonction *Dis-*

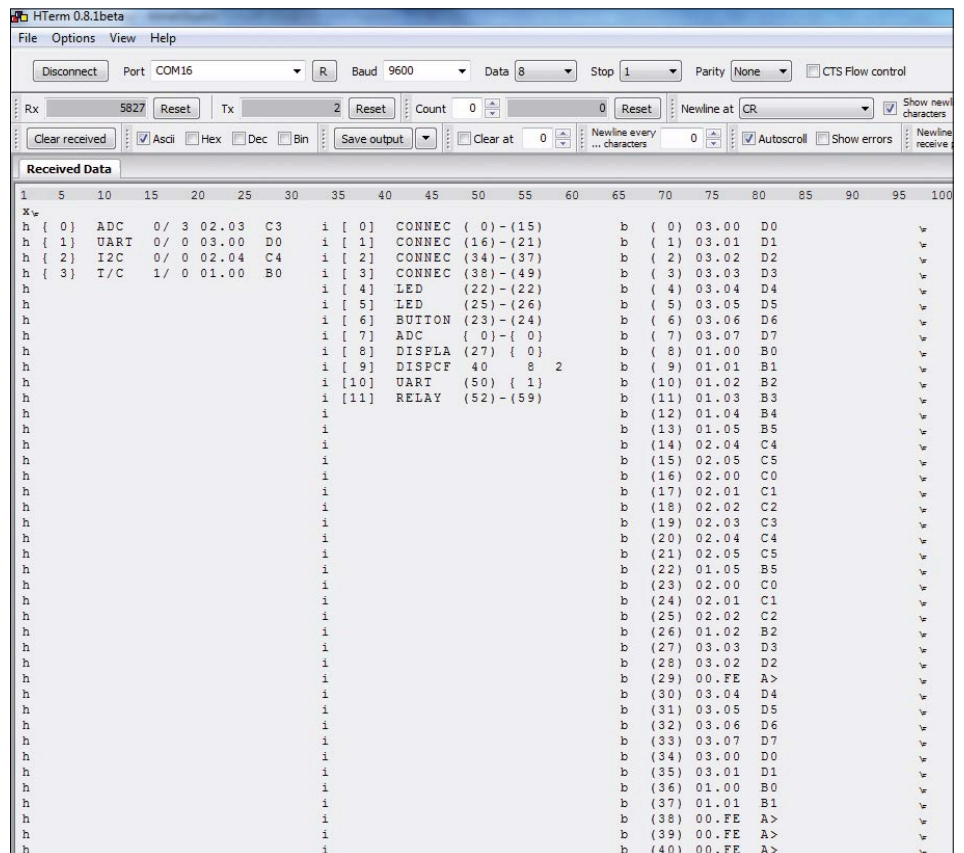


Figure 7. Dans le programme du terminal, nous voyons quelle doit être la taille des tableaux EFL dans le cas de notre application. Nous pouvons, maintenant, optimiser les besoins en mémoire RAM, de sorte que le progiciel tourne également avec des µC plus petits.

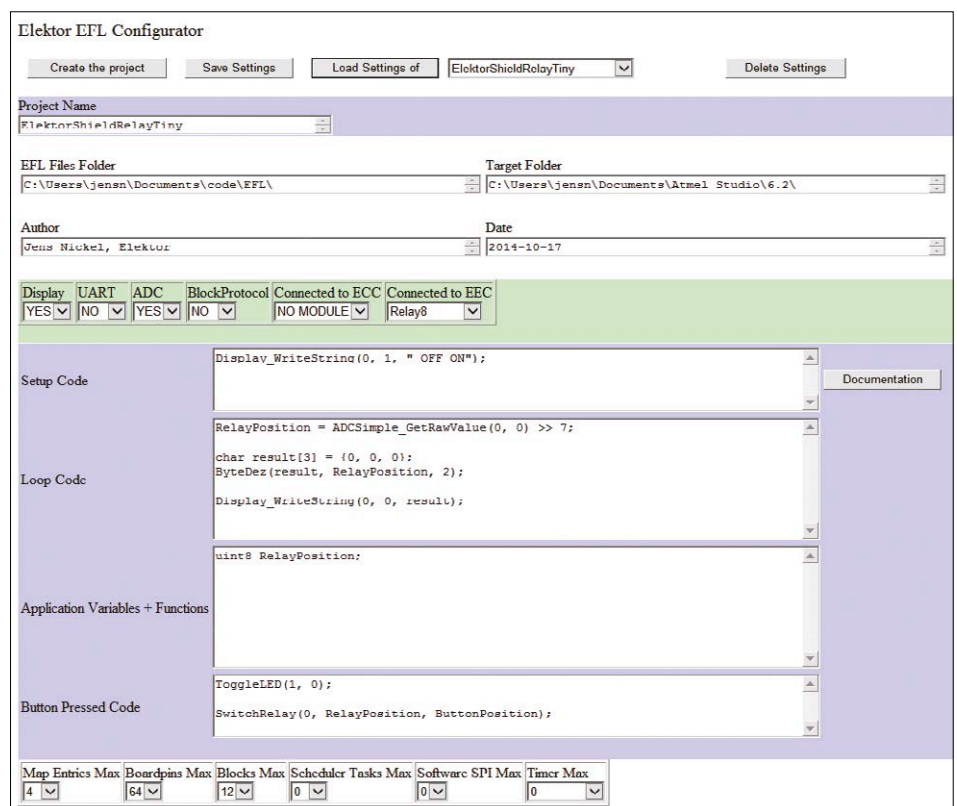
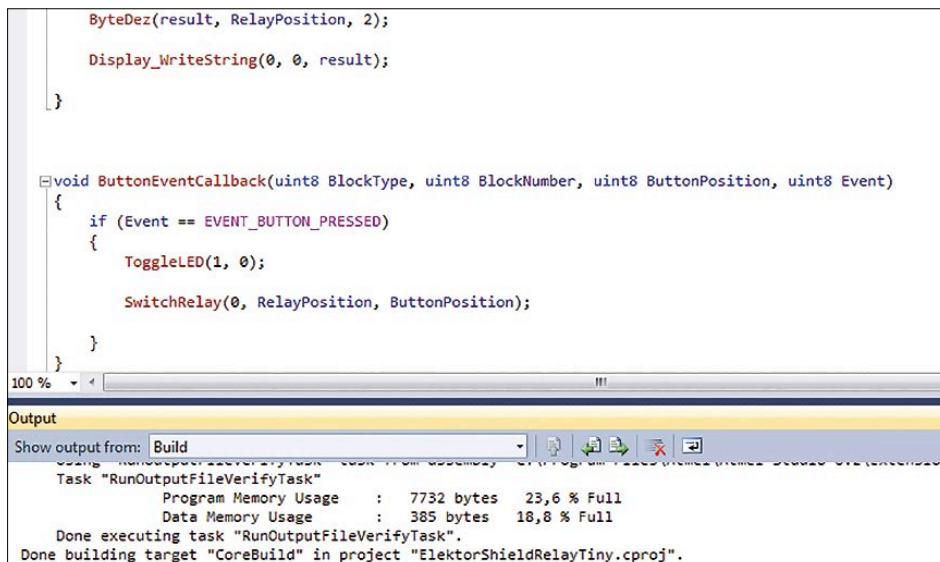


Figure 8. Nous paramétrons à zéro les besoins en mémoire de l'interface SPI logicielle et du temporisateur (*timer*) (en bas à droite). Nous n'avons en effet besoin ni de l'un ni de l'autre.



play_WriteNumber(...) fait, de son côté, appel, dans la bibliothèque *DisplayEFL*, à la fonction *sprintf(...)*. Pour afficher les chiffres 0 à 7, nous n'avons pas besoin d'une fonction aussi puissante. Nous remplaçons donc *Display_WriteNumber(...)* par les lignes que visualise le champ de texte *Loop Code* de la figure 8 et produisons à nouveau le projet depuis EFL Configurator. Le **figure 9** nous montre le résultat. Le projet *ElektorShield RelayTiny* peut être trouvé, lui aussi, sous les paramètres de projet fournis.

Figure 9. En l'absence d'accès à distance, le code de la commande du relais trouve également place dans un ATmega88.

Scripts

Les scripts utilisés sont écrits dans le langage de script *Sheets* développé par l'auteur ; ils sont enregistrés, dans une application locale telle qu'EFL Configurator, dans des fichiers texte. Un script peut avoir plusieurs valeurs d'entrée et une valeur de sortie. Comme les scripts ont été initialement utilisés pour créer dynamiquement un document XML ou (X)HTML, la valeur de sortie a pour nom *XMLResult*. Le code de script lui-même a lui aussi le format XML, toutes les commandes de script sont des éléments XML. Dans le code du script, les commandes peuvent être mélangées à d'autres éléments XML ou (X)HTML, qui lors de l'exécution du script, se voient simplement écrits dans *XMLResult*. Il est donc relativement facile, par son biais, d'écrire des scripts, des générateurs de pages HTML (interfaces utilisateur par. ex.) ou encore des documents XML (Cf. p. ex. le script *Base.txt* d'EFL Configurator).

La commande TEXT ne fait rien de plus qu'écrire le texte pris entre les balises dans *XMLResult* :

```
<DIV>
<TEXT>Hello World!</TEXT>
</DIV>
```

Le XMLResult contient alors :

```
<DIV>
Hello World!
</DIV>
```

Avec la commande

```
<TEXT Press='DoSomething' >Do something!</TEXT>
```

c'est, au lieu d'un simple texte, un bouton HTML portant le texte *Do something!* que l'on crée. Plus tard, lorsque l'on appuie sur le bouton du formulaire HTML, le script *DoSomething* est lu depuis le fichier texte *DoSomething.txt* et exécuté. Ce script donne accès aux commandes HTML de

l'interface utilisateur d'où il a été fait appel au script. Il pourra s'agir de champs de texte par exemple. Nous pouvons aussi sans problème produire ces champs de texte par un script :

```
<ENTER ID='EnterText' >Replace the text here</ENTER>
```

Dans le script *DoSomething*, il est possible d'accéder au contenu actuel de ce champ de texte au moyen de l'expression *@EnterText*. Lors de l'appel du script, ce contenu est copié dans la variable *EnterText* puis transmis au script comme valeur d'entrée.

Nous pouvons également créer ce genre de variables simples dans le script lui-même et y mettre une valeur. Les commandes

```
<SET RName='TargetFolder' >C:\MyFolder\</SET>
<TEXT>@TargetFolder</TEXT>
```

écrivent le texte *C:\MyFolder* dans *XMLResult*.

Le caractère @ donne accès au contenu de variables, mais, au lieu de @ suivi d'un nom de variable, nous pouvons aussi utiliser le résultat d'un calcul dans notre script. Il suffit d'encadrer l'expression arithmétique par @{ et } .

Les commandes

```
<SET RName='TargetFolder' >
@{Add(C:\MyFolder\,MySubFolder)}</SET>
<TEXT>@TargetFolder</TEXT>
```

produisent le texte *C:\MyFolder\MySubFolder*.

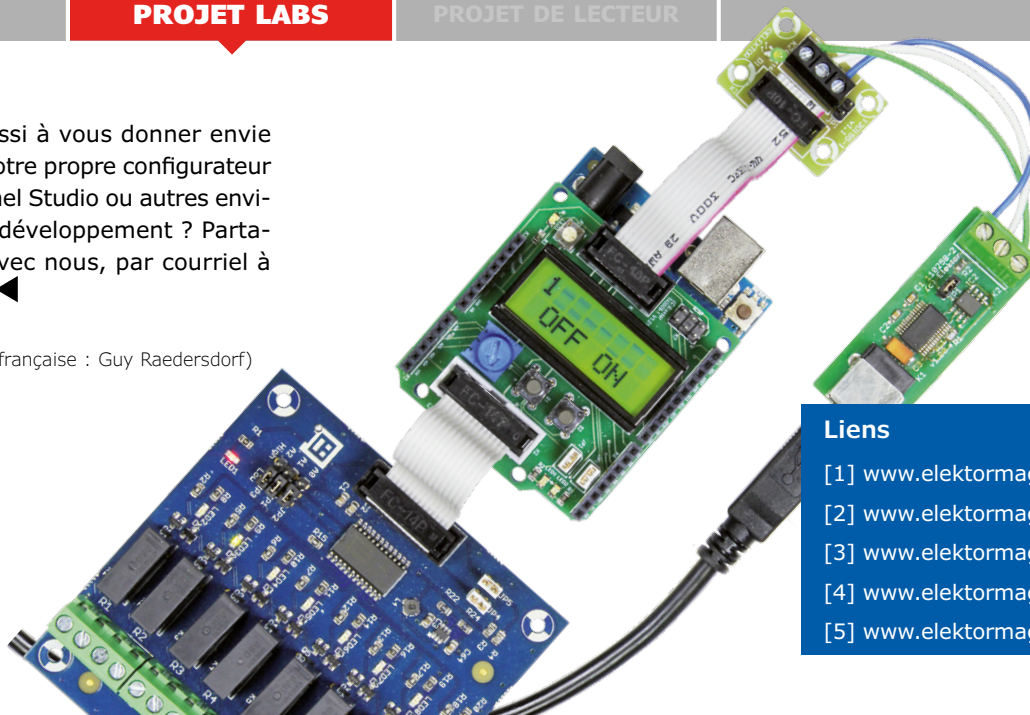
```
<TEXT>@{Add(C:\MyFolder\,MySubFolder)}</TEXT>
```

donne le même résultat, plus directement.

Dans des expressions, tout caractère spécial (virgule, ouverture et fermeture de parenthèse etc.) doit être remplacé

Avons-nous réussi à vous donner envie de développer votre propre configurateur pour projets Atmel Studio ou autres environnements de développement ? Partagez vos idées avec nous, par courriel à la rédaction ! ◀

(140372 – version française : Guy Raedersdorf)



Liens

- [1] www.elektormagazine.fr/140328
- [2] www.elektormagazine.fr/140372
- [3] www.elektormagazine.fr/140009
- [4] www.elektormagazine.fr/130154
- [5] www.elektormagazine.fr/120668

par des constantes prédéfinies qui commencent toujours par @@ : @@Comma, @@Open, @@Close. Les variables d'état de l'interpréteur de script commencent elles aussi par @@ : @@DateToday par exemple contient toujours la date actuelle sous la forme AAAAY-MM-JJ.

Le type de données (complexe) le plus important du langage de script sont des *sheets* (feuilles de tableau) éponymes, composés à leur tour de 1 à 64 colonnes pouvant, à chaque fois, contenir des données de types différents. Chaque ligne du *sheet* contient un ensemble de données reprenant les données de toutes les colonnes. Un *sheet* peut comporter jusqu'à 1 023 lignes. Au cœur d'un ensemble de données, chaque colonne porte un nom pour référencer spécifiquement toute entrée du tableau.

Tous les *sheets* utilisés dans un script (jusqu'à 32) doivent, au début, faire l'objet d'une déclaration de nom, plusieurs entrées successives étant séparées par un point-virgule :

```
<SHEETS>Files;Symbols;</SHEETS>
```

La commande SET permet de remplir non seulement de simples variables comme mentionné plus haut, mais également des *sheets* entiers. La commande

```
<SET Array='True' RName='Folder;File;' RS='Files' >
Common;BoardpinEFL.c;
Common;CommonEFL.h;
Libraries;LEDButtonEFL.h;
</SET>
```

crée dans le *sheet Files* deux colonnes dénommées *Folder* et *File* et remplit le tableau avec les valeurs prises entre les balises. Ici encore, il est permis de faire référence à des variables (@...), variables d'état (@@...) et expressions mathématiques (@{...}).

L'expression @Sheetname.Columnname donne alors accès au contenu d'une colonne donnée du jeu de données (série) actuel d'un *sheet* spécifique. L'expression @Columnname suffit lorsque le nom de la colonne n'est utilisé que dans un seul *sheet*. Une fois le *sheet* rempli, c'est toujours le premier ensemble de données qui est à jour. La commande FOREACHROW permet, par exemple, de parcourir ligne par ligne le *sheet Files* défini plus haut. Au cœur de la boucle, on a alors accès aux données dans chacune des colonnes. Les commandes

```
<FOREACHROW Of='Files' >
<TEXT>@Files.File</TEXT>
<TEXT> ; </TEXT>
</FOREACHROW>
```

donnent le texte suivant dans XMLResult :

```
BoardpinEFL.c ; CommonEFL.h ; LEDButtonEFL.h ;
```

READ et WRITE sont les instructions pour le chargement et l'enregistrement de données dans une base de données. Dans le cas d'une application locale comme EFL Configurator, la base de données est constituée de fichiers texte qui, tout comme les scripts, se trouvent dans le dossier APP, le sous-dossier portant le nom de la base de données (dans notre exemple DATA tout simplement). Chaque fichier texte contient un tableau qui, à l'image d'un *sheet*, est organisé en colonnes et lignes. FILEREAD et FILEWRITE permettent la lecture et la création de fichiers texte (et binaires). FILERC (RC= Rename/Copy) décale et copie ou renomme des fichiers. MAKEFOLDER permet de créer un dossier, la création de tous les dossiers-parents se faisant automatiquement en parallèle. MESSAGE fait apparaître un message à l'écran. Il existe encore bien d'autres commandes.

accordeur de guitare (basse) de scène

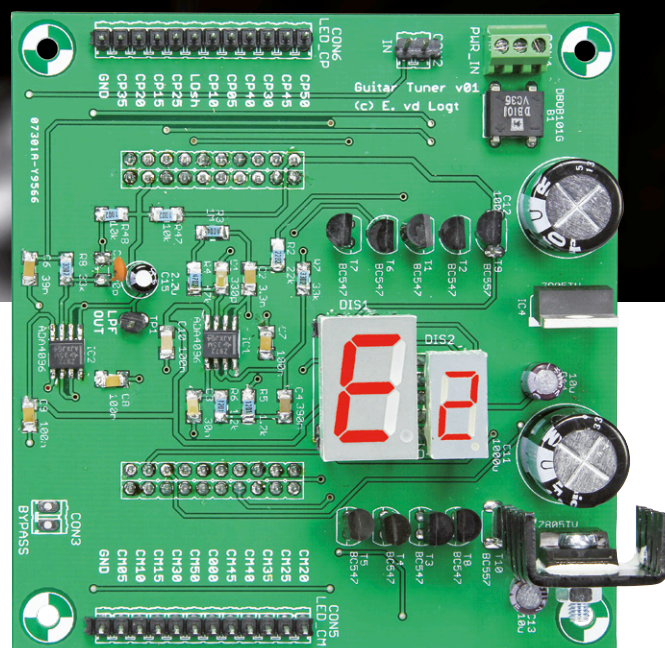
rapide et précis

Emile van de Logt (Pays-Bas)

Pour accorder sa guitare, on peut bien sûr acheter un appareil professionnel de scène, mais le développer et le construire soi-même procure bien plus de satisfaction. Déterminer la fondamentale d'une note de guitare n'est pas aussi facile qu'on le pense tant qu'on n'a pas essayé. Pour ce projet, j'ai utilisé un processeur rapide Cortex ARM M4 et une méthode d'analyse de fréquence par autocorrélation.

Il y a un bout de temps que je cherchais un accordeur rapide et précis pour mes guitares. On peut utiliser pour cela des pédales, mais les bonnes sont chères et pas toujours faciles à lire. Après avoir acheté un modèle professionnel d'accordeur de scène pour guitare, j'ai vite eu l'idée d'en construire un moi-même, pensant que déterminer la fréquence fondamentale d'une note de guitare ne devait pas être bien difficile. Pas plus que d'afficher sur des LED la différence par rapport à la fréquence voulue.

Mon projet a été stimulé par la découverte de la carte d'évaluation TM4C123GXL (**fig. 1**). Elle porte un microcontrôleur ARM Cortex M4 qui, outre de nombreuses entrées/sorties (E/S), dispose de capacités de traitement numérique du signal (DSP) embarquées. Une carte puissante pour... 13 \$ seulement, plus le port, chez TI. L'idéal donc pour saisir un signal et le travailler. Commençons par examiner la carte TM4C123GXL, pour savoir de quoi elle est vraiment capable.



Le circuit imprimé définitif diffère légèrement du prototype représenté ici.

Comme pour tout projet, ce qui importe, c'est de découvrir les choix à faire et les compromis à accepter. S'attacher aussi à discerner les options qui conduiraient à l'échec du système, pour mieux appréhender de quelle manière il doit travailler. C'est ce qui m'a poussé à une pause dans le premier projet, celui de TFR, pour ensuite aboutir à la réalisation finale.

TM4C123G Launchpad Evaluation Kit

Le fabricant de cette carte, *Texas Instruments*, est connu pour la qualité de la documentation et du soutien au matériel qu'il présente, pour favoriser une prise en main rapide. Il y a un Wiki pour cette carte, avec manuel de travail [2], présentations et même des vidéos. L'environnement de développement (IDE) est *Code Composer Studio* (CCS) qui ne laisse rien à désirer. Il s'agit de bien s'accrocher au manuel, de connecter la carte TM4C123GXL au PC par USB et de s'y mettre. On suit les pré-

sentations avec plaisir, les explications sont claires et soutenues par des vidéos et des exemples de projets. On y trouve aussi une interface de programmation d'application (API) du nom de *Tiva* qui permet de commander toutes les E/S. Pour tout étudiant, ingénieur ou amateur, une manière idéale de se lancer dans cette série de microcontrôleurs perfectionnés !

On trouve sur la carte d'évaluation un microcontrôleur ARM Cortex M4 du type TM4C123GH6PM, avec une fréquence d'horloge maximum de 80 MHz, une unité à virgule flottante (FPU) avec capacité de traitement numérique du signal (DSP), 256 Ko de flash, 32 Ko de SRAM, 2 Ko d'EEPROM, 8 UART, 4 interfaces série multiples, 4 modules I²C, CAN, USB, 2 convertisseurs A/N à 12 bits et plein de ports d'E/S ! Ce sont particulièrement les convertisseurs A/N à 12 bits, les nombreuses E/S et les possibilités de DSP qui rendent cette carte tellement adaptée à mon objectif : la réalisation d'un accordeur de guitare de niveau professionnel !

Ébauche : transformation de Fourier rapide (TFR)

Enhardi par les spécifications impressionnantes de la carte, je me suis lancé dans l'implémentation d'une TFR. On en trouve différentes routines prêtes à l'emploi. Et c'est là que j'ai rencontré des difficultés. Calculons ensemble. Le spectre d'un signal de guitare couvre environ 2 kHz. Selon la loi de Shannon [3], il faut une fréquence d'échantillonnage d'au moins 4 kHz. Pour réaliser ensuite une TFR avec une résolution de 0,5 Hz, pas moins de 8 192 points de transformation sont nécessaires ! Côté vitesse, le microcontrôleur peut s'en tirer, mais la mémoire nécessaire rien que pour ceci approche du maximum de 32 Ko de SRAM. Après révision à la baisse, partant d'une fréquence d'échantillonnage de 1 kHz, le spectre tombe à 500 Hz et on peut loger 2 048 points de TFR en mémoire.

Il fallait encore rédiger le reste du programme. Par conséquent il est clairement apparu que le résultat escompté ne serait pas atteint.

Second projet : autocorrélation et détection de hauteur du son

En cherchant une manière adéquate de déterminer les composantes de fréquence dans un signal, j'ai trouvé un article scientifique [4] qui décrivait comment trouver avec précision la fréquence fondamentale d'un signal par une fonction d'autocorrélation. Les auteurs y détaillent leur méthode de détermination de la fondamentale d'un signal – ils la nomment *Yin* – avec les mathématiques nécessaires. Exactement ce qu'il me fallait (cf. encadré) !

Sur la base de cet article, Joren Six [6] a écrit une bibliothèque en C++ que Ashok Fernandez a traduite en une excellente bibliothèque en C [7]. Bel exemple de l'utilité d'un article scientifique pour aboutir à une application concrète. Cette bibliothèque en C (*Yin.h* et *Yin.c*), je l'ai reprise presque intégralement dans le projet et tout a fonctionné du premier coup, autre preuve de la qualité de cette bibliothèque !

Des fréquences et des notes

Pour y comprendre quelque chose dans le projet d'un accordeur de guitare, les notions de note, intervalle et fréquence sont indispensables. Pour donner le ton en musique, on part souvent du **la** (A en notation anglo-saxonne) du diapason, à

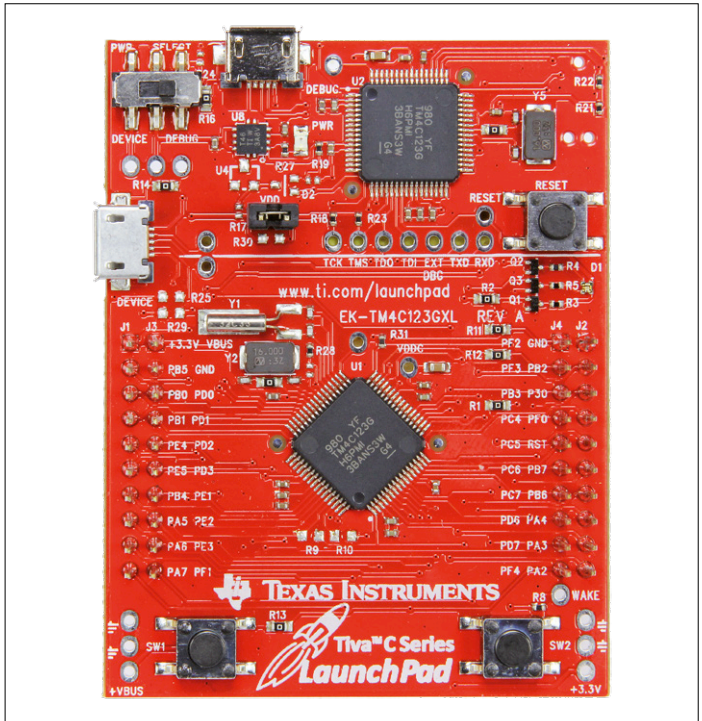


Figure 1. La carte TM4C123GXL de Texas Instruments équipée d'un processeur ARM Cortex M4 (photo TI).

440 Hz. Les autres notes de musique sont **si** (B), **do** (C), **ré** (D), **mi** (E), **fa** (F) et **sol** (G). Entre deux notes voisines, il y a soit un ton soit un demi-ton : entre **si** et **do**, ainsi qu'entre

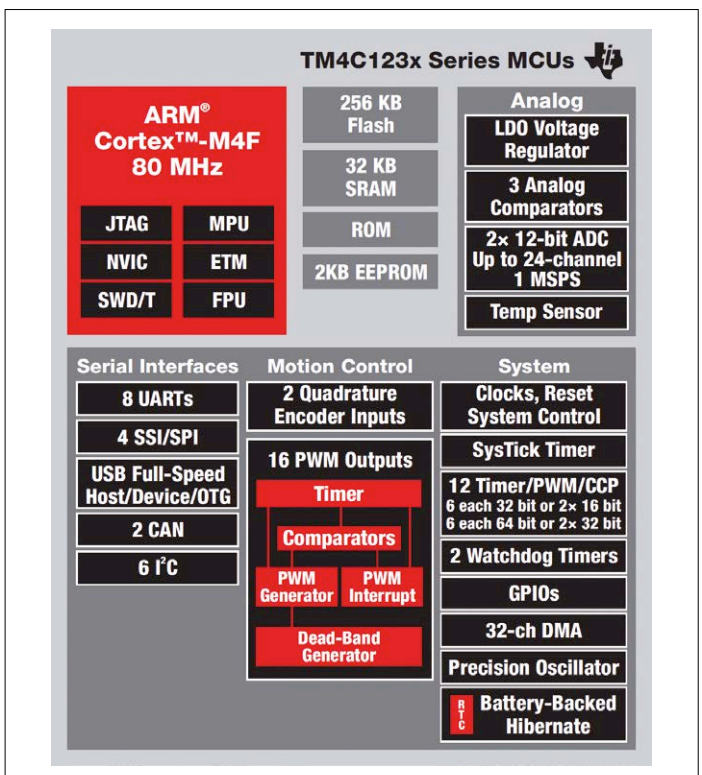


Figure 2. Synoptique des fonctions et des raccords sur la carte TM4C123G (illustration TI).

L'auteur

Emile van de Logt est directeur des cours d'électrotechnique et de technologie des soins de santé à l'université de Rotterdam. Il a étudié l'électrotechnique à l'Université technique d'Eindhoven et les sciences du management à l'Université ouverte.

Passionné d'électronique, il a suivi des ateliers sur la programmation embarquée en C et sur les FPGA-VHDL, il construit des amplificateurs à tubes, des générateurs d'effets pour guitare... Il est aussi brasseur amateur et dispose d'un système complètement automatisé de fabrication de bière dont il a personnellement développé le matériel et le logiciel.



Le processus de Yin : autocorrélation et interpolation parabolique

L'algorithme Yin n'utilise pas la transformation de Fourier, mais se base sur une autre méthode fort répandue dans le traitement numérique du signal, la fonction d'autocorrélation [5]. Elle effectue la somme des multiplications, point par point, d'un signal par une version décalée de ce même signal.

Supposons que le signal d'entrée soit du bruit blanc entre les valeurs -1 et $+1$. Si pareille valeur est multipliée avec une autre valeur, décalée dans le temps, le résultat sera toujours compris entre $+1$ et -1 . Ensuite, quand on additionne toutes ces valeurs, à condition qu'il y en ait suffisamment, la somme va tendre vers 0, parce que finalement elles s'équivalent toutes.

Il en va tout autrement pour un signal qui n'est pas aléatoire, comme celui d'une corde de guitare pincée. Si nous exécutons sur lui la même opération, la valeur sera plus grande que 0 et dépendra du décalage. L'intérêt, c'est que la valeur sera la plus grande quand le décalage retombe précisément sur le signal original, donc une période plus tard.

C'est ce que l'algorithme met à profit. Au pas de la fréquence d'échantillonnage ($1/6\,000$ de s), le signal est décalé et multiplié par lui-même. On regarde ensuite pour quelle grandeur de décalage l'autocorrélation est la plus forte.

C'est la mesure de la fréquence dominante dans le signal !

L'algorithme Yin applique encore un certain nombre de raffinements, comme l'interpolation parabolique, pour atteindre plus de précision. Disons que l'algorithme trouve que l'autocorrélation est la plus haute quand le signal est décalé de 16 échantillons. On arrive à une fréquence de $6\,000/16 = 375$ Hz. La fréquence suivante qui connaît le même cas, c'est $6\,000/15 = 400$ Hz (ces deux fréquences sont toujours un multiple de la période d'échantillonnage). La fréquence recherchée se situe quelque part entre les deux valeurs. L'interpolation parabolique permet alors de trouver la fréquence optimale entre ces deux points, ce qui améliore sensiblement la précision.

mi et **fa**, l'intervalle est d'un demi-ton. Entre les autres notes, il y a chaque fois l'intervalle d'un ton entier.

La suite des 12 demi-tons voisins forme une octave. Les noms correspondants sont do, do#, ré, ré#, mi, fa, fa#, sol, sol#, la, la# et si. Le signe # se prononce dièse (le *hashtag* des Twitteurs) et indique que la note est haussée d'un demi-ton par rapport à la note du même nom sans #.

Après le si revient le do de l'octave supérieure. Les octaves sont numérotées. Le la du diapason à 440 Hz appartient à l'octave n°4, c'est le **la4** (ou A4). Monter d'une octave correspond à un doublement de la fréquence. C'est ainsi que la fréquence du **la5** est de 880 Hz. Celle du la3 de 220 Hz.

Les six cordes d'une guitare sont accordées comme suit : mi2 (E2), la2 (A2), ré3 (D3), sol3 (G3), si3 (B3) et mi4 (E4), avec le bourdon, la corde de mi inférieur, à 82,407 Hz et le mi de la chanterelle à 329,628 Hz. La 12^e frette de la corde de mi du haut, le mi5 (E5), sonne à 659,255 Hz. Si vous pincez la même corde en touchant la 24^e frette (ça existe sur certaines guitares), elle donne le mi6 (E6) qui est à la fréquence de 1 318,51 Hz.

C'est le haut du spectre pour l'accordeur de guitare. À l'autre bout du spectre, la guitare basse à quatre cordes vibre une octave plus bas : mi1 ou E1 (41,30 Hz), la1 ou A1, ré2 ou D2, sol2 ou G2. Une guitare basse à cinq cordes dispose d'une fréquence encore plus basse : le si0 ou B0 à 30,868 Hz. C'est le bas du spectre pour un accordeur de guitare.

La fréquence n'augmente pas linéairement avec la hauteur, c'est-à-dire que la différence en Hz entre deux notes séparées par un $\frac{1}{2}$ ton est différente pour chaque $\frac{1}{2}$ ton. Sans entrer dans les détails du calcul de la fréquence d'une note par rapport au $\frac{1}{2}$ ton voisin, retenir qu'il équivaut à une multiplication par 1,059 (approximativement racine douzième de deux). Exemple : la note la4 (ou A4) est à 440 Hz, le la#4, qui est le $\frac{1}{2}$ ton voisin, est à 466,164 Hz, soit une différence de 26,164 pour cet intervalle d'un $\frac{1}{2}$ ton. En revanche, à l'intervalle d'un $\frac{1}{2}$ ton entre p. ex. la5 et la#5 correspond une différence en Hz de 52,328 comme le montre le calcul des valeurs du **tableau 1**, rédigé directement en code C.

Pour mémoire :

C	D	E	F	G	A	B
do	ré	mi	fa	sol	la	si

Exigences essentielles pour un accordeur de guitare

Avant tout, résumons les prétentions de cet accordeur.

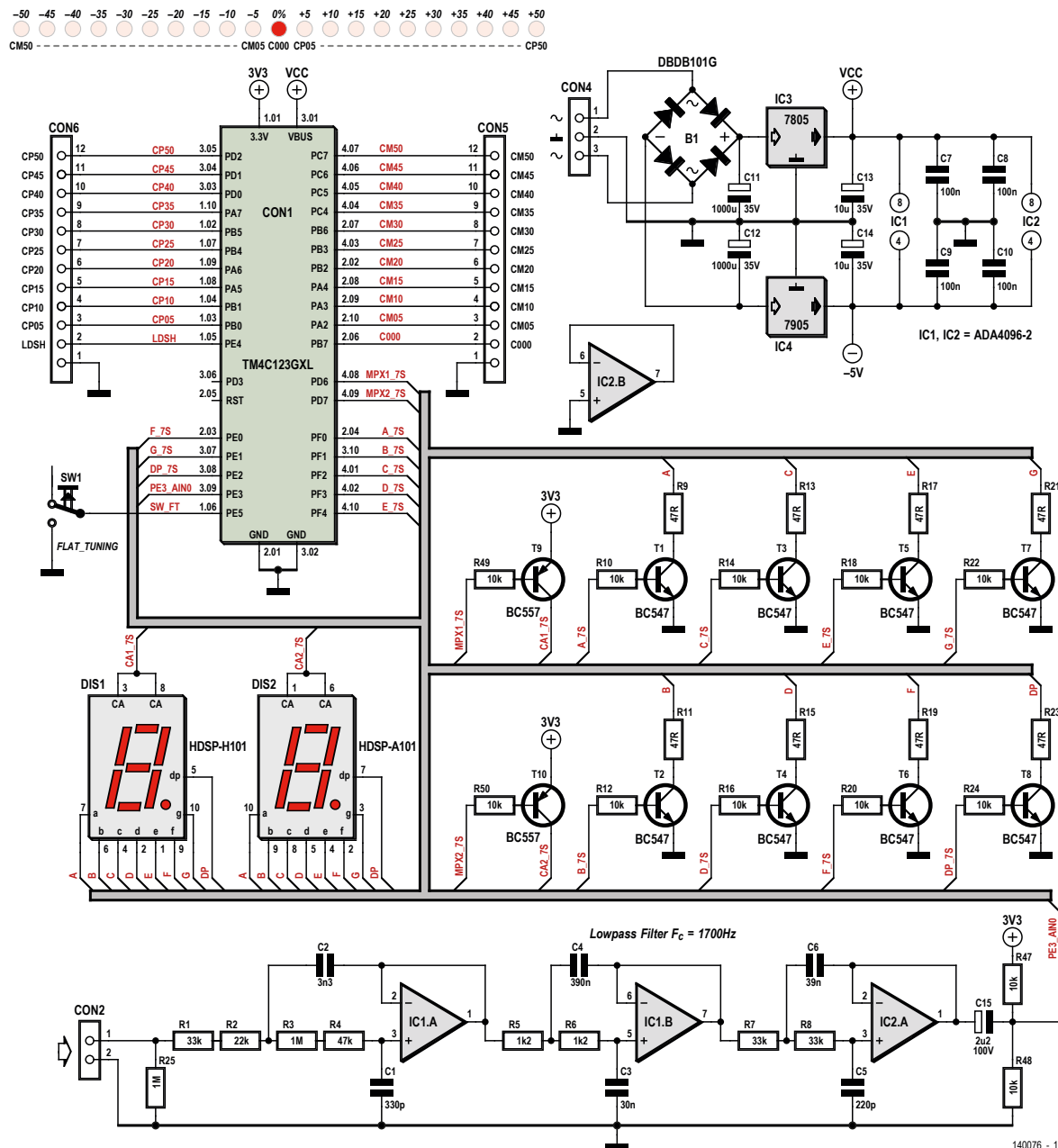
- **Vitesse** : rafraîchissement dix fois par seconde
- **Précision** : marge d'erreur sur la fondamentale : moins de 5 %
- **Affichage** : la note et le numéro d'octave plus le dièse quand il le faut.
- **Écart** : écart relatif de -50% à $+50\%$ au moyen de 21 LED (5 % par LED)
- **Jeu de lumière** : reproduction du parcours lumineux vers le centre jusqu'à l'accord.

Figure 3. Comme la plupart des fonctions sont prises en charge par la carte TM4C123GXL, le schéma complet ne demande que peu de composants.

```
const float note_freq[MAX_NOTES] = { 20.602, 21.827, 23.125, 24.500, 25.957, 27.500, 29.135, 30.868,
32.703, 34.648, 36.708, 38.891, 41.203, 43.654, 46.249, 48.999, 51.913, 55.000, 58.270, 61.735,
65.406, 69.296, 73.416, 77.782, 82.407, 87.307, 92.499, 97.999, 103.826, 110.000, 116.541, 123.471,
130.813, 138.591, 146.832, 155.563, 164.814, 174.614, 184.997, 195.998, 207.652, 220.000, 233.082, 246.942,
261.626, 277.183, 293.665, 311.127, 329.628, 349.228, 369.994, 391.995, 415.305, 440.000, 466.164, 493.883,
523.251, 554.365, 587.330, 622.254, 659.255, 698.456, 739.989, 783.991, 830.609, 880.000, 932.328, 987.767,
1046.502, 1108.731, 1174.659, 1244.508, 1318.510, 1396.913 };
```

```
const char note_name[MAX_NOTES][4] = { "E0", "F0", "F#0", "G0", "G#0", "A0", "A#0", "B0",
"C1", "C#1", "D1", "D#1", "E1", "F1", "F#1", "G1", "G#1", "A1", "A#1", "B1",
"C2", "C#2", "D2", "D#2", "E2", "F2", "F#2", "G2", "G#2", "A2", "A#2", "B2",
"C3", "C#3", "D3", "D#3", "E3", "F3", "F#3", "G3", "G#3", "A3", "A#3", "B3",
"C4", "C#4", "D4", "D#4", "E4", "F4", "F#4", "G4", "G#4", "A4", "A#4", "B4",
"C5", "C#5", "D5", "D#5", "E5", "F5", "F#5", "G5", "G#5", "A5", "A#5", "B5",
"C6", "C#6", "D6", "D#6", "E6", "F6" };
```

Tableau 1.



	+3V3	VBUS	+5V
CP30	PB5	GND	GND
CP05	PB0	PD0	CP40
CP10	PB1	PD1	CP45
LED dièse	PE4	PD2	CP50
Flat-tuning	PE5	PD3	
CP25	PB4	PE1	G_7seg
CP15	PA5	PE2	DP_7seg
CP20	PA6	PE3	AIN0
CP35	PA7	PF0	B_7seg

C_7seg	PF2	GND	
D_7seg	PF3	PB2	CM20
CM25	PB3	PE0	F_7seg
CM35	PC4	PF0	A_7seg
CM40	PC5	RESET	
CM45	PC6	PB7	C000
CM50	PC7	PB6	CM30
mpx1_7seg	PD6	PA4	CM15
mpx2_7seg	PD7	PA3	CM10
E_7seg	PF4	PA2	CM05

Tableau 2.

- **Flat tuning** ou **accord grave** : quand la guitare est accordée délibérément plus bas que son accord normal, d'un ton ou même plus, il faut prévoir la possibilité d'un décalage de 0 à -6 demi-tons vers le grave, de manière à

retrouver un affichage normal.

- **Portée** : détection depuis le bourdon de la basse, si0 sur une cinq cordes, jusqu'à la chanterelle de la guitare ordinaire mi6.

Liste des composants

Résistances :

(toutes CMS1206)

R1, R7, R8 = 33 k Ω

R2 = 22 k Ω

R3, R25 = 1 M Ω

R4 = 47 k Ω

R5, R6 = 1,2 k Ω

R9, R11, R13, R15, R17, R19, R21, R23 = 47 Ω

R10, R12, R14, R16, R18, R20, R22, R24, R47, R48, R49, R50 = 10 k Ω

Condensateurs :

(C1 à C10 CMS1206)

C1 = 330 pF

C2 = 3,3 nF

C3 = 30 nF

C4 = 390 nF/16 V, 5 %

C5 = 220 pF/16 V, 5 %

C6 = 39 nF/50 V, 10 %

C7, C8, C9, C10 = 100 nF

C11, C12 = 1 000 μ F/35 V radial

(au pas de 5 mm, diam. 12,5 mm)

C13, C14 = 10 μ F/35 V radial

(au pas de 2,54 mm, diam. 5 mm)

C15 = 2,2 μ F/100 V radial

Semi-conducteurs :

B1 = pont redresseur DB101

T1 à T8 = BC547

T9, T10 = BC557

IC1, IC2 = double ampli op ADA4096-2

ou LM833 (SOIC-8)

IC3 = 7805

IC4 = 7905

21 LED rouge haut rendement 3 mm

(pour CP05 à CP50, C000, CM05 à CM50)

1 LED rouge haut rendement 5 mm

(pour LDSH)

Divers :

TM4C123GXL carte d'évaluation

DIS1 = chiffre à 7 seg. rouge anode commune, 14,22 mm, 13 mcd@10 mA (p.ex. Avago HDSP-H101)

DIS2 = chiffre à 7 seg. rouge anode commune, 7,62 mm, 10 mcd@10 mA (p.ex. Avago HDSP-A101)

CON2, CON3 = embase à 2 picots

au pas de 2,54 mm

CON4 = embase à 3 picots au pas de 2,54 mm

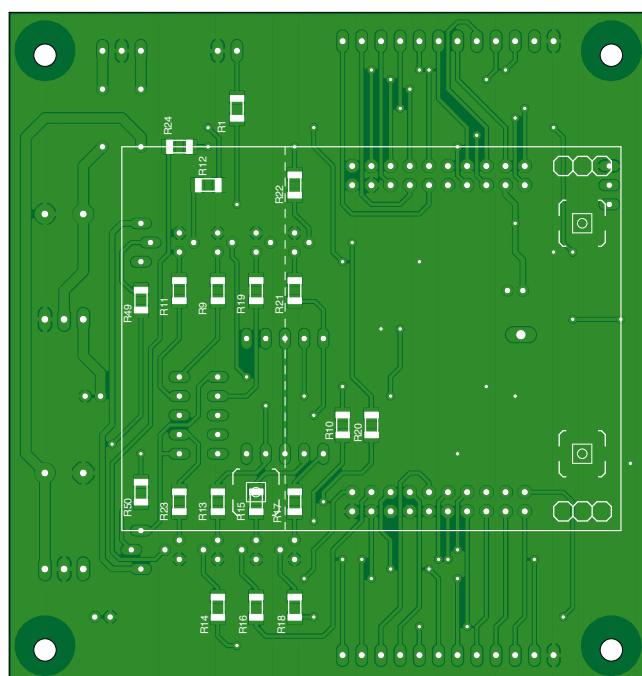
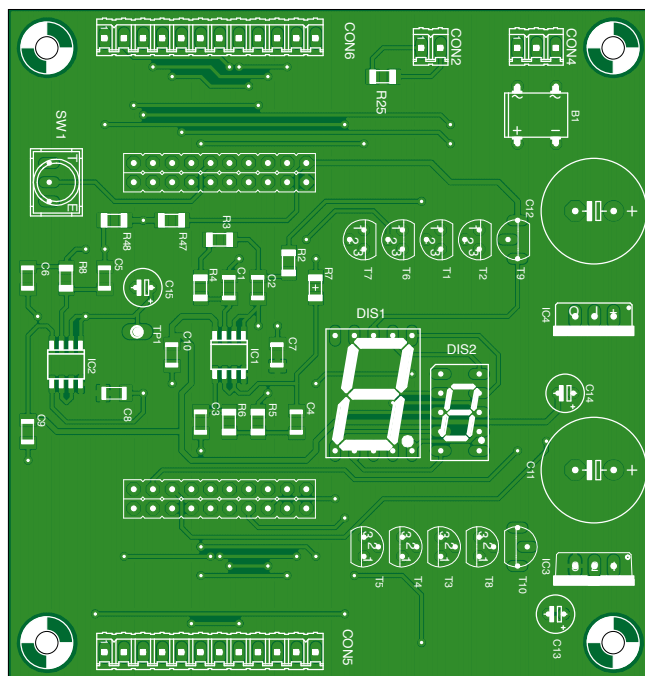
CON5, CON6 = barrette de dominos à 12 voies au pas de 2,54 mm

SW1 = bouton-poussoir normalement ouvert

2 connecteurs 2x10 contacts femelles pour carte d'évaluation au pas de 2,54 mm

3 prises pour jack en façade, diam. 6,3 mm

circuit imprimé 140076, tracé des pistes téléchargeable [1]



Le projet

L'accordeur de guitare résulte d'une étroite connivence entre le matériel de la **figure 3** et son logiciel. C'est pourquoi il serait intéressant que vous disposiez déjà du logiciel pour suivre plus aisément cette description intégrée. Il est disponible au téléchargement sur [8].

Voici de quoi se compose le projet.

- Le signal de guitare est appliqué à un filtre passe-bas du 6^e ordre dont la fréquence de coupure est à 1 700 Hz. Il a été développé avec l'outil *WEBENCH Filter Designer* de TI. Le fabricant prescrit son propre amplificateur opérationnel, ADA4096, mais un modèle standard comme le LM833 fait aussi bien l'affaire.
- L'entrée AIN0, entrée 0 du convertisseur A/N du microcontrôleur, lit la sortie du filtre.
- La fréquence d'horloge de ce CA/N est de 125 kHz et le matériel réalise une moyenne sur 16 échantillons consécutifs, ce qui prend 128 μ s. Le code de l'opération est dans la routine `init_timer0_for_adc()`.
- Il faut à l'algorithme Yin au moins deux périodes complètes du signal de guitare pour arriver à déterminer la fréquence fondamentale. Avec la note la plus basse (si 0 à 30,868 Hz) cela prend 65 ms. Comme l'actualisation a lieu toutes les 100 ms, l'algorithme Yin peut détecter des fréquences jusqu'à 20 Hz (mi 0).
- La fréquence d'échantillonnage FS est fixée à 6 000 Hz. En 100 ms, on lit donc 600 échantillons. C'est la routine `ADC0_SS3_IntHandler()` qui s'en occupe. Elle les écrit dans un tampon en anneau. Il a ceci de particulier qu'un processus peut y écrire tandis qu'un autre peut y lire.
- Après lecture de 600 échantillons, on donne par la variable `sampled_new` le signal à la boucle `while` de `main()` de lire les 600 échantillons du tampon en anneau pour les écrire dans le tampon d'entrée de l'algorithme Yin que l'on appelle immédiatement. Et c'est l'algorithme Yin qui détermine finalement la fréquence fondamentale, la variable `pitch` du signal de guitare.
- On appelle alors la routine `find_nearest_note()` pour déterminer la note la plus proche dans le tableau, ce sera la variable `nnote`. Elle déterminera aussi l'écart relatif en pourcentage, la variable `cents`.
- On passe ces deux variables à une routine d'interruption, `Timer1IntHandler()`, qui tourne à 200 Hz. Elle inscrit le nom de la note et le numéro d'octave sur les afficheurs à 7 segments commandés en multiplex par la routine `display_note()`. Il ne reste qu'à montrer l'écart relatif par rapport à la note affichée au moyen des 21 LED. La raison d'être de la routine `display_cents()`. Plus l'écart est grand, plus il y a de LED à allumer. Elle se sert d'un diagramme d'état qui règle aussi le balayage lumineux quand la corde approche de l'accord.

Figure 4. Le circuit imprimé pour l'accordeur de guitare. La carte TM4C123GXL se place à l'arrière et se branche sur les connecteurs correspondants.

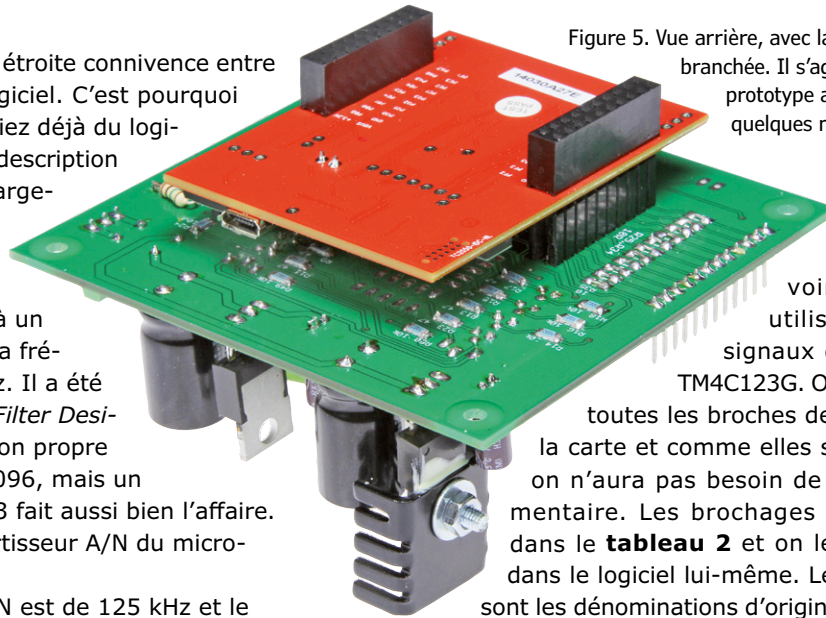


Figure 5. Vue arrière, avec la carte TM4C123GXL branchée. Il s'agit d'un précédent prototype auquel on a apporté quelques modifications.

Il nous reste à voir comment sont utilisés les différents signaux d'E/S de la carte TM4C123G. On emploie presque toutes les broches des connecteurs de la carte et comme elles sont nombreuses, on n'aura pas besoin de matériel supplémentaire. Les brochages sont rassemblés dans le **tableau 2** et on les retrouve aussi dans le logiciel lui-même. Les noms en rouge sont les dénominations d'origine des broches, les autres décrivent chaque fois la fonction exercée dans l'accordeur.

- CP05 à CP50 : connexions pour les dix LED d'écart positif ;
- CM05 à CM50 : connexions pour les dix LED d'écart négatif ;
- C000 : connexion pour la LED qui indique quand la corde sonne juste ;
- *Flat tuning* : entrée pour l'interrupteur de déclenchement du changement d'accord. À la première pression, s'affiche le réglage actuel, ensuite l'accord est abaissé par pas successifs jusqu'à -6. Puis il est remis à 0. Exemple : l'accord abaissé est à -1, sur la guitare, la corde de mi est accordée sur ré#. En pinçant cette corde, c'est pourtant mi qui est affiché. Si la fonction *flat tuning* était sur 0, ce serait ré# qui serait indiqué ;
- LED-sharp (dièse) : connexion pour la LED qui indique la détection d'une note diésée ;
- `mpx1_7seg` et `mpx2_7seg` : pour la commande de multiplexage des afficheurs à 7 segments ;
- Les signaux `xx_7seg` contiennent les niveaux de commande des afficheurs à 7 segments.

La construction

Installer le logiciel peut s'effectuer de deux manières.

Étendue : télécharger et installer *Code Composer Studio* (CCS) version 6 et l'API *Tivaware* [10]. Avec le manuel *Getting Started* spécifique, pas de souci, tout est expliqué en détail dans les premiers chapitres. Ensuite, télécharger le projet accordeur de guitare du site Elektor [8] et le placer dans l'espace de travail de CCS. Au passage, on peut adapter le logiciel et en faire un nouveau fichier .bin. Brancher ensuite la carte TM4C123GXL au PC par câble USB et cliquer sur le bouton *Debug*. Le nouveau micrologiciel est immédiatement chargé dans la carte.

Minimale : pour qui veut seulement programmer la carte TM4C123GXL par le micrologiciel accordeur de guitare sans installer l'environnement de développement CCS complet.



Figure 6. Photo de mon prototype que j'emmène régulièrement sur scène.

Installer alors le programme *LM Flash Programmer* [11] et utiliser le fichier *Guitar_Tuner_Yin.bin* du répertoire Debug du projet pour le charger dans la carte TM4C123GXL. La procédure adéquate est aussi décrite dans le fichier *Workbook*.

Il est possible de commander la carte TM4C123GXL directement chez le fabricant, c'est rapide et meilleur marché, mais aussi chez un distributeur comme *Farnell*. La carte nécessite une petite adaptation pour fonctionner avec le micrologiciel accordeur de guitare. Pour des raisons de compatibilité avec d'autres cartes TI, quelques broches de port sont reliées ensemble : PB7 avec PD1 par la résistance R10 de 0 Ω et PB6 avec PD0 par la résistance R9 de 0 Ω . Ces deux résistances se trouvent juste sous le microcontrôleur et devant le texte du logo du fabricant, marquées R9 et R10. Enlevez-les toutes les deux prudemment avec une panne bien chaude.

Le circuit imprimé porteur du circuit du projet est réalisé sous la forme d'un « *shield* » à placer sur la carte TM4C123GXL. Il contient le filtre passe-bas, l'alimentation et les connexions pour toutes les LED et les afficheurs à 7 segments. Plusieurs CMS sont utilisés, mais ils sont aisément soudables avec un fer à pointe fine. C'est encore plus facile à l'aide d'une station à air chaud. Commencez par construire l'alimentation sur le *shield*. Le transformateur, qui n'est pas représenté dans le schéma, peut être du type 2×6 V ou 2×9 V et ne doit fournir que peu de puissance.

Après la construction, vérifiez d'abord si vous avez du +5 V et du -5 V bien stables. Débranchez tout et soudez le restant des composants sur le circuit imprimé. Les afficheurs à 7 segments peuvent se souder directement ou se placer dans des connecteurs surélevés. Montez les embases femelles à 2×10 contacts sur la face inférieure du *shield* pour pouvoir y brancher la carte du TM4C123GXL.

Les 21 LED sont reliées aux connecteurs CON5 et CON6. Celle du milieu vient sur la broche 2 de CON5 (C000), avec à sa droite les LED CP05 à CP50 (écarts de +5 % à +50 %) et à sa gauche les LED CM05 à CM50 (écarts de -5 % à -50 %). Il y en a encore une autre, celle qui indique le dièse, elle vient à la broche 2 de CON6. Chaque LED est alimentée entre la sortie précitée des connecteurs et la broche de masse (1) du même connecteur.

Vérifiez, après sa construction, le bon fonctionnement du filtre passe-bas en lui envoyant un signal sinusoïdal de 500 mV_{pp} par CON2. Vous pouvez mesurer sur la broche de test TP1 (LPF_OUT) le signal de sortie du filtre. Il doit laisser passer les fréquences jusqu'à 1 700 Hz et atténuer les fréquences supérieures.

S'il travaille bien, vous pouvez raccorder la carte à TM4C123GXL : la carte sous le *shield*, le connecteur USB en direction de C11/C12/IC3/IC4. Vérifiez alors que le point décimal sur le petit afficheur clignote au rythme de 1 Hz. Appuyez sur le poussoir *Flat Tuning* et assurez-vous que les nombres -0, -1 jusqu'à -6 s'affichent successivement. Remettez-le sur 0, branchez une guitare et pincez une corde. La note et le numéro d'octave doivent s'afficher et les LED s'allumer. Réglez la guitare pour faire s'allumer celle du milieu (C000). Mon prototype détecte même le ré5 (D5) sur la 22^e frette de la chanterelle ! À l'autre extrémité, sur la corde de mi grave (E2), aucun souci. À défaut de guitare basse, j'ai fait des essais avec un générateur de fonction. Le prototype détectait parfaitement les fréquences jusqu'à 20 Hz. On peut encore mettre le projet dans un boîtier, j'ai utilisé à cette fin un châssis Hammond (515-0950) et je lui ai ajouté une face avant. C'est aussi pour cette raison que les LED ne sont pas implantées sur le circuit imprimé, mais reliées par CON5 et CON6.

Gardez bien l'accord !

(140076 – version française : Robert Grignard)

Liens

- [1] www.ti.com/tool/EK-TM4C123GXL
- [2] http://processors.wiki.ti.com/index.php/Getting_Started_with_the_TIVA%E2%84%A2_C-Series_TM4C123G_LaunchPad
- [3] http://fr.wikipedia.org/wiki/Théorème_d'échantillonnage_de_Nyquist-Shannon
- [4] YIN, a fundamental frequency estimator for speech and music, 2002, Alain de Cheveigne & Hideki Kawahara
- [5] <http://fr.wikipedia.org/wiki/Autocorrélation>
- [6] <https://github.com/JorenSix/Pidato>
- [7] <https://github.com/ashokfernandez/Yin-Pitch-Tracking>
- [8] www.elektormagazine.fr/140076
- [9] www.ti.com/lstds/ti/analog/webench/webench-filters.page
- [10] www.ti.com/tool/ccstudio
- [11] www.ti.com/tool/Imflashprogrammer

MAXQ à la rescousse

un multimètre sauvé du dépôt



Il y a quelque temps, nous avons trouvé un multimètre Philips endommagé pour une bouchée de pain sur *Marktplaats* (site néerlandais de vente d'objets d'occasion). L'écran à cristaux liquides (LCD) n'était plus très lisible, mais nous étions persuadés que nous pourrions facilement réparer l'appareil. C'était présomptueux : avec la meilleure volonté du monde, rien ne pouvait sauver l'écran, et un remplacement par d'autres composants s'imposait.

Fons Janssen, Maxim Integrated (Pays-Bas)

Ce qu'il y a de bien avec les appareils de mesure des années 80-90, c'est qu'ils sont construits avec des composants

standard, et que la documentation technique est souvent disponible sur l'internet. Après étude du manuel de notre multimètre PM2535 [1], nous avons estimé qu'il devait être possible de remplacer l'écran par un autre modèle.

Commande de l'écran d'origine

L'écran LCD est commandé par un circuit intégré Philips PCF8576. Celui-ci reçoit les données du processeur central via une interface I²C. Notre idée était donc d'intercepter et de décoder ces données,

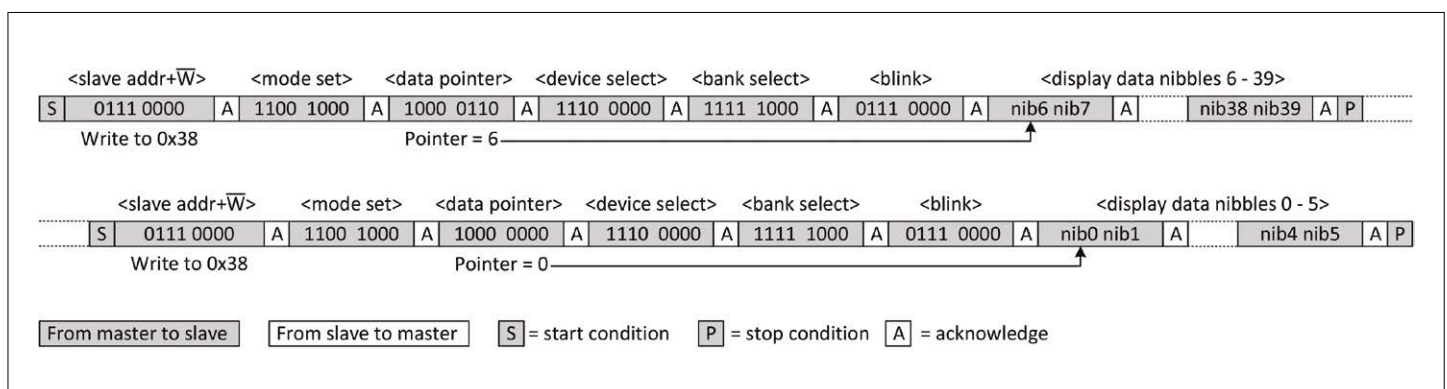


Figure 1. Transmission de données à l'écran par I²C.

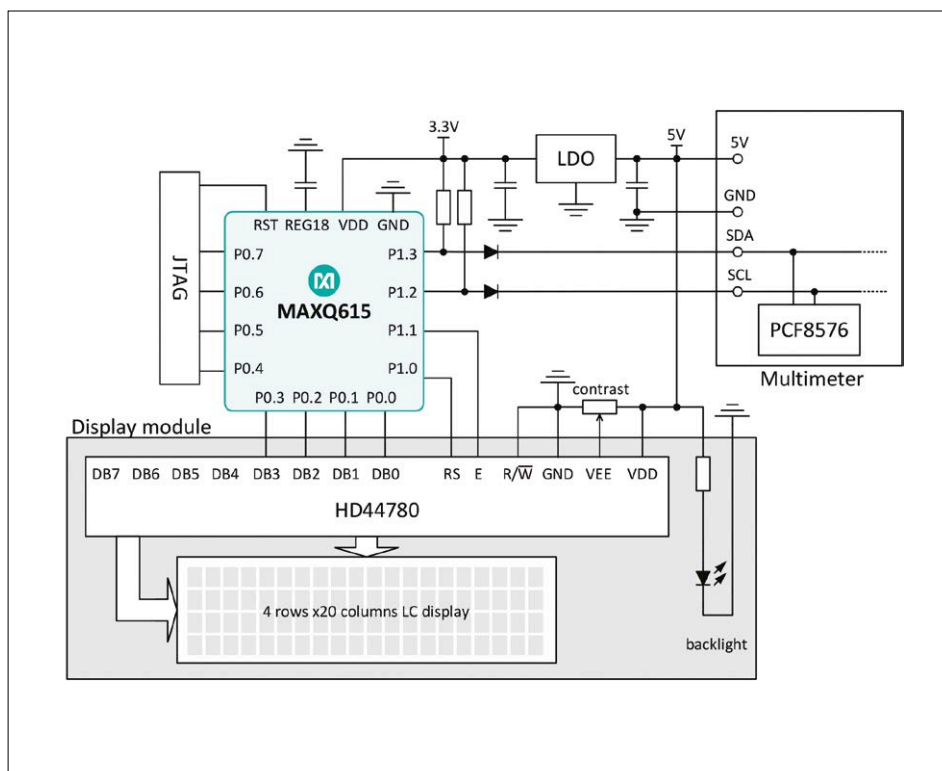


Figure 2. Le circuit de commande du nouvel écran comprend un circuit de commande HD44780, un écran de 4x20 caractères, et un rétroéclairage. Pour la liaison, quatre lignes : alimentation + 5 V, masse, et transmission de données SDA et SCL.

prises au PCF8756, les décode et les transmet au nouvel écran. Nous avons opté pour un MAXQ615 de chez Maxim Integrated ; en sus de son interface I²C, il possède juste assez de lignes d'E/S pour commander l'écran en mode 4 bits. Le circuit (**figure 2**) est élaboré autour d'une platine d'évaluation du MAXQ615, dont une partie est constituée d'un circuit imprimé à trous, ce qui permet le montage des composants additionnels.

Le MAXQ615 fonctionne sous 3,3 V et ne peut donc être connecté directement aux circuits logiques 5 V du multimètre. Le contrôleur ne fait que recevoir des données ; en munissant les lignes SDA et SCL d'une résistance de rappel et d'une diode d'isolation, le problème est résolu. Lorsque le maître I²C met les lignes au niveau bas, les diodes conduisent et les entrées sont aussi au niveau bas ; lorsque les lignes sont au niveau haut, les diodes sont bloquées et la tension aux entrées est limitée à 3,3 V. Lorsque le MAXQ615 met les lignes au niveau bas (par exemple pour un accusé de réception), les diodes se bloquent également ; ceci empêche toute transmission de données, qui perturberait la communication entre le processeur central et le PCF8576.

Logiciel

L'interface I²C du MAXQ615 est matérielle. L'écriture des valeurs de quelques registres par le programme suffit à régler les fonctions I²C et à configurer le contrôleur en esclave.

Par défaut (routine principale), le contrôleur attend les données I²C. Dès qu'il reconnaît sa propre adresse (identique à celle du PCF8576), il sait que le premier octet sera une commande. Toutes les commandes suivantes sont ignorées, sauf celle contenant le pointeur de données (*data pointer*), utilisé pour déterminer l'emplacement mémoire des données. Les données reçues après le dernier octet de commande sont entreposées dans une zone de 20 octets (deux quartets par octet) ; ce tableau est la copie bit à bit de la mémoire de l'écran dans le PCF8576. La valeur du pointeur correspondant est la moitié du pointeur de données au niveau quartet. Rappelons que la transmission s'effectue en deux temps.

Après réception de l'ensemble des données, leur traitement peut commencer.

► l'écran LCD d'origine est remplacé par un modèle de 4x20 caractères, commandé par un HD44780

puis de les envoyer vers un autre écran. Le PCF8756 peut commander jusqu'à 160 segments. La documentation technique du multimètre décrit avec précision la relation entre les segments de l'écran et la mémoire du circuit intégré, organisée en 40 lignes de 4 bits (quartets, *nibbles* en anglais). Chaque bit détermine la visibilité d'un segment.

La fiche de caractéristiques du PCF8756 [2] explique en outre comment remplir la mémoire via l'I²C : on envoie une ou plusieurs commandes, suivies des données. Avec l'aide d'un analyseur logique, on voit clairement que la transmission s'effectue en deux temps : d'abord les quartets 6 à 39, et ensuite les quartets 0 à 5. La séquence se répète en continu, pour assurer le rafraîchissement de l'écran. Le flux de données est représenté en **figure 1**.

Les commandes *<mode set>*, *<device select>*, *<bank select>* et *<blink>* permettent la configuration du pilote de l'écran ; leurs valeurs restent constantes, nous n'avons donc pas à nous en préoccuper. La commande *<data pointer>* est en revanche importante : elle indique à quel emplacement de la mémoire les données sont inscrites. Les données (de deux quartets chacune) sont envoyées à la suite des commandes, et automatiquement inscrites au bon emplacement dans la mémoire.

Nouvel écran

Pour remplacer l'écran d'origine, nous avons choisi un modèle de 4 lignes de 20 caractères commandé par un HD44780 ; c'est tout juste suffisant pour les données à afficher.

Un microcontrôleur configuré en mode esclave I²C intercepte les données trans-

Le contenu de l'écran est stocké dans une chaîne de 80 caractères ; celle-ci est parcourue par le programme, qui détermine ce qui doit apparaître à l'écran et à quel emplacement. Pour des segments uniques c'est relativement simple, ça l'est moins pour les caractères composés de 7 ou 16 segments. Les bits d'un caractère de 7 segments sont regroupés dans un octet, ceux d'un caractère de 16 segments dans deux octets ; une table de correspondance permet d'établir la relation octet(s)-caractère. Dès que la dernière correspondance est établie, la chaîne est transmise en une fois à l'écran. La **figure 3** montre une représentation schématique du traitement des données. Le programme pour le MAXQ615 est disponible sur le site d'Elektor [3], quoique nous ne nous attendions pas à ce que beaucoup de lecteurs tombent sur un multimètre identique avec le même défaut...

Montage

Pour pouvoir monter le nouvel écran, nous avons scié l'ancien écran et l'avons rangé dans un espace vide du boîtier, à côté de la platine du MAXQ615. Le nouvel écran peut alors prendre place dans l'espace libéré et être collé au boîtier.

Deux photos montrent le multimètre, avant et après. L'affichage est non seulement à nouveau bien visible, mais aussi bien plus lisible grâce au rétroéclairage.

À suivre ?

Le multimètre est opérationnel, mais il y a sans doute encore moyen de l'améliorer : en changeant la référence de tension, par exemple. Rendez-vous à plus tard, peut-être... ◀

(150766 – version française : Jean-Louis Mehern)

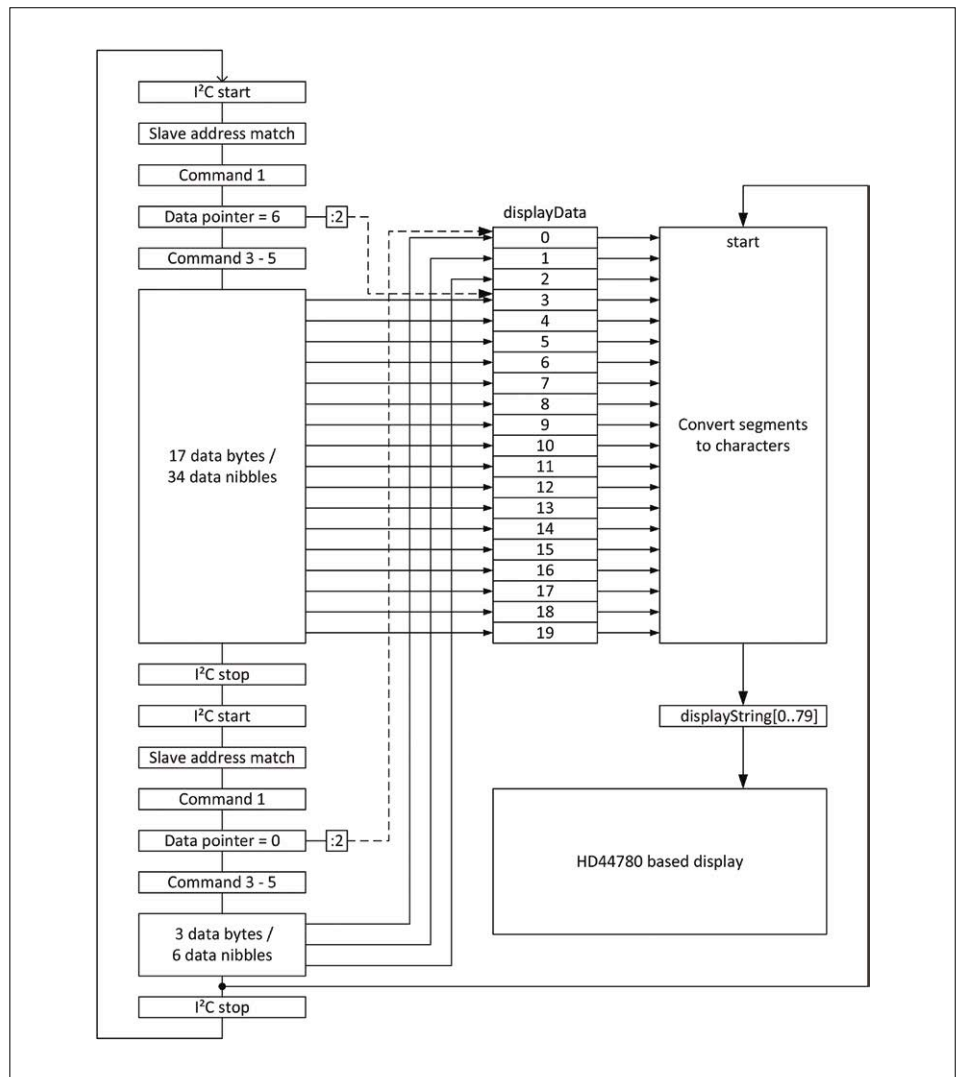


Figure 3. Représentation schématique du traitement des données.



Figure 4. L'écran endommagé : rien à récupérer.

Liens

- [1] www.download-service-manuals.com/download.php?file=Philips-6930.pdf
- [2] www.nxp.com/documents/data_sheet/PCF8576.pdf
- [3] www.elektormagazine.fr/150766



bienvenue dans votre e-choppe

Elektor recommande



Mooshimeter, instrument de mesure et enregistreur de données sans fil

Le multimètre est un instrument incontournable de la boîte à outils des électroniciens. Depuis quelques années, les multimètres avec des fonctions « sans fil » se multiplient. Le multimètre Mooshimeter appartient à cette nouvelle génération, mais il peut également remplacer un multimètre standard. Logé dans un coffret relativement petit, en polycarbonate, il se connecte à un ordiphone Android ou iOS via Bluetooth 4.0. Une appli permet d'afficher les

valeurs mesurées ou de régler plusieurs paramètres. Grâce aux deux convertisseurs A/N à 24 bits, il est possible d'effectuer des mesures sur deux canaux à la fois. En outre ce multimètre enregistre les mesures sur carte SD. Ne vous privez pas de cet excellent instrument aux caractéristiques uniques !

Harry Baggen
(labo d'Elektor)

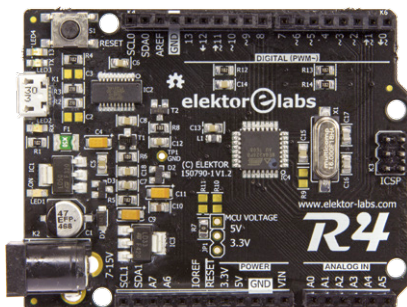
www.elektor.fr/mooshimeter



Vos favoris :

1. Elektor Uno R4

www.elektor.fr/elektor-uno-r4



2. Horloge Nixie à six chiffres

www.elektor.fr/horloge-nixie

3. Raspberry Pi 3 (modèle B)

www.elektor.fr/raspberry-pi-3-b

4. Mooshimeter

www.elektor.fr/mooshimeter

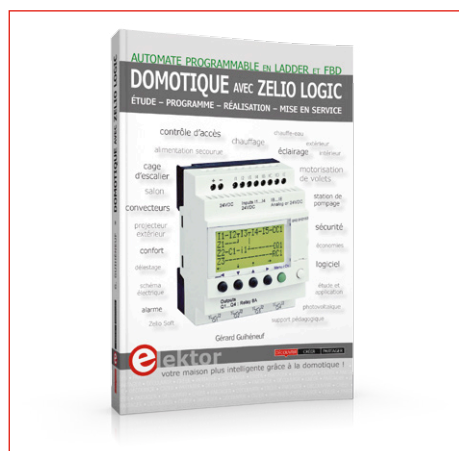
5. Livre Tuto Arduino Uno

www.elektor.fr/tuto-arduino-uno

6. DVD Elektor 1990-1999

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

Domotique avec Zelio Logic



Le module programmable compact Zelio Logic avec douze entrées/sorties est une solution technique bon marché et facile à mettre en œuvre, dans un logement, pour automatiser l'éclairage, piloter le chauffage, optimiser la consommation d'électricité ou encore pour se protéger contre les intrusions. Ce livre montre comment exploiter Zelio Logic avec son outil de programmation graphique en LADDER ou FBD (logiciel pour PC Zelio Soft 2).



Prix (membres) : 41,85 €

www.elektor.fr/domo-zelio

Tuto Arduino Uno 45 projets électroniques originaux



Alarme silencieuse, détecteur de présence, luxmètre, commande de moteur, accès depuis l'internet, commande sans fil sont quelques-uns des projets Arduino Uno brillants et passionnants de ce livre, vérifiés et testés par l'auteur. Ces 45 projets sont faciles à réaliser et bien décrits (schémas, photos). Les explications et la description des composants utilisés facilitent l'adaptation des montages à ses besoins.



Prix (membres) : 31,32 €

www.elektor.fr/tuto-arduino-uno

Atlas DCA75 Pro

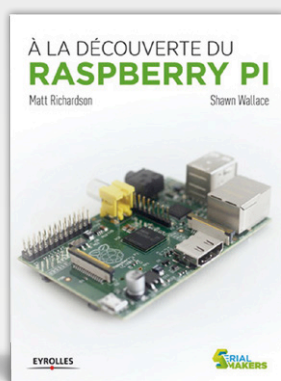


Cet instrument d'analyse des semi-conducteurs dernier cri affiche les caractéristiques des composants sur son écran graphique ou sur PC. Reliez les sondes à n'importe quel composant pour voir sur l'écran LCD les résultats de test (brochage, caractéristiques). Cet instrument reconnaît les transistors IGBT, MOSFET, JFET entre autres, les régulateurs de tension, les diodes, les LED, les triacs (jusqu'à 10 mA), les thyristors (jusqu'à 10 mA), etc.



Prix (membres) : 112,46 €

www.elektor.fr/atlas-dca75-pro

**NOUVEAU !**notre sélection de
**livres
EYROLLES**
pour petits et grandsLIBRAIRIE **EYROLLES****NOUVEAU !****Elektor
Uno R4**Encore mieux
qu'Arduino Uno R3**e-choppe
Elektor en
ligne**Votre panier d'achats
pour l'électronique**Uniquement pour les membres Elektor :
5% de remise et frais de port offerts****offre valable
jusqu'au 11 août 2016****www.elektor.fr/eyrolles****Raspberry Pi 3 Starter Kit
(version Deluxe)**

Le *Raspberry Pi 3 Starter Kit* inclut tout ce dont vous avez besoin pour démarrer le nano-ordinateur le plus populaire du monde ! Le kit comprend une carte Raspberry Pi 3 (modèle B), un luxueux boîtier en ABS pour la carte, une alimentation micro-USB pour Raspberry Pi 3 (5,1 V, 2,5 A), un câble HDMI haute vitesse (1 m), un câble Ethernet cat. 5e (2 m) et une carte micro-SD Transcend (16 Go, classe 10) avec adaptateur (avec NOOBS pré-installé).

**Prix (membres) : 80,96 €****www.elektor.fr/rpi-starter-kit-deluxe****Pro Tech Toolkit**

Un kit d'outils aux talents multiples pour tout réparer.

Indispensable aux professionnels et amateurs

Tout pour réparer les appareils électroniques
tournevis, pinces...

Kit totalement repensé

**Prix (membres) : 58,46 €****www.elektor.fr/all-new-pro-tech-toolkit****Analyseur logique ScanaQuad SQ100**

ScanaQuad (SQ) est une série d'analyseurs logiques à quatre canaux et de générateurs de signaux numériques. Ils vous seront d'une grande utilité lorsque vous travaillerez sur des protocoles série comme UART, SPI, I²C, 1-Wire, USB, I²S, CAN, LIN, RS232, RS485, etc. Avec les analyseurs logiques ScanaQuad, vous pouvez capturer des signaux et les analyser, ainsi que produire vos propres signaux de test !

**Prix (membres) : 128,66 €****www.elektor.fr/logic-analyzer-scanaquad**

Rejoignez la communauté Elektor

Devenez membre GOLD maintenant !



formule GOLD

- ✓ 10x magazine imprimé
- ✓ 10x magazine numérique
- ✓ l'accès à l'archive d'Elektor (des milliers d'articles)
- ✓ l'accès à plus de 750 projets d'Elektorlabs
- ✓ le DVD annuel d'Elektor
- ✓ 10% de remise dans l'e-shoppe
- ✓ des offres exclusives

formule GREEN


- ✓ 10x magazine numérique
- ✓ l'accès à l'archive d'Elektor (des milliers d'articles)
- ✓ l'accès à plus de 750 projets d'Elektorlabs
- ✓ 10% de remise dans l'e-shoppe
- ✓ des offres exclusives

Également disponible
abonnement « zéro papier » GREEN !

www.elektor.fr/membres



layar
INTERACTIVE PRINT

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

SCANNEZ CETTE PAGE ET REGARDEZ LA VIDÉO



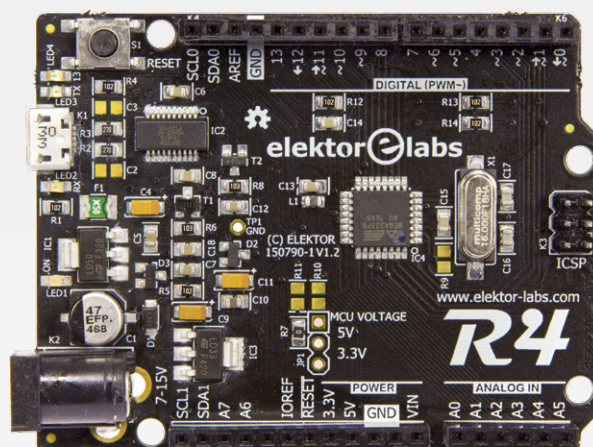


Nouveau microcontrôleur, plus d'E/S et plus de possibilités !

Quand un fabricant propose la version B d'un produit donné, les différences entre nouveau et ancien modèles sont généralement peu significatives. Ce n'est pas le cas pour l'ATmega328PB d'Atmel au cœur de l'Arduino Uno R3. La version B de ce μ C a tellement de nouvelles fonctions que l'abandon de la version R3 est justifié : **passer à l'Elektor Uno R4**.

Points forts :

- ATmega328PB à 16 MHz
- 2x UART
- 2x I²C
- 2x SPI
- 9 sorties à MLI
- 8 entrées analogiques
- 24 broches d'E/S à usage général
- Régulateurs de tension embarqués 5 V et 3,3 V
- Paquet (installation + bibliothèque) compatible Arduino
- Code source ouvert, matériel libre



NOUVEAU !

notre sélection de
livres
EYROLLES
pour petits et grands

NOUVEAU !

**Elektor
Uno R4**

Encore mieux
qu'Arduino Uno R3

e-choppe
Elektor en
ligne

Votre panier d'achats
pour l'électronique



Prix (membres) : 26,95 €
www.elektor.fr/elektor-uno-r4

DVD Elektor 1990-1999



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



Prix (membres) : 80,10 €

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

DVD Elektor 2015



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



Prix (membres) : 24,75 €

www.elektor.fr/dvd-2015

Horloge Nixie à six chiffres



Voici une nouvelle version de notre horloge Nixie qui affiche l'heure à la seconde près ! En plus elle tient compte automatiquement du changement d'heure (été/hiver). L'horloge est maintenant pilotée par un PIC18F4420. Le kit contient tous les composants (circuits imprimés nus, tubes Nixie, contrôleurs programmés, autres composants électroniques et pile de 9 V). Il faut un fer à souder pour assembler ce kit.



Prix (membres) : 116,96 €

www.elektor.fr/horloge-nixie

bienvenue dans la section **PARTAGER**


Thijs Beckers (Elektor)

avionique

Avez-vous déjà réfléchi aux équipements électroniques et électriques à bord d'un avion ? Sans électronique, on ne pourrait pas piloter les avions actuels ; décollage et atterrissage par mauvais temps seraient très dangereux, voire impossibles. Divers systèmes sont installés afin de transporter les passagers en toute sécurité, d'autres sont là pour le confort. Pourquoi doit-on dès lors arrêter ses propres appareils électroniques dans un avion de ligne ? Il y a bien le « mode Avion », mais quand même... En dehors du fait que l'on se trouve dans une cage de Faraday et que plus de 150 téléphones mobiles chercheraient désespérément une antenne-relais à puissance maximale, mes recherches dans ce domaine n'ont rien donné. Si vous en savez plus à ce sujet, n'hésitez pas à nous contacter ! Une autre bizarrerie : pourquoi atténue-t-on l'éclairage au décollage et à l'atterrissage ? Là, j'ai une explication plausible : l'éclairage dans la cabine est fonction de la luminosité extérieure, pour que la différence soit la plus faible possible. En cas de besoin, nos yeux mettraient moins de temps à s'adapter, ce qui nous ferait gagner de précieuses secondes. C'est aussi la raison pour laquelle les stores des hublots doivent rester ouverts.

Le réseau de bord est aussi un sujet intéressant, notamment le choix des alimentations : tension alternative (à 400 Hz) de 115 V en triphasé, et de 28 V en continu. À 400 Hz, un transformateur spécifié pour une charge donnée sera plus petit et plus léger qu'à 50 ou 60 Hz, ce qui est un facteur très important dans un avion. Une règle de base nous apprend que pour chaque kilo épargné, on en économise cinq autres sur la carcasse de l'appareil, et sur le carburant qui serait nécessaire pour déplacer ce kilo. Une fréquence élevée a cependant ses inconvénients, entre autres la chute de tension due à l'inductance des câbles : à 400 Hz elle est jusqu'à sept fois supérieure à ce qu'elle serait à 60 Hz. Et pour finir, saviez-vous que certains avions peuvent être commandés à partir d'un *iPad*, qui se substituera au pilote (pas complètement...) en cas de nécessité ? Le pilote automatique est géré par l'*iPad*, et l'avion peut encore se poser en toute sécurité. ◀

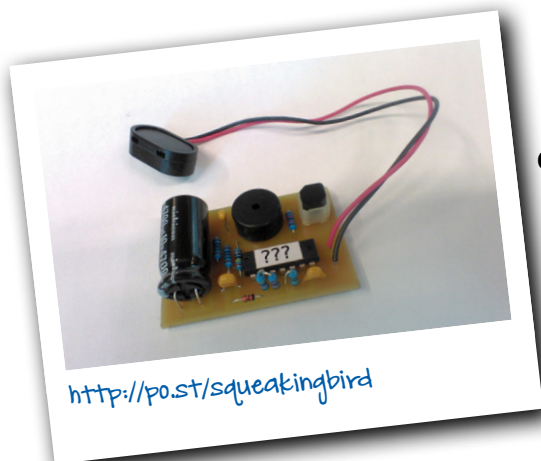
Crédit photo : www.popsi.com

(160022 – version française : Jean-Louis Mehren)



bruits de labo...

Effet du printemps, marque de sympathie, remerciement ou improbable conjonction des astres, toujours est-il que nous avons reçu ces dernières semaines plusieurs cadeaux. Pour notre plus grand plaisir bien sûr : au labo d'Elektor, nous adorons les surprises !



Cafouillis et gazouillis

Au siège d'Elektor, nous ne nous étonnons plus de recevoir des colis à l'adresse incomplète et/ou accompagnés d'un texte amphigourique écrit sur un bout de papier. Quand il y en a un. Malgré cela nous parvenons généralement à relier le contenu de ces colis à tel ou tel projet d'Elektor.Labs. Pas avec ce cadeau expédié anonymement d'Allemagne, une petite carte qui produit une suite de sons aigus. Tente-t-elle de nous dire quelque chose ? Manifestez-vous si vous connaissez l'auteur de ce gadget, pour que nous puissions lui gazouiller un merci personnalisé !

Times Square à la maison

L'autre jour, nous avons reçu de Suisse une grande boîte. Des chocolats ? Non, une clé USB, un adaptateur et un circuit joliment conçu, à savoir une carte dotée d'un PIC et de 180 LED. Visiblement un bandeau défilant, que nous avons bien sûr allumé. Une des phrases du message disait : « Bonjour Elektor ! Merci pour tout ce que vous m'avez appris en électronique ! Ce projet est *open source*, et ne demande qu'à être publié et amélioré. » Ce que nous avons fait sur Elektor.Labs. A-f-i-n q-u-e t-o-u-t l-e m-o-n-d-e e-n p-r-o-f-i-t-e !



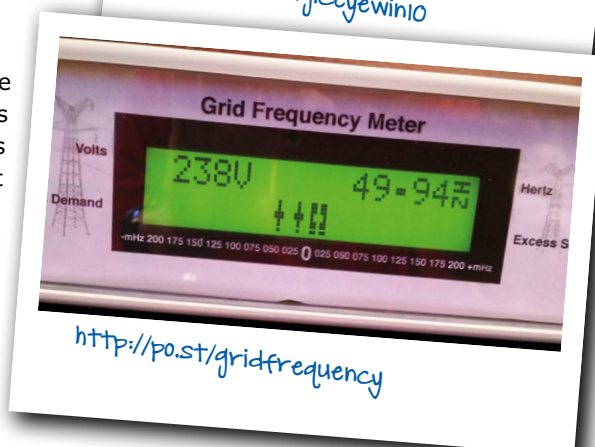
Œil magique pour Windows 10

En janvier 2010, Elektor publiait le projet d'un « œil magique », ou tube indicateur cathodique, qui affichait le taux d'utilisation de l'unité centrale d'un PC. Plus esthétique qu'utile, le tube lumineux n'avait d'autre but que d'égayer les ternes boîtiers des ordinateurs avec un petit côté rétro. Le pilote fonctionnait parfaitement à l'époque, mais il avait été écrit pour Windows XP et est devenu incompatible avec les nouvelles versions de l'OS. Que faire ? Le mettre à jour ? Appliquer une rustine ? L'auteur a choisi de court-circuiter Windows et d'utiliser une puce séparée pour la communication ATtiny-PC.

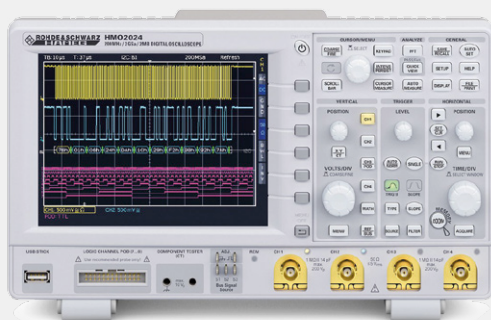


Construisez votre fréquencemètre de réseau

La qualité de la tension du réseau préoccupe plus de personnes qu'on ne le pense. Pourquoi ? Je n'en ai aucune idée. Parce que les consommateurs veulent s'assurer qu'ils paient bien pour la fréquence nominale de leurs contrats ? Ou vérifier qu'elle est aussi stable qu'on le prétend ? Pourtant elle fluctue. Il semble facile de la mesurer puisqu'elle est relativement basse, mais il faut pour cela une référence de temps stable et précise, ainsi qu'un point de déclenchement stable. Familier du sujet, l'auteur de ce projet a récemment amélioré le micrologiciel de son fréquencemètre de réseau. ◀



(160023 – version française : Hervé Moreau)



un oscilloscope, mais lequel ? (1)

Elektor 05/2016, p.14 (150769)

Cet article doit aider les électroniciens à choisir l'oscilloscope adapté à leurs besoins. Il comporte malheureusement une petite erreur que nous signalons à Rohde & Schwarz. Dans le paragraphe « Interfaces de télécommande », nous avons écrit qu'Hameg utilise toujours et encore l'interface RS232, qu'il n'est plus nécessaire de vous présenter. Toutefois, cela fait des années que les oscilloscopes Hameg sont également dotés d'une interface USB. Nous vous prions de nous pardonner cette erreur. Dans le labo d'Elektor, nous utilisons différents oscilloscopes Hameg équipés d'une telle interface double (RS232 + USB). L'utilisateur peut ainsi choisir l'interface appropriée.

projet 2.0

corrections, mises à jour et courrier des lecteurs



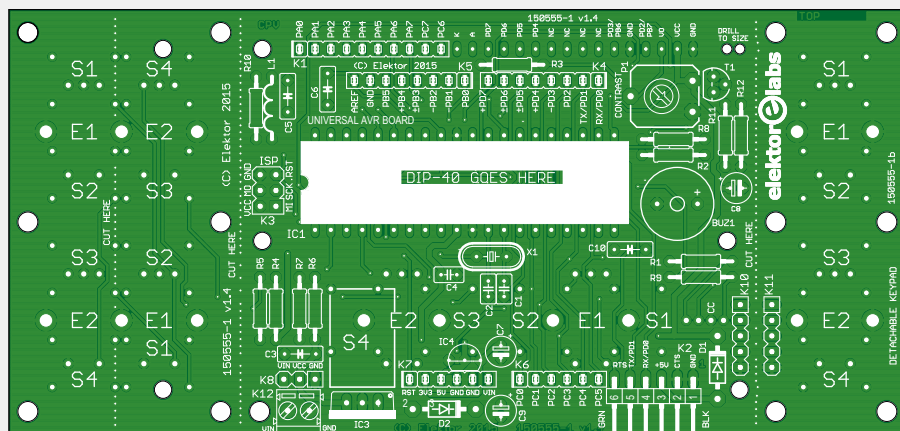
Platino, le retour

Elektor 03/2016, p.54 (150555)

Le projet comporte deux erreurs.

Pour le schéma et la liste des composants, T1 est un BC547C, mais le circuit imprimé est prévu pour un 2N3902. Si les deux types de transistors sont utilisables, ils ne sont malheureusement pas compatibles broche à broche, l'émetteur et le collecteur sont inversés. Il faut souder un BC547 ou un transistor avec le même brochage côté LCD de la carte, ou côté composants après une rotation de 180°. Si l'on prévoit que le rétroéclairage de l'écran LCD soit/puisse être allumé en permanence, il suffit de court-circuiter les bornes de collecteur et d'émetteur à l'aide d'un pont et de ne pas implanter T1.

Le tableau 1 de l'article (*Cavaliers*) comporte une erreur sur la ligne JP3 (rétroéclairage LCD), « PC7 ». Il faut lire : *Port PB5 ou PC5*. Les cavaliers sur le schéma sont corrects (JP3 = PB5, PC5).



climatiseur réglable

Elektor 06/2015, p.54 (140345)

Depuis la préparation de cet article, son auteur, Goswin Visscher, a acquis de l'expérience avec le climatiseur réglable. Au cours des deux dernières années, il a vu le capteur de température et d'humidité tomber en panne trois fois sans raison apparente. Par la suite, il a modifié le circuit et opté pour un autre capteur. Il s'agit d'un DHT22 (Adafruit, de spécifications similaires) ; son prix n'est d'ailleurs qu'un quart de celui du capteur précédent. L'utilisation du nouveau capteur requiert quelques ajustements sur la carte. Le DHT22 est un capteur « One-Wire » et donc sans interface I²C. R13 doit être remplacée par une résistance de 10 kΩ. On relie ensuite la broche DATA du capteur à la broche SDA du connecteur (reliée elle à RC4 du microcontrôleur). La broche SCL ne sert pas.

Bien sûr, le logiciel doit être modifié. Les données fournies par le capteur n'ont plus à être converties, d'où de grosses économies de code. Nous avons donc pu intégrer une fonction RS232 dans le même programme.

Dans le menu, il est possible d'activer l'option RS232 pour la sortie 1 avec la commande « O1=1[CR][LF] » et de la désactiver avec « O1=0[CR][LF] » (idem pour la sortie 2). La chaîne de caractères qui contient les informations d'état est transmise une fois par seconde, quelle que soit l'option du menu sélectionnée.

La nouvelle version du logiciel est disponible au téléchargement sur la page du projet (www.elektormagazine.fr/140345).



un oscilloscope, mais lequel ? (2)

Elektor 05/2016, p.14 (150769)

Je possède un DSO UNI-T doté d'un afficheur riquiqui de 320×240, c'est une source de frustrations. Heureusement que ledit oscilloscope met les échantillons à disposition via USB, ce qui permet de visualiser les données sur un écran d'ordinateur. J'ai écrit un programme qui permet de visualiser plus de 10 000 échantillons (à condition d'avoir un nombre de moniteurs suffisant).

En tout cas, il est possible de faire défiler cette page. La résolution verticale est de 9 bits. Le programme est écrit en Python et fonctionne sur un PC sous Windows ou Linux, et sur des processeurs ARM (ODROID, Raspberry Pi 2). C'est avec plaisir que je mets à disposition le logiciel et la documentation (en anglais) et je serais très heureux que vous transmettiez ma proposition à vos lecteurs. On trouvera une capture d'écran en pièce jointe.

Hermann Hamann

Bonjour M Hamann,

Il n'y a pas à dire, vous avez indéniablement amélioré les capacités d'affichage du DSO UNI-T !

Un grand merci pour votre offre : nous nous faisons un plaisir de proposer au téléchargement votre logiciel sur la page de l'article concerné (www.elektormagazine.fr/150769).

Ralf Schmiedel, service des lecteurs Elektor



Balaise, l'ampli Hi-Fi

Elektor 05/2016, p.74 (150738)

Cher M. Scherer,

Un grand merci pour le souvenir nostalgique concernant votre GIGANT 1986 ! Vous savez probablement que le nom de « GIGANT » est une tradition dans le monde de l'audio. Dès 1961, il y avait, dans le livret DIY (*Do It Yourself*) de la société RIM de Munich, le kit d'un amplificateur à tubes pour table de mixage, qui portait ce nom. Il fournissait, à partir de deux tubes de sortie EL 34, une puissance de 30 W, que l'on pouvait abaisser à 12 W en commutant la tension d'alimentation. Quelques années plus tard, la puissance réduite fut rehaussée à 15 W – mais sans modification du circuit. Vers la fin des années 1960, les tubes de puissance passèrent d'une tension de polarisation automatique à une tension fixe, les transformateurs grossirent et l'amplificateur fut baptisé GIGANT S, avec 45 W cette fois. Ici, plus d'option de réduction de la puissance. Je possède l'un de ces amplis.

En 1969, la société Dynacord proposa son GIGANT, un ampli de scène, qui tirait une puissance de 160 W de quatre EL 34 – à l'époque un véritable géant parmi les amplis à tubes, même si sur cet appareil les tubes se trouvaient exclusivement dans la section amplificateur, les étages petits signaux étaient en général « transistorisés ». Je possède quatre exemplaires de ce GIGANT. Puis arriva leur Baby de 500 W par canal. Je ne l'ai jamais construit ; cependant sa forte ressemblance avec le Crescendo de décembre 1982 me donna l'idée de rehausser la tension de service de la section des *drivers* de mon propre Crescendo non modifié par ailleurs, par la mise en place sur chaque transformateur torique de deux nouveaux enroulements de 5 V. La section des *drivers* disposait (à la différence de la tension non régulée du GIGANT) d'une tension de ± 77 V, la tension de service des FET de puissance Hitachi était, avec leurs ± 70 V, identique à celle du GIGANT. Et, surprise, mon Crescendo fournit, suite à cette modification, sans distorsion, plus de 300 W bien stables par canal !

Merci donc, avec « un certain retard », des idées que m'a fournies votre GIGANT !

Uwe Menrath

Cher M. Menrath,

Un grand merci pour ces infos ô combien intéressantes. Je les découvre vu que je n'ai commencé à manier le fer à souder que dans les années 70 et qu'à l'époque je trouvais beaucoup trop compliqué, tout composant comportant plus de trois broches - un tube par exemple ;-).

Je suis heureux que l'astuce du rehaussement de la tension de l'étage des *drivers* fonctionne si bien dans le cas du Crescendo aussi !

Thomas Scherer



téléphone Telefunken pour voiture de 1965

« *Schätzchen**, je serai en retard pour le dîner »



Reproduit avec l'aimable autorisation du Museum für Kommunikation de Francfort.

Gerd Kowalewski (Allemagne)

Aujourd'hui, pour appeler à l'autre bout du monde, on prend un téléphone portable, ça ne coûte (presque) plus rien et ça n'a plus rien à voir avec les communications vocales des téléphones d'antan. Nous communiquons par le signal vocal numérisé, par le texte et les illustrations de notre courrier électronique et même par vidéo ; tout y passe, fichiers de données, musique, photos, vidéos en direct...

La diffusion de flux à haut débit par l'internet nous permet d'avoir une présence virtuelle partout dans le monde.

Les performances d'un téléphone portable standard relié à un réseau cellulaire de type D, E ou maintenant LTE, sont bien supérieures à celles des ordinateurs de bureau d'il n'y a pas si longtemps, pour une fraction seulement de leur consommation énergétique. S'il convient de saluer le progrès technologique en électronique et surtout en

microélectronique, il ne faut pas oublier cependant les systèmes d'infrastructure des réseaux mobiles radio, sans lesquels tout cela serait impossible.

Trouvaille dans un garage

Ma curiosité et mon envie d'explorer l'origine et l'historique des réseaux mobiles dans mon pays, l'Allemagne, sont nées le jour où j'ai trouvé une radio mobile personnelle VHF (PMR) — aujourd'hui très rare, à cause des faibles quantités produites — parmi les biens d'une personne décédée. Soigneusement emballées dans un carton poussiéreux, deux caisses de métal, banales et a priori peu engageantes, arboraient magnifiquement le nom d'un fabricant allemand de renom : *Telefunken* (fig. 1). Le tré-

EST^D 2004

www.elektor.tv



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



Figure 1. Trouvées dans un garage : deux boîtiers métalliques de taille identique à monter dans un coffre de voiture avec la marque Telefunken sur leur face avant.



Figure 2. Les boîtiers jumeaux de l'installation de téléphonie mobile avec les câbles correspondants.

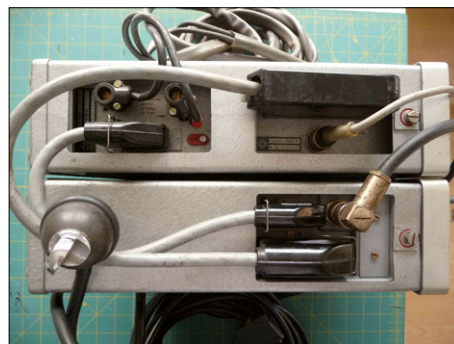


Figure 3. (3a) Connexions câblées entre la radio S/E 160E15 öbl B et le boîtier auxiliaire. (3b) Les connecteurs du boîtier radio.

sort que je venais de découvrir était un émetteur/récepteur pour voiture de type S/E 160E15 öbl B en excellent état, avec alimentation, unité de sélection d'appels et tous les câbles (fig. 2). Le tout (fig. 3a, 3 b et 4a, 4 b) date des débuts de la téléphonie mobile en Allemagne : le réseau A des années 60.

Il s'agit d'un émetteur/récepteur (*Sender/Empfänger*), autour de 160 MHz, avec 15 voies de communication sélectionnables sur les 17 du réseau A1 à cette époque. Deux porteuses VHF, décalées de 4,5 MHz permettaient de créer un canal radio duplex avec l'aide de stations de base, surélevées et distantes d'environ 50 km.

Cet appareil a été conçu à Ulm par *Telefunken*, spécialiste de la radio, comme successeur significativement plus compact à la S/E160E11. La production et l'assemblage, à la main, en petites quantités ont commencé en 1965 (comme l'indique le numéro d'autorisation). Le sigle öbl du réseau A [1] correspond à *öffentlicher beweglicher Landfunkdienst*, le service public de radio terrestre, réservé alors à une poignée de clients (fortunés). La limite de capacité de son implantation finale a été atteinte en 1971 avec le nombre (difficilement imaginable aujourd'hui) de seulement 10.784 abonnés.

Ce qui explique la rareté actuelle des téléphones mobiles pour voiture de ce type. Une telle installation de téléphonie mobile pour voiture comprenant deux boîtiers (près de 15 kg en tout) à placer dans le coffre et une unité de commande pour le tableau de bord coûtait une fortune ! Il utilisait déjà des transistors, même pour le convertisseur HT *push-pull* à commuta-

tion (fig. 5), mais pas pour le multiplieur, le pilote et l'ampli de puissance 10 W de l'émetteur qui utilisaient, entre autres, deux doubles tétrodes de type 6360 (QQE03/12) (fig. 6). Il y avait quelques cavaliers à manipuler pour adapter l'appareil au système électrique du véhicule (fig. 7). Et non, tout le monde n'utilisait pas du 12 V avec négatif sur le châssis dans les années 60.

Le prix : 6550 DM sans l'installation ! [6]. Soit 23.500 € d'aujourd'hui. Sans parler des frais de fonctionnement ni de la licence. L'abonnement mensuel coûtait à lui seul l'équivalent de 250 €. Plus tard, les autorités ont jugulé l'afflux de nouveaux intéressés en passant l'abonnement à quelque 1000 € actuels.

La clientèle comprenait non seulement politiciens et hommes d'affaires aisés, mais également des équipes d'entretien routier ou forestier, le personnel technique au sol des aéroports et les commandants des navires sur le réseau fluvial allemand. La *Deutsche Bundesbahn* (chemins de fer) utilisait ce réseau pour les communications radiophoniques à bord des trains.

Sur le réseau A1, un canal suffisait pour 20 à 25 abonnés, qui pouvaient donc écouter les communications les uns des autres. Le réseau A2 ajoutait 19 canaux duplex (38 fréquences porteuses). Le canal 39 était le canal d'appel. Une tonalité à 2280 Hz (canal libre et utilisable), signifiant aux utilisateurs qu'ils étaient à portée d'une antenne et reliés au réseau de téléphonie fixe, était transmise par les stations de base, tout comme la tonalité à 1750 Hz qui indiquait qu'un canal était occupé. Ces tonalités étaient transmises sur les canaux de communication. Sans la



Figure 4. Les connecteurs de l'alimentation STV E 6/12 (4a) et de l'unité de sélection d'appels SRS-1098/12 (4 b). *Ruf-Nr* = identifiant de l'unité de sélection d'appels.



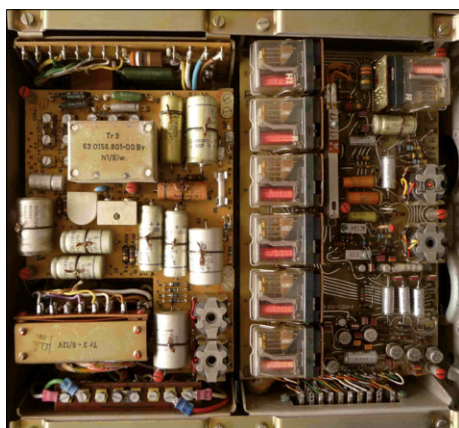


Figure 5. À l'intérieur du convertisseur DC-DC push-pull STV E 6/12 et de l'unité de sélection d'appels SRS-1098/12.

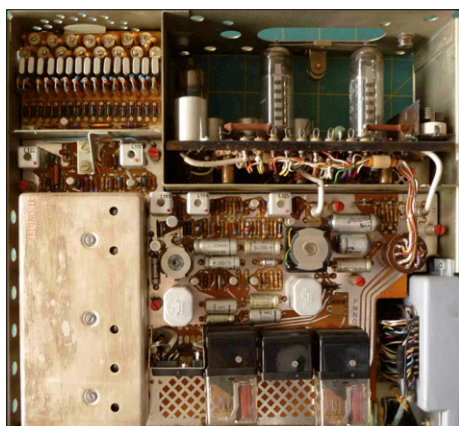


Figure 6. Vue interne de l'émetteur/récepteur. (Quasiment) que des transistors !

tonalité 'canal libre' de la station de base, il était impossible d'attribuer un canal voix.

Sélection des appels

Le codage du numéro d'appel à sept chiffres [2] [3] d'un abonné était relativement complexe. Les deux premiers étaient la clé (*Subscriber Key*) de la zone au sein du réseau radio, pour le réseau A1 (21 à

25, 31) ou A2 (61 à 65, 32). La tolérance des fréquences de la signalisation à fréquences audio utilisée sur le canal d'appel était de seulement 1 %. La *Group Key* [3] était codée dans le troisième chiffre du numéro mobile. Ces chiffres figuraient dans le système de sélection « 4 parmi 30 » suivant un tableau de clés spéciales à partir duquel on choisissait quatre fréquences individuelles dans une liste de 30 fréquences audio qui correspondaient à un nombre à quatre chiffres (chiffres 4 à 7 du numéro du radiotéléphone).

Cette technique rendait possible l'utilisation d'un « signal de sonnerie continue » sur le canal d'appel qui permettait des appels codés semi-automatiques pour un maximum de 27.405 abonnés, en supposant que la position du client était connue à l'avance au moins approximativement. Pas d'itinérance, de triangulation, ni de transfert intercellulaire à l'époque.

Tout ceci demandait des circuits accordés à bande très étroite dans le récepteur qui devait détecter simultanément quatre fréquences audio sur le canal de service (canal de supervision pour les experts). Les fréquences étaient de la forme $f(n) = 337,5 \text{ Hz} + n \times 15 \text{ Hz}$ avec $n=1$ à 30. Initialement, la valeur maximale pour n était de 20, puis de 30 avec le réseau A3.

Ce qui se trouve dans...

Le boîtier num. 1 : contient l'émetteur/récepteur de radiotéléphonie VHF 160E15 öbL, variante B, type G 50-3. Il contient donc tous les éléments d'un système de communication commandé à distance (par câbles) avec les caractéristiques suivantes :

- 15 canaux sur les 17 canaux duplex du réseau A1, soit 34 fréquences de porteuses VHF, réglées par quartz (plus de 30 quartz) pour ce modèle équipé de canaux.

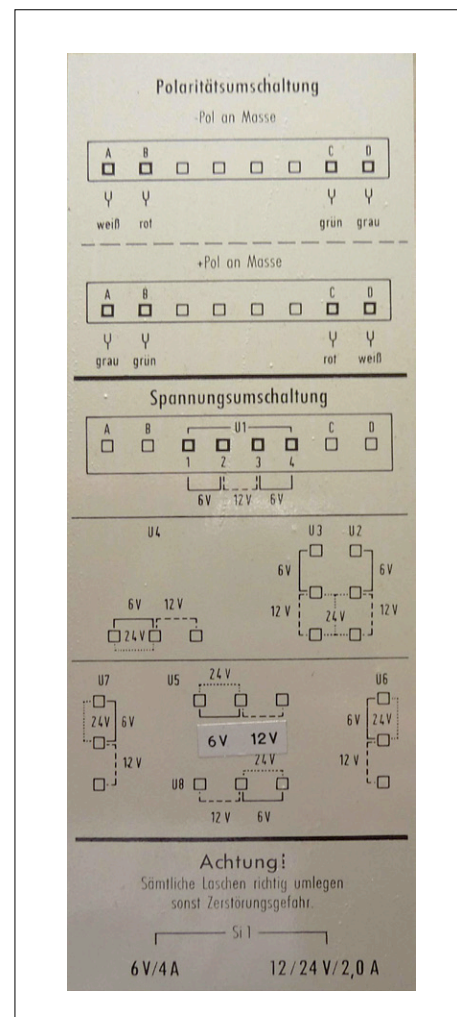


Figure 7. Alimentation au choix. 6, 12 ou 24 V? Positif ou négatif sur le châssis ?

- CH30 : 157,55 MHz / 162,05 MHz à CH44 : 158,25 MHz / 162,75 MHz.
- décalage E/R de 4,5 MHz pour fonctionnement en duplex intégral avec espacement de 50 kHz.
- Modulation : FM (F3), déviation de 4 kHz.



Figure 8. Boîtier de commande standard série AT400 « Becker » pour montage sur le tableau de bord. (Copyright oebL.de, merci !)



Figure 9. Version spéciale du boîtier de commande pour tableau de bord de Mercedes W108/W109.



Figure 10. Les versions Automatik ou deluxe versions du boîtier de commande ont très vite remplacé la version standard.

L'auteur



Gerd Kowalewski a commencé à collectionner et réparer des postes de télévision à tubes. Plus tard, après l'école, il a étudié l'ingénierie électrique et rejoint la jeune pousse CPV, initialement pour concevoir des appareils de transmission de données avant de finir par s'occuper du développement des produits et de la production en Asie. Il a travaillé plusieurs années sur des véhicules électriques et sur l'instrumentation. Son passage au groupe de réflexion *Mannesmann Pilotentwicklung (mpe)* à Munich a engendré quelques brevets. Après avoir dirigé sa propre entreprise, *GK Electronic Consulting (GKEC)*, pendant 10 ans, des ennuis de santé l'ont poussé à une retraite anticipée. Lecteur d'*Elektor* depuis 40 ans, Gerd s'intéresse toujours à la conception de circuits compliqués et aux solutions matérielles efficaces, ainsi qu'aux avancées techniques, futures ou passées.

- Tonalité de signalisation : 1750 Hz ± 1 Hz.
- Puissance RF : 10 W [avant duplexeur]
- Connecteur d'antenne : type UHF (SO-239).

Le boîtier num. 2 : contient la partie STV / SRS, avec l'assemblage STV E 6/12. Il s'agit du conditionnement de l'alimentation 6, 12 ou 24 V du véhicule avec le positif ou négatif relié au châssis, à l'aide d'un convertisseur DC-DC, complètement réalisé en transistors, utilisé pour produire et stabiliser toutes les tensions de fonctionnement.

Consommation énergétique pour une tension de batterie de 12,6 V : 24 W.

Ce boîtier renferme également l'unité de sélection d'appels SRS 1098/12 gérant la signalisation d'appels par fréquences sur le canal de supervision.

Le câble de batterie, les fils de connexion d'origine et même le câble coaxial de la vieille antenne avec base (sans le fouet toutefois) étaient encore présents après un demi-siècle !

... et ce qui n'y est plus

Malheureusement, le boîtier de commande, typiquement un Becker AT400 [4], a disparu, sans doute avec le véhicule dans lequel il était installé. Le boîtier (**fig. 8**) ne comprenait qu'une paire d'indicateurs lumineux, un buzzer et quelques commutateurs, le tout accompagné d'un combiné de téléphone

Funk 70 [5]. Les fonctions principales de communication étaient dans les deux autres boîtiers.

Le boîtier de commande alertait l'utilisateur de l'arrivée d'un appel depuis la station de base à l'aide d'indicateurs lumineux et du buzzer. Une loupote verte indiquait la possibilité de passer un appel via la station la plus proche. Un commutateur rotatif rendait possible (avec l'aide d'une carte des zones radio) la sélection du canal à utiliser en fonction de l'emplacement de l'utilisateur. Des boîtiers de commande mieux finis (**fig. 9**), ainsi que des modèles « automatiques » (**fig. 10**) sont apparus plus tard sur le marché. En plus d'une indication de l'intensité du signal radio, des versions *deluxe* affichaient la tension de batterie : malheur à qui oubliait d'éteindre la radio en quittant

la voiture. Le combiné était un modèle très similaire au W48 utilisé sur le réseau filaire [5]. Le combiné et le boîtier de commande étaient fixés soit sur le tableau de bord du véhicule, près de l'autoradio, soit à l'arrière de la voiture quand elle était conduite par un chauffeur.

Pour conclure

Ce S/E 160E15 öbL de *Telefunken* montre à quelle vitesse la radio mobile a changé en Allemagne en 50 ans. On mesure l'ampleur du développement des réseaux, depuis l'antique réseau terrestre de radio mobile A jusqu'aux réseaux LTE (*Long Term Evolution*) à large bande d'aujourd'hui. L'appareil que j'ai eu la chance de récupérer fonctionne peut-être encore, mais je ne l'ai pas testé. ◀

(140268 – version française : Kévin Petit)



Sources

(en allemand, mais Google Chrome en propose une traduction potable) :

[1] www.oebL.de (remarquable collection privée)

[2] www.oebL.de/A-Netz/Technik/Technik.html

[3] www.oebL.de/A-Netz/Technik/Tabelle.html

[4] www.oebL.de/A-Netz/Geraete/becker/AT400/AT400.html

[5] www.oebL.de/sonstiges/Hoerer/hoerer.html

[6] www.oebL.de/A-Netz/Doku/AT400_Angebot.JPG

Historique de la radio mobile analogique en Allemagne de l'Ouest

Les premières expériences de radiotéléphonie sur les chemins de fer allemands remontent à 1918, d'abord en Grandes Ondes puis en Ondes Courtes autour de 30 MHz. À partir de 1926, les passagers des trains express sur la ligne Berlin-Hambourg pouvaient y passer des appels radiotéléphoniques et, six ans plus tard, en 1932, l'entreprise Mitropa mettait en relation les appels entre des trains en marche et des navires en mer en utilisant la radio marine.

Les services de radiotéléphonie mobile, bien développés avant la prise du pouvoir par les nazis en 1933, furent suspendus pendant la guerre. Après la levée des sanctions et des restrictions imposées par les forces alliées victorieuses, l'administration postale fédérale a progressivement acquis les droits d'utilisation de la radio conformément aux accords internationaux. À partir de 1958, l'administration postale, autorité suprême des télécommunications, a opéré une fusion des services radio disparates en formant le réseau A dans la bande de 156 à 174 MHz, étendu et renommé ensuite A1, A2 puis A3. En 1968, sa couverture était de 80 %. Le déploiement d'un réseau mobile plus efficace a commencé en 1972 avec le réseau B1 et ses 38 canaux. En dépit de l'augmentation significative des prix, le nombre d'abonnés a continué d'augmenter fortement jusqu'à la fin des années 70. À cause du nombre limité de canaux radio libres, il a fallu réutiliser les fréquences porteuses du réseau A sur le nouveau réseau B2, pour atteindre en 1980 un total de 75 canaux voix. Au passage, l'espacement des canaux a été réduit de 50 kHz à 20 kHz et le canal 19 est devenu le canal d'appel standard pour tous les états fédéraux. Grâce aux multiples réutilisations de ces canaux de fréquence,

sur des cellules radio séparées convenablement, le service disposait de 850 fréquences utilisables simultanément pour accéder au réseau de téléphonie fixe. Le réseau B a pour la première fois rendu possible le passage d'appels sans mise en relation par les fameuses opératrices, à condition que l'abonné sache dans laquelle des 158 zones radio différentes de la République fédérale il se trouvait, ainsi que le préfixe de numérotation à utiliser pour atteindre le réseau fixe. L'itinérance entre les états n'était pas encore d'actualité.

Le premier combiné portable, le Dynatac 8000 de *Motorola*, connu en Allemagne comme *The Bone* (NdT : l'os, à cause de sa forme), avec environ 30 min d'autonomie, a plus de 30 ans ! En 1986, la croissance du nombre d'abonnés poussait le réseau B, avec 27.000 utilisateurs, dans ses retranchements. Apparu fin 1985, le réseau C offrait une alternative pour la téléphonie mobile dans la bande des 450 MHz, avec commutation automatique et code d'accès unique (0161) dans tout le pays : une première, suivie par le transfert intercellulaire, qui permettait de poursuivre une conversation tout en passant d'une zone radio à une autre, et de localiser les utilisateurs au sein du réseau. Le réseau B n'a été mis hors service qu'en 1994.

À son apogée, le réseau C comptait 800.000 utilisateurs. Le prix de l'abonnement avait peu à peu baissé de 84 %. Grâce aux progrès de la miniaturisation, les combinés devenaient portables. Le réseau C a été mis hors service en 2000. La suite, vous la connaissez.

Sources

(principalement en allemand, mais Google Chrome en propose une traduction portable) :

www.izmfm.de

www.oebf.de/A-Netz/ANetz.html

www.oebf.de/B-Netz/BNetz.html

www.oebf.de/C-Netz/CNetz.html

www.mobilfunk-geschichte.de

wissen.de/die-geschichte-der-mobiltelefone

de.wikipedia.org/wiki/A-Netz

de.wikipedia.org/wiki/B-Netz

de.wikipedia.org/wiki/C-Netz

de.wikipedia.org/wiki/D-Netz

de.wikipedia.org/wiki/E-Netz

www.3gpp.org

www.bundesnetzagentur.de/DE/Sachgebiete/Telekommunikation/Unternehmen_Institutionen/Marktbeobachtung/Deutschland/Mobilfunkteilnehmer/Mobilfunkteilnehmer_node.html

en.wikipedia.org/wiki/Universal_Mobile_Telecommunications_System

[en.wikipedia.org/wiki/LTE_\(telecommunication\)](http://en.wikipedia.org/wiki/LTE_(telecommunication))

en.wikipedia.org/wiki/LTE_Advanced

www.focus.de/digital/handy/mobilfunkgeschichte/

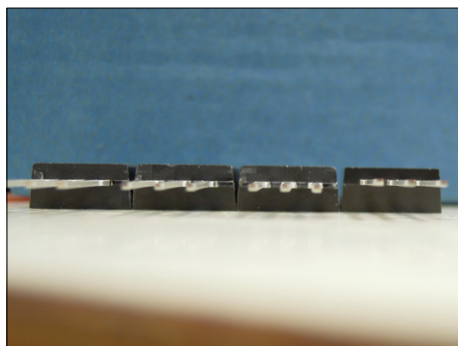
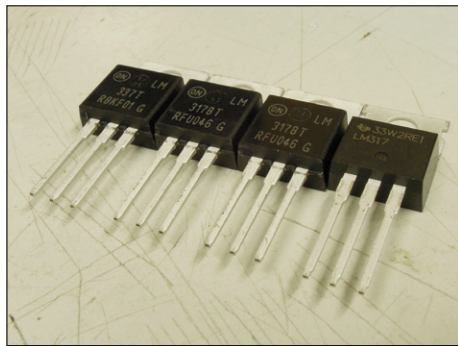


implantation des c.i.

Thijs Beckers (rédaction des Pays-Bas)

1^{ère} partie : TO-220 ≠ TO-220

Notre collègue Ton avait besoin pour un projet de diverses alimentations. Il avait opté pour un circuit simple avec quelques régulateurs standard LM317 en boîtier TO-220, à monter sur des radiateurs, vu qu'ils devraient fournir pas mal de courant. Lors de l'implantation sur le circuit imprimé, une surprise attendait Ton ; il semblait que les trous n'étaient pas alignés : un des LM317 était bien adossé à son radiateur, alors que la semelle de l'autre formait un angle avec le radiateur. Les broches de ce circuit intégré ont dû être légèrement pliées pour assurer un bon contact avec le radiateur. C'était très curieux.



Après inspection du montage, l'alignement des trous du circuit imprimé était hors de cause ; c'étaient bel et bien les LM317 qui posaient problème ! On peut voir sur les photos que les broches ne sortent pas toutes du boîtier au même endroit. Pour les trois circuits de gauche, de chez ON Semiconductor, les broches sont bien alignées et s'échappent du boîtier à hauteur de la jonction des deux parties de la coque. Pour le quatrième LM317, de chez Texas Instruments, les broches émergent du boîtier juste au-dessus de cette jonction.

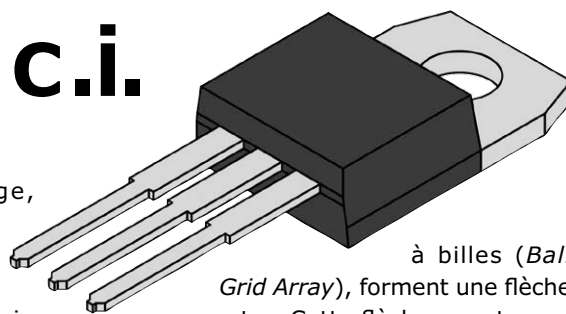
Les dimensions des boîtiers sont bien indiquées dans les fiches de caractéristiques des fabricants, mais au vu des marges nous ne sommes guère plus avancés : hauteur entre semelle et broches de 2,03 à 2,93 mm chez Texas Instruments, de 2,04 à 2,79 mm chez ON Semiconductor. Cela ne nous gênera pas trop lors d'une expérimentation occasionnelle, mais si vous n'étiez pas au courant de ces divergences, vous voilà prévenus !

2^e partie : où est la broche 1 ?

Au labo d'Elektor, il nous arrive de tomber sur des boîtiers de circuits intégrés sur lesquels la broche 1 n'est pas repérée (point de couleur, encoche en demi-cercle, ou petite marque au laser), mais identifiée grâce à un côté du boîtier biseauté. Sur ce circuit intégré, un TS922 de chez STMicroelectronics en boîtier SO-8, la broche 1 est « en haut à gauche » sur le côté biseauté à 45°.

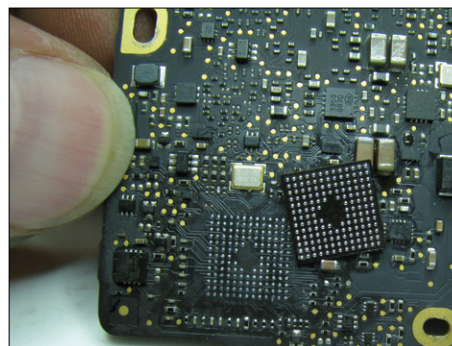
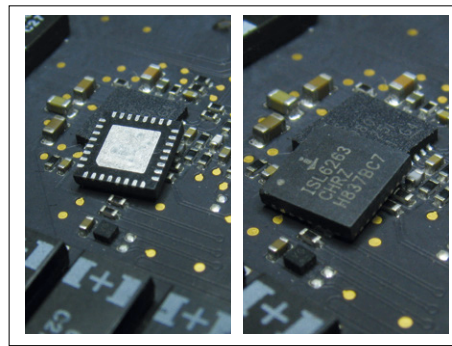
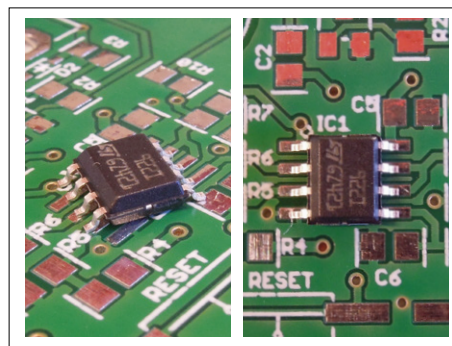
Nous en avons encore que la plupart d'entre vous ne rencontreront pas tous les jours. Ainsi la broche 1 de ce circuit est identifiée par un coin biseauté du PowerPAD™. Ceci ne se voit que lorsque le circuit n'est pas encore muni de son dissipateur, ou alors il faut disposer d'un scanner à rayons X. Il y a néanmoins un point de couleur sur le dessus du boîtier, ce qui permet de vérifier l'orientation correcte lors du montage.

En voici un que vous ne verrez pas souvent non plus, et son concepteur avait sans aucun doute le sens de l'humour ! Les contacts de ce boîtier, appelé matrice



à billes (Ball Grid Array), forment une flèche au centre. Cette flèche se retrouve bien entendu sur le circuit imprimé, ce qui permet une orientation correcte lors de l'implantation. La broche 1 est aussi indiquée par un point de couleur sur le boîtier, comme pour le circuit précédent. Et si vous voulez vraiment savoir où trouver ce circuit particulier : il s'agit du contrôleur de gestion du système (System Management Controller) d'un Apple MacBook Pro A1425, avec les entrailles à l'air sur la table de réparation (en fait une table de cuisine...). ◀

(150369 - version française : Jean-Louis Mehren)





NIDays

coupe de robotique

NXT 2016

Mariline Thiebaut-Brodier (Elektor)

La passionnante finale de la coupe de robotique NXT 2016, organisée par National Instruments, a eu lieu le 10 mars 2016. La compétition est ouverte aux lycées (enseignement secondaire et BTS), IUT, classes préparatoires aux grandes écoles (CPGE) ainsi qu'aux classes préparatoires intégrées. Le cœur du robot était le même pour tous, mais les robots engagés dans la compétition ne se ressemblaient pas du tout. Faites la connaissance de l'équipe gagnante ici.

Principe du concours

Les participants conçoivent un robot autonome à l'aide du kit LEGO® MINDSTORMS® NXT. En moins de cinq minutes, le robot doit parcourir un terrain de jeu et réussir une succession d'épreuves. Ce périple doit être effectué en totale autonomie, c'est-à-dire sans intervention de la part des participants. Une fois le robot lancé, il est interdit aux membres de l'équipe de toucher les robots, l'aire de jeu ou un quelconque élément.

Dès que le robot a quitté la zone de départ, il doit zigzaguer entre des parois de 30 cm de haut, éviter des Rubik's cubes et enfin pousser un interrupteur de type va-et-vient.

Si cette première mission se déroule en moins de 50 s, le nombre de points attribué est doublé. La deuxième mission consiste à récupérer une balle NXT de couleur rouge ou bleue et à la déposer dans une zone de sécurité de la même couleur. Cette couleur est définie avant le début de l'épreuve par tirage au sort. Des lignes de suivi sur le terrain permettent de guider le robot vers cette zone de sécurité. Enfin, pour sa dernière mission, le robot doit faire demi-tour pour regagner la zone de collecte, repérer la ligne verte à suivre pour grimper quelques marches d'escalier ou une rampe (montée et descente). La zone d'arrivée se trouve juste après ces obstacles.

Travail d'équipe

L'ensemble scolaire Jeanne d'Arc (de la maternelle au BTS) situé à Argentan (61) a présenté une équipe issue d'une classe de terminale S, spécialité Informatique et Sciences du Numérique (ISN), accompagnée par M. Jafari, leur professeur de physique/chimie et ISN. Parmi les différents sujets abordés (création de site web, algorithmique, son...), le programme ne prévoit que 4 à 5 h de robotique sur l'année. Yohan Bisson et Arthur Pain entre autres ont décidé de se lancer dans l'aventure après en avoir entendu parler par un de leurs camarades. Le grand frère de ce camarade, ancien élève, aujourd'hui ingénieur, a travaillé chez NI et leur a présenté ce concours.

Au début, les deux élèves ont travaillé tous les mercredis matins pendant une heure de permanence. Ils ont pu se frotter à la programmation du robot. Pendant les vacances de février, ils se sont réunis avec leur professeur. L'agent d'entretien du lycée leur avait préparé une réplique du plateau utilisé pour la coupe, et l'avait installée dans une pièce réservée à cet effet (**figure 1**). Il s'agit d'une surface plane de 2 m par 3 m, avec un revêtement en PVC blanc mat, entourée d'une bordure de 30 cm de haut de couleur anthracite, en plastique ou en bois. Le quatuor pouvait donc tester en grandeur nature le comportement du robot. Enfin, les derniers jours, nuits comprises, avant

la compétition, toute l'équipe a été sur le pont pour régler nombre de détails. Que de stress !

Un chemin semé d'embûches

Dès le départ, les élèves ont décidé de fractionner le travail ; chacun program-mait des morceaux du programme et les testait. Les problèmes sont apparus lorsqu'ils ont tout mis dans un seul programme : le logiciel de développement (V2 *Lego Mindstorms Education NXT*) a planté ! Au bout de plusieurs heures, les élèves ont réussi à contourner le problème ; il fallait ajouter les segments de programme un à un. Apparemment ce problème est lié à la mauvaise gestion de la mémoire tampon. Le programme se décompose en trois parties :

- Chemin dans le labyrinthe et pression sur l'interrupteur
- Récupération de la balle et dépôt
- Pente – dès le départ, ils ont choisi la pente. Le franchissement de l'escalier semblait plus hasardeux. Ils ont limité les risques.

Ensuite ils ont eu la joie de découvrir les problèmes matériels. Leur première version du robot était équipée de roues montées sur des axes. Au bout d'une demi-heure de fonctionnement, les roues glissaient petit à petit sur les axes et finissaient par tomber. C'est pourquoi ils sont passés aux chenilles.

Au bout de deux heures d'utilisation, les déplacements du robot devenaient si désordonnés qu'il lui était impossible de suivre une ligne droite. En fait, c'est la tension de la batterie qui était devenue insuffisante pour que les moteurs tournent tous correctement. Toutefois il a fallu un certain temps pour comprendre d'où venait le problème. En résumé, ils n'ont eu que de « petits » soucis.

Stratégie gagnante

Les règles imposées pour le concours n'ont posé aucun problème particulier à nos compétiteurs. Leur secret pour gagner a été de jouer sur les moteurs pour ne pas aller trop vite ni trop lentement. Si le robot est trop rapide, ses lignes droites ne sont pas parfaites, et il a du mal à s'arrêter juste avant l'obstacle. Si le freinage est violent, les liaisons mécaniques le supportent mal, faute de stabilité et de souplesse adéquates. D'ailleurs un robot concurrent très, très véloce a fini coincé... La précipitation n'est pas payante. Il faut associer vitesse et efficacité.

Toutefois il ne faut pas être trop lent pour espérer obtenir le bonus attribué si on réussit la première mission en moins de 50 s. Pour la première mission, le temps de parcours du robot est passé de 1 mn 27 à 46 s.

Matériel

Outre le contenu du kit, il n'est possible d'ajouter au robot que des éléments LEGO®, le tout sans dépasser la taille maximale du robot (contenu dans un cercle de 43 cm). Les élèves ont veillé à bien situer le centre de gravité du robot. Leur robot est équipé de trois types de capteurs :

- photosensible pour détecter la couleur de la balle
- de couleurs pour suivre les lignes sur le terrain
- à ultrasons à l'avant et sur les côtés pour se déplacer

Toutefois comme les Lego® Mindstorms® NXT sont en fin de vie (remplacés par Lego® Mindstorms® EV3), l'équipe a eu beaucoup de mal à trouver le capteur de couleurs.

Les derniers soucis sont apparus sur le terrain parisien. En effet, pour augmenter la stabilité du robot, une tige munie d'un



Figure 1. Arthur Pain et Yohann Bisson aidés par Amélie Leriche et Dimitri Le Conte travaillent sur la reproduction à taille réelle du terrain de jeu.

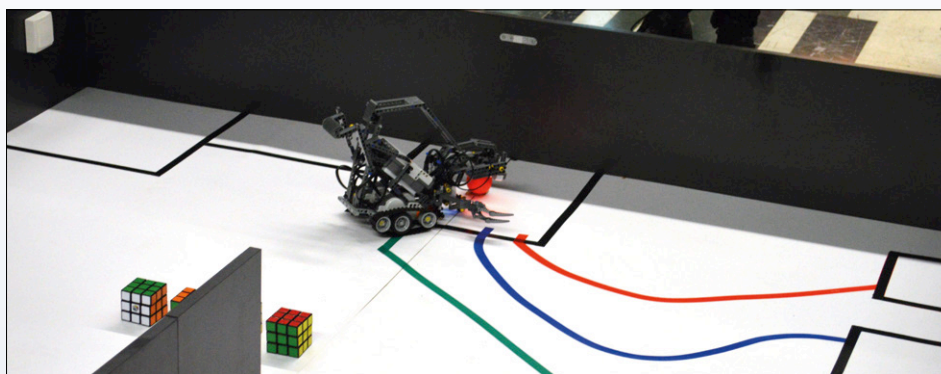


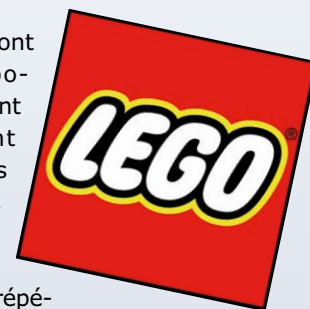
Figure 2. Le robot en piste : il récupère la balle rouge pour la mettre dans la zone de sécurité correspondante.

embout rond avait été ajoutée à l'avant. Toutefois comme les deux planches de la rampe ne joignaient pas parfaitement, cette tige restait coincée dans l'interstice entre ces deux planches. Il a fallu faire une modification de dernière minute. Bien entendu, les élèves avaient apporté des pièces supplémentaires pour réaliser toutes les adaptations nécessaires ainsi qu'un deuxième robot (en cas de panne du premier).

Et si c'était à refaire ?

Même si la préparation de la coupe représente un énorme travail, les élèves se lanceraient sans hésiter à nouveau dans l'aventure. La joie de la victoire a chassé instantanément la fatigue. Grâce à cette expérience, les élèves ont appris à travailler en groupe et à se coordonner.

Non seulement ils ont découvert la robotique, mais ils savent aussi maintenant qu'ils sont capables de se dépasser, conformément aux promesses de leur professeur qui, en répétant qu'il faut « bosser » pour réussir, leur a transmis la valeur du travail et de l'effort.



Ils sont très fiers d'avoir battu des équipes de Prépa PCSI, d'IUT et de terminale S option SI. La 2^e marche du podium est occupée par l'ISEN de Lille, et la 3^e par le lycée Condorcet Schoeneck et l'IUT MP Montbéliard. ◀

(160092)

Kodi : lecteur multimédia pour PC, Raspberry Pi, tablette et bien d'autres



Harry Baggen (labo d'Elektor)

PC, ordinateurs portables, ordiphones, tablettes..., aujourd'hui ils sont tous équipés d'un logiciel pour regarder des photos et des vidéos, et écouter des fichiers audio. Mais en général cet outil n'est pas universel ni assez souple : certains formats ne sont pas pris en charge, des paramètres manquent ou encore il n'est pas possible de lire un fichier en ligne. Quelle est la meilleure alternative ?

Il existe beaucoup de lecteurs alternatifs pour différentes plateformes, mais en termes de polyvalence et d'extensibilité, un logiciel sort du lot : Kodi, peut-être plus célèbre sous son ancien nom XBMC. Il s'agit d'un lecteur multimédia à code source ouvert, développé par la Fondation XBMC (un groupe de programmeurs/utilisateurs enthousiastes). Kodi/XBMC est un lecteur qui a été conçu en 2004 pour la première Xbox, d'où son nom *Xbox Media Center* (abrégié en XBMC). Les versions ultérieures ont été portées sur d'autres systèmes d'exploitation, notamment Windows, Linux, iOS et Android. Il y a également eu plusieurs versions autonomes pour des téléviseurs interactifs, des décodeurs et des lecteurs multimédias. Désormais Kodi prend en charge plus de 75 langues.

Un des points forts de Kodi/XBMC est la possibilité d'ajouter des modules complémentaires (*add-ons*) qui permettent de multiplier les fonctions ou d'avoir accès à des sources de médias externes via l'internet. Outre les innombrables additifs officiels, il en existe de nombreux autres qui permettent de consulter toutes sortes de sources audio et vidéo illégales. La Fondation XBMC prend expressément ses distances par rapport à ces sources.

Dans cet article, je vais vous présenter les nombreuses possibilités de Kodi et des appareils sur lesquels vous pouvez l'utiliser. Si de nombreux collègues et amis semblent connaître le nom de Kodi, ils ne savent pas comment s'en servir. En fait, c'est assez facile et vous trouverez sur YouTube des centaines de vidéos pédagogiques pour vous aider à vous y mettre.

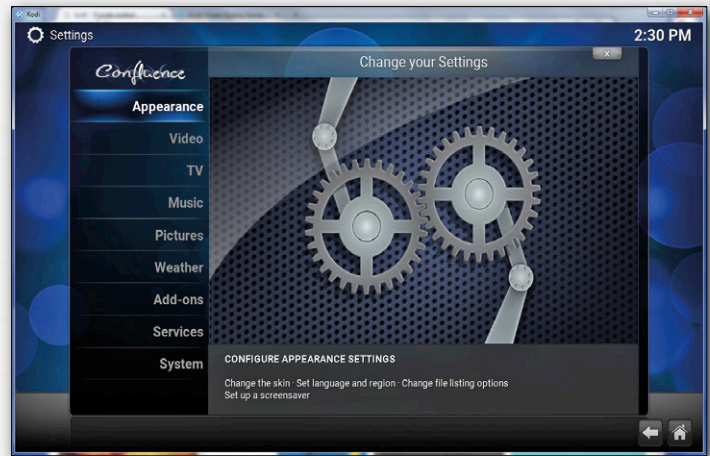
Abondance de versions

Sur le site officiel de Kodi [1], la page *Downloads* présente toutes les versions actuellement disponibles pour Windows, Linux, Mac OS X, iOS, Android et une série de plateformes matérielles spécifiques comme Raspberry Pi et Amazon Fire TV. Pour la plupart des systèmes, l'utilisateur n'a qu'à télécharger un fichier d'installation et à l'exécuter sur son ordinateur. Pour les appareils Android, Kodi est également disponible dans le Google Play Store. Aujourd'hui, un nombre croissant de lecteurs multimédias qui tournent sous Android arrivent sur le marché, Kodi y est de plus en plus présent.

Le nano-ordinateur Raspberry Pi est une plateforme matérielle très populaire, qui permet de construire soi-même et à moindre coût un lecteur multimédia.

Un grand nombre d'entre eux sont sans doute achetés dans ce but. Dès la première version du RPi, on trouvait des versions de Kodi spécialement adaptées à cet effet. OpenELEC (*Open Embedded Linux Entertainment Center* [2]) est l'une des plus populaires. C'est ce qu'on appelle un *Just Enough Operating System* (JeOS) ; vous n'avez donc pas à installer Linux ou un autre système d'exploitation, tout le nécessaire est déjà intégré dans le logiciel. OpenELEC est disponible pour Raspberry Pi A, B, 2 et 3, mais aussi pour plusieurs autres plateformes matérielles comme les PC équipés d'un processeur x86 (idéal pour un PC serveur multimédia, pas besoin de Windows) et des « boîtes » multimédias équipées du processeur Freescale IMX6 (Cubox). Bien qu'il ne soit pas difficile d'installer le logiciel OpenELEC sur un RPi, il fonctionne un peu différemment de la version Kodi ordinaire. Il faut télécharger une image du disque (une copie exacte du contenu de la mémoire) qui doit ensuite être transférée sur une clé USB ou une carte (micro-)SD. Elle ne peut pas être copiée en utilisant l'Explorateur de Windows ou un autre programme du même genre. Sur une page du site d'OpenELEC [3], vous trouverez

► Kodi et RPi, un mariage idéal pour donner naissance à un lecteur multimédia autonome



comment faire sous Linux, Windows et Mac OS X. La plupart des lecteurs d'Elektor utilisent Windows, auquel cas il faut avoir recours à un petit programme, Win32Diskimager [4], pour transférer fidèlement l'image sur une clé USB ou une carte SD. Il suffit ensuite de connecter la clé ou d'enficher la carte dans le RPi, et de le démarrer.

Utilisation

Au début, habituez-vous à l'interface de Kodi et aux nombreux réglages possibles. Le mieux c'est de d'abord jeter un œil au guide de démarrage rapide (*Quick Start Guide*) du wiki Kodi [5] pour découvrir l'interface et ses réglages. Sous *System/Settings*, il y a un grand nombre d'onglets avec des possibilités de réglage ; vous pouvez y choisir votre niveau entre *Basic*, *Standard*, *Advanced* et *Expert* (commencez avec *Standard*, tous les paramètres importants vont s'afficher). La plupart des réglages doivent rester tels quels, sauf peut-être la langue de l'interface sous *Appearance/International* et la langue de sous-titrage des vidéos sous *Videos/Subtitles*. Dans *System/Audio output*, vous pouvez sélectionner le format de sortie audio. Dans un RPi, cela dépend si la sortie HDMI est connectée à un récepteur *surround*, ou directement à un téléviseur.

En plus de lire des vidéos et des photos de votre collection personnelle, il est bien sûr intéressant d'activer un certain nombre

d'additifs qui permettent d'accéder à une variété de sources multimédias sur l'internet, depuis les émissions que vous avez ratées jusqu'aux programmes de cuisine et les informations de divers pays. En sélectionnant *Videos/Add-ons*, et ensuite *Get more...* (il en va de même pour *Pictures* et *Music*), vous voyez une liste des additifs présents par défaut dans Kodi. Ici, vous pouvez installer les additifs que vous souhaitez. Ensuite, ils seront visibles dans *Videos/Add-ons*. La liste peut être étendue à de nombreux autres additifs. Un aperçu de tous les additifs officiels de Kodi est disponible en [6]. Ce faisant, attention à la version de Kodi que vous utilisez. Chaque version de Kodi a son propre nom sous lequel sont regroupés les additifs. La version la plus récente est la 16, baptisée « Jarvis ». L'installation

d'additifs est décrite dans le wiki Kodi [7], mais il est conseillé de consulter YouTube sur ce sujet.

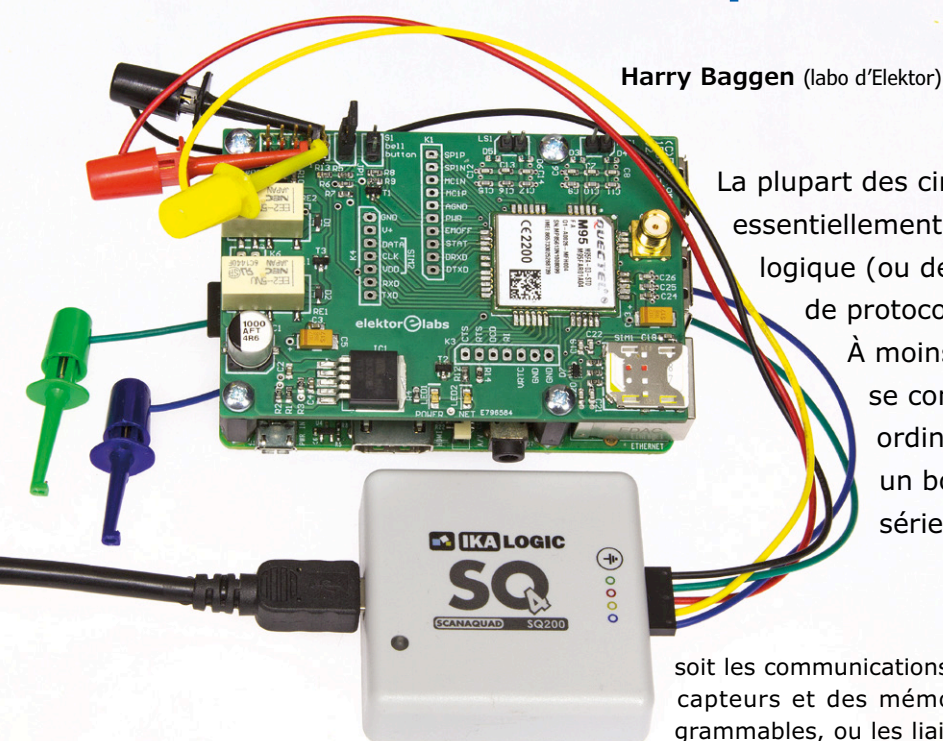
Tous les additifs ne sont pas approuvés par les développeurs de Kodi, mais il y a en circulation beaucoup d'additifs non officiels intéressants (à ne pas confondre avec les additifs illégaux). Citons deux sources qui ont réuni un grand nombre d'additifs et les ont assemblés dans des paquets zip bien pratiques ; l'utilisateur a ainsi accès en une fois à un grand nombre d'additifs : Super Repo [8] a réuni plus de 2 000 additifs, et TVAddons [9] en compte 1 200. Outre ces deux sources, il existe beaucoup de modules complémentaires illégaux, faites donc attention à ce que vous installez ! ◀

(160031 version française : Eric Dusart)

Liens

- [1] <https://kodi.tv/>
- [2] <http://openelec.tv/>
- [3] http://wiki.openelec.tv/index.php/HOW-TO:Installing_OpenELEC/Creating_The_Install_Key#tab=DiskImage
- [4] <https://sourceforge.net/projects/win32diskimager/>
- [5] http://kodi.wiki/view/Quick_start_guide
- [6] http://kodi.wiki/view/Category:All_add-ons
- [7] <http://kodi.wiki/view/Add-ons>
- [8] <https://superrepo.org/>
- [9] <https://www.tvaddons.ag/>

ScanaQuad, le plus petit analyseur logique décode de nombreux protocoles série



Harry Baggen (labo d'Elektor)

La plupart des circuits électroniques actuels sont essentiellement numériques ; l'acquisition d'un analyseur logique (ou devrions-nous plutôt dire un « analyseur de protocole » ?) s'avère de plus en plus pertinente.

À moins d'avoir gagné à l'EuroMillions, on devra se contenter d'un appareil USB connecté à un ordinateur. Mais lequel est pratique, versatile et a un bon rapport qualité-prix ? Nous avons testé la série ScanaQuad de la société française Ikalogic.

Si vous concevez ou réparez essentiellement des circuits numériques, un analyseur logique vous sera plus utile qu'un oscilloscope, mais faut-il vraiment acquérir un tel appareil ? Les analyseurs logiques proposés par les principaux fabricants restent inabornables pour les amateurs, et même pour une jeune société. Quant aux solutions bon marché, elles ne répondent pas aux besoins réels des utilisateurs ou souffrent d'un logiciel mal conçu ou incomplet. La situation s'améliore, lentement mais sûrement.

Quatre modèles

La société française Ikalogic propose depuis 2010 des analyseurs logiques avec connexion USB vers un ordinateur. Rien d'extraordinaire direz-vous, mais fin de l'année dernière Ikalogic a sorti une nouvelle série d'analyseurs à quatre canaux. Pourquoi pas huit, ou plus ? Quand on y réfléchit, la raison semble évidente. La plupart des microcontrôleurs et autres circuits numériques actuels utilisent des liaisons série pour économiser sur le nombre de broches et de connexions : que ce

soit les communications avec des capteurs et des mémoires programmables, ou les liaisons avec le monde extérieur, tout passe par I²C, I²S, USB, 1-Wire, etc. Tous ces bus n'ont besoin que de deux, trois ou quatre lignes, alors pourquoi en demander plus ? Ikalogic y a réfléchi et a sorti la série d'analyseurs ScanaQuad, qui intègre dans son logiciel la plupart des protocoles série courants et se limite donc à quatre canaux. Et si on en veut vraiment plus – jusqu'à 24 – les modules ScanaQuad peuvent être connectés en parallèle par un adaptateur.

La série comprend quatre modèles (SQ25, SQ50, SQ100 et SQ200) qui diffèrent essentiellement par leur fréquence d'échantillonnage (le chiffre indique cette fréquence, en MHz : SQ25=25 MHz). Il est à noter que la fréquence est indépendante du nombre de canaux utilisés, ce qui n'est pas le cas de bon nombre d'analyseurs, où elle est inversement proportionnelle au nombre de canaux en service. Les deux modèles les plus rapides offrent une plage d'entrée plus étendue (± 15 V), plus de configurations d'entrées/sorties, et peuvent effectuer des mesures différentielles.

L'utilisateur peut définir lui-même chaque connexion, par exemple laquelle est la ligne SDA et laquelle SCL pour un bus I²C. En outre, chaque connexion peut aussi fonctionner en sortie, sur laquelle on retrouvera par exemple un signal carré de fréquence et rapport cyclique réglables, ou même des séries complètes de données définies à l'aide d'instructions en JavaScript. Cette souplesse d'emploi vaut aussi pour le décodeur de protocole : vous pouvez créer les vôtres, mais il en existe déjà de nombreux prédéfinis. Les données lues peuvent bien entendu être enregistrées, mais aussi modifiées et « rejouées » via un des canaux.

Protocoles actuellement disponibles : 1-Wire, CAN, DHT11 et 22, DMX-512, HC-SR04, I²C, I²S, JTAG, 4-bits LCD, LIN, Manchester, Maple, MIDI, NMEA 0183, nRF24L01, Oregon Scientific, bus parallèle, PWM, SENT, SPI, SPI WIZnet W5100, UART, Atmel SWI, MODBUS, HDMI-CEC et USB1.1. Certains capteurs sont aussi supportés par le logiciel : pour le moment quelques capteurs de température (1-Wire et I²C) et d'humidité (I²C).

À l'usage

Nous utilisons un ScanaQuad (SQ100) au labo d'Elektor depuis quelque temps.

► Cet analyseur est incroyablement petit, à peine 5 cm sur 5 !

Lorsque nous avons reçu le colis – qui n'était pas bien grand – et l'avons ouvert, une surprise « de taille » nous attendait : l'analyseur est incroyablement petit, à peine 5 cm sur 5 ! Ce boîtier gris contient pourtant tout ce qu'il faut pour un analyseur rapide à quatre canaux. Le colis comprend aussi un câble USB et un cordon de mesure avec cinq griffe-fils.

Le logiciel ScanaStudio, téléchargeable sur le site d'Ikalogic pour Windows, OS X ou Linux, est facile à installer. L'interface est très intuitive, et après quelques tâtonnements on maîtrise déjà les fonctions principales. Certaines tâches, comme la création d'un décodeur de protocole, sont bien sûr compliquées, mais il est évident que ce programme a été écrit par des techniciens pour des techniciens. Tout est clair (logique ?) et bien organisé.

Lors du démarrage, on ouvre un « espace de travail », nouveau ou existant, une fenêtre où les quatre canaux sont affichés et peuvent être configurés. On peut ouvrir plusieurs espaces de travail en même temps, avec par exemple des protocoles différents ; ceci permet de passer facilement de l'un à l'autre, mais seuls les paramètres de la fenêtre active sont utilisés. Les quatre canaux sont bien visibles, et on peut en masquer un ou plusieurs pour simplifier l'affichage.

Un menu permet de configurer chaque entrée ou sortie selon divers paramètres : niveau logique à mesurer, définition de

chaque ligne comme entrée ou sortie, drain (collecteur) ouvert ou E/S I²C, entrée RS-232, valeur de la résistance de rappel vers le bas et éventuellement vers le haut. Pratiquement toutes les situations possibles sont configurables.

Le nombre d'options de déclenchement est lui aussi quasiment illimité : manuel, front montant ou descendant ; un menu « glisser-déposer » permet même de créer une série de niveaux et fronts. On peut aussi entrer une valeur décimale, hexadécimale ou ASCII pour le déclenchement. On regrettera peut-être l'absence d'un mode non asservi, qui permettrait de vérifier la présence d'un signal sur une ligne.

Après déclenchement, les données sont inscrites dans le tampon de l'appareil, dont la taille dépend du modèle (1 à 16 MB).

Les données ne sont pas transmises via l'USB : le tampon interne est bien plus rapide, et en principe de taille suffisante pour contenir toutes les données à analyser. On peut naviguer et zoomer dans l'ensemble du tampon à partir de l'espace de travail, placer des marqueurs et effectuer diverses mesures : par exemple le rapport cyclique, ou le nombre de flancs (montants ou descendants) entre deux marqueurs. On peut également ouvrir une autre fenêtre pour y afficher les données au format hexadécimal ou par paquets avec horodatage associé.

Plus on passe de temps avec un



Figure 1. L'électronique du ScanaQuad est bâtie autour d'un circuit FPGA Spartan de Xilinx et d'une mémoire rapide.

module ScanaQuad, plus on apprécie la simplicité et la richesse de l'interface de ScanaStudio, et nous n'avons vraiment rien trouvé à lui reprocher. On pourrait bien entendu remarquer que la définition d'un protocole personnel est difficile, surtout pour les débutants ; c'est une tâche complexe qui demande de la concentration, mais là encore Ikalogic a prévu des instructions détaillées sur son site.

Avec le ScanaQuad on en a pour son argent, c'est indéniable. À vous de choisir le modèle qui vous convient, en fonction de la vitesse des signaux à analyser. *Riquiqui, mais maous costaud !* ◀

(160101 – version française : Jean-Louis Mehren)

Lien

www.elektormagazine.fr/news/banc-d-essai-scanaquad-analyseurs-logiques-minuscules

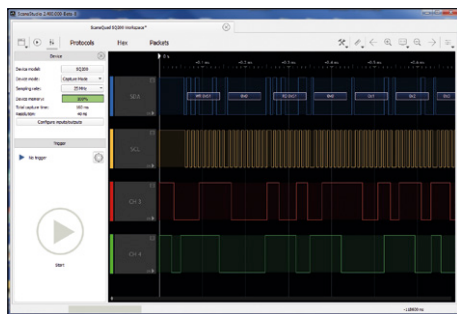


Figure 2. L'espace de travail bien ordonné de l'interface ScanaStudio.

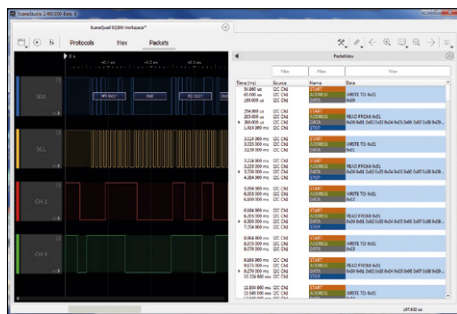


Figure 3. Les données, à droite de l'écran, peuvent être affichées en format hexadécimal ou – comme ici – par paquets.

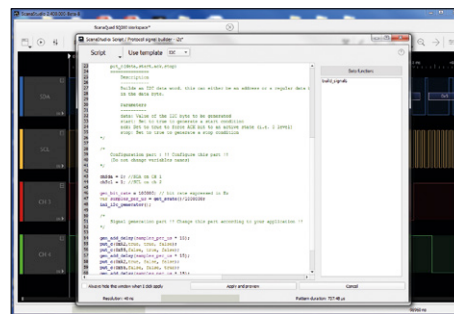


Figure 4. Le script du générateur de signal du protocole contient des modèles que l'on peut utiliser comme exemple.

compilées par **Robert van der Zwan**

Debout l'Australie !



Membre de longue date de la communauté Elektor, Bill Morgan travaille à Sydney comme ingénieur en électronique. Il entend « réveiller » ses compatriotes, car lance-t-il : « Nous ne pouvons plus compter uniquement sur nos ressources naturelles. » Que faire ? Se tourner vers les ressources intellectuelles, enseigner la programmation aux enfants, répond Bill. Il part pour un voyage qui démarre avec Scratch, un langage de programmation visuel très ludique. Nous mettrons bientôt cette initiative sous le feu des projecteurs elektoiriens.

Une publication annuelle ?

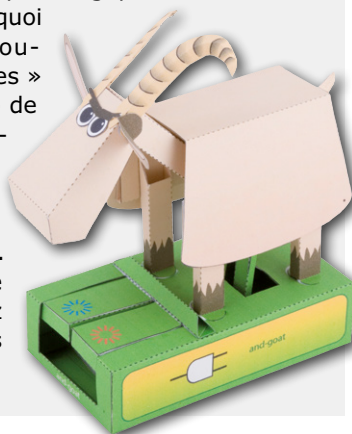
Les éditions française et allemande du magazine sont mensuelles, les versions anglaise et néerlandaise sont bimestrielles. Et il y a bien sûr les nouvelles quasi quotidiennes du site web. Nous songeons maintenant à une publication annuelle ! Nous vous tiendrons informés bien sûr. En attendant, si vous avez des suggestions, écrivez-nous à redaction@elektor.fr.



READ ONLY MEMORY

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

Une compréhension fine des processeurs passe par celle des microscopiques interrupteurs et portes logiques qui les composent. Tout le monde comprend le fonctionnement d'un interrupteur, celui d'une porte logique est moins intuitif. C'est pourquoi Rob Ivens a mené son troupeau de « Biquettes Logiques » jusqu'aux pages du numéro de juillet & août 2008 d'Elektor. Les Biquettes sont des modèles en papier à monter soi-même de portes logiques ET, OU, NON et OU Exclusif. Les Biquettes broutent encore les pages de l'internet. Tapez « Logic Goats » pour les retrouver !



Projet TABULA

Pour dispenser ses cours de technologie de l'information (TI), l'université technique d'Aix-la-Chapelle (RWTH) prévoit d'utiliser un matériel tellement à la pointe de la technologie qu'il est encore en cours de développement : un écran tactile de la taille d'une table, sur laquelle les étudiants placent des objets interactifs. Le laboratoire Elektor allemand fait partie de ce projet novateur baptisé TABULA.

TABULA révolutionne l'apprentissage des TI. Les objets, ou « tangibles », peuvent être placés et déplacés librement sur la table-écran. Lorsqu'un *tangible* est déplacé, son électronique interne interagit avec la table et modifie la configuration ou les informations affichées sur l'écran.

L'étudiant peut utiliser plus d'un *tangible* à la fois et ainsi étudier l'effet de diverses relations inter-objets. TABULA repose sur une technologie conçue par l'université RWTH : PERCS, pour *Persistently Trackable Tangibles on Capacitive Multi-Touch Displays*. PERCS traite également la détection permanente des objets, pas uniquement la détection de leur mouvement.

D'après le professeur Jan Borchers, du centre d'interaction homme-machine de l'université RWTH : « Grâce à l'écran tactile, les étudiants étudient concrètement des concepts utilisés par de nombreuses branches des TI. Ils apprennent par exemple à concevoir des algorithmes d'insertion d'objets dans un arbre binaire, ou à programmer par nœuds et graphes des filtres de traitement d'image. »

L'équipe allemande du labo Elektor conçoit



INDISCRÉTIONS • Notre graphiste Giels Dols a fêté son soixantième anniversaire en mai dernier, un de nos bureaux de Limbricht (et derrière lesquelles l'attendaient une tasse de café et une friandise locale) programme collaboratif TABULA présenté ci-dessus • Tom Verboven met au point un logiciel de gestion des répondre à leurs attentes.

en collaboration avec Elektor

actuellement des prototypes de *tangibles*. Ils seront dotés d'une certaine intelligence locale, notamment pour communiquer leur position à l'unité centrale de la table, probablement via Bluetooth Low Energy. Ils sont en outre équipés d'un capteur optique.

L'université RWTH se concentre quant à elle sur le support pédagogique proprement dit, en particulier sur les structures de données et la programmation par flux de données. Parmi les membres du projet TABULA figure aussi une société de formation en ligne d'Aix-la-Chapelle.

<http://hci.rwth-aachen.de/tabula>



événement que nous lui avons rappelé sur chaque porte

- Ferdinand te Walvaart a été désigné chef de projet du relations avec les clients qui nous permettra de mieux

PROFIL D'EXPERT

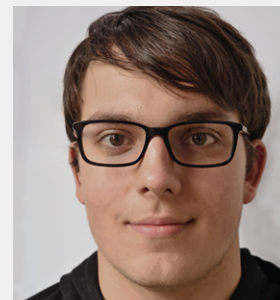
Elektor est au cœur d'un réseau de plus de 1 000 experts et auteurs engagés dans la publication de livres, d'articles, de DVD, de webinaires et autres événements. Coup de projecteur !

Nom : **Viacheslav Gromov**

Âge : **16 ans**

Études : **lycéen**

Projet en cours : **une unité de commande de serre à alimentation solaire, suivi d'un livre sur l'IdO**



Peux-tu te présenter, Viacheslav ?

J'ai 16 ans, je suis lycéen, mais aussi auteur indépendant spécialisé en électronique. Cela fait maintenant quelques années que j'écris des livres et des articles, principalement sur le numérique, les microcontrôleurs et les FPGA par exemple. Mais l'analogique m'intéresse aussi. D'ailleurs j'ai écrit un cours sur les μ C ARM pour Elektor.

Tu as commencé l'électronique très tôt ?

J'ai d'abord découvert la mécanique grâce à mes deux grands-pères. J'étais petit et nous habitions encore en Russie. Ce n'est que lorsque nous avons déménagé en Allemagne que je me suis intéressé à l'électronique. Près de chez nous, il y avait un magasin appelé *Wultschner Elektronik*. On pouvait m'y apercevoir plusieurs fois par jour, pour moi c'était le paradis ! Je pouvais y trouver tous les composants dont j'avais besoin. C'est à cette époque que j'ai commencé à acheter des livres d'électronique et à arpenter des salons.

Selon toi, d'où viendra la prochaine révolution de l'électronique ?

Je suis persuadé que les recherches sur les microsystèmes à base de semi-conducteurs organiques conduiront à une percée en matière de rendement. Idem pour les réseaux de neurones, qui sont bien plus avancés que les structures utilisées actuellement pour traiter l'information. Ils nous apporteront toutes sortes d'avantages, mais avant cela il faudra que nous trouvions une nouvelle façon de penser.

Qu'est-ce qui différencie l'Allemagne des États-Unis en matière d'innovation électronique ?

Malheureusement, il faut bien reconnaître que la recherche et l'industrie allemandes sont très en retard, que ce soit en électronique ou plus généralement en technologie. Pour ce qui est de l'électronique, les centres de gravité actuels sont bien connus : la Silicon Valley et l'Asie. Et je pense que ce n'est pas près de changer...

Si je t'offrais 500 € en bons d'achat pour l'e-choppe, que choisirais-tu et pourquoi ?

N'importe comment il y aurait des livres dans mon choix ! J'opterais aussi pour une carte FPGA, avec un choix situé quelque part entre une carte d'initiation et le Red Pitaya – puisque sa puce fournit un FPGA. ◀

(160030 – version française : Hervé Moreau)

hexadoku casse-tête pour elektorniciens

Dring dring ! Le téléphone sonne encore !
 Vous souhaiteriez être tranquille pendant que vous remplissez votre grille estivale d'hexadoku. Avez-vous pensé à vous inscrire sur la liste Bloctel (bloctel.gouv.fr) ? Depuis le 1^{er} juin, le nouveau système d'anti-démarchage téléphonique géré par Opposetel est entré en vigueur. Les sanctions envers les entreprises qui ne respectent pas cette nouvelle liste ont été alourdies. Le téléphone restera-t-il enfin silencieux ?

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 **trois** des réponses internationales correctes reçues dans les délais ; leurs auteurs recevront chacun un bon d'achat Elektor d'une valeur de **50 €**.
 À vos crayons !

Envoyez votre réponse (les chiffres sur fond grisé) avec vos coordonnées par courriel, avant le **31 août 2016** à l'adresse **hexadoku@elektor.fr**

Les gagnants

La solution de la grille du numéro de mai est **BD4FE**

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

René **Niel** (France), Kurt **Kuehni** (Suisse), Marianne **Meyers** (Luxembourg).

Bravo à tous les participants et félicitations aux gagnants !

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

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

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.

S'ABONNER MAINTENANT !

Newsletter

Recevez chaque semaine les
toutes dernières informations

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



TEMPÉRATURE
DETECTEUR DE PROXIMITE
ENREGISTREUR DE TRAJECTOIRE
DEBIT
ULTRASON
HUMIDITE
EFFET HALL
PRESSION
FORCE

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

Les langues de notre boutique:

Capteurs effet Hall

Capteurs Hall

SIEMENS



Boîtier : P-SSO-3-2

Número de commande	Magnetic-Range	
TLE 4905L 0,56	±17 mT	unipolaire
TLE 4935L 0,60	±20 mT	bipolaire / latch

Capteur radiométrique, linéaire

Honeywell

- Tension de fonctionnement : 4,5 ... 10,5 VDC
- Consommation de courant : 8,7 mA (@ 5 VDC)
- Température de fonctionnement : -40 ... +150 °C
- Linéarité : 1 % typ.
- Temps de réponse : 3 µs



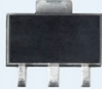
Boîtier : TO-92
Sortie : 0,2 V / 1,5 mA

Número de commande	Magnetic-Range	Sensibilité
SS 496 A1 2,06	±84 mT	2,5 mV/G
SS 495 A 1,18	±67 mT	3,125 mV/G
SS 495 A1 3,19	±84 mT	3,125 mV/G

Capteur radiométrique, linéaire, SMD

Honeywell

- Tension de fonctionnement : 2,7 ... 6,5 VDC
- Consommation de courant : 10 mA
- Température de fonctionnement : -40 ... +100 °C
- Linéarité : 1 % typ.
- Temps de réponse : 3 µs



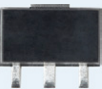
Boîtier : SOT-89
Sortie : 1,0 V / 1,5 mA

Número de commande	Magnetic-Range	Sensibilité
SS 59 ET 1,35	±65 mT	1,0 mV/G

Capteur effet Hall, numérique, SMD

Honeywell

Capteurs effet Hall numériques compensés en température



Unipolaire, boîtier : SOT-89
Sortie : 0,4 V / 20 mA

Número de commande	Magnetic-Range
SS 543 AT 1,89	7,5 ... 18,0 mT
SS 549 AT 2,51	23,5 ... 39,0 mT

Détecteur de proximité inductif

CONTRINEX

Excellents capteurs inductifs pour détecter sans contact les pièces métalliques dans le quotidien de l'automatisation.
Idéal pour une utilisation dans des conditions environnementales difficiles.



Extrait de notre gamme.
Vous trouverez la gamme complète à l'adresse suivante :

<http://rch.it/kj>



Número de commande	Ø x longueur	Distance de commutation	Pose
DWAD 509 M8 390 76,89	M8 x 45 mm	0 ... 4 mm	Câble de 2 m quasi-affleurant
DWAD 623 M5 40,25	M5 x 25 mm	1,5 mm	Câble de 2 m affleurant
DWAD 617 M12 32,52	M12 x 50 mm	4,0 mm	Câble de 2 m non affleurant
DWAS 603 M8 129 40,59	M8 x 29 mm	1,5 mm	Prise M8 3 pôles affleurant
DWAS 713 M8 001 63,45	M8 x 60 mm	6,0 mm	Prise M8 3 pôles non affleurant
DWAS 603 M12 28,07	M12 x 60 mm	2,0 mm	Prise M12 4 pôles affleurant
DWAS 503 M12 41,01	M12 x 60 mm	6,0 mm	Prise M12 4 pôles affleurant

Testeur capteur jusqu'à 100 mA, LED, Vibreur, micro-USB

CONTRINEX

- Contrôles de champs rapides différents types de capteurs
- Accu



Número de commande
ATE 0000 010 45,34

Capteur d'humidité 0 ... 100 % rF, TO 39

B+B SENSORS

Capteur d'humidité numérique avec interface IC dans le boîtier TO39 résistant à la pression (jusqu'à 16 bars), convient pour les mesures du point de rosée.



Número de commande
HYT 939 26,39 TO 39

Capteurs de pression, UsV 4,75-5,25

freemove

Número de commande	PR kPa	Sens mV/kPa	Lin %
MPX 5010DP 11,43	0 - 10	450	±5
MPX 5050DP 13,78	0 - 50	90	±2,5
MPX 5100DP 14,24	0 - 100	45	±2,5
MPX 5500DP 8,91	0 - 500	9	±2,5



Capteurs de distance SHARP

Très haute fiabilité et précision plus élevée que les capteurs classiques.



Typen: GP2Y0A

Número de commande	Portée (cm)	Long. x larg. x prof. (mm)
GP2-0215 5,63	20 ... 150	29,5 x 13,0 x 21,6
GP2-0430 4,29	4 ... 30	37,0 x 18,9 x 13,5
GP2-1080 4,66	10 ... 80	29,5 x 13,0 x 21,6

Número de commande	
DMC01-SC150 1,81	3 pôles Câble de données

Capteurs à ultrasons

Récepteur et émetteur céramique à ultrasons de 40 kHz



Ø 9,9 mm, H 7,1 mm
Número de commande

Número de commande	
MUS-40E 2,51	Récepteur
MUS-40S 2,51	Émetteur

Prix du jour ! Prix à la date du : 01.06.2016
Prix en € hors T.V.A., les frais de transports
reichelt elektronik, Elektronikring 1, 26452 Sande (D)

Modes de paiement internationaux:



Commander maintenant! www.reichelt.fr
Assistance téléphonique en anglais : +49 (0)4422 955-360

Ne faites plus de compromis !

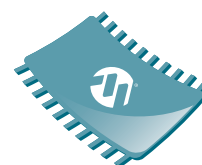
Ajoutez des fonctionnalités à votre système grâce aux microcontrôleurs PIC® 8 bits



Dans la vie réelle, la conception de systèmes embarqués demande de faire des compromis à chaque phase. Les ajustements entre performances, fonctionnalités et coûts sont souvent un frein pour commercialiser les meilleures idées. Nous croyons qu'il existe des solutions. C'est pourquoi nous avons doté nos derniers microcontrôleurs PIC® 8 bits de blocs matériels intelligents, flexibles et indépendants du cœur, qui réagissent rapidement, consomment très peu d'énergie et demandent bien moins de code qu'une approche logicielle. Les périphériques indépendants du cœur (CIP) sont conçus pour vous aider à combiner facilement de nombreuses fonctions système complexes dans un seul microcontrôleur, pour en augmenter la vitesse et la flexibilité tout en réduisant la consommation électrique et les coûts. Intégrez les microcontrôleurs PIC® 8 bits à vos systèmes, et vous n'aurez plus de compromis à faire.

Améliore les fonctions systèmes :

- Une flexibilité maximale
- Une latence minimale
- Réduction des coûts



**FLEXIBLE
INTELLIGENCE
MADE EASY**
8-BIT PIC® MICROCONTROLLERS

microchip
DIRECT
www.microchipdirect.com

 **MICROCHIP**

www.microchip.com/8-bit