

nouvelle formule !

mai / juin 2017 | n° 465

www.elektormagazine.fr

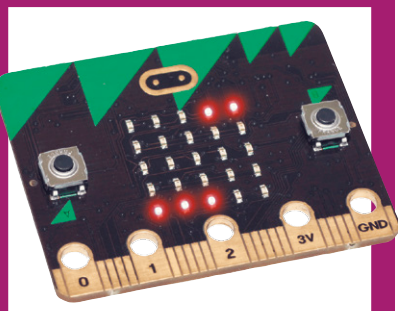
elektor

DÉCOUVRIR

CRÉER

PARTAGER

radiocommande XBee/nRF24



BBC micro:bit
pour électroniciens
premiers pas avec mbed



cartes et kits
de développement
pour l'Internet des Objets



préamplificateur
pour microphones
professionnels

bidirectionnelle à 4 voies

prototypage Bluetooth : module PSoC BLE

**au format L-board • Phantastron • analyseur de
signal par Ethernet : nouveaux logiciels •**

Sigfox : donner vie aux objets • éclairage extérieur

variable (lampes à LED de 12 V) • transformateur de Tesla •

simulateur de présence • le bus I²C • chargeur d'accus Li-ion •

**utilisation de l'eC-stencil-fix • hors-circuits de R. Lacoste : corrélation
numérique • Android sur Raspberry Pi (2) • Rétronique : PET fête**

ses 40 ans • WaterDeck : analyse de la consommation d'eau • loi sur les

robots ? • passerelle IoT et nœuds sans fil (2)

Dans cette édition :
5 projets du labo
6 projets de lecteur
Bluetooth, L-board,
SPI, Li-ion, I²C,
XBee ...

ISSN 0181-7450

(B) 16,50 € • (CH) 29,00 FS • (CAN) 20,99 \$ca • (And) 15,50 €
DOM surface 16,50 € • DOM avion 16,90 € • N Cal/S 2000 cfp

L 19624 - 465 - F: 15,50 € - RD



36 Expériences de Physique avec Arduino + Kit de démarrage



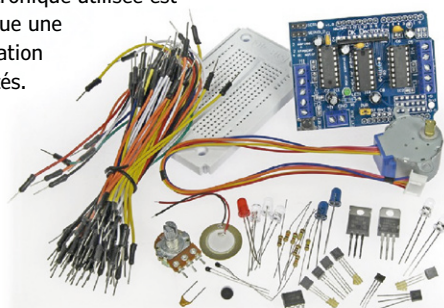
La rencontre de la physique et du microcontrôleur ne devrait plus étonner personne. Il existe d'excellents enregistreurs de données, ainsi que de nombreux programmes pour les traiter et les présenter sous forme de graphiques colorés et attrayants. La physique rébarbative, c'est fini !

J'ai choisi l'Arduino, car cette plate-forme est d'un accès facile et sa documentation abondante. La famille Arduino offre des ressources extraordinaires à un prix dérisoire. Ajoutez-y le logiciel gratuit CoolTerm, et vous pouvez enregistrer toutes les données de mesure pour les retravailler sous Excel et créer aisément des tableaux ou des graphiques.

Ce livre n'est pas un manuel de physique. Vous n'y trouverez ni équations différentielles ni courbes abstraites. Nous étudierons des phénomènes physiques de la vie de tous les jours. Sans chercher à être exhaustif, mon modeste ouvrage apporte aux leçons de physique une approche nouvelle et créative grâce aux techniques modernes de mesure et de traitement des données. L'électronique utilisée est simple, et constitue une belle démonstration des possibilités.

Un kit a été spécialement conçu pour réaliser les expériences décrites dans ce livre.

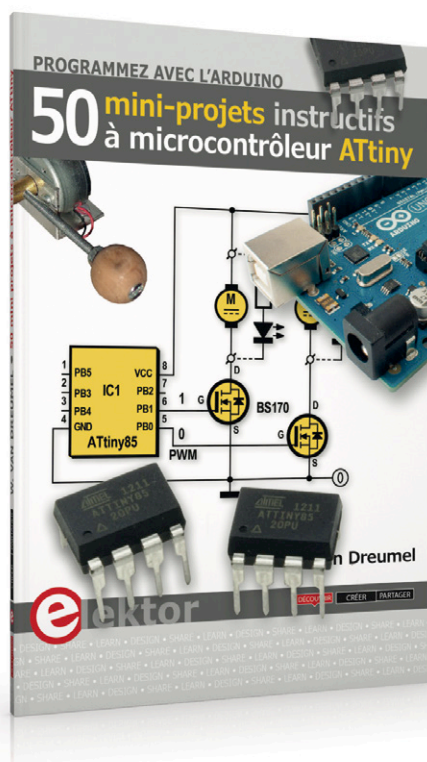
Contenu détaillé du kit, voir www.elektor.fr/159009-71



www.elektor.fr/arduino-36-experiences

Programmez avec l'Arduino

50 mini-projets instructifs à microcontrôleur ATtiny



Pour tester et mettre au point de petits montages à microcontrôleur, la carte Arduino est parfaite. Ensuite, on peut obtenir le même résultat avec un autre μC , plus petit et moins cher, comme l'octopède ATtiny. Ce livre pour les débutants montre comment faire à l'aide de 50 applications originales, instructives et astucieuses. La plupart des projets sont décrits en détail.

Dans ces pages, ni théorie ni mathématiques, que des considérations électroniques pratiques : un livre qui se lit le fer à souder à la main !

L'ATtiny est programmé avec une carte Arduino ; les croquis disponibles sur cette plateforme sont facilement transposables pour l'ATtiny, le livre explique comment.

ISBN 978-2-86661-204-7

www.elektor.fr/50-mini-projets-attiny-arduino

40^{ème} année,
n° 465 – mai/juin 2017

ISSN 0181-7450
Dépôt légal : avril 2017
CPPAP 1113 U 83713
Directeur de la publication : Donatus Akkermans

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

@ : service@elektor.fr

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

Fax : (+33) 01.49.19.22.37

www.elektor.fr | www.elektormagazine.fr

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

Publicité :

Margriet Debeij
Tél. : 02 41/955 09 174
Fax : 02 41/955 09 013
Gsm : (+49) 170 550 539 6
margriet.debeij@eimworld.com

DROITS D'AUTEUR :

© 2017 Elektor International Media B.V.

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

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

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



Géolocalisation pour tous

Un rhinocéros abattu au zoo de Thoiry, sa corne découpée à la tronçonneuse !

En mars dernier, la France qui était déjà une zone de transit pour le trafic de cornes de rhinocéros est devenue également une zone de braconnage. Quel rapport avec l'électronique ? Faut-il incriminer le système d'alarme et les caméras de surveillance du zoo qui n'ont pas empêché ce vol funeste ?

Pas du tout ! La même semaine, la Fondation Sigfox a présenté l'expérimentation d'un système de suivi à distance des rhinocéros en Afrique pour mieux les protéger : projet « Now Rhinos Speak ». Il semble que désormais la préservation des espèces menacées est nécessaire partout dans le monde.

Pour mémoire, le réseau Sigfox a déployé des antennes sur trois continents, dans une trentaine de pays. Il écoute et relaye les messages envoyés par des objets connectés estampillés Sigfox. Bien sûr, vous connaissez les balises Argos bientôt quarantennaires qui permettent aux communautés scientifiques de collecter des données géolocalisées et d'œuvrer à la sauvegarde de l'environnement et des espèces. Donc rien de nouveau sous le soleil africain !

À un détail près : les réseaux à bas débit comme Sigfox, LoRa et consorts sont accessibles à tous. Pas besoin d'être un institut de recherche ou une association de protection de l'environnement pour y entrer, comme dans le réseau Argos. Un électronicien peut facilement créer des balises personnalisées et souscrire un abonnement pour veiller sur son propre écosystème. Quand vous aurez mis en place la surveillance de votre cheptel de moutons, loups ou rhinos, n'hésitez pas à venir partager votre expérience avec nous.

Mariline Thiebaut-Brodier

PS La fondation Sigfox cherche des donateurs pour produire les 29.000 capteurs à installer dans les oreilles des rhinocéros. Le capteur permettrait de mesurer chaque signe du comportement de l'animal : mouvement, signes de stress, rythme cardiaque, etc.

Notre équipe

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

118

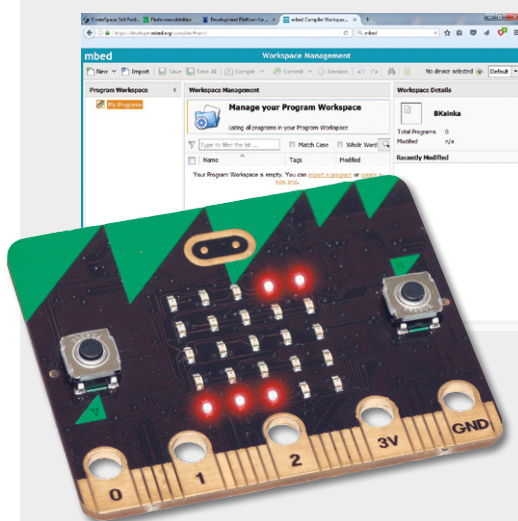
- 5 bientôt dans Elektor
- 13 Elektor : guide de connexion(s)
- 63 vol tous azimuts
pilote par Clemens Valens
- 66 le Phantastron
drôle de composant n°30
- 88 projet 2.0
corrections, mises à jour
et courrier des lecteurs
- 90 Rétronique
PET fête ses 40 ans, bon anniversaire !
- 94 Q & R
(presque) tout ce que vous avez toujours
voulu savoir sur... la différence entre
immothèque et domothèque
- 126 questions d'éthique
avons-nous besoin d'une loi sur les robots ?
- 128 l'e-choppe d'Elektor
- 130 Hexadoku - casse-tête pour elektorniciens
- 131 agenda - mai-juin 2017

en coulisse

- 6 tremplins vers l'IdO
cartes et kits de développement
pour l'Internet des Objets
- 26 trucs et astuces
panne de fer à souder *gull-wing*
- 40 microscope USB Andonstar V160
indispensable pour travailler avec les CMS
- 42 le bus I²C
1^{re} partie : le protocole
- 82 Android sur Raspberry Pi (2)
afficheur piloté via le bus SPI
- 99 Sigfox
donner vie aux objets

BBC micro:bit pour électroniciens premiers pas avec mbed

20



La carte BBC micro:bit ne convient pas seulement aux étudiants, c'est aussi un superbe microcontrôleur pour les électroniciens. On dispose sous un faible volume de presque tout ce que l'on a régulièrement besoin : entrées/sorties numériques avec MLI, convertisseur A/N ainsi qu'une interface USB pour l'alimentation, la programmation et l'échange de données. Divers capteurs et Bluetooth complètent le tout.

- 104 expérimenter avec
un transformateur de Tesla
quelques composants
pour un effet électrisant
- 112 hors-circuits de R. Lacoste
corrélation numérique :
détection de motif et de période
- 116 utilisation de l'eC-stencil-fix
pour l'e-BoB avec capteur BME280 et
d'autres cartes

6

cartes et
kits de développement
pour
l'Internet des Objets



chargeur d'accus Li-ion

à courbes caractéristiques correctes

- Charge toutes sortes d'accus Li-ion de tension nominale de 3,6 V.
- Mode « formatage » pour essayer de sauver des cellules ayant subi une décharge profonde.
- Mode « décharge » pour déterminer la capacité réelle de l'accum.
- Toutes les informations pertinentes affichées sur l'écran LCD.

préamplificateur simple pour microphones professionnels

28

Il y a belle lurette que les enregistreurs à bande sont tombés en désuétude et ont été remplacés par des ordinateurs portables équipés d'une « carte son USB » de bonne qualité. Alors que l'entrée « ligne » d'un PC présente généralement des caractéristiques acceptables, l'entrée « micro » se caractérise par de médiocres niveaux de bruit et de distorsion. Surtout, elle ne restitue qu'un signal monophonique et n'est pas adaptée à la connexion de microphones professionnels à sortie symétrique, qui souvent nécessitent une alimentation « fantôme » de 48 V. Je vous propose ici un préamplificateur qui permet de contourner ces deux écueils et ravira sûrement les possesseurs de cartes son « haut de gamme ».



sur la scène : les projets

- 14 module PSoC BLE au format L-board adapté aux platines d'essais**
pour un prototypage Bluetooth tout confort
- 20 BBC micro:bit pour électroniciens**
premiers pas avec mbed
- 28 préamplificateur simple**
pour microphones professionnels

- 46 retour sur l'analyseur de signal par Ethernet**
plus rapide, toujours plus facile à utiliser, et quelques nouveautés
- 54 radiocommande bidirectionnelle à 4 voies**
avec modules radio XBee ou nRF24
- 68 passerelle IoT et nœuds sans fil**
2^e partie : le logiciel
- 74 éclairage extérieur variable**
réglage de la luminosité de lampes à LED de 12 V
- 79 simulateur de présence**
la crainte des monte-en-l'air
- 96 WaterDeck**
analyse de la consommation d'eau des villes intelligentes
- 106 capteurs (4)**
pour Arduino et Cie
- 118 chargeur d'accus Li-ion**
à courbes caractéristiques correctes



bientôt sur ces pages

Extrait du sommaire du prochain numéro :

- Afficheurs à 7 segments pour tableau de scores
- Espion pour RFM12
- Générateur d'images de test
- LoRa avec cartes ST
- Raspberry Pi : CN/A audio
- Raspberry Pi : mise à jour de l'analyseur OBD
- Simulateur du bruit de l'océan
- Traceur de courbe pour tube à vide
- Variateur pour LED
- Etc.

Sous réserve de modification.

Le numéro de juillet-août 2017 paraîtra le 21 juin 2017.

tremplins vers l'IoT

cartes et
kits de développement
pour
l'Internet des Objets

Viacheslav Gromov (Allemagne), readers@gromov.de





Notre lectorat semble friand d'Internet des Objets (IdO), spécialement de tout ce qui touche à la domotique. Auparavant pour se lancer dans de tels projets, il fallait composer soi-même un système de développement et de prototypage rapide. Aujourd'hui, toute la branche « embarquée » propose des kits de développement incroyablement peu onéreux et pourtant riches en matériel indispensable, pour qu'en quelques minutes vous puissiez réaliser votre prototype, même avec peu d'expérience.

Les kits de développement pour l'IdO sont maintenant très complets : du câble USB en passant par la carte truffée de capteurs jusqu'à l'accumulateur et au boîtier nécessaires. Peut-être est-ce en raison de cette énorme diversité que l'on s'y perd vite dans les choix à faire. Quel processeur adopter ? Quels capteurs ? Quel système radio ? Comment l'alimenter ? Et puis combien cela va-t-il me coûter, combien de temps y passer, comment économiser l'énergie ? Et plein d'autres questions. Le marché est rempli à ras bord, pour ne pas dire saturé, de plateformes pour toutes les technologies existantes et tous les champs d'application.

Ce survol n'abordera pas les célèbres plateformes IdO plus ou moins professionnelles comme micro:bit, certaines cartes

d'Arduino, les nombreuses cartes ESP8266 et bien d'autres encore, pour mieux se concentrer sur les solutions des grands fabricants de semi-conducteurs, c'est quand même de là que viennent les innovations les plus rapides des technologies IdO. Sans vouloir être exhaustives, ces quelques pages devraient vous montrer par coups de sonde quelques produits phares de l'IdO et les caractéristiques du matériel comme du logiciel ; nous irons voir les processeurs, en passant par les capteurs jusqu'aux modules radio, flamboyants neufs ou déjà éprouvés, célèbres ou moins connus, bon marché ou chers, universels ou spécifiques, de quoi permettre à chacun d'y trouver son bonheur !

Hexiwear de MikroElektronika et NXP

NXP (ex Freescale, avant de devenir Qualcomm - Californie) a certainement aussi ses propres cartes de développement pour l'IdO, mais ses coopérations avec d'autres fabricants et distributeurs présentent encore plus d'intérêt. C'est ainsi qu'a pris son envol **Hexiwear** (figure 1) [1]. Alors que la plateforme est produite et commercialisée par MikroElektronika, les microcontrôleurs et la plupart des capteurs viennent de NXP. En outre, NXP stimule à fond cette plateforme, notamment en adaptant son environnement de développement Kinetis Design Studio. Mais, c'est quoi, au juste Hexiwear ? C'est un grand gadget en forme de bracelet-montre à assembler avec différents boîtiers et bracelets, à porter sur soi ou à garder indépendant, en l'accompagnant des nombreuses extensions possibles pour divers usages. La carte Hexiwear est ainsi très diversifiée, on peut lui adjoindre le BLE, l'USB, de nombreux capteurs (pouls, lumière,

température, pression atmosphérique, hygrométrie, accélération, gyroscope et champ magnétique), un écran couleur OLED de 28 mm et un accumulateur dont les 190 mAh permettent de tenir en usage régulier pendant plusieurs heures. La manipulation se fait par six boutons tactiles à l'avant, mais au besoin, on peut encore l'intensifier avec le moteur vibreur embarqué pour une rétroaction. Il y a encore à l'avant une LED RVB programmable. La carte fait travailler deux microcontrôleurs : le K64F bien connu (M4F, 120 MHz, 1 Mo de flash), il sert de processeur principal au centre de la carte et commande tous les périphériques, ainsi que le KW40Z (M0+, 48 MHz, 160 Ko) de la série radio à double mode (BLE + IEEE 802.15.4) qui assure la liaison BLE (**B**luetooth **L**ow **E**nergy) sur Hexiwear. La carte dispose d'un connecteur à 50 broches sur la face inférieure, en particulier pour la station d'accueil disponible comme accessoire. Cette station est équipée de trois boutons-poussoirs libres, d'un autre pour la RàZ des deux microcontrôleurs, de LED, d'un socle pour carte micro-SD, de connecteurs pour I²S, micro-USB et SWD. Au milieu de cette carte d'extension se trouvent les trois connecteurs pour y glisser les cartes populaires Click au format MikroElektronika. Il y a des centaines de ces cartes avec divers capteurs, actionneurs et interfaces. Avec Hexiwear et les cartes Click, vous terminez votre prototype en quelques minutes.

Un micrologiciel chargé en usine fonctionne sur Hexiwear pour faire la démo de toutes les fonctions des capteurs. En outre, on peut relier la carte Hexiwear par Bluetooth avec l'appli *Hexiwear* de WolkAbout sur un ordiphone, sur lequel on peut suivre en direct les données transmises au nuage WolkAbout et finalement les visualiser sous différents aspects. Vous pouvez aussi, sans fil, mettre à jour le micrologiciel par OTAP (*Over-The-Air-Programming*).

Le plus simple pour la programmation de Hexiwear est de passer par l'interface micro-USB, un débogueur OpenSDA fait alors passer le code vers le microcontrôleur. Il existe trois variantes [2] pour rédiger son propre logiciel pour Hexiwear : pour commencer, on

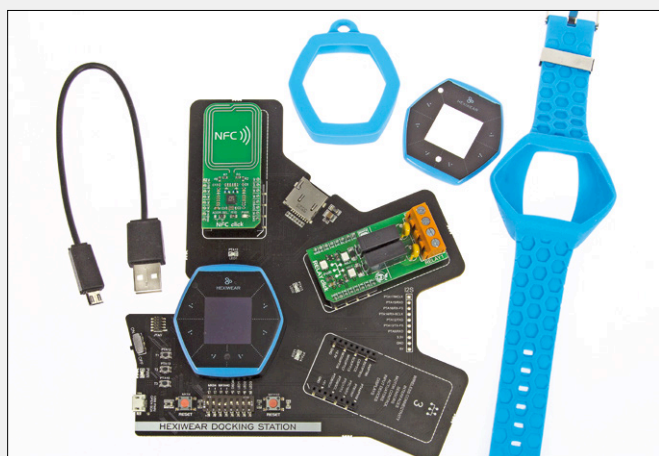


Figure 1. Voici Hexiwear sur sa station d'accueil avec deux des cartes d'extension Click disponibles. Il y a d'autres accessoires comme le câble USB ou des boîtiers de différentes couleurs.

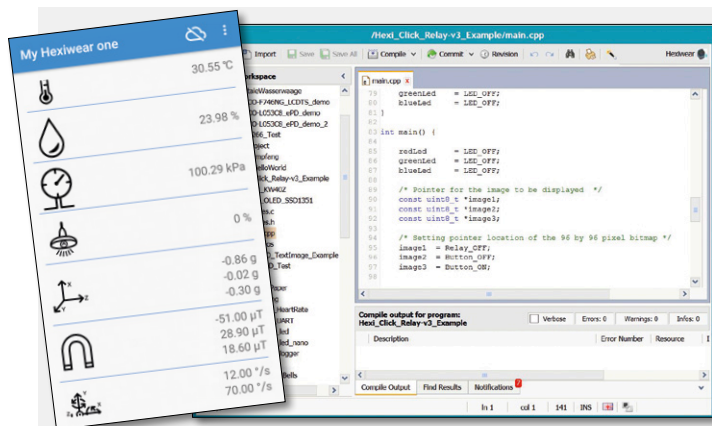


Figure 2. À gauche, l'appli Hexiwear avec les données actuelles ; à droite, un fragment de code de la page mbed, dans laquelle on reconnaît bien la simplicité et la structure des commandes.

peut utiliser l'environnement en ligne mbed (figure 2), les adeptes de Linux disposent de la distribution Zephyr et si vous voulez utiliser les options matérielles, vous choisirez KDS (Studio Kinetis Design). On peut aussi chercher l'inspiration sur Hackster.io qui présente de nombreux projets d'IdO avec Hexiwear. La documentation (intégralement à source ouverte !) et l'assistance sont dignes d'éloges. Le prix du seul Hexiwear avoisine les 50 €, avec la station d'accueil, le boîtier et les accessoires, environ le double.

À propos de NXP, il ne faut pas oublier la petite carte **WaRP7** (figure 3) qui a vu le jour en coopération avec Farnell/Element 14 [3]. Elle est livrée avec accumulateur de 180 mAh et dispose d'USB, NFC, BLE et WLAN comme interfaces sans fil, d'une caméra embarquée et de nombreux capteurs. Son processeur i.MX7 Solo (A7 + M4) est entouré de quelques puces de mémoire. Elle accepte aussi les cartes Click. En complément, on peut lui raccorder un écran tactile rond en couleurs. On peut recommander cette plateforme aux développeurs qui souhaitent travailler par lignes de commande et avec des distributions Linux dont ils ont une connaissance approfondie. Autre solution : Android comme système d'exploitation. Le prix est d'environ 90 €.

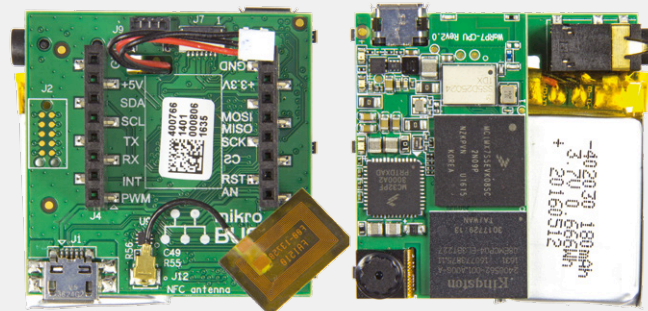


Figure 3. La carte WaRP7 sous toutes ses faces.

SensorTile de STMicroelectronics

STMicroelectronics (ST) a depuis peu commercialisé un kit très intéressant du nom de **SensorTile** [4] dont le prix avoisine 80 € (figure 4). Intéressons-nous au SensorTile de la taille d'une pièce de monnaie, basé sur le microcontrôleur STM32L476 (M4, 80 MHz, 1 Mo de flash) et le BlueNRG-MS pour la communication BLE. Il dispose d'un microphone, d'un baromètre et d'un accéléromètre, d'un gyroscope et d'un magnétomètre. Les connexions

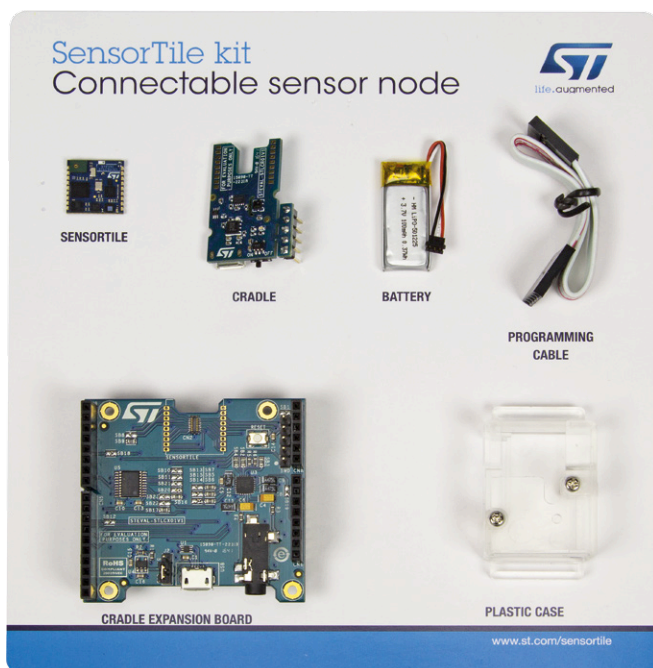


Figure 4. Le kit SensorTile avec tous ses composants.

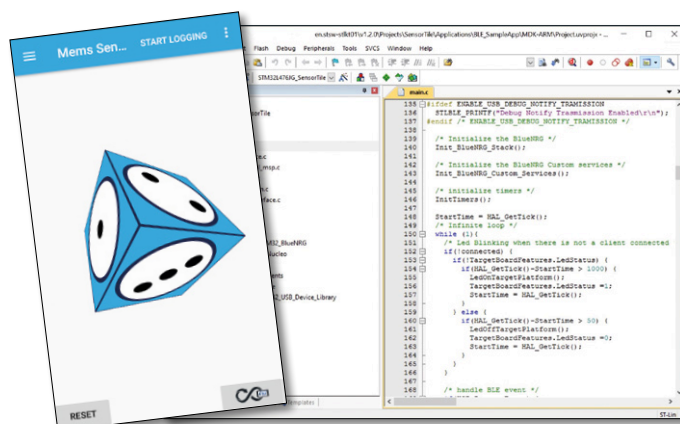


Figure 5. À gauche, un exemple d'accéléromètre avec l'appli pour ordiphone, à droite, un fragment du modèle d'exemple de démarrage dans l'environnement de Keil.

principales sont sur des pastilles à souder le long des bords, mais aussi sur le petit connecteur situé sous la carte.

Puisqu'il faut bien alimenter le SensorTile, deux cartes d'adaptations sont prévues dans le kit. L'une lui sert de berceau (*cradle*) sur lequel le souder et lui octroie en supplément des capteurs d'humidité et de température, un socle pour carte micro-USB, un connecteur de programmation SWD ainsi qu'un connecteur micro-USB pour l'alimentation. Le kit comporte aussi un accumulateur de 100 mAh à brancher sur le berceau pour fournir du courant au SensorTile. Et le boîtier transparent pour tout l'assemblage est aussi compris.

L'autre possibilité d'adaptation est constituée d'un *shield* Arduino avec sortie audio DAC (numérique vers analogique) pour jack. On y trouve aussi un connecteur micro-USB et un SWD. Ce *shield* peut travailler indépendamment avec le SensorTile, mais

aussi avec une carte Nucleo ou une autre de la série Arduino de manière à ce que le SensorTile soit piloté par un autre microcontrôleur (et inversement).

Les deux cartes d'adaptation sont équipées du connecteur SWD par lequel on programme le SensorTile, de préférence avec un ST-Link, que l'on trouve aussi sur toute carte Nucleo à 64 broches. Il y a encore dans le kit un câble de pontage. ST a aussi développé une appli du nom de ST BlueMS qui, dès le déballage, fait la démonstration des capacités du SensorTile (**figure 5**). Pour plusieurs exemples sur des capteurs, il faut cependant demander

une licence, ce qui se fait automatiquement par courriel et ne prend pas beaucoup de temps. L'appli permet aussi de charger des données dans le nuage Watson IoT d'IBM.

On peut rédiger le logiciel avec l'habituel compilateur STM32. Cela peut mener au modèle un peu plus complexe BlueMicrosystem2 que montre l'exemple de programme qui tourne à la sortie d'usine sur le SensorTile. Ou bien, on utilise le modèle aux fonctions de base simplifiées *Starter Firmware* pour lequel il existe trois programmes de mise en train.

Kit IdO solaire de Cypress Semiconductor

Au premier abord, le kit [Solar-Powered IoT Device](#) [7] pour à peine 50 €, basé sur le module BLE de Cypress, ne se démarque pas des innombrables solutions IdO. Le module BLE de Cypress est vraiment bien et l'environnement de développement PSoC Creator lui convient parfaitement dans ce genre d'applications, mais tout l'intérêt provient de la cellule solaire sur la carte destinée au glanage d'énergie ! Parce que les tendances s'influencent en sens contraire : l'Internet des Objets exhorte à l'optimisation de la puissance et cela débouche parfois sur le glanage d'énergie, quand c'est possible et souhaitable. On ne peut pas négliger la récolte d'énergie en IdO, c'est même un atout.

Ce kit montre les capacités du module EZ-BLE PRO (CYBLE-022001-00, M0, 48 MHz, 256 Ko de flash). Accompagnent ce

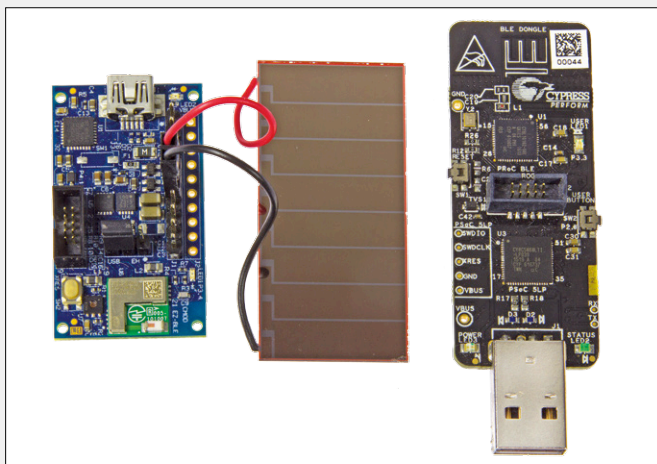


Figure 8. On retrouve ici les deux cartes décrites ainsi que la cellule solaire du kit.

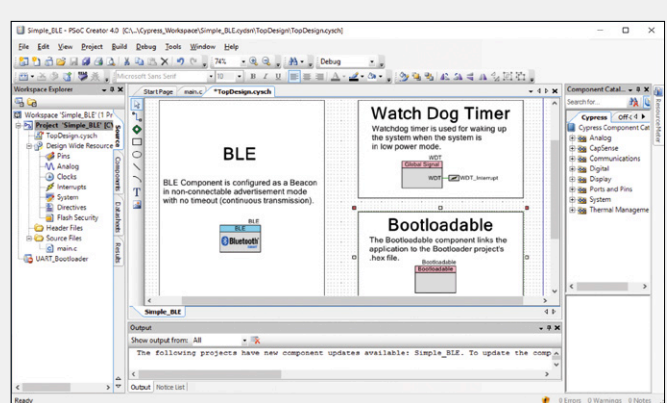


Figure 9. La philosophie LEGO du Creator de PSoC simplifie aussi la programmation de la liaison BLE, tout y est configuré d'avance automatiquement.

module sur la carte, le circuit de glanage d'énergie et quelques capteurs pour la température, l'humidité et la lumière, mais on peut y brancher d'autres capteurs. Une carte USB BLE est jointe au lot pour communiquer avec la carte de glanage d'énergie (**figure 8**). Ainsi, en supplément de l'habituel flux de données, on peut déterminer la portée et la force du signal, pour maximiser la récolte dans différents environnements (en plus du solaire, la piézoélectricité par exemple).

On peut programmer le processeur de la carte de récolte au moyen d'un débogueur externe, non compris dans le kit, de préférence un MiniProg 3 sur le connecteur SWD, avec le Creator de PSoC [8], modulaire et assez simple (**figure 9**). Il y a encore dans le kit quelques composants à fils importants pour la mise en œuvre des projets d'exemple.

SimpleLink SensorTag de Texas Instruments

Vous pouvez acheter pour environ 30 € l'un des [SensorTag](#) de TI (**figure 10**) [9]. Au déballage, vous trouverez une balise de la grandeur d'une boîte d'allumettes avec un boîtier en caoutchouc. Cela peut sembler inutile et contrariant, mais à y regarder de plus près, on découvre tout ce qu'il y a dessous. Après l'installation de l'appli *SensorTag* de TI (**figure 11**), tout devient clair : on peut interroger en direct tous les capteurs des balises par BLE ou WLAN (selon la balise) et transférer ces données dans le nuage IBM Watson IoT pour une analyse.

L'une des nouvelles balises est pilotée par un microcontrôleur CC1350 Dual Mode (M3, 48 MHz, 128 Ko) qui communique en BLE et, en outre, dans la bande de 1 GHz sur des distances jusqu'à 2 km ! De l'extérieur, on ne voit que quelques boutons, mais derrière les étroites ouvertures dans le boîtier intérieur en plastique se cachent bon nombre de capteurs : un microphone, des capteurs de lumière, d'humidité, de pression d'air et de température (normal et infrarouge) et enfin un accéléromètre, un gyroscope et un magnétomètre. Il y a aussi une LED utilisateur. La petite pile bouton au dos

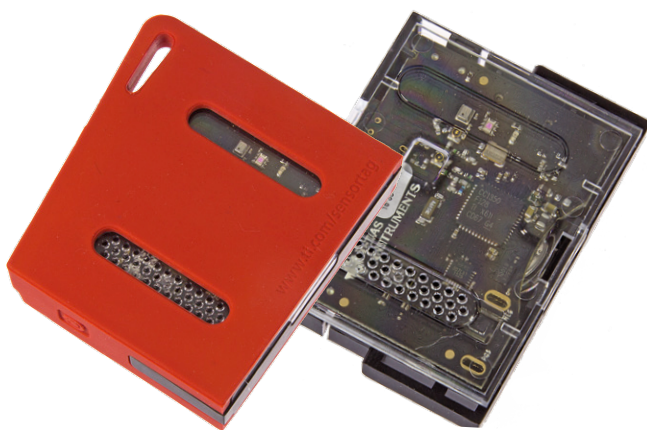


Figure 10. C'est sous le boîtier que se cache le SensorTag CC1350.

devrait, en raison du concept à basse puissance, durer un an en usage intermittent.

Ce serait bien si la balise sortait toute prête de l'usine, mais ce n'est évidemment pas le cas. Par le connecteur JTAG accessible de l'extérieur, on peut programmer la balise avec le Studio Code Composer (CCS, version compatible radio) de TI [10]. Comme le matériel est déjà prêt, on ne s'occupe que du logiciel dans la phase de prototypage. On dispose pour cela de nombreuses bibliothèques et d'une documentation complète, même pour la programmation de l'appli pour ordiphone.

Il existe encore une série d'outils logiciels supplémentaires, comme le Studio Sensor Controller, qui configure (d'avance) les applications de capteur pour microcontrôleur. Les SensorTags acceptent d'ailleurs aussi une programmation sans fil (OTAP) :

on peut par exemple dans cette application effectuer la mise à jour du micrologiciel de la balise par BLE. En outre, il existe depuis peu la possibilité de raccorder sur le connecteur de la balise à 20 contacts des cartes d'extension, des DevPack pour avoir un écran ou une LED RVB. L'offre de ces DevPack ne cesse de croître. Tout est fort à l'étroit dans ces emballages en caoutchouc, aussi pratique-t-on une incision à la bonne place, d'une main ferme et avec un couteau acéré, ce petit travail manuel est toujours récompensé.

Particulièrement remarquable, le débogueur à 15 € DevPack permet de programmer confortablement la balise par interface USB, ce qui rend superflus les gros débogueurs bien chers. Un excellent rapport prix/efficacité. À côté du CC1350, il y a encore le CC2650 avec BLE et ZigBee ou 6LoWPAN (déjà mentionné comme IEEE 802.15.4). Le fabricant sortira prochainement une balise WLAN.

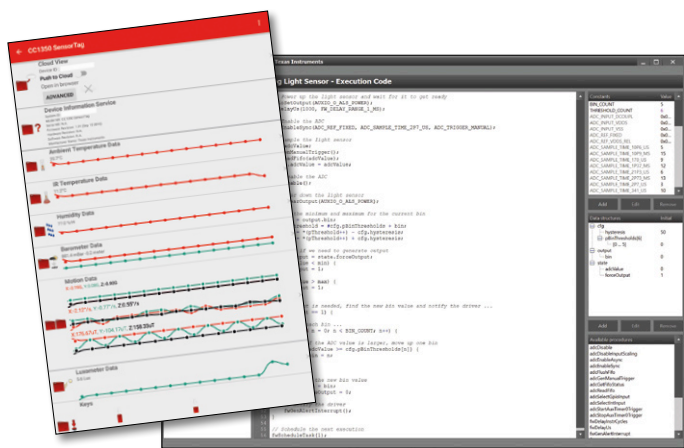


Figure 11. À gauche, l'interface de l'appli SensorTag, à droite, une fenêtre de Studio Sensor Controller.

XDK de Bosch Connected Devices and Solutions

Le **XDK110** (figure 6) de Bosch est là depuis longtemps, pourtant il reste dans le coup [5]. Pour la grande qualité du boîtier, des accessoires, la présence en ligne et la carte elle-même, on



Figure 6. Les principaux composants du kit XDK110 ; un support (mural) est aussi disponible. La photo montre bien l'étonnante petitesse de l'objet.

paie près de 200 €. Bosch est mondialement connu pour ses magnifiques capteurs, dont bon nombre sont installés sur XDK. Il est logé dans un boîtier en plastique avec quatre LED, deux boutons et il y a pour les capteurs une ouverture transparente (champ magnétique, accélération, inertie, gyroscope, lumière, température, hygrométrie, pression atmosphérique et audio). Sur le côté, on voit un socle pour carte micro-SD, un connecteur de programmation et une prise creuse à 26 contacts pour le câble de liaison vers un T-Board (les deux sont joints à la livraison). Ce T-Board (XDK Gateway, passerelle XDK), on peut aussi bien l'insérer dans une plaque d'expérimentation pour se faire rapidement un prototype. Le XDK s'occupe ainsi de tâches de contrôle et de surveillance plus complexes en dehors de son boîtier : de par son microcontrôleur ARM Cortex M3 avec 1 Mo de mémoire flash, il dispose d'une puissance de calcul suffisante. Si la mémoire de données ne suffit pas, il peut les enregistrer sur la carte micro-SD. L'accumulateur de 560 mA lui autorise les applications mobiles pour un bon bout de temps. On le programme avec le Workbench XDK [6] gratuit. On y trouve de nombreux exemples de programme et plusieurs notices explicatives progressives. La carte est équipée de BLE et WAN pour la communication avec un ordiphone (cf. l'exemple de programme VirtualXDK avec l'appli de la figure 7) ainsi qu'avec les autres appareils IdO. Et on peut en même temps accéder à l'internet.

D'autres producteurs de logiciel comme Relayr apportent

leur aide à XDK avec leurs propres solutions (nuage, analyse, etc.).

La programmation s'opère normalement au moyen du chargeur d'amorçage programmé en usine et l'interface micro-USB, vous ne devrez vous servir d'un débogueur externe branché sur les connexions de programmation que dans des cas exceptionnels.


En conclusion, on peut affirmer que quand on recherche une plateforme professionnelle, bien pensée et assistée industriellement, ce n'est certainement pas le prix élevé de XDK qui fera obstacle, bien au contraire ! 



Figure 7. À gauche, la vue d'ensemble de l'appli VirtualXDK. Le bureau XDK basé sur Eclipse, à droite, est familier aux développeurs avertis qui ne devront plus apprendre que la programmation et les relations avec le chargeur d'amorçage. Rien de pénible donc.

(160343 – version française : Robert Grignard)

Qu'y a-t-il d'autre ?

AT88CKECC-AWS-XSTK de Microchip, http://goo.gl/FPXVoi	Ce kit modulaire à plus de 200 € montre une des plus récentes techniques de sécurité pour l'IdO de Microchip en collaboration avec les services internet d'Amazon (AWS).
P-NUCLEO-LRWAN1 de ST, http://goo.gl/VLyLCT	Ce kit à environ 40 € est une carte Nucleo L073RZ avec <i>shield</i> LoRa, ce qui constitue la nouvelle tendance et rend plus accessible le marché de l'IdO.
P-NUCLEO-USB001 de ST, http://goo.gl/wjDplk	USB-C est une interface typique (aussi pour le chargement !) sur les appareils IdO modernes qui, pour environ 50 €, montre les possibilités de la carte Nucleo F072RB avec une carte d'extension pour fournir la puissance.
Kit IdO C027 de u-blox basé sur mbed, http://goo.gl/QSz6Oh	Cette carte IdO vaut, selon les modèles, à peu près 100 €, et met l'accent sur la localisation GPS et la réception UMTS/GSM, souvent indispensables sur les appareils évolués.
Kit Thunderboard React/Sense de Silicon Labs, http://goo.gl/dfGhfy	Ces deux cartes valent environ 30 € avec de nombreux capteurs, et d'autres périphériques offrent, selon leur type, des techniques radio différentes pour un bon rapport prix/puissance.
NuMaker Uni de Nuvoton, http://goo.gl/Nm9XkG	Dans le segment de prix inférieur, ce fabricant chinois propose quelques cartes IdO de la série NuMaker, ici aussi avec capteurs, Bluetooth et WLAN pour moins de 30 €.
Kit de prototypage rapide IdO Synergy S3A7 de Renesas, http://goo.gl/ov4UsG	Ce kit modulaire IdO pour un bon 150 €, avec son assez grand écran peut, sur base du concept « bac à sable » être combiné avec de nombreux capteurs différents et des modules radio.

... et bien d'autres encore !

Liens

- [1] www.hexiwear.com
- [2] www.hexiwear.com/getting-started/
- [3] www.element14.com/community/docs/DOC-79058/l/warp7-the-next-generation-iot-and-wearable-development-platform
- [4] www.st.com/sensortile
- [5] <http://xdk.bosch-connectivity.com/>
- [6] <http://xdk.bosch-connectivity.com/software-downloads>
- [7] www.cypress.com/documentation/development-kitsboards/s6sae101a00sa1002-solar-powered-iot-device-kit
- [8] www.cypress.com/products/psoc-creator-integrated-design-environment-ide
- [9] www.ti.com/ww/en/wireless_connectivity/sensortag2015/
- [10] www.ti.com/tool/ccstudio-wcs

la communauté d'Elektor

DÉCOUVRIR CRÉER PARTAGER

82

pays

248153

membres actifs

1040

experts & auteurs

489

publications

235332

visiteurs (par mois)

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



Magazine Elektor : six numéros par an d'électronique inédite (projets, bancs 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.elektorpcbsevice.com



Elektor GRATUIT, version numérique : 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



Livres & DVD Elektor : 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

- ✓ accès à l'archive d'Elektor
- ✓ 10% de remise dans l'e-shoppe
- ✗ 6x magazine imprimé
- ✓ 6x magazine numérique
- ✓ des offres exclusives
- ✓ accès à plus de 1 000 fichiers Gerber
- ✗ le DVD annuel d'Elektor

connectez-vous

www.elektor.fr/formule-greencard

formule

GOLD 127,50 € par an

- ✓ accès à l'archive d'Elektor
- ✓ 10% de remise dans l'e-shoppe
- ✓ 6x magazine imprimé
- ✓ 6x magazine numérique
- ✓ des offres exclusives
- ✓ accès à plus de 1 000 fichiers Gerber
- ✓ le DVD annuel d'Elektor

connectez-vous

www.elektor.fr/formule-goldcard

formule

GRATUITE

- ✗ accès à l'archive d'Elektor
- ✗ 10% de remise dans l'e-shoppe
- ✗ 6x magazine imprimé
- ✗ 6x magazine numérique
- ✓ des offres exclusives
- ✗ accès à plus de 1 000 fichiers Gerber
- ✗ le DVD annuel d'Elektor

connectez-vous

www.elektor.fr/inscription



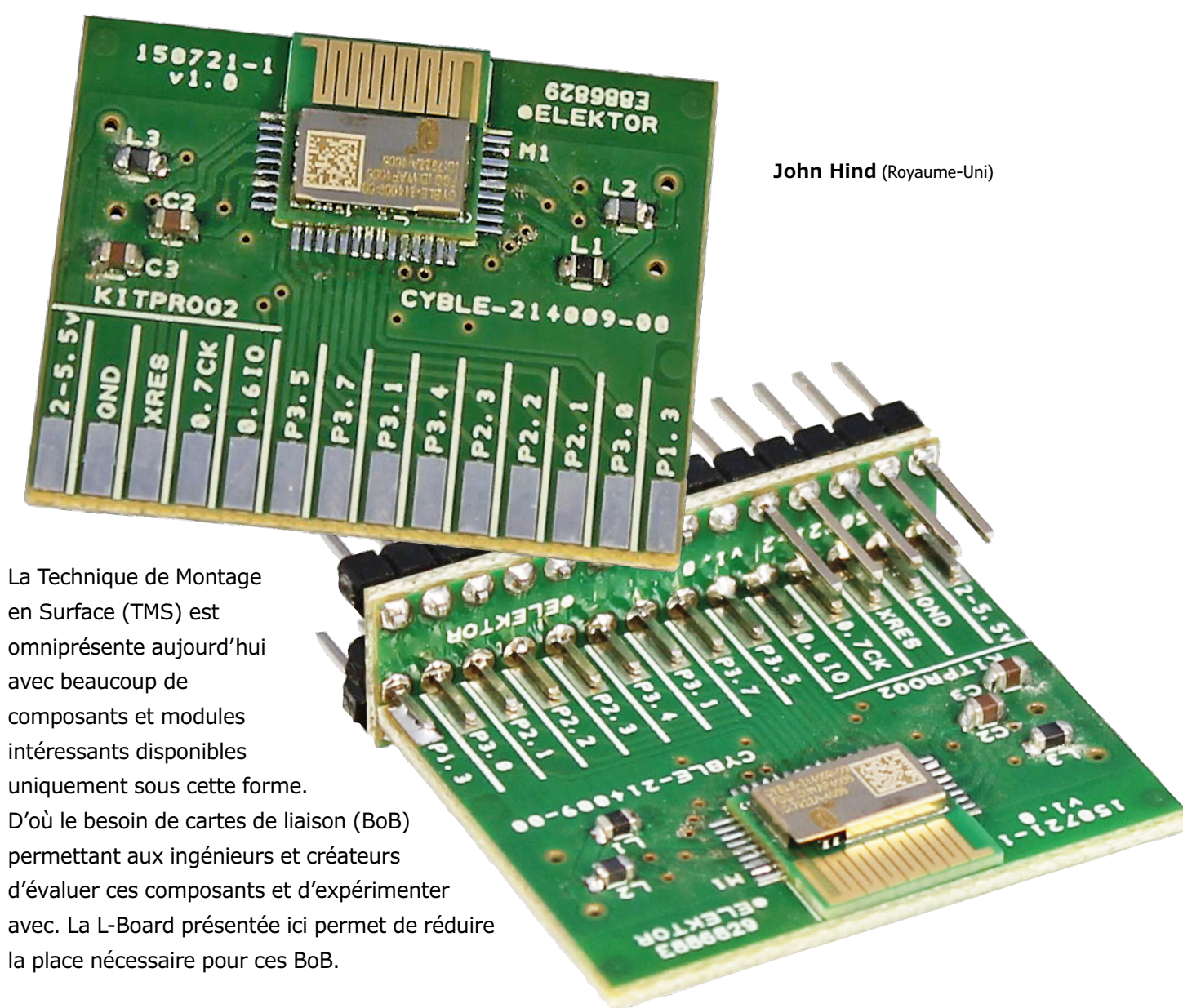
facebook.com/ElektorFR



twitter.com/Elektor

module PSoC BLE au format L-board adapté aux platines d'essais pour un prototypage Bluetooth tout confort

John Hind (Royaume-Uni)



La Technique de Montage en Surface (TMS) est omniprésente aujourd'hui avec beaucoup de composants et modules intéressants disponibles uniquement sous cette forme. D'où le besoin de cartes de liaison (BoB) permettant aux ingénieurs et créateurs d'évaluer ces composants et d'expérimenter avec. La L-Board présentée ici permet de réduire la place nécessaire pour ces BoB.

Normalement les BoB déploient les minuscules contacts TMS vers le pas standard à taille plus humaine de 2,54 mm (0,1"), utilisé pour les bar-

rettes et les platines d'essais. Pour les composants avec un nombre relativement faible de broches comme les capteurs et actionneurs, une seule rangée

de broches suffit pour les enficher dans une platine d'essais standard (les BoB utilisés dans l'actuelle série d'articles « Capteurs pour Arduino et Cie » en sont

Fonctions

- Module DIP, gain de place sur les platines d'essais
- PSoC avec Bluetooth Low Energy (BLE) 4.1
- Interface graphique de programmation

un bon exemple). Toutefois, lorsque le nombre de broches augmente, les BoB à simple rangée (SIL) deviennent trop longues et instables sur une platine d'essais. Une configuration à double rangée (DIL) améliore cette situation et beaucoup de BoB utilisent ce format (par ex. les diverses « T-Board » publiées par Elektor ces dernières années). Mais celles-ci posent leurs propres problèmes lorsqu'on les utilise sur une platine d'essais standard avec « gouttière » centrale conçue pour les boîtiers de circuits intégrés DIP traditionnels. Si la carte fait plus de 10 mm (0,4") de large, des points de connexion de la platine vont être masqués, ce qui limite la place pour des circuits périphériques. Une carte de 10 mm avec des contacts traversants de chaque côté laisse au mieux 6 mm (0,24") entre les contacts, insuffisant pour la plupart des boîtiers ou même pour router plus de quelques pistes vers le côté comme exigé par le format T-Board. Du coup la carte est toujours plus large que les 10 mm idéaux.

Introduisons la L-Board

Le format L-Board proposé ici résout ces problèmes. C'est une carte en deux parties avec une carte horizontale en gouttière de 10 mm soudée à une carte verticale portant la puce. Cette structure s'enfiche sur une platine d'essais standard, en recouvrant la gouttière sans occulter aucun point de connexion. On peut placer plusieurs L-Boards côte à côte sur la platine d'essais tout comme les bons vieux boîtiers DIP. L'ennui est que, comme les célèbres meubles Suédois, le kit doit être assemblé avant d'être prêt à l'emploi.

Ma première L-Board

Les modules PSoC BLE de Cypress Semiconductors constituent une intéressante application du concept L-Board. PSoC signifie Système sur Puce Programmable et combine ici un puissant processeur ARM à 32 bits avec un périphérique radio Bluetooth basse consommation (BLE) et des périphériques programmables numé-

riques logiques et analogiques avec intercommunication interne. Ajoutons qu'il y a aussi une famille PSoC, Radio sur Puce Programmable. Les puces PSoC BLE sont en fait une extension de la fonction PSoC, cette dernière n'incluant pas le périphérique numérique programmable. Mais elles sont largement interchangeables au plan du développement.

Comme l'indique leur nom, ces composants sont disponibles sous forme de puce, mais les micromodules que nous utiliserons ici ont une antenne intégrée, un oscillateur à quartz et sont préqualifiés, simplifiant grandement la fabrication en petite série. Malheureusement, ces modules sont presque impossibles à souder sans un équipement professionnel pour l'assemblage des CMS.

Les développeurs peuvent débuter avec la gamme PSoC/PROC à un coût particulièrement bas grâce à des modules d'étude subventionnés et à un Envi-

À PROPOS DU PROJET

T-board

L-board

BLE

PSoC

prototypage

BoB

débutant

→ connaisseur

expert

env. 2 h

Fer à souder, platine d'essais,
 programmeur Kitprog2,
 PC Windows, PSoC Creator

env. 50 €

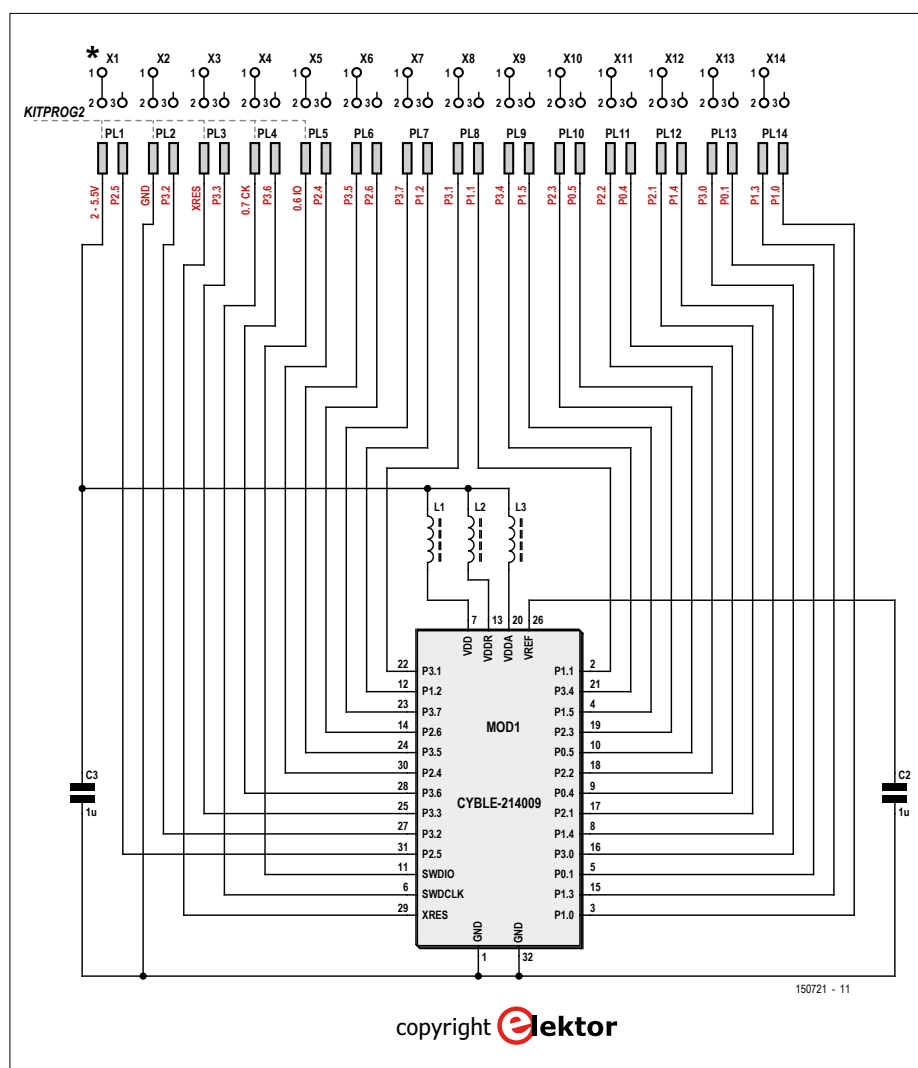
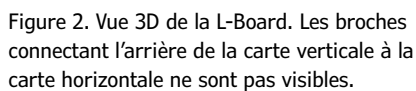
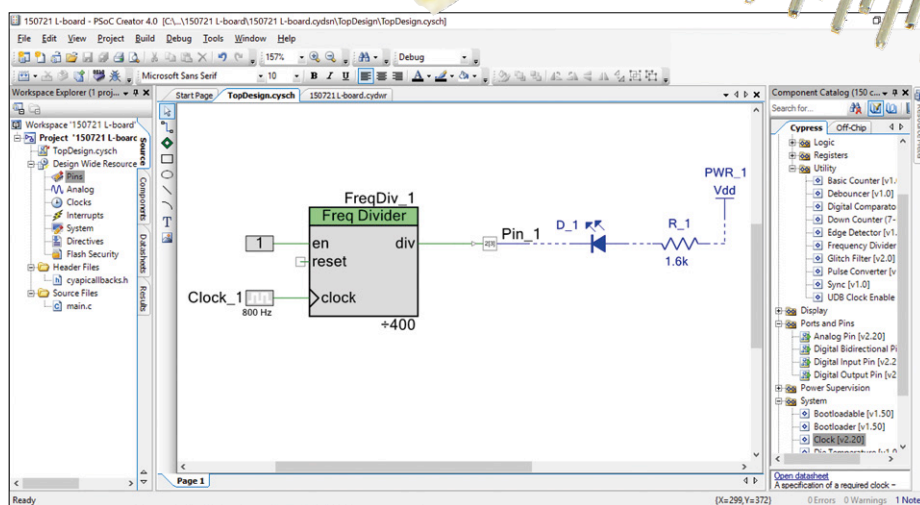


Figure 1. Schéma de la L-Board pour le module PSoC CYBLE-214009-00 avec support pour Bluetooth LE 4.1. PL1 à PL14 représentent le connecteur imprimé de la carte principale, X1 à X14 les trois rangées de barrettes sur la carte « gouttière ».



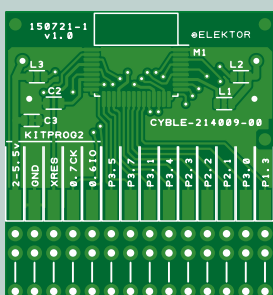
Il aurait été idéal d'utiliser le module en gamme étendue CYBLE-224116-01 actuellement en échantillons et supportant BLE4.2, mais il n'était pas prêt à temps pour cet article et le CYBLE-214009-00 BLE4.1 a été utilisé

La L-Board est conçue avec une découpe en « V » qui se casse pour séparer proprement les deux parties de la carte. Le montage final nécessite le soudage manuel de trois barrettes à 14 broches sur la carte gouttière (pour cette version à 28 broches). La barrette arrière a des broches des deux côtés de la carte. La barrette du milieu n'a des broches que sur le dessus, il faut donc couper les broches du dessous après les avoir soudées. C'est beaucoup plus facile de le faire



LISTE DES COMPOSANTS

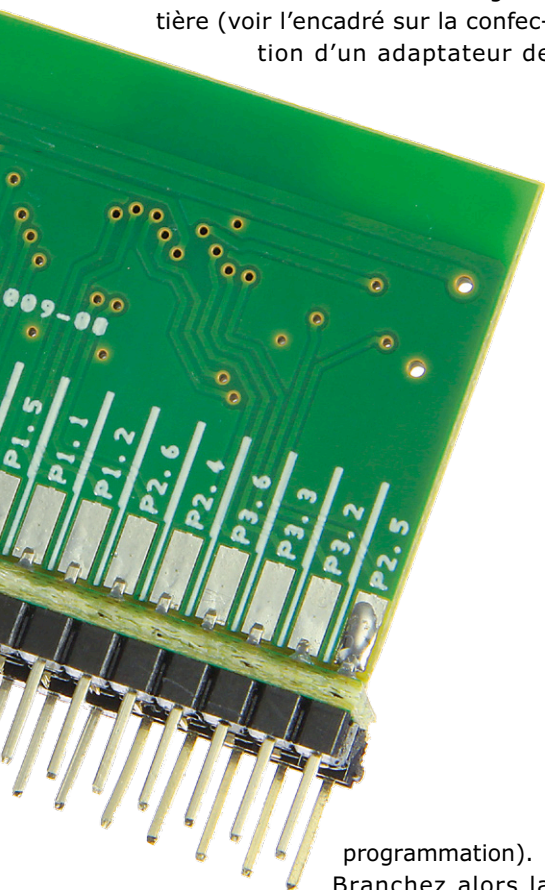
longueur de broche ~7 mm
circuit imprimé réf. 150721-1



avant d'ajuster la barrette avant. Celle-ci a aussi des broches sur le dessous de la carte, mais ici on doit laisser les cinq broches de gauche sur le dessus, car elles constituent le connecteur pour l'adaptateur Kitprog2. Enfin, la partie principale de la carte est placée sur le dessus de la carte gouttière entre la première et la seconde rangée de broches qui sont alors soudées directement sur les pastilles de chaque côté (**fig. 2**). L'ensemble peut maintenant être placé sur la gouttière d'une platine d'essais.

Hello, world !

Pour commencer à programmer, téléchargez l'EDI PSoC Creator sur le site de Cypress et installez-le sur un PC (malheureusement Windows seulement). Connectez le câble plat du Kitprog2 sur les broches de droite de la carte gouttière (voir l'encadré sur la confection d'un adaptateur de

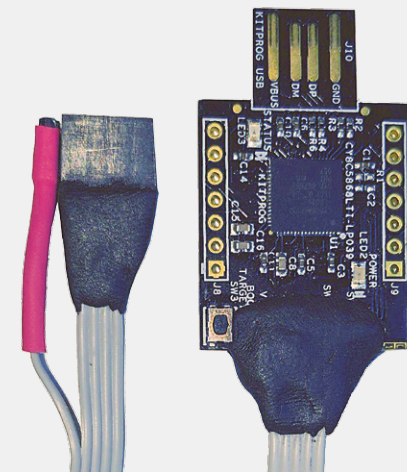


programmation). Branchez alors la prise USB du Kitprog2 au PC (une rallonge mâle/femelle USB-A convient). Sur la page d'accueil de PSoC Creator, sélectionnez [Create New Project](#). Sélectionnez le CYBLE-214009-00 comme module cible. Sur la page suivante, sélectionnez [Empty schematic](#) et sur la dernière page sélectionnez un dossier approprié pour stocker les fichiers. À ce stade, on peut être agréablement surpris en voyant un schéma vierge plutôt que l'habituel éditeur de code !

Adaptateur de programmation

Pour faire un adaptateur de programmation, détachez la partie Kitprog2 d'un kit d'étude (qui est prédécoupée pour ça) et soudez un morceau de câble plat à 5 conducteurs dans les cinq trous en face de la prise USB. Soudez l'autre extrémité du câble plat à une embase femelle simple rangée à 5 contacts.

Il vaut mieux la séparer en deux parties : une broche pour VTARG et quatre broches



pour les autres signaux. Ainsi la carte peut être alimentée soit par la connexion USB du Kitprog2, soit par la broche de la platine d'essais (un avantage singulier de l'architecture PSoC est l'inhabituelle large plage de tension de fonctionnement qui s'étend de 1,9 V à 5,5 V).

On peut aussi souder une barrette à cinq broches sur la carte cible initiale pour pouvoir l'utiliser avec le Kitprog2 détaché. La photo montre un montage Kitprog2, avec un produit de finition en caoutchouc malléable appelé Sugru, idéal dans le prototypage pour recouvrir l'arrière des connecteurs et autres montages électromécaniques isolés.

On trouve à droite le [Component Catalog](#). Glissez une [Digital Output Pin](#) depuis la section [Ports and Pins](#) sur le schéma, et sélectionnez [Configure](#) par un clic droit. Dans l'onglet [General](#), cochez les cases [Digital Output](#), [HW connection](#) et [External terminal](#) puis cliquez sur [OK](#). Ajoutez une LED, une résistance et une alimentation depuis l'onglet [Off-Chip](#) du [Component Catalog](#), disposez-les et connectez-les (**fig. 3**). Dans l'arborescence de l'espace de travail à gauche du schéma, double-cliquez sur [Pins](#) sous [Design Wide Resources](#). Vous verrez une image du brochage du module et une liste des broches sur le schéma, constituée pour le moment de l'unique broche de sortie ajoutée plus tôt. Dans le menu [Port](#), assignez-la à un port phy-

sique. Peu importe lequel, mais prenez [P2\[3\]](#) pour l'instant. Il est repéré [P2.3](#) sur la L-Board. Bien sûr le circuit physique de la LED doit être assemblé sur la platine d'essais (les composants hors circuit du schéma sont seulement pour la documentation).

Cliquez sur l'onglet [TopDesign.cysch](#) pour revenir au schéma. Depuis l'onglet [Cypress](#) du [Component Catalog](#), ajoutez un composant [Clock](#) (situé dans le dossier [System](#)) et réglez-le sur 800 Hz. C'est presque la fréquence minimale accessible avec les diviseurs d'horloge intégrés. Ensuite ajoutez un composant [Frequency Divider](#) (situé dans le dossier [Digital\Utility](#)) et réglez-le pour diviser par 400, en connectant sa borne horloge à l'horloge et sa sortie [div](#) à la bro-

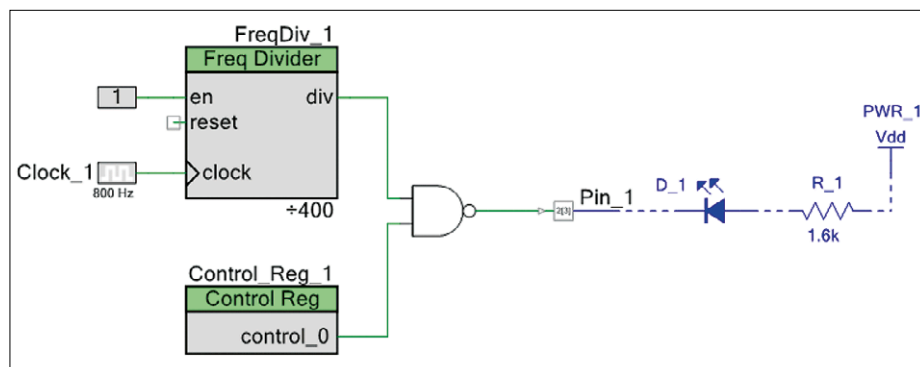


Figure 4. Dans la version étendue de « Hello, world ! », la LED est pilotée à la fois par le microcontrôleur et les blocs de logique numérique programmable.

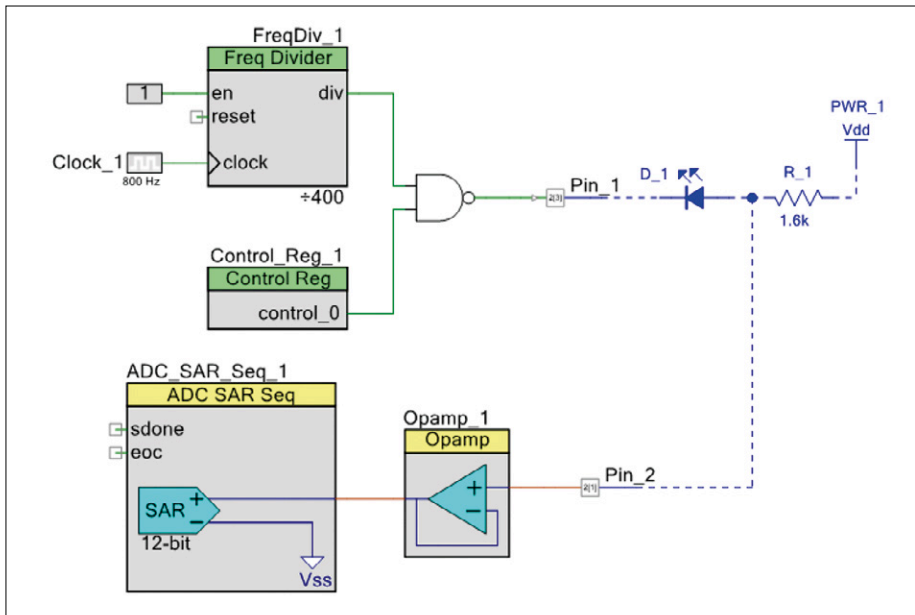


Figure 5. L'ajout d'un bloc CA/N au circuit permet de mesurer la tension directe de la LED.

Listage 1. Un programme simple pour essayer le débogage et le saut pas à pas dans le code.

```
#include "project.h"

int main(void)
{
    CyGlobalIntEnable; /* autoriser les interruptions */

    for(;;)
    {
        Control_Reg_1_Write(0x01);
        Control_Reg_1_Write(0x00);
    }
}
```

Listage 2. Le programme qui complète le circuit de la figure 5.

```
#include "project.h"

int main(void)
{
    CyGlobalIntEnable; /* autoriser les interruptions */

    volatile float32 volts = 0;
    ADC_SAR_Seq_1_Start();
    ADC_SAR_Seq_1_StartConvert();

    for(;;)
    {
        Control_Reg_1_Write(0x01);
        volts = ADC_SAR_Seq_1_CountsTo_Volts(0,
        ADC_SAR_Seq_1_GetResult16(0));
        Control_Reg_1_Write(0x00);
    }
}
```

che. Enfin, ajoutez un signal **Logic High** (voir le dossier [Digital\Logic](#)) et connectez-le à la broche **Enable** du diviseur. Vous devriez maintenant avoir quelque chose de similaire à la figure 3. Sauvegardez votre travail.

Sélectionnez alors l'option **Program** du menu **Debug**. Une fois le projet construit et téléchargé, la LED devrait commencer à clignoter. C'est l'équivalent embarqué du traditionnel *Hello, world* ! réalisé sans écrire une seule ligne de code. En fait, le processeur ARM ne tourne même pas : cela fonctionne entièrement en logique programmable !

Ajouter au schéma un **Control Register** avec une seule sortie et une porte **Nand**. Faites les connexions (**fig. 4**). Dans le **Workspace Explorer** sur la gauche, ouvrez le fichier `main.c` et ajoutez deux lignes de code à l'intérieur de la boucle `for` (cf. **listage 1**) :

```
Control_Reg_1_Write(0x01);
Control_Reg_1_Write(0x00);
```

Cette fois, sélectionnez **Debug** dans le menu **Debug**. Le processeur ARM démarre et s'arrête avant la première ligne de code. Pressez F10 deux fois et la LED commence à clignoter. Maintenez la pression sur F10 et constatez que la LED clignote quand le registre de contrôle est actif et reste éteinte quand il est remis à zéro.

Mesure d'une tension

Pour explorer les aptitudes analogiques, mesurons la tension directe de la LED. Allez d'abord sur la page [System de Design Wide Resources](#) et réglez les tensions d'alimentation sur 5,0 V (nous considérons que la carte est alimentée par le Kitprog2). Ajoutez au schéma une broche **Analog**, un **Opamp** et un **Sequencing SAR ADC**. Configurez la broche pour avoir un **External terminal**, l'ampli-op en **Follower** et le CN/A (DAC) à une cadence d'échantillonnage de 100 000 éch/s. Réglez **Vref select** sur **VDDA**, **Sequenced channels** sur 1, activez le canal 0 dans l'onglet **Channels** et mettez son mode sur **Single**. Dans l'onglet **Pins**, assignez **P2[1]** à la broche. Câblez comme sur la **figure 5**, sans oublier d'ajouter physiquement sur la platine d'essais le fil entre la LED et la broche analogique ! Ajoutez alors quelques lignes de plus au listage 1 pour obtenir le **listage 2** :

```
volatile float32 volts = 0;
ADC_SAR_Seq_1_Start();
```



```
ADC_SAR_Seq_1_StartConvert();
```

Dans le bloc `for`, entre les deux instructions d'écriture du registre de contrôle ajoutées plus tôt, insérez :

```
volts = ADC_SAR_Seq_1_
CountsTo_Volts(0,
ADC_SAR_Seq_1_GetResult16(0));
```

Débuguez ceci et observez la variable `volts`. Elle doit être entre 1,8 et 4,5 V selon la couleur et le type de LED, ou 0 V si la lecture est faite dans la phase d'extinction du clignotement. Examinez l'onglet *Analog* de *Design Wide Resources* pour voir comment le schéma analogique a été implanté dans la puce. Cela vaut aussi la peine de vérifier le *Resource Meter* accessible par l'onglet en haut à droite de l'application. Consommer près de la moitié des ressources UDB disponibles pour faire clignoter une LED pourrait vous inquiéter, mais comme toujours il y a plus d'une façon d'accommoder un lapin. On aurait pu utiliser un des quatre *Timer Counter PWM* ou la méthode traditionnelle avec une interruption de temporisateur et du code.

Ajoutons Bluetooth au mélange

Le Bluetooth Low Energy est traité dans un projet exemple téléchargeable en [1], mais jetons juste un œil. Glissez un composant *Bluetooth Low Energy* depuis la section *Communication* du catalogue et ouvrez sa fenêtre de configuration. On peut appliquer un des nombreux profils standard enregistrés, ou choisir *Custom* pour créer un nouveau profil. Normalement ce sera un serveur GATT et un périphérique GAP (ce sont les types pouvant être associés aux ordiphones, tablettes ou PC). Les propriétés et descripteurs sont définis dans l'onglet *Custom Service* de la page *Profiles*. Notez toutefois que ce module peut aussi jouer le rôle central et même dynamiquement changer de rôle. Ceci permet éventuellement les liaisons point à point et même de créer des réseaux en utilisant plusieurs modules sans recourir à un ordiphone, une tablette ou un PC. Le téléchargement fournit plus de détails et du code générique pour mettre en œuvre un service personnalisé ainsi qu'un exemple pleinement fonctionnel.

Considérez aussi le micrologiciel EZ-Serial de Cypress, surtout si vous n'êtes pas à l'aise avec la programmation en C. L'appli gratuite *CySmart* pour Android, iOS et Windows est un excellent outil pour déboguer et tester les disposi-

tifs BLE (quoique la version Windows nécessite l'achat d'une clé électronique dédiée).

Conclusion

Cet article a présenté la L-Board, un nouveau format pour les cartes de liaison qui présente des avantages significatifs sur le classique format T-Board et autres cartes de liaison « plates ». Un module PSoC BLE sophistiqué a été réalisé au format L-Board et quelques exemples montrant ses fonctions avancées ont été décrits. Cypress échantillonne actuellement une nouvelle génération de ces modules supportant BLE4.2 avec une vitesse et une capacité accrues, une meilleure sécurité et, très important, une portée améliorée. Il devrait être possible de poursuivre avec un de ces modules au format L-Board lorsqu'ils seront livrables. L'amélioration de la portée va ouvrir BLE vers de nouveaux champs d'applications passion-



nants au-delà du remplacement du câble actuel ou des contraintes du réseau personnel. Ceci comprend la domotique et peut-être une nouvelle génération d'instrumentation combinée bon marché, à la fois pour le laboratoire et le travail de terrain. ◀

(150721 – version française : Denis Lafourcade)

Liens

[1] www.elektormagazine.fr/labs/l-board

[2] www.elektormagazine.fr/150721

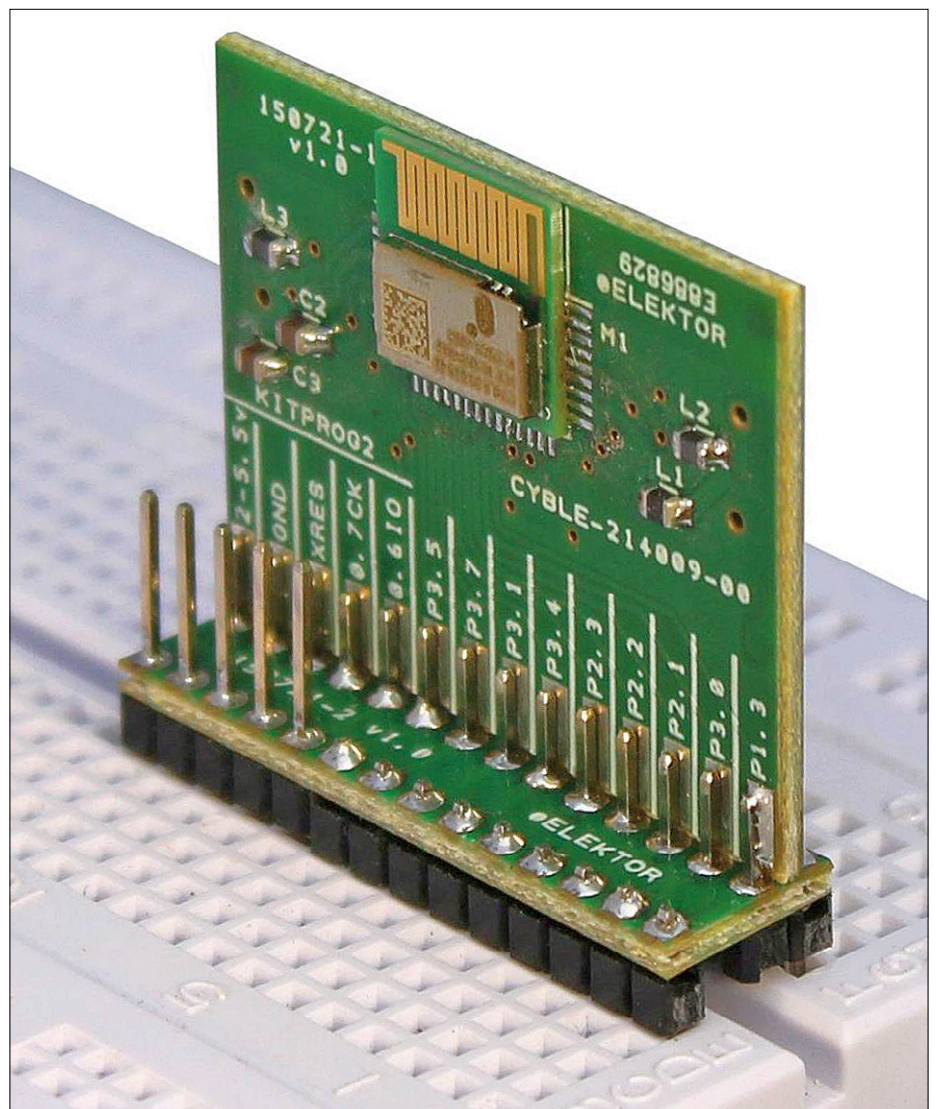


Figure 6. L-Board prototype assemblée, enfichée dans une platine d'essais.

BBC micro:bit pour électroniciens

premiers pas avec mbed

Burkhard Kainka

La carte BBC micro:bit ne convient pas seulement aux étudiants, c'est aussi un superbe microcontrôleur pour les électroniciens. On dispose sous un faible volume de presque tout ce que l'on a régulièrement besoin : entrées/sorties numériques avec MLI, convertisseur A/N ainsi qu'une interface USB pour l'alimentation, la programmation et l'échange de données. Divers capteurs et Bluetooth complètent le tout.

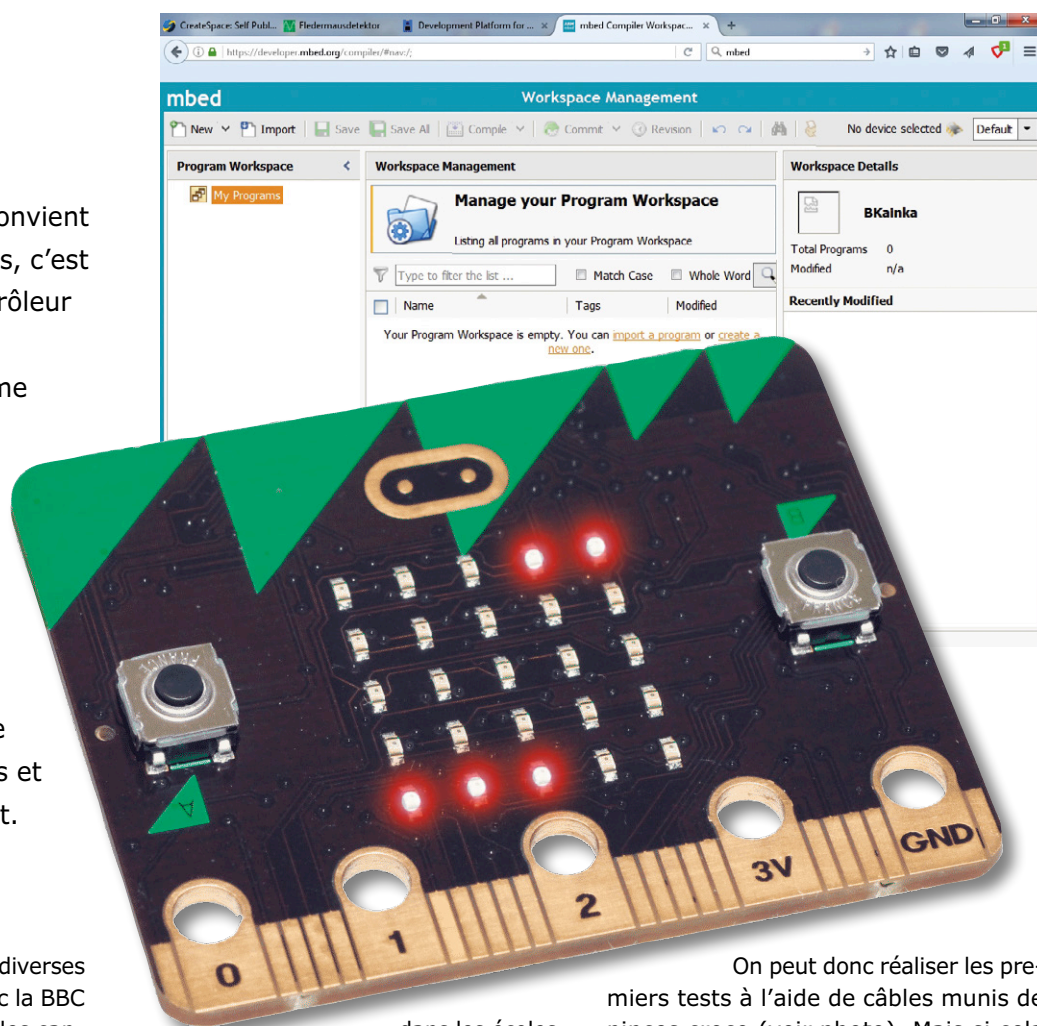
La carte Arduino offre également diverses entrées/sorties, en revanche avec la BBC micro:bit s'y rajoutent de multiples capteurs et éléments de commande : deux boutons, un affichage à 5x5 LED, une boussole et un accéléromètre à 3 axes, ainsi que des capteurs de lumière et de température. Cerise sur le gâteau : la transmission des données par Bluetooth Low Energy.

Contacts

La carte a été développée spécialement par la BBC en collaboration avec l'université de Lancaster pour être utilisée

dans les écoles, c'est pourquoi elle est équipée de cinq grands points de contact avec des trous de 4 mm auxquels on peut facilement connecter des pinces crocodile (**fig. 1**). Comme il y a un risque de toucher les broches voisines plus petites, une solution a été trouvée : les broches voisines des connexions de masse GND et 3 V sont au même potentiel. En outre les broches voisines des grandes connexions des ports 0,1 et 2 supportent une connexion directe sans problème.

On peut donc réaliser les premiers tests à l'aide de câbles munis de pinces croco (voir photo). Mais si cela devient plus complexe et que l'on a besoin de plus de connexions, il conviendra de trouver une solution sous forme d'un connecteur si l'on ne souhaite pas simplement souder de petits fils. Avec ses connecteurs au pas de 1,27 mm, la carte correspond en fait aux connecteurs d'extension d'une carte-mère de PC mise au rebut. Il est certainement possible de réaliser quelque chose à l'aide de pince, scie et fer à souder. Mais c'est plus confortable d'utiliser un connecteur avec



2 × 40 broches correspondant parfaitement. Des cartes compatibles sont disponibles à l'e-choppe [6][7]. La **figure 2** montre la carte du projet « station météo pour BBC micro:bit ».

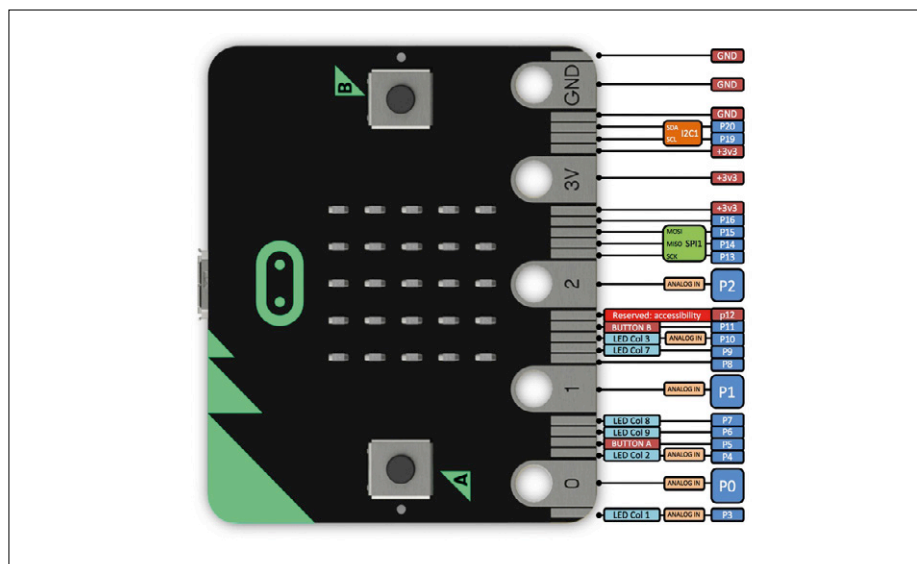
Une solution alternative est l'utilisation de barrette de 2 × 20 broches, que l'on soudera directement aux contacts (**fig. 3**). Les broches de la barrette sont néanmoins au pas de 2,54 mm, de sorte qu'elles ne correspondent malheureusement qu'à une broche sur deux de la carte BBC micro:bit.

Le verso de ce « connecteur maison » est similaire, mais on notera que toutes les connexions y sont complètement isolées. Les broches qui y sont soudées assurent avant tout une stabilité mécanique. Les broches du verso ont aussi une fonction de réserve : si l'on a quand même besoin d'un signal de la carte micro:bit non connecté jusqu'à présent, on pourra raccorder un petit fil à une broche du verso. On veillera cependant à supprimer la broche correspondante dans la rangée avant, afin d'éviter un court-circuit. Il y a suffisamment de broches inutilisées, telles que les connexions supplémentaires pour GND et VCC. En revanche, ce qui peut devenir important ce sont par ex. les connexions I2C, dont toutes les broches ne sont pas accessibles par la méthode de la barrette. C'est pourquoi on supprimera la broche GND superflue. C'est très facile : bien la chauffer au fer à souder, puis la retirer. Un petit fil peut à présent être passé dans le trou puis soudé à la broche souhaitée (SDA, P20). Le petit fil sera soudé au verso à la broche correspondante. Et voilà le bus I2C retrouvé.

Premiers programmes

Les applications pratiques dans le domaine de l'électronique sont très faciles à réaliser. Il existe à ce sujet de nombreux langages de programmation développés spécifiquement pour l'école [1][5]. Mais nous utiliserons C++, car avec lui pratiquement tout est réalisable, alors que d'autres langages de programmation ont toujours quelques restrictions. C'est particulièrement simple avec la plateforme mbed [2], car on travaille en ligne et rien n'est à installer localement.

Si vous ne l'avez déjà fait pour d'autres projets, connectez-vous à mbed et créez votre propre espace. Si vous essayez alors de charger un exemple disponible, il vous sera rappelé de choisir d'abord



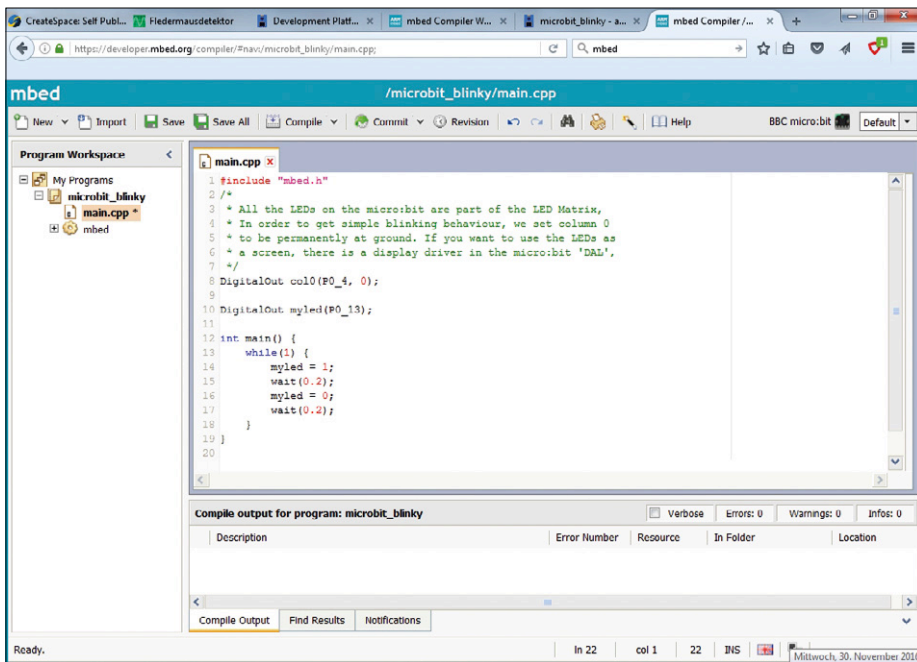


Figure 4. Le projet *microbit_blinky*.

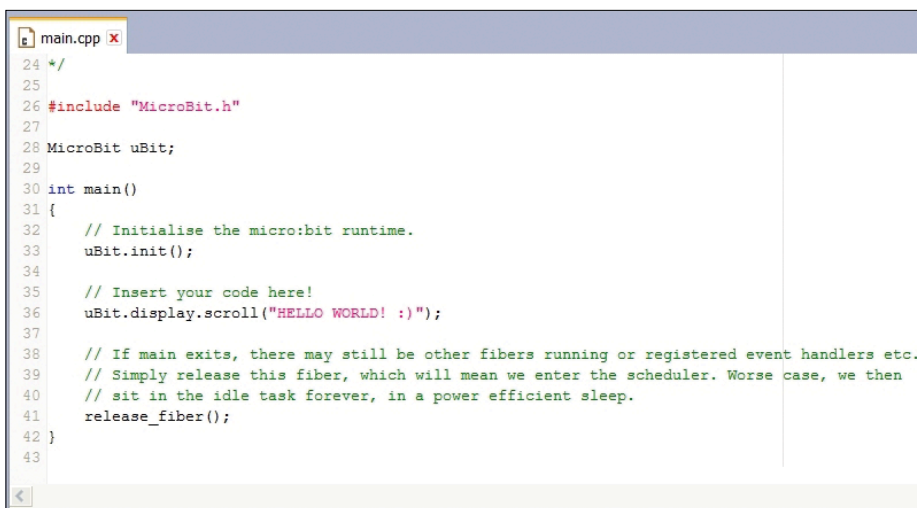


Figure 5. Le projet *microbit-hello-world*.

Listage 1. Mesure de tension.

```
//Voltage1
#include "MicroBit.h"

MicroBit uBit;

int main()
{
    uBit.init();
    MicroBitSerial serial(USBTX, USBRX);
    while (1) {
        int u = 3300 * uBit.io.P0.getAnalogValue() / 1023;
        uBit.display.scroll(u);
        uBit.serial.printf("%d\r\n", u);
        uBit.sleep(500);
    }
}
```

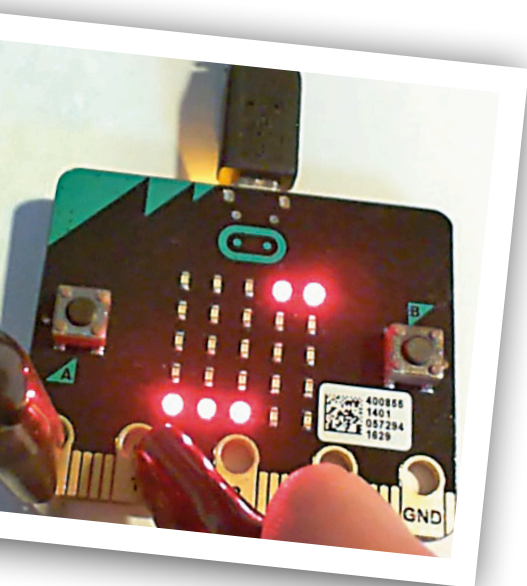
une plateforme, à savoir un système avec lequel vous voulez travailler. Vous serez guidé vers la page du matériel, vous serez étonné par le nombre de systèmes pouvant être utilisés, l'un d'entre eux étant par ailleurs d'Elektor. Dans notre cas, micro:bit est notre cible. On trouve alors d'autres informations comme le schéma de câblage de la carte micro:bit et l'option « add to your mbed Compiler ». On y trouve aussi les liens essentiels vers la documentation de l'université de Lancaster et les premiers exemples. Le premier exemple s'appelle *microbit_blinky* et doit à présent être chargé. Le compilateur affiche la plateforme nouvellement intégrée et le programme importé. Un clic sur *main.cpp* ouvre un code source clair (fig. 4).

Ce qui frappe immédiatement est la désignation des ports avec P0_4 et P0_13, donc toute autre que la désignation officielle. Cela vient du fait qu'à cause de l'inclusion de la bibliothèque *mbed.h*, on s'adresse bien au contrôleur de micro:bit, mais pas encore à la carte avec toutes ses caractéristiques et possibilités. Mais le programme peut désormais être testé. Avec *compile*, on produit un fichier *hex* qu'il suffit de copier dans la mémoire USB de micro:bit. La LED jaune d'état se met aussitôt à clignoter sur la carte, et indique que le système s'apprête à programmer le contrôleur. Le processus prend fin après une seconde et on peut voir le résultat : la LED en haut à gauche de l'afficheur de 5x5 LED clignote.

Le deuxième exemple s'appelle *microbit-hello-world* et inclut *microbit.h* (fig. 5). Il est alors possible d'utiliser toutes les spécificités de la carte. L'importation du programme dure particulièrement longtemps, ce qui signifie que beaucoup de fichiers sont chargés. C'est le grand point positif de mbed : mbed prend lui-même en charge tout ce que l'on devait auparavant péniblement copier et inclure. Cela serait faisable aussi avec une autre plateforme sur son propre ordinateur, peut-être en y travaillant dur et au risque d'y renoncer finalement. On peut travailler avec mbed sans même se douter de tout ce qui se déroule en arrière-plan. Un texte défilant est produit dans notre

Listage 2. Affichage de la mesure de capteurs.

```
//Sensors
#include "MicroBit.h"
MicroBit uBit;
int main()
{
    uBit.init();
    MicroBitSerial serial(USBTX, USBRX);
    while (1) {
        uBit.serial.printf("Time: %d ms \r\n", uBit.systemTime());
        uBit.serial.printf("Temp: %d deg \r\n", uBit.thermometer.getTemperature());
        uBit.serial.printf("P0: %d mV \r\n", uBit.io.P0.getAnalogValue());
        uBit.serial.printf("P1: %d mV \r\n", uBit.io.P1.getAnalogValue());
        uBit.serial.printf("P2: %d mV \r\n", uBit.io.P2.getAnalogValue());
        uBit.serial.printf("X: %d mG \r\n", uBit.accelerometer.getX());
        uBit.serial.printf("Y: %d mG \r\n", uBit.accelerometer.getY());
        uBit.serial.printf("Z: %d mG \r\n", uBit.accelerometer.getZ());
        uBit.serial.printf("B: %d µT \r\n", uBit.compass.getFieldStrength());
        uBit.sleep(1000);
    }
}
```



exemple. La ligne clé est explicite : `uBit.display.scroll(«HELLO WORLD! :»)` ; compilez, transmettez, ça marche.

On pourrait penser d'abord, tant d'efforts pour un simple texte défilant ? Mais en fait mbed a bien plus à offrir, à savoir l'accès complet à tous les composants matériels importants du système, donc non seulement à l'afficheur, mais aussi aux ports avec convertisseur A/N et MLI, aux capteurs spéciaux tels que la boussole et l'accéléromètre, et à bien plus encore. Sur notre plan de travail se trouve désormais le répertoire *microbit* et dans celui-ci le sous-répertoire *microbit-dal* (Device Abstraction Layer) avec un nombre incroyable de fichiers que l'on pourrait fouiller des heures durant pour y découvrir toutes les possibilités.

Mais c'est plus facile avec la documentation de l'université de Lancaster [3]. On y trouve pour chaque thème toutes les informations importantes, mais aussi des extraits de code qu'il est possible de copier et de coller dans son propre code source. On pourrait donc simplement compléter le code source disponible `main.cpp` dans notre projet *microbit-hello-word* et tester le tout après coup. Une alternative est de cloner le projet et de continuer à travailler sur la copie. Le projet cloné s'appelle désormais par ex. *microbit-test* et dans un premier temps fait la même chose. On présente ci-dessous quelques petits codes source qu'on peut simplement copier dans `main.cpp` et tester. Tous les programmes sont disponibles en téléchargement au format txt sur le site d'Elektor [8].

Mesure de tension

On a toujours besoin dans un labo d'électronique de CA/N. Micro:bit dispose d'un CA/N à 10 bits et à six entrées possibles dont trois situées sur les grands points de connexion. Un simple programme est présenté (**listage 1**) afin d'en tester les caractéristiques. On obtient avec `uBit.io.P0.getAnalogValue()` une valeur à 10 bits pour une plage de tension jusqu'à 3,3 V. Celle-ci est convertie en mV puis affichée.

La méthode d'affichage typique sur la carte micro:bit est le texte défilant sur l'afficheur à LED. Cela fonctionne bien,

mais la lecture requiert une certaine concentration. Toutefois il y a aussi l'interface série, on peut ainsi visualiser les données sur un terminal. Il faut installer le pilote *mbed Windows serial port* pour utiliser le tout sur un PC sous Windows. La procédure est décrite en [4].

En reliant la borne 3,3 V de la carte à l'entrée 0, on obtient la valeur attendue de 3300 mV. Des mesures comparatives montrent que la tension réelle est légèrement inférieure. En fait la tension de 5 V de l'USB de la carte est stabilisée à 3,3 par un régulateur, puis connectée à VCC par une diode Schottky.

Avec une entrée flottante, la mesure obtenue est environ 880 mV. Une mesure simultanée à l'oscilloscope en haute impédance indiquera une tension plus élevée qui s'interrompt brièvement à chaque mesure. En fait les connexions les plus importantes (0, 1 et 2) comportent des résistances de rappel de 10 MΩ afin que les broches de port numériques puissent facilement être utilisées en capteurs tactiles. La tension à vide réelle devrait donc être de l'ordre de 3,3 V. Pour mesurer celle-ci, on place un condensateur de 100 nF à l'entrée. Celui-ci se charge alors à la tension à vide puis fait office de tampon durant la mesure.

Tous les capteurs

Le deuxième petit exemple (**listage 2**) montre de nombreuses mesures des différentes entrées et de capteurs, y

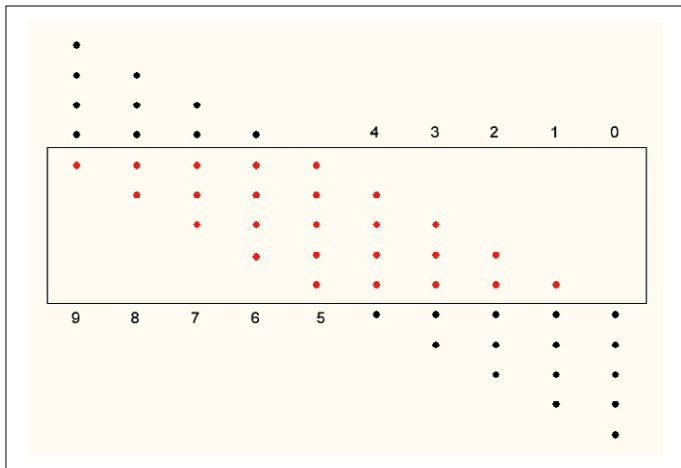


Figure 6. Affichage des chiffres.

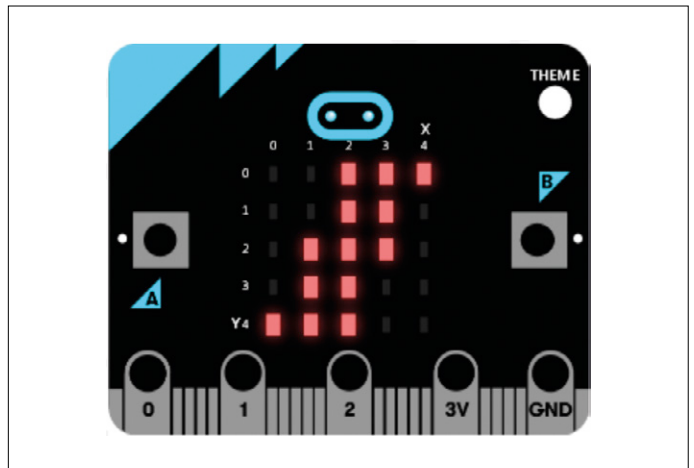


Figure 7. Affichage du nombre 13579.

compris la température du contrôleur et la durée en millisecondes depuis le dernier démarrage, les trois tensions disponibles sur les broches 0 à 2, les valeurs d'accélération sur trois axes et le champ de force magnétique combiné XYZ. Deux capteurs complexes sont simplement lus grâce au bus I2C, le gros du travail ayant déjà été réalisé par d'autres. Pour en savoir plus, vous aurez à vous frayer un chemin dans les arcanes de nombreux fichiers.

Toutes les mesures sont affichées par le terminal. Pendant que l'on procède à tous ces essais nous viennent des idées pour des applications telles que des mesures avec des aimants. Ou un accéléromètre pour la voiture, qui informerait son conducteur d'une conduite trop sportive ? L'affichage produit pour-

rait ressembler à ceci :

```
Time: 1715018 ms
Temp: 22 deg
P0 465 mV
P1 251 mV
P2 252 mV
X 0 mG
Y 1008 mG
Z 64 mG
B: 169311 uT
```

Afficheur numérique statique

Un texte défilant sur un afficheur à LED n'est pas facile à lire, il faut beaucoup se concentrer. Il faudra attendre le prochain cycle d'affichage s'il l'on a raté un chiffre. Mais 25 LED offrent d'autres possibilités. Nous sommes habitués à des afficheurs statiques à plusieurs chiffres ou à des

instruments analogiques à aiguille.

Un afficheur fixe pourrait fonctionner en binaire ou en BCD. Dans les deux cas, on a besoin de quatre bits par chiffre. Une autre possibilité s'offre à nous sachant que l'on dispose de cinq LED par colonne. Les chiffres de 1 à 5 sont représentés à partir du bas par une à cinq LED. La barre lumineuse se déplace vers le haut pour les chiffres de 6 à 9. On peut s'imaginer disposer de cinq points, dont certains ne seraient pas toujours dans le domaine visible (**fig. 6**).

La lecture d'un tel afficheur (**fig. 7**) est intuitive après un peu d'entraînement. La représentation correspond par hasard au codage en morse des chiffres 0 à 9, une LED allumée représente un point, une LED éteinte un trait. Le sept se codera donc en morse « --... ».

Listage 3. Mesure de tension avec affichage statique.

```
//LED-Display
#include "MicroBit.h"
MicroBit uBit;

void ledDisplay (int n){
    int x;
    int y;
    int d;
    uBit.display.enable();
    MicroBitImage image(5,5);
    image.clear();
    for(int i = 0; i < 5; i++){
        d = 9 - n % 10;
        n = n / 10;
        x = 4 - i;
        for(int j = 0; j < 5; j++){
            y = d - j;
            image.setPixelValue(x,y,255);
        }
    }
}
```

```
}
uBit.display.print(image);
}

int main()
{
    uBit.init();
    uBit.io.P1.setDigitalValue(0);
    uBit.io.P2.setDigitalValue(1);

    MicroBitSerial serial(USBTX, USBRX);
    while (1) {
        int u = 3300 * uBit.io.P0.getAnalogValue() / 1023;
        uBit.serial.printf("%d\r\n", u);
        ledDisplay (u);
        uBit.sleep(500);
    }
}
```

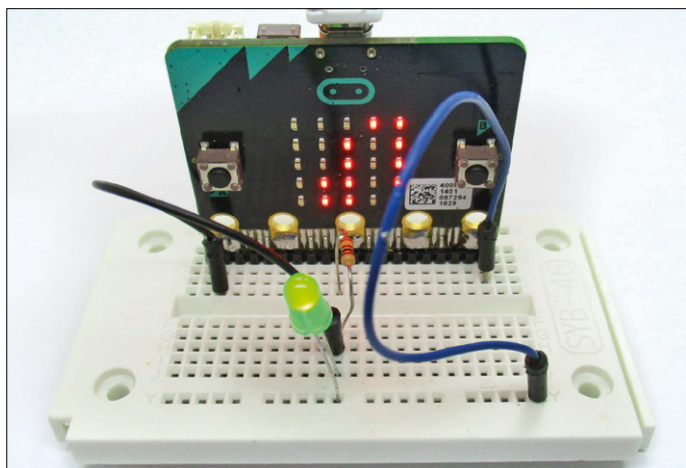



Figure 8. Mesure de la tension d'une LED : 2,496 V.

Une fonction pour ce type d'affichage a été écrite en C. Le chiffre à afficher est transmis dans un nombre entier n . Une tension jusqu'à 3300 mV est affichée (**listage 3**).

Deux broches de port sont commutées en plus dans le programme principal. P1 devient bas, P2 devient haut. Un changement de direction des données ne doit pas être défini explicitement, celui-ci s'effectue automatiquement en arrière-plan. On peut à présent facilement examiner la capacité de charge du port, et utiliser à cet effet son propre CA/N. Il s'avère que celui-ci a clairement une impédance plus élevée que les ports d'un contrôleur AVR. Officiellement les ports sont conçus pour 5 mA dans les deux directions. Les mesures montrent qu'il se produit une chute de tension de 300 mV. La résistance de conduction du transistor FET du port est donc environ 60 Ω . Des mesures avec des charges plus élevées montrent que le courant de sortie ne dépasse jamais 15 mA. Un courant d'environ 10 mA circule dans une LED connectée directement sans résistance.

On utilisera normalement une résistance série. On peut alors mesurer la tension de la LED, comparer différentes LED, et procéder à d'autres expériences (**fig. 8**). ◀

(160273 – version française : Xavier Pfaff)

Liens

- [1] <http://microbit.org/code/>
- [2] <https://developer.mbed.org/>
- [3] <https://lancaster-university.github.io/microbit-docs/ubit/>
- [4] <https://developer.mbed.org/handbook/Windows-serial-configuration>
- [5] B. Kainka, *BBC Micro:bit Tests Tricks Secrets Code*, CreateSpace 2016
- [6] www.elektormagazine.fr/150652
- [7] www.elektormagazine.fr/160274
- [8] www.elektormagazine.fr/160273

Les dernières nouveautés pour vos conceptions les plus récentes™



MOUSER
ELECTRONICS

Plus de
nouveaux produits
en stock que
n'importe quel
autre distributeur.



Commandez dès maintenant sur

mouser.fr



trucs et astuces

les lecteurs écrivent aux lecteurs

Encore des solutions futées qui facilitent la vie des électroniciens.

Souder des pattes de 0,5 mm

Peter Krengel

En feuilletant le numéro de décembre d'Elektor, je suis tombé sur l'article « Souder au pas de 0,5 mm » qui a piqué ma curiosité. Quelle était l'astuce proposée ? Ma curiosité se mua vite en un sourire, car je ne connais que trop bien ces petites bêtes à 144 pattes, je suis un gros utilisateur de FPGA d'Altera et de Xilinx.

Mais la solution du couteau pour gratter m'a vraiment choqué, car il y a tout de même des méthodes beaucoup plus simples, propres et rapides !

En fait, il faut d'abord souder tous les composants standards CMS au four à refusion, sauf ces mille-pattes. Ma méthode vraiment simple fonctionne même si le soudage au four n'est pas parfait et a laissé sur le circuit imprimé des petites billes ou des bavures. Ma solution à ce problème s'appelle panne de fer à souder gull-wing (en aile de mouette) qu'on peut se procurer chez

Weller ou Ersa [1][2].

Il s'agit d'une panne qui présente une cavité que l'on remplit d'une petite goutte de soudure. J'utilise une telle panne avec la technique ci-dessous. Tout d'abord, gratter

est totalement inutile, car, en fait, il ne se forme pas de petites billes, même avec un four défectueux ou mal réglé. Et même les petites pattes de 0,3 mm se laissent souder proprement et sans court-circuit. Voici comment faire :

1. Nettoyer les pastilles en les frottant avec du flux de soudure liquide de type No-Clean (sans nettoyage) et enduire toutes les pattes de la puce avec ce flux. C'est facile avec un stylo applicateur de flux, il suffit d'appuyer sa pointe quelque part pour en faire jaillir un peu du liquide. Il faut y aller doucement, sans tordre les pattes, juste épandre la goutte qui pend à la pointe du stylo, puis laisser sécher.
2. Au moyen d'une loupe (grossissement $\times 10$), positionner le mille-pattes aussi précisément que possible sur les pastilles,

puis souder l'une des pattes de coin avec une panne de 0,2 mm (par ex. Weller RT).

3. Contrôler encore une fois à la loupe le positionnement de la puce et le corriger légèrement au besoin.

4. Remplacer la panne de 0,2 mm par la gull-wing. Régler la température pour que la soudure soit bien liquide, mais ne s'oxyde pas. Il faut au préalable tester ce réglage avec précision.

5. Remplir la panne gull-wing de soudure, l'appuyer immédiatement sur une patte de coin et balayer lentement les pattes. Effectuer l'opération sur les quatre côtés.

6. Pour finir, contrôler les soudures et, au besoin, répéter le balayage avec la panne vide.

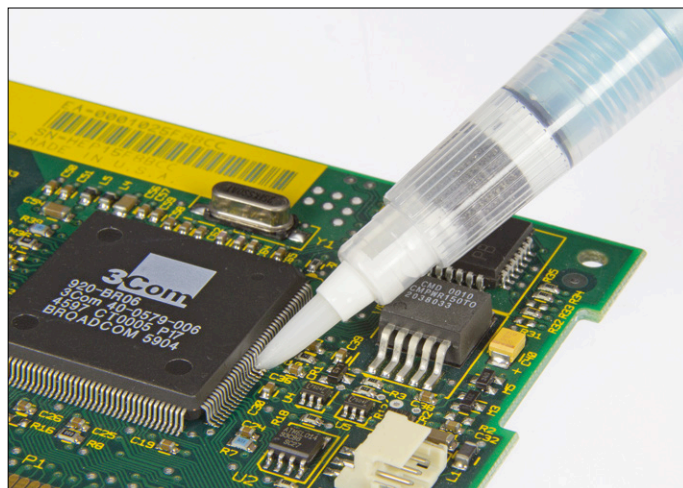
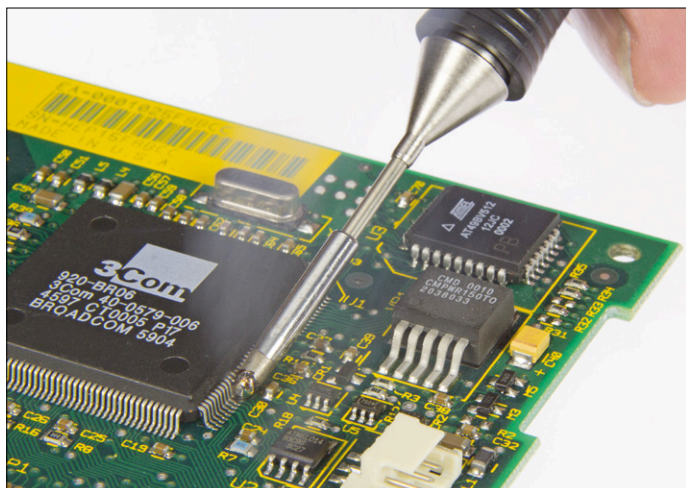
Et le meilleur, c'est que les 144 pattes sont traitées en deux minutes au maximum (y compris le contrôle). Une surchauffe est impossible, vu la courte durée de l'opération de soudage. Essayez, vous m'en direz des nouvelles !

Les points de soudure semblent réalisés à la machine, mais ont aussi un brillant qu'on n'obtient pas avec la machine. Cela tient sans doute au fait que le temps de soudage par patte est très court du fait du balayage, ce qui ne laisse pas à la soudure le temps de s'oxyder, même en cas de température légèrement trop élevée. De plus, la tension superficielle de la soudure non oxydée agit dans le sens d'une répartition très régulière de la soudure, car les liquides ont tendance à se diviser en corps de masse égale, comme l'eau en apesanteur (mais je digresse ;)). En tout cas, la soudure se retrouve finalement également répartie et seulement là où elle doit être, c'est-à-dire sur les pastilles.

[1] Weller : www.weller.de/en/Weller--Products--Product-details.html?article_id=D0719205001379336391A111723

[2] Ersa Micro : www.ersa-shop.com/serie-f%C3%BCr-microtool-hohlkehle-163mm-c-80_81_125_130.html





Sauvetage des soudures ratées

Peter Krengel

Même les soudures ratées peuvent être sauvées avec une panne gull-wing. Pour cela, on a à nouveau besoin du stylo applicateur de flux liquide.

1. Répandre du flux sur les pattes de la puce soudée assez abondamment pour qu'il en pénètre un peu sous la puce.
2. Immédiatement après l'écoulement du flux (qui sèche très vite !), balayer les pattes sur tous les côtés de la puce avec la panne gull-wing vide.
3. C'est terminé !

À la place du flux No-Clean, on peut aussi utiliser de la colophane, nettement moins chère. On se procure une allumette (usagée), une bonne brosse à dents propre et un flacon de 100 ml d'alcool éthylique à 99% (qui n'est pas là pour être bu !). Une aussi petite quantité d'alcool peut s'acheter en pharmacie (à un prix de pharmacien) ou, pour moins cher, sur l'internet. Il faut éviter l'alcool à brûler ou l'isopropanol qui laisseraient sur le circuit imprimé des vilaines taches blanches très difficiles à enlever.

On liquéfie la colophane dans un récipient métallique, avec l'allumette on en prélève quelques gouttes qu'on répartit sur les pattes de la puce. Ici aussi on doit faire vite. Si la colophane se solidifie trop vite, on peut préchauffer le circuit imprimé entre 120 et 130 °C avec un sèche-cheveux, ou mieux, une soufflante à air chaud.

Comme précédemment, on balaie les pattes avec la panne gull-wing vide, plusieurs fois au besoin. Toutes les bavures de soudure devraient maintenant s'être accumulées sur les pattes ou sur la panne, laissant des points de soudure propres.

La brosse à dents et l'alcool entrent maintenant en action, car contrairement à la méthode No-Clean, l'usage de la colophane rend un nettoyage à l'alcool obligatoire, car les pattes continuent

à baigner dans le flux après le balayage. Mais auparavant, il faut que le circuit imprimé soit parfaitement refroidi, au besoin en le mettant au congélateur pendant cinq minutes.

On pulvérise la colophane en excès par un léger raclage avec l'allumette (pour éviter des rayures sur le circuit imprimé). Après élimination de la poussière par tapotage et soufflage, on nettoie tous les résidus par un brossage généreux à l'alcool. Terminé !

La colophane à prix d'ami

Peter Krengel

Voici un petit tuyau pour tous ceux qui pestent contre le prix excessif d'une quantité minime de colophane. Sur l'internet, cherchez colophane+charcuterie. Vous y trouverez à moindres frais de la poudre de résine de colophane pure en vrac, utilisée pour épiler les cochons ou plumer la volaille. On peut faire fondre cette résine en la chauffant et la conserver ensuite sous forme solide dans une petite boîte de crème préalablement bien nettoyée à l'alcool et au produit à vaisselle.

Nettoyage d'une tresse à dessouder

Peter Krengel

La colophane est aussi utile dans le cas d'une tresse à dessouder qui n'absorbe plus. On plonge la tresse dans de l'acide chlorhydrique dilué entre 5 à 10% (disponible en grande surface de bricolage), ce qui élimine l'oxyde de cuivre soluble dans l'acide. Ensuite on rince la tresse à l'eau et on la fait passer dans la colophane fondue. On attend qu'elle durcisse et on élimine l'excédent de résine. Votre tresse n'aura jamais aussi bien fonctionné, on parie ?

(160324 – version française : Helmut Müller)

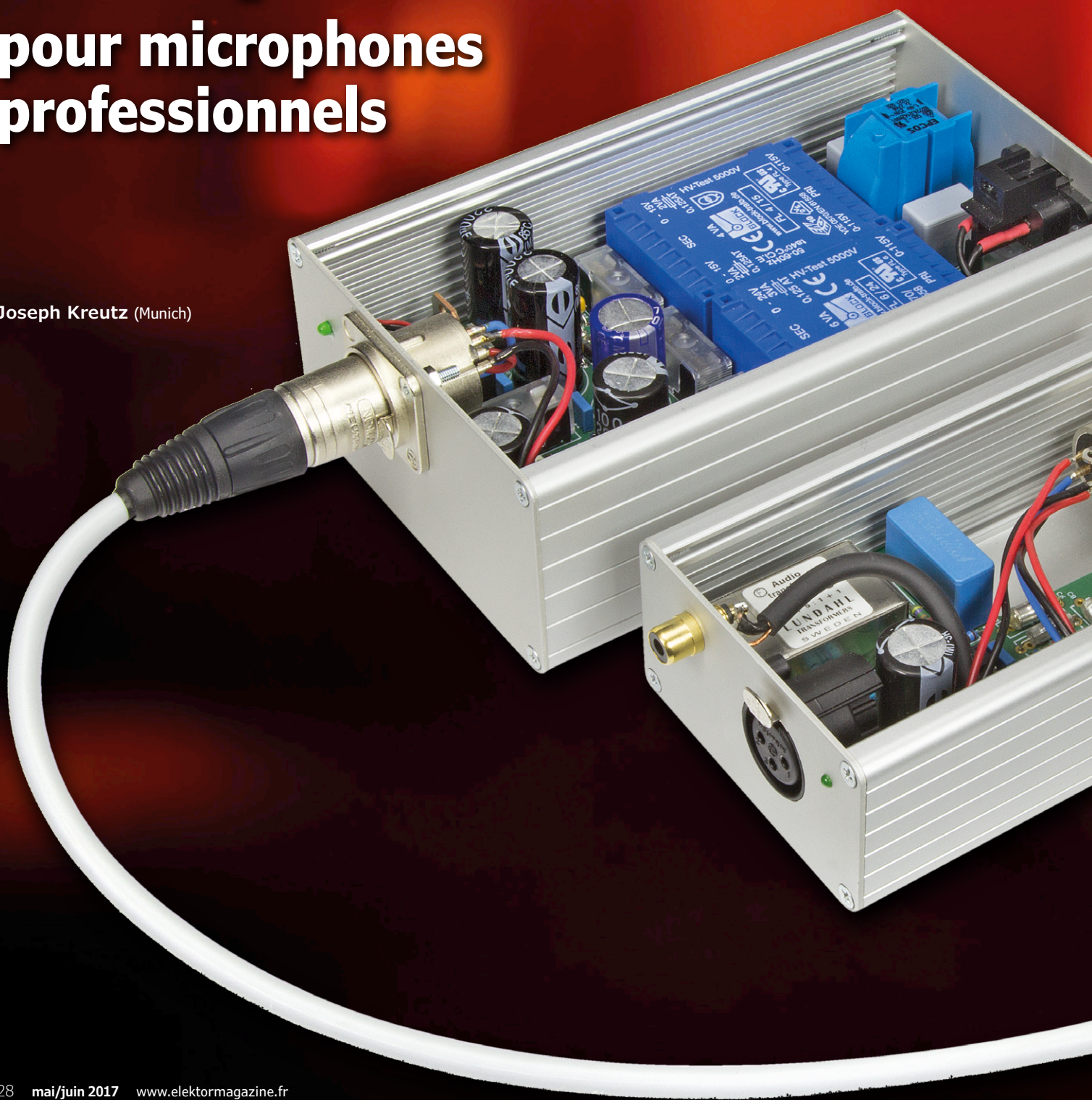
Vous avez une solution fûtée pour arranger une bricole... Une façon bien à vous d'utiliser un composant ou un outil... Vous savez comment résoudre un problème plus facilement ou mieux qu'avec la solution actuelle... Écrivez-nous – chaque astuce publiée vous rapportera 40 € !



préamplificateur simple

pour microphones professionnels

Joseph Kreutz (Munich)





Il y a belle lurette que les enregistreurs à bande sont tombés en désuétude et ont été remplacés par des ordinateurs portables équipés d'une « carte son USB » de bonne qualité. Alors que l'entrée « ligne » d'un PC présente généralement des caractéristiques acceptables, l'entrée « micro » se caractérise par de médiocres niveaux de bruit et de distorsion. Surtout, elle ne restitue qu'un signal monophonique et n'est pas adaptée à la connexion de microphones professionnels à sortie symétrique, qui souvent nécessitent une alimentation « fantôme » de 48 V. Je vous propose ici un préamplificateur qui permet de contourner ces deux écueils et ravira sûrement les possesseurs de cartes son « haut de gamme ».

Circuit d'amplification

La **figure 1** représente le schéma du préamplificateur. Apparemment surannée, l'utilisation d'un transformateur (TR1) permet de concevoir un circuit simple avec un isolement galvanique efficace de l'entrée et une bonne réjection du signal en mode commun. Le transformateur réduit substantiellement les interférences telles que celles provoquées par les téléphones portables. Vous trouverez facilement des modèles pour

microphones avec un rapport de 1:10. Ce type de transfo élève de 20 dB la tension du signal, ce qui améliore considérablement le rapport signal/bruit du préamplificateur.

Pour conserver tous ces bénéfices, il faut un amplificateur opérationnel à entrées FET dont le courant de bruit est négligeable. L'ampli op AD743 est idéal parce que son bruit d'entrée est très faible (2,9 nV/√Hz [1]), mais il est cher. Le modèle OPA134 de Burr-Brown (TI) meil-

leur marché est un excellent choix ; toutefois pour le prototype du préampli, j'ai utilisé un TL071 avec un résultat étonnamment bon [2]. Les amplificateurs opérationnels doubles comme OP2134 ou TL072 conviendront parfaitement à des réalisations stéréophoniques.

Pour acheter le transformateur, jetez un œil chez Bürklin [3] qui propose des modèles avec un bon rapport qualité/prix. Les sociétés Lundahl [4], Sowter [5] et Jensen [6] proposent des produits

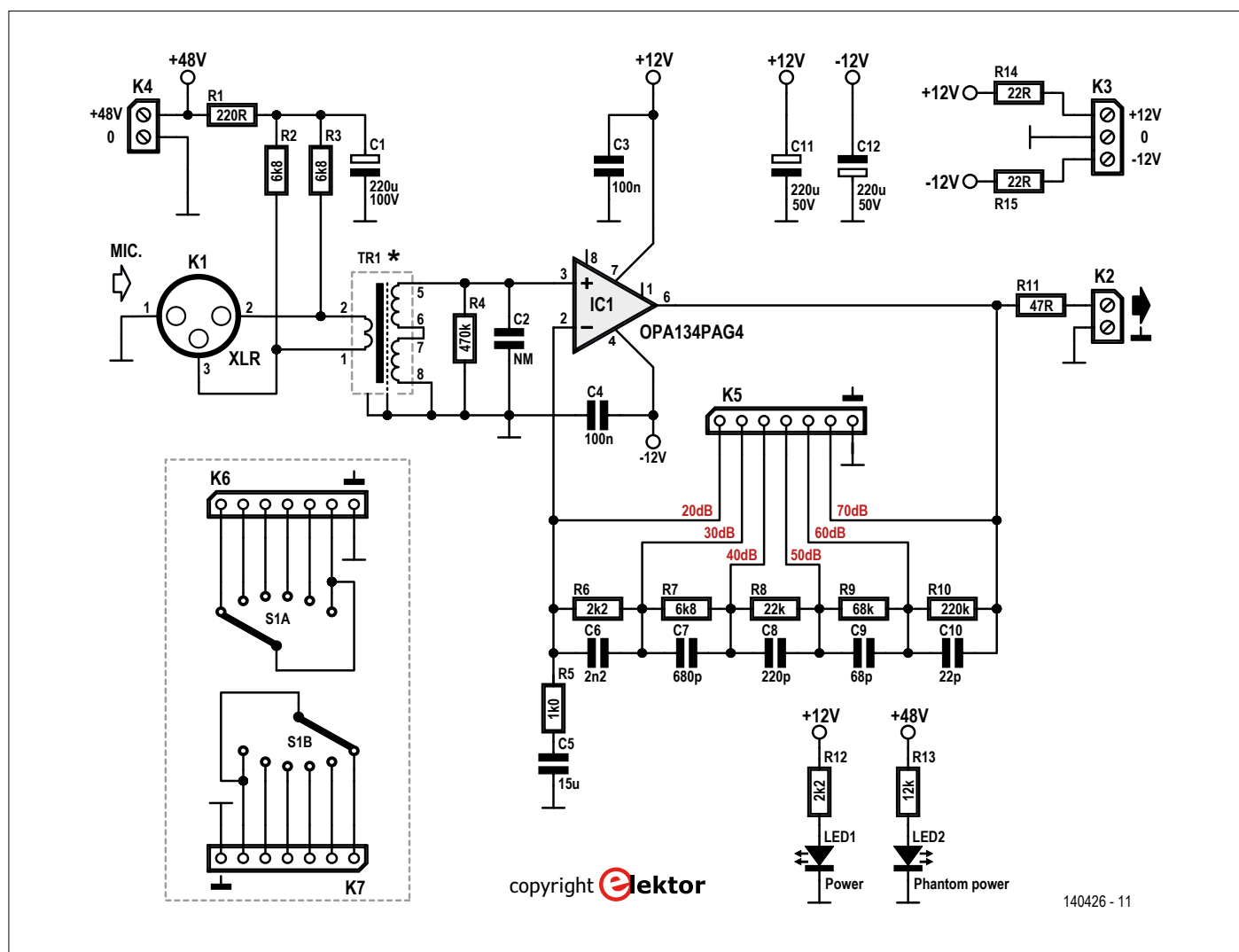


Figure 1. L'ampli op est le cœur de cet amplificateur.

de qualité professionnelle, mais plus onéreux. N'hésitez pas à chercher ailleurs sur la Toile. Retenez que le circuit imprimé peut accueillir plusieurs modèles pour TR1 : ÜP3095M et ÜP3096M de Pikatron, LL1935 de Lundahl et JT-115K-EPC de Jensen. Le transformateur LL1935 de Lundahl possède deux enroulements primaires et deux enroulements secondaires [note du labo : ces enroulements peuvent être utilisés en sens inverse (c'est clairement indiqué dans la feuille de caractéristiques, c'est probablement vrai pour les autres transformateurs)]. Revenons au schéma : pour appliquer le signal au primaire de TR1, il faut brancher un micro de préférence via un connecteur aux normes XLR usuelles dans le domaine professionnel ou par un *jack* stéréo de 6,35 mm. Dans ce cas, l'extrémité du connecteur correspond au point chaud (canal gauche), l'anneau intermédiaire au point froid (canal droit) et la partie restante à la masse. Les micros à condensateur nécessitent une tension d'alimentation de 48 V, appliquée via les résistances R2 et R3 de 6,8 k Ω . La valeur exacte de R2 et R3 n'est pas essentielle, mais ces deux résistances doivent être appariées avec une précision de 0,4 % ou mieux [7, 8]. Sortez l'ohmmètre pour les sélectionner. Elles doivent dissiper au moins 0,5 W pour éviter de griller en cas de court-circuit à la masse.

Le secondaire de TR1 est connecté d'une part à la masse, d'autre part à l'entrée positive de l'ampli op (IC1) ; l'entrée négative est reliée au réseau de rétroaction qui fixe le gain du préampli. Le commutateur rotatif à six positions relie la sortie de l'ampli op à son entrée négative, directement ou via les résistances R6 à R10 montées en série. Chaque pas augmente le gain de 10 dB. Comme le transformateur élève déjà le niveau de 20 dB, le gain du préampli peut varier entre 20 et 70 dB. Il est préférable que le commutateur soit de type « make before break » : la nouvelle connexion est fermée avant que la précédente soit ouverte. Cela permet d'éviter de brusques sauts du gain, voire un gain temporairement maximal accompagné d'un effet de Larsen. Les condensateurs en parallèle sur R6 à R10 limitent la bande passante du préampli à 32 kHz. Ce sont des modèles axiaux, au polystyrène, de qualité supérieure. Toutefois les empreintes sur le circuit imprimé permettent de monter des condensateurs

céramique ou à film plastique. Sur le secondaire de TR1, vous disposez de place pour, si nécessaire, monter un condensateur (C2) destiné à amortir des résonances indésirables du transformateur. Généralement vous pouvez omettre C2.

Encore quelques conseils : les condensateurs de découplage C3 et C4 de 100 nF seront installés aussi près que possible de l'ampli op. Je préconise d'utiliser des résistances à film métallique qui présentent une tolérance de 1 % ; c'est impératif pour R5 à R10. Vous pouvez omettre R4 (470 k Ω) si le micro ne requiert pas une charge à une impédance particulière. Dans le cas contraire, il faut choisir une valeur égale à cent fois l'impédance préconisée. Les condensateurs électrolytiques doivent supporter une tension de 25 V, à l'exception de C1 qui sera un modèle pour 63 ou 80 V. Les tensions d'alimentation de l'ampli op ne sont guère critiques pour autant qu'elles correspondent aux spécifications du fabricant.

Si vous n'utilisez que des microphones électrodynamiques, donc sans alimentation 48 V, les composants C1 et R1 à R3 ne sont pas nécessaires.

Le condensateur C5 a une valeur de 15 μ F, mais pour une fréquence de coupure plus basse (ou moins de variation de la phase dans la plage de fréquence basse), un condensateur de 22 μ F de la même série convient.

INFO SUR LE PROJET

audio

préamplificateur

microphone

débutant

connaissseur

expert

5 h (sans la mise en boîte)

rien de spécial

env. 270 € (boîtiers compris)

Alimentation ± 12 V

L'ampli op (IC1) est alimenté avec les tensions de +12 et -12 V produites par la partie basse du circuit de la **figure 3**. Cette alimentation construite avec deux régulateurs intégrés est ultra classique. Notez qu'il est judicieux d'installer les condensateurs de découplage de 100 nF aussi près que possible des régulateurs. Le redressement est assuré par un pont de diodes 1N4007. Le fusible F5

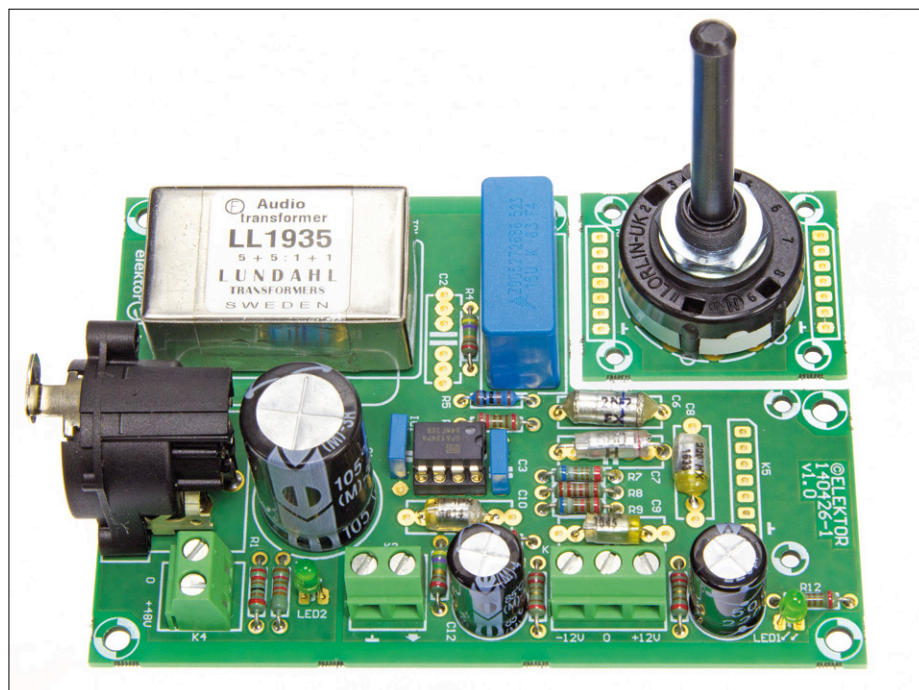


Figure 2. Circuits imprimés garnis de l'amplificateur.

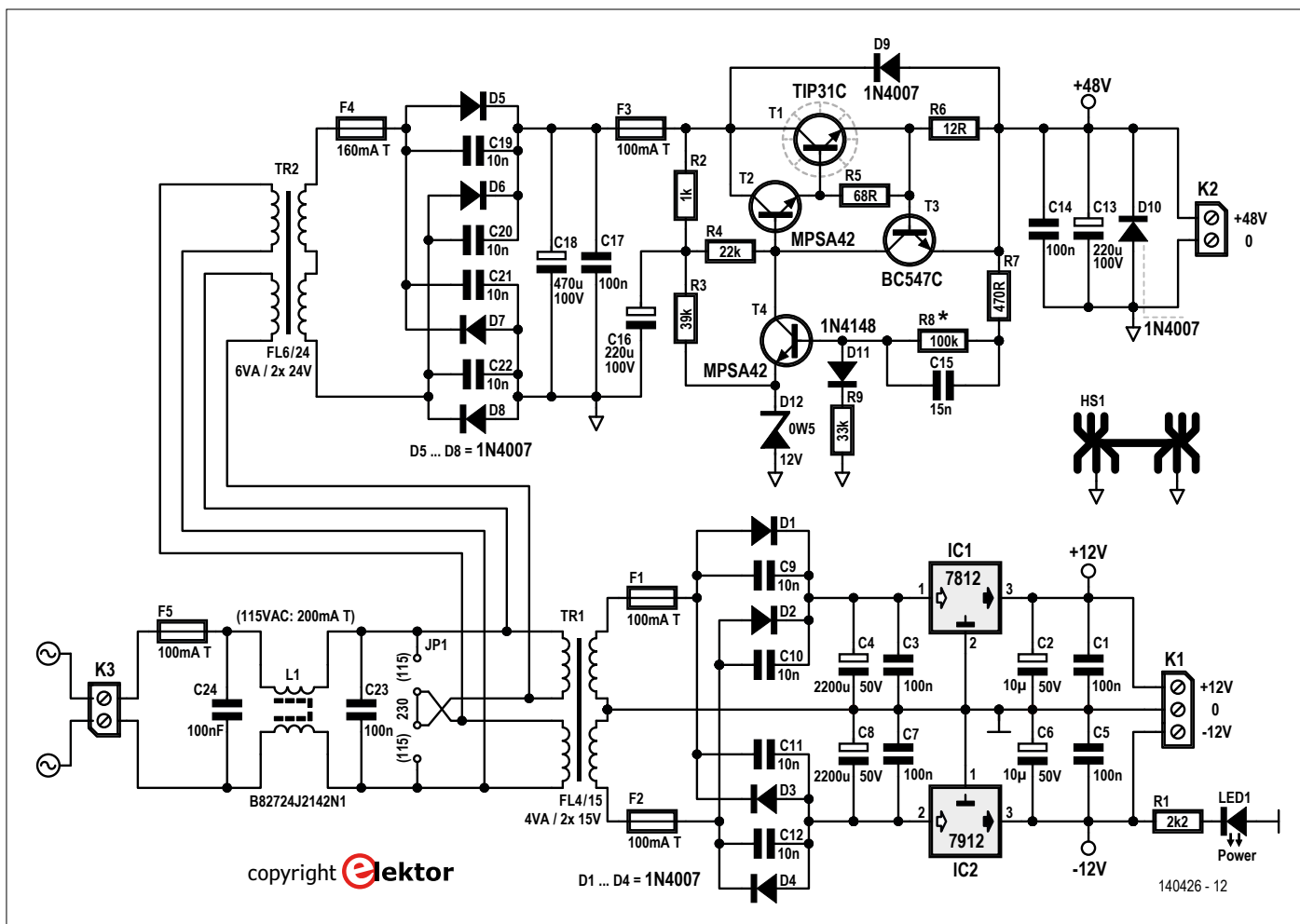


Figure 3. Schéma de la double alimentation : ± 12 V et 48 V.

qui protège le transformateur TR1 est dimensionné selon les prescriptions du constructeur. La diode électroluminescente LED1 montée sur la face avant de l'alimentation signale quand l'appareil

est en fonction. Certains exemplaires du régulateur 7912 ne délivrent leur tension nominale que s'ils sont chargés par un courant minimal de 5 mA. Il est donc pertinent de connecter la sortie du régu-

lateur IC2 à la masse du circuit via une résistance de 2,2 k Ω si la LED1 n'est pas montée. L'ampli op OPA134 consomme au maximum 5 mA à $T_{amb} = 25^\circ\text{C}$. Les régulateurs n'ont donc pas besoin de radiateur, même lorsque deux amplificateurs sont raccordés.

Les ordinateurs de bureau produisent tous des tensions de +12 et -12 V que l'on peut utiliser pour alimenter le préampli. Si vous choisissez cette solution, filtrez soigneusement ces tensions : elles seront certainement polluées par le bruit produit par la carte-mère et d'autres matériels. Il est prudent d'insérer des fusibles lents de 100 mA dans chacune des lignes d'alimentation afin de ne pas endommager l'ordinateur en cas d'incident.

Alimentation 48 V

La partie haute de la figure 3 montre l'alimentation qui fournit si nécessaire la tension de 48 V. La norme EN 61938 [8] prévoit pour les micros un courant de 10 mA tout au plus, et tolère un écart de ten-

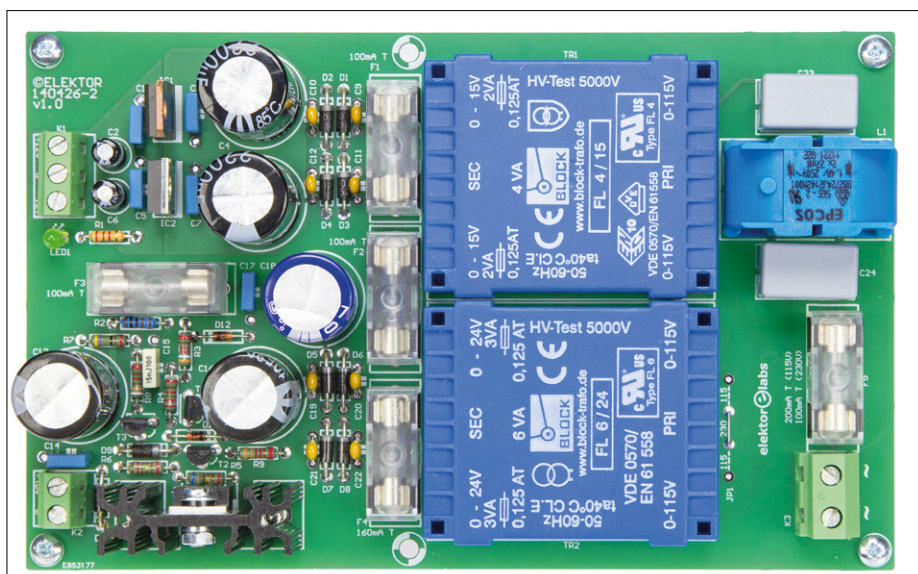


Figure 4. Circuit imprimé garni de la double alimentation.



LISTE DE COMPOSANTS – PRÉAMPLIFICATEUR

Résistances

Toutes à film métallique

R1 = 220 Ω , 1%, 250 mW
R2, R3 = 6,8 k Ω , 0,1%, 500 mW
R4 = 470 k Ω , 1%, 400 mW
R5 = 1 k Ω , 1%, 600 mW
R6, R12 = 2,2 k Ω , 1%, 400 mW
R7 = 6,8 k Ω , 1%, 400 mW
R8 = 22 k Ω , 1%, 250 mW
R9 = 68 k Ω , 1%, 250 mW
R10 = 220 k Ω , 1%, 250 mW
R11 = 47 Ω , 1%, 250 mW
R13 = 12 k Ω , 1%, 250 mW
R14, R15 = 22 Ω , 1%, 250 mW

Condensateurs

C1 = 220 μ F, 100 V, 20%, au pas de 5/7,5 mm, \varnothing_{max} 17,8 mm
C2 = non monté
C3, C4 = 100 nF, 100 V, 5%, au pas de 5/7,5 mm, PET
C5 = 15 μ F, 63 V, 10%, au pas de 5/7,5/10/15/22,5 mm, PET
C6 = 2,2 nF, 1%, 160 V, au pas de 5/7,5/10/14,6 mm, polystyrène, axial
C7 = 680 pF, 2,5%, 160 V, au pas de 5/7,5/10/14,6 mm, polystyrène, axial
C8 = 220 pF, 2,5%, 160 V, au pas de 5/7,5/10/14,6 mm, polystyrène, axial
C9 = 68 pF, 2,5%, 160 V, au pas de 5/7,5/10/14,6 mm, polystyrène, axial
C10 = 22 pF, ± 1 pF, 160 V, au pas de 5/7,5/10/14,6 mm, polystyrène, axial
C11, C12 = 220 μ F, 50 V, 20%, au pas de 5 mm, \varnothing_{max} 10 mm

Transformateur

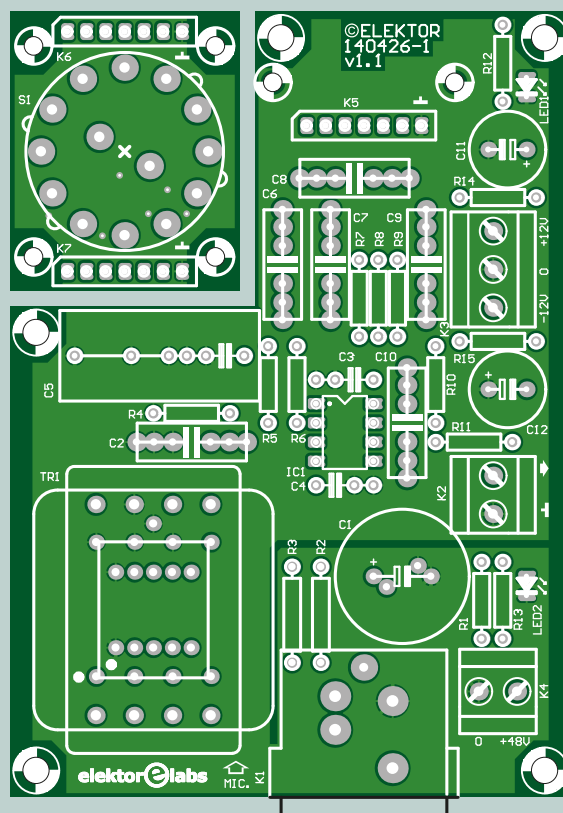
TR1 = LL1935, 1:10 (1:1:5+5), Lundahl, 38 \times 23 \times 16 mm

Semi-conducteurs

LED1 = LED, vert, 3 mm, T-1
LED2 = LED, bleu, 3 mm, à forte intensité, T-1
IC1 = OPA134PAG4, DIP-8

Divers

K1 = connecteur audio XLR, 3 contacts, femelle, montage sur circuit imprimé (Neutrik NC3FAAH1)
K2, K4 = bornier à 2 voies, au pas de 5,08 mm, 630 V
K3 = bornier à 3 voies, au pas de 5,08 mm, 630 V
K5 = barrette mâle à 7 contacts, au pas de 2,54 mm, coudé
K6, K7 = non monté (à l'opposé de K5)



commutateur rotatif à 2 pôles, 6 positions, commutation MBB, Lorlin CK1060

Circuit imprimé 140426-1

Boîtier 1455K1201, Hammond Manufacturing

Autres transformateurs adaptés au circuit imprimé :

TR1 = Pikatron ÜP 3095 M, 1:10, 18 \times 14,5 \times 13,2 mm (voir [3])

TR1 = Pikatron ÜP 3096 M, 1:15, 18 \times 14,5 \times 13,2 mm (voir [3])

TR1 = Jensen transformers JT-115K-EPC, 1:10, 31,75 \times 31,75 \times 23,62 mm (voir [6])

sion de ± 4 V autour de la valeur nominale. La tension du secteur est appliquée au transformateur TR2 qui délivre 2 \times 24 V. Le régulateur repose sur un schéma conventionnel et ne comprend que des éléments discrets. La diode Zener de 12 V constitue la référence de tension ; pour le réglage de la tension, le signal de rétroaction est appliqué au transistor T4 via R7 et R8. Le transistor T1, un TIP31C, est généreusement dimensionné afin de résister aux mauvais traitements. En outre T1 est monté sur un radiateur parce qu'il chauffe à cause de la tension d'entrée élevée du régulateur. Une résistance thermique d'environ 20 $^{\circ}\text{C}/\text{W}$ sera suffisante pour

refroidir T1. Recours ultime, le fusible lent F3 de 100 mA limitera sans aucun doute les dommages dans les situations désespérées... Le transistor T3 et la résistance R6 limitent à 50 mA l'intensité du courant produit par cette alimentation, c'est largement suffisant pour plusieurs micros. Les condensateurs électrolytiques de l'alimentation de 48 V doivent supporter une tension de service de 100 V. Enfin le labo d'Elektor a mesuré une ondulation résiduelle inférieure à 0,2 mVcc à 10 mA, éliminée par la carte de l'amplificateur.

Assemblage du tout

Les primaires des transformateurs TR1 et TR2 supportent les deux tensions de

secteur : 115 V et 230 V. Pour le 230 V, il faut un fil de liaison pour JP1 ; pour le 115 V, il en faut deux. Attention : il n'y a jamais trois fils, c'est clairement indiqué sur le circuit imprimé. Pour le 115 V, il faut doubler le calibre du fusible. Le fabricant des transformateurs ne recommande de mettre des fusibles qu'au secondaire. Toutefois, pour plus de sécurité, nous avons ajouté celui des primaires.

Pour habiller les deux circuits imprimés, le labo d'Elektor a choisi deux boîtiers de la série 1455 de Hammond (voir photo en début d'article). L'ampli est dans le petit boîtier.

Attention : l'enveloppe métallique du transformateur Jensen peut court-circuiter les pistes du circuit imprimé : lorsque vous soudez ce composant, laissez de l'espace entre lui et le circuit imprimé ou interposez un film isolant d'un demi-millimètre d'épaisseur.

Sécurité : les câbles de la prise secteur sont connectés directement sur le circuit imprimé, et aucun contact ne doit être établi avec le boîtier, ce qui garantit un isolement de classe II. Tous les fils, toutes les broches et toutes les pistes du circuit imprimé reliés directement ou indirectement

à la tension secteur doivent se trouver à au moins 6 mm des parties métalliques du boîtier. Sous le circuit imprimé, coupez aussi court que possible les fils et broches soumis à la tension secteur. Pour le câblage de l'alimentation, il faut utiliser du fil souple de 0,75 mm².



LISTE DE COMPOSANTS – ALIMENTATION

Résistances

R1 = 2,2 k Ω , 5%, 0,25 W, à couche de carbone
 R2 = 1,0 k Ω , 1%, 0,6 W, à film métallique
 R3 = 39 k Ω , 1%, 0,25 W, à film métallique
 R4 = 22 k Ω , 1%, 0,25 W, à film métallique
 R5 = 68 Ω , 1%, 0,4 W, à film métallique
 R6 = 12 Ω , 1%, 0,25 W, à film métallique
 R7 = 470 Ω , 1%, 0,25 W, à film métallique
 R8 = 100 k Ω , 1%, 0,25 W
 R9 = 33 k Ω , 1%, 0,25 W

Condensateurs

C1, C3, C5, C7, C14, C17 = 100 nF, 100 V, 5%, au pas de 5/7,5 mm, PET
 C2, C6 = 10 μ F, 50 V, 20%, au pas de 2 mm, \varnothing_{max} 6,5 mm.
 C4, C8 = 2200 μ F, 50 V, 20%, au pas de 5/7,5 mm, 17,8 mm max.
 C9, C10, C11, C12, C19, C20, C21, C22 = 10 nF, 200 V, 10%, au pas de 5,08 mm, céramique X7R
 C13, C16 = 220 μ F, 100 V, 20%, au pas de 5/7,5 mm, 17,8 mm max.
 C15 = 15 nF, 100 V, 5%, au pas de 5/7,5 mm
 C18 = 470 μ F, 100 V, 20%, au pas de 5/7,5 mm, 17,8 mm max.
 C23, C24 = 100 nF, X1, 440 VAC, au pas de 10/12,5/15 mm, polypropylène

Inductance/Transformateur

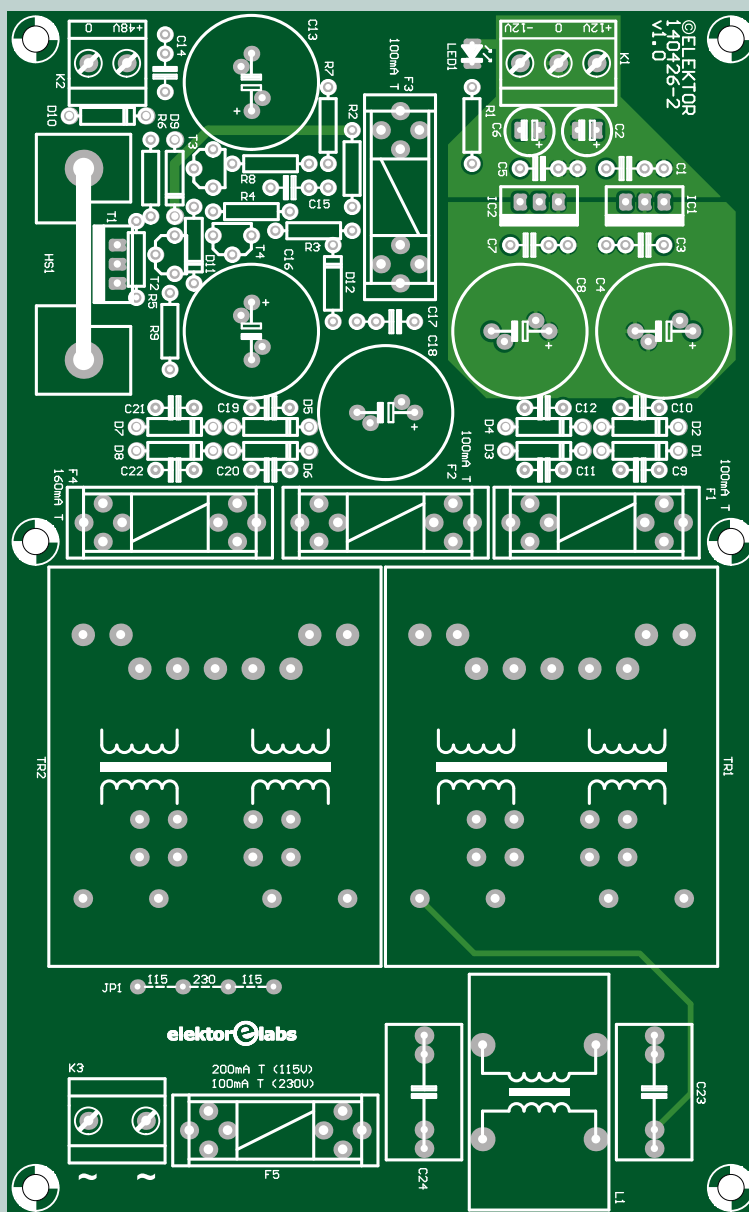
L1 = B82724J2142N1, bobine d'arrêt, mode commun, 2 \times 27 mH, 1,4 A
 TR1 = FL4/15 Block, prim. 2 \times 115 V, sec. 2 \times 15 V, 2 VA, montage sur circuit imprimé
 TR2 = FL6/24 Block, prim. 2 \times 115 V, sec. 2 \times 24 V, 6 VA, montage sur circuit imprimé

Semi-conducteurs

D1, D2, D3, D4, D5, D6, D7, D8, D9, D10 = 1N4007, 1000 V, 1 A, DO-41
 D11 = 1N4148, 100 V, 200 mA, DO-204AH
 D12 = BZX55C12, diode zener 12 V/0,5 W, DO-35
 LED1 = LED, vert, 3 mm, T-1
 T1 = TIP31C, NPN, TO220 (100 V/40 W/3 A)
 T2, T4 = MPSA42, NPN, TO-92 (300 V/625 mW/500 mA)
 T3 = BC547C, NPN, TO-92 (45 V/500 mW/100 mA)
 IC1 = 7812, TO-220-3
 IC2 = 7912, TO-220AB-3

Divers

JP1 = fil de 0,6 mm (voir texte)
 K1 = bornier à 3 voies, au pas de 5,08 mm, 630 V
 K2 = bornier à 2 voies, au pas de 5,08 mm, 630 V
 K3 = bornier à 2 voies, 7,62 mm, 630 V
 F1, F2, F3, F5(*) = fusible, 100 mA, à action retardée, 250 V, 20 \times 5 mm



F4 = fusible, 160 mA, à action retardée, 250 V, 20 \times 5 mm
 F1, F2, F3, F4, F5 = porte-fusible, 20 \times 5 mm, 500 V, 10 A
 F1, F2, F3, F4, F5 = cache pour porte-fusible 20 \times 5 mm
 HS1 = radiateur SW25-4 Aavid Thermalloy, 13 °C/W, 34,5 \times 12,5 \times 25 mm
 Circuit imprimé 140426-2
 *1 pour 115 VAC : 200 mA, à action retardée, 250 V, 5 \times 20 mm
 Boîtier 1455N1601, Hammond Manufacturing

Il ne reste plus qu'à confectionner le câble pour relier l'ampli et l'alimentation. Il est muni de connecteurs XLR à cinq pôles ; cela peut paraître extrême.

Une autre solution est d'utiliser deux câbles séparés, l'un pour la tension symétrique de ± 12 V, l'autre pour l'alimentation fantôme. ◀

(140426)

Bien entendu, vous pouvez avoir recours à quantité d'autres modèles de connecteurs.

LISTE DE COURSES

→ 140426-1
Circuit imprimé

→ 140426-2
Circuit imprimé

Connecteurs XLR et câble pour relier l'alimentation au préamplificateur		
Alimentation	Préampli de micro	Câble
Connecteur femelle XLR Neutrik NC5FD-LX, montage sur panneau (Farnell 1390124)	Connecteur mâle XLR Neutrik NC5MP, montage sur panneau (Farnell 250820)	Choix du labo d'Elektor : Pro Power PPCY4C0.5 (au mètre), blindé, 4 conducteurs de 0,5 mm ² , \varnothing_{ext} 6,5 mm
Connecteur mâle XLR Neutrik NC5MX, montage sur câble (Farnell 250764)	Connecteur femelle XLR Neutrik NC5FFX, montage sur câble (Farnell 1108204)	Autre : Pro Power PPCY4C0.75 (au mètre), blindé, 4 conducteurs de 0,75 mm ² , \varnothing_{ext} 7 mm

Courbes relevées par le labo d'Elektor avec son analyseur audio de précision

Conditions dans les deux cas :

- impédance de la source = 200 Ω ;
- gain min. = 19,4 dB ;
- tension de sortie max. = 7,8 V (THD 0,1%, gain 70 dB)

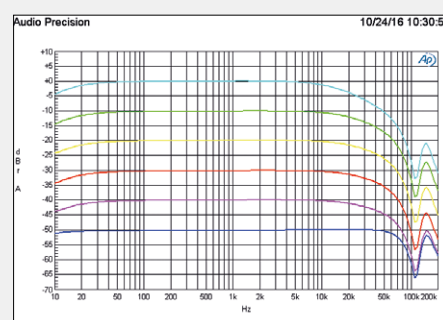
Couleur de la courbe en fonction du gain total : 70 dB (bleu clair), 60 dB (vert), 50 dB (jaune), 40 dB (rouge), 30 dB (rose) et 20 dB (bleu foncé).

Tableau 1. Mesure avec **Lundahl LL1935**

Gain	THD+N (source 200 Ω) 1 kHz, 1 V _{sortie} B 22 kHz	S/N (source = 200 Ω) par rapport à 1 V _{sortie} B 22 kHz	Fréquence de coupure à -3 dB basse/haute
20	0,0032 %	110 dB / 112,7 dBA	< 10 Hz / 74 kHz
30	0,0014 %	100 dB / 103 dBA	12,2 Hz / 37 kHz
40	0,0028 %	90,8 dB / 93,1 dBA	12,8 Hz / 34 kHz
50	0,0087 %	80,8 dB / 83 dBA	12,8 Hz / 34 kHz
60	0,027 %	71 dB / 73 dBA	12,8 Hz / 30,6 kHz
70	0,078 %	62 dB / 63,5 dBA	12,8 Hz / 17 kHz

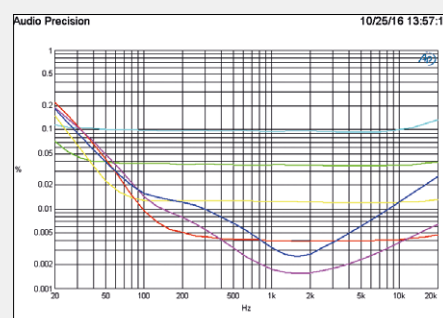
Amplitude en fonction de la fréquence pour les différentes valeurs du gain, sur la plage de 10 à 200 kHz.

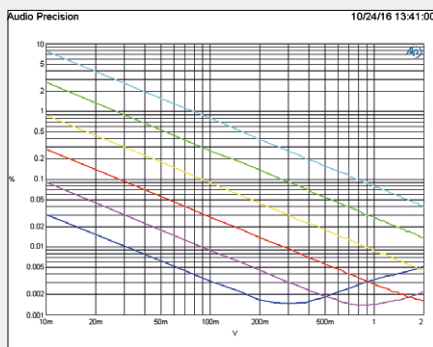
À 20 dB (bleu foncé), la courbe est presque celle du transformateur. À 151 kHz il y a un pic de résonance dont l'amplitude et la fréquence dépendent grandement de l'impédance et du type de la source (asymétrique ou symétrique). Pour les gains de 30, 40 et 50, les fréquences de coupure sont plus ou moins la même, mais à 60 et 70 dB, la bande passante au gain unité de l'ampli op commence à apparaître. Avec le gain de 50 dB, la bande passante de l'OPA134 est égale à 8 MHz/316, soit 25,3 kHz ! Si vous voulez plus de bande passante avec les gains les plus élevés, le LME49710 (produit gain-bande passante = 55 MHz) ou l'OPA627 (produit gain-bande passante = 16 MHz) sont des options possibles. Remplacer le condensateur C5 par un 22 μ F abaissera la fréquence de coupure à 9,6 Hz et le gain relatif à 20 Hz de -1,48 dB à -0,87 dB.



Distorsion + bruit en fonction de la fréquence pour V_{sortie} = 1 V et les différentes valeurs du gain sur une bande passante de 80 kHz.

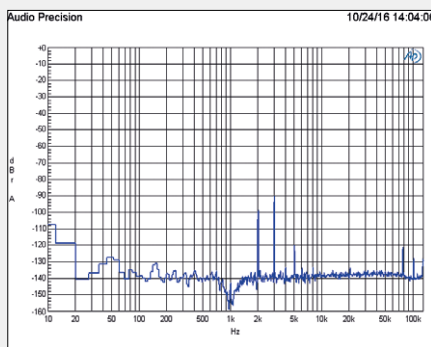
Pour que la tension de sortie reste constante quelle que soit la valeur du gain, on augmente le niveau du signal d'entrée en conséquence. Pour le gain le plus élevé, un niveau d'entrée de 335 μ V suffit pour obtenir 1 V en sortie. Avec le gain le plus élevé, il y a surtout du bruit, mais comme pour les gains plus faibles, la tension d'entrée augmente la distorsion et devient le facteur essentiel. L'augmentation de la distorsion aux extrémités des courbes pour les niveaux d'entrée plus élevés est due uniquement au transformateur.





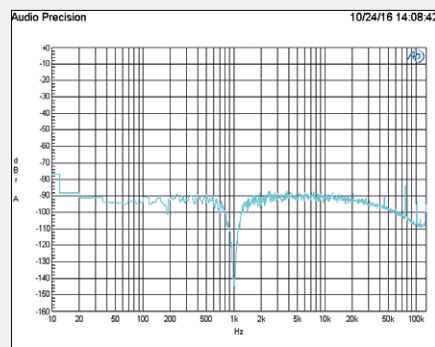
Distorsion + bruit en fonction du niveau de sortie à 1 kHz et pour une bande passante de 22 kHz.

La distorsion repart à la hausse aux extrémités des courbes pour les gains les plus élevés.



Transformée de Fourier rapide (FFT) à 1 kHz pour le gain total le plus faible (20 dB) et $V_{\text{sortie}} = 1$ V.

Il y a essentiellement un 3^e harmonique à -90 dB, responsable du résultat « distorsion harmonique totale + bruit » égal à 0,0032%.



FFT à 1 kHz pour un gain total plus élevé (70 dB) et $V_{\text{sortie}} = 1$ V.

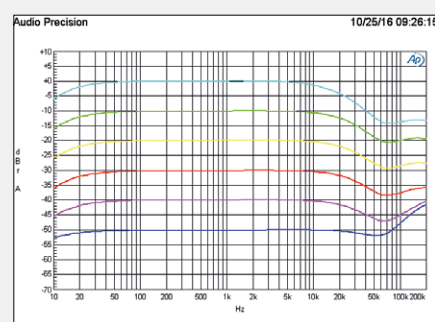
Comme toutes les fréquences (lire « bruit ») sont proches du niveau de 90 dB, « distorsion harmonique totale + bruit » est égal à 0,078%.

Tableau 2. Mesure avec **Pikatron ÜP3095M**

Gain	THD+N (source 200 Ω) 1 kHz, 1 V_{sortie} B 22 kHz	S/N (source = 200 Ω) par rapport à 1 V_{sortie} B 22 kHz	Fréquence de coupure à -3 dB basse/haute
20	0,0034%	109 dB / 112 dBA	< 10 Hz / ? > 200 kHz)
30	0,0016%	100 dB / 102 dBA	14,5 Hz / 30 kHz
40	0,0031%	90,6 dB / 92,6 dBA	15,3 Hz / 28,3 kHz
50	0,0097%	80 dB / 82,2 dBA	15,3 Hz / 26,2 kHz
60	0,030%	70,4 dB / 72,5 dBA	15,3 Hz / 23,3 kHz
70	0,086%	61,1 dB / 62,7 dBA	15,3 Hz / 16,2 kHz

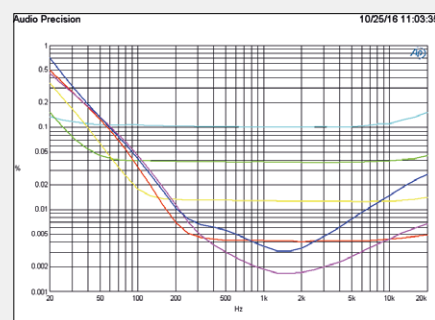
Amplitude en fonction de la fréquence pour les différentes valeurs du gain, sur la plage de 10 à 200 kHz.

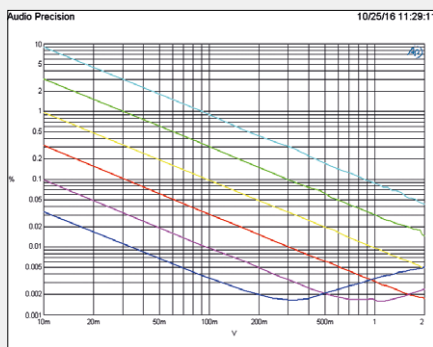
Il y a un pic de résonance juste après 200 kHz. La bande passante totale est inférieure à celle observée avec le transformateur Lundahl, mais satisfaisante pour la pratique. Remplacer le condensateur C5 par un 22 μ F abaissera la fréquence de coupure de 15,3 à 12 Hz.



Distorsion + bruit en fonction de la fréquence pour $V_{\text{sortie}} = 1$ V et les différentes valeurs du gain sur une bande passante de 80 kHz.

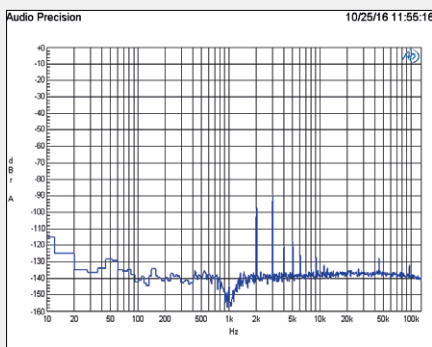
Ce n'est qu'aux fréquences les plus basses que la somme « distorsion totale + bruit » est supérieure à celle observée avec le transformateur Lundahl.





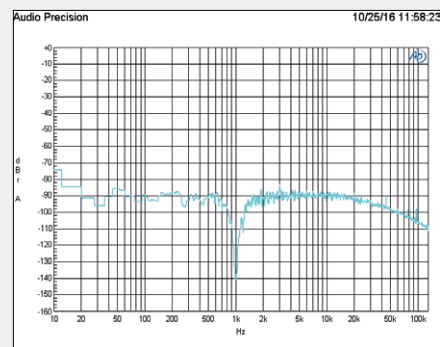
Distorsion + bruit en fonction du niveau de sortie à 1 kHz et pour une bande passante de 22 kHz.

Il est difficile de relever une différence avec la courbe du Lundahl. Les niveaux sont légèrement supérieurs.



Transformée de Fourier rapide (FFT) à 1 kHz pour le gain total le plus faible (20 dB) et $V_{\text{sortie}} = 1 \text{ V}$.

Le 3^e harmonique est le facteur essentiel et le reste des harmoniques est un peu supérieur à ce qui est observé sur la courbe du Lundahl. La somme « distorsion harmonique totale + bruit » est égale ici à 0,0034%.



FFT à 1 kHz pour un gain total plus élevé (70 dB) et $V_{\text{sortie}} = 1 \text{ V}$.

Le 3^e harmonique est à peine visible et le reste n'est que du bruit. La somme « distorsion harmonique totale + bruit » est égale à 0,086%

Analyse du bruit du préamplificateur

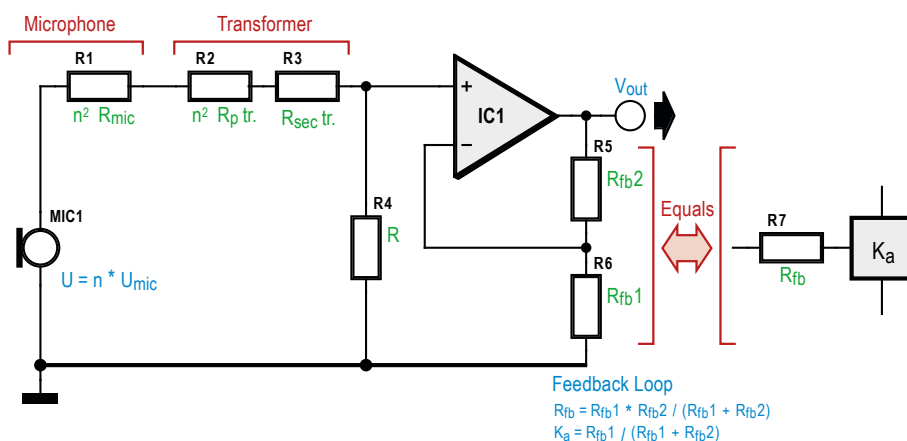


Figure. Schéma équivalent utilisé pour l'analyse du bruit du préampli.

1. Théorie

Les sources de bruit considérées sont d'une part le bruit thermique inhérent aux composants résistifs, d'autre part le bruit produit par les éléments actifs, ici l'amplificateur opérationnel. D'autres sources de bruit, telles que celles résultant de champs électromagnétiques ou électrostatiques, ne sont pas prises en compte. Il est d'ailleurs usuel d'en réduire les nuisances, par ex. grâce à un blindage adéquat.

La **figure** représente le schéma équivalent utilisé pour l'analyse du bruit du préamplificateur, dû à deux sources : les effets thermiques dans les différentes résistances ainsi que la composante causée par le composant actif, ici l'ampli op. Seules les parties réelles des impédances contribuent effectivement au bruit.

La source de signal est constituée par le micro, qui présente la résistance interne R_{mic} et produit le signal U_{mic} , appliqué via le transformateur T1 à l'entrée positive de l'ampli op U1. Chaque enroulement du transformateur présente une résistance interne : $R_{p\ tr}$ au primaire, $R_{sec\ tr}$ au secondaire. Le transfo est également caractérisé par le rapport de transformation n entre le primaire et le secondaire. Vue depuis l'entrée positive de l'ampli op, la résistance totale de la branche du circuit qui comprend le micro et le transfo est donc égale à :

$$(1) \quad R_a = n^2 \cdot (R_{mic} + R_{p\ tr}) + R_{sec\ tr}$$

Cette résistance est le cas échéant mise en parallèle avec R4

lorsque cette dernière est utilisée. Il en résulte une résistance totale égale à :

$$(2) \quad R_{inp\ pos} = \frac{1}{\frac{1}{R_a} + \frac{1}{R_4}} = \frac{1}{\frac{1}{n_2 \cdot (R_{mic} + R_{p\ tr}) + R_{sec\ tr}} + \frac{1}{R_4}}$$

L'entrée négative de l'ampli op est reliée aux résistances R_{fb1} et R_{fb2} . Ce réseau de rétroaction se caractérise par une atténuation K_a et une résistance équivalente valant respectivement :

$$(3) \quad K_a = \frac{R_{fb1}}{R_{fb1} + R_{fb2}}$$

Et

$$(4) \quad R_{fb} = \frac{1}{\frac{1}{R_{fb1}} + \frac{1}{R_{fb2}}}$$

Ramené aux entrées de l'ampli op, le bruit thermique sera alors égal à :

$$(5) \quad e_{nth} = \sqrt{4 \cdot k_b \cdot T \cdot (R_{inp\ pos} + R_{fb}) \cdot B}$$

avec k_b la constante de Boltzmann, T la température en kelvins et B la bande passante du signal.

La tension de bruit imputable à l'ampli op sera égale à :

$$(6) \quad e_{n\ comp\ actifs} = \sqrt{\left\{ e_{n\ op}^2 + [i_{n\ op} \cdot (R_{inp\ pos} + R_{fb})]^2 \right\} \cdot B}$$

avec respectivement $e_{n\ op}$ et $i_{n\ op}$ les densités de tension et de courant de bruit publiées dans les fiches techniques de ce composant. Il en résulte finalement la tension de bruit globale :

$$(7) \quad e_{n\ totale} = \sqrt{e_{nth}^2 + e_{n\ comp\ actifs}^2}$$

Comme les sources de bruit sont décorréliées, la tension qui résulte de leur combinaison n'est pas calculée par une simple addition, mais en extrayant la racine carrée de la somme de leurs amplitudes élevées au carré, ceci revient à dire que dans ce cas, ce sont les puissances individuelles qui entrent en jeu.

2. Exemple chiffré

Intéressons-nous aux valeurs numériques du schéma de la figure 1 pour comparer quatre amplis ops : OPA134 de Burr Brown, AD743 d'Analog Devices, NE5534 et TL071. Nous considérons une bande passante de 30 kHz, grosso modo celle du préamplificateur.

L'auteur de l'article manifestant une évidente préférence pour les chaudes après-midis d'été : la température sera donc de 300 K, ou 26,85 °C si vous préférez. Le rapport de transformation n est égal à 10 ; les résistances au primaire et au secondaire valent respectivement 50 et 1500 Ω. Nous supposons que le micro présente une résistance interne de 100 Ω, valeur réaliste pour ce genre d'équipement. Le niveau de bruit est calculé avec la plus grande valeur de R_{fb} qui correspond au gain maximal. Sur le prototype du préampli, R_a n'est pas utilisée. Le bruit thermique dû aux résistances est calculé avec les équations

(1), (4) et (5) :

$$R_{inp\ pos} = 10^2 \cdot (100 + 50) + 1500 = 16500$$

et

$$R_{fb} = \frac{1}{\frac{1}{1000} + \frac{1}{319 \cdot 10^3}} = 996,87 \approx 997$$

On introduit ces valeurs dans l'éq. (5) :

$$e_{nth} = \sqrt{4 \cdot 1,38 \cdot 10^{-23} \cdot 300 \cdot (16500 + 997) \cdot 30000}$$

Après calcul, on obtient :

$$e_{nth} = 2,95 \cdot 10^{-6} \text{ V}$$

La contribution au bruit de l'ampli op est déterminée avec l'éq. (6), et sa valeur totale ramenée à l'entrée avec l'éq. (7). Les résultats pour les quatre amplis ops sont présentés dans un tableau, c'est plus pratique pour la comparaison. Le rapport signal/bruit de la dernière ligne du tableau est donné pour une tension de 5 mV fournie par le microphone.

ampli op	AD743	OPA134	NE5534	TL 071
$e_{n\ op}$ (nV / $\sqrt{\text{Hz}}$)	2,9	8,0	4,0	18,0
$i_{n\ op}$ (pA / $\sqrt{\text{Hz}}$)	0,0069	0,003	1,5	0,010
$e_{n\ comp\ actif}$	502 nV	1,39 μV	5,24 μV	3,12 μV
$e_{n\ totale}$	2,99 μV	3,26 μV	5,47 μV	4,29 μV
(S+N)/N (dB)	84,46	83,72	79,23	81,33

À l'exception du NE 5534, tous ces amplis ops sont à entrée FET et se caractérisent par un courant de bruit à l'entrée négligeable. Les résultats obtenus avec l'ampli op OPA134 (fig. 1) sont très bons. Plus étonnant encore, ce sont les performances honorables du TL071. De type bipolaire, le NE5534 est pénalisé par son courant de bruit et donnera le meilleur de ce qu'il peut dans d'autres configurations.

3. Influence du transformateur sur le bruit thermique

L'éq. (1) peut encore s'écrire :

$$(8) \quad R_a = n^2 \cdot R_{mic} + n^2 \cdot R_{p\ tr} + R_{sec\ tr}$$

Il est évident que cette somme comprend deux termes, l'un pour le microphone, l'autre pour le transformateur, pour lequel on peut définir une résistance équivalente rapportée au secondaire :

$$(9) \quad R_{eq\ tranfo} = n^2 \cdot R_{p\ tr} + R_{sec\ tr}$$

La contribution du microphone et du transformateur au bruit thermique vaut alors :

$$(10) \quad e_{nth\ mic\ et\ transf} = \sqrt{4 \cdot k_b \cdot T \cdot (n^2 \cdot R_{mic} + R_{eq\ tranfo}) \cdot B}$$

Le transformateur multiplie la tension d'entrée U_{mic} fournie par le microphone par un facteur n . Le calcul du rapport signal/bruit au secondaire donne alors :

$$(11) \quad \frac{S+N}{N} = \frac{n \cdot U_{mic} + \sqrt{4 \cdot k_b \cdot T \cdot (n^2 \cdot R_{mic} + R_{eq\ transfo})} \cdot B}{\sqrt{4 \cdot k_b \cdot T \cdot (n^2 \cdot R_{mic} + R_{eq\ transfo})} \cdot B}$$

À titre de comparaison, considérons le même calcul pour un transformateur dépourvu de résistance interne :

$$(12) \quad R_{eq\ transfo} = 0$$

Après simplification, l'éq. (11) devient :

$$(13) \quad \frac{S+N}{N} = \frac{U_{mic} + \sqrt{4 \cdot k_b \cdot T \cdot R_{mic}} \cdot B}{\sqrt{4 \cdot k_b \cdot T \cdot R_{mic}} \cdot B}$$

Comparons les équations (11) et (13) : un transformateur sans résistance interne ne dégraderait ni n'améliorerait le rapport signal/bruit de la tension délivrée par le micro ; dans le cas d'un transformateur réel, ces résistances peuvent contribuer substantiellement à l'augmentation du bruit. Le choix de ce composant requiert donc une attention particulière.

Liens

- [1] www.analog.com
- [2] www.ti.com
- [3] www.buerklin.com (en anglais et allemand)
- [4] www.lundahl.se/our-products/microphones/
- [5] www.sowter.co.uk/pro-audio-transformers.php#
- [6] www.jensen-transformers.com/mic_in.html

- [7] *Mikrofonaufsätze* de Jörg WUTTKE, p. 83, en allemand (mais qui vaut la peine d'être lu !) : www.schoeps.de/documents/Mikrofonbuch_komplett.pdf
- [8] Norme EN 61938, systèmes audio, vidéo et audiovisuels - Interconnexions et valeurs d'adaptation - Valeurs d'adaptation recommandées des signaux analogiques.
- [9] Pages du projet : www.elektormagazine.fr/140426 et www.elektormagazine.fr/labs/1161





banc d'essai

microscope USB Andonstar V160

indispensable pour travailler avec les CMS

La fabrication manuelle de prototypes qui comportent des CMS peut vite tourner au cauchemar tellement certains CMS sont minuscules. Sans microscope, il est impossible de détecter les défauts d'assemblage. À côté des microscopes classiques, on trouve des microscopes USB qui permettent d'afficher l'image sur l'écran d'un PC ou d'une tablette. Le labo d'Elektor a essayé pour vous le microscope USB Andonstar V160.



Luc Lemmens (labo d'Elektor)

En soudure manuelle de CMS, le risque de défaut est élevé : soudure froide et pâte entre deux pastilles sont difficiles à éviter. Une bonne loupe, si possible éclairante, peut aider à découvrir de telles imperfections. Cependant, son grossissement est insuffisant pour détecter certains défauts. L'espace entre la loupe et le point de soudure est souvent si réduit qu'il est impossible d'y glisser la panne d'un fer à souder. Les fabricants de matériel optique professionnel proposent de quoi se sortir de ces situations épineuses. La palette s'étend des classiques microscopes binoculaires aux caméras de contrôle à

haute résolution et même aux appareils à rayons X spécialisés. Naturellement les prix sont tout aussi professionnels : haut de gamme ! Le microscope USB peut constituer une alternative bon marché. L'offre est tellement vaste que le choix n'est pas évident.

Immobilité parfaite

En utilisant un microscope USB, on se rend vite compte qu'il n'y a pas que la résolution et l'agrandissement qui comptent. Les qualités mécaniques ont une influence déterminante sur

son aptitude à rendre le service attendu. Les microscopes USB bon marché sont la plupart du temps équipés d'une mécanique peu robuste et peu stable si bien que la moindre vibration, ou le simple fait de toucher la caméra suffisent à déranger le cadre et/ou la mise au point de l'image.

Le microscope USB Andonstar V160 fait exception à la règle : sa platine massive garantit une image stable et nette. Les réglages sont précis et faciles. Le champ de vision dépend de la molette de réglage de la platine. La mise au point est confiée à une bague tournante placée sur le boîtier de la caméra. La caméra est petite et de forme étroite. Installée à quelques centimètres de l'objet, elle fournit des images nettes. Il reste alors suffisamment de place pour manipuler le fer à souder tout en regardant l'écran. L'Andonstar V160 permet même une visée oblique pour examiner les composants et les soudures latéralement. Dans une telle position, on a encore aussi plus de liberté de mouvement pour l'outillage.

Pour PC et tablettes

Les microscopes USB sont étroitement apparentés aux caméras électroniques, ils sont compatibles avec la plupart des logiciels d'imagerie pris en charge par les systèmes d'exploitation. L'Andonstar V160 est livré avec un CD qui contient le logiciel pour Windows. Le site d'Oasis Scientific [1] propose également les logiciels pour tous les systèmes d'exploitation habituels ou presque, Linux est le grand absent.

Nous avons essayé ce microscope USB avec une tablette Android, une association pertinente, surtout si la place sur l'espace de travail est mesurée. L'appli que nous avons utilisée s'appelle *mScope*, elle a donné toute satisfaction. Pour *mScope*, la tablette doit prendre en charge l'USB-OTG et le raccordement nécessite un câble d'adaptation USB-OTG. La plupart des applis pour caméra peuvent enregistrer des photos et même des petites vidéos, cela peut être utile pour documenter des projets.

Éclairage annulaire à LED

L'Andonstar V160 est équipé d'un éclairage annulaire à LED dont la luminosité est réglable par un bouton sur le câble de raccordement. Les petits objets peuvent être éclairés pratiquement sans zone d'ombre, mais il faut faire attention aux reflets provoqués par des surfaces réfléchissantes (comme des îlots de soudure ou d'autres surfaces brillantes sur les circuits imprimés). Dans ce cas, il est recommandé de couper l'éclairage annulaire et de placer une source de lumière diffuse à proximité du microscope.

Un outil indispensable

Au labo d'Elektor, depuis que nous avons l'Andonstar V160, les loupes restent dans les tiroirs, surtout pour souder les CMS. Avant d'arriver à coordonner le mouvement de vos outils tout en regardant l'écran, il faut un temps d'apprentissage. À la résolution maximale (1600 × 1200), il y a un léger retard de l'affichage qui force l'utilisateur à adapter ses gestes. Il faut en toutes circonstances garder un œil sur le fer à souder chaud pour ne pas trop l'approcher du microscope et risquer de l'endommager. Finalement, on s'habitue rapidement à cette observation indirecte via l'écran. Dans le coin CMS de notre labo, le microscope USB Andonstar V160 est devenu rapidement indispensable. ◀

(160367 – version française : Yves Georges)



Fig. 1. L'Andonstar V160 en action, associé à une tablette Android.

▶ Seule une platine massive garantit des images sans tremblement.

Lien

[1] www.oasis-scientific.com/downloads.html

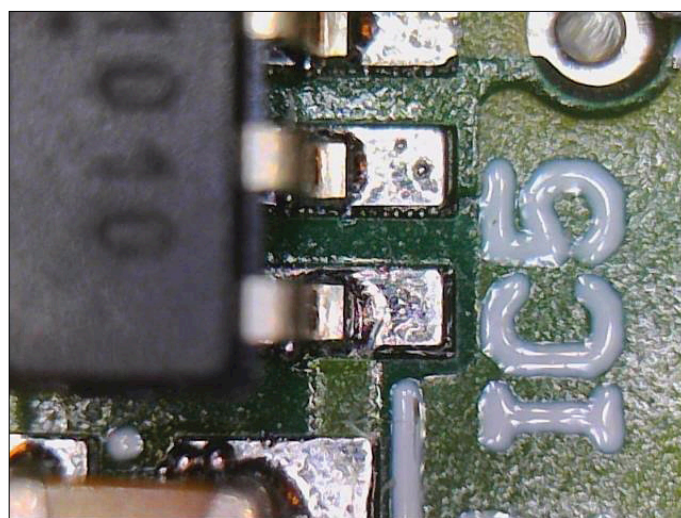


Fig. 2. Broches d'un circuit intégré de type CMS affichées sur l'écran avec un fort grossissement.

le bus I²C

1^{re} partie : le protocole

Josef Möllers

Le bus *Inter-Integrated Circuit*, en abrégé bus I²C ou encore TWI (*Two Wire Interface*) chez certains fabricants, permet de relier facilement deux puces à l'aide de deux fils quand il y a peu de données à transmettre et à petite vitesse. En trois épisodes, nous examinerons ses caractéristiques essentielles et les moyens de le mettre en œuvre avec Raspberry Pi, Arduino et d'autres systèmes.



Il n'est pas nécessaire d'aller dans les détails techniques pour découvrir à quel point ce bus I²C est d'emploi facile. Nous examinerons d'abord le protocole de transmission des données, puis nous ferons des essais de communication avec différents systèmes, depuis Arduino et RPi jusqu'au PC lui-même et finalement, nous irons à la rencontre de quelques puces I²C spécialisées et verrons comment lever une panne de transfert d'information.

C'est Philips Semiconductors qui a conçu le bus I²C dans les années 1980. Si vous ne tenez pas à aborder le bus I²C comme une boîte noire associée à des bibliothèques de logiciel tout fait, mais que vous voulez tout savoir jusqu'au niveau du bit,

vous trouverez la spécification officielle complète de NXP (qui a repris Philips Semiconductors) sur [1] et dont la lecture est évidemment recommandée.

Il y a sur l'internet une infinité de ressources sur le thème du bus I²C. Citons par exemple le site en anglais [2] de la société hambourgeoise *telos Systementwicklung GmbH* où l'on trouve à peu près tous les aspects de la communication sur ce bus.

Petit détour par la couche physique

En général, un microcontrôleur programmable commande des blocs périphériques à fonction précise, comme extension de port, horloge ou EEPROM. Le contrôleur est le maître, il pour-

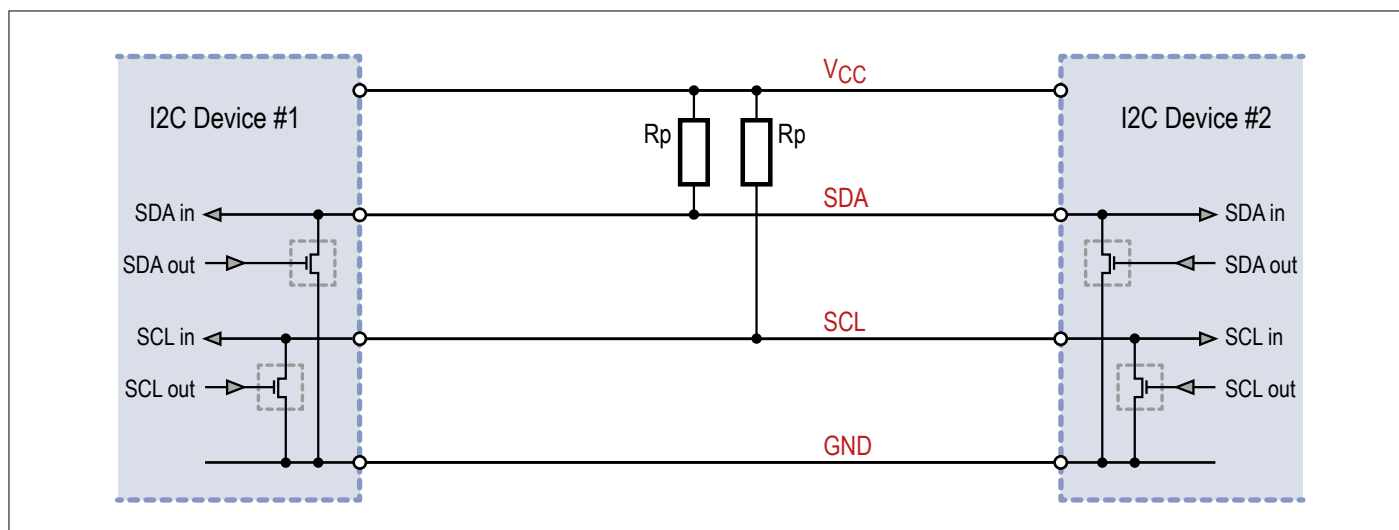


Figure 1. Le bus I²C constitue une porte ET câblée.

voit au signal d'horloge, tandis que les puces périphériques sont les esclaves. Par I²C, on peut aussi faire converser deux microcontrôleurs, si l'un occupe la fonction de maître. Quand il faut commander plusieurs puces esclaves, tous les blocs, en rang d'oignons, partagent les mêmes lignes de bus. Mais on peut aussi connecter plusieurs maîtres au même bus I²C. L'une des deux lignes du bus véhicule le signal d'horloge SCL, l'autre le signal de données SDA. Un câble plat à quatre conducteurs suffit ainsi pour commander les périphériques et les alimenter. Pour limiter la diaphonie, mieux vaut utiliser pour la masse le fil situé entre ceux des deux signaux.

Les deux lignes adhèrent à la logique positive : une tension élevée sur SDA représente un 1 et pour une impulsion d'horloge, la ligne SCL est aussi placée brièvement au niveau haut. Au repos, les résistances de polarisation haute maintiennent les deux lignes au niveau haut. Les tampons dans les puces I²C sont en configuration à collecteur ouvert ou à drain ouvert de manière à ce que les deux lignes constituent toujours une grande porte ET câblée (**figure 1**). Une ligne ne peut être haute par rapport au point commun que si **toutes** les sorties sont au niveau haut. Si l'une des sorties passe au niveau bas, la ligne commune est également au niveau bas. Chaque membre du bus doit donc vérifier le niveau des lignes : s'il veut y mettre un 1, il doit être en mesure d'exclure la possibilité qu'un autre membre y place un 0 en même temps.

La grandeur de la résistance est peu critique, elle devrait se situer dans la plage de quelques milliers d'ohms. Sur RPi, les résistances de polarisation (1,8 kΩ vers le 3,3 V) sont déjà installées à demeure. Sur les microcontrôleurs Atmel de la série ATmega, ces résistances sont commutées par logiciel, il ne faut donc pas en ajouter, puisqu'elles valent déjà entre 20 et 50 kΩ. Selon la spécification, le niveau haut doit atteindre au moins 0,7 V et le niveau bas être inférieur à 0,3 V ; en pratique, les puces de 5 V fonctionnent aussi avec une résistance vers le 3,3 V, par exemple sur un RPi. Si d'aventure un circuit ne fonctionnait pas, cela peut provenir de différences de niveau, il faut alors installer un convertisseur de niveau.

Il existe aussi des modules esclaves à résistances de polarisation intégrées. Il faut les enlever ou ne pas les monter s'il s'agit d'un kit, parce que quand plusieurs esclaves mettent ensemble leurs résistances, elles se retrouvent en parallèle et la résistance résultante est alors trop petite. Sur un bus I²C, il ne faut qu'une paire de résistances de polarisation, de préférence du côté du maître.

Il faut aussi faire attention quand on mélange des blocs de tensions différentes. Certains de ces composants alimentés sous 3,3 V ne supportent pas une tension supérieure à l'entrée. Relier les résistances de polarisation au +5 V risque de détériorer la puce. Pour que le bus travaille en toute sécurité, il y a lieu de s'en tenir aux niveaux prescrits. Pour assembler sur le même bus des composants de 5 V et de 3,3 V, des convertisseurs de niveau bidirectionnels sont nécessaires, comme ceux décrits sur [3]. Deux MOSFET et les résistances nécessaires relient alors la partie sous 5 V à celle sous 3,3 V sur le bus (**figure 2**).

Le protocole

Le bus I²C est un simple bus maître/esclave sur lequel la communication démarre toujours d'un maître qui s'adresse à un esclave avec lequel il échangera des données. Un esclave ne

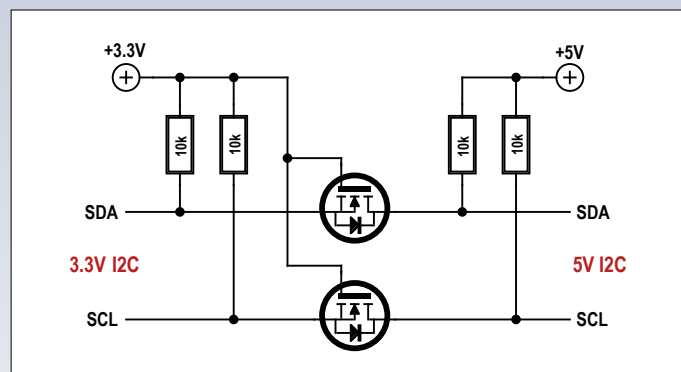


Figure 2. Des MOSFET compartimentent le bus entre la partie sous 5 V et celle sous 3,3 V.

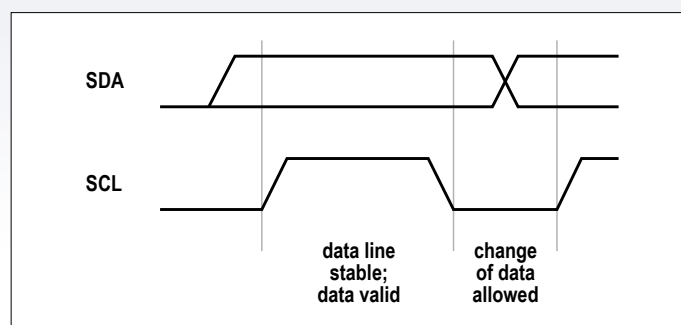


Figure 3. Diagramme temporel d'un transfert de données.

peut pas entreprendre lui-même une conversation comme il est possible de le faire en SCSI, par exemple. Mais un microcontrôleur peut aussi bien jouer le rôle de maître que d'esclave. En théorie, il serait possible d'inverser les rôles en cours d'exécution, mais d'habitude, un membre du bus est soit l'un, soit l'autre.

C'est toujours le maître qui décide de la vitesse de transmission. Il y a officiellement quatre débits binaires possibles :

- 100 Kb/s (mode standard, Sm, par défaut),
- 400 Kb/s (mode rapide, *Fast mode*, Fm),
- 1 Mb/s (mode plus rapide, Fm+) et
- 3,4 Mb/s (mode à haute vitesse *High Speed*, Hs).

Le maître peut transmettre des données à l'esclave, et inversement, mais c'est toujours le maître qui délivre le signal d'horloge sur la ligne SCL. Le transfert de données (**figure 3**) se déroule donc synchronisé par l'horloge. Pour envoyer un bit, l'émetteur met le bit de donnée sur la ligne SDA, en logique positive. Le maître relâche la ligne d'horloge SCL qui est tirée au niveau haut et après un certain temps la ramène au niveau bas. C'est ainsi qu'un bit est transmis et l'émetteur peut envoyer le bit suivant. Pendant la transmission, aussi longtemps que la ligne SCL reste haute, la ligne SDA ne peut pas varier.

1. Pour lancer la transmission, le maître abaisse d'abord la ligne de données SDA, puis celle d'horloge SCL (**figure 4**). Cette séquence s'appelle la condition de départ, *START condition*.

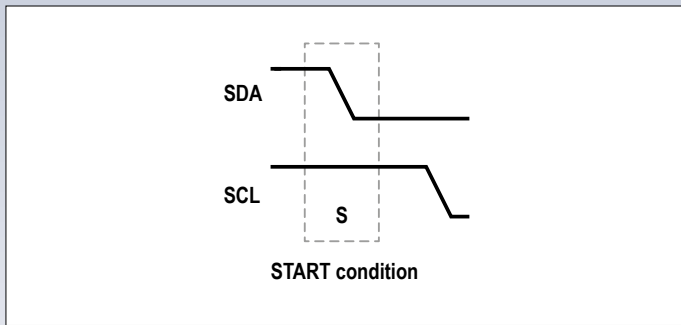


Figure 4. Pour démarrer une communication, il faut remplir la condition de *START*.

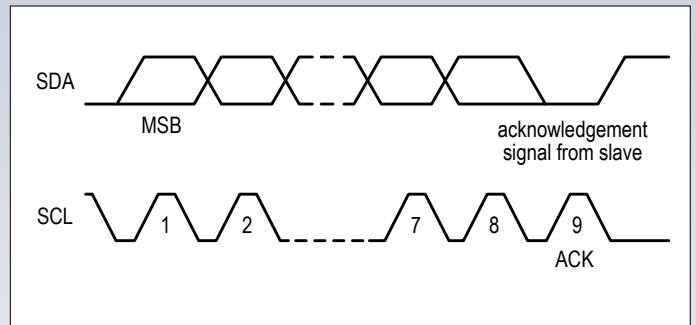


Figure 5. La manière de passer les bits d'adresse.

2. Après quoi, le maître, comme l'indique la **figure 5**, envoie d'abord l'adresse de l'esclave, bit de poids fort en tête (MSB), suivie du bit de lecture/écriture (L/E) (0 = donnée de l'esclave au maître ou 1 = donnée du maître à l'esclave). L'adresse d'esclave compte généralement sept bits (le standard I²C permet aussi des adresses plus longues sur 10 bits, mais c'est rarement utilisé). Le fabricant des blocs périphériques fournit lui-même l'adresse, en partie du moins. La plupart du temps, le fabricant laisse à disposition quelques bits que l'on peut définir soi-même par câblage extérieur de manière à associer sur le bus plusieurs blocs identiques en leur conférant des adresses différentes. Les adresses des composants périphériques sont indiquées dans les fiches techniques, mais on peut les obtenir avec un outil tel que [i2cdetect](#). Au total, l'adresse d'esclave et le bit de L/E font 8 bits, donc un octet.

3. Après l'envoi de l'adresse, le maître transmet à coups d'horloge SCL les données utiles, toujours sous forme d'octets avec le bit de poids fort MSB en tête (**figure 6**). Si le bit de L/E dans l'octet d'adresse était un 0, le maître attend que l'esclave mette au bon moment les bits de donnée sur SDA pour produire lui-même les impulsions en mesure sur SCL. En revanche, si ce fameux bit était un 1, c'est au tour du maître de passer les bits de donnée sur SDA et toujours de produire les coups d'horloge. Pour chaque octet, y compris ceux d'adresse avec l'indicateur de direction L/E, chaque destinataire doit envoyer un neuvième bit comme accusé de réception. Un niveau bas pour une bonne arrivée (ACK), un niveau haut pour un déni de réception (NACK). Un NACK ne signifie d'ailleurs pas que la transmission a raté ou que les données étaient fausses, mais seulement que c'est la fin de la transmission. Quand il s'agit de l'octet d'adresse, un NACK signale qu'aucun esclave n'a reconnu l'adresse ou que la combinaison de l'adresse et du bit de L/E n'était pas valide.

4. À la fin de la transmission, le maître libère d'abord la ligne d'horloge SCL, puis la ligne de données SDA et donc les deux lignes retournent au niveau haut à cause de résistances de polarisation. Cette séquence (**figure 7**) s'appelle *STOP condition*. Le cas échéant, le maître peut abaisser en premier la ligne de données SDA.

Les conditions *START* et *STOP* sont des exceptions à la règle

qui veut que la ligne SDA ne puisse pas varier tant que la ligne SCL est au niveau haut.

Le nombre d'octets qu'une communication peut transmettre est en fait illimité. En revanche, des protocoles plus élaborés, comme *Power Management Bus* (PMBus) ou *System Management Bus* (SMBus), qui sont basés sur le bus I²C, définissent des limites supérieures ou des formats d'enregistrement, qui contiennent par exemple la longueur de la transmission. En fait, la transmission « sans frontière », n'importe quelle plateforme ou bibliothèque ne la permet pas, parce que généralement, il faut préciser dès le début de la transmission le nombre d'octets à recevoir. Par exemple, la bibliothèque *Wire* d'Arduino définit dans ce protocole un temps limite, par exemple pour la mise en pause de l'horloge (*clock stretching*, cf. plus loin) ou la détection d'erreur par contrôle de redondance (*Packet Error Checking*, *PEC*).

Une communication (entre *START* et *STOP*) est toujours unidirectionnelle. Le maître fixe le sens de la communication avec le bit de L/E pour toute sa durée.

Seules quelques puces ont une structure si élémentaire qu'elles ne peuvent communiquer que d'une seule manière, par exemple le PCF8574 (que nous utiliserons dans la dernière partie de cette série). Avec la plupart des composants I²C, le maître doit, avant de pouvoir lire, effectuer une opération d'écriture pour envoyer à l'esclave un paramètre tel qu'un numéro de registre ou une adresse mémoire. S'il veut ensuite envoyer des données à l'esclave, il peut le faire en effectuant une écriture à l'endroit désigné par le paramètre. Si le maître veut lire des données issues de l'esclave, il doit lancer une nouvelle opération de lecture. Pour éviter qu'un autre maître s'immisce dans la conversation en envoyant au même esclave un autre paramètre, le maître actif peut remplacer la condition de *STOP* par une nouvelle condition de *START*, ce qui s'appelle alors une condition de *REPEATED START* :

Start - Adresse+Write - Registernumber - (Repeated)
Start - Adresse+Read - Data- ... - Stop

De nombreux blocs périphériques, comme des mémoires ou le RV-8523 de RTC, incrémentent automatiquement le numéro de registre ou l'adresse mémoire après chaque accès, d'autres, comme le capteur de température LM75, ne le font pas. Avec les

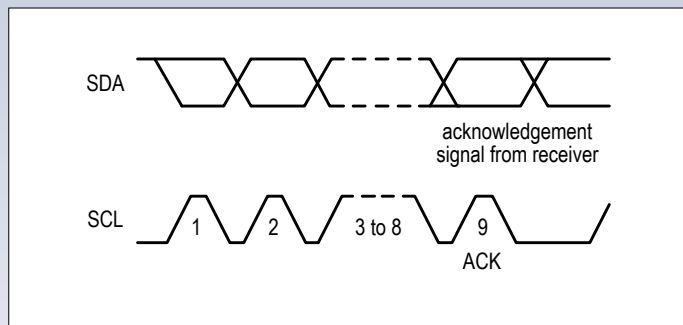


Figure 6. Le transfert des bits de données.

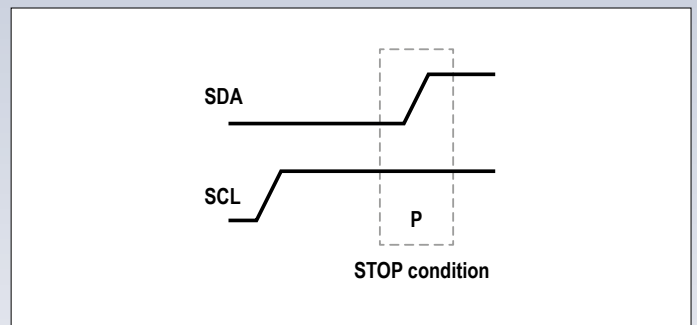


Figure 7. La clef pour clore la communication : *STOP*.

premiers, on peut dans une seule communication transmettre tout le banc de registres ou de mémoire, avec les autres, il faut répéter chaque fois l'adresse. D'un autre côté, quand on veut tout le temps lire le même registre, par exemple le registre de température du LM75, on peut se passer de répéter le numéro de registre. Les détails sur le sujet sont dans la fiche technique. Le protocole I²C ne prévoit pas les interruptions. Bien sûr, un esclave pourrait déclencher une interruption chez le maître par une ligne séparée. Après quoi, celui-ci appelle l'esclave par I²C, mais le protocole du bus ne dispose d'aucune norme dans ce but.

Mise en pause de SCL et arbitrage

Quand un esclave doit transmettre des données, il peut s'écouler un certain temps avant que la puce ne soit prête à le faire. L'esclave doit alors pendant ce délai arrêter ou différer l'horloge du maître. Comme les deux lignes de signal du bus forment une grande porte ET, l'esclave en a le pouvoir en passant momentanément la ligne SCL au niveau bas. Le maître s'en rend compte puisque SCL ne remonte pas quand il libère la ligne et il attend. Ce mécanisme s'appelle *Clock Stretching*, étirement du coup d'horloge, mais à proprement parler, ce n'est pas vrai, puisque l'esclave ne peut qu'allonger la pause entre deux impulsions d'horloge, pas l'impulsion elle-même.

Il est vrai qu'on peut raccorder plusieurs maîtres sur un bus I²C. Aussi, entre les conditions *START* et *STOP*, le bus affiche complet, aucun autre maître ne peut émettre de condition *START* pendant ce temps-là. Un maître (potentiel) doit donc continuellement tenir à l'œil le bus. Un maître ne peut émettre de *START* que quand le bus est libre. Imaginons que deux maîtres veuillent lancer un *START* au même instant. Il y a fort à parier que les deux adresses d'esclaves ou, au bout du compte, les bits de L/E seront différents et alors, un maître aura voulu transmettre un 1 et l'autre un 0. À cause du ET câblé, celui qui veut envoyer un 0 le verra sur le bus, tandis que l'autre, avec son 1, ne l'apercevra pas et devra cesser immédiatement la transmission. C'est donc celui qui a présenté un 0 qui est vainqueur, il ne remarque rien, acquiert tout de suite le contrôle du bus et poursuit sa transmission.

On trouve parfois à ce sujet le terme d'arbitrage, même dans de la documentation officielle. C'est inapproprié parce qu'il n'y a pas le moindre déphasage ou quoi que ce soit de comparable à des priorités ou des délais d'attente. Celui qui veut

transmettre attend simplement que le bus se libère pour commencer, tout en surveillant ce qui se passe sur le bus. On ne trouve rien d'autre dans la documentation de NXP.

Tout maître peut d'ailleurs à tout moment interrompre une communication avec une condition *de STOP*. Un récepteur aussi peut le faire d'un simple NACK dans l'octet d'accusé de réception, mais jamais un esclave comme émetteur. Même si l'esclave, de son côté, a cessé d'envoyer les données dont il dispose, le maître, imperturbable, continue à lire ce qui se trouve sur SDA, éventuellement sans signification.

Dans la deuxième partie

La prochaine fois, nous établirons des communications avec différents microcontrôleurs, comme ATmega et Arduino, mais aussi Raspberry Pi et un PC, utilisés comme maître I²C et esclaves. Des exemples de programmes en C et Python montreront comment s'adresser à un membre du bus, lui envoyer des commandes et recevoir ses réponses. ◀

(160095 – version française : Robert Grignard)



Liens

- [1] www.nxp.com/documents/user_manual/UM10204.pdf
- [2] www.i2c-bus.org/
- [3] <http://playground.arduino.cc/Main/I2Cbi-directionalLevelShifter>

retour sur l'analyseur de signal par Ethernet

plus rapide, toujours plus facile à utiliser et quelques nouveautés

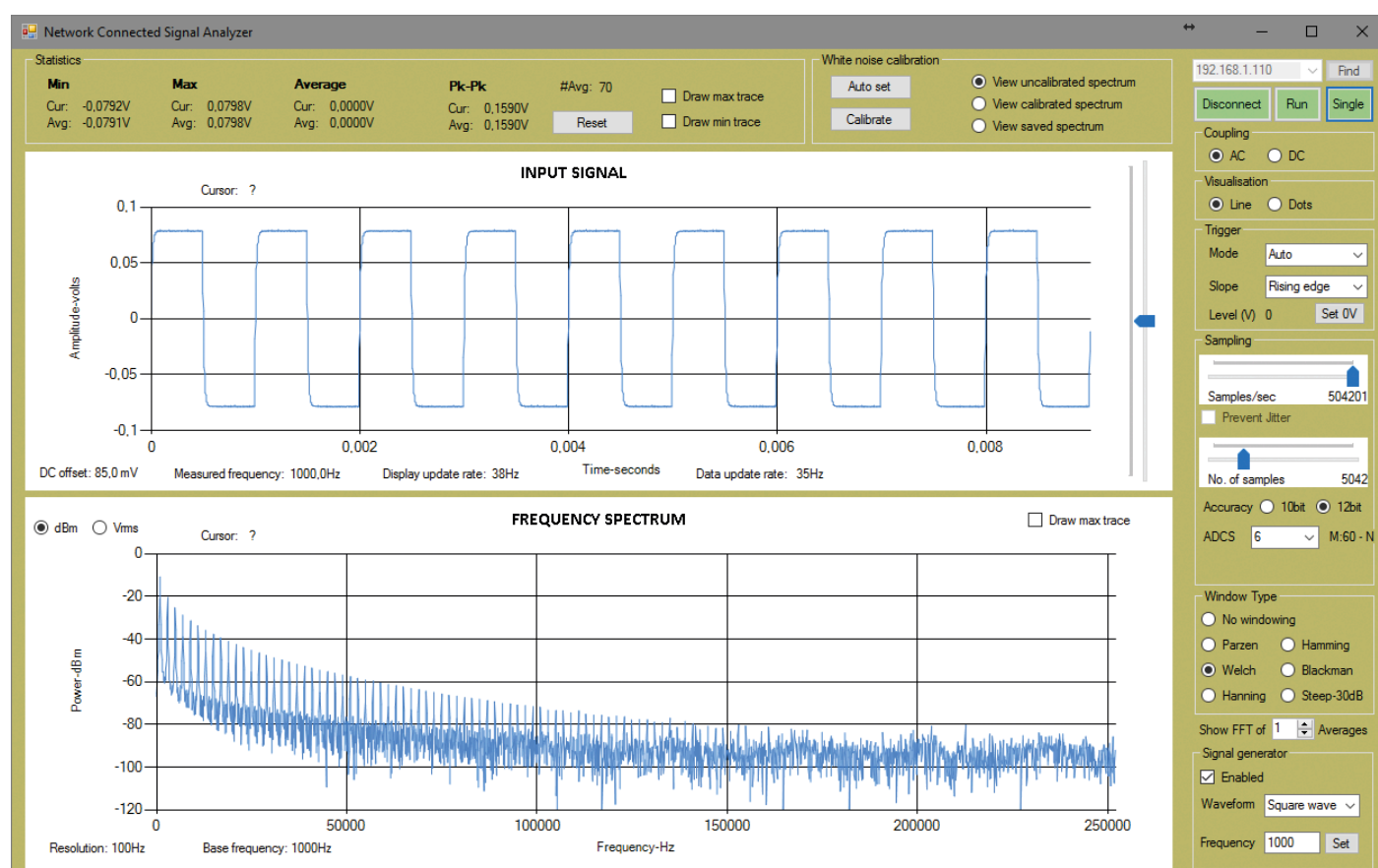


Figure 1. La nouvelle interface utilisateur.

En mars et avril 2016, Elektor vous proposait de réaliser un analyseur de spectre par Ethernet (ASPE). Son logiciel a été revu, ce qui permet – sans toucher au matériel – une nette amélioration des caractéristiques de l'instrument : nouvelles fonctions, rapidité accrue, connexion doublée et convivialité affinée. Voici l'ASPE II !

Joost Breed (Pays-Bas) et
Neal Martini (États-Unis)

Commençons par voir les améliorations de l'instrument.

Plus de connexions

Deux ASPE II peuvent être connectés en même temps et à chaud (*plug and play*) au même réseau local (*Local Area Network*). Pour ce faire, nous utilisons le protocole de configuration dynamique des hôtes (*Dynamic Host Configuration*

Protocol – DHCP) qui attribue automatiquement une adresse IP à chaque instrument.

Rapidité accrue

L'ASPE II est beaucoup plus rapide que son prédécesseur, ce qui permet des

taux de rafraîchissement plus élevés pour l'oscilloscope et la transformée de Fourier rapide (*Fast Fourier Transform* – FFT). Cette amélioration résulte essentiellement de la capacité multitâche (*multithreading*) de l'ordinateur client et de l'accès direct à la mémoire (*Direct Memory Access* – DMA) du dsPIC33.

Plus de fonctions

Nous avons étendu les capacités de l'analyseur de forme d'onde synthétique (*Synthetic Waveform Analyzer* – SWA) avec des fonctions définies par l'utilisateur (*User Defined Functions* – UDF). L'oscilloscope dispose également de nouvelles fonctions : couplage CA ou CC, déclenchement (*triggering*) et production de statistiques. Les fonctions définies par l'utilisateur et l'analyseur de forme d'onde synthétique permettent notamment de calculer la transformée de Fourier de n'importe quelle fonction continue définie mathématiquement, un outil vraiment puissant !

Plus de convivialité

L'interface utilisateur (*User Interface* – UI) est beaucoup plus intuitive et cohérente avec l'ASPE II. On se sent maintenant en face d'un instrument de mesure moderne.

Rien à changer dans le matériel pour ces améliorations, il suffit de modifier le logiciel de l'ordinateur client pour accéder à la convivialité et aux nouvelles fonctions, et celui de l'ASPE pour la rapidité et la connectivité.

L'interface utilisateur

Passons en revue l'interface utilisateur de l'ASPE II. La **figure 1** en montre un exemple typique.

Se connecter à l'ASPE

Au démarrage, nous ne connaissons pas l'adresse IP de l'ASPE. Si nous appuyons sur le bouton **Find**, en haut à droite, une recherche d'un ou deux ASPE sur le réseau local est lancée. L'adresse IP des instruments détectés sera ajoutée à la liste ; après sélection de l'instrument souhaité, la liaison pourra être établie avec le bouton **Connect**.

Échantillonnage

Tout comme dans la version précédente, les paramètres d'échantillonnage peuvent être réglés dans le panneau **Sampling** : taux (échantillons par s), nombre

d'échantillons, résolution (10 ou 12 bits), et ADCS (constante de division pour l'obtention de la fréquence d'horloge du CA/N). Une fonction d'élimination de la gigue (*Prevent Jitter*) a été ajoutée, pour permettre la sélection du taux d'échantillonnage optimal. Lorsque cette fonction est sélectionnée, la fréquence d'échantillonnage sera ajustée automatiquement à la plus proche valeur présentant une gigue minimale. Cette fréquence est un multiple entier du nombre de cycles nécessaires pour la conversion A/N, divisé par la fréquence souhaitée.

Déclenchement

Les réglages du déclenchement (*trigger*) sont réalisés par le logiciel ; ils sont semblables à ceux rencontrés sur un oscilloscope. Lorsqu'un seuil de déclenchement est sélectionné (p. ex. 1 V), ainsi que le

Tableau 1. Le seuil de déclenchement est fixé à 1 V, flanc montant. L'index correspondant est 1039.

Index	1034	1035	1036	1037	1038	1039	1040	1041	1042
Tension	0,800	0,820	0,883	0,950	0,969	1,003	1,012	1,133	1,306

sens du flanc (p. ex. montant), l'algorithme va rechercher parmi les données reçues par l'ASPE les échantillons dont la valeur est supérieure ou égale au seuil de déclenchement, et dont la valeur de l'échantillon précédent est inférieure à ce niveau (voir **tableau 1**). Ceci nous donnera une liste des échantillons, avec leur index, qui répondent à ces critères ; le signal représenté commencera toujours au premier point rencontré dans cette liste.

Dans le panneau **Trigger**, à droite, le seuil de déclenchement peut être choisi entre –250 mV et +250 mV, et le déclenchement sur un flanc montant ou descendant. On peut aussi sélectionner un mode **Auto** ou **Normal**. Dans ce dernier cas, le signal ne sera montré qu'après un déclenchement effectif. En mode **Auto**, si aucun déclenchement n'a eu lieu après 200 ms ou plus, une capture unique sera effectuée automatiquement. Ce mode est le mode par défaut, comme sur la plupart des oscilloscopes ; il permet à l'utilisateur de voir à quoi le signal ressemble, même s'il ne répond pas aux paramètres de déclenchement. On peut aussi supprimer le déclenchement avec la commande **Off**. Le seuil de déclenchement est réglé avec le curseur à droite du

Caractéristiques

- Connexion réseau
- *Plug and play*
- Générateur de bruit
- Oscilloscope
- Analyseur de spectre
- Possibilité de sous-échantillonnage

graphique supérieur ; ce seuil est remis à 0 avec **Set 0V**.

L'analyseur est par défaut à l'arrêt (mode **Stop**) ; en appuyant sur le bouton rouge **Stop**, la capture commence, jusqu'à une pression sur le bouton **Run**. On peut aussi capturer un événement unique avec le bouton **Single** ; le système attend le déclenchement, puis se remet en mode **Stop** immédiatement.

Couplage

Le panneau **Coupling** permet la sélection du couplage souhaité. En mode **DC**, le signal est montré tel qu'il a été mesuré. En mode **AC**, la composante continue, calculée par transformée de Fourier rapide (TFR – FFT), est soustraite au signal avant qu'il soit montré.

Statistiques

Diverses statistiques sont établies pour chaque groupe de données reçues par l'ASPE II, comme tensions minimale, maximale, moyenne et crête à crête. Des moyennes sont aussi calculées et montrées ; le bouton **Reset** permet de les remettre à zéro. Les statistiques sont montrées en haut de l'interface utilisateur.

En sélectionnant **Show max trace** et **Show min trace**, des lignes représentant respectivement les valeurs maximale et minimale capturées seront affichées dans le graphique du domaine temporel. La remise à zéro est possible avec le bouton **Reset**, ou en les désélectionnant puis en les sélectionnant à nouveau. La même fonction existe pour le niveau maximal de la TFR.

Le décalage continu, la fréquence mesurée, et le taux de rafraîchissement sont visibles en bas du graphique du domaine temporel. Le décalage continu provient

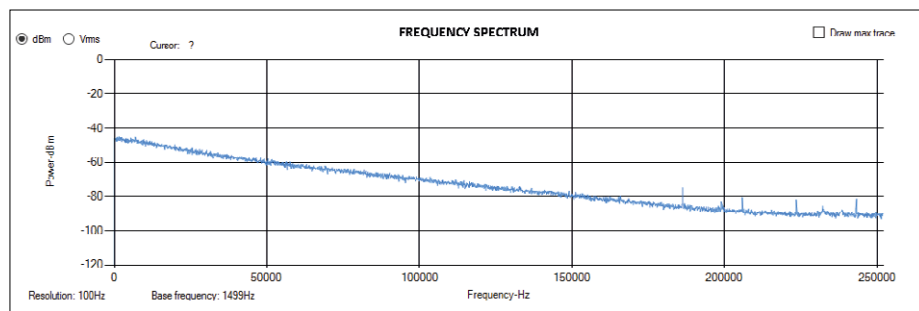


Figure 2. Entrée et sortie raccordées, mais à calibrer.

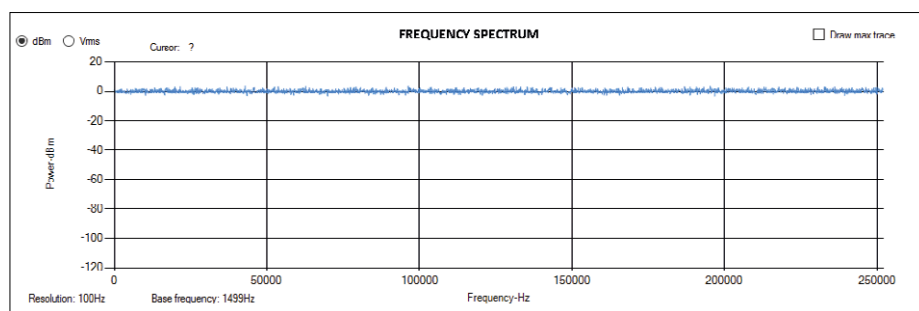


Figure 3. Entrée et sortie raccordées et calibrées.

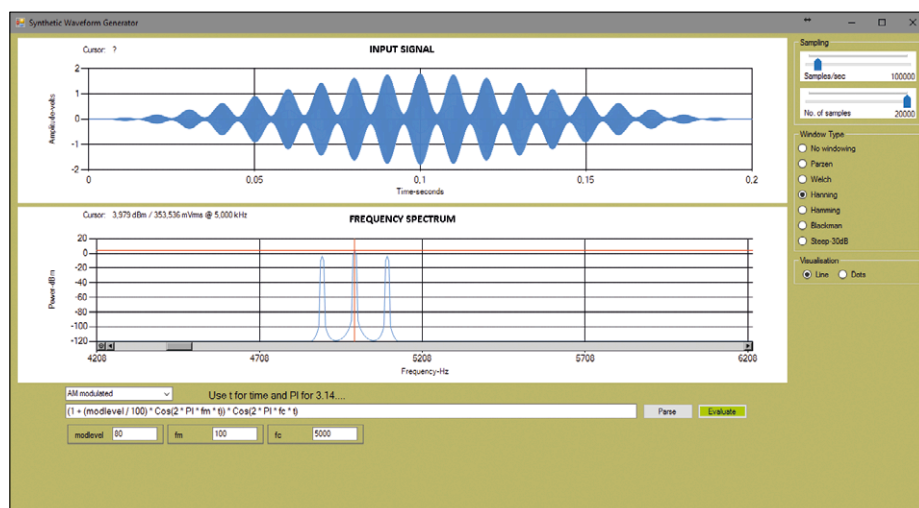


Figure 4. Analyse d'un signal modulé en amplitude avec l'analyseur de forme d'onde synthétique (SWA). Les valeurs du taux de modulation et des fréquences de la porteuse et de la modulation peuvent être entrées comme paramètres.

de la TFR, et la fréquence est calculée à partir du temps entre deux déclenchements. Le graphique du domaine fréquentiel montre également la fréquence de base, qui est la fréquence de l'échantillon de spectre de la TFR (*FFT bin*) qui possède l'amplitude la plus élevée.

Calibrage du bruit blanc

L'ASPE II permet la mesure de la réponse en fréquence d'un filtre. On utilise pour ce faire la source de bruit blanc interne de l'instrument comme signal d'entrée du filtre. La TFR permettra de mesurer l'amplitude des composantes du bruit à la sortie du filtre, et d'en montrer la bande passante.

Il y a cependant un filtre passe-bas à la sortie du générateur ; la densité spectrale de puissance n'est donc pas la même pour toutes les fréquences, ce qui fausse le résultat de la mesure. Nous avons prévu une fonction de calibrage, afin de corriger ce défaut. Il suffit de connecter la sortie du générateur de bruit à l'entrée de l'analyseur, et d'effectuer un grand nombre (p. ex. 500) de moyennes de la TFR. En appuyant sur le bouton **Calibrate**, en haut de l'interface utilisateur, l'enveloppe de la TFR obtenue précédemment servira de référence pour les mesures suivantes (voir **figures 2 et 3**). Vous pouvez connecter à nouveau le filtre entre sortie et entrée de l'instrument, vous obtiendrez une représentation plus correcte de sa réponse en fréquence.

Le processus de calibrage peut être automatisé, il suffit d'appuyer sur le bouton **Auto set**. Encore une remarque concernant l'exactitude des mesures : afin que l'impédance vue par le générateur de bruit soit la même, qu'il soit connecté à l'analyseur ou au circuit à tester, l'impédance d'entrée de ce dernier doit être de 50 Ω . Il suffira de connecter une résistance de cette valeur en parallèle sur l'entrée du circuit si ce n'est pas le cas. Il faudra aussi veiller à éventuellement insérer un condensateur (d'une valeur appropriée pour ne pas perturber la réponse en fréquence) entre sortie du circuit et entrée de l'analyseur, cette dernière acceptant les tensions continues.

Analyseur de forme d'onde synthétique

L'analyseur de forme d'onde synthétique (*Synthetic Waveform Analyzer* – SWA) a été amélioré, pour permettre à l'utilisateur d'entrer ses propres signaux (sous forme mathématique), et de les évaluer

Tableau 2. Configuration de JP2.

RB13 (broche 3)	RB14 (broche 1)	Adresse IP
Masse	Masse	192.168.1.123 (adresse fixe)
Masse	Ouvert	192.168.0.123 (adresse fixe)
Ouvert	Masse	DHCP 1
Ouvert	Ouvert	DHCP 2

en sélectionnant fréquence d'échantillonnage, nombre d'échantillons, et fenêtre de la TFR.

L'analyseur utilise une bibliothèque libre appelée *NCalc* [4] pour évaluer et analyser la formule entrée. Après avoir entré cette formule et appuyé sur le bouton **Parse**, les paramètres seront affichés comme champs d'entrée sous la fenêtre contenant la formule (**figure 4**). Ces paramètres seront utilisés pour évaluer la formule pour chaque échantillon (bouton **Evaluate**). Les résultats sont montrés dans les domaines temporel et fréquentiel.

Connexion à chaud (plug and play)

L'adresse IP utilisée par l'ASPE original pour se connecter à un réseau local était sélectionnée à l'aide d'un cavalier parmi quatre adresses fixes. Avec le protocole de configuration dynamique des hôtes (*Dynamic Host Configuration Protocol* – DHCP), l'adresse IP est désormais attribuée dynamiquement lors de la connexion au réseau. Ceci permet en outre la connexion simultanée de deux ASPE II au même réseau.

Attribution de l'adresse IP

Rappelons tout d'abord que nous voulons établir une connexion entre un ordinateur, considéré comme client, et l'ASPE II, le serveur. Le serveur DHCP, généralement dans le routeur du réseau, commande le processus d'attribution des adresses IP ; nous utilisons les facilités de débogage de l'ASPE II pour en visualiser les différentes phases. Pour le débogage, nous devons relier un ordinateur au connecteur K6 de l'ASPE à l'aide d'un adaptateur série-USB. Un émulateur de terminal (p. ex. *Tera Term*) permet alors de suivre le processus DHCP (**figure 5**).

Le cavalier JP2 permet la sélection d'une adresse IP parmi quatre (**tableau 2**). Les deux premières sont des adresses fixes, comme pour l'ASPE original, les deux autres sont attribuées par le protocole DHCP, une par ASPE connecté au réseau.

Nous avons maintenant une adresse IP attribuée à l'ASPE par le routeur du réseau, mais elle n'est pas encore connue du logiciel client tournant sur l'ordinateur. Le problème de découverte de l'adresse de l'ASPE par l'ordinateur est plus ardu qu'il n'y paraît !

Reconnaissance de l'ASPE

Le fichier *NetworkScanner.cs* du programme en C# contient le code qui permet à l'ordinateur de trouver l'adresse IP de l'ASPE. Examinons les différentes étapes et les fragments de code du fichier qui sont essentiels au processus :

1. (**Listage 1**) Trouver l'adresse IP des accès au réseau (en principe dans le routeur), et vérifier qu'elles sont actives. Si c'est le cas, acquérir les adresses (un routeur peut avoir par exemple l'adresse 192.168.0.1).
2. (**Listage 2**) Rechercher un ASPE II dont l'adresse IP est parmi celles des accès au réseau. Ceci se fait à l'aide de l'outil *Ping*, qui

Figure 5. Les étapes du processus DHCP.

envoie un signal vers toutes les adresses possibles jusqu'à ce qu'il reçoive une réponse. Par exemple, si l'adresse IP du routeur est 192.168.0.1, les adresses de 192.168.0.2 à 192.168.0.255 seront essayées. Ceci pourrait prendre pas mal de temps si les adresses devaient être vérifiées à la queue leu leu, le programme teste donc plusieurs adresses en parallèle (*multithreading*).

Listage 2. Recherche d'un ASPE.

```
private static void Ping(IPAddress host, int attempts, int timeout)
{
    ...
    Ping ping = new Ping();
    ping.PingCompleted += PingCompleted;
    ping.SendAsync(host, timeout, host);
    ...
}
```

Listage 1. Recherche de l'adresse IP de la passerelle d'accès au réseau.

```
static IPAddress NetworkGateway()
{
    ...
    foreach (NetworkInterface f in NetworkInterface.GetAllNetworkInterfaces())
    {
        if (f.OperationalStatus == OperationalStatus.Up)
        {
            foreach (GatewayIPAddressInformation d in f.GetIPProperties().GatewayAddresses)
            {
                if (d.Address.AddressFamily == AddressFamily.InterNetwork)
                {
                    ip = d.Address;
                }
            }
        }
    }
}
```

3. (**Listage 3**) Lorsqu'un test d'adresse est terminé, vérifier si une réponse a été reçue. Si c'est le cas, essayer d'établir une connexion TCP/IP avec cette adresse et le port 4000 (numéro de port inscrit dans le code TCP/IP de l'ASPE II). Si cela fonctionne, acquérir l'adresse MAC (*media access control*) de l'ASPE.

4. (**Listage 4**) Pour acquérir l'adresse MAC de l'ASPE, il faut consulter la table ARP (*address resolution protocol*) de l'ordinateur. Cette table contient les adresses IP et les adresses MAC correspondantes qui sont actives sur le réseau et connues de l'ordinateur.

À chaque connexion TCP/IP réussie, l'ordinateur ajoute une paire IP-MAC à la table ARP. Si nous consultons cette table, nous pouvons donc retrouver

l'adresse MAC de l'ASPE (**figure 6**).

En fait, c'est l'adresse MAC qui permet d'avoir deux appareils sur le même réseau. L'ASPE modifie le champ le plus significatif de l'adresse MAC par défaut du W5500, en fonction du cavalier JP2. L'ordinateur voit deux adresses MAC différentes, et sait qu'il y a deux adresses IP correspondantes différentes. Vous pourriez même ajouter un troisième ASPE au réseau, en lui attribuant une adresse IP fixe !

On y est : l'ASPE a son adresse IP, elle est connue de l'ordinateur, et ils peuvent l'utiliser pour communiquer. Ceci facilitera la connexion fructueuse de l'ASPE au réseau.

Amélioration des performances

Les améliorations résultent essentiellement de l'utilisation du *multithreading* dans le logiciel du client, et de l'accès

direct à la mémoire (*direct memory access* – DMA) pour le dsPIC33 du serveur.

Multitâches

Si l'on ne tient pas compte de l'exécution « simultanée » de plusieurs tâches lors de l'écriture d'un programme pour PC, toutes les instructions seront exécutées séquentiellement, dans une seule et même tâche (la principale), lors du démarrage par le système d'exploitation. Cette unique tâche ne tient bien entendu pas compte des possibilités de l'unité centrale de traitement (CPU), qui possède huit processeurs logiques si elle est de type quatre cœurs. Nous n'utilisons donc que 12,5% des capacités de la CPU, ce qui est un véritable gâchis ! C'était bien le cas dans la première version du logiciel client : demande de données à l'ASPE, réception des données, conversion des volts en décibels, calcul de la TFR, établissement des graphiques dans les domaines temporel et fréquentiel. Si l'une ou plusieurs des tâches prennent du temps, c'est l'ensemble qui est ralenti. La gestion de l'interface utilisateur est aussi une de ces tâches séquentielles ; la réaction à une action de l'utilisateur peut paraître très lente, de même que le rafraîchissement de la fenêtre. La solution était (*of course*) de traiter chaque tâche principale indépendamment des autres, et de distribuer la charge de travail aux huit processeurs logiques de la CPU.

Avec le nouveau logiciel, la demande et la réception des données sont des tâches séparées, de même que le calcul de la TFR. Les performances sont nettement améliorées, le rafraîchissement de l'interface utilisateur et sa réactivité sont beaucoup plus rapides. Lorsqu'un signal est affiché à l'écran, d'autres échantillons sont déjà traités et leur TFR calculée. Le taux de rafraîchissement de l'écran est désormais d'environ 40 Hz, au lieu de 8 précédemment, pour une acquisition de 5 042 éch/s (**tableau 3**). C'est cinq fois mieux !

Tableau 3. Taux de rafraîchissement des deux versions ($f_s = 504\,201$ Hz).

N	Taux de rafraîchissement (Hz) de l'ASPE	Taux de rafraîchissement (Hz) de l'ASPE II
5 000	8,33	42
10 000	3,96	22
15 000	2,48	14
20 000	1,70	11

Listage 3. Réponse d'un ASPE ?

```
private static void PingCompleted(object sender,
PingCompletedEventArgs e)
{
    ...
    if (e.Reply != null && e.Reply.Status == IPStatus.Success)
    {
        // Try to connect to the port
        if (TryConnect(ip, _port))
        {
            // Can connect.
            string macaddress = GetMacAddress(ip);
            ...
        }
    }
}
```

Listage 4. Utilisation de la table ARP pour obtenir l'adresse MAC.

```
public static string GetMacAddress(IPAddress ip)
{
    ...
    if (SendARP(BitConverter.ToInt32(ip.GetAddressBytes()), 0), 0, macAddr, ref macAddrLen) != 0)
        return "00:00:00:00:00:00";
    ...
}
```


Accès direct à la mémoire (DMA)

Dans le logiciel original, l'interface SPI du dsPIC33 était utilisée par les fonctions `wiznet_read()` et `wiznet_write()` de la bibliothèque d'E/S du W5500 pour contrôler le transfert de données. La vitesse de transfert était limitée par le fait que les instructions devaient être exécutées par le dsPIC. Nous utilisons désormais les possibilités d'accès direct à la mémoire, ce qui permet de transférer directement des données entre la mémoire et l'interface SPI sans passer par le processeur central du dsPIC. Nous atteignons maintenant 1,3 Mo/s au lieu de 700 Ko/s.

C'est même presque trivial, et deux canaux DMA sont nécessaires. Le premier canal traite l'envoi au W5500, et le second effectue une lecture fictive après chaque transfert d'octet, comme requis par l'interface SPI du dsPIC33. Les deux canaux opèrent en mode *One Shot* : transfert d'un bloc de données, déclenchement d'une interruption, attente d'une nouvelle demande de transfert. Le **listage 5** montre le déroulement du processus.

La lecture fictive après chaque transfert d'octet est loin d'être idéale, de même que le démarrage manuel du transfert du premier octet. On devrait pouvoir se contenter d'un seul canal DMA, mais les concepteurs du dsPIC33 en ont décidé autrement...

Mise à jour de l'ASPE

L'ASPE original fonctionnera avec le nouveau logiciel client (sur PC), la nouvelle interface utilisateur, ainsi que les fonctions définies par l'utilisateur et l'analyseur de forme d'onde synthétique. Pour la connexion dynamique à un réseau (DHCP) et certaines améliorations des performances, le logiciel du dsPIC33 devra cependant être mis à jour.

Pour mettre à jour le logiciel client, il faut télécharger la version gratuite de *Visual Studio 2015* de Microsoft, puis *NCSA Visual Studio* sur la page du projet [3], et l'exécuter.

Pour mettre à jour le logiciel du dsPIC, il faut télécharger l'environnement de développement gratuit *MPLABS* de Microchip, puis *MPLABS.X* sur la page du projet [3]. Après compilation du code, un programmeur sera nécessaire pour le charger sur le dsPIC33, par exemple le *PICKit 3* de Microchip, qui est bon marché ; celui-ci se raccorde directement au connecteur ICSP K1. Si cela vous

Listage 5. SPI avec DMA sur le dsPIC33.

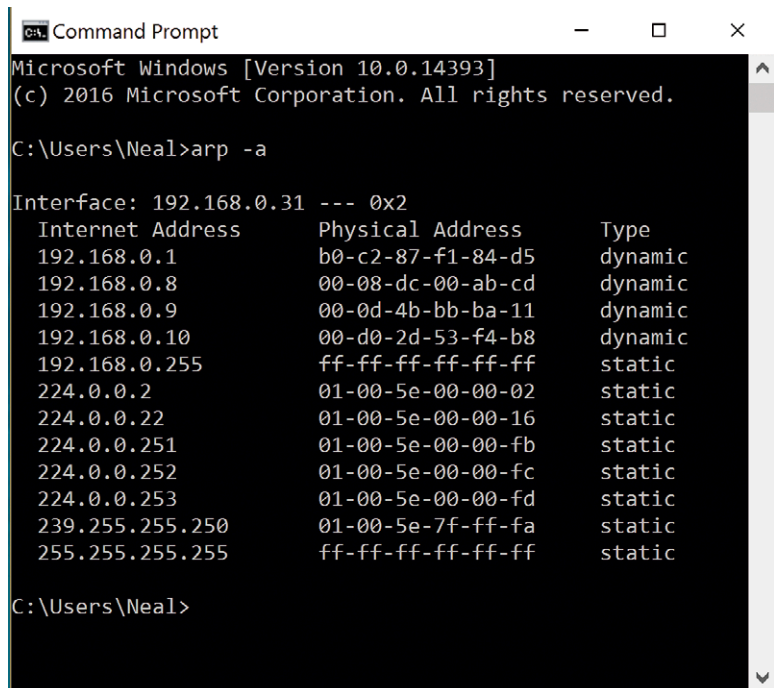
```
done = false; //using dma and spi for faster uP to W5500 transfers
SPI1STATbits.SPIEN=0; //keep SPI off until ready for transfer

//Set up DMA 1 to transfer from RAM at pBuf to SPI
DMA1CNT = len-1; //block size
DMA1STAL = (unsigned int) pBuf; //point to data buffer
IFS0bits.DMA1IF = 0; //clear DMA interrupt flag
IEC0bits.DMA1IE = 1; // Enable DMA interrupt
DMA1CONbits.CHEN = 1; //turn DMA1 on

//Set up DMA 2 to do dummy reads after each write;
//required for dsPIC33 DMA SPI
DMA2STAL = (unsigned int) &dummy;
DMA2CNT = len-1;
IFS1bits.DMA2IF = 0;
IEC1bits.DMA2IE = 1;
DMA2CONbits.CHEN=1;

//force the first byte transfer after enabling SPI;
//required by dsPIC33 to start process
SPI1STATbits.SPIEN=1; //turn on SPI
DMA1REQbits.FORCE=1;
while (DMA1REQbits.FORCE==1);

//wait for DMA transfer to complete; done set true by DMA interrupt
while (done==false);
```



```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Neal>arp -a

Interface: 192.168.0.31 --- 0x2
   Internet Address      Physical Address        Type
192.168.0.1              b0-c2-87-f1-84-d5      dynamic
192.168.0.8              00-08-dc-00-ab-cd      dynamic
192.168.0.9              00-0d-4b-bb-ba-11      dynamic
192.168.0.10             00-d0-2d-53-f4-b8      dynamic
192.168.0.255            ff-ff-ff-ff-ff-ff      static
224.0.0.2                01-00-5e-00-00-02      static
224.0.0.22               01-00-5e-00-00-16      static
224.0.0.251              01-00-5e-00-00-fb      static
224.0.0.252              01-00-5e-00-00-fc      static
224.0.0.253              01-00-5e-00-00-fd      static
239.255.255.250          01-00-5e-7f-ff-fa      static
255.255.255.255          ff-ff-ff-ff-ff-ff      static

C:\Users\Neal>
```

Figure 6. La table ARP, obtenue en tapant `arp -a` dans une fenêtre de commande. L'entrée pour l'adresse IP 192.168.0.8 est celle de notre ASPE avec l'adresse MAC par défaut 00-08-dc-00-ab-cd du W5500.

Réponse en fréquence avec le générateur de bruit calibré

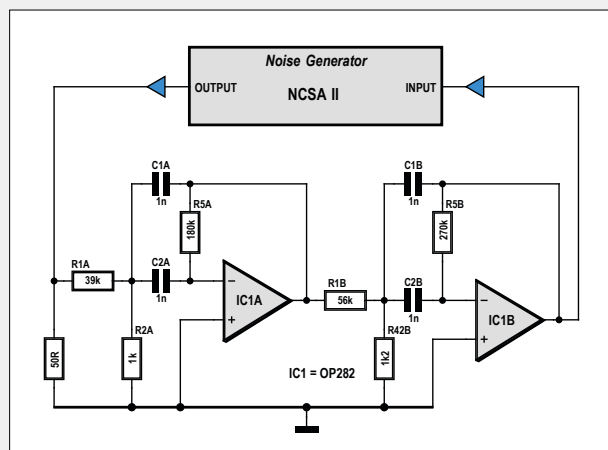


Figure 7. Analyse d'un filtre passe-bande avec le générateur de bruit blanc intégré.

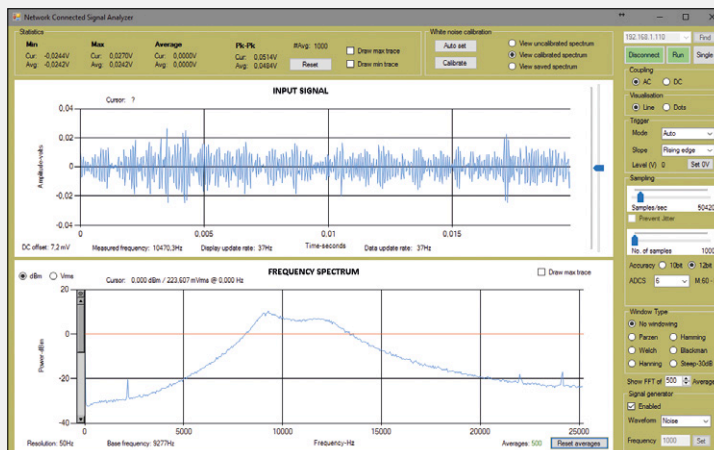


Figure 8. Réponse en fréquence du filtre de la figure 7, après calibrage du bruit.

Utilisons le générateur interne de bruit blanc de l'ASPE pour déterminer la réponse en fréquence d'un filtre passe-bande (**figure 7**). Le générateur est tout d'abord connecté à l'analyseur, afin d'établir une référence ; celle-ci est ensuite utilisée pour établir la réponse en fréquence du filtre, alimenté

par le générateur de bruit (**figure 8**). Nous déduisons du graphique que la fréquence centrale du filtre est de 10 kHz, sa bande passante de ± 5 kHz, et sa pente d'environ 24 dB par octave, ce qui est normal pour un filtre à quatre pôles.

Sous-échantillonnage

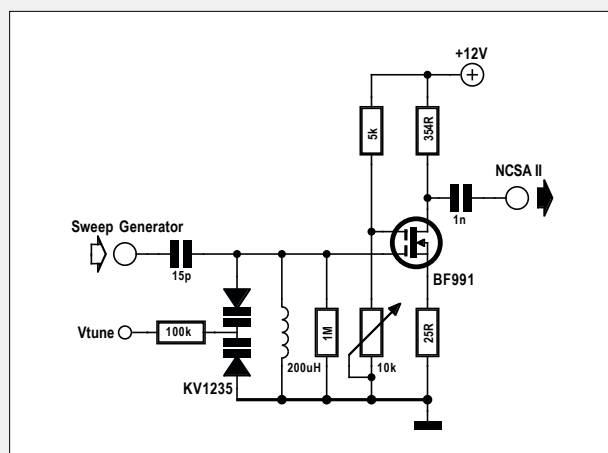


Figure 9. Filtre passe-bande ajustable à MOSFET d'une radio MA.

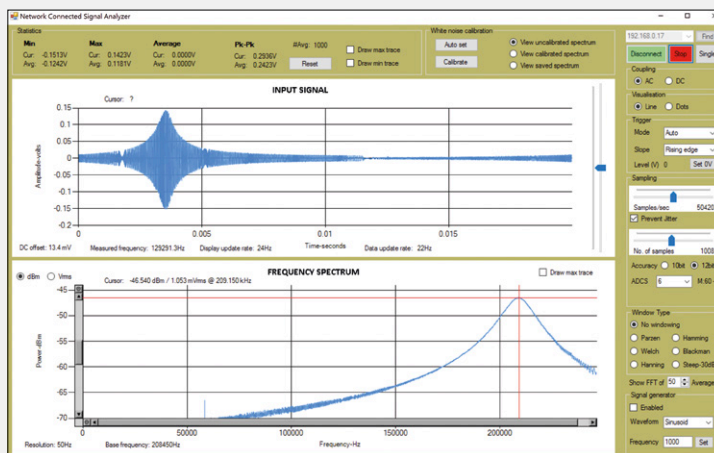


Figure 10. Réponse en fréquence du filtre passe-bande, centré sur 1,201 MHz. Un sous-échantillonnage a été utilisé.

Nous souhaitons maintenant analyser un filtre passe-bande dont la fréquence centrale est de 1,2 MHz (**figure 9**). Il s'agit d'un filtre passe-bande LC ajustable à MOSFET, pour l'étage d'entrée d'une radio MA. La fréquence centrale est ajustée (entre 580 kHz et 1,6 MHz) en modifiant la tension aux bornes de deux diodes à capacité variable, en parallèle avec une inductance.

La fréquence d'échantillonnage de l'ASPE ne peut atteindre de telles valeurs, il faut donc utiliser le sous-échantillonnage, décrit dans un article précédent [2].

Un générateur à balayage externe est utilisé : 1 à 1,5 MHz en 10 ms ; le filtre est centré sur 1,2 MHz. La réponse du filtre est en **figure 10**, elle est normale pour un filtre passe-bande.

Le graphique indique une fréquence centrale de 209,150 kHz ; à cause du sous-échantillonnage, il faut ajouter des multiples de la fréquence d'échantillonnage, la fréquence centrale exacte du filtre est donc égale à 1 217 552 Hz (209 150 Hz + 2 × 504 201 Hz).

Modulation d'amplitude en quadrature (QAM)

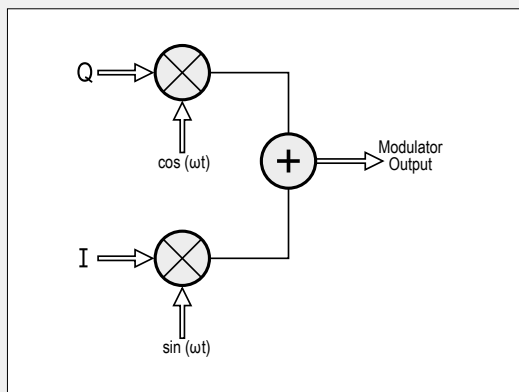


Figure 11. Synoptique du modulateur à quadrature (IQ).

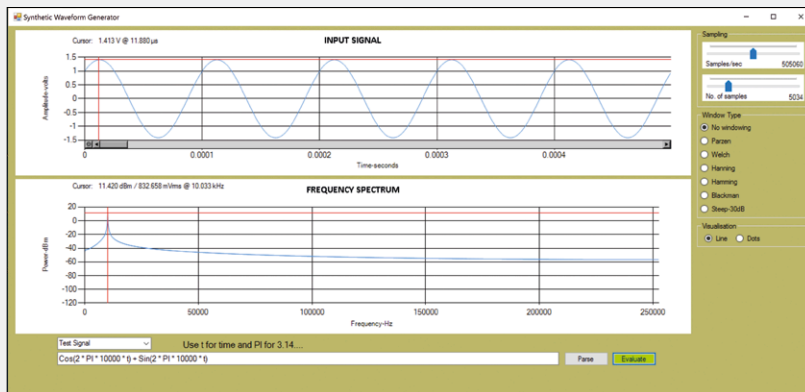


Figure 12. Signal de la sortie du modulateur à quadrature du générateur de l'ASPE.

Cet exemple illustre les possibilités des fonctions définies par l'utilisateur (UDF) et de l'analyseur de forme d'onde synthétique (SWA). Nous analysons ici un modulateur QAM (auss appelé I-Q), très utilisé dans les systèmes de télécommunication (**figure 11**). La modulation 256QAM est fréquemment utilisée pour la télévision numérique par câble, où I et Q peuvent prendre seize valeurs d'amplitude distinctes chaque. L'équation définissant le modulateur est la suivante :

$$Q \cos(\omega t) + I \sin(\omega t) = C \cos(\omega t + \theta)$$

$$\text{où } C = \sqrt{Q^2 + I^2} \text{ et } \theta = \tan^{-1}(Q / I)$$

En analysant cette équation, nous en déduisons que pour une paire I-Q, la sortie du modulateur sera une onde cosinusoidale dont l'amplitude et la phase résultent de l'équation. Voyons ce qu'en dit l'analyseur de forme d'onde synthétique.

Avec I et Q égaux à 1 V, le signal devrait être de forme cosinusoidale avec une phase de 45° à $t = 0$, et une amplitude de 1,414 V. Les résultats, en **figure 12**, confirment la théorie. L'utilisateur peut modifier les valeurs de I et Q, et observer les résultats.

semble exagéré, vous pourrez toujours faire appel au service de reprogrammation d'Elektor (voir ci-après).

Et la suite ?

On peut encore améliorer l'ASPE, c'est un fait. Un double tampon permettrait d'accroître la vitesse jusqu'à avoir un flux de données continu, soit un oscilloscope et une TFR en temps réel. Une radio

logicielle (*software defined radio* – SDR) est alors aussi possible.

L'analyseur de forme d'onde synthétique (SWA) pourrait traiter des fonctions non linéaires (*Rect, Sign, Delta...*), ce qui permettrait l'analyse de beaucoup plus de signaux.

Nous avons un outil très puissant, et évolutif. Restez connecté ! ◀

(160362 – version française :
Jean-Louis Mehren)

Liens

- [1] analyseur de signal par Ethernet (1), 03/2016 : www.elektormagazine.fr/150211
- [2] analyseur de signal par Ethernet (2), 04/2016 : www.elektormagazine.fr/150694
- [3] www.elektormagazine.fr/160362
- [4] ncalc.codeplex.com



DANS L'E-CHOPPE

→ 150211-92
Module assemblé de l'ASPE

→ 150211-71
Boîtier pour l'ASPE

→ 160362-41
Reprogrammation de l'ASPE en ASPE II

radiocommande

avec modules radio XBee ou nRF24

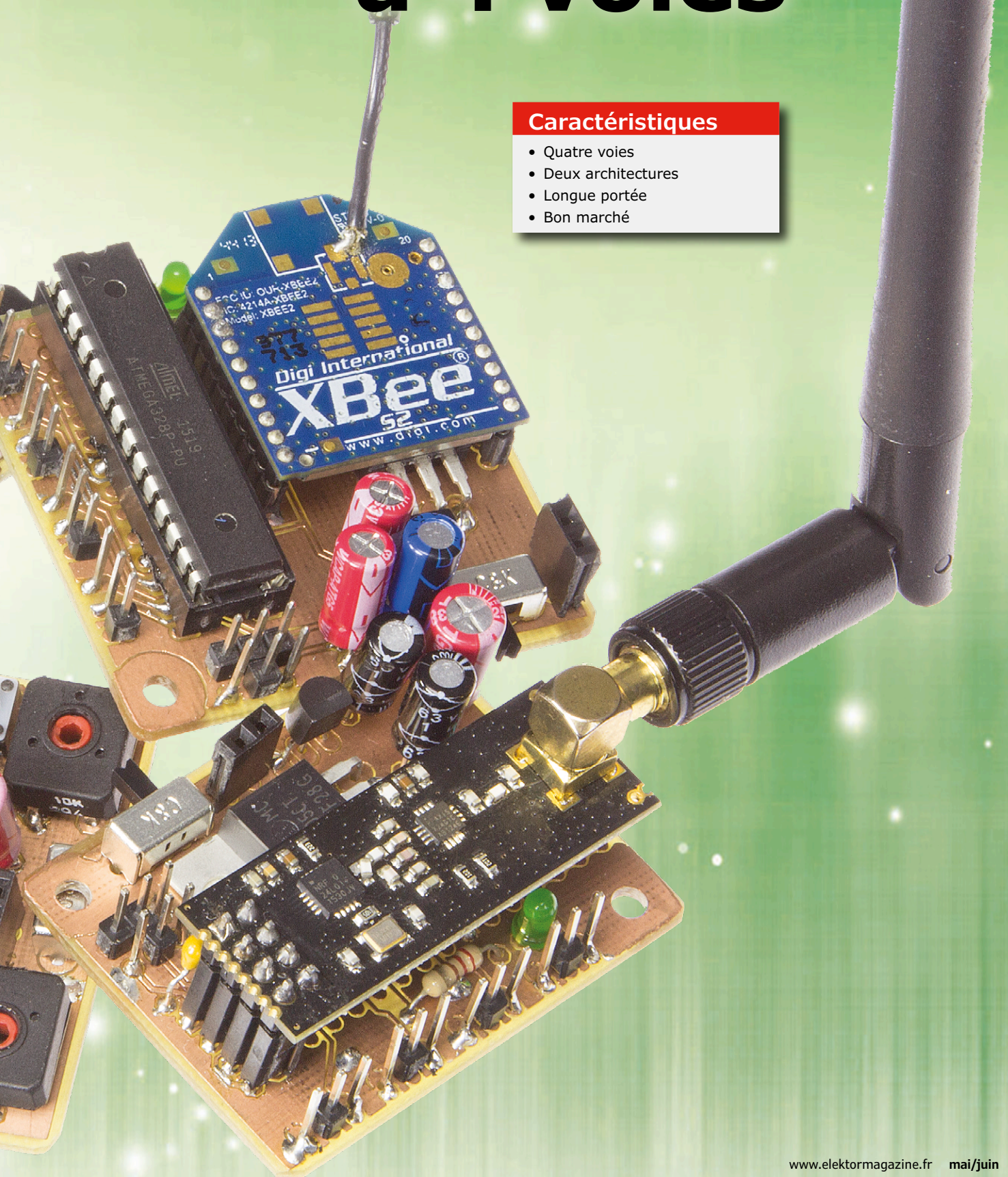
Somnath Bera (Inde) & Roy Aarts (labo d'Elektor)



bidirectionnelle à 4 voies

Caractéristiques

- Quatre voies
- Deux architectures
- Longue portée
- Bon marché



Les modules radio bon marché pour la bande ISM sont parfaits pour commander à distance des jouets, tant qu'il s'agit de basculer des interrupteurs ou de faire tourner un ou deux moteurs. Mais si le système de commande est plus complexe, par ex. commande de la gouverne de direction, du gouvernail d'altitude et des ailerons d'un avion tout en actionnant un moteur sans balais et en écoutant un signal de retour pour chacune de ces actions, les modules élémentaires ne sont plus à la hauteur.

À cause de sa complexité, l'application décrite ci-dessus demande un système radio plus élaboré. Les célèbres modules XBee de Digi International répondent précisément à ce type de besoin. Au départ, ils ont été conçus pour les liaisons point à point et en étoile, à la vitesse de 250 kbits/s ; aujourd'hui c'est une

famille de modules émetteur/récepteur sans fil, faciles à utiliser dans différentes architectures : cellulaire, ZigBee, Thread, 802.15.4 et Wi-Fi entre autres. Nous utiliserons ici des modules XBee S2C (anciennement ZB) qui comportent une pile logicielle ZigBee.

Les modules XBee sont certes faciles à

utiliser, mais ils sont chers ; c'est pourquoi notre ingénieur au labo s'est lancé le défi d'atteindre les mêmes résultats avec des modules nRF24 meilleur marché. Cet article présente donc deux versions de la télécommande à quatre voies, l'une avec des modules XBee, l'autre des nRF24. Commençons par la première.

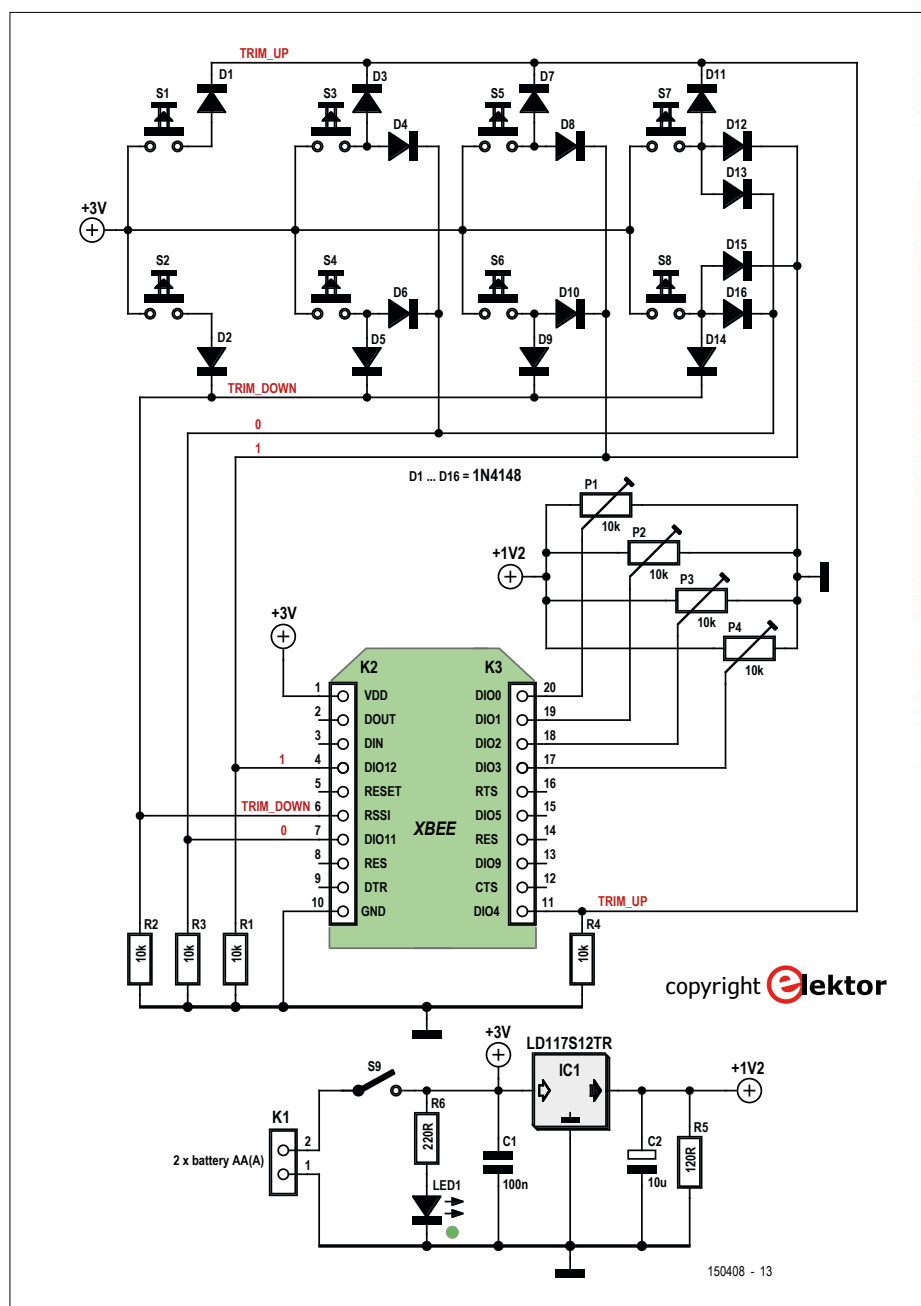
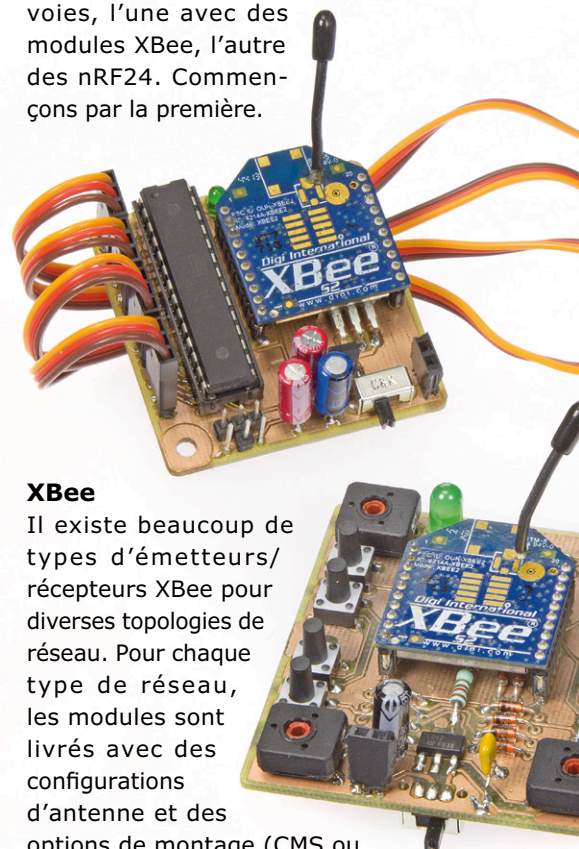


Figure 1. Schéma de l'émetteur XBee.



XBee

Il existe beaucoup de types d'émetteurs/récepteurs XBee pour diverses topologies de réseau. Pour chaque type de réseau, les modules sont livrés avec des configurations d'antenne et des options de montage (CMS ou traversant) différentes. En outre, il y a deux versions des modules ZigBee : normale et *pro* (meilleure portée). La version normale a une portée d'environ 60 m en intérieur et en milieu urbain, et de 1200 m en terrain dégagé. Avec la version *pro*, c'est 90 m en intérieur/urbain, et 3,2 km en terrain dégagé. Avec une antenne directionnelle Yagi, la portée en extérieur de la version *pro* peut atteindre 20 km. Attention toutefois : le XBee-Pro n'est pas certifié pour une utilisation en Europe.

Les modules ont grosso modo la taille d'une pièce de 2 € (en imaginant une pièce carrée), soit 24 × 28 mm ; la version *pro* est un peu plus longue. Non seulement les modules *pro* sont un peu plus gros, mais ils consomment également plus d'énergie. Les modules pos-

Les modules ZigBee sont prévus pour les réseaux maillés et les routeurs. Il existe d'autres appareils pour créer des réseaux plus gros couvrant des zones plus étendues. Toutefois, il est possible de les utiliser pour une communication point à point comme nous allons le faire.

L'émetteur XBee

La **figure 1** montre le schéma de l'émetteur à cœur XBee. Comme le module XBee

Cortex-M3 préprogrammé, aucun μC externe n'est nécessaire. Le micrologiciel du XBee balaie les entrées numériques et analogiques, et transmet les valeurs au récepteur « intéressé » via un lien sériel virtuel (sans fil).

Quatre des broches d'E/S du module sont utilisées pour mesurer des tensions. Nous avons utilisé des ajustables, mais pour commander un avion ou une voiture, mieux vaut les remplacer par deux joysticks. L'axe horizontal de chacun des joysticks peut se substituer à P1 ou P3, l'axe vertical à P2 ou P4. Il est également possible de remplacer un ajustable par une photorésistance avec une résistance en série ou n'importe quel autre capteur qui délivre une tension. Les entrées analogiques mesurent des tensions jusqu'à 1,2 V. Les ajustables sont alimentés par le régulateur de tension de 1,2 V IC1, afin que la tension d'alimentation soit indépendante de celle des piles.

L'idée est de pouvoir commander des servomoteurs. Mais, les servos ont la plupart du temps besoin d'un petit décalage dans la commande pour compenser leurs imprécisions mécaniques : c'est le rôle des poussoirs S1 à S8. Les poussoirs impairs augmentent le décalage,

les autres le diminuent. La paire S1-S2 commande la voie n°1, S3-S4 la voie n°2 et ainsi de suite. Seize diodes, D1 à D16, sont utilisées pour coder les neuf combinaisons, la neuvième correspond à l'état dans lequel tous les poussoirs sont au repos, une combinaison valide et importante. Les pressions simultanées sur plusieurs des poussoirs seront rejetées par le récepteur.

Cela conclut la description du matériel de l'émetteur ; sa programmation sera décrite plus loin. Bonne nouvelle : vous n'aurez pas à apprendre un langage de programmation.

Le récepteur XBee

Le schéma du récepteur (**fig. 2**) diffère de celui de l'émetteur : il faut un microcontrôleur pour décoder les données sérielles reçues. Le micrologiciel d'usine du module XBee n'a aucune idée de la signification des données reçues, il ne

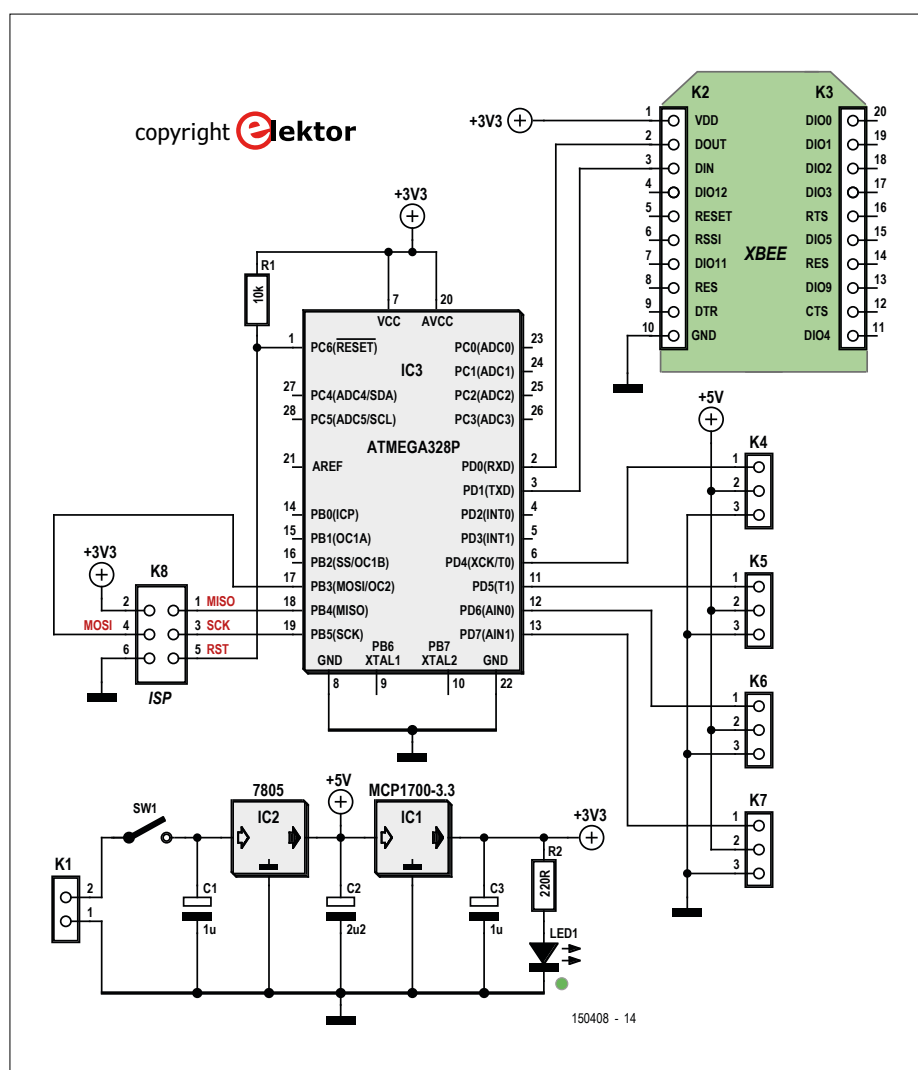


Figure 2. Le récepteur XBee comporte un microcontrôleur pour décoder les données transmises par l'émetteur.

PROJECT INFORMATION

modèles réduits

XBee nRF24 Arduino

télécommande RC

débutant

connaisseur

expert

env. 2 h

Ordinateur, fer à souder,
adaptateur USB/Série,
programmeur AVR ISP

version XBee : env. 100 €
version nRF24 : env. 35 €



LISTE DES COMPOSANTS ÉMETTEUR XBEE (150408-3)

Résistances

P1-P4 = 10 kΩ, ajustable de précision
R1-4 = 10 kΩ
R5 = 120 Ω
R6 = 220 Ω

Condensateurs

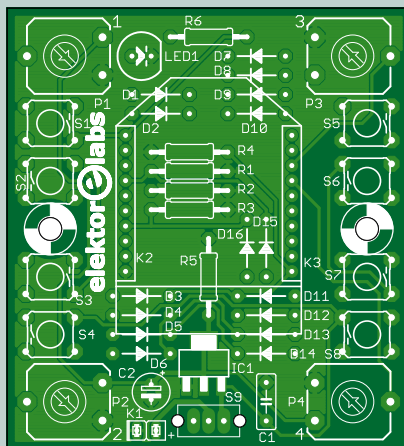
C1 = 100 nF
C2 = 10 µF, 6,3 V, au pas de 2,5 mm

Semi-conducteurs

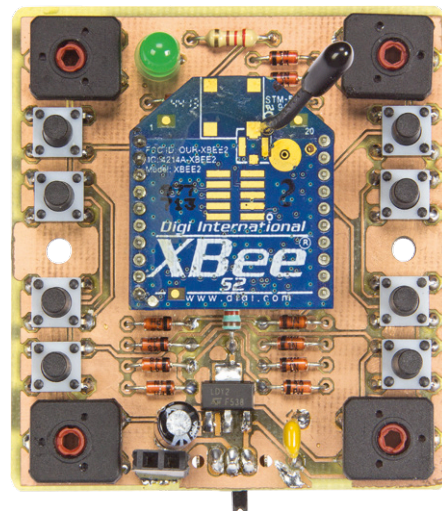
IC1 = LD1117S12TR
D1-16 = 1N4148
LED1 = LED, verte, 3 mm

Divers

K1 = barrette sécable à 2 voies,
au pas de 2,54 mm
K2, K3 = barrette sécable femelle à 10 voies,
au pas de 2 mm
S1 à S8 = poussoirs 6×6 mm
S9 = interrupteur à glissière
(OS102011MS2QN1C)
XBee S2C ou XBee-Pro S2C (+ antenne)



circuit imprimé réf. 150408-3
Pas de micrologiciel



peut donc pas s'acquitter automatiquement de cette tâche ; il se contente de pousser ces données sur son port série. C'est le microcontrôleur IC3, un ATmega328P, qui se charge du décodage. Il est possible, lorsque l'on achète la version programmable des modules XBee, d'en modifier le micrologiciel et ainsi de supprimer le µC externe. Malheureusement, cette approche n'est viable que pour ceux (lisez les entreprises) qui veulent utiliser les modules XBee à grande échelle. Le µC reçoit les données sur son port série et les convertit en signaux pour les servos sur les connecteurs K4 à K7. IC2 a été ajouté pour fournir une alimentation 5 V propre pour les servos, le reste du circuit tourne en 3,3 V. Le logiciel pour le récepteur a été écrit avec l'EDI Arduino et sera donc facile à modifier (voir l'**encadré**).

Programmation des microcontrôleurs

Les trois microcontrôleurs utilisés dans cet article sont cadencés par leur oscillateur RC interne à 8 MHz, et la configuration de leurs fusibles sera la même (**tableau 3**).

Pour compiler le logiciel pour ces µC dans l'EDI Arduino, faites attention à sélectionner la bonne carte (menu **Tools**). Plusieurs choix sont valides, mais nous vous recommandons de choisir **Arduino Pro** ou **Pro Mini** avec l'option **Processor** (également dans le menu **Tools** après avoir sélectionné la carte) réglée sur **ATmega328 (3,3 V, 8 MHz)**. Vous pouvez aussi opter pour **Atmega328 on a breadboard (8-MHz internal clock)** comme décrit sur cette page : www.arduino.cc/en/Tutorial/ArduinoToBreadboard

Pour compiler le logiciel du système à nRF24, vous devrez installer la bibliothèque Arduino nRF24 : <https://github.com/TMRh20/>



LISTE DES COMPOSANTS RÉCEPTEUR XBEE (150408-4)

Résistances

R1 = 10 kΩ
R2 = 220 Ω

Condensateurs

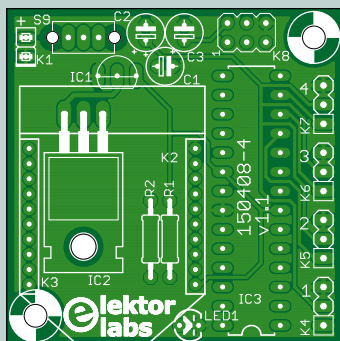
C1, C3 = 1 µF, 6,3 V, au pas de 2,5 mm
C2 = 2,2 µF, 6,3 V, au pas de 2,5 mm

Semi-conducteurs

IC1 = MCP1700-3302E/TO
IC2 = 7805
IC3 = ATmega328P-PU, programmé
(réf. 150408-43)
LED1 = verte, 3 mm

Divers

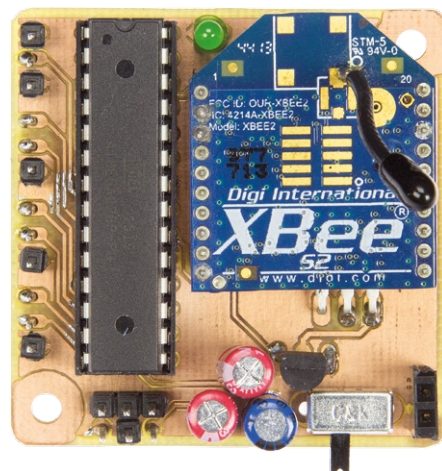
K1 = barrette sécable à 2 voies, au pas de 2,54 mm
K2, K3 = barrette sécable femelle à 10 voies, au pas de 2 mm
K4-7 = barrette sécable à 3 voies, au pas de 2,54 mm



K8 = barrette sécable à double rangée à 6 voies (2×3), au pas de 2,54 mm
S9 = interrupteur à glissière
(OS102011MS2QN1C)
XBee S2C ou XBee-Pro S2C (+ antenne)
circuit imprimé réf. 150408-4

Relier l'émetteur au récepteur

Les modules XBee doivent être configurés avant de pouvoir les utiliser comme souhaité. Pour ce faire, vous aurez besoin d'un ordinateur et probablement d'un convertisseur USB/série (un câble ou une carte, voir dans l'e-choppe), mais



une carte de liaison XBee (BoB) vous sera sans doute utile (fabriquez-en une ou rendez-vous dans l'e-choppe). Vous devez également télécharger et installer l'utilitaire gratuit XCTU [1] sur votre ordinateur. Les lecteurs expérimentés pourront se passer d'XCTU et envoyer eux-mêmes les incantations nécessaires sous la forme de commandes AT. D'abord, configurons l'émetteur. Reliez le module au PC via une ligne série (convertisseur USB/série si nécessaire) et une BoB Xbee.

Sur l'ordinateur, démarrez XCTU. Une fois prêt, cliquez sur le bouton *Discover radio modules*. Il faudra peut-être remettre à zéro le module avant qu'il ne soit détecté. Une fois le module détecté, un écran de configuration apparaît. Cliquez sur le bouton *Read* pour lire la configuration courante du module (fig. 3).

Configuration de l'émetteur XBee

Il faut programmer le module avec le jeu de fonctions *ZigBee Router API*. Pour cela, cliquez sur *Update*, sélectionnez la famille de produits *XB24-ZB* ainsi que le jeu de fonctions *ZigBee Router API* et la version *23A7* du micrologiciel. Enfin, cliquez sur *Update* (fig. 4).

Maintenant, reconfigurez le module XBee. Dans la section *Networking*, saisissez un *PAN ID*, c.-à-d. un nombre entre 0 et 0xFFFF FFFF FFFF FFFF (64 bits). Cet ID permet d'éviter les interférences entre réseaux, oubliez *1234* (trop simple !). Notez le nombre saisi, vous en aurez besoin pour le récepteur.

Pour le champ *Channel Verification*, sélectionnez *Enabled [1]*.

Faites défiler la liste de paramètres jusqu'à la section *Addressing* pour y saisir l'adresse MAC du récepteur. Cette dernière est visible sur le module, mais vous pourrez aussi la déterminer avec XCTU. La première partie (les 32 bits de poids fort) de l'adresse sur 64 bits est saisie dans le champ *Destination Address High* ; les 32 bits restants, dans le champ *Destination Address Low*.

Dernière chose : configurez les broches d'E/S du module dans la section *I/O settings* (presque tout en bas de la liste de paramètres). Le **tableau 2** et la **figure 5** montrent les valeurs à utiliser. Réglez *I/O Sampling* sur 64 (au-dessous des *IO settings*).

Enfin, cliquez sur *Write*.

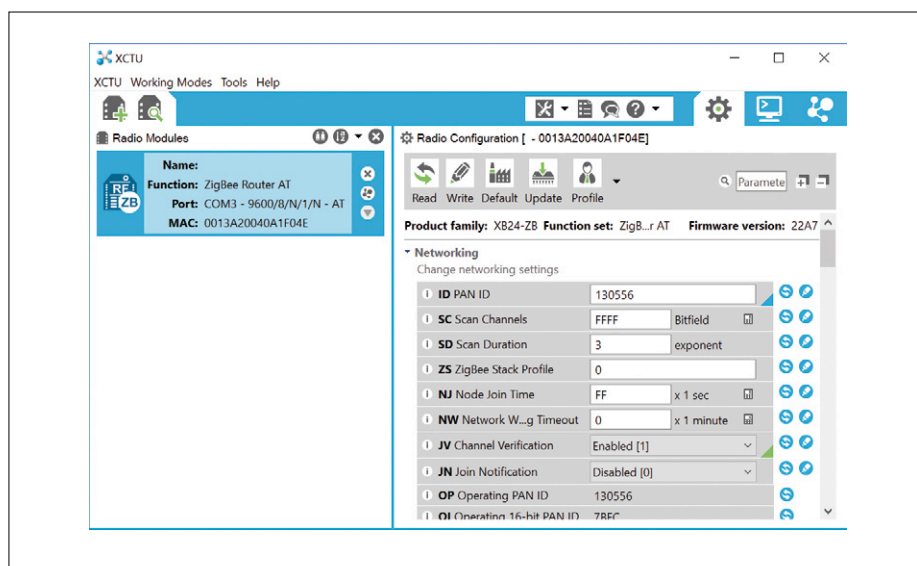


Figure 3. XCTU a détecté un module XBee sur COM3.

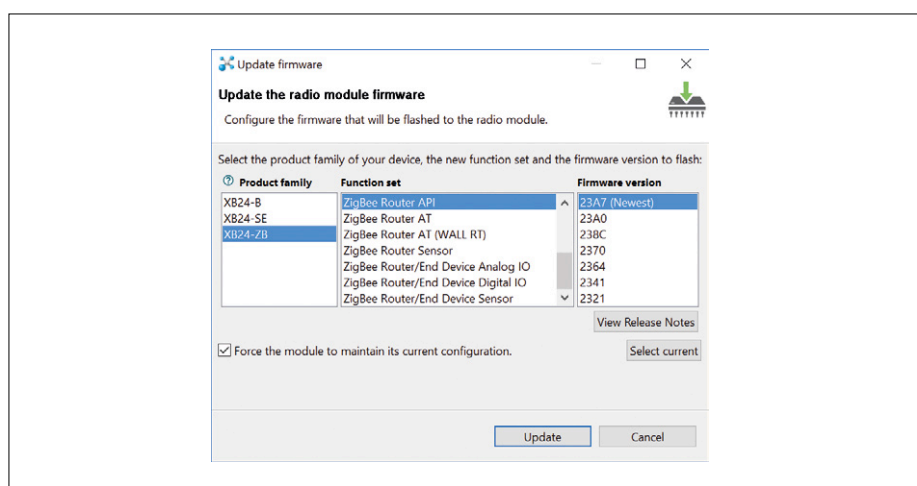


Figure 4. Il faut activer la fonction *ZigBee router API* sur l'émetteur.

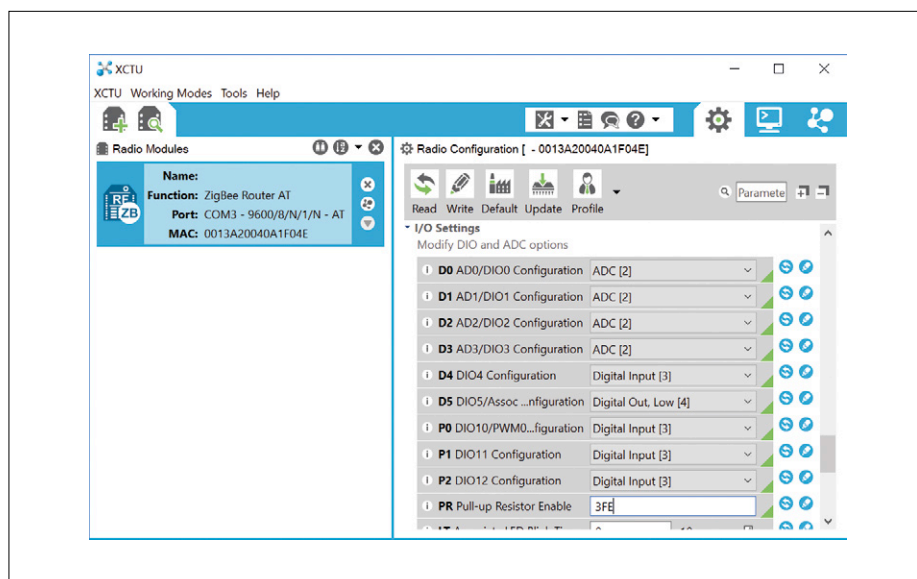
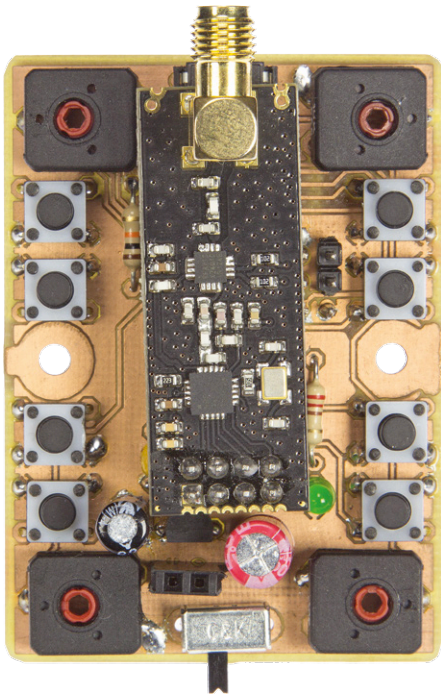


Figure 5. Configuration des broches d'E/S de l'émetteur XBee.



Configuration du récepteur XBee

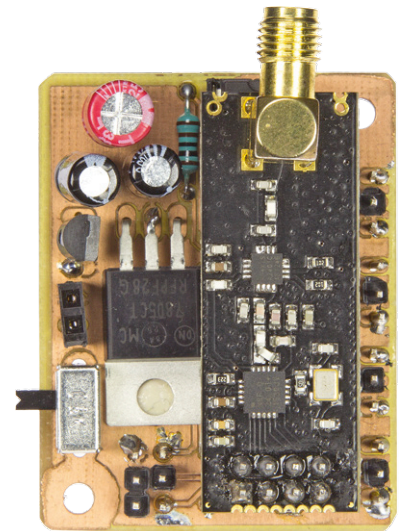
Dans votre dispositif de programmation des modules XBee, remplacez maintenant l'émetteur par le récepteur. Reliez le tout à l'ordinateur ; dans XCTU, cliquez sur le bouton *Discover radio modules*.

Cliquez ensuite sur le bouton *Read* une fois le module détecté.

Le module récepteur sera programmé avec le jeu de fonctions *ZigBee Coordinator API*. Cliquez sur *Update*, sélectionnez la famille de produits *XB24-ZB*, choisissez la fonction *ZigBee Coordinator API* et la version *21A7* du micrologiciel puis cliquez sur le bouton *Update*.

Dans la section *Networking*, saisissez le même *PAN ID* que celui utilisé pour l'émetteur. Dans la section *Addressing*, saisissez l'adresse de l'émetteur de la même façon que vous avez saisi celle du récepteur lors que la configuration de l'émetteur.

Cliquez ensuite sur *Write*.

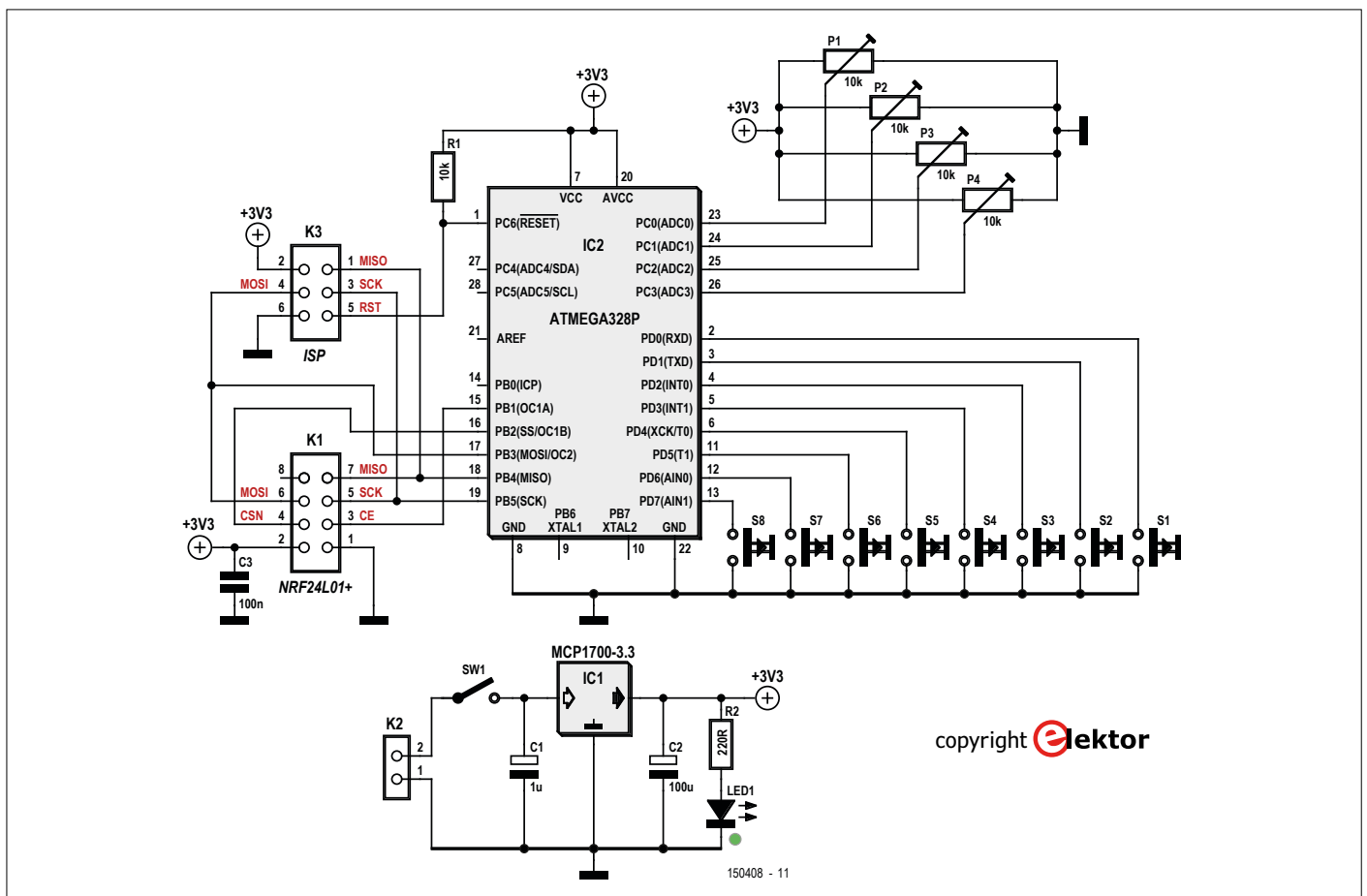


Avant de pouvoir utiliser le récepteur, il faut bien entendu programmer son microcontrôleur ; le micrologiciel est disponible sur [2]. Reliez un programmeur AVR *in situ* à K8. N'oubliez pas de régler les valeurs des fusibles (**tableau 3**). Enfichez maintenant les modules XBee sur les cartes ; attention à ne pas les intervertir. Reliez également les servos à la carte du récepteur. Mettez sous tension le récepteur et l'émetteur, l'ordre n'a pas

d'importance. Une fois la connexion entre les deux cartes établie (cela peut prendre quelques secondes), vous devriez être en mesure de commander les servos en réglant les ajustables de l'émetteur. Si vous reliez l'émetteur à un port série, vous pourrez voir les messages de retour envoyés par le récepteur.

Passons aux modules nRF24...

Une radiocommande basée sur des



copyright **elektor**

Figure 6. L'émetteur à nRF24 utilise un bus SPI pour la communication entre le µC et le module.

modules XBee est sans nul doute intéressante grâce à toutes les configurations réseau possibles, mais cela reste cher. Une solution moins onéreuse est de remplacer les modules XBee par des modules radio nRF24L01 plus abordables et disponibles partout sur l'internet. Avec une bonne antenne, il est possible d'atteindre la même portée qu'avec un XBee-Pro. Même s'il faut ajouter un µC externe, le coût total reste inférieur à celui de la solution à base de modules XBee.

L'émetteur nRF24

La figure 6 montre le schéma de l'émetteur à cœur nRF24. Il arbore les mêmes poussoirs et ajustables que son homologue XBee (fig. 1) ; le module XBee est remplacé par un µC ATmega328 et le connecteur K1 pour connecter le module radio nRF24. Nota : la liaison entre le µC et le module est un bus SPI, et non pas une ligne série asynchrone.

Grâce au nombre élevé de ports d'E/S du µC, les diodes pour coder les signaux des poussoirs ne sont pas nécessaires. Aussi, l'alimentation de 1,2 V n'est plus nécessaire : tout peut fonctionner en 3,3 V.

Table 1. Réglage du décalage des servos.

	'1'	'0'	'TRIM_DOWN'	'TRIM_UP'	Décalage du servo
S1	0	0	0	1	voie 1 +
S2	0	0	1	0	voie 1 -
S3	0	1	0	1	voie 2 +
S4	0	1	1	0	voie 2 -
S5	1	0	0	1	voie 3 +
S6	1	0	1	0	voie 3 -
S7	1	1	0	1	voie 4 +
S8	1	1	1	0	voie 4 -

Le récepteur nRF24

Le schéma du récepteur à nRF24 (fig. 7 ; notre prototype fig. 8) est presque identique à celui du récepteur XBee (fig. 2), à la différence près que le module XBee est remplacé par K1 pour connecter le module radio nRF24. Comme pour l'émetteur nRF24, la liaison entre le module radio et le µC est un bus SPI.

Relier l'émetteur et le récepteur nRF24

Cette étape est plutôt simple : les logiciels de l'émetteur et du récepteur font le boulot. Il vous suffit de programmer les micros avec le bon logi-

ciel. Consultez l'encadré pour les détails. Pour éviter les interférences entre plusieurs systèmes de radiocommande à nRF24, vous pouvez changer les adresses réseau utilisées. En haut des croquis de l'émetteur et du récepteur, vous trouverez une ligne semblable à la suivante :

```
uint64_t address[] =
    {0xFFFFFFFF, 0xFFFFFFFF};
```

Remplacez dans un des fichiers les valeurs ci-dessus par deux autres nombres à 64 bits (ce que vous voulez) tout en vous assurant que les deux valeurs sont différentes puis copiez la ligne modifiée dans l'autre fichier. Recompilez, reprogrammez, et c'est fait ! Reliez les servos à la carte du récepteur ; mettez le récepteur et l'émetteur sous tension (l'ordre n'a pas d'importance). Une fois la connexion établie

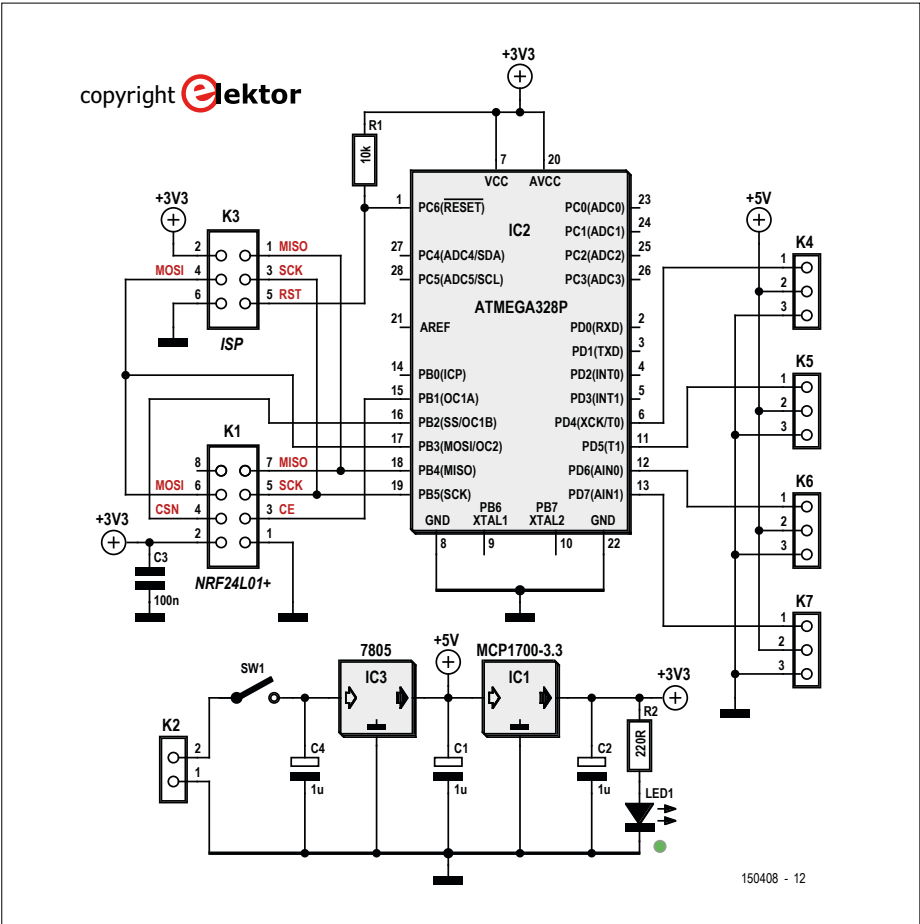


Figure 7. Le récepteur nRF24 est presque identique au récepteur XBee.

Table 2. Paramètres pour les ports d'E/S de l'émetteur XBee.

E/S	Valeur
D0	ADC [2]
D1	ADC [2]
D2	ADC [2]
D3	ADC [2]
D4	Digital Input [3]
D5	Digital Out, Low [4]
P0	Digital Input [3]
P1	Digital Input [3]
P2	Digital Input [3]
PR	3FE

Table 3. Valeurs à programmer dans les fusibles de tous les microcontrôleurs utilisés dans cet article.

Fusible	Valeur
Low	0xE2
High	0xDA
Extended	0x05

(cela peut prendre quelques secondes), vous devriez pouvoir commander les servos avec les ajustables. Si vous reliez le récepteur à un terminal série, les messages de retour s'afficheront.

Le mot de la fin

Nous avons présenté deux architectures fondamentales pour un système de radiocommande avec échange bidirectionnel de données : XBee et nRF24. Le choix dépend de votre application et de votre budget.

Le logiciel n'est capable de commander que des servos ; vous l'adapterez pour répondre à vos besoins. Cela devrait être facile : nous avons mis à votre disposition les sources des programmes, compatibles avec l'EDI Arduino. ◀

(150408 -
version française :
Kévin Petit)

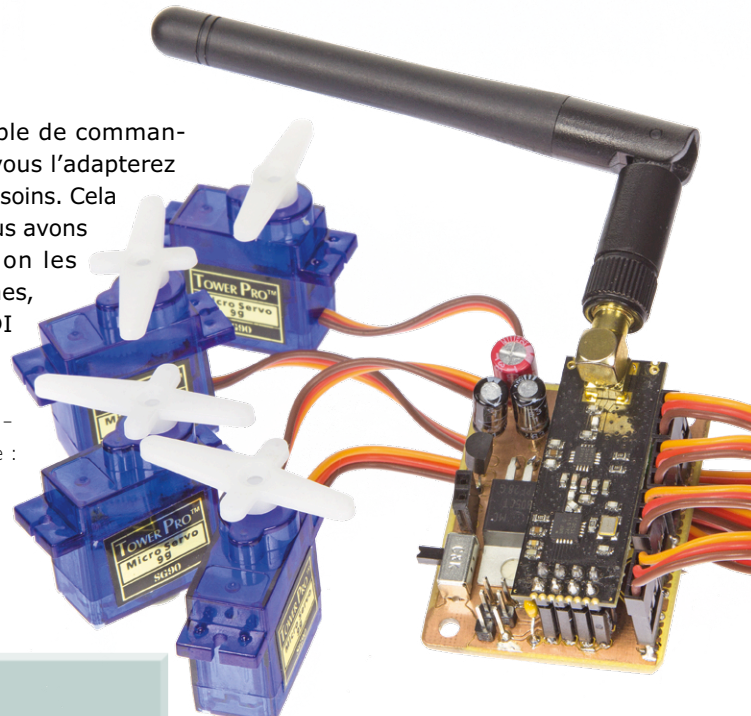


Figure 8. Notre prototype du récepteur nRF24 relié à quatre petits servos.



LISTE DES COMPOSANT ÉMETTEUR nRF24 (150408-1)

Résistances

P1-P4 = 10 kΩ ajustable de précision
R1 = 10 kΩ
R2 = 220 Ω

Condensateurs

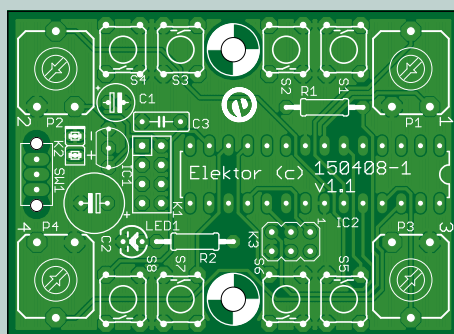
C1 = 1 µF, 6,3 V, au pas de 2,5 mm
C2 = 100 µF, 6,3 V, au pas de 3,5 mm
C3 = 100 nF

Semi-conducteurs

IC1 = MCP1700-3302E/TO
IC2 = ATmega328P-PU, programmé
(réf. 150408-41)
LED1 = verte, 3 mm

Divers

K1 = barrette sécable femelle double rangée
à 8 voies (2×4), au pas de 2,54 mm
K2 = barrette sécable à 2 voies,
au pas de 2,54 mm
K3 = barrette sécable double rangée à 6 voies



(2×3), au pas de 2,54 mm
S1 à S8 = poussoirs 6×6 mm
SW1 = interrupteur à glissière
(OS102011MS2QN1C)
nRF24L01+PA+LNA (+ antenne)
circuit imprimé réf. 150408-1



LISTE DES COMPOSANTS RÉCEPTEUR nRF24 (150408-2)

Résistances

R1 = 10 kΩ
R2 = 220 Ω

Condensateurs

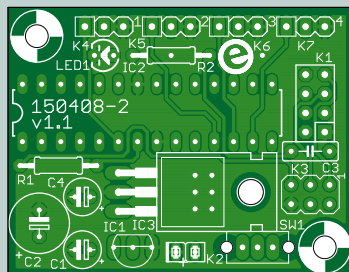
C1, C4 = 1 µF, 6,3 V, au pas de 2,5 mm
C2 = 100 µF, 6,3 V, au pas de 3,5 mm
C3 = 100 nF

Semi-conducteurs

IC1 = MCP1700-3302E/TO
IC2 = ATmega328P-PU, programmé
(réf. 150408-42)
IC3 = 7805
LED1 = verte, 3 mm

Divers

K1 = barrette sécable femelle double rangée
à 8 voies (2×4), au pas de 2,54 mm
K2 = barrette sécable à 2 voies,
au pas de 2,54 mm



K3 = barrette sécable double rangée à 6 voies
(2×3), au pas de 2,54 mm
K4-K7 = barrette sécable à 3 voies,
au pas de 2,54 mm
SW1 = interrupteur à glissière
(OS102011MS2QN1C)
NRF24L01+PA+LNA (+ antenne)
circuit imprimé réf. 150408-2

Liens

- [1] www.digi.com/products/xbee-rf-solutions/xctu-software/xctu
- [2] www.elektormagazine.fr/150408



DANS L'E-CHOPPE

version nRF24

→ 150408-1

circuit imprimé de l'émetteur

→ 150408-2

circuit imprimé du récepteur

→ 150408-41

microcontrôleur de l'émetteur
programmé

→ 150408-42

microcontrôleur du récepteur
programmé

version XBee

→ 150408-3

circuit imprimé de l'émetteur

→ 150408-4

circuit imprimé du récepteur

→ 150408-43

microcontrôleur du récepteur
programmé

Outils

→ 140374-91

T-Board sans fil

→ 150387-1

Port série USB à haute vitesse



vol tous azimuts

piloté par **Clemens Valens** (labo d'Elektor)

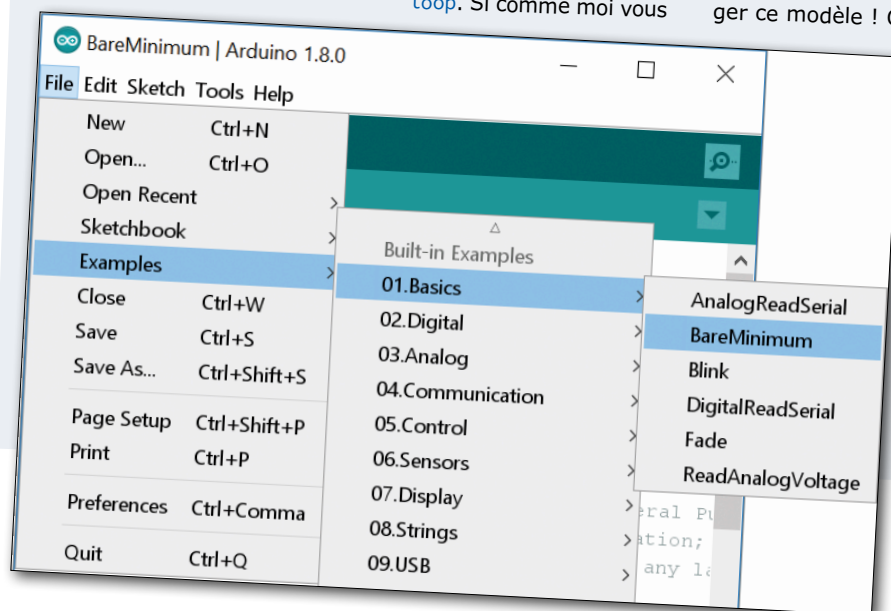
Secret d'Arduino : nouveau modèle de croquis

Lorsque vous ouvrez l'EDI Arduino, ou lorsque vous créez un nouveau croquis par le menu Fichier → Nouveau, l'éditeur contient déjà des squelettes vides pour les fonctions `setup` et `loop`. Si comme moi vous

n'aimez pas la façon dont sont écrits ces squelettes, vous devez les modifier à chaque fois, ce qui est presque aussi long que de les écrire à partir de rien. Si seulement on pouvait changer ce modèle ! On peut et voici comment. Ouvrez le fichier

`<arduino>\examples\01.Basics\BareMinimum\BareMinimum.ino` dans Notepad ou un autre éditeur, modifiez-le et sauvez-le. La prochaine fois que vous créez un nouveau croquis, il utilisera ce modèle. Vous ne pouvez pas faire ça dans l'EDI parce qu'il empêche l'écrasement du modèle. Un modèle personnalisé est un très bon moyen d'ajouter automatiquement des lignes de code (comme `Serial.begin(115200);`) ou un texte de licence que vous utilisez dans tous vos croquis.

(160354-a)



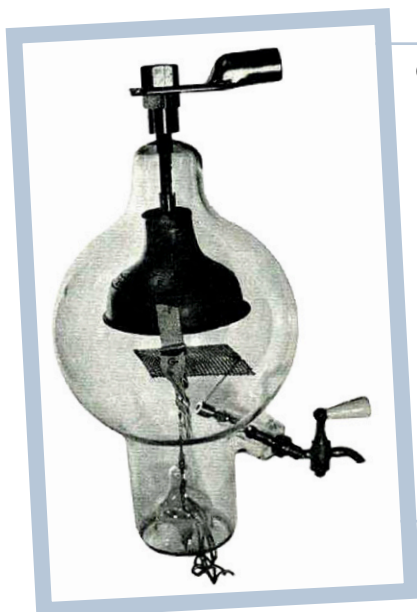
Outil de labo indispensable

Les bricoleurs amateurs aiment les gadgets et les outils pratiques (allo ? 14 est-il le numéro atomique du silicium ?) et nous sommes toujours à l'affut de bons candidats à ajouter à la pile déjà impressionnante sur leurs étagères. Cette fois nous présentons un outil à énergie solaire qui devrait plaire à ceux qui sortent de leur atelier de temps en temps (oui, ça existe !) pour aller se balader dans la nature où ils risquent d'être attaqués par des moustiques et même de se perdre. Dans ce cas, la boussole leur permettra de retrouver le chemin du retour en toute sécurité sans être piqués par ces pénibles insectes. En forme de mousqueton bien pratique, il s'attache à une boucle de ceinture et vous accompagne partout.

(160354-d)

Répulsif à moustiques Digisonic 12D8 solaire, avec boussole, disponible sur l'internet.





Composants particuliers

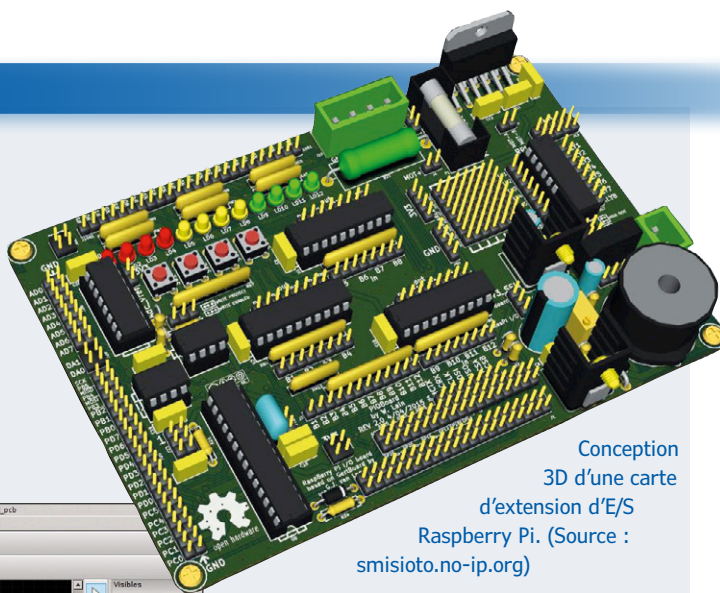
Ailleurs dans ce numéro, vous pouvez découvrir le Phantastron, un circuit multivibrateur spécial de grande précision inventé pendant la Seconde Guerre mondiale. En effet le Phantastron est un circuit très intelligent, mais je suis sûr que si son créateur Alan Dower Blumlein avait connu le phantasatron Umac 606, le circuit aurait été encore meilleur. L'Umac 606, commercialisé à peu près à la même époque, était un phantasatron à faisceau hélicoïdal et anode infernale avec un filament en tungstène uriné obtenu par un procédé spécial de triple distillation d'un isotope d'uranium duquel tous les neutrons avaient été éliminés. La grille était faite du métal rare senileium, choisi pour son absence totale d'émission. Un vide doublement pompé permettait de voir clairement la plaque non émettrice de type plongeur à triple traitement. Grâce à sa construction autonettoyante à grande eau, ce tube restait utilisable pendant toute sa durée de vie. Pour plus de détails, consultez la fiche technique : www.tubecollectors.org/archives/606.pdf

(160354-c)

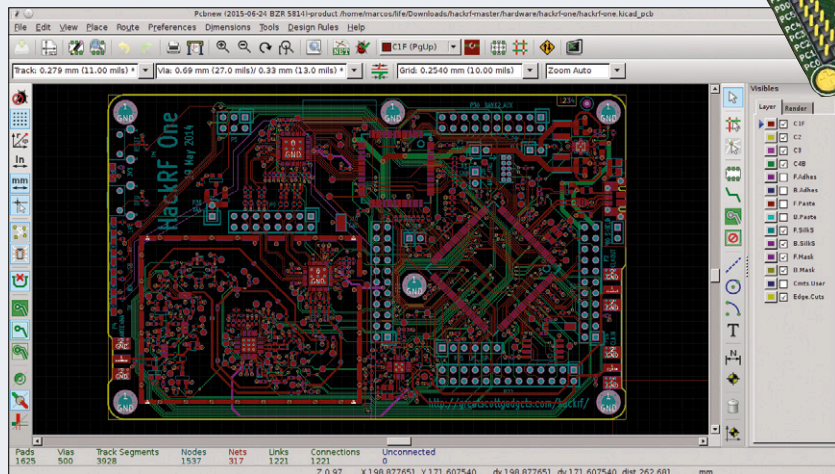
Le Phantasatron Umac 606.

Bibliothèque CAO 3D KiCad

Ces dernières décennies, la communauté du logiciel ouvert a produit beaucoup de très bons produits, certains complètement gratuits même pour des applications commerciales. Créés et maintenus par des milliers d'enthousiastes, en équipes ou par des individus qui ne se rencontrent presque jamais dans la réalité, on ne peut qu'admirer l'énergie et l'effort déployés sans rien attendre en retour. Linux est le premier exemple qui vient à l'esprit, mais il y en a bien d'autres. KiCad EDA (prononcez « Kikad »), une suite multiplateforme d'automatisation de la conception électronique, en est un autre. Selon Wikipédia le



Conception 3D d'une carte d'extension d'E/S Raspberry Pi. (Source : smisioto.no-ip.org)



PcbNew, éditeur de circuits imprimés de KiCad. (Source : kicad-pcb.org)

projet a été lancé en 1992 par Jean-Pierre Charras. Ma première expérience avec KiCad remonte à 2003. À cette époque il était utilisable, mais inadapté à mon travail quotidien. Toutefois, KiCad n'a cessé d'évoluer et aujourd'hui, 25 ans après, c'est un outil de niveau professionnel pour la conception de schémas et circuits imprimés avec visualisation 3D.

Le succès de ces projets communautaires repose sur la contri-

bution de nombreuses personnes. Dans le cas de KiCad, les contributions se font sous la forme de bibliothèques de composants et de tutoriaux, mais aussi d'extensions fonctionnelles comme le routeur interactif de type « push-and-shove » développé par la très sérieuse équipe « Hardware and Timing » du laboratoire international de recherche nucléaire du CERN. Quant à moi ma contribution personnelle se limite à ce modeste article pour faire passer le mot. Mon attention a été attirée récemment par un site italien maintenu par un certain Walter, de Padoue, offrant une bibliothèque de

composants très complète pour KiCad. Outre les bibliothèques, le site propose aussi quelques techniques pour concevoir de jolis modèles de composants 3D, aussi jetez-y un coup d'œil avant de concevoir les vôtres.

<http://smisioto.no-ip.org/elettronica/kicad/kicad-en.htm>
<http://kicad-pcb.org/>

(160354-b)

Fais-tu confiance à ton assistante ?

Je crois que ça a commencé avec Siri d'Apple, l'assistant personnel et navigateur de la connaissance « intelligent » intégré aux iPhones. Les concurrents lui ont emboîté le pas et aujourd'hui nous avons Cortana de Microsoft, l'Assistant de Google et Viv de Samsung. Ils ont tous des qualités et des défauts qu'ils corrigeront sans doute rapidement. Étonnamment le concurrent le mieux placé pour tous les surclasser est Alexa d'Amazon. Implantée dans le très réussi « haut-parleur »

à commande vocale Echo et ses déclinaisons Echo Dot et Tap, Alexa était associée à plus de 30 nouveaux produits présentés lors du *Consumer Electronics Show* (CES) qui se tient tous les ans à Las Vegas. Commande vocale ne signifie pas ici qu'elle peut aussi prononcer des mots, mais qu'elle est commandée par la voix. Vous lui demandez de passer un disque d'Elvis, Alexa passe Elvis.

Alexa doit son succès à son ouverture. Siri et Cortana étaient d'abord fermés, mais Alexa a été une plateforme ouverte dès le début, incitant les concepteurs à ajouter des « savoir-faire ». Par exemple commander une pizza ou changer la couleur d'une LED simplement en le disant. Google Assistant est ouvert aussi, puisqu'il autorise les actions de tiers. Devant le succès d'Alexa, Microsoft et Apple ont depuis peu permis l'accès des concepteurs à Cortana et Siri.

Contrairement à Siri et Cortana qui utilisent la puissance de calcul de la machine qui les héberge, Alexa passe par les services en ligne d'Amazon pour le gros du travail, ce qui en fait un système de commande vocale léger pour toutes sortes d'applications. Service en ligne implique que cela ne fonctionne qu'avec une connexion à l'internet. Alexa se met en éveil dès qu'elle entend le mot déclic « Alexa » (qui peut être personnalisé). Elle capture alors le son et crée un flux de données sur l'inter-

net vers le serveur où il est analysé et interprété. Il y est aussi stocké pour de futures références, pour améliorer les capacités de reconnaissance vocale, et c'est là que les problèmes de vie privée surgissent. En effet, un dispositif capable d'envoyer des conversations privées vers le *Cloud* où elles sont stockées est une atteinte potentielle à la vie privée. Les *Autorités*, ou des pirates informatiques, pourraient pour des raisons plus ou moins avouables, vouloir profiter d'une base de données si intéressante. Amazon déclare avoir refusé



ROGERS © 2015 Pittsburgh Post-Gazette. Reproduit avec la permission de ANDREWS MCMEEL SYNDICATION. Tous droits réservés.

nette demande d'accès à ces données jusqu'à maintenant, mais nul ne sait ce qu'il en sera dans le futur. Les usagers peuvent effacer leurs enregistrements, mais ce faisant ils dégradent l'assistant intelligent, donc sans doute peu de personnes le feront. Êtes-vous prêts à renoncer à votre vie privée pour commander une pizza sans avoir à composer un numéro ?

(160354-e)

25 000 000 000 de microcontrôleurs ont été expédiés en 2016.

Onyx Coonect, la première fabrique d'ordiphones et tablettes en Afrique.

Vous voulez participer ?

Envoyez vos commentaires, suggestions, trucs et astuces à labs@elektor.com

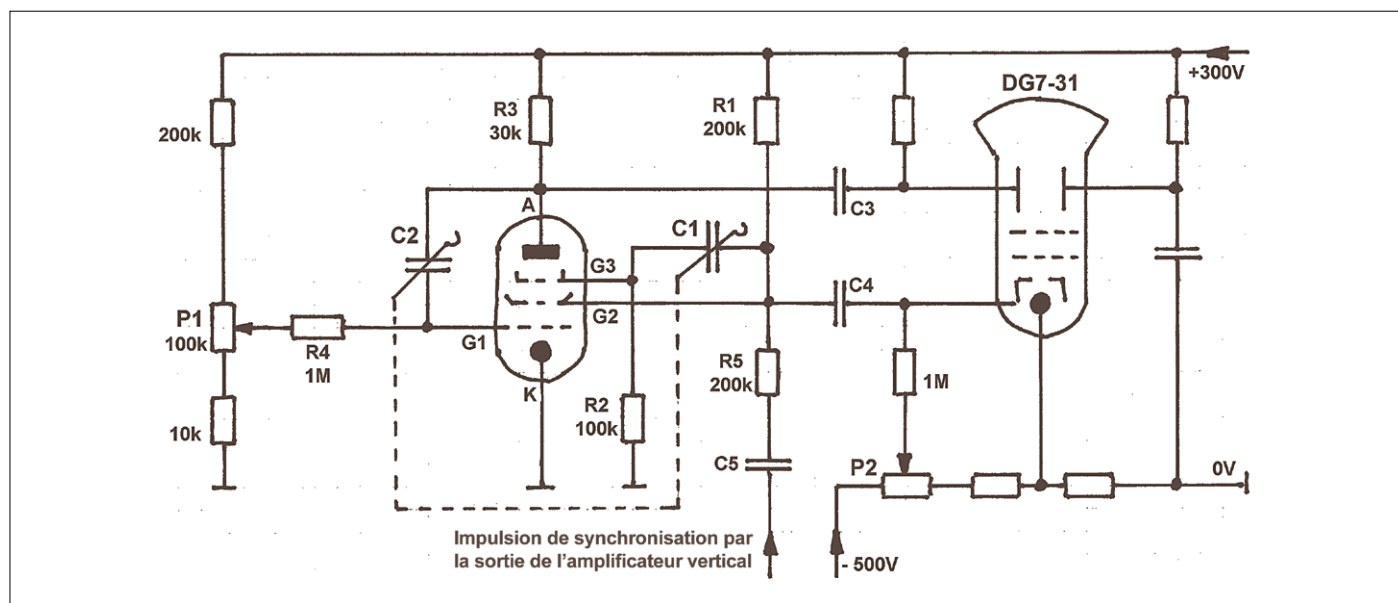
(160354 – version française : Denis Lafourcade)

le phantatron

drôle de composant n°30

Lutz Bergmann (Allemagne)

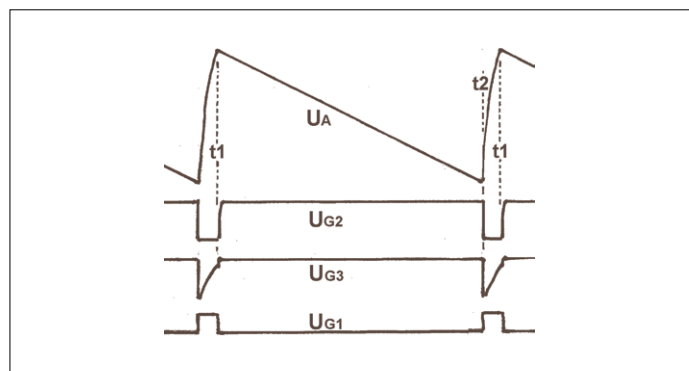
En dépit de son extrême linéarité et de sa précision, ce circuit très ingénieux est presque tombé dans l'oubli aujourd'hui. Il mérite une meilleure reconnaissance.



Il y a des composants électroniques au nom à désinence -tron, nous avons vu le Fetron, le magnétron, le klystron, le thyatron. Ce sont des composants bizarres, mais cette fois-ci, nous ne dévoilerons pas un objet physique, mais un type de circuit hors du commun. Il a été créé en 1940 par un personnage tout aussi étrange, Alan Blumlein, l'inventeur du son stéréophonique, de l'enregistrement sonore, de la télévision et du radar [1]. Dans le phantatron, on utilise une penthode d'une manière non conventionnelle et même sérieusement décriée comme amplificateur. Mais ici, on ne l'utilise pas comme amplificateur, mais comme oscillateur à relaxation ! Le circuit exploite un effet spécial pour former une tension en dents de scie [2, 3] en se basant sur le fait que le courant de grille-écran (G2) de la penthode augmente brutalement quand la tension d'anode

passse en dessous de celle de grille-écran. L'attraction des électrons se réduit par rapport à la plaque et augmente en faveur de l'électrode qui porte la plus haute tension. Si la grille-écran reçoit son courant à travers une résistance, ce moment est marqué d'une chute de sa tension, alors qu'elle était jusque-là indépendante du potentiel d'anode. On peut transférer cette variation négative au moyen du condensateur C1 à la grille d'arrêt (G3). Celle-ci agit alors comme une grille de commande supplémentaire qui, maintenant négative, empêche le passage du restant de courant d'anode, lequel s'annule subitement. La tension d'anode peut de nouveau grimper vite. Avec ce fonctionnement bistable, un intégrateur de Miller peut se transformer en oscillateur en dents de scie auto-entretenu.

Examinons le cycle à partir de la fin de la coupure de courant, disons l'instant t_1 . La tension de la plaque y est au maximum, vu qu'il n'y a pas de courant. Comme le condensateur C1 ne peut rendre la grille d'arrêt négative qu'un temps limité, le courant d'anode recommence à passer. La chute de tension sur la résistance R3 augmente et le potentiel d'anode U_A descend. Cette baisse provoque un courant vers C2 qui tend à faire descendre la tension sur la grille de commande G1, ce qui modère la hausse du courant d'anode. Un équilibre s'établit entre les courants dans C2 et R4. Comme le courant dans R4 est constant, il en va de même avec C2. La tension d'anode décroît alors avec une linéarité enviable. Cet effet a été découvert en 1919 par John Milton Miller [4]. On peut régler le taux de descente de la tension de plaque d'une infinité de façons



avec le potentiomètre P1 dans un rapport de 1 à 10 et par bonds selon des valeurs échelonnées de la capacité de C2. Le processus se répète indéfiniment, pour autant que la tension de plaque ne devienne pas plus basse que celle de grille-écran, ce qui annulerait le courant d'anode, dont le potentiel remonterait encore une fois, mais pas à n'importe quelle vitesse, puisqu'il faut recharger C2. C'est la constante de temps $C2 \times R3$ qui en décide. Le potentiel de la grille d'arrêt G3 doit rester négatif assez longtemps pour cela. En conséquence, C1 et C2 doivent rester appariés. Si vous modifiez la valeur de C2 au moyen d'un commutateur, il faut effectuer le même changement sur C1. À partir de 1945, cet oscillateur à relaxation a pris une grande importance pour la construction d'oscilloscopes simples, en raison du prix encore élevé des tubes à vide à l'époque. Une seule penthode permettait alors d'une manière « fantastique » de fournir tous les signaux requis pour assurer la déflexion horizontale (axe des x et du temps) sur les tubes à rayons cathodiques. La tension en dents de scie atteignait 200 V_{c.-à-c.} sans composant additionnel pour la déflexion d'un tube cathodique DG7-31. Le condensateur C3 l'appliquait alors à l'une des plaques de déflexion horizontale (déclenchement asymétrique pour un tube DG7-31 à écran de 7 cm de diamètre). Avec le condensateur C4, on pouvait encore envoyer l'impulsion négative de la grille-écran au cylindre de Wehnelt [5] pour masquer à l'écran (*blanking*) le retour du faisceau.

Un pareil oscilloscope n'offrait que peu de facilités d'utilisation. Le balayage horizontal était par principe relaxé, sans synchronisation. Il fallait ajuster la fréquence de l'oscillateur pour stabiliser l'affichage du signal à observer. Pour la synchronisation,

on reliait la sortie de l'amplificateur vertical (axe des y) à la grille-écran par une cellule RC (R5/C5).

Avec les progrès de la déflexion horizontale déclenchée et la chute des prix des tubes à vide, le phantatron a quitté la scène depuis si longtemps qu'on a du mal à s'en souvenir. Les transistors sont apparus, puis les circuits intégrés, ils ont renvoyé dans l'occultation les lampes de radio, alors que les générateurs de balayage ont tellement facilité le déclenchement qu'on les a presque oubliés.

Vous trouverez davantage d'information sur l'internet, y compris la version à semi-conducteurs de ce circuit [6] et la stupéfaction lors de la déclassification de son degré de sécurité [7].

(160281 – version française : Robert Grignard)

Liens

- [1] Alan Blumlein : https://en.wikipedia.org/wiki/Alan_Blumlein
- [2] Transistron Miller : www.r-type.org/articles/art-135.htm
- [3] Base de temps Miller : www.r-type.org/articles/art-136.htm
- [4] Effet Miller : https://fr.wikipedia.org/wiki/Effet_Miller
- [5] Cylindre de Wehnelt : https://en.wikipedia.org/wiki/Wehnelt_cylinder
- [6] Site Radar Basic : www.radartutorial.eu/17.bauteile/bt52.en.html
- [7] Sécurité (p. 380) : <http://americanradiohistory.com/Archive-Wireless-World/40s/Wireless-World-1946-11.pdf>

Trouvez Elektor en kiosque



avec votre code postal...

Où trouver Elektor près de chez vous ?



Si vous cherchez où acheter votre magazine Elektor, nous vous aidons à le trouver facilement.

Rendez-vous sur le site www.elektormagazine.fr, cliquez sur l'onglet **MAGAZINE**, puis sur **Elektor en kiosque**.

Ensuite indiquez votre code postal, puis cliquez sur **TROUVER**.

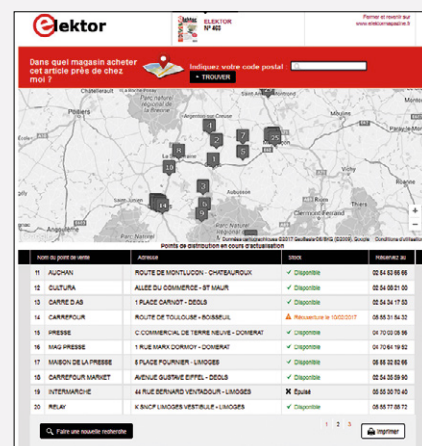
Vous verrez apparaître une carte avec des repères chiffrés qui renvoient chacun à un des points de vente énumérés. Vous connaîtrez aussitôt l'état du stock chez ces différents revendeurs : magazine disponible ou épuisé.

Surveillez la carte pour découvrir les jours et horaires d'ouverture de chaque magasin.

Il ne reste qu'à choisir le plus commode pour y acheter votre magazine.

Ce moteur indique également la date de réouverture des points de vente fermés au moment de votre recherche.

Toutes ces informations, fournies par notre diffuseur (Messageries de Presse Lyonnaises), sont mises à jour quotidiennement et proviennent des 14.000 magasins informatisés de son réseau.



N°	Nom du magasin	Adresse	Statut	Horaires
11	AUDIN	ROUTE DE MONTAUDOU - ORTALBOURG	✓ Disponible	08.04.01.01.01
12	CULTURA	ALLÉE DU DOMINIQUE - ST SAUVEUR	✓ Disponible	02.04.02.01.01
13	CARTE D'AS	1 PLACE CARNOT - DESLIS	✓ Disponible	02.04.03.01.01
14	CARTEPOUR	ROUTE DE TOULOUSE - ROSSIGNOL	✓ Disponible	08.04.01.01.01
15	IMPRESSE	C COMMERCIAL DE TERRE NEUVE - DONNET	✓ Disponible	04.04.01.01.01
16	IMPRESSE	1 RUE MARX DONNET - DONNET	✓ Disponible	04.04.01.01.01
17	MAISON DE LA PRESSE	5 PLACE FOURNIER - LINDOIS	✓ Disponible	08.04.01.01.01
18	CARTEPOUR MARNET	AVENUE GUSTAVE EIFFEL - DESLIS	✓ Disponible	02.04.03.01.01
19	INTERMARCHÉ	44 RUE BERNARD VENTROUX - LINDOIS	✓ Disponible	08.04.01.01.01
20	RELAY	10000 LINDOIS VERTICALE - LINDOIS	✓ Disponible	08.04.01.01.01

passerelle IoT et nœuds sans fil

2^e partie : le logiciel



L'auteur a cherché un système domotique commercial qui réponde à ses exigences. Resté sur sa faim, il a décidé de concevoir le sien : dans la bande des 868 MHz, plusieurs « nœuds terminaux » (capteurs et actionneurs) communiquent avec une passerelle centrale. Celle-ci utilise le protocole MQTT pour envoyer des données de mesure à un serveur OpenHAB, qui les traite et les supervise [1]. Dans le 1^{er} article [2], nous avons abordé le matériel. Ici nous nous intéressons au logiciel qui anime le tout.

Hennie Spaninks (Pays-Bas)

Nous traiterons ici du code pour les nœuds terminaux, la passerelle et le serveur sur RPi. L'auteur a écrit lui-même les croquis (*sketches*) Arduino et il a bien entendu fait appel à diverses bibliothèques à code source ouvert.

Nœud terminal

Croquis Arduino pour le nœud terminal : du code pour piloter la liaison radio, lire les capteurs et commander les sorties.

Passerelle (*gateway*)

Croquis Arduino pour la passerelle : du code pour piloter la liaison radio et décoder les messages. Un client MQTT échange les messages avec le courtier (*broker*).

Serveur domotique (Raspberry Pi)

Sur le Raspberry Pi tournent à la fois le courtier MQTT (Mosquitto) et le serveur domotique OpenHAB.

Pour compiler le code Arduino, nous utilisons l'EDI Arduino

standard, épaulé par les bibliothèques mentionnées plus loin. Pour le Raspberry Pi, on utilisera de préférence la version la plus puissante (Rpi 3), encore que cela marche aussi avec le Raspberry Pi 2. Notre SE sera Raspbian.

Nœud terminal

Le code pour les différents nœuds terminaux est disponible en [3]. Le noyau du code est identique pour tous les nœuds terminaux. Nous prenons comme exemple le nœud terminal DHT [4]. Ce nœud comporte une entrée numérique (un bouton-poussoir) sur l'entrée 8, une sortie numérique (relais/LED) sur la broche 9 et un capteur d'humidité/température sur la broche 4. Si on presse le bouton, le nœud peut exécuter localement deux fonctions :

- Démarrer un temporisateur (*timer*) et mettre la sortie au niveau haut. Réinitialiser la sortie une fois la durée du temporisateur écoulée. La durée de la temporisation est réglée avec l'appareil 7 (cf. tableau 2 dans [2]). Un « 0 » bloque la fonction.
- Basculer la sortie après pression du bouton-poussoir. Ceci

permet d'activer et de désactiver la sortie manuellement en local. Pour ce faire, l'appareil 6 doit être sur « ON ».

Indépendamment de ces deux fonctions, chaque changement de niveau de l'entrée numérique (broche 8) provoque l'envoi d'un message d'état via l'appareil 40. Les valeurs du capteur d'humidité/température peuvent être lues via l'appareil 48/49. Des transmissions périodiques sont aussi possibles.

Ce code fait appel à trois bibliothèques :

- Bibliothèque RFM69 de Felix Ruso [5].
- Bibliothèque SPI pour la commande du bus SPI.
- Bibliothèque DHT pour la lecture du capteur DHT11.

Le code commence par quelques paramètres de configuration (à partir de la ligne 60) :

```
#define NODEID 2
// unique node ID within the closed network
#define GATEWAYID 1
// node ID of the Gateway is always 1
#define NETWORKID 100
// network ID of the network
#define ENCRYPTKEY "xxxxxxxxxxxxxxxx"
// 16-char encryption key; same as on Gateway!
//#define DEBUG
// uncomment for debugging
#define VERSION "DHT V2.2"
// this value can be queried as device 3
```

NODEID attribue au nœud terminal une adresse unique sur le réseau sans fil. Cet ID(entificateur) est également utilisé dans le sujet (*topic*) MQTT pour s'adresser au nœud. Le nodeID de la passerelle est toujours « 1 ». Un nœud terminal ne communique qu'avec la passerelle ; il n'y a donc pas de trafic internœuds terminaux.

NETWORKID permet de définir des réseaux fermés. Seuls les nœuds du même réseau peuvent communiquer entre eux. Dans ce projet, nous utilisons un réseau fermé, de sorte que l'ID du réseau doit être le même sur la passerelle et tous les nœuds terminaux.

ENCRYPTKEY est la clé de cryptage du trafic sans fil. Cette clé comporte seize caractères et doit être la même pour tous les nœuds du réseau.

Le paramètre DEBUG est une bascule d'activation du mode de débogage. Si ce paramètre n'est pas défini (en faisant de cette ligne un commentaire par l'ajout de « // »), aucune information de débogage n'est produite. La suppression de « // » active le débogage, des messages d'état sont alors envoyés sur le port série.

Une commande adéquate du module RFM69 requiert quelques paramètres supplémentaires :

```
#define FREQUENCY RF69_868MHZ
#define IS_RFM69HW // uncomment only for RFM69HW!
#define ACK_TIME 50 // max # of ms to wait for an ack
```

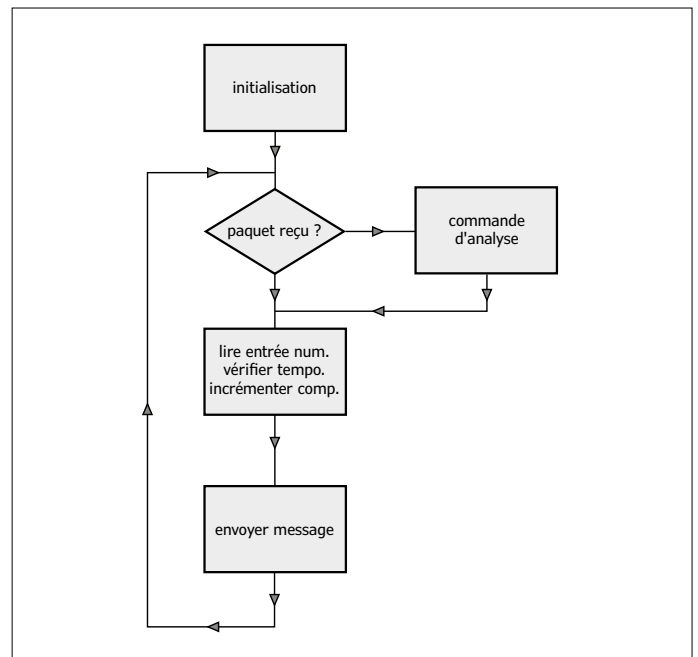


Figure 1. Ordigramme du nœud terminal.

FREQUENCY indique la fréquence utilisée (en Europe, 433 ou 868 MHz). IS_RFM69HW indique que vous utilisez la version forte puissance. ACK_TIME donne la durée d'attente maximale, après envoi d'un message, avant réception d'un acquittement.

À partir de la ligne 78, on trouve quelques définitions pour le capteur DHT, les broches et variables utilisées. Les commentaires dans le code source indiquent les fonctions de ces variables. À partir de la ligne 135 suit l'initialisation du programme (*setup()*), dans laquelle est définie la broche de sortie et initialisé le module sans fil. Voir l'ordinogramme de la **figure 1**.

On commence (ligne 167) par voir si le module RFM69 a reçu un paquet radio. Si c'est le cas, il y a appel de la fonction *parseCmd*. En fonction du *deviceID* et de l'instruction de lecture/écriture dans le bloc de données reçu, le drapeau d'envoi adéquat est levé. Après cela, on vérifie s'il y a eu pression du bouton et expiration de la temporisation. Il y a ensuite incrémentation du compteur de temps de fonctionnement. Si l'intervalle de transmission périodique est écoulé, il y a positionnement des drapeaux d'envoi de ces paramètres.

Enfin, à la ligne 235, il y a appel de la fonction *sendMsg*. Selon le drapeau d'envoi levé, *sendMsg* prépare les données requises pour la transmission radio, réinitialise le drapeau d'envoi et envoie le paquet après un appel de la fonction *txRadio()*.

En cas de demande de la température ou de l'humidité (appareils 48 et 49), le module DHT est interrogé (lignes 478 et 485).

Passerelle (*gateway*)

La passerelle remplit les fonctions suivantes :

- Réception et décryptage d'un message MQTT, création et envoi du paquet radio.
- Réception et décryptage d'un message radio, création et publication du message MQTT.

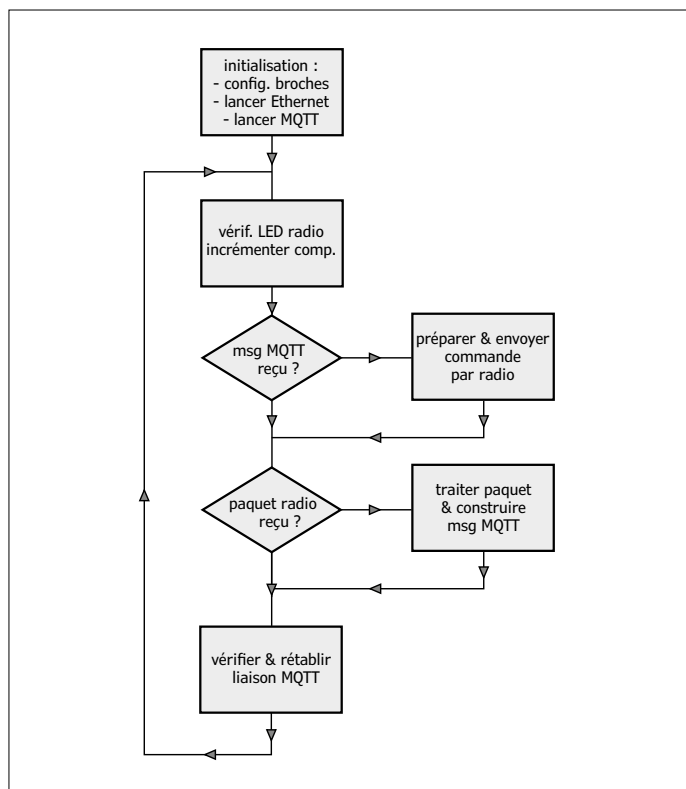


Figure 2. Ordinoigramme de la passerelle.

- Réception, décryptage et réponse aux demandes sur la passerelle elle-même, telles que temps de fonctionnement, version, etc.

La passerelle est reliée à une connexion Ethernet fixe et reçoit ses informations de réseau par DHCP. La passerelle comporte deux LED. La LED radio indique l'activité de la liaison sans fil. La LED MQTT s'allume lorsqu'il existe une liaison avec Mosquitto.

Comme le programme de la passerelle occupe beaucoup de mémoire, la fonction de débogage est subdivisée en deux parties : l'une pour déboguer la liaison sans fil et l'autre pour déboguer la connexion au courtier MQTT. Il ne peut y avoir qu'une seule partie active, sinon le programme ne se compile pas. La description du programme repose sur la dernière version (2.4), disponible en [6].

Le croquis de la passerelle utilise les bibliothèques suivantes :

- *RFM69*
- *SPI*
- *Ethernet*
- *PubSubClient* [7]

Lors de la compilation du croquis de la passerelle, nous modifions quelque peu des paramètres par défaut :

- Dans la passerelle, le bus SPI est utilisé par deux appareils. Les deux appareils utilisent « 10 » comme adresse d'appareil par défaut. Nous choisissons le code d'appareil 8 pour le module RFM et 10 pour le module Ethernet (voir configuration à la ligne 80 du code de la passerelle).

- Le module RFM utilise des interruptions pour indiquer la réception d'un paquet radio. Tant que le pilote Ethernet (*W5100.h*) est actif sur le bus SPI, le traitement de ces interruptions se fait incorrectement d'où blocage de la passerelle. Nous utilisons la solution proposée par Martin Harazinov et adaptons le fichier *w5100.h* pour résoudre ce problème (cf. [8]). Dans le fichier *w5100.h* (à trouver dans `../Documents/Arduino/libraries/Ethernet/src/utility`), on ajoute aux lignes suivantes (à partir de la ligne 342) des déclarations `cli()` et `sei()`.

```

#else
inline static void initSS() { DDRB |= _BV(2); };
inline static void setSS() { cli(); PORTB &= ~_BV(2); };
inline static void resetSS() { PORTB |= _BV(2); sei(); };
#endif

```

Cela permet de bloquer les interruptions tant qu'il y a du trafic Ethernet sur le bus SPI.

Le croquis de la passerelle commence par un bloc de déclaration très similaire à celui du nœud terminal. On y définit tous les paramètres de la liaison sans fil. C'est ensuite le tour des paramètres suivants pour Ethernet :

- *Ip*, pour l'adresse IP de la passerelle. C'est un repli (*fall-back*) ; la passerelle utilise DHCP et reçoit donc une adresse dynamiquement.
- *Mac*, pour l'adresse MAC de la connexion Ethernet. Remarque : celle-ci doit être unique dans un sous-réseau IP, donc attention s'il y a plusieurs Arduino dans le réseau.
- *Mqtt_server*, l'adresse du serveur Mosquitto (nous en reparlerons plus loin).

La **figure 2** montre l'ordinoigramme du code de la passerelle. L'initialisation commence à la ligne 135. On y configure les broches de sortie des deux LED, lance la connexion Ethernet et démarre la connexion avec le courtier MQTT.

Le client MQTT est initialisé à la ligne 129. On y demande un abonnement, défini dans la fonction `mqtt_subs` (ligne 382). L'objet de l'abonnement est `home/rfm_gw/sb/#` (ligne 121) pour recevoir tous les messages *Southbound*. Dès qu'un message MQTT arrive, la variable `mqttToSend` est mise à `TRUE`. Dans le programme principal, on commence par voir si la LED du sans fil doit être éteinte puis on augmente le temps de fonctionnement de la passerelle.

À la ligne 223, on vérifie avec la variable `mqttToSend`, s'il faut envoyer des messages sur le réseau sans fil, fonction dont se charge `sendMsg` (ligne 248). Le cas échéant, l'envoi de la transmission se répète, au maximum cinq fois, toutes les demi-secondes, jusqu'à ce que la réception du paquet soit confirmée par le nœud terminal.

Ensuite, à la ligne 225, on contrôle si l'on a reçu un paquet d'un des nœuds terminaux. Si c'est le cas, on appelle la fonction `processPacket` (ligne 288). Un message MQTT de la passerelle se réfère toujours à *Northbound*. Il est constitué de `nodeID` et `deviceID`, et stocké dans la chaîne `buff_topic` à la

ligne 303. En fonction du `deviceId`, on détermine quelle donnée du paquet radio est valide (*integer*, *float*, *string* ou état), et quel est le formatage requis pour le message MQTT. Pour finir, le message MQTT se trouve dans la chaîne `buff_mess`. Enfin on vérifie à la ligne 227, si la liaison MQTT est toujours active. Si ce n'était pas le cas, on tente à la ligne 230 de la redémarrer, toutes les deux secondes. La LED MQTT est allumée (ligne 235) lors de l'établissement de la liaison.

OpenHAB

On voit en **figure 3** l'architecture d'OpenHAB qui repose sur un bus d'événement(s). Par le biais de ce bus, il y a échange d'événements entre les interfaces d'entrée et de sortie et le reste d'OpenHAB. Cette construction permet d'utiliser côte à côte plusieurs interfaces d'E/S (« *bindings* »). Il est possible ainsi de créer des « *bindings* » spécifiques pour différents protocoles. À l'heure actuelle il existe des « *bindings* » pour des compteurs intelligents, les systèmes audio SONOS, les lampes Hue de Philips, les thermostats Nest ou encore les voitures Tesla. La page [9] en donne un aperçu.

Installation

Nous utilisons un Raspberry Pi 3 avec une carte SD de 16 Go ainsi que la version 2 d'OpenHAB. La façon la plus simple d'installer OpenHAB est d'utiliser OpenHABian. C'est une image avec un système Linux minimal combiné à des scripts d'installation. On pourra télécharger cette image depuis [10]. Avec l'outil adéquat (WinDiskImager [11] par ex.), nous écrivons cette image sur une carte SD. Cette dernière est enfichée dans le Raspberry Pi et après le « *boot* », l'installation démarre. Cela peut durer un certain temps.

Au cours de cette installation, il y a également installation de *Samba* qui permet d'accéder aux dossiers de configuration d'OpenHAB par le réseau, ce qui facilite les changements de configuration.

Lorsque l'installation est terminée, nous démarrons Mosquitto avec l'outil de configuration :

```
pi@openHABianPi: sudo openhabian-config
```

... et voyons apparaître, si tout se passe bien, la fenêtre de la **figure 4**. Le choix 22 installe Mosquitto. Il faut aussi que le Raspberry Pi reçoive toujours la même adresse IP. Nous y veillons en modifiant le fichier `/etc/network/interfaces`. Il faut modifier `address` et `gateway` en fonction de la configuration souhaitée, par ex. :

```
auto eth0
iface eth0 inet static
    address 192.168.2.7
    netmask 255.255.255.0
    gateway 192.168.2.254
```

Un redémarrage active le nouveau paramétrage.

Configuration

Pour qu'OpenHAB fonctionne, il faut configurer correctement la liaison MQTT. Le fichier à modifier se trouve sur le RPi dans `/etc/openhab2/services/mqtt.cfg`. L'URL du courtier (*broker*) est le seul paramètre à modifier comme suit :

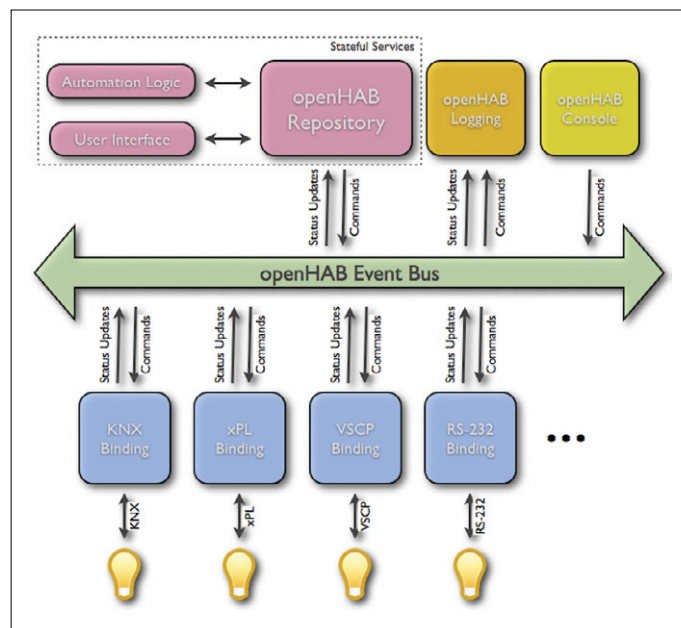


Figure 3. Architecture d'OpenHAB (source : OpenHAB.org).

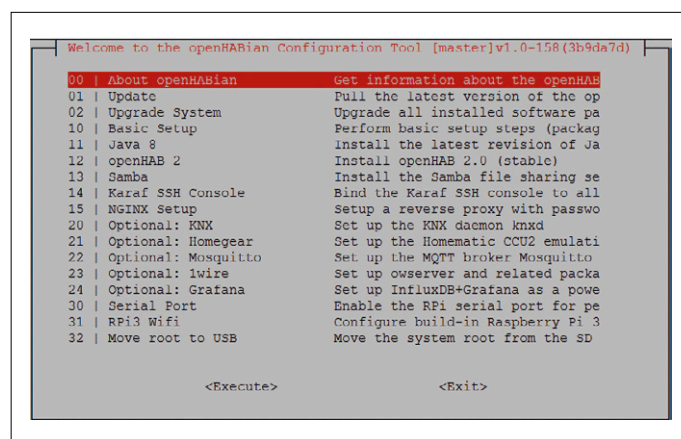
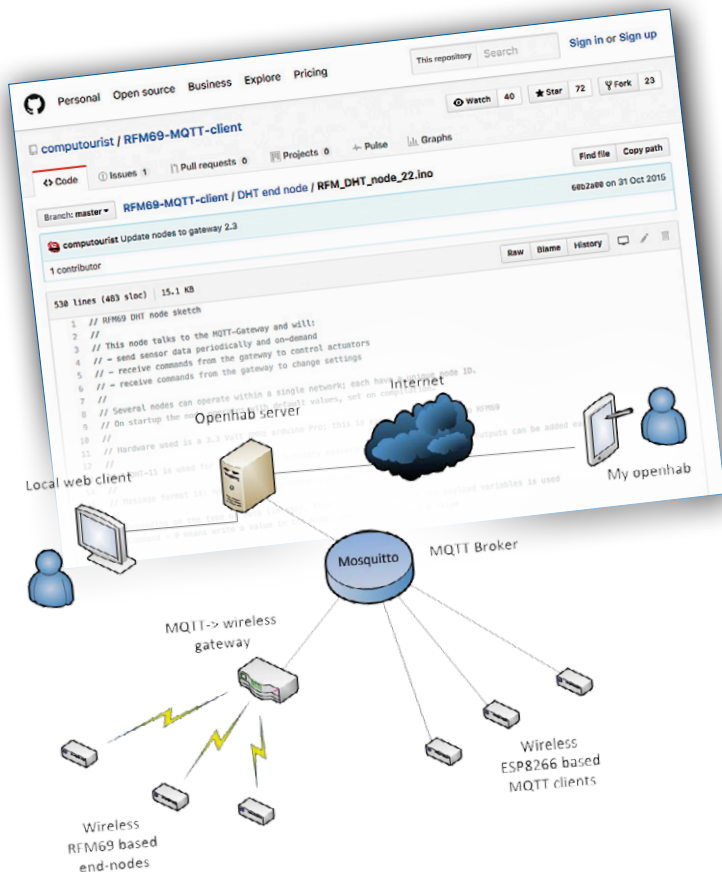


Figure 4. Configuration d'OpenHABian.

```
# Define your MQTT broker connections here for use
# in the MQTT Binding or MQTT
# Persistence bundles. Replace <broker> with an ID
# you choose.
# URL to the MQTT broker, e.g. tcp://localhost:1883
# or ssl://localhost:8883
mosquitto.url=tcp://localhost:1883
```

C'est un renvoi vers l'hôte local (*localhost*) pour le serveur Mosquitto. Veillez à ce qu'ici `mosquitto` soit le nom (ID) du courtier. Dans la configuration d'objet (*item*) (cf. le texte plus loin), nous renvoyons à cet ID (entifiant) pour établir la communication. Le port 1883 est utilisé par défaut pour le trafic MQTT. OpenHAB utilise trois fichiers de configuration pour décrire un système :

- *Items* dans le dossier `/etc/openhab2/items`, qui contient les définitions des paramètres utilisés et leur lien vers les dossiers *item*.



- *Sitemap* dans le dossier `/etc/openhab2/sitemaps`, qui décrit la mise en page de l'interface utilisateur.
- *Rules* dans le dossier `/etc/openhab2/rules`, qui indique les actions à effectuer (*scripting*).

À titre d'exemple, nous configurons un nœud DHT terminal avec le *nodeID* 2. Il doit envoyer périodiquement la température, l'humidité et l'état de sortie. Dans le dossier *item* (`/etc/openhab2/items`), nous créons un fichier *home.items* avec le contenu du **listage 1**. Le Wiki OpenHAB [12] explique la structure de la configuration d'un élément (*item*). Ensuite, dans le dossier `/etc/openhab2/sitemaps`, nous créons le fichier *home.sitemap* avec le contenu suivant :

```
sitemap home label="MyHome"

Frame label="Controls" {
    Switch item=OUT2 label="Myhome lamp"
    icon="light" }

Frame label="Sensoren" {
```

```
Text item=TEMP2
Text item=HUM2
Text item=RSSI2 }
```

La première ligne donne le nom de cette configuration (*home*), pour que lors de l'accès à OpenHAB par un navigateur ou une appli, il y ait chargement de la cartographie (*sitemap*) correcte. Ensuite on définit cadre pour superviser la sortie du nœud 2 et un autre pour l'affichage des température, humidité et intensité de champ. Enfin, nous créons un fichier *home.rules* dans `/etc/openhab2/rules` comportant les lignes suivantes :

```
//
// refresh rules
//

rule "refresh RSSI2" // periodically refresh
value for signal strength
when
    Time cron "0 0/1 * * * ?"
then
    sendCommand(getRSSI2, "READ")
end
```

Grâce à la commande *cron*, on envoie, une fois par minute, le message MQTT « READ » vers l'appareil 2 sur le nœud 2 aux fins d'obtenir la force de champ.

Il est temps de procéder aux premiers essais. Ouvrez un navigateur et allez à :

<http://192.168.xx.xx:8080/basicui/app?sitemap=home>

Il faut bien entendu remplacer l'adresse IP par l'adresse correcte. La fenêtre de la **figure 5** montre le résultat de cette instruction. Il se peut que toutes les valeurs ne soient pas disponibles (on a dans ce cas-là affichage d'un « - »), mais au bout d'un certain temps, après une première actualisation par le nœud, cela devrait être résolu.

Débogage

Et, si tout ne se passe pas comme prévu ?

MQTT fonctionne-t-il ?

Si la connexion MQTT entre la passerelle et le courtier fonctionne, la LED MQTT s'allume, ce qui indique que les messages transitent de la passerelle vers Mosquitto. Étape suivante : installer un client de test MQTT (MQTT.fx [13] ou MQTTLens [14], voir Chrome Web Store). MQTT.FX a un onglet *broker status*

Listage 1. Configuration d'un *item* OpenHAB pour un nœud terminal température/humidité de l'air.

```
Number RSSI2 «RSSI [%d dBm]» {mqtt=><[mosquitto:home/rfm_gw/nb/node02/dev02:state:default]}
String getRSSI2 «get RSSI» {mqtt=>>[mosquitto:home/rfm_gw/sb/node02/dev02:command:*.default]}
Number TEMP2    "Temperature [%1f °C]" {mqtt="<[mosquitto:home/rfm_gw/nb/node02/dev48:state:default]}"
Number HUM2     "Humidity [%1f %]" {mqtt="<[mosquitto:home/rfm_gw/nb/node02/dev49:state:default]}"
Switch OUT2 "attic node" <PushBtn> {mqtt=">[mosquitto:home/rfm_gw/sb/node02/dev16:command:ON:ON],>[mosquitto:home/rfm_gw/sb/node02/dev16:command:OFF:OFF]}"
```

pour demander des informations sur Mosquitto. Abonnez-vous au sujet `home/rfm_gw/#` et des messages devraient apparaître ; si tout est OK, il s'agit de messages provenant du nœud terminal. Sinon, il est temps de compiler la passerelle ou le nœud avec l'option `DEBUG` et de connecter un PC au port sériel. Démarrez PUTTY [15], paramétrez le port COM correct et réglez la vitesse à 115200 baud. La sortie du nœud/passerelle indiquera l'existence (ou non) d'une connexion et les valeurs en cours de transfert. L'envoi de commandes client MQTT au nœud (par ex. `READ` à `/home/rfm_gw/node02/dev03`) montrera où bloque le flux de données.

OpenHAB fonctionne-t-il ?

Si la connexion MQTT fonctionne, l'étape logique suivante consiste à voir du côté d'OpenHAB. L'examen des fichiers journaux (*log*) est instructif. L'instruction

```
> tail -f /var/log/openhab2/events.log
```

permet d'afficher les dix dernières lignes du journal d'événements. On devrait y voir des mises à jour entrantes venant du nœud ou de l'interface utilisateur. Petit coup d'œil aussi à `/var/log/openhab2/openhab.log`, fichier où sont rangés les événements système d'OpenHAB.

Et après...

Votre première extension sera sans doute un accès à l'internet ; `myopenhab1` le permet. OpenHAB a une appli qui communique avec votre propre système OpenHAB par l'internet via une passerelle sécurisée (voir [1]).

OpenHAB permet également de créer des graphiques des variations dans le temps des valeurs de mesure ; cela requiert d'activer une base de données locale et de spécifier les valeurs de mesure à mémoriser. On peut travailler localement, mais

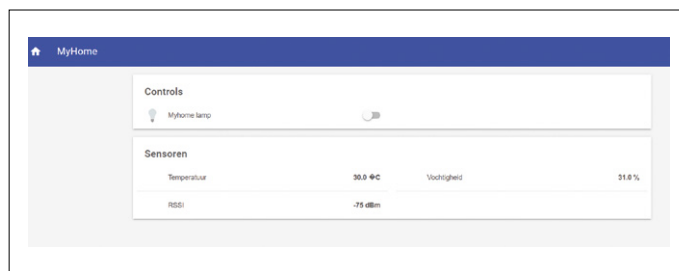


Figure 5. Fenêtre de navigateur.

aussi accéder à des bases de données externes (voir [16]). Le nœud décrit mesure la température et l'humidité et peut commuter une sortie numérique. L'utilisation d'un circuit électronique différent pour le nœud, et la modification du logiciel ouvrent d'autres perspectives. On trouvera sur Github [3] des solutions pour un LCD et un lecteur RFID [17]. On y décrit également un nœud qui commande depuis OpenHAB des interrupteurs du système de domotique *KlikAanKlikUit*.

Parallèlement à la solution décrite ici, une variante à base d'ESP8266 a été développée. Ce nœud établit une liaison par Wi-Fi avec le courtier MQTT et OpenHAB. On trouvera en [8] la description du projet.

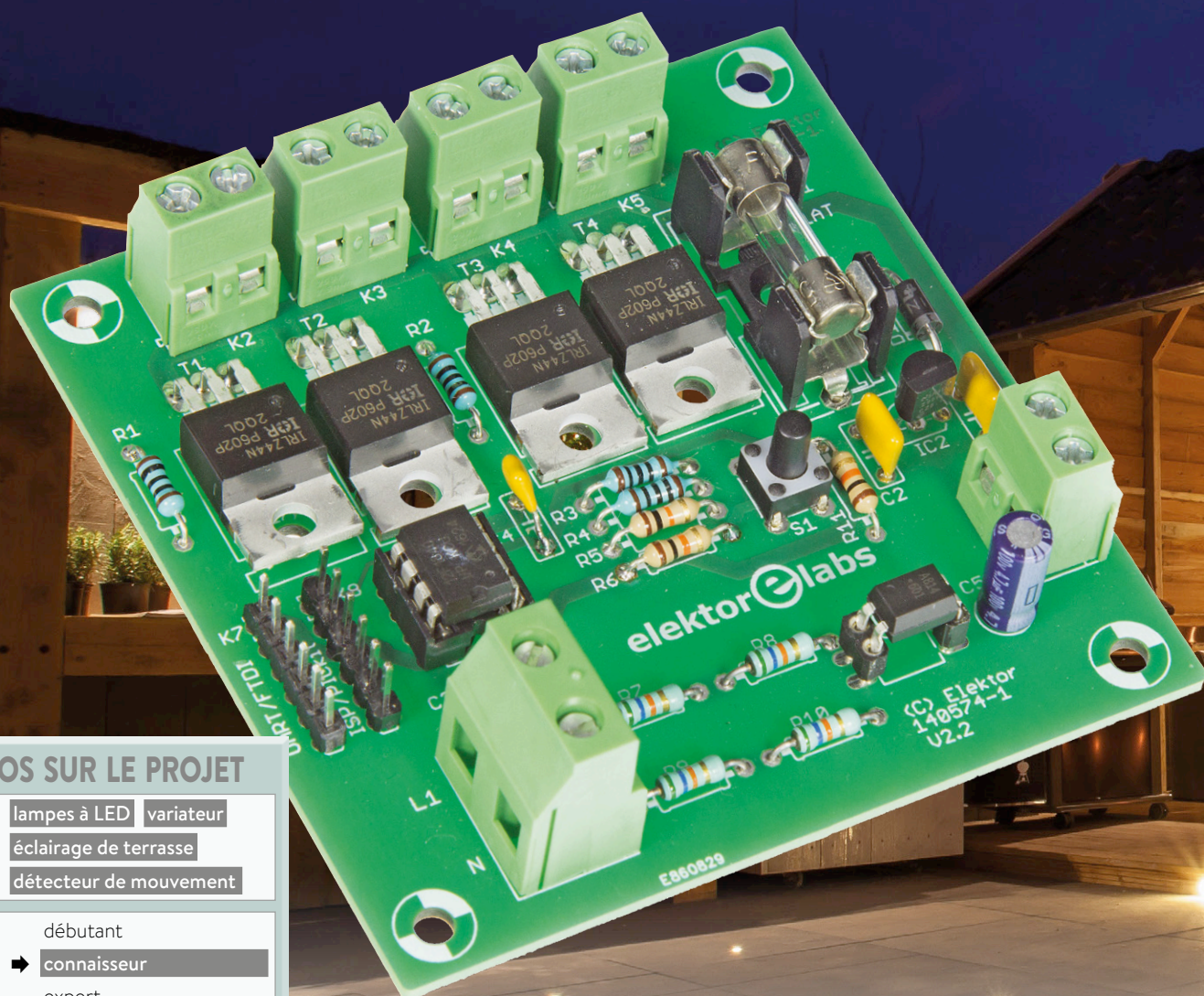
Jetez donc un coup d'œil au forum [19] où différents utilisateurs présentent des solutions pour des détecteurs PIR, des volets roulants, des détecteurs de niveau et bien d'autres applications encore. ◀

(160318 – version française : Guy Raedersdorf)

Liens

- [1] www.myopenhab.org
- [2] www.elektormagazine.fr/150085
- [3] <https://github.com/computourist/RFM69-MQTT-client>
- [4] https://github.com/computourist/RFM69-MQTT-client/blob/master/DHT%20end%20node/RFM_DHT_node_22.ino
- [5] <https://github.com/LowPowerLab>
- [6] https://github.com/computourist/RFM69-MQTT-client/blob/master/Gateway_2.4/RFM_MQTT_GW_24.ino
- [8] <http://harizanov.com/2012/04/rfm12b-and-arduino-ethernet-with-wiznet5100-chip>
- [7] <https://github.com/knolleary/pubsubclient>
- [9] <http://docs.openhab.org/addons/bindings.html>
- [10] <http://docs.openhab.org/installation/openhabian.html>
- [11] <https://sourceforge.net/projects/win32diskimager>
- [12] <https://github.com/openhab/openhab1-addons/wiki/Explanation-of-items>
- [13] <http://mqttfx.org>
- [14] <https://github.com/sandro-k/MQTTLensChromeApp>
- [15] www.putty.org
- [16] <http://docs.openhab.org/configuration/persistence.html>
- [17] <https://github.com/computourist/RFM69-MQTT-client>
- [18] <https://github.com/computourist/ESP8266-MQTT-client>
- [19] <http://homeautomation.proboards.com/board/2/openhab-rfm69-based-arduino>

éclairage extérieur variable



INFOS SUR LE PROJET



lampes à LED variateur
éclairage de terrasse
détecteur de mouvement



débutant
→ connaisseur
expert



env. 3 h



Câble FTDI USB/série, version
de démo de PIC MikroBasic,
Visual Basic 2015 (tous optionnels)



env. 25 €

Andreas Meyer (Allemagne)

Par un beau jour d'été, notre lecteur Andreas Meyer a installé un éclairage de terrasse, composé de quatre spots à LED de 3 W. Mais le résultat obtenu était vraiment trop puissant. Comme personne n'aime passer une soirée en terrasse avec des lunettes de soleil, le nouvel éclairage n'a pas beaucoup servi. Il fallait installer un variateur qui affaiblisse en douceur la luminosité afin d'améliorer le confort.

réglage de la luminosité de lampes à LED de 12 V



La « règle d'airain de l'éclairage » suivante s'applique, et pas seulement aux électroniciens : *là où une lampe éclaire trop, il faut un variateur, et vite (sic) !* Hélas, les quatre spots à LED d'Andreas n'étaient pas compatibles avec un variateur. Ce n'est pas toujours le cas, certains modèles alimentés en 12 V fonctionnent correctement avec un variateur (mais ce n'est quasiment jamais le cas pour les lampes à LED alimentées par le réseau).

Un simple test avec un signal MLI issu d'un microcontrôleur et un MOSFET de puissance en amplificateur de courant montre que les spots sont faciles à commander, bien que rien n'ait été écrit sur le sujet. Le principe de fonctionnement est différent entre les modèles alimentés en 220 V et ceux en 12 V parce que les lampes à LED prévues pour le réseau comportent en général un circuit d'alimentation qui transforme le cou-

Caractéristiques

- Possibilité de connecter quatre lampes à LED
- Deux niveaux de variation pour des lampes à LED basse tension
- Transition douce entre les niveaux d'éclairage
- Commande des niveaux avec un détecteur de mouvement



Figure 1. Montage de test « vite fait » de l'auteur : programmeur avec le microcontrôleur et quatre MOSFET sur une plaque d'essai.



Une lumière parfaite pour les douces soirées d'été !

rant alternatif 230 V en courant continu adapté aux LED.

Naturellement ce bloc d'alimentation

doit être construit spécialement pour la variation et doit réduire le courant de sortie proportionnellement à la tension

d'entrée moyenne. Par ailleurs ce bloc essaie de maintenir le courant constant aussi longtemps que possible ; lorsque rien ne va plus, la lampe s'assombrit, scintille ou libère une odeur suspecte « du côté de l'électronique ». A contrario, les lampes de 12 V sont très souvent de conception plus simple : généralement il s'agit de trois LED blanches branchées en série avec une résistance adéquate, de telle façon que sous une tension continue de 12 V circule un courant constant raisonnable. Comme il n'y a pas de régulation électronique, il n'y a pas de problème pour faire varier l'éclairage par une simple variation du rapport impulsion/pause. Voilà pour la situation de départ et la théorie.

Cahier des charges du circuit

Le circuit final comporte au moins une puce puisque le montage fait appel à un microcontrôleur. Mais quelles sont les autres exigences ?

Sur la base de l'expérience acquise avec ce nouvel éclairage de la terrasse (bien que peu utilisé), il est évident que la luminosité doit être très nettement inférieure à celle observée à « pleine puissance ». En outre, l'éclairage doit être commandé par un commutateur photosensible (modèle du commerce) et/ou un interrupteur d'alimentation ordi-

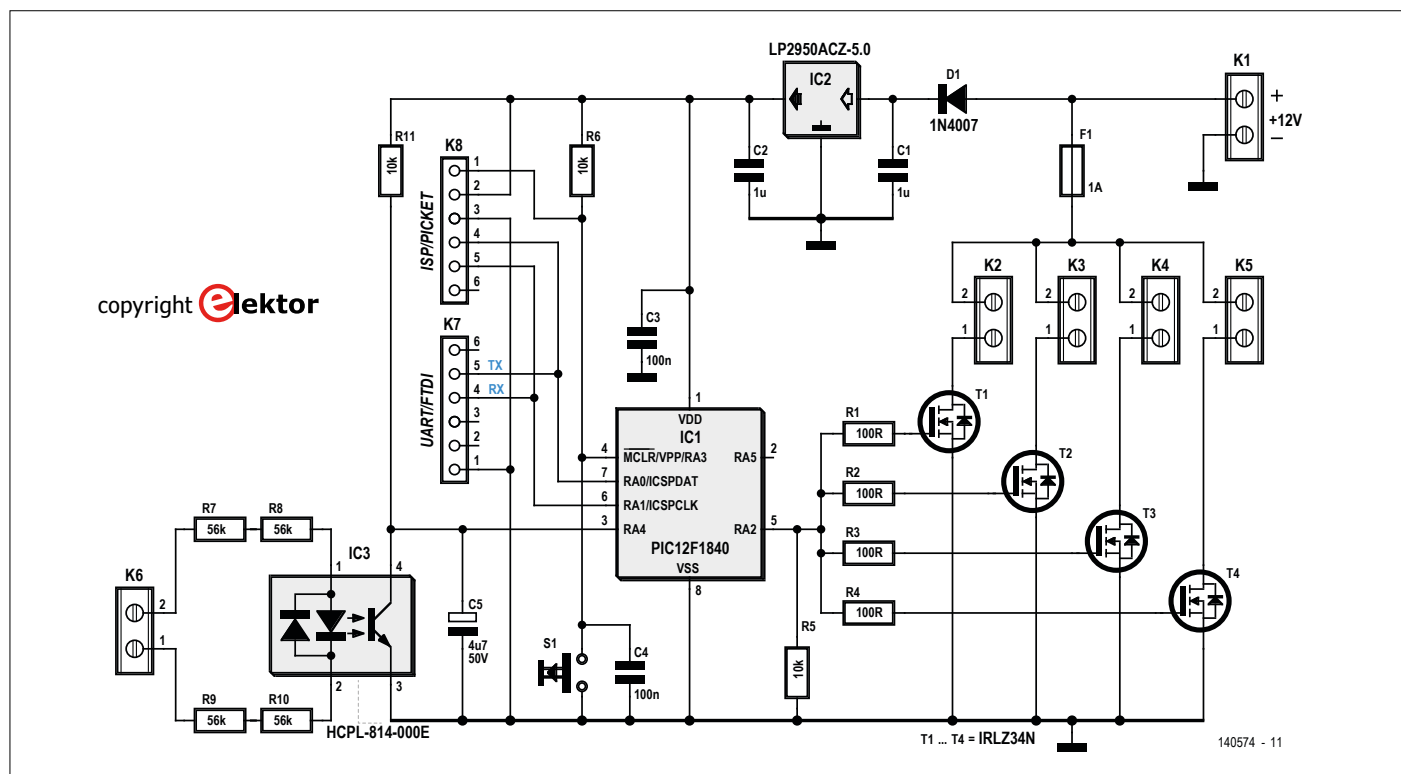


Figure 2. Le schéma de l'éclairage extérieur variable se compose d'un microcontrôleur, de quatre MOSFET de puissance, d'une entrée 230 V pour un détecteur de mouvement et d'une alimentation régulée de 5 V.

naire, afin de l'éteindre en journée et de l'allumer le soir si nécessaire. De plus, l'éclairage est plus lumineux quand quelqu'un se déplace afin que personne ne trébuche par exemple lors d'un ravitaillement en bière, vin ou eau ! Enfin, et non des moindres, pour des raisons de confort, les niveaux de luminosité ne doivent pas changer de façon brusque, mais de façon progressive.

La variation même ne pose pas de problème avec une modulation de largeur d'impulsion (MLI) effectuée par un μC . Si besoin, achetez un interrupteur crépusculaire et montez-le en série avec l'interrupteur. Pour la détection de mouvement, on peut ajouter un détecteur de mouvement approprié (voir en grande surface de bricolage). Le circuit électronique n'a donc pas besoin d'un capteur de lumière, mais d'une entrée 230 V pour le détecteur de mouvement, de sorte que l'éclairage puisse basculer entre les deux niveaux de luminosité souhaités. L'auteur a déterminé comme convenables des rapports cycliques MLI de 10% pour l'éclairage d'ambiance normal et de 50% si un mouvement est détecté. Le circuit électronique nécessite quatre sorties MLI, ainsi qu'un circuit d'alimentation qui peut fournir 12 V aux lampes à LED. Pour adapter les valeurs de luminosité, comme on ne veut pas à chaque fois changer les constantes dans le code source, le compiler et flasher le μC , un port série est ajouté afin de pouvoir régler les valeurs de MLI de l'extérieur via un PC. Cela fait partie des exigences essentielles pour ce montage.

Passons au schéma

Andreas a d'abord dessiné le schéma autour d'un μC qu'il avait sous la main, de type PIC16F628, et il a utilisé un optocoupleur pas très sensible comme entrée de pilotage du 230 V pour la détection de mouvement. Suite aux commentaires sur le site du labo d'Elektor [1] et aux travaux complémentaires du labo, le μC de la version finale est le PIC12F1840, bien plus que suffisant, plus petit et moins cher (cf. **fig. 2**).

Le schéma est rapide à expliquer : en haut à droite se trouve la partie alimentation, capable de délivrer au moins 1 A sous 12 V stabilisés. Un ampère suffit pour les quatre lampes de 3 W. Pour des lampes plus puissantes, il faut adapter le circuit d'alimentation et le fusible F1. Les quatre pilotes de lampe à base de MOSFET de puissance fonctionnent



LISTE DES COMPOSANTS

Résistances

(toutes 5%, 1/4 W, 250 V)
R1 à R4 = 100 Ω
R5, R6, R11 = 10 k Ω
R7 à R10 = 56 k Ω *

Condensateurs

C1, C2 = 1 μF / 50 V, film,
pas de 5,08 mm
C3, C4 = 100 nF / 50 V,
film, pas de 5,08 mm
C5 = 4,7 μF / 35 V,
électrolytique,
vertical, 5 x 11 mm,
pas de 2 mm

Semi-conducteurs

D1 = 1N4007
T1 à T4 = IRL34NPBF
IC1 = PIC12F1840-I/P,
programmé*
IC2 = LP2950ACZ-5.0
IC3 = HCPL-814-000E,
optocoupleur

Divers

5 x connecteur à vis à
deux pôles pour circuit
imprimé, pas de 5,08 mm
1 x connecteur à vis à deux pôles pour circuit
imprimé, pas de 7,62 mm
2 x barrette à 6 contacts, une rangée,
pas de 2,54 mm
1 x porte fusible pour circuit imprimé (par ex.

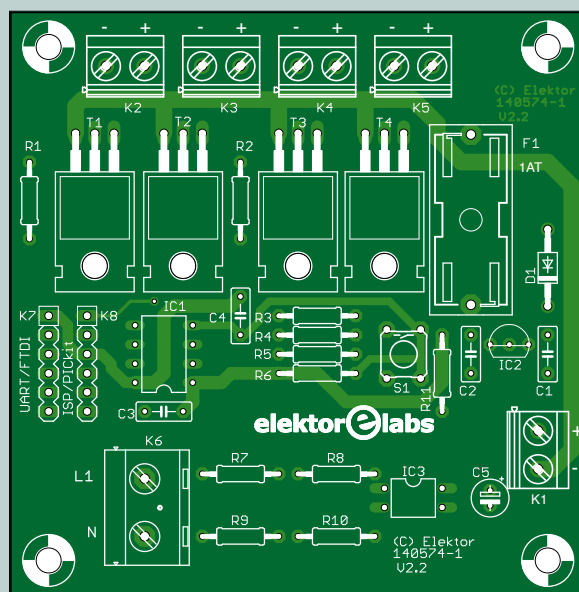


Figure 3. Le circuit imprimé de l'éclairage extérieur variable a une longueur d'environ 7,7 cm et prend place dans une boîte de dérivation suffisamment grande.

Schurter 0031.8201)

1 x fusible 1 A, 20 x 5 mm*

1 x interrupteur pour circuit imprimé,
contact à fermeture, 6 x 6 mm

1 x circuit imprimé 140574-1*

1 x support de CI à 8 broches, DIP, pour IC1

* voir texte

sans refroidissement et peuvent débiter plus de courant. Le μC est alimenté par IC2, un régulateur de tension 5 V de type *micropower* (micropuissance, quelques mA seulement). Lorsque le détecteur de mouvement se met en fonction, le 230 V est appliqué sur K6 et atteint l'optocoupleur au travers de résistances série. Ces quatre résistances de 56 k Ω servent à répartir la puissance dissipée (env. 240 mW au total), mais aussi à assurer une meilleure rigidité diélectrique. À la sortie d'IC3, le condensateur C5 lisse la tension, ce qui évite ainsi les pics à 100 Hz. L'UART interne du μC est raccordé à K7 auquel on peut connecter directement un câble FTDI USB/série, mais aussi à K8, prévu pour téléverser du code dans le μC . Enfin, S1 est le bouton de remise à zéro. La fréquence MLI est réglée dans le microprogramme à 150 Hz. Cela suffit presque toujours pour une variation sans scintillement et permet également une

faible fréquence d'horloge interne du μC de seulement 200 kHz pour une petite consommation d'énergie. Si la fréquence MLI vous semble trop faible, vous pouvez choisir une fréquence d'horloge supérieure dans le microprogramme, mais gardez à l'esprit que les constantes de temps pour la transition entre les deux niveaux de luminosité sont également fonction de cette fréquence d'horloge.

Circuit imprimé, programme et Cie

Le circuit imprimé conçu pour ce montage (**fig. 3**) de 7,7 cm de côté prend place dans une boîte de dérivation. Comme vous pouvez le voir, il n'y a pas un seul CMS. Grâce aux composants traversants, la réalisation devrait donc être facile. En plus du circuit imprimé, Elektor propose le microcontrôleur préprogrammé [2]. Dans le micrologiciel, les deux valeurs de départ pour la MLI sont réglées sur 10% et 50%, mais il est pos-

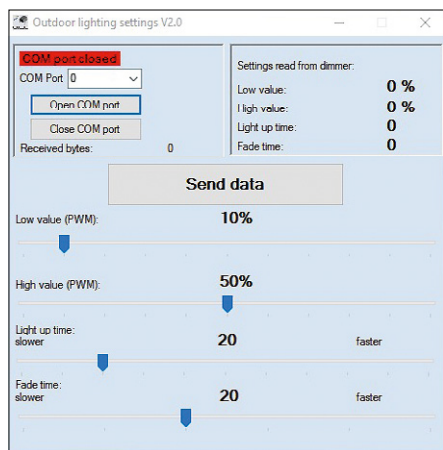


Figure 4. Fenêtre de l'application sur PC avec laquelle on peut régler les deux niveaux de luminosité et les vitesses de transition entre ces niveaux.

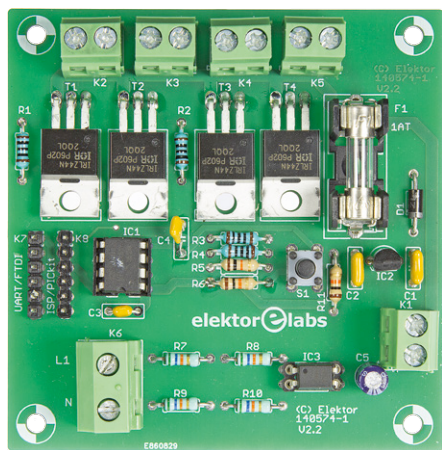


Figure 5. Voilà à quoi ressemble le prototype fabriqué par le labo d'Elektor.

sible de les modifier par la suite dans le logiciel sur PC. Pour IC1, l'utilisation d'un support est recommandée (c'est prévu dans la liste de composants). Comme déjà mentionné, les quatre transistors MOSFET n'ont pas besoin d'être refroidis. Avec une résistance typique drain-source de 35 mΩ, si chaque exemplaire est parcouru par un courant de 1 A, la puissance dissipée est égale à 35 mW ; 1 A est suffisant pour 4×12 W, cela fait déjà beaucoup de lumière avec des LED. Il faut installer autant de MOSFET que l'on a de canaux de sortie. Si vous ne connectez que deux lampes, vous pouvez supprimer deux MOSFET y compris les résistances en série et les connecteurs (par ex. T3 et T4 conjointement avec R3 et R4, ainsi que K4 et K5). Il va sans dire que l'on doit adapter F1 et le bloc d'alimentation si la puissance totale est supérieure à 12 W.

À propos de l'auteur

Andreas Meyer est électronicien industriel et se passionne pour la programmation des microcontrôleurs et la conception de schémas et circuits imprimés. Actuellement il s'occupe surtout de la programmation d'applications Windows qui échangent des données avec des microcontrôleurs et les supervisent.

Le micrologiciel du PIC12F1840 a été développé avec l'EDI mikroBasic de MikroElektronika. Comme il est très court à cause du nombre limité de fonctions, il peut être modifié avec la version de démo gratuite de l'EDI. En outre, Andreas a développé avec Visual Studio une petite application pour PC sous Windows (**fig. 4**), avec laquelle les deux valeurs de MLI correspondant aux deux luminosités peuvent être transmises au µC via un câble convertisseur FTDI USB/série. Attention si vous modifiez le code, les nouvelles valeurs ne seront prises en compte qu'après une remise à zéro du µC ! Le code source complet, le programme complet pour le PC et le fichier hex du logiciel sont bien sûr disponibles sur le site d'Elektor [2]. La **figure 5** montre les prototypes du labo d'Elektor.

Et encore...

Il faut définitivement se retirer l'idée que ce montage a quelque chose de commun avec l'un de ces « transformateur électronique 12V » en plastique blanc bon marché comme ceux proposés pour les lampes halogènes. Ce sont généralement des alimentations à découpage simples, qui offrent 12 V au secondaire, mais sortent une tension alternative de quelques dizaines de kHz. Ce type d'alimentation n'est valable que pour les ampoules à incandescence et n'est pas adaptée à des lampes à LED (préférez le circuit électronique présenté ici). En effet les MOSFET contiennent tous une diode parasite tête-bêche, comme vous pouvez le voir sur le symbole de composant. Avec une alimentation alternative, les diodes intégrées aux MOSFET conduisent et appliquent une tension inverse élevée aux bornes des LED (> 12 V), cela peut réduire la durée de vie des LED.

Le deuxième point à observer concerne l'installation elle-même : sur les quatre bornes K2 à K5 reliées aux lampes se trouve une tension rectangulaire, dont la fréquence correspond à la MLI. Bien que 150 Hz ne semble pas énorme, la tension rectangulaire a tout de même des flancs relativement raides et cela engendre une forte proportion d'harmoni-

ques qui entrent bien dans la bande HF classique. En d'autres termes : si les lignes vers les lampes à LED ne sont pas vraiment courtes, les interférences HF seront rayonnées dans l'éther et le voisinage. Les installations domestiques habituelles avec des câbles normaux ou des fils rigides ne sont pas faites pour la HF, et ne sont pas non plus « étanches à la HF ». Dans de tels cas, une paire torsadée est bien meilleure ; il est même préférable d'utiliser des câbles blindés pour les lignes de plus de quelques mètres de longueur. Bien que les paires torsadées ou le câble coaxial RG-58/U n'acceptent normalement que 2 A et ne sont peut-être pas prévus à cet effet, un tel câblage en basse tension limite les rayonnements parasites.

Si vous utilisez l'éclairage extérieur réglable comme Andreas en combinaison avec un détecteur de mouvement (avec deux niveaux de variation), vous adapterez probablement à plusieurs reprises les deux valeurs de luminosité en fonction de l'installation des lampes. Il est donc intéressant de ne pas installer le circuit de manière définitive tant que vous n'avez pas trouvé le parfait compromis à la lumière du soir – il faut au moins faciliter l'optimisation par un accès aisé à K7 pour la connexion d'un PC ou d'un ordinateur portable (ou K8, si vous voulez mettre à jour le micrologiciel avec les valeurs modifiées). Et alors tout sera prêt pour les douces soirées d'été...

(140574 – version française : Denis Langrand)

Liens

[1] <http://goo.gl/izezWr>

[2] www.elektormagazine.fr/140574

DANS L'ÉCHOPPE

→ 140574-1
Circuit imprimé

→ 140574-41
Contrôleur préprogrammé

simulateur de présence

la crainte des monte-en-l'air

Theo de Wijs (Pays-Bas)

Les membres de la guilde des cambrioleurs confondent souvent le « mien » et le « tien », peuvent difficilement se targuer d'un soutien inconditionnel du public, arrivent toujours à l'improviste et ne craignent réellement qu'une chose : la présence des « locataires » légitimes...



La meilleure prévention anti-intrusion est en fait de ne jamais quitter la maison. Pour des raisons évidentes, une présence permanente est impossible, et une absence prolongée en particulier (lors de vacances par ex.) augmente le risque de visiteurs indésirables. Nous n'avons pas le choix ; il faut entrer en action. Deux options : la première est d'acheter un système d'alarme – mais cela peut valoir son pesant de « cacahouètes ». Les magasins *discount* proposent parfois des systèmes à bas prix, mais lors de tests comparatifs ils ne font pas le poids et ne constituent donc pas vraiment une solution. De plus, lorsque l'alarme se déclenche, il est en fait... déjà trop tard... La seconde option est de simuler une

présence constante ; c'est l'approche adoptée ici. Nous n'allons bien évidemment pas utiliser un chronoprogrammeur qui, jour après jour allume/éteint une lampe toujours au même moment. Tout criminel un tant soit peu « invétéré » le constate immédiatement. Il va falloir nous creuser plus les méninges. Le circuit proposé ici commence par réagir à la lumière ambiante et, le soir venu, abaisse un volet roulant. Peu après, une lampe ou un simulateur de TV est allumé puis plus tard (à l'heure du coucher) coupé. Et le matin, le volet roulant remonte.

Le schéma

Un examen du schéma de la **figure 1**, de la gauche vers la droite, permet de

découvrir de manière logique les diverses fonctions. Tout à gauche, l'alimentation : la tension d'alimentation dérivée via R1 et C1 de la tension du réseau est redressée par D1 à D4 et stabilisée à une valeur continue de 12 V par D5/C2.

Attention : l'ensemble du circuit est relié directement au secteur ! Donc, non seulement le circuit doit être réalisé correctement et mis en boîtier sans le moindre risque de contact, mais tout test ou toute mesure mal faite comporte des **risques létaux**.

À droite de l'alimentation, nous trouvons un diviseur de tension photosensible, LDR1/P1. Tant qu'il y a (suffisamment) de lumière sur la photorésistance (LDR = **Light Dependent Resistor**), de jour donc,

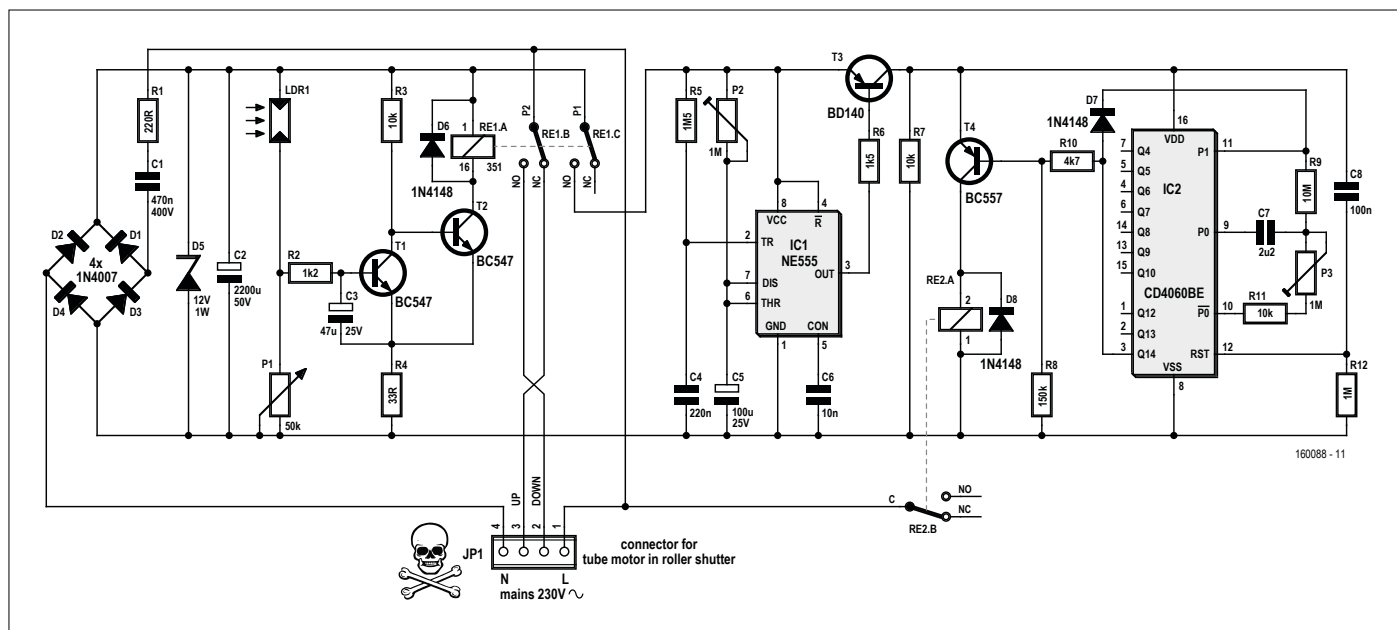


Figure 1. Le schéma se subdivise en trois parties : alimentation, capteur, commande.

sa résistance est faible et la tension au nœud avec P1 élevée. Le transistor T1 est alors passant et sa tension de collecteur faible. Par conséquent, le transistor T2 est bloqué et aucun courant ne circule pas à travers la bobine de RE1.A, le relais reste donc décollé et le circuit au repos. Par le contact NC (*Normally Closed* = fermé au repos) du relais bipolaire RE1, la tension du secteur est relayée au contact UP de JP1, contact auquel est connectée la connexion « remontée » du moteur de volet roulant (celle qui en assure la montée et le maintien dans cette position).

Dans cet état, la tension d'alimentation n'est pas (via RE1.C) transmise au reste du circuit centré sur IC1 et IC2. Lorsque, dans la soirée, le niveau de lumière baisse, la résistance de la photorésistance augmente. La tension sur la jonction de LDR1 et P1 diminue, et à un moment donné cette tension devient trop faible pour maintenir T1 en conduction (P1 permet de régler l'instant précis où cela se produit). Lorsque T1 est bloqué, son collecteur est haut et T2 devient passant. Il circule alors un courant à travers la bobine de RE1.A, le relais colle, activant ainsi le circuit. Via RE1.B, la tension du secteur arrive maintenant

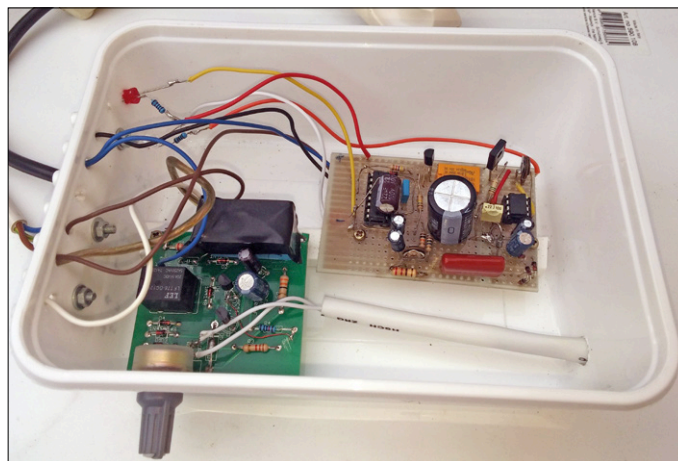


Figure 2. Le prototype de l'auteur dans un « boîtier » fortement « améliorable ».

au contact DOWN de JP1, la connexion éponyme du moteur de volet roulant est maintenant alimentée et le volet descend.

Temporisateurs

Via le contact NO (*Normally Open* = ouvert au repos) de RE1.C, la seconde partie du circuit, à savoir le tempori-

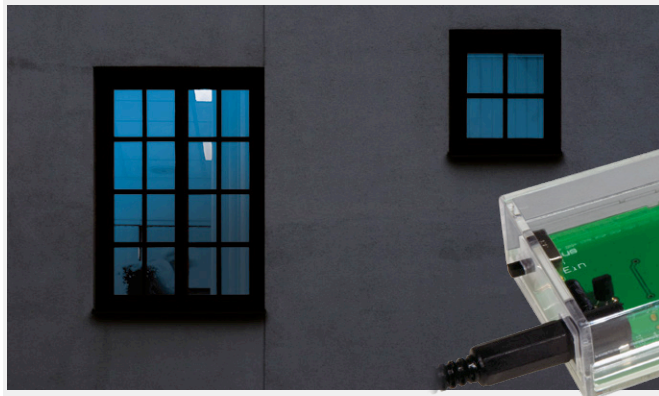
sateur IC1 (l'incontournable NE555), se trouve alimentée. Ce temporisateur remplit un rôle important. En effet, si à l'instant même où le volet roulant se met à descendre, une lampe (ou tout autre dispositif) est activée, cela paraîtrait louche. Des personnes « malintentionnées » pourraient avoir l'idée que les deux événements se déroulent vraiment de façon très (trop) synchrone. Une grande partie de l'efficacité du simulateur partirait en fumée.

À l'instant où la tension d'alimentation est appliquée au temporisateur IC1, sa sortie (broche 3) est haute. Le transistor PNP T3 est bloqué alors, de sorte que le second temporisateur (IC2) n'est pas alimenté. Après écoulement de la durée de basculement de IC1 (réglable à l'aide du potentiomètre P2), de quelque 30 à 50 s, la broche de sortie 3 passe au niveau bas et T3 commute la tension d'alimentation vers le second temporisateur (IC2).

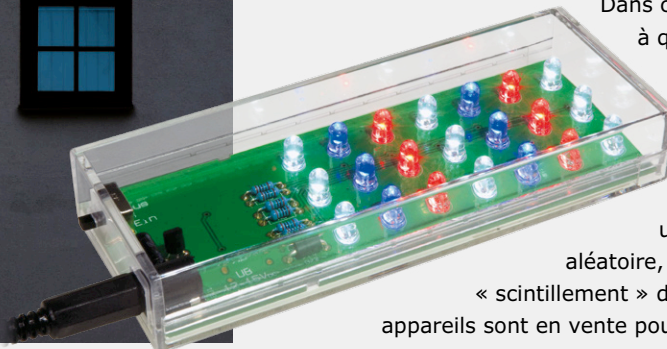
En fait, IC2 (un 4060), n'est pas un « vrai » temporisateur, mais un compteur numérique à oscillateur intégré doté d'un grand nombre de sorties amenées vers l'extérieur. Dès que la tension d'alimentation est commutée par T3, l'entrée de réinitialisation du CI (broche 12) reçoit une impulsion

L'auteur

Theo de Wijs (73 ans) est un cadre d'ONG à la retraite. Le cours d'électronique dont il a bénéficié date de l'époque des tubes électroniques et des premiers transistors. Jusqu'en 1976, il a été actif dans l'électronique (entre autres chez Philips) ; ensuite il a été appelé à d'autres tâches. L'électronique a toujours été son violon d'Ingres.



Simulateur de TV



Dans cet article, on parle à quelques reprises de simulateur de TV (télévision). Il s'agit là d'un circuit simple comportant quelques LED de couleur et un (pseudo) générateur aléatoire, qui permet d'imiter le « scintillement » d'une TV couleur. Ces appareils sont en vente pour moins de 10 €.

par l'intermédiaire de C8/R12 et toutes les sorties Q du CI sont au niveau bas – y compris la sortie Q14 (broche 3). Le transistor PNP T4 devient alors passant et alimente le relais RE2.A. Ce dernier colle et on peut, via le contact NO RE2.B, activer une lampe ou un simulateur de TV (ou quoi que ce soit d'autre). Après écoulement d'une durée réglable sur une plage large (allant jusqu'à un maximum de l'ordre de 24 h) avec le potentiomètre P3, la sortie Q14 passe au niveau haut. T4 est alors bloqué et le relais RE2.A retombe. Il simule l'heure de coucher : le simulateur de TV et/ou la lampe de salon s'éteint(en)t. Dans ce temporisateur, la diode D7 joue un rôle important : elle bloque l'oscillateur du CI lorsque Q14 passe au niveau haut, ce qui l'empêche ainsi de poursuivre son comptage.

Nous nous retrouvons maintenant dans un état stable : le compteur IC2 est à l'arrêt, le relais RE2.A est décollé et il le restera jusqu'au lever du soleil le jour suivant (du moins nous l'espérons). La lumière frappant la LDR augmente, la tension au niveau du nœud de la LDR1 et de P1 augmente et le transistor T1 passe en conduction. T2 est alors bloqué, le relais RE1.A retombe et le circuit se retrouve au repos. IC1 et IC2 ne sont

plus alimentés et peuvent donc, au repos et réinitialisés, démarrer le cycle suivant.

Quelques détails

Nous en avons pratiquement terminé en ce qui concerne ce circuit. Il est conçu de façon à ce que les deux relais consomment du courant tant qu'il fait « nuit », ce qui semble le plus logique : habituellement nous allons en vacances en été et en cette saison les nuits sont plus courtes que les jours. Cependant si l'on souhaite utiliser ce circuit durant la saison « sombre » (votre attention SVP, les amateurs de sports d'hiver), on peut par ex. utiliser pour T2 une version PNP et intervertir les connexions des contacts des relais. Les relais sont alors au repos quand il fait « nuit », de sorte que la consommation d'énergie du circuit diminue quelque peu.

Il est important que le moteur tubulaire assurant la montée et la descente du volet roulant soit pourvu de butées de fin de course. En d'autres termes, lorsque les positions finales (entièrement remonté/abaissé) sont atteintes, le courant d'alimentation du moteur doit être coupé par les butées de fin de course (le relais RE1.A du circuit reste actif lui). Cela est d'ailleurs le cas avec la plupart des volets roulants.

Il faut bien entendu que les contacts de

commutation du relais soient capables d'encaisser le courant absorbé par la charge. Pas de problème pour RE2 ; pour ce qui est de RE1, jetez un coup d'œil à la documentation du volet roulant. Un exemplaire 10 A devrait cependant faire l'affaire dans la plupart des cas.

Au début de cet article, nous avons souligné que le circuit est relié directement au secteur – l'alimentation ne comporte pas de transformateur. Il n'y a pas de raison que cela constitue un problème, mais la réalisation requiert plus d'attention que d'habitude. Il doit être impossible d'entrer en contact avec le circuit qui devra être logé dans un boîtier en matière plastique adéquat ; caractéristique spécifique du potentiomètre P1 : il doit avoir un axe en plastique !

La **figure 2** montre le prototype de l'auteur dont une partie est construite sur un morceau de carte de prototypage et bien qu'il ait été réalisé rapidement en tant que *proof-of-concept* (test de concept), il remplit parfaitement son rôle sous cette forme depuis un certain temps déjà. Néanmoins nous recommandons, pour un circuit qui doit rendre service de façon fiable et autonome pendant de nombreuses semaines, une construction un peu plus « robuste ». ◀

(160088 – version française : Guy Raedersdorf)

Moteurs tubulaires

Un moteur tubulaire est un moteur électrique allongé, monté dans un logement en forme de tube. Ce type de moteur est beaucoup utilisé dans les stores pare-soleil et les volets roulants. On les monte dans un arbre creux du volet roulant ou du store pare-soleil. Le moteur tubulaire est couplé à l'arbre par l'intermédiaire d'un adaptateur et d'un dispositif d'entraînement. Dans la description du moteur, les fabricants utilisent deux nombres : le couple en Nm

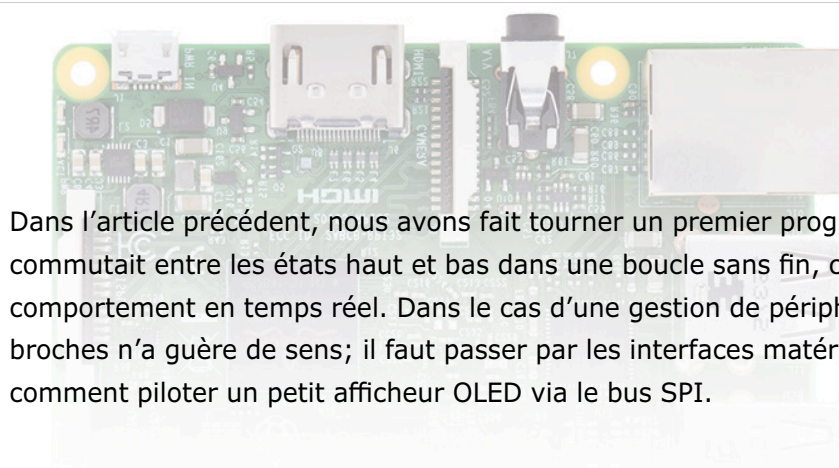
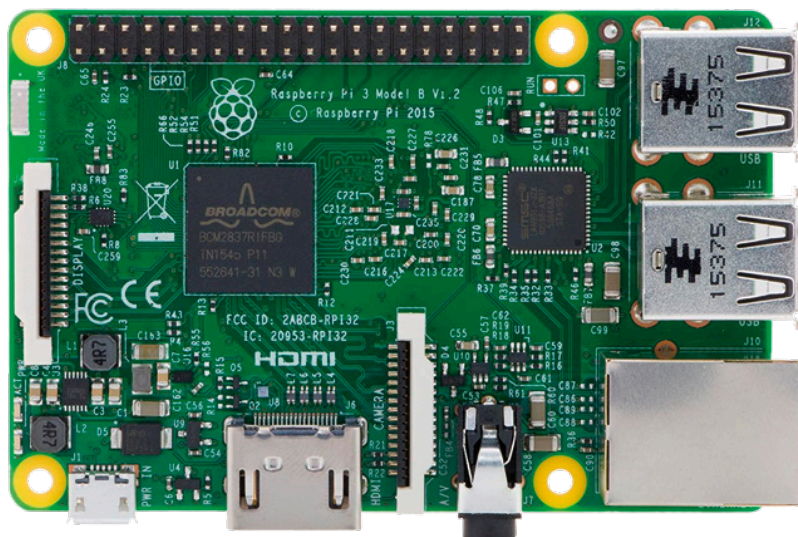
et le nombre de tours par minute. Un moteur tubulaire 6/17 a ainsi un couple de 6 Nm et une vitesse de rotation de 17 tours par minute.

(Source : Wikipedia)



Android sur Raspberry Pi (2)

afficheur piloté via le bus SPI



Dans l'article précédent, nous avons fait tourner un premier programme sous Android : une sortie numérique commutait entre les états haut et bas dans une boucle sans fin, ce qui nous a permis d'examiner son comportement en temps réel. Dans le cas d'une gestion de périphérique, la commande « manuelle » des broches n'a guère de sens; il faut passer par les interfaces matérielles du contrôleur. Nous montrons ici comment piloter un petit afficheur OLED via le bus SPI.

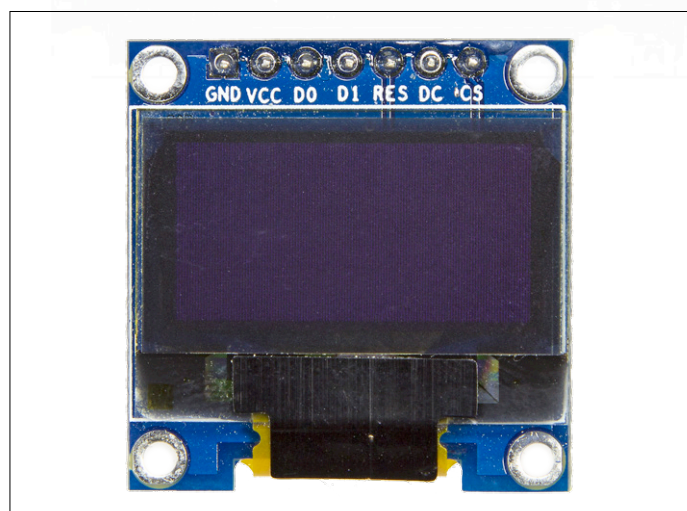


Figure 1. Cet afficheur est bien plus petit qu'un moniteur HD, aussi bien par sa taille que par sa consommation.

Tam Hanna (Slovaquie)

Si l'on veut gérer des périphériques avec Android Things, on a pour l'instant le choix entre SPI et I2C. Dans la suite, nous allons utiliser un afficheur OLED piloté par SPI, largement répandu et disponible chez AliExpress pour quelques dollars (**figure 1**).

Circuit

Un Raspberry Pi sous Android se comporte d'un point de vue matériel comme un Raspberry Pi ordinaire. Le circuit de la **figure 2** n'est donc guère différent de celui utilisé dans la série « Windows sur la carte RPi » [2].

Android Things identifie les bus SPI (de même que les broches GPIO) par des chaînes de caractères. Notre première tâche est d'identifier ceux du RPi qui sont à notre disposition :


```

@Override
protected void onCreate(Bundle savedInstanceState) {

    . . .

    PeripheralManagerService manager = new
        PeripheralManagerService();
    List<String> deviceList = manager.getSpiBusList();

    if (deviceList.isEmpty()) { } else {
        Log.i(TAG, "List of available devices: " +
            deviceList);
    }
}

```

Le programme affiche dans la console de débogage une liste de tous les bus disponibles. Au moment de la publication, le RPi 3 de l'auteur fournit deux références :

```

I/MainActivity: List of available devices: [SPI0.0,
    SPI0.1]

```

Ici, il faut éviter un petit piège tendu par Gradle ou Android Studio : si vous voulez recueillir les données au niveau d'un point d'arrêt, il faut prendre garde de démarrer le programme en cliquant sur l'icône de débogage. Si vous cliquez sur la flèche du démarrage normal, le débogueur n'est pas lié au programme.

Dans un premier projet de démonstration (dont on peut comme d'habitude télécharger le code depuis le site web d'Elektor [3]), nous devons commencer par initialiser le bus SPI. Pour cela, faisons appel à une nouvelle méthode appelée `configureAndFindDevice`. En premier lieu, munissons la classe d'activité principale `MainActivity` d'un objet GPIO global qui contrôle la ligne de réinitialisation de l'afficheur. En complément, ajoutons un objet `SpiDevice` qui encapsule la communication entre le RPi et l'afficheur.

```

public class MainActivity extends Activity {

    SpiDevice myDevice;

    Gpio myPinReset;

```

L'étape suivante inclut l'initialisation – plus ou moins classique – du bus SPI à l'intérieur de `configureAndFindDevice` (**listage 1**). Un appel à `setFrequency` fixe la fréquence de communication : les valeurs de 4 MHz ou 8 MHz ont donné toute satisfaction à l'auteur.

Les lecteurs attentifs de l'article précédent se demandent sans doute ce que `Thread.Sleep` vient faire dans le fil GUI. En fait, le délai de 10 ms n'est pas gênant et ne justifie aucunement la mise en place d'un fil spécifique.

Nous incorporons l'appel de cette méthode à `onCreate` de sorte qu'elle soit exécutée au démarrage du programme (**listage 2**). Ici, nous initialisons une autre broche GPIO qui permet de commander la ligne DC (*Data Control*). De plus, nous appelons la fonction `initDisplay` responsable de l'initialisation de divers paramètres de l'afficheur – le contenu de cette méthode est présenté dans la section suivante.

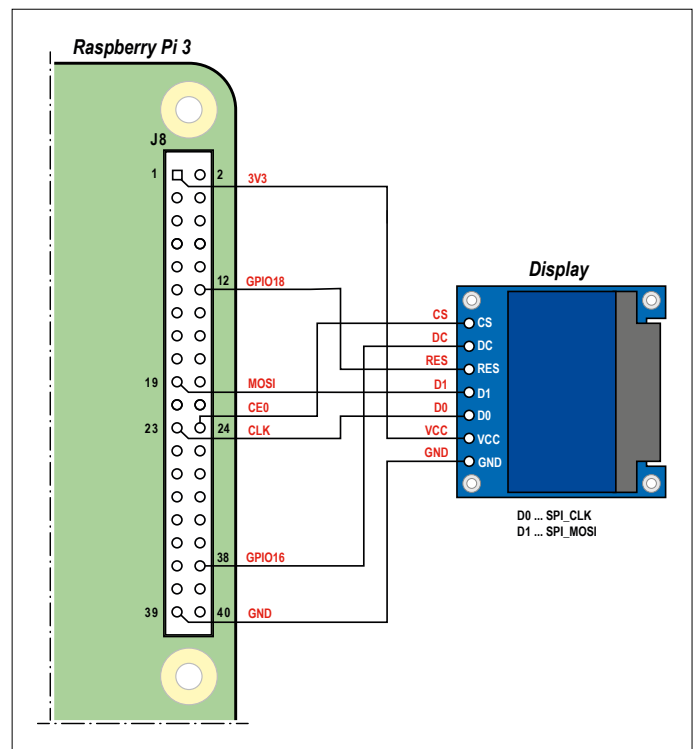


Figure 2. Il suffit de quelques fils pour raccorder le RPi à l'afficheur.

Pour être parfaitement complet du point de vue didactique, notez qu'il faut détruire l'objet `SPIdevice` en fin d'activité – c'est le rôle de la méthode `onDestroy` (**listage 3**).

Listage 1. Initialisation de l'interface SPI

```

void configureAndFindDevice(){

    PeripheralManagerService manager = new
        PeripheralManagerService();

    try {

        myPinReset = manager.openGpio("BCM18");
        myPinReset.setDirection(Gpio.
            DIRECTION_OUT_INITIALLY_LOW);
        myPinReset.setValue(false);

        myDevice = manager.openSpiDevice("SPI0.0");
        myDevice.setMode(SpiDevice.MODE0);
        myDevice.setFrequency(4000000);
        myDevice.setBitsPerWord(8);
        myDevice.setBitJustification(false);
        Thread.sleep(10);
        myPinReset.setValue(true);
    }

    catch (Exception e) { . . . }
}

```

Listage 2. Appelée au démarrage du programme, la méthode onCreate effectue les initialisations de toutes les broches.

```
@Override
protected void onCreate(Bundle savedInstanceState) {

    . . .

    configureAndFindDevice();

    try {

        PeripheralManagerService manager = new
            PeripheralManagerService();
        myPinDataCommand = manager.
            openGpio("BCM16");
        myPinDataCommand.setDirection(Gpio.
            DIRECTION_OUT_INITIALLY_LOW);

    }catch(Exception x){}

    initDisplay();

}
```

Si ces opérations de nettoyage font défaut, il peut se produire au débogage des exceptions du genre « android.os.ServiceSpecificException: BCM18 is already in use ».

Notons encore que des classes d'activité du système d'exploitation Android peuvent être supprimées en cours de fonctionnement. Il s'agit là d'une décision de conception des développeurs qui provoque des troubles en bien des lieux de l'univers Android.

Listage 3. À la fin du programme, on fait le ménage.

```
@Override

protected void onDestroy() {

    super.onDestroy();

    Log.d(TAG, "onDestroy");

    try {
        if(myDevice!=null) myDevice.close();
        if(myPinDataCommand!=null) myPinDataCommand.
            close();
        if(myPinReset!=null) myPinReset.close();
        myDevice = null;

    } catch (IOException e) {

        Log.w(TAG, "Unable to close SPI device", e);

    }

}
```

Nous initialisons...

Après la configuration réussie du *framework* SPI, nous pouvons nous occuper de la transmission des données qui mettront l'afficheur dans le mode correct ou seront affichées sur l'écran. Même si la feuille de caractéristiques du SSD1306 est très répandue, il est toujours bon, lors du premier emploi d'un circuit intégré, de rechercher chez Adafruit ou parmi la documentation d'autres fabricants, des bibliothèques complètes qui présentent le mode opératoire.

Le code d'`InitDisplay` est présenté dans le **listage 4**.

Pour commencer, on effectue un appel supplémentaire à `Thread.sleep` : cela garantit un temps de démarrage suffisant au contrôleur lorsqu'il se réveille après la réinitialisation. On envoie ensuite trois commandes qui démarrent la pompe de charge et mettent l'afficheur en marche. Avant l'envoi de la commande `sendData`, qui gère l'envoi de données à afficher, on s'assure que le tampon d'affichage contient des données plus ou moins aléatoires.

La déclaration des constantes des commandes prend une forme particulière, dont la plus notable est que chaque constante nécessite une conversion de format (`cast`) explicite vers un octet, car, en son absence, le compilateur Java créerait une arborescence :

```
private byte[] CMD_DISPLAY_ON = { (byte)0xAF };
```

Ensuite, nous pouvons utiliser les deux méthodes d'envoi des données vers le contrôleur de l'afficheur. Remarquons que pour la réception des données aussi bien que pour la collecte des commandes, le SD1306 se comporte comme un périphérique SPI. C'est le niveau de tension présent sur l'entrée numérique qui détermine le mode. Si le niveau est haut, les données vont

Listage 4. Affichage de lignes sur l'afficheur.

```
void initDisplay()

{

    try {

        Thread.sleep(100);
        SendCommand(CMD_CHARGE_PUMP_ON);
        SendCommand(CMD_MEMORY_MODE);
        SendCommand(CMD_DISPLAY_ON);

        for(int i=0;i<myDisplayBuffer.length;i++){
            myDisplayBuffer[i]=22;
        }

        sendData();

    }

    catch(Exception x){Log.e("Elektor",x.
        getMessage());}

}
```


dans le tampon d'affichage ; s'il est bas, elles sont stockées dans la mémoire de commande. Cela conduit au code suivant :

```
void SendCommand(byte[] _what){
    try {
        myPinDataCommand.setValue(false);
        myDevice.write(_what, _what.length);
    }

    catch (Exception x){
        Log.e("Elektor",x.getMessage());
    }
}
```

La version pour l'envoi de données d'affichage ne présente qu'un minimum de différences avec `SendCommand`. Au lieu de prendre comme données le code de commande, la fonction accède à un tableau défini comme membre (`myDisplayBuffer`).

```
void sendData(){
    try {
        myPinDataCommand.setValue(true);
        myDevice.write(myDisplayBuffer,
            myDisplayBuffer.length);
    }

    catch (Exception x){
        Log.e("Elektor",x.getMessage());
    }
}
```

Et voilà une première version du programme déjà opérationnelle. Exécutez-la ; si tout est correct, une série de lignes horizontales apparaissent sur l'écran (**figure 3**).

... et communiquons

Pour n'afficher qu'un motif constitué de lignes, nul besoin d'un système d'exploitation en temps réel dédié ; une mémoire d'affichage de 1 Ko sur un AVR fera parfaitement l'affaire. La puissance du RPi n'entrera en jeu que lorsque le système d'exploitation aura à afficher des informations complexes.

Une application serait de pouvoir afficher sur l'écran des éléments de commande GUI (comme un diagramme) sous la forme d'une matrice de points (*bitmap*). Pour cela, nous allons incorporer **MPAndroidChart** à notre programme – il s'agit d'une bibliothèque de diagrammes très puissante qui joue également un rôle dans le développement d'applications Android classiques. Commencez par ouvrir le fichier *build.gradle* du projet et complétez-le avec le code suivant :

```
allprojects {
    repositories {
        jcenter()
        maven { url "https://jitpack.io" }
    }
}
```

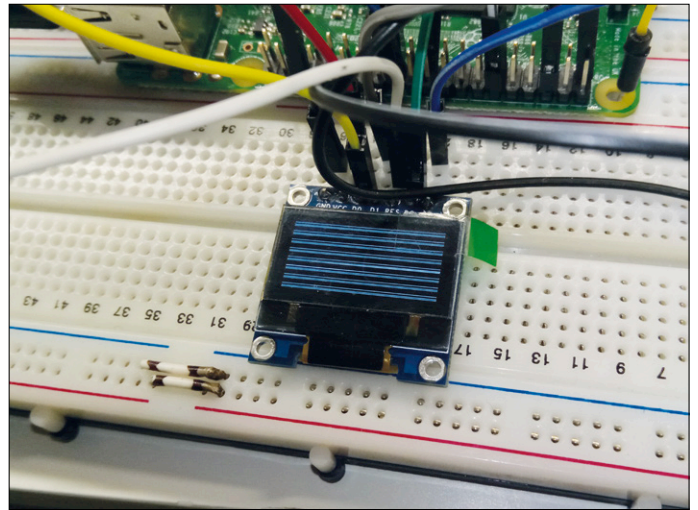


Figure 3. Le motif de test montre que tout fonctionne bien. Un afficheur OLED noir donne, par contre, l'impression d'être « mort ».

Il faut compléter le fichier *build.gradle* (*Module:app*) de l'appli de la manière suivante :

```
dependencies {
    provided 'com.google.android.
        things:androidthings:0.1-devpreview'
    compile 'com.github.PhilJay:MPAndroidChart:v3.0.1'
}
```

Après l'enregistrement de ces deux fichiers, Android Studio affiche un bandeau jaune dans lequel il faut cliquer sur *Sync Now*, ce qui demande à l'EDI d'actualiser l'ensemble du projet. L'une des plus grandes forces de Gradle est sa capacité de télécharger sur l'internet les bibliothèques et autres informations nécessaires au cours de la compilation.

Ensuite, faites un clic droit sur le dossier du projet et choisissez l'option *New* → *File* → *Android Resource File*. Dans la fenêtre de dialogue qui s'affiche, saisissez les paramètres tels que représentés sur la **figure 4** pour créer un fichier de mise en page (*layout*).

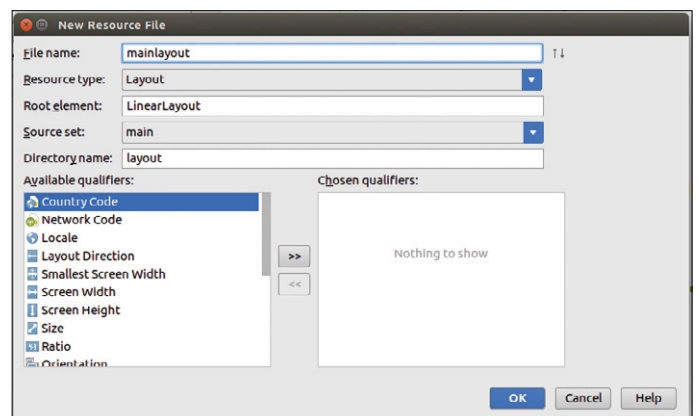


Figure 4. Ces paramètres servent à créer un nouveau fichier de mise en page.

Par défaut, le sous-système de mise en page d'Android produit un élément de commande affichable dont la taille est ajustée dynamiquement, alors que nous voulons un *widget* d'une taille fixe de 128 × 64 pixels. Il nous faut donc modifier le code de la manière suivante :

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android=http://schemas.android.com/
    apk/res/android
    android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

    <com.github.mikephil.charting.charts.LineChart
        android:id="@+id/chart1"
        android:layout_width="128px"
        android:layout_height="64px"/>

</LinearLayout>
```

Le détournement par l'écran principal du RPi utilisé ici n'est pas absolument nécessaire, mais il facilite le travail, car il permet d'observer l'affichage produit sur un moniteur Android Studio connecté.

Nous pouvons maintenant nous occuper de la bibliothèque graphique, qui a besoin d'une directive pour l'affichage de diagrammes. Celle-ci prend la forme suivante :

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.mainlayout);
}
```

La première différence avec une appli Android Things habituelle est l'appel de `setContentView`, qui s'occupe du chargement des informations contenues dans le fichier XML de mise en page. Ensuite nous déclarons un tableau rempli de données d'affichage plus ou moins aléatoires :

```
ArrayList<Entry> values = new ArrayList<Entry>();

for (int i = 0; i < 20; i++) {
    float val = (float) (Math.random() * 25) + 3;
    values.add(new Entry(i, val));
}
```

Nous en arrivons au paramétrage : la bibliothèque *MPAndroidChart* est conçue pour le traitement de vastes ensembles de données, ce qui rend son initialisation relativement complexe, dont voici seulement quelques extraits :

```
LineChart mChart = (LineChart) findViewById(R.
    id.chart1);
mChart.getDescription().setEnabled(false);
LineDataSet s1=new LineDataSet(values,"");

. . .

ArrayList<ILineDataSet> dataSets = new
    ArrayList<ILineDataSet>();
```

```
dataSets.add(s1);
LineData data = new LineData(dataSets);
mChart.setData(data);
```

Le premier problème, à l'origine de la taille énorme du code, est l'élimination des commandes pour les caractères (inutiles ici) : la résolution de l'afficheur de 128 × 64 pixels est tellement petite qu'elle n'a sans doute jamais été testée par les développeurs de *MPAndroidChart*.

Le problème suivant est de convertir le résultat en une matrice de points qui peut être envoyée à l'afficheur. Ici, il importe surtout d'éliminer l'information de couleur, notre afficheur ne gère bien entendu que le noir et blanc. À cause de la procédure assez particulière appliquée aux caractères par *MPAndroidChart*, nous devons retarder la conversion ; c'est possible avec un pilote, voici le squelette de son code :

```
final Handler handler = new Handler();
handler.postDelayed(new Runnable() {
```

```
    @Override
    public void run() {
```

```
        //Code ist hier
```

```
    }
}, 700);
```

`postDelayed` accepte un objet qui doit comporter une méthode `run()` qui contient elle-même le code à exécuter après écoulément du délai. C'est assez compliqué – commençons par la conversion du widget en une instance de la classe `Bitmap` :

```
public void run() {
    LineChart mChart = (LineChart) findViewById(R.
        id.chart1);

    Bitmap returnedBitmap = Bitmap.createBitmap(128,
        64, Bitmap.Config.ARGB_8888);
    Canvas canvas = new Canvas(returnedBitmap);

    Drawable bgDrawable =mChart.getBackground();
    if (bgDrawable!=null)
        bgDrawable.draw(canvas);
    else
        canvas.drawColor(Color.WHITE);

    mChart.draw(canvas)
```

À ce moment, `returnedBitmap` contient une version couleur du diagramme produit par *MPAndroidChart*. Nous devons donc convertir cette matrice de points (*bitmap*) en un tampon de trame (*frame buffer*). Et voici notre première cause d'irritation : la structure de la mémoire de l'écran ; elle est divisée en un groupe de pages !

La solution la plus simple à ce genre de problème est de traiter les octets de mémoire l'un après l'autre dans des boucles `for`. Dans notre cas, il nous faut trois boucles :

```
byte counter=0;
```

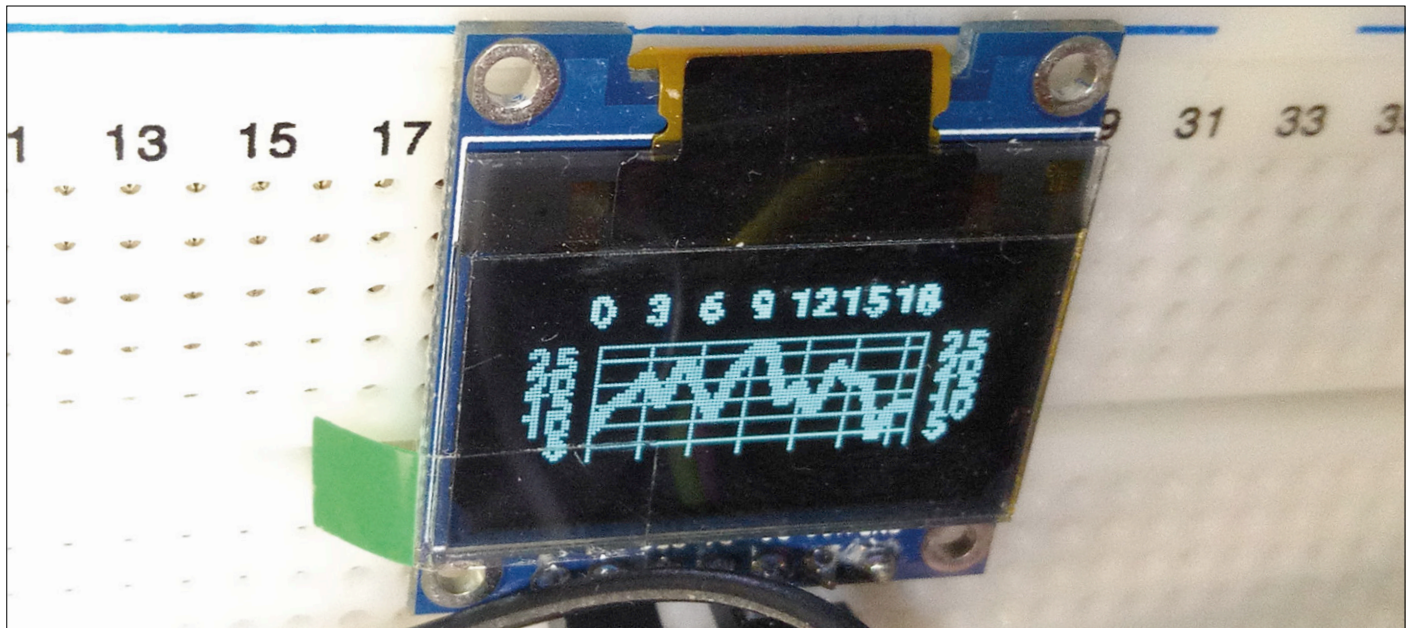



Figure 5. Grâce à Android, on peut afficher sans gros efforts un sympathique diagramme linéaire sur une grille.

```
byte store=0;
int counter2=0;
for(int ctr=0;ctr<8;ctr++)
    for(int x=0;x<returnedBitmap.getWidth();x++)
    {
        for (int y = ctr*8; y < (ctr+1)*8; y++)
        {
```

L'étape suivante consiste à rassembler huit pixels dans un octet. Pour cela, nous commençons par acquérir l'information de couleur de chaque pixel pour ensuite la convertir en blanc ou noir par une évaluation simple de ses composantes :

```
int myVal=returnedBitmap.getPixel(x,y);
if(Color.red(myVal)!=255 || Color.
    green(myVal)!=255 || Color.blue(myVal)!=255)
    { //Set
        store=(byte)(store | ((byte)(1 <<
            (counter))));
    }
}
```

Le remplissage d'un octet se fait donc en huit pas. Les informations sont rangées une à une dans `store` au moyen d'instructions de manipulation de bits ; dès qu'un octet est plein, il est copié dans `myDisplayBuffer`.

```
counter++;
if(counter==8) {
    myDisplayBuffer[counter2]=store;
    store=0;
    counter=0;
    //Byte raus
    counter2++;
}
}
```

Pour finir, nous envoyons quelques commandes au contrôleur de l'afficheur pour le mettre dans un état bien défini. `sendData` envoie alors les informations à l'afficheur via le bus SPI :

```
SendCommand(CMD_RESETCOLADDR);
SendCommand(CMD_RESETPAGEADDR);

sendData();

}
```

Le diagramme de la **figure 5** montre le fruit de nos efforts.

Conclusion

Nous voici arrivés à la fin de nos expérimentations avec Android Things. Ce n'est certes pas une plateforme pour tout le monde, mais si vous arrivez à en tirer quelque chose, vous aurez à votre service des fonctions nombreuses et puissantes. Il y a beaucoup d'applications qui ne sont pas ou seulement très difficilement à la portée des contrôleurs à 8 bits, alors mettre à la tâche un puissant processeur à 32 bits permet de gagner beaucoup de temps et d'effort. Nous vous souhaitons beaucoup de succès dans ce domaine ! ◀

(160369 – version française : Helmut Müller)

Liens

- [1] www.elektormagazine.fr/160361
- [2] www.elektormagazine.fr/150520
- [3] www.elektormagazine.fr/160369



projet 2.0

corrections, mises à jour et courrier des lecteurs

carte de commande Wi-Fi : le retour

Elektor 11/2016, p. 32 (150402)

La commande par PC (via la page web [150402.html](#)) n'était pas possible jusqu'ici avec les navigateurs Internet Explorer et Edge. Lorsque les pages sont stockées localement sur le disque dur, ces navigateurs ne permettent pas d'écrire en local (ici, l'adresse IP de la carte contrôleur WAN, saisie une seule fois). Clemens Valens, du labo d'Elektor, a conçu une nouvelle page qui interroge le type de navigateur et bloque l'enregistrement local si celui-ci n'est pas possible. Les données du formulaire sont néanmoins envoyées, ce qui fait que la commande fonctionne. La nouvelle version est disponible sur la page du labo d'Elektor [www.elektormagazine.fr/labs/remake-wifi-controller-board-150402](#) ainsi que sur la page du projet [www.elektormagazine.fr/150402](#).

le déwattteur, ampli en classe D (1)

Elektor 12/2016, p. 36 (150115)

Dans l'article, page 38, on peut lire en bas à droite : « Le gain est actuellement de un, mais vous pouvez le modifier en changeant R3 : $A = R4/R3 + 1$ ». Il faut lire : « Le gain est égal à 1 lorsque la résistance R3 n'est pas montée. On peut l'augmenter en câblant R3 ($A = R4/R3 + 1$) ».

le déwattteur, ampli en classe D (2)

Elektor 12/2016, p. 36 (150115)

L'article ne précise pas comment régler P1. Procédez de la manière suivante : une fois toutes les connexions établies, mettez l'amplificateur sous tension et vérifiez la fréquence MLI des deux canaux, qui devrait être égale à 400 kHz. Mesurez la fréquence sur L1 (côté connexion à T1) en l'absence de signal d'entrée et réglez P1 pour ajuster la fréquence. Si vous ne disposez pas d'un instrument approprié pour mesurer la fréquence, positionnez P1 au milieu de sa course sur les deux canaux. Vous pouvez retrouver ces explications sur le site du labo d'Elektor ([www.elektormagazine.com/labs/200w-class-d-audio-power-amplifier-150511](#)).



hors-circuits - ampli de classe A, B, C, D, E, F, G, H : quésaco ?

Elektor 11/2015, p. 62 (150527)

Il est indiqué dans l'article que le rendement théorique d'un amplificateur en classe B peut aller jusqu'à 100%.

Cela ne cadre pas avec mes connaissances ni avec la littérature, dans laquelle on trouve un rendement maximal de $\pi/4 = 78,53\ldots\%$ pour des signaux sinusoïdaux. L'article dit aussi qu'il est possible d'atteindre un rendement plus élevé avec des MOSFET qu'avec des transistors bipolaires. Je ne vois pas pour quelle raison, les deux transistors travaillant en régime non saturé.

Plus loin, il est dit qu'un amplificateur en classe C peut atteindre un rendement maximal de 70%. Je suis là d'un autre avis. Théoriquement, la classe C permet de s'approcher aussi près que l'on veut de 100% en réduisant l'angle de conduction (qui donne la fraction de la période pendant laquelle le transistor est passant). Dans la pratique, on peut atteindre jusqu'à 90%, mais il faut alors travailler avec des courants élevés. On ne peut atteindre des rendements convenables qu'en remplaçant la résistance de collecteur R1 (**figure 3**) par une inductance.

Encore plus loin, il est dit que pour un étage de classe E, il faut des transistors à commutation très rapide comme pour

la classe D. Cela non plus ne correspond pas à mes connaissances. On peut utiliser des transistors dont les temps de commutation sont de l'ordre de la moitié de la période de fonctionnement. Ou inversement : les transistors rapides permettent d'amplifier des signaux à très haute fréquence. Il est dommage qu'un avantage décisif de la classe E ne soit pas mentionné, à savoir que la capacité de sortie du commutateur n'a pas d'effet négatif.

Martin Obmann

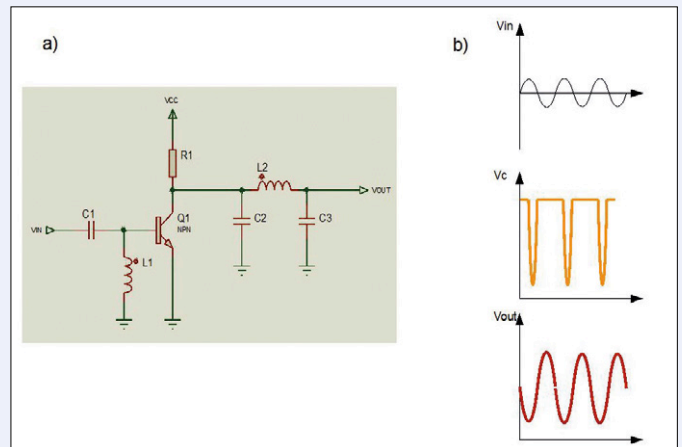
Bonjour Martin,

Vous avez tout à fait raison pour la classe B, c'est une erreur dans l'article. Comme vous l'avez mentionné, le rendement maximal est de 78%, au moins pour des signaux sinusoïdaux. Toutefois je pense toujours que le rendement des MOSFET est légèrement supérieur à celui des transistors bipolaires, surtout parce qu'ils permettent de s'approcher davantage des tensions d'alimentation dans l'étage de sortie, ce qui limite ainsi les pertes globales. J'aurais dû préciser : « typiquement autour de 70%, jusqu'à 78% pour les circuits particulièrement bien optimisés ».

À propos de la classe C : là encore, vous avez théoriquement raison, le rendement peut aller jusqu'à 100%... si les transistors et les composants passifs sont tous parfaits. La valeur de 70% que j'ai mentionnée provient de ma propre expérience avec des amplificateurs RF. C'est un véritable défi d'en obtenir davantage aux Ultra Hautes Fréquences ou au-dessus, mais j'aurais dû apporter là-dessus quelques précisions.

En ce qui concerne la classe E, je ne pense pas qu'un transistor aussi lent puisse faire l'affaire, même si je suis loin d'être un expert dans ce domaine. Voir par ex. : http://people.physics.anu.edu.au/~dxt103/160m/class_E_amp_design.pdf

Robert Lacoste



SDR d'Elektor réinventé (1) + (2)

Elektor 07-08/2016, p. 50 (150515), Elektor 09/2016, p. 46 (160048)

Bonjour Jan,

J'ai le plaisir de vous informer que mon logiciel G8JCFSDR (Build 274) supporte désormais complètement le *shield* SDR d'Elektor.

J'étais déjà en contact avec Burkhard (Kainka, NDRL), qui a déjà testé la version « rechargée » de G8JCFSDR.

La version de G8JCFSDR qui supporte le *shield* SDR d'Elektor offre une précision de réglage de 1 Hz, sans problème de transitoire, un dispositif de calibrage facile à utiliser (pour garantir que votre *shield* SDR est toujours calé sur la bonne fréquence), ainsi qu'une fonction intégrée de chargement du micrologiciel Arduino, ce qui simplifie la mise à niveau. Le logiciel G8JCFSDR est disponible au téléchargement sous www.g8jcfldr.dyndns.org. Je serais très heureux si vous pouviez informer vos lecteurs de ces nouvelles fonctions.

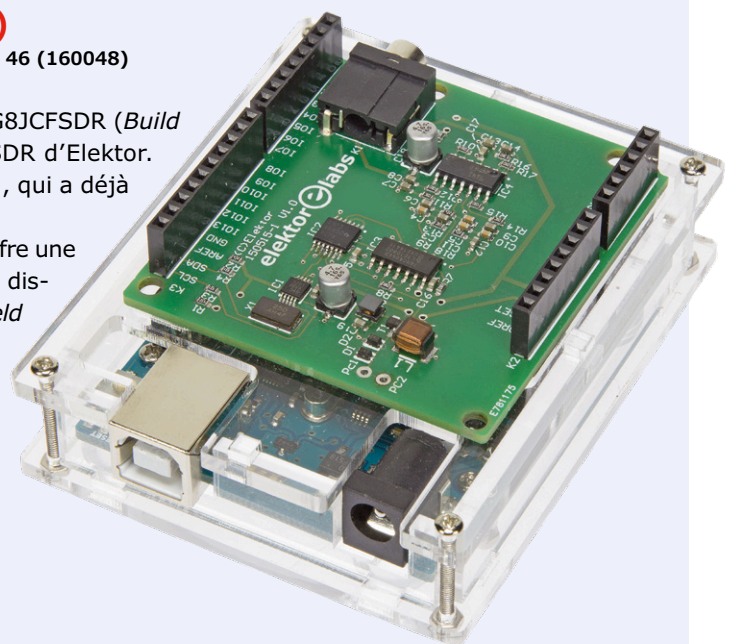
Peter, GM8JCF

g8jcf@g8jcfldr.dyndns.org

Bonjour Peter,

Ces nouvelles sont réjouissantes et je suis très content de les partager avec nos lecteurs. Nous allons également publier ton adresse de courriel avec la mise à jour.

Jan Buiting, PE1CSI



chien de garde pour ordinateur

Elektor 06/2016, p. 51 (140344)

Un nouveau logiciel est disponible au téléchargement sur la page du projet (www.elektormagazine.fr/140344).

PET fête ses 40 ans, bon anniversaire !

finies les soudures,
vive le BASIC pour
tous

Nous sommes en janvier 1977, au « CES », le salon professionnel dédié à l'électronique grand public. Chicago est figée par le froid, mais pas le public qui découvre avec admiration le premier ordinateur personnel complet, prêt à l'emploi : le *Personal Electronic Transactor* de Commodore ou pour les intimes « PET 2001 ».

Karl-Ludwig Butte
(Allemagne)

Photo :
Wikimedia Commons

Deux ans auparavant, le magazine *Popular Electronics* fait sensation en publiant un article sur un microordinateur, l'Altair 8800, fourni en kit. À l'époque, les passionnés du microordinateur bouillonnent déjà d'enthousiasme (fig. 1). Les créations per-

sonnelles prolifèrent, des centaines de livres sont publiés sur « Comment fabriquer son propre ordinateur » (fig. 2), les clubs informatiques comme

le *Homebrew Computer Club* [1] poussent comme des champignons. Cependant, le temps est venu de franchir l'étape suivante et de présenter des ordinateurs complets, comportant processeur, alimentation électrique, mémoire vive, clavier et mémoire de masse, le tout logé dans un seul et même boîtier ! PET offre tout cela et le clou est que l'interpréteur de



EST[®] 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

BASIC Commodore annonce sa prise de service quelques secondes après la mise sous tension pour exécuter immédiatement les instructions BASIC saisies par l'utilisateur. Plus besoin de commutateurs binaires comme pour l'Altair 8800 et plus besoin non plus de taper des valeurs hexadécimales comme c'était le cas avec KIM-1 (le tout premier Commodore lancé sur ce marché prometteur). Au contraire, des mots (BASIC) clairs, compréhensibles tels que GOTO, PRINT ou IF ...THEN suffisent pour programmer le PET 2001 et assurer qu'il exécute les choses les plus remarquables. Une véritable révolution, dont nous avons du mal à saisir l'ampleur aujourd'hui, pour nous qui disposons désormais de processeurs à huit cœurs, de gigaoctets de mémoire, de téraoctets de disque dur et d'interfaces utilisateurs graphiques.

Le matériel

Qu'est-ce qu'on obtient à l'époque pour 595 \$ US (soit 2 500 € aujourd'hui) [2] ? Le cœur du PET est un processeur 6502 doté de 4 Ko de mémoire vive. Pour 8 Ko de mémoire vive, il faut rajouter 200 \$ US [3]. L'écran monochrome de neuf pouces affiche des caractères blancs ou verts sur fond noir. Le clavier issu de la fabrication de calculatrices de bureau de la maison Commodore comprend 73 touches. L'avantage est que le clavier prend moins de place que celui d'une machine à écrire et qu'il est possible d'ajouter à gauche du clavier un magnétophone capable de lire des cassettes compactes et d'enregistrer données et programmes. Le port dit *datasette*, quant à lui, est destiné à connecter une *datasette* externe. Doté de deux autres interfaces, PET a l'avenir devant lui. La première interface sert de port utilisateur pour accéder, entre autres, aux connexions d'E/S numériques du processeur 6502. L'utilisateur peut ainsi y connecter son propre matériel.

La plupart des ordinateurs personnels de cette époque tels que l'Apple II, le TRS-80 ou le C64 sont dotés d'extensions similaires. Cette option disparaît néanmoins des générations de PC qui suivent pour réapparaître en 2012 sur le Raspberry Pi sous le nom de « port GPIO ». Le port utilisateur fait non seulement le bonheur des bricoleurs, mais aussi des jeunes pousses de l'époque qui peuvent ainsi développer des dispositifs auxiliaires pour ordinateurs personnels.

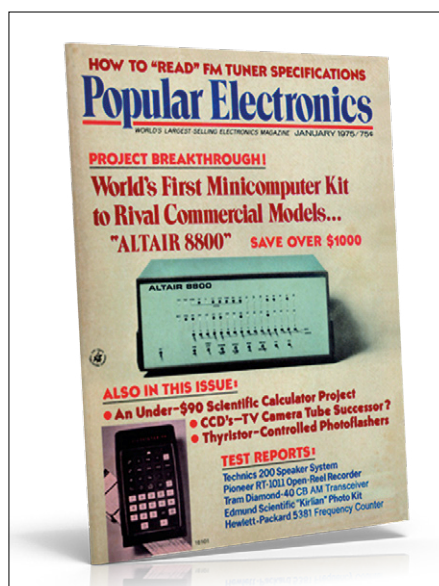


Figure 1. Couverture du numéro de janvier 1975 de « Popular Electronics » (Ziff Davis Publishing Company).

Commodore utilise la deuxième interface, un bus IEEE488, pour connecter des périphériques tels qu'imprimante ou lecteur de disquettes (**fig. 3**). Ce système de bus est remarquable à bien des égards. En effet, il assure le pilotage simultané des périphériques reliés en série à PET (principe de la guirlande, *Daisy Chain*). Commodore n'a pas besoin de décider dès le départ du nombre et du type de périphériques qui seront utilisés plus tard. Le bus permet de raccorder jusqu'à quinze appareils, ce qui est largement suffisant. Hormis ce bus IEEE-488, pas besoin de monter d'autre matériel sur la carte mère de PET. Certes, cela implique que les périphériques doivent être dotés de leur propre intelligence sous la forme d'un système à microcontrôleur complet, mais cela présente l'avantage aux fabricants de pouvoir se faire payer en conséquence. Commodore va ainsi gagner du temps et donc de l'argent en lançant PET rapidement sur le marché. À l'époque, Apple fait exactement l'inverse. Sept emplacements libres disposés sur la carte mère sont disponibles pour les extensions. Ceci permet p. ex. de connecter un contrôleur de disquettes qui constitue uniquement l'interface électrique avec l'électronique du lecteur. Le processeur de l'Apple II limite à deux le nombre de lecteurs qu'il peut commander. PET, quant à lui, n'a rien d'autre à faire que d'envoyer une commande au lecteur du disque magnétique amovible qui copie la disquette 1 sur la disquette 2 comme

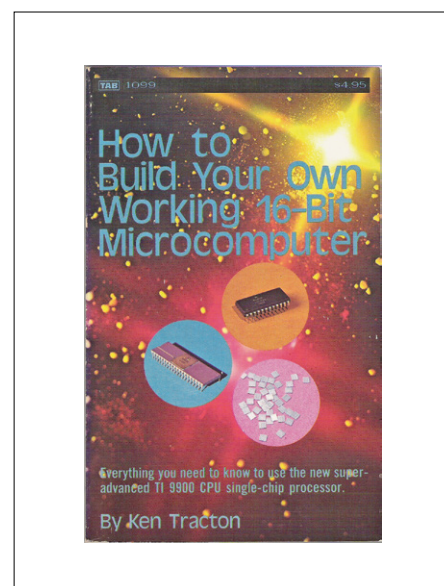


Figure 2. Un des nombreux ouvrages « Comment construire soi-même ».

un grand, pour que PET puisse se vouer à d'autres tâches.

L'ensemble du matériel est logé dans un boîtier métallique qui s'ouvre comme un capot moteur. Il dispose même d'une béquille (**fig. 4**). Cette construction solide a certainement contribué au fait que beaucoup de PET ont résisté au temps. On les retrouve sur les sites de vente aux enchères en ligne pour un prix



Figure 3. Système informatique complet : PET 2001, imprimante et lecteur de disquettes.



Figure 4. Il suffit de desserrer deux vis pour soulever le capot.

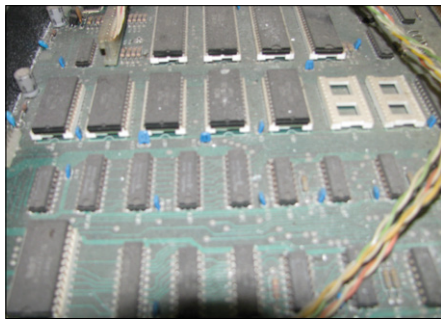


Figure 5. Carte mère à son arrivée chez l'auteur.

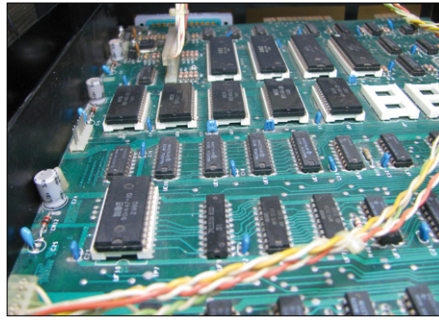


Figure 6. Carte mère après une bonne session de nettoyage.



Figure 7. Écran de démarrage BASIC (Tom Knox ; domaine public ; wikipedia).

variant entre 200 et 800 €. En général, ils arrivent dans un piteux état chez leur nouveau propriétaire. Avec un peu de chance, il suffit de les dépoussiérer (**fig. 5**) pour qu'ils retrouvent leur splendeur d'antan (**fig. 6**).

Le logiciel

L'interpréteur de BASIC de PET qui se trouve dans une ROM de 14 Ko (**fig. 7**) sert à la fois de système d'exploitation et d'environnement de développement. Il repose sur un BASIC Microsoft et il est doté d'un « éditeur plein écran » [4]. Ceci permet de visualiser les lignes de programme et de les éditer ultérieurement en amenant tout simplement le curseur à l'endroit à modifier. Pour confirmer les modifications, il suffit d'appuyer sur la touche « Return ». Quel confort pour l'époque ! En plus des commandes BASIC usuelles, le PET dispose également de PEEK et POKE, pour lire ou écrire des octets dans la mémoire.

Les noms de variables se différencient à l'aide des deux premiers caractères. Les noms peuvent être plus longs, mais les autres caractères sont tout simplement ignorés. Ainsi pour l'interpréteur de BASIC la variable « SO » peut correspondre à SOURIS ou SOURCE.

Le BASIC Commodore connaît trois types de données différents : les nombres flottants sur 40 bits, les nombres entiers sur 16 bits et les chaînes de caractères (*strings*) limitées à 255 caractères. Pour les distinguer, on se sert d'un caractère spécial placé à la fin du nom de la variable. Les chaînes de caractères sont dotées d'un \$ tandis que les variables entières se reconnaissent au %. Si le caractère \$ manque, la variable est traitée comme un nombre flottant. Les variables ne peuvent pas être déclarées.

Si l'interpréteur détecte une nouvelle variable, il lui réserve un certain nombre d'octets en fonction du type de données. En revanche, l'interpréteur ne pardonne pas les fautes de frappe dans les noms des variables. Malheur à celui qui saisit FRACTURE\$ au lieu de FACTURE\$. Il risque de se casser la tête pour trouver l'erreur, à moins d'avoir un ami relecteur attentif. En effet, l'expérience montre qu'il est difficile de détecter ses propres fautes de frappe.

Si tout va bien, la commande BASIC est presque exécutée de manière instantanée. PRINT 2*SQR(25), un coup de <Return> et le résultat « 10 » s'affiche à l'écran en un rien de temps. Si l'instruction est précédée d'un numéro de ligne, PET attend que l'on saisisse RUN avant d'exécuter la commande. Au bout de deux ou trois programmes, on s'aperçoit rapidement qu'il vaut mieux compter large et numéroter ces lignes de programmation de 10 en 10. En effet, si l'on a un programme de 10 lignes et qu'on les numérote de 1 à 10, il est impossible de rajouter une onzième ligne de programmation par exemple entre la ligne 4 et 5. Il faut renommer manuellement les lignes 5 à 10 pour avoir les lignes 6 à 11. De plus, il convient de ne pas oublier d'adapter les lignes en conséquence pour les sauts de ligne conditionnels (IF) et les sauts de lignes inconditionnels (GOTO). Au départ, il n'y a pas de commande du type RENUMBER pour

automatiser la numérotation. Les programmeurs doivent se creuser la tête, car très vite il devient clair qu'il sera difficile pour un programme BASIC de renuméroter un autre programme écrit en BASIC. Il faut donc prendre le taureau par les cornes et recourir aux grands moyens. La solution est alors trouvée, elle s'appelle « assembleur », un programme écrit en langage machine, logé dans une partie de la mémoire protégée. Heureusement, PET est doté d'un moniteur de langage machine (au départ sur une cassette, plus tard dans la ROM). Les choses deviennent alors intéressantes et chaque programmeur de PET connaît bien la commande secrète qui lui permet d'aller plus loin. SYS 1024 <Return>. Résultat : le langage machine prend la relève du BASIC. Les magazines, les livres et les lettres d'information de nombreux clubs informatiques sont bourrés de listings en code hexadécimal. Ces derniers permettent p. ex. de mettre en œuvre la commande RENUMBER, mais aussi de connecter des appareils bien pratiques tels qu'un lecteur de code à barres.

Les applications

PET 2001 présente également des aspects ludiques. Bien sûr, ces jeux n'ont rien à voir avec les jeux megapixels multicolores que nous avons maintenant sur nos ordiphones. À l'époque, PET n'a même pas d'affichage graphique pour pouvoir piloter chaque pixel séparément. En revanche, il dispose de toute

Liens

- [1] https://fr.wikipedia.org/wiki/Homebrew_Computer_Club
- [2] www.minneapolisfed.org
- [3] www.kuto.de/cbmmuseum/cbm_pet.html
- [4] https://en.wikipedia.org/wiki/Commodore_BASIC

une série de symboles graphiques avec lignes, blocs ou caractères qui s'adaptent à la trame des lettres et des chiffres affichables à l'écran. Le « graphique bloc », comme il s'appelle, permet d'afficher des graphiques en barres, des masques d'écran, mais aussi de jouer.

Les applications mathématiques ne manquent pas, du simple calcul au programme qui résout des équations compliquées en passant par le plan de remboursement. BASIC dispose de fonctions mathématiques telles que LOG (logarithme) et SQR (racine carrée).

L'apparition des imprimantes et des lecteurs de disquettes ouvre la voie aux applications commerciales comme le traitement des commandes, la facturation et la gestion des stocks. Les petites entreprises se risquent alors à faire le pas, parce qu'un ordinateur Commodore doté d'une imprimante et d'un lecteur de disquettes coûte beaucoup moins cher que le modèle de base proposé par les géants de l'informatique de l'époque. La *killer app* du moment s'appelle VisiCalc, un tableur lancé sur le marché en 1979. Écrit au départ pour Apple II, ce logiciel est porté peu de temps après sur PET. Ainsi PET doit son entrée dans le monde de la bureautique à l'apparition des premiers traitements de texte.

De nombreux programmeurs amateurs font leurs premiers pas dans ce domaine en connectant des périphériques au port utilisateur. On trouve même des programmes pour transformer le PET en boîte à musique. Le PET permet également de résoudre des petites tâches de mesure et de pilotage.

Conclusion

L'ordinateur PET a non seulement écrit un chapitre important de l'histoire de l'informatique, mais aussi de celle de l'économie. Il a certainement permis d'introduire l'informatique dans l'entreprise et d'élargir le marché. Les programmeurs et les conseillers en informatique ont rapidement remplacé les typographes. Peu sont ceux qui ont réalisé que le PET a révolutionné l'informatique de la fin des années 1970. ◀

(160353 – version française : Pascal Duchesnes)



Réactions tardives

compilées par Jan Buiting

Les articles publiés dans notre rubrique « rétronique » ont toujours suscité de nombreuses réactions de la part de nos lecteurs depuis 2004. Comme vous pouvez le constater, celles-ci ne se limitent pas aux courriers. Ci-après, vous trouverez quelques exemples de dons et autres contributions qui ont suivi le tout premier article de cette série. Bien entendu, les personnes nommées ont donné leur accord pour cette publication.

Objet 04 : horloge MOS 5314 Elektor

L'article sur l'horloge MOS a déclenché toute une série de réactions. Alan Woodman s'est mis à construire de nombreuses horloges numériques. Aujourd'hui, nous avons le plaisir de présenter son horloge sur oscilloscope. Reginald Neale a monté la grosse puce MM5316 sur une platine d'expérimentation. Bernd Rinke a numérisé l'article original datant de 1974. Il a refondu les circuits imprimés. Le tout peut être téléchargé gratuitement sur www.elektormagazine.fr/160102. Philippe Decock, quant à lui, nous a envoyé une série d'anecdotes amusantes, en nous signalant également qu'il y avait un lien qui manquait !

État : sourires et gratitude.

Déclencheur 04 : horloge MOS 5314 (1974), Elektor 11/2016.



Objet 05 : radio Roberts type 49041

Le monde est petit ! Mon médecin traitant et mon dentiste lisent eux aussi Elektor. Ils ne manquent aucune occasion pour apporter leur grain de sel aux formules physiques et mathématiques présentées dans le magazine, mais aussi leur humour. Avec l'ordonnance, mon médecin, Gaspard Knops, me remet une « radio Roberts » des années 50, dotée de tubes sur batterie. « J'ai bien digéré Colette et Ronald » me dit-il en souriant, faisant ainsi référence à mon article sur la radio portative Philips et à celui de Ronald Dekkers sur la radio Pye P87BQ. Comme on voit, les lecteurs d'Elektor continuent de consommer les tubes sur batterie. C'est peut-être parce qu'ils utilisent une tension anodique « hypocalorique ». Pour moi, les types DF et DL sont des pièces qui n'ont rien perdu de leur attrait.

État : en attente d'une bonne rénovation.

Déclencheur 05 : radio portable Pye P87BQ, Elektor 5/2015.



Objet 06 : pH-mètre d'Electrofact

Beaucoup d'électrotechniciens ont touché à l'électronique médicale. Nombreux sont les stagiaires Elektor qui ont travaillé par la suite pour le fabricant Pie Medical, installé près de notre laboratoire. Qui dit médecine dit aussi chimie et physique. Vu sa taille, je suppose que ce pH-mètre issu de l'ère des tubes provient de l'environnement sécurisé d'un laboratoire d'analyse. Electrofact est une entreprise néerlandaise qui a disparu. L'appareil de mesure qui m'est offert est complet, il comporte la verrerie d'origine ainsi que quelques capteurs. La documentation a été égarée ce qui ne facilite pas son utilisation. En tout cas, rien à lui reprocher du point de vue électrique.

État : nécessitera plus de recherche et de restauration.

Déclencheur 06 : analyseur de pH sanguin/O₂/CO₂ Radiometer PHM22 / PHA928a₂, Elektor 9/2013.



Prêt à parler de vos trésors dans un prochain article Rétronique ?

Courriel : editor@elektor.com ; **objet :** *Trigger [nn]*.



(presque) tout ce que vous avez toujours voulu savoir sur...

la différence entre immotique et domotique

Les maisons connectées sont à la mode de nos jours. Nous avons demandé à Emmanuel François de nous en dire plus sur son expertise : les bâtiments intelligents.

Q Y a-t-il une grande différence entre la domotique et l'immotique ?

R Il y a deux différences majeures : une maison intelligente contient moins de dispositifs techniques qu'un bâtiment intelligent ; en outre, dans le bâtiment intelligent, l'efficacité est beaucoup plus importante. Toutefois, une autre différence est que dans ces bâtiments, les solutions informatiques locales priment. La GTB (**g**estion **t**echnique de **b**âtiment, *Building Management System*) est en grande partie « analogique » et pas nécessairement interactive ou contrôlable à distance. Une maison intelligente est plutôt entièrement « numérique » et contrôlable à distance par l'internet. L'appli « Home » récemment publiée (disponible sur la plupart des appareils Apple et Android) permet de commander les systèmes et les appareils connectés à la maison, c'est une « preuve » de cette différence. Ce qui se produit dans les maisons intelligentes maintenant suivra dans les bâtiments intelligents. La domotique est une sorte de précurseur de ce qui attend les bâtiments intelligents.

Q Les protocoles (logiciels) sont-ils semblables à ceux utilisés en domotique ?

R D'abord, « intelligent » suggère l'incorporation de solutions informatiques, dans l'immotique et la domotique. Par conséquent, dans les deux cas, vous aurez besoin d'une épine dorsale IP (*backbone*). Les protocoles TCP/IP sont fréquemment choisis, en raison de leur compatibilité avec l'internet. C'est aussi dû aux exigences des bâtiments intelligents : flexibilité et modularité. La solution la plus logique est la liaison sans fil. On a donc recours aux protocoles standardisés pour le sans-fil comme le Wi-Fi et le Bluetooth (LE). Aujourd'hui, la plupart des systèmes d'éclairage à LED par ex. emploient le Bluetooth LE. Ces systèmes sont généralement connectés à une épine dorsale IP. Pour les réseaux locaux, ZigBee, EnOcean [1] et Thread [2] sont les plus populaires. KNX est un protocole bien connu, mais il n'est pas vraiment utilisé dans des applications sans fil. Il y a naturellement d'autres protocoles, mais généralement le protocole choisi est un protocole ouvert. Vous verrez rarement des maisons et bâtiments intelligents qui utilisent un protocole propriétaire.

Q Quels sont les avantages de l'immotique ?

R En premier lieu, l'efficacité ! Efficacité énergétique aussi bien qu'efficacité de la gestion de l'espace. En France, environ 50% des bureaux disponibles dans un bâtiment sont inutilisés à cause du télétravail. Réduire de 10% la surface de bureaux dans un immeuble de 10 000 m² permet d'économiser jusqu'à 1 000 000 € par an, l'utilisation des espaces de travail doit donc devenir flexible. Deuxièmement, un environnement de travail bien adapté, confortable, avec de l'air sain et un bon éclairage peut améliorer la productivité des employés. Et troisièmement il y a un impact sur la gestion des biens. La GTB permet d'optimiser la commande du bâtiment.



Liens

- [1] <https://en.wikipedia.org/wiki/EnOcean>
- [2] [https://en.wikipedia.org/wiki/Thread_\(network_protocol\)](https://en.wikipedia.org/wiki/Thread_(network_protocol))
- [3] <http://ovgrealestate.com>
- [4] <https://flexom.bouygues-immobilier.com>
- [5] https://fr.wikipedia.org/wiki/Haute_qualité_environnementale
- [6] www.breeam.com

Q Actuellement quel est le meilleur exemple d'immatique ?

R Mon préféré est le bâtiment *The Edge* à Amsterdam [3]. Il est très efficace, en énergie et en gestion d'espace. Au départ, le bâtiment devait compter 100 000 m². Grâce aux solutions intelligentes de gestion de l'espace, il ne fait finalement que 60 000 m², une réduction de presque 50%, donc une diminution significative des coûts. Autre aspect : il semble que les bâtiments environnants ont perdu 40% de leur valeur à cause de la conception efficace de *The Edge*.

Du côté maison intelligente, la société française de BTP Bouygues construit à présent environ 10 000 maisons entièrement connectées par an en utilisant « Flexom » [4]. Elle peut le faire sans surcoûts grâce aux liaisons sans fil et au protocole EnOcean. L'installation de systèmes sans fil est peu coûteuse. De plus, Bouygues utilise la communication sans contact (NFC) pour la mise en service des systèmes : il faut seulement 20 min par maison pour la configuration.

Q Quand cette technologie parviendra-t-elle à maturité ?

R La clé c'est la confiance que l'utilisateur place dans la technologie et les solutions. C'est pourquoi nous avons travaillé sur une norme pour les bâtiments intelligents, appelée *Ready2Services*, qui peut être intégrée dans les normes comme HQE [5] et BREEAM [6]. Puisque nous travaillons dans un environnement informatique en perpétuelle évolution, cette norme assure aux investisseurs que la GTB restera à jour et résistera à l'épreuve du temps.

Q L'intégration entre les maisons intelligentes et les voitures intelligentes, c'est pour quand ?

R Je pense que dans 15 ans il n'y aura presque plus de voiture avec un moteur à combustion sur la route. Je suis sûr que d'ici 2025, un pays comme la Suède décidera même d'interdire les moteurs de combustion. Ainsi, presque tous les véhicules seront entièrement électriques. La batterie

et l'ensemble du système électrique de la voiture seront reliés au système d'énergie de votre maison intelligente.

Q Qu'est-ce qui est déterminant pour la recherche à l'heure actuelle ?

R Bien, nous avons parlé de maisons et voitures intelligentes. Le point commun et important dans les deux cas c'est le cerveau. L'intelligence artificielle mène l'innovation, parce que l'utilisateur exige que les voitures et maisons intelligentes s'adaptent à ses besoins. C'est une grande question pour toutes les parties prenantes. Tous doivent être à la pointe en matière d'IA ou ils manqueront le train. ◀

(160259 version française : Thierry Destinobles)

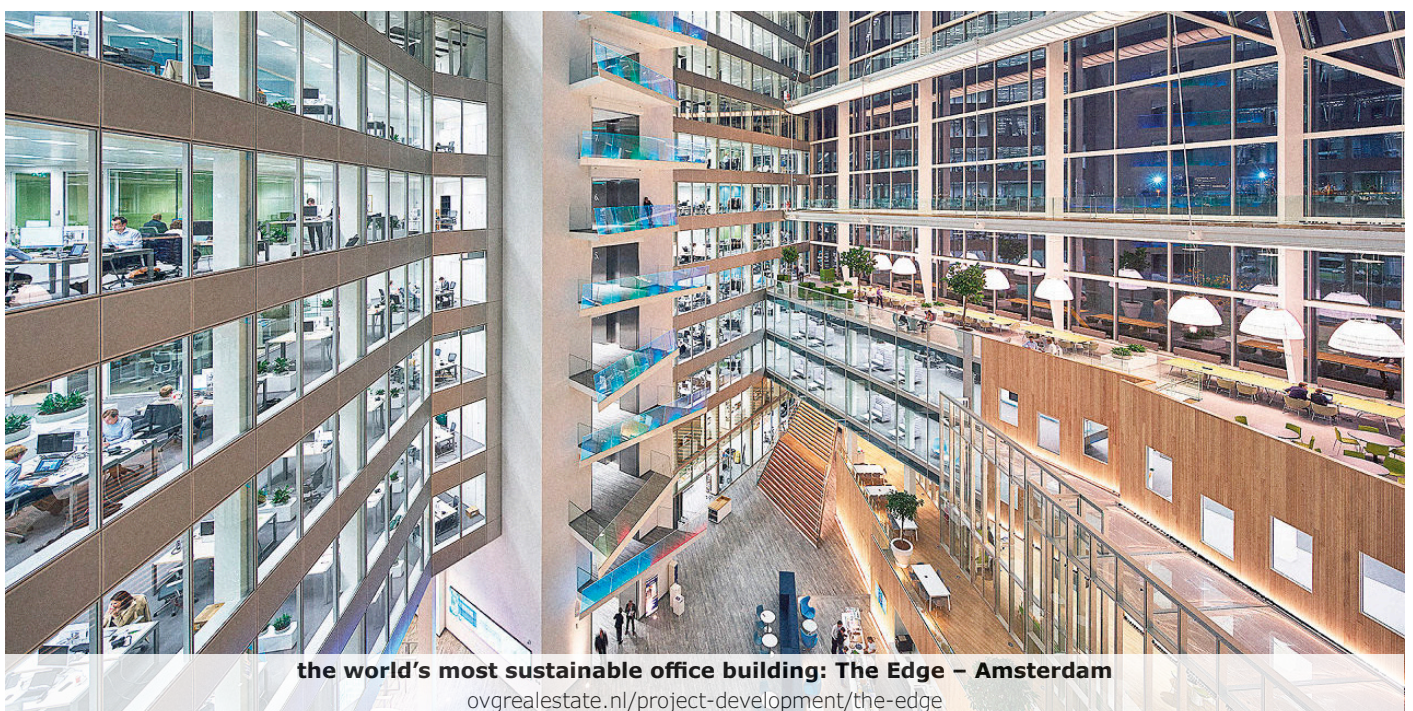


Qui est Emmanuel François ?

Emmanuel a toujours travaillé dans l'immatique. En 2014, alors âgé de 55 ans, il devient Président de la *Smart Building Alliance for Smart Cities* (SBA), qu'il a fondée avec Serge Le Men de *Newron System* en 2012. Cette alliance qui compte 140 membres guide toutes les parties impliquées avec la transition des bâtiments/villes

intelligents vers le monde numérique. La SBA a également produit *Ready2Services* et *Ready2Grid*, deux prérequis nécessaires pour les bâtiments/villes intelligents.

Il est également responsable des ventes d'EnOcean GmbH pour l'Europe de l'Ouest. Il a aussi travaillé dans l'univers de l'électrotechnique en tant que directeur général ou responsable des ventes successivement chez Wieland Electric France, CAPRI (groupe Cooper), Soulé (groupe ABB) et Merlin Gerin (Schneider Electric).



the world's most sustainable office building: The Edge – Amsterdam

ovgrealestate.nl/project-development/the-edge

WaterDeck

analyse de la consommation d'eau des villes intelligentes

Waldemir Cambiucci (Brésil)

La pénurie d'eau frappe de plus en plus de régions – plusieurs villes du Brésil et l'état de Californie par exemple. L'impact sur les usagers domestiques et l'industrie devient critique. Même si les autorités travaillent sur des solutions pratiques pour résoudre ce problème et apprennent aux usagers à économiser l'eau, nous manquons toujours d'information sur notre consommation d'eau en temps réel. Il faut une acquisition de données au travers d'une interface web.

Le projet WaterDeck est une solution pour la surveillance et l'analyse de la consommation d'eau des villes intelligentes (et plus !), qui utilise les technologies Freescale et ARM. WaterDeck collecte les données de nombreux dispositifs et capteurs, comme des jauges de niveau, des réservoirs, des débitmètres et des commandes de pompes à eau. Chacun d'eux est connecté à une carte FRDM-K64F [1]. Les données des capteurs sont collectées via des connexions à l'internet et utilisées pour fournir en temps réel aux usagers domestiques et industriels des tableaux de bord qui montrent les réserves et la consommation d'eau. Ces informations sont accessibles sur des sites web, des API et des applis pour tablettes et ordiphones. Le nuage (*cloud*) permet plus encore : à savoir regrouper l'énorme masse de données accumulées au fil du temps non seulement pour examiner l'historique, mais aussi pour créer des rapports et faciliter les analyses ultérieures.

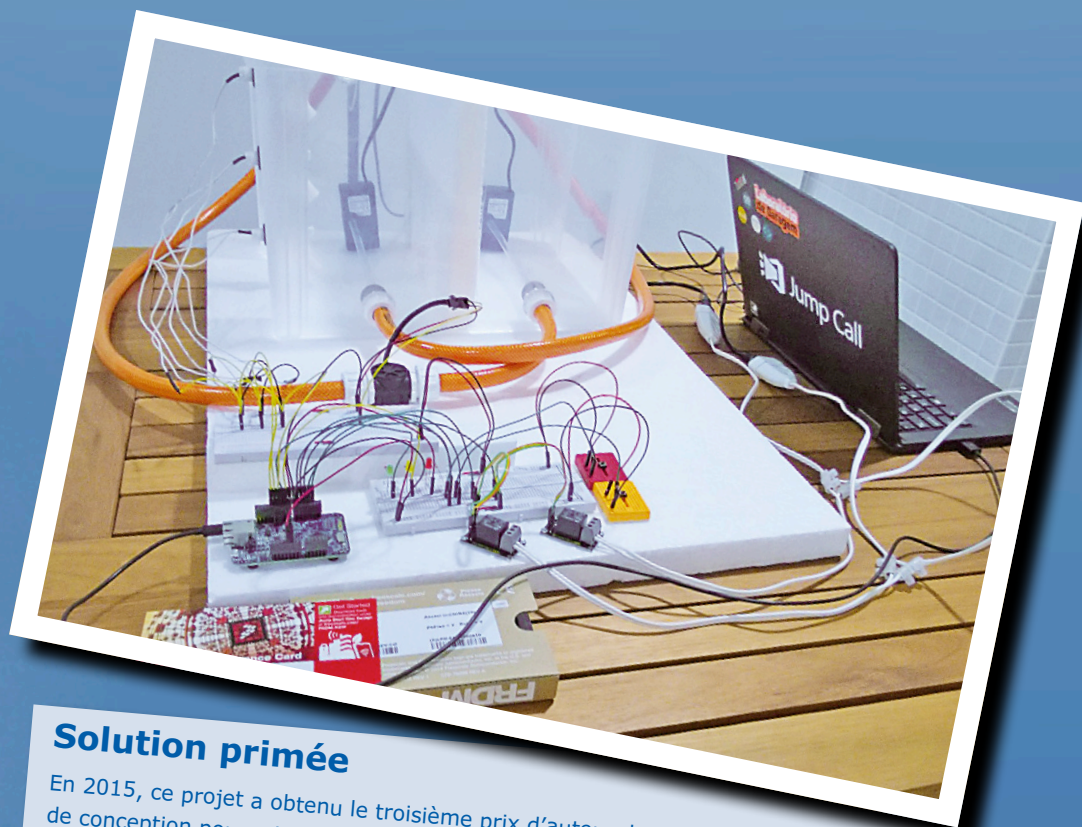
Avec l'Internet des Objets (IdO), il est important de rappeler que les données collectées et les scénarios associés sont aussi importants que les capteurs, microcontrôleurs et dispositifs qui en assurent le fonctionnement. La combinaison bien choisie d'un solide microcontrôleur et d'une collecte de données de bonne qualité peut créer une formidable opportunité pour comprendre l'environnement qui nous entoure, avec de vrais scénarios d'IdO et des bénéfices substantiels pour les

usagers. Cette même solution s'applique ainsi à la fois aux usagers domestiques et commerciaux qui peuvent comparer les données réelles de consommation avec celles des factures de leurs fournisseurs.

Avec une population mondiale qui frisera les neuf milliards en 2050 et une consommation d'eau en plein essor sur cette période, le projet WaterDeck peut être d'un grand secours, non seulement en temps de crise, mais aussi comme outil permanent d'optimisation de la consommation et de surveillance de l'eau, partout et n'importe quand dans le monde.

Modules de base

Pour ce projet dédié à la gestion de l'eau pour les particuliers et les petites entreprises, nous avons choisi la carte FRDM-K64F (**fig. 1**) et le modèle de conception « Acquisition de Données avec Interface Web ». La FRDM-K64F a été conçue par Freescale en collaboration avec mbed pour prototyper toutes sortes de dispositifs, en particulier ceux dans la gamme de prix et de taille des Cortex-M4. Elle se présente comme une carte de développement avec connecteurs pour plaques à bandes ou platines d'essais, et comporte un programmeur Flash USB. Avec son connecteur compatible Arduino, la carte FRDM-K64F fournit aux développeurs et ingénieurs une plateforme simple, robuste et extensible.



Solution primée

En 2015, ce projet a obtenu le troisième prix d'auteur du concours de conception pour microcontrôleurs ARM Keil, pour la création d'un dispositif domestique ou industriel intelligent basé sur un processeur ARM Cortex-M. La compétition a attiré plus de 700 candidatures. Waldemir résume son expérience ainsi : *Le concours de conception pour microcontrôleurs ARM organisé en coopération avec Elektor était une magnifique occasion pour améliorer mes compétences dans l'IdO, notamment en utilisant une carte récente comme la Freescale FRDM-K64F. Ce fut aussi un bon moyen pour rencontrer de nouveaux amis et passionnés du monde entier via la communauté ARM. Enfin ce fut un honneur d'être désigné comme l'un des gagnants de ce concours et j'attends déjà le prochain concours ARM Keil.*

Lors du concours 2015 de conception pour microcontrôleur ARM (voir encadré **Solution Primée**), nous avons sélectionné un des quatre modèles de conception disponibles. Nous avons choisi le modèle « Acquisition de Données avec Interface Web » qui utilise un composant réseau intergiciel pour afficher sur une page web les données provenant d'un accéléromètre et d'un magnétomètre. Un intergiciel (ou logiciel médiateur) est une couche de logiciel qui sert d'interface entre deux parties de code informatique distinctes pour les faire fonctionner ensemble, telles qu'un programme applicatif et son environnement fonctionnel. Les utilisateurs peuvent modifier la fréquence d'enregistrement des données avec des scripts CGI et JavaScript.

Le but (**fig. 2**) est de collecter de multiples informations à partir de capteurs, jauges de niveau, pompes, débitmètres, etc., et de créer dans une interface web des notifications *push* vers l'utilisateur final. Il est ainsi possible d'afficher le tableau de bord sur n'importe quel terminal, comme tablettes, ordinateurs ou ordinateurs de bureau, depuis un jeu de données hébergé dans le nuage.

WaterDeck en action

Nous avons créé un environnement de simulation complet pour ce projet, avec deux réservoirs qui simulent la consommation d'eau. Le premier récipient est un réservoir d'eau domestique

ordinaire équipé de trois capteurs de niveau de marque ICOS [2]. Ces capteurs sont connectés à la FRDM-K64F via des broches numériques standard, le signal est lu comme haut ou bas. Nous avons installé un débitmètre sur la conduite d'alimentation, en amont de la pompe principale. Ainsi, dès que l'eau s'écoule dans le réservoir principal, on peut lire le débit de l'eau qui pénètre dans le réservoir domestique en gallons par heure ou litres par minute.

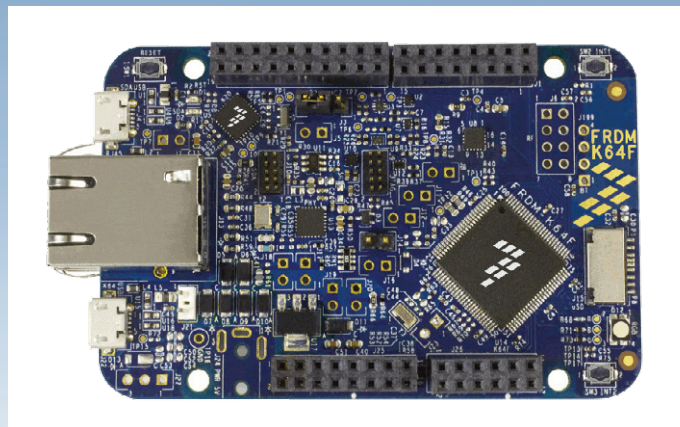


Figure 1. La carte Freescale FRDM-K64F.

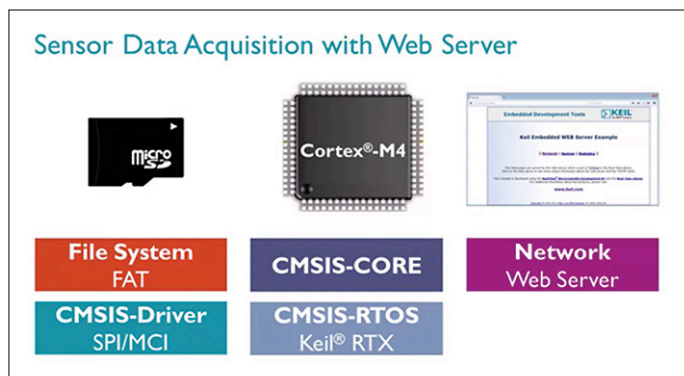


Figure 2. Acquisition de données des capteurs avec serveur web.

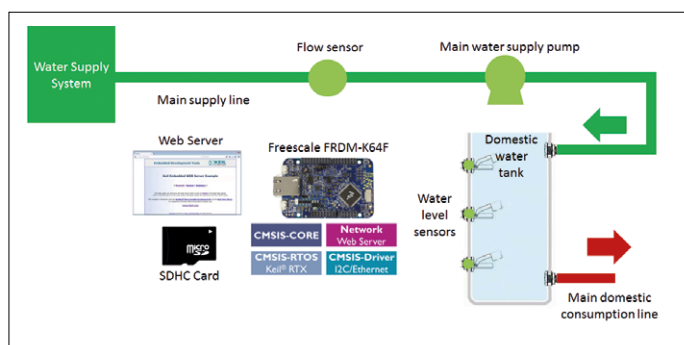


Figure 3. Utilisation de la carte Freescale FRDM-K64F pour commander et surveiller la consommation d'eau pour usage domestique avec consultation des données sur l'internet.

Enfin une autre pompe simule simplement la consommation issue du réservoir domestique. Quand cette pompe est actionnée, l'eau est aspirée du réservoir principal et refoulée dans le réservoir externe.

La carte Freescale FRDM-K64F, pièce maîtresse de cette simulation, commande les nombreuses entrées/sorties numériques du système. Une fonction remarquable de cette carte est sa capacité à exporter vers le web toutes les données collectées, ce qui permet à de nombreux utilisateurs potentiels d'y accéder. Pendant la simulation, les données sont sauvegardées sur une carte SDHC pour archiver l'historique de tous les capteurs et

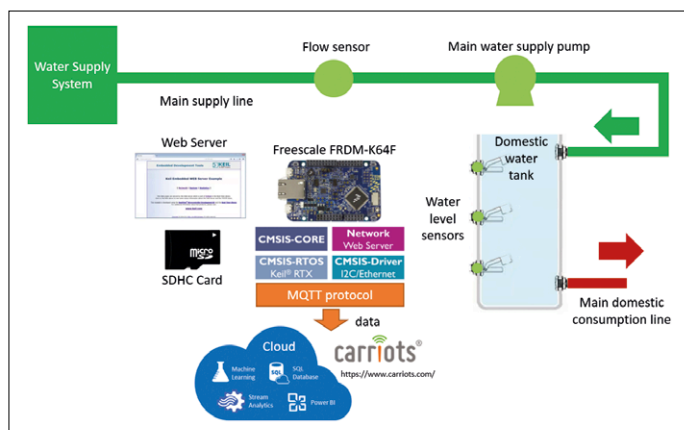


Figure 4. Diagramme du circuit principal, avec l'adaptateur pour le protocole MQTT qui connecte Freescale aux différents tableaux de bord ou plateformes IdO hébergés dans le nuage.

composants du système. On peut utiliser ces données pour alimenter des sites ou autres systèmes sur le web.

Connexion au nuage

Toujours avec le modèle de conception « Acquisition de Données avec Interface Web », ce projet peut être considéré comme un exemple pour la connexion de la plateforme Freescale à différents tableaux de bord pour l'IdO dans le nuage, comme Windows Azure et Amazon Web Services, ou à des solutions SaaS (« Software As A Service ») comme la plateforme IdO Carriots [3].

Pour cela, il faut un adaptateur ou un client pour le protocole MQTT afin de connecter le système à différents fournisseurs et d'envoyer les messages à rassembler. MQTT (« Message Queuing Telemetry Transport ») prend en charge beaucoup de scénarios possibles pour la messagerie dans les projets de l'IdO. Nous évaluons en ce moment la solution Mosquitto pour nous connecter à Freescale dans nos futurs projets. Mosquitto est un courtier de messages à code source ouvert (licence BSD) qui met en œuvre le protocole MQTT versions 3.1 et 3.1.1. Plus d'informations en [4]. Le diagramme de la **figure 3** présente une vue d'ensemble de notre projet.

Conclusions

Comme le montre cet article, ce projet est conçu pour afficher et surveiller la consommation d'eau domestique. Il est basé sur des réservoirs, des jauges, des débitmètres et actionneurs pour pompes à eau d'usage courant. Il utilise une connexion à l'internet standard pour fournir en temps réel un tableau de bord sur la disponibilité de l'eau au domicile de l'utilisateur, accessible sur des sites web, des API et des applis sur tablettes et ordinateurs qui traitent ces informations.

Pendant le projet nous avons évalué et exploité beaucoup de fonctions importantes de la carte Freescale FRDM-K64F, qui s'est révélée très polyvalente pour les applications de l'IdO. L'exercice a aussi été l'occasion d'expérimenter l'environnement de développement Keil uVision5 avec lequel le projet a été configuré et testé.

Fonctions principales utilisées par ce projet :

- fichier journal sur carte SDHC ;
- interruptions pour l'acquisition des données du débitmètre ;
- broches pour les vannes d'entrée et sortie, ainsi que pour des LED et capteurs de divers types ;
- relais pour commander des circuits externes 220 V, comme des pompes à eau ;
- boutons-poussoirs pour entrées manuelles de l'utilisateur.



(150502 – version française : Denis Lafourcade)

Liens

- [1] www.nxp.com/products/software-and-tools/hardware-development-tools/freedom-development-boards/freedom-development-platform-for-kinetis-k64-k63-and-k24-mcus:FRDM-K64F
- [2] www.icos.net
- [3] www.carriots.com
- [4] <http://mosquitto.org>



sigfox

donner vie aux objets

Donner la vie à des millions d'objets, c'est l'objectif de Sigfox qui a créé un réseau capable de connecter à l'internet des milliards d'objets. Ce réseau est aujourd'hui présent dans 29 pays, donc accessible à environ 470 millions de personnes.

Les entreprises qui veulent développer des services et des produits dans le domaine de l'Internet des Objets disposent aujourd'hui d'une large gamme de solutions, grâce aux toujours plus nombreux partenaires de Sigfox. Partons en voyage dans le monde de Sigfox.



Mariline Thiebaut-Brodier (Elektor)

Sur le terrain

Fondée en 2010 par Ludovic Le Moan et Christophe Fourtet, cette entreprise française est installée à Labège, près de Toulouse, au sein de l'*IoT Valley*. Elle a créé un réseau cellulaire à bas débit, sur des bandes de fréquence sans licence, dédié aux communications entre machines (M2M) et à l'IdO.

Le réseau Sigfox présent dans 29 pays (essentiellement européens) devrait s'étendre à une soixantaine de nouveaux pays d'ici 2018 (voir [1]). Sigfox installe des antennes, mais se déploie également grâce à des alliances avec des opérateurs de télécommunications comme SFR (France) ou Albertis Telecom (Espagne). Les clients de ces opérateurs bénéficient de la connectivité Sigfox en plus des services existants (4G, Wi-Fi...). Cette société est aussi entrée dans le monde des compteurs connectés pour le télérelevé et le suivi des consommations à distance (eau, électricité) ou encore la surveillance des réseaux de distribution (détection de fuite d'eau, courbe de charge électrique). Sigfox s'est allié avec des fournisseurs de services en nuage comme le *cloud* Microsoft Azure. Les données collectées par les objets Sigfox sont transmises à la plateforme Azure pour y être valorisées.

Aujourd'hui les secteurs demandeurs de connectivité sont divers et variés : ville intelligente, logistique, agriculture, sécurité (maisons, entreprises), véhicules, environnement... Le suivi à distance (ou pistage GPS) d'objets, d'animaux ou de personnes est un secteur en plein essor. La société Sensolus propose le système *Chickenrun* qui permet de calculer les temps de parcours des volailles. Les Français de *HideSeek* (voir **encadré**)

commercialisent différentes balises autonomes qui renvoient la position du sujet/objet connecté, mais aussi la température et la pression atmosphérique.

Technologie

Sigfox a déployé un réseau à bas débit (100 bits/sec) qui transmet des messages courts sur de grandes distances. Le record de distance en production est établi à 1.024 km entre un objet et une antenne, mais dans la réalité on oscille entre 2 et 130 km en fonction de la topologie du terrain. Les objets connectés consomment alors très peu d'énergie pour la transmission radio qui est bidirectionnelle et à bande ultra-étroite (UNB, *ultra narrow band*). L'émetteur module la porteuse du signal émis par déplacement de phase à deux états (BPSK, *binary phase-shift keying*). Le récepteur écoute une bande très étroite du spectre pour détecter les messages. Ensuite les messages reçus et valides sont envoyés dans le nuage Sigfox ou à la plateforme choisie par le propriétaire de l'objet.

Le réseau est bidirectionnel, mais uniquement sur demande de l'objet. Il faut comprendre que dans le protocole Sigfox, le réseau est au service de l'objet, et pas l'inverse. L'objet demande donc lui-même une « mise à jour » au réseau, qui lui répond si besoin. Le reste du temps, il est « sourd ».

Cette technologie est pensée et optimisée pour l'utilisation de la batterie, il est prévu que certains objets déjà en production pourront tenir jusqu'à 10 ans sur de simples piles. En effet, il est très simple de calculer la consommation de courant d'un objet Sigfox car elle est constante, contrairement à ce qui se passe avec d'autres technologies. Cela facilite grandement la mise en œuvre. Bien évidemment, la durée de l'autonomie est liée au nombre de messages qu'on a besoin d'envoyer par jours (max. 140 messages par objet et par jour). Chaque message contient max. 12 octets de données utiles.

Tableau 1. Répartition de la couverture Sigfox à travers le monde.

Zone de communication radio	Région du monde
RCZ1	Europe (868 MHz, ETSI 300-220), Oman, Afrique du Sud
RCZ2	États-Unis (902 MHz, FCC part 15), Mexique, Brésil
RCZ3	Japon (923 MHz, ARIB STD-T108)
RCZ4	Australie, Nouvelle-Zélande, Singapour, Taiwan, Hong Kong, Colombie, Argentine (920 MHz, ANATEL 506, AS/NZS 4268, IDA TS SRD)

L'entreprise Sigfox se distingue d'un point de vue technologique, mais également commercial des autres initiatives de type « LPWAN ». Sigfox opère avec une approche très ouverte pour son protocole puisque tous les fabricants de semi-conducteurs peuvent intégrer gratuitement le protocole Sigfox à leurs produits. Contrairement aux technologies concurrentes qui obligent à reverser des frais de licences sur chacune des puces produites, ce qui entraîne un monopole de fait. Sigfox ne se rémunère que sur les abonnements à son service. Cette approche devrait stimuler la concurrence entre les fabricants de modules et tirer les prix des objets vers le bas. Les modules les moins chers pour entrer dans le monde Sigfox sont proposés aux alentours de 2 €.

La structure technique du réseau est elle aussi très différente des autres technologies, ce qui contribue grandement à son essor si rapide dans le monde entier. Elle semble nécessiter un nombre d'antennes bien plus faible que ses concurrents pour couvrir une zone équivalente (grâce à sa longue portée), et avoir une capacité à connecter un nombre d'objets bien plus élevé.

Mise en pratique

Pour envoyer des messages sur le réseau Sigfox, il faut du matériel compatible Sigfox, un module ou un kit de développement. En premier lieu, il faut choisir la zone dans laquelle vous travaillerez. Le **tableau 1** détaille ces différentes zones, établies principalement en fonction des normes à respecter en matière de transmission et de spectre radio.

Ensuite il faut un émetteur-récepteur qui parle Sigfox ou bien un module de transmission qui en contient un ainsi qu'un microcontrôleur. Le **tableau 2** donne quelques références des circuits de ce type. Les modules sont pilotés soit directement avec des commandes AT (communication à bas niveau) sur une interface série (UART), soit via une interface de type API (bibliothèques).

Si vous êtes un utilisateur chevronné d'Arduino ou de Raspberry Pi, vous n'aurez pas de mal à trouver une carte d'extension Sigfox pour votre carte préférée. Les autres pourront se tourner vers une carte de développement autonome. Nous avons rassemblé dans les **encadrés** ci-après une sélection de cartes et kits actuellement disponibles. La plupart ont obtenu le label « Sigfox Ready » après avoir passé une certification dédiée et gérée par Sigfox. La certification coûte 2.500 € pour un objet. À cette occasion, la puissance émise par l'objet testé est mesurée afin de le classer selon une échelle basée sur des normes ETSI ou FCC (voir **tableau 3**). La classe U0 garantit une bonne couverture de l'objet.

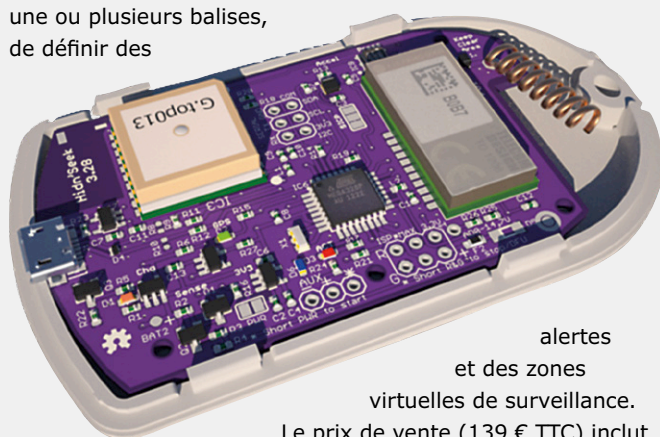
Revenons aux kits. La bonne surprise est que souvent un accès d'un an au réseau Sigfox est inclus, de quoi expérimenter dès que vous aurez ouvert le colis. Le coût d'un accès au réseau seul varie entre 1 et 20 € par an.

Sigfox facilite la vie des jeunes pousses et propose à celles qui veulent connecter un produit des abonnements au réseau ainsi que des outils gratuits. Le programme « startup » y est même dédié afin d'accélérer tous les projets qui germent.

Si votre kit de développement ou système sur puce (SoC) est vendu sans abonnement au réseau Sigfox, vous devrez enre-

HideSeek – carte compatible avec l'EDI Arduino, prête à l'emploi

La société HideSeek commercialise le pisteur **HIDNSEEK ST1-A/B** [4], il s'agit d'une balise GPS autonome (sans carte SIM) et rechargeable. Une application mobile ainsi qu'un site internet permettent de suivre une ou plusieurs balises, de définir des



alertes et des zones virtuelles de surveillance. Le prix de vente (139 € TTC) inclut un an d'abonnement au réseau Sigfox, renouvelable chaque année (19,90 €).

Le plus intéressant c'est que les API ainsi que le micrologiciel de cet objet sont ouverts. Il se transforme donc en carte de développement que l'on peut adapter à ses propres usages en ajoutant des capteurs. Le fabricant met à disposition sur GitHub ([5] et [6]) tous les outils nécessaires pour reprogrammer l'objet avec l'EDI Arduino et un câble USB. La carte embarque un µC ATmega328P à 12 MHz, une passerelle Sigfox TD1207 et un module GPS PA6H de GlobalTop.

Caractéristiques

- Localisation par GPS
- Capteur de mouvement à 3 axes
- Capteurs de température et pression
- Mise à jour du micrologiciel par interface USB
- Géorepérage (*geofencing*) avec appli Apple iOS et Android
- E/S d'usage général (GPIO) libres pour des extensions (I2C, série, ISP, E/S analogique/numérique)
- Batterie LiPo de 500 mAh, rechargeable et surveillée (plusieurs mois d'autonomie)


gistrer votre objet sur une plateforme pour le suivre, voir la liste en [2]. Certaines sont réservées à des applications particulières comme le suivi de déchets ou d'abeilles. D'autres sont généralistes et utilisables par tous comme Microsoft IoT, Amazon AWS ou Thethings.io.

Les informations suivantes permettent d'identifier chaque objet Sigfox :

- Un identifiant unique, SIGFOX ID (*read-only device unique identifier*). En général cet ID apparaît sur une étiquette collée sur l'objet. Il est non modifiable et indispensable pour enregistrer un objet sur le réseau (attention à ne pas le perdre). Ex : 16B87.
- Un code PAC (*Porting Authorization Code*). Ce code est également indispensable pour enregistrer un objet, mais il change si l'objet change de propriétaire. Ex : CF14xxxx-xxx.
- Une clé de sécurité cachée, chiffrée AES128.

La liste de produits proposés ici n'est pas exhaustive. Elle

montre simplement que l'offre pour atterrir sur la planète Sigfox est variée. Il n'y a que l'embarras du choix. En outre Sigfox a créé un site dédié aux développeurs (voir [3]) qui contient des tutoriels ainsi que des liens vers diverses ressources.

En résumé, la création d'un objet connecté est facilitée par les modules disponibles. L'accès au réseau Sigfox est simple et peu onéreux par rapport à Lora par exemple. Vous avez donc les outils nécessaires pour réaliser votre premier objet connecté. Est-ce que ce sera pour pister votre chien qui se sauve régulièrement ou pour récupérer des données environnementales dans vos vergers ? Nous ne doutons pas que vous trouverez des utilisations originales du réseau Sigfox. N'oubliez pas de partager vos expériences avec nous. 



(160167)

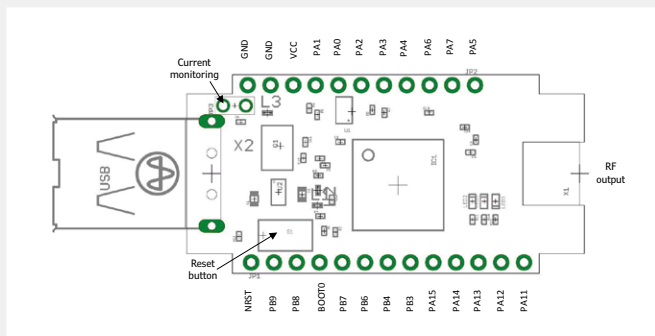
Tableau 2. Circuits intégrés pour la communication Sigfox, utilisés dans différents shields, kits de développement et cartes d'extension.

Fabricant	Référence	Type	Pour passer à la pratique
Atmel	ATA8520D-GHQW	émetteur/récepteur + AVR µC à 8 bits	
	ATA8520E-GHQW	émetteur/récepteur + µC AVR à 8 bits	kit d'évaluation : ATA8520-EK4-E (RCZ1) / ATA8520-EK1-F (RCZ2) ; carte d'extension pour Xplaine Pro : ATA8520-EK6-E (RCZ1) / ATA8520-EK3-F (RCZ2)
	ATA8520-GHQW	émetteur + µC AVR à 8 bits	
OnSemiconductor	AX-SFEU / AX-SFEU-API	émetteur/récepteur	kit de développement : DVK-SFEU-[API]-1-GEVK
	AX-SFUS / AX-SFUS-API (RCZ2)	émetteur/récepteur	
Radiocrfts	RC1682-SIG	module radio (contrôleur + pile Sigfox)	kit de démonstration : RC1682-SIG-DK
STMicroelectronics	S2-LP	émetteur/récepteur	kit de développement : STEVAL-FKI868V1 ou STEVAL-FKI915V1 (carte NUCLEO-L152RE + S2-LP)
TD Next	TD1204	émetteur Sigfox + récepteur GPS	carte d'évaluation : TD1204 EVB
	TD1205P	émetteur Sigfox + récepteur GPS + antenne + accéléromètre	
	TD1207R/08R	passerelle (émetteur + µC ARM Cortex-M3)	carte d'évaluation : LGA25 EVB
	TD1508 (RCZ2)	passerelle (émetteur + µC ARM Cortex-M3)	carte d'évaluation : LGA25 EVB
Telit	LE51-868 S	émetteur/récepteur	kit d'évaluation : LE51-868S EVK
TexasInstruments	CC1120 / CC1125 / CC1190	émetteur/récepteur	kit de développement : MSP430F5529 LaunchPad ou CC1120 CC1190 BoosterPack ou CC1125 BoosterPack
Wisol	WSSFM10R1	émetteur/récepteur	carte d'extension : BRKWS01 de SNOO

Tableau 3. Classes de liaison montante (de l'objet vers le réseau) en fonction de la puissance émise.

	Classe U0	Classe U1	Classe U2	Classe U3
ETSI	14 dBm ±2 dB	12 dBm > P > 7 dBm	7 dBm > P > 0 dBm	< 0 dBm
FCC	22 dBm ±2 dB	20 dBm > P > 15 dBm	15 dBm > P > 5 dBm	< 5 dBm

Nemeus – clé USB



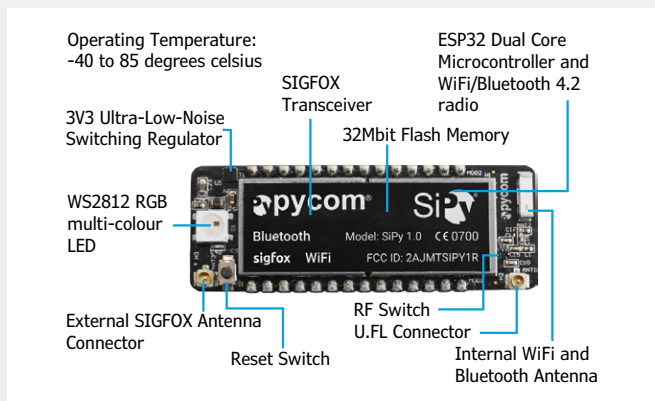
Nemeus propose le **MM002-LS-EU** [7], un modem bi-mode (LoRa/Sigfox) qui contient un µC STM32L151X et un émetteur-récepteur RF SX1272. Le tout est animé par le système d'exploitation Contiki. Le connecteur JTAG permet

de charger son code et les ports I2C, SPI, UART, GPIO, CA/N, CN/A de dialoguer avec des capteurs.

Pour expérimenter plus rapidement avec le réseau Sigfox, vous pouvez vous tourner vers la version prête à l'emploi du MM002-LS-EU : la clé **USB MK002-LS-EU** [8]. Il suffit de la connecter à un ordinateur (Windows ou Linux) pour lui envoyer des commandes AT. Une appli JAVA est fournie pour effectuer des tests. Il est également possible d'alimenter directement la clé (broche Vcc) et d'accéder à l'UART (broches PA2 et PA3).

À l'écriture de ces lignes, Nemeus annonçait la sortie de l'**IoT Smart Sensor**, une carte pour expérimenter avec les liaisons sans fil (Sigfox, LoRa, Bluetooth, GPS/GNSS...). Cette carte compatible avec l'EDI Arduino comporte un accéléromètre, un capteur de température, un capteur de pression, mais également une LED RVB programmable par l'utilisateur.

PyCom – programmation en Python



La société PyCom propose la carte de développement **SiPy MicroPython Wi-Fi/Bluetooth/Sigfox** [9]. Elle est compatible MicroPython et conçue spécifiquement pour les applications de l'IdO. Elle repose sur le système sur puce (SoC) Espressif ESP32 doté d'un µC à double cœur, du Wi-Fi (portée jusqu'à 1 km), du Bluetooth (classique et BLE), de Sigfox et de 512 Ko de mémoire RAM, sur une surface réduite (55 × 20 × 3,5 mm). L'EDI propre à PyCom, *Pymakr*, facilite la programmation.

Nota : il faut une antenne externe pour le réseau Sigfox. Une utilisation sans antenne peut endommager la carte, voir

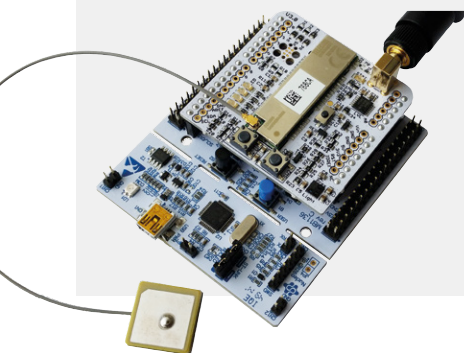
réf. 125-9535 chez RS. L'antenne Wi-Fi et Bluetooth est interne. Pour accéder facilement aux broches de la SiPy, vous pouvez utiliser une *carte d'extension universelle Pycom*. Elle est équipée d'un connecteur micro-USB pour l'alimentation et les communications série, d'un chargeur de batterie LiPo, d'une protection contre l'inversion de polarité, d'un emplacement pour carte micro-SD, d'un bouton et d'une LED utilisateur.

Caractéristiques

- 2× UART
- 2× SPI
- I2C
- I2S
- Carte micro-SD
- CA/N à 8 voies, 12 bits
- Jusqu'à 24 GPIO
- LED multicolore RVB WS2812
- Commutateurs de RESET et RF
- Émetteur-récepteur CC1125 de TI
- Version RCZ1/RCZ3 et version RCZ2/RCZ4

PyCom annonce la sortie de la **FiPy WiFi/Bluetooth/LTE CAT M1 / NB1/LoRa/Sigfox** [10] qui, sur le modèle de la SiPy, donne accès à cinq réseaux.

Quicksand – carte de développement STM32



L'entreprise belge Quicksand propose le kit de développement **QW GPS Sigfox** [11] construit autour d'un émetteur-récepteur TD1204 et d'une carte NUCLEO-L152RE (µC STM32 mbed).

Caractéristiques

- Capteurs : GPS, accéléromètre, température, proximité et lumière ambiante
- 2 boutons utilisateur, 1 bouton de RàZ et 4 LED utilisateur
- Compatible Arduino via convertisseur de niveau
- Compatible 3,3 V et 5 V
- Logiciel à source ouverte disponible
- Interface JTAG vers TD1204

ATIM – l’embarras du choix

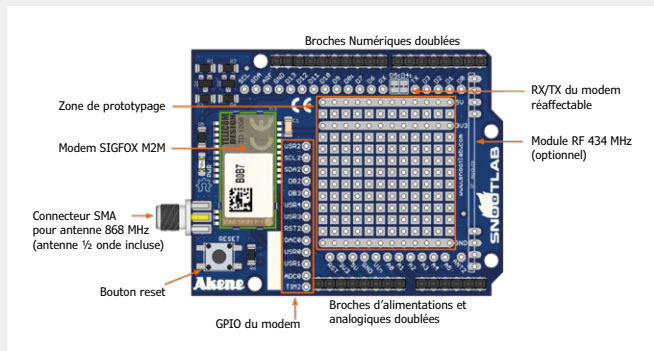


Réf.	Adaptateur	Entrée	Sortie (antenne)
ACW-DUINO	Arduino	GPIO	connecteur SMA ou UFL
ACW-MangOH	ordinateur mangOH	GPIO, UART	connecteur UFL
ACW-MPCIE	mini PCI-Express	UART	connecteur UFL
ACW-RPI	Raspberry Pi	GPIO	connecteur SMA ou UFL
ACW-SDK	USB (<i>starter kit</i>)		connecteur SMA
ACW-USB	clé USB	USB	connecteur SMA
SigBee	XBee	GPIO	connecteur SMA ou UFL
ARM-N8-SIGFOX	Version Sigfox du module radio ARM-NANO Advanced Radio Modem®)		

La société française ATIM propose plusieurs cartes d’extension radio et multimode (Sigfox & Lora) [12]. Le

tableau suivant en donne une vue d’ensemble. La plupart de ces cartes incluent un an d’abonnement au réseau.

SNOOTLAB – Arduino et Raspberry Pi



La société SnootLab propose plusieurs cartes avec la passerelle TD1208 ainsi que les bibliothèques correspondantes [13] :

Akeru 3.3 : carte compatible avec l’environnement Arduino/Genuino. Cette carte est également disponible dans un kit qui contient plusieurs capteurs.

Akene : *shield* pour Arduino/Genuino.

Breakout TD1208 : carte d’extension pour ajouter la connectivité Sigfox à ses projets, sans réaliser un circuit spécifique pour accéder aux signaux de la passerelle.

Foquinha-Pi : *shield* pour Raspberry Pi (toutes versions).

Toutes ces cartes sont proposées avec un an d’abonnement au réseau.

NetTrotter – carte 100% compatible Arduino

Cette société italienne propose la **BiB, Basic IoT Board** [14] : c’est une carte personnalisable (ajout de capteurs de température et d’humidité, lecture de balises NFC, communication Bluetooth Low Energy). Comme elle est

100% compatible Arduino, elle peut accueillir nombre de *shields* Arduino et être programmée avec l’EDI Arduino. Elle comporte un GPS et un accéléromètre/magnétomètre/gyroscope.

Liens

- [1] Sigfox (couverture, partenaires) : www.sigfox.com, partners.sigfox.com
- [2] Plateformes Sigfox pour enregistrer ses objets : partners.sigfox.com/products/platform
- [3] Site Sigfox pour les développeurs : makers.sigfox.com
- [4] Pisteur HIDNSEEK ST1-A : www.hidnseek.fr/hidnseek_st1a
- [5] GitHub pour HidnSeek : github.com/hidnseek/hidnseek
- [6] Tutoriel pour HidnSeek : github.com/Bucknalla/sigfox-hidnseek-tutorial
- [7] MM002-LS-EU : www.nemeus.fr/en/nemeus-mm002-2
- [8] USB MK002-LS-EU : www.nemeus.fr/en/mk002-usb-key

- [9] SiPy : www.pycom.io/product/sipy
- [10] FiPy : www.pycom.io/product/fipy-preorder-shipping-april-2017
- [11] QW GPS Sigfox : lpwan.be/wp/product/qw-gps-sigfox-development-kit
- [12] ACW-DUINO, ACW-MangOH, ACW-MPCIE, ACW-RPI, ACW-SDK, ACW-USB, SigBee, ARM-N8-SIGFOX : www.atim.com/en/products/catalog
- [13] Akeru 3.3, Akene, Breakout TD1208, Foquinha-Pi : snootlab.com/lang-en/72-03-iot-et-sans-fil
- [14] BiB, Basic IoT Board : www.nettrotter.io/index.php/it/ecosystem-it/basic-iot-board

expérimenter avec un transformateur de Tesla quelques composants pour un effet électrisant

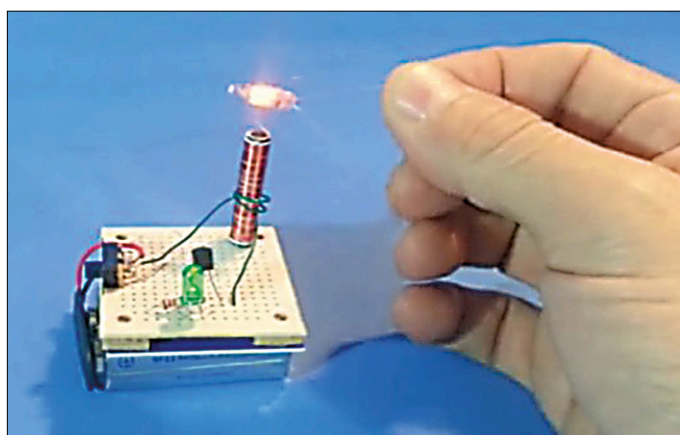
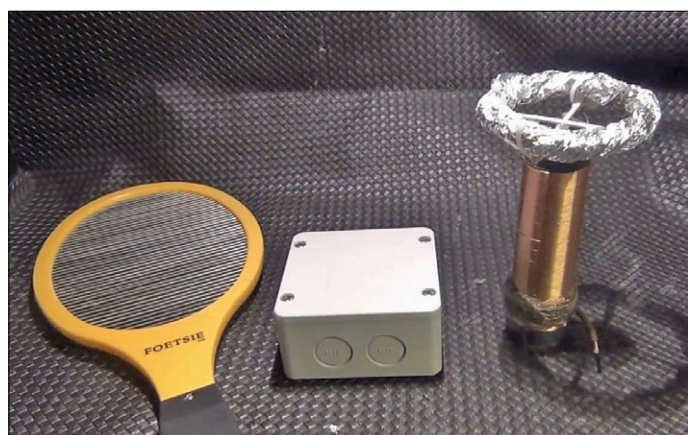
La production d'arcs électriques à l'aide d'un transformateur de Tesla (aussi appelé bobine Tesla) est un sujet souvent abordé dans le magazine, néanmoins il fascine toujours autant, et pas seulement les électroniciens ! Les dispositifs sont petits ou grands, et les plus imposants produisent des arcs de plusieurs mètres. Nous nous contenterons ici d'en décrire quelques-uns qui ne produisent que des arcs modestes ou ne permettent que l'allumage de tubes ou lampes fluorescentes, toutefois ils sont faciles à construire.

Harry Baggen (labo d'Elektor)

Vous avez sans doute déjà vu, en photo ou en vidéo, ce gros appareil d'où semblent s'échapper des éclairs. Il s'agit d'un transformateur de Tesla qui, comme son nom l'indique, est un transformateur, à noyau d'air, qui permet d'obtenir de très hautes tensions. Le secondaire du transformateur utilise la résonance d'un circuit LC, qui est entretenue par la tension au primaire. Par le passé, on utilisait souvent un éclateur à étincelles en parallèle ou en série avec l'enroulement primaire pour

déclencher les oscillations dans le circuit LC, ce qui produisait pas mal de parasites et du bruit. On utilise actuellement un circuit à semi-conducteurs ou à tubes pour alimenter le primaire avec une tension HF, généralement entre 50 kHz et 1 MHz.

Le transformateur de Tesla a une forme typique : l'enroulement secondaire consiste en une longue bobine, munie en son sommet d'une pièce métallique servant d'électrode de décharge et souvent de forme toroïdale. La tension de claquage dans l'air est de 1 à 3 kV par mm, et la forme de cette électrode est importante : le claquage se produira plus vite avec une pointe



▶ avec un excitateur de Slayer, on peut allumer lampes et tubes fluorescents

qu'avec un bol. Avec une bonne marge de sécurité et pour une tension de claquage de 3 kV/mm, nous aurions besoin d'une tension de 300 kV pour un arc de 10 cm !

Vous trouverez plus d'informations sur Wikipédia [1], ainsi que sur le site *hvtesla* [2], plus pratique et avec de belles photos de transformateurs. Et si vous voulez en apprendre encore plus, le guide de conception, construction et mise en œuvre de Kevin Wilson [3] vous dira tout (ou presque) sur le sujet.

Ne cherchez pas les complications

La construction d'un transformateur de Tesla pour de grands arcs électriques n'est pas évidente. En plus d'un travail mécanique conséquent, vous devrez aussi concevoir un circuit de commande capable de fournir une haute tension et une puissance non négligeable. Pour une simple expérimentation, il vaut mieux se contenter d'un dispositif plus simple, et ça permet déjà de beaucoup s'amuser.

Pour la production de la haute tension, le plus simple est de récupérer le circuit ad hoc d'un appareil bon marché. Le circuit imprimé avec l'oscillateur d'un appareil photo jetable (circuit du flash) ou d'un appareil (ou d'une lampe) antimoustiques convient très bien. La vidéo en deux parties *How to make a small Tesla Coil* [4] montre la construction d'un transformateur qui utilise le circuit d'une « raquette » antimoustiques. Malgré sa simplicité, ce dispositif permet l'obtention d'arcs passables. Vous pouvez aussi utiliser des composants discrets et construire un oscillateur autour d'un seul transistor. Le site *Learn to discover* [5] vous montre comment faire, avec schémas et vidéo. Vous obtiendrez des petits arcs, n'en attendez tout de même pas trop.

Il existe d'autres circuits simples semblables, appelés excitateurs de Slayer. Il s'agit de dispositifs dont le bobinage secondaire produit un champ magnétique élevé, ce qui permet d'allumer des lampes néon ou des tubes fluorescents ; le but principal n'est donc pas l'obtention d'arcs électriques. C'est très amusant aussi !

Si cela vous tente, rendez-vous sur *hackaday*, on vous présentera la construction d'une *Miniature Tesla Coil* [6]. L'oscillateur est bâti autour d'un 2N2222 courant et la bobine secondaire semble vraiment petite, mais les lampes et tubes fluorescents s'allument pourtant bien. Vous trouverez sur *YouTube* d'autres vidéos montrant des excitateurs de Slayer, par exemple celles en [7]. ◀

(160352 – version française : Jean-Louis Mehren)

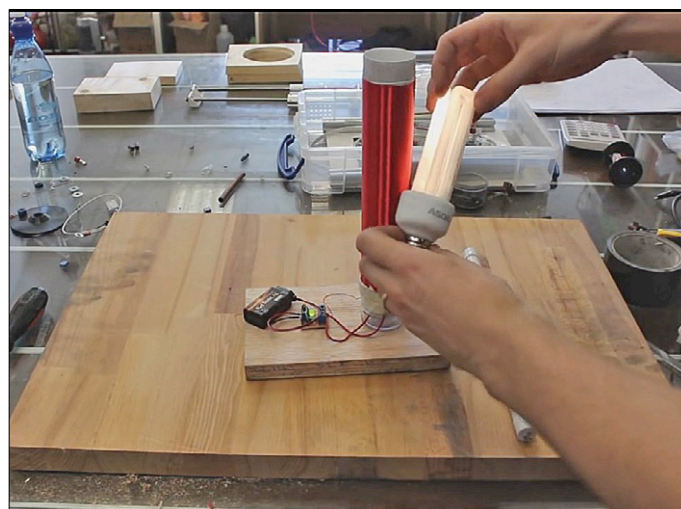
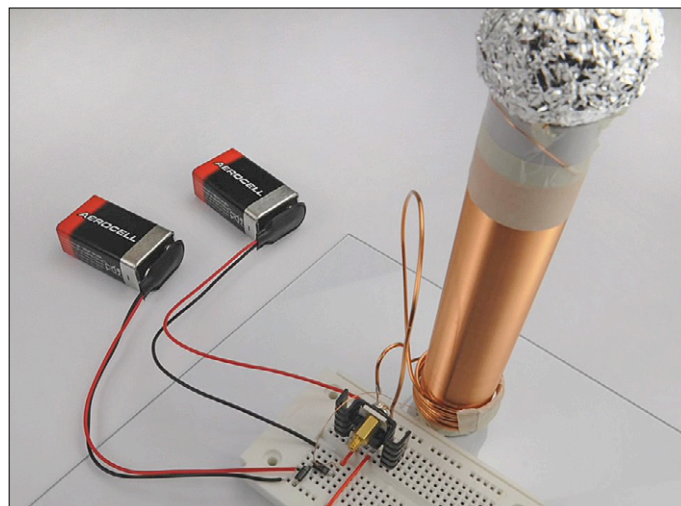
Liens

[1] fr.wikipedia.org/wiki/Bobine_Tesla

[2] www.hvtesla.com/

[3] www.teslacoildesign.com/

[4] www.youtube.com/watch?v=QDZnCOLZ394
www.youtube.com/watch?v=X2PrPHgOy04



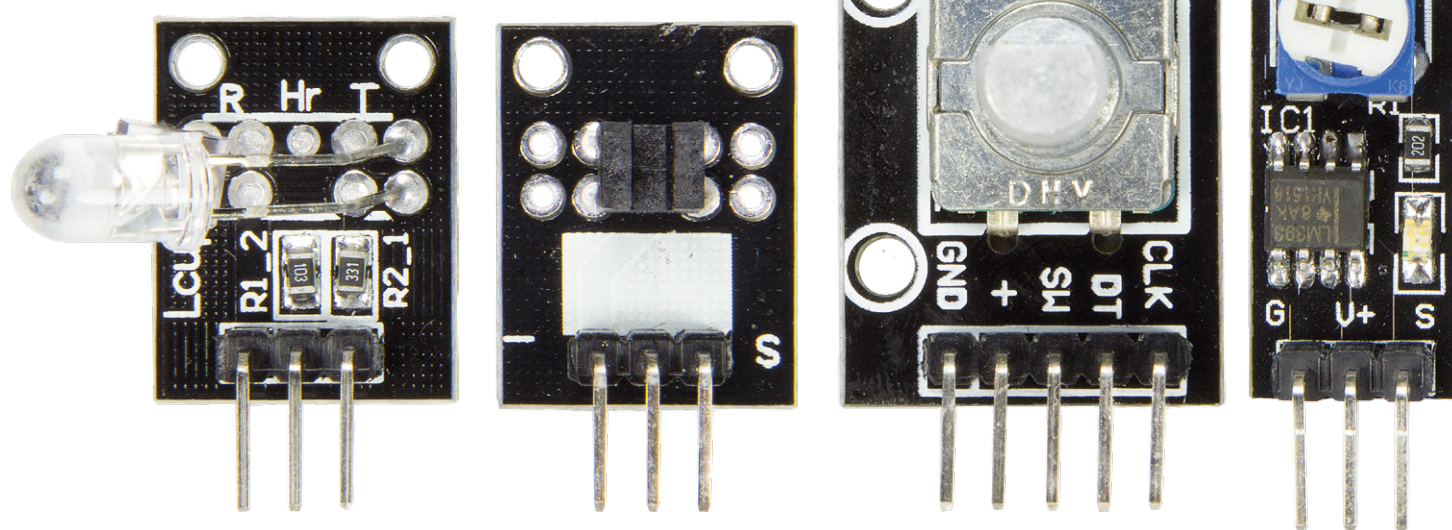
[5] learntodiscover-science.blogspot.nl/2016/10/simple-tesla-coil.html

[6] hackaday.com/2014/12/22/micro-tesla-coil-makes-a-perfect-stocking-stuffer/

[7] www.youtube.com/watch?v=MwG-EzVEEII
www.youtube.com/watch?v=3zjnX41K1pE
www.youtube.com/watch?v=LHCXqhhxGqA
www.youtube.com/watch?v=gSaGzhdI-QM

capteurs (4) pour Arduino et Cie

La commutation et le pilotage avec la lumière sont des techniques répandues qui présentent de nombreux avantages. En effet, elles ne nécessitent aucun contact mécanique, les contacts de commutation ne sont pas soumis à l'usure, les modules sont isolés électriquement et bien plus encore.



Burkhard Kainka

Nous présentons ici les barrières photoélectriques et leurs applications, du capteur à fourche au capteur de lumière réfléchi en passant par le pulsomètre. Nous verrons comment interpréter ces signaux de manière optimale. Une fois de plus, nous utilisons les capteurs du kit de 35 capteurs disponible dans l'e-choppe d'Elektor [1]. Comme toujours, vous pouvez télécharger tous les programmes de cet article [5].

Capteur de poursuite

Tout le monde connaît bien ces petits robots sur deux roues qui suivent une ligne blanche. Le principe est simple, c'est la capacité du sol à réfléchir la lumière qui est utilisée pour conduire le robot. Une diode émettrice IR illumine le sol tandis qu'un phototransistor IR capte la lumière réfléchi. En principe, on peut parler de barrière lumineuse à réflexion. Pour s'adapter à la sensibilité des différentes surfaces de sol et aux distances, le capteur de poursuite se sert d'un potentiomètre et d'un comparateur (**fig. 1**).

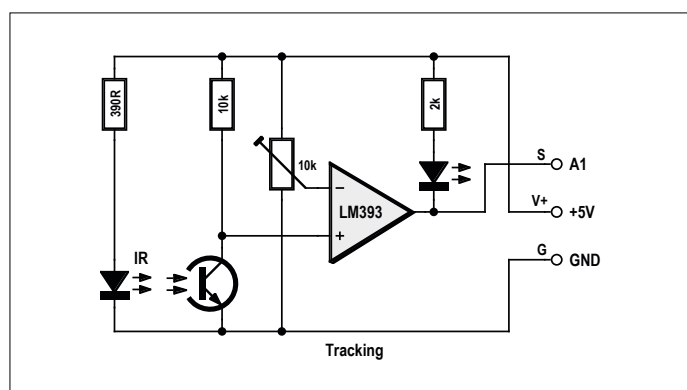


Figure 1. Barrière lumineuse à réflexion réglable.

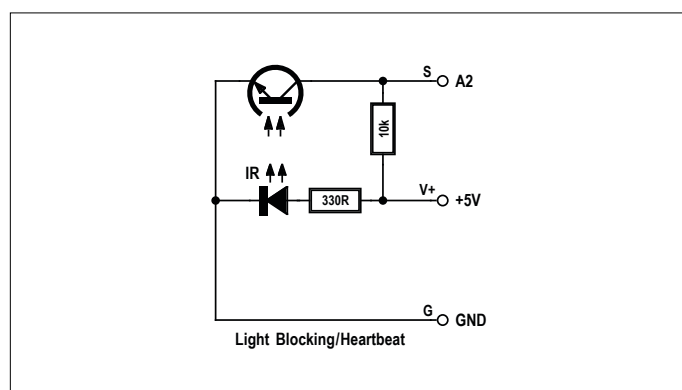


Figure 2. Barrière lumineuse à fourche.

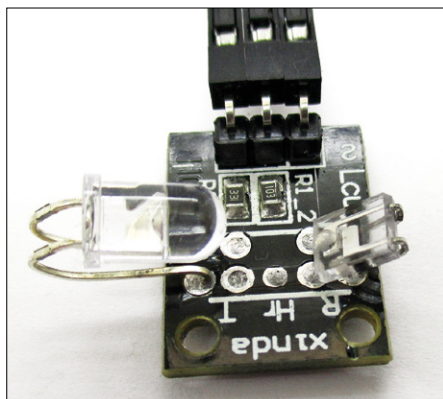


Figure 3. Pulsomètre une fois préparé.

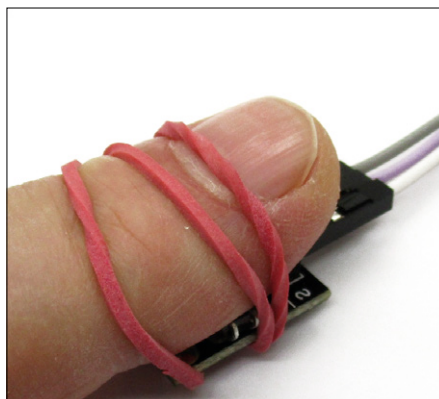


Figure 4. Pulsomètre en action.

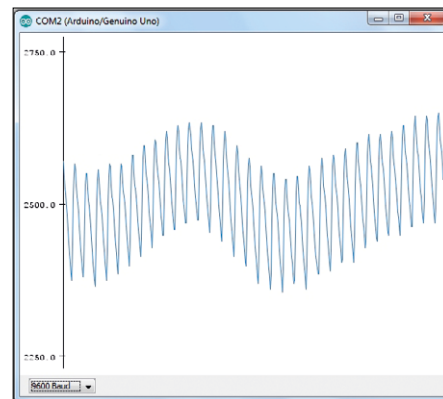


Figure 5. Tension sur le pulsomètre.

La LED située à la sortie du comparateur permet de tester facilement le fonctionnement. Un doigt placé à env. 1 cm du capteur peut servir de surface réfléchissante. Le point de commutation optimal peut être ajusté avec le potentiomètre.

À la sortie, on obtient un signal numérique susceptible d'être interprété comme à l'accoutumée. Vous pouvez soit vous servir du logiciel présenté dans le 2^e article de cette série [3], soit raccorder directement un actionneur. Maintenant, place à votre imagination et n'oubliez pas qu'il n'y a pas que les robots. Vous avez peut-être une lampe ou un ventilateur qui ne demande qu'à être commandé au doigt et à l'œil.

Barrière lumineuse à fourche

Beaucoup de machines et d'appareils (du scanner à l'imprimante 3D) sont équipés d'un interrupteur de fin de course qui permet de détecter la position d'une pièce mobile. Le problème des contacts mécaniques ou des microinterrupteurs est leur stabilité à long terme, de sorte qu'ils doivent être remplacés après un certain temps. La barrière lumineuse à fourche présente un avantage : la diode IR et le phototransistor ne s'usent pas à condition que le courant de la LED reste modéré.

La diode émettrice est pointée vers le phototransistor (**fig. 2**), l'écart entre les deux doit être faible. Le boîtier en forme de fourche présente une fente au travers de laquelle il est possible de faire passer un objet opaque. La position de commutation est reproductible avec une précision de moins d'un millimètre. Des capteurs à fourche similaires sont utilisés dans les souris, ce qui démontre bien leur fiabilité au quotidien.

Le signal de sortie du collecteur du phototransistor peut être interrogé de manière soit analogique soit numérique. Le signal est traité en fonction de la tâche envisagée et du logiciel utilisé. Le capteur permet un positionnement au millimètre près, ce qui est nécessaire par exemple pour les imprimantes 3D. Dans les cas moins critiques, il suffit d'une simple interrogation numérique de type « lumière ? » ou « pas de lumière ? ».

Pulsomètre

La mesure optique est souvent utilisée dans les moniteurs de fréquence cardiaque. La lumière émise à une longueur d'onde adéquate par une diode au travers d'un doigt ou d'un lobe d'oreille est récupérée de l'autre côté des tissus par un phototransistor. La lumière est partiellement absorbée par le sang qui se trouve dans le corps. Le rythme cardiaque modifie périodiquement la circulation sanguine et module ainsi légèrement le flux lumineux. Les signaux doivent bien sûr être interprétés de manière

analogique, car il ne faut pas s'attendre à de grandes variations. Le capteur de rythme cardiaque est conçu comme un capteur à fourche (voir figure 2). En revanche, l'écart est plus grand entre l'émetteur et le récepteur. Comme la diode émettrice IR et le phototransistor ont des pattes de raccordement longues, il est facile de trouver la position optimale. La carte est identique à celle du capteur à fourche sauf que le connecteur et les composants sont montés de l'autre côté.

Pour que cela fonctionne, il faut donc courber la diode IR et le phototransistor de manière à ce qu'un doigt puisse être coincé entre les deux. Attention, le doigt ne doit toucher aucune partie conductrice de la carte, car ceci risquerait d'entraîner des ronflements supplémentaires. La **figure 3** montre comment nous avons courbé les capteurs. De cette manière, la diode IR pointe presque directement vers le phototransistor. La tension de sortie approche maintenant le zéro. La lumière est coupée dès qu'on pose le doigt dessus. Comme le phototransistor conduit moins, la tension augmente. Le doigt peut être calé à l'aide d'un élastique (**fig. 4**) si l'on veut faire plusieurs tests d'affilée. Il est important que le doigt soit relaxé et que la pression ne varie pas de manière à ne pas fausser le signal. Pour notre premier test dans l'EDI Arduino, nous nous servons du programme VoltageAD2 (voir 1^{er} article de la série [2]). La tension sur phototransistor peut être représentée à l'aide du traceur série (**fig. 5**). On voit clairement la modulation de la lumière par le pouls. Les oscillations plus lentes, superposées au signal, proviennent de mouvements incontrôlables. Pour améliorer le traitement du signal, nous utilisons d'abord un filtre passe-bas (**listage 1**). Ce filtre codé sur deux lignes donne simplement la moyenne glissante des valeurs mesurées. Chaque nouvelle valeur mesurée est ajoutée à la somme de valeurs moyennes *mean*, après avoir réduit cette somme de 1/20. Chaque valeur mesurée n'a donc qu'une influence de 5% sur la moyenne. La constante de temps du filtre passe-bas est la période d'échantillonnage multipliée par 20, soit dans ce cas 0,4 s. La fréquence d'échantillonnage a été fixée à 50 Hz, pour atténuer les éventuels signaux parasites du réseau électrique. La fréquence de coupure du filtre correspond à $f = 1 / (2 \pi T)$, soit 0,398 Hz. Pour l'illustration, les valeurs mesurées non filtrées et filtrées sont toujours affichées en alternance.

Le traceur série permet de montrer le fonctionnement du filtre. La moyenne glissante suit le signal avec une certaine inertie (**fig. 6**). Si l'on retire rapidement le doigt du capteur, on peut voir la réponse aux impulsions du filtre passe-bas (**fig. 7**).

Il ne nous reste plus qu'à calculer la différence entre le signal

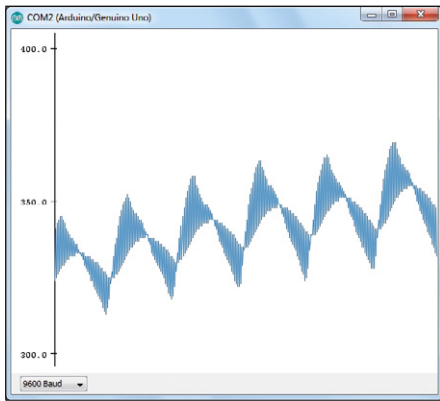


Figure 6. Valeurs d'origine et valeurs moyennes.

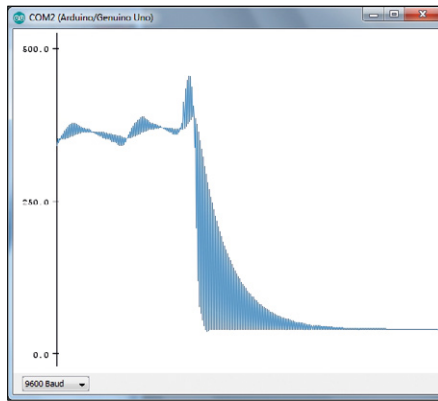


Figure 7. Réponse impulsionnelle.

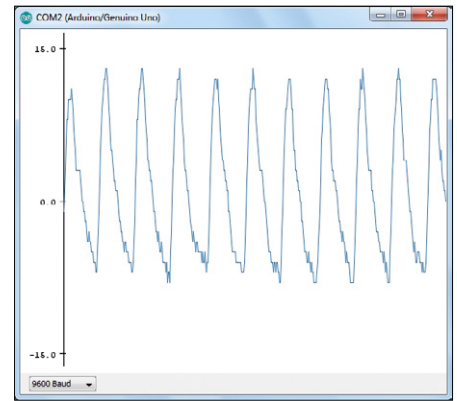


Figure 8. Signaux d'impulsion filtrés.

entrant et la valeur moyenne (**listage 2**). Et voilà, notre filtre passe-haut est lui aussi terminé ce qui nous permet d'obtenir un signal de pouls propre, dépourvu de toute oscillation lente. Passons maintenant à l'interprétation de la fréquence du pouls. Le montage permet de détecter d'autres détails. Le signal grimpe abruptement pour redescendre plus tranquillement (**fig. 8**). On voit qu'avec chaque battement, le cœur pompe du sang dans le doigt au travers des artères et que le sang repart doucement au

travers des veines. Le pulsomètre permet de détecter également les éventuelles irrégularités du rythme cardiaque.

Pour parachever le pulsomètre, il convient maintenant de mesurer les écarts entre les flancs positifs. L'Arduino dispose d'une fonction simple pour mesurer le temps : `millis()`. Le temps écoulé depuis la dernière initialisation est restitué en millisecondes. Ceci permet de mesurer l'écart entre deux battements cardiaques. Le programme (**listage 3**) indique la durée du

Listage 1. Calcul de la valeur moyenne

```
//VoltageAD2 0...1023 at AD2 filter
int sensorPin = 2;
int value;
int mean;

void setup() {
  Serial.begin(9600);
  lcd.begin(16, 2);
}

void loop() {
  value = analogRead(sensorPin);
  mean = mean - mean / 20;
  mean = mean + value;
  Serial.println(value);
  Serial.println(mean / 20);
  delay(19);
}
```

Listage 2. Filtre passe-haut

```
//Filter2AD2 0...1023 at AD2 filter
int sensorPin = 2;
int value;
int mean;

void setup() {
  Serial.begin(9600);
}

void loop() {
  value = analogRead(sensorPin);
  mean = mean - mean / 20;
  mean = mean + value;
  value = value - mean / 20;
  Serial.println(value);
  delay(19);
}
```

Listage 3. Pulsomètre Arduino

```
void loop() {
  value = analogRead(sensorPin);
  mean = mean - mean / 20;
  mean = mean + value;
  value = value - mean / 20;
  if ((old < 0) & (value > 0)) {
    time2 = millis();
    pulseTime = time2 - time1;
    time1 = time2;
    //Serial.println(pulseTime);
    n = n + 1;
    pulseFreq = 60000 / pulseTime;
    if (pulseFreq > 45) {
```

```
      Serial.println(pulseFreq);
      lcd.setCursor(0, 0);
      lcd.print(pulseTime);
      lcd.print(" ms ");
      lcd.setCursor(0, 1);
      lcd.print(pulseFreq);
      lcd.print(" /min ");
    }
  }
  old = value;
  delay(20);
}
```


pouls en ms ainsi que le nombre de battements par minute sur l'afficheur à cristaux liquides du *shield* d'extension Elektor [6]. Le résultat ne se laisse pas attendre, puisque la fréquence est calculée à chaque battement de cœur à partir de la durée de l'impulsion. Cependant, nous avons constaté que le moindre mouvement supprime des impulsions, ce qui conduit à multiplier la durée du pouls par deux ou trois. C'est la raison pour laquelle les valeurs mesurées ne sont présentées que lorsque la fréquence cardiaque dépasse les 45 battements par minute.

Pulsomètre en Bascom

En Bascom, il faut tout d'abord s'occuper de la mesure du temps puisque celle-ci n'est pas prévue. Pour cela, on se sert d'une interruption de *timer*, ce sera *timer0*. La routine d'interruption est interrogée toutes les 10 ms ce qui correspond à la résolution avec laquelle le temps est enregistré.

Au point où on en est, appliquons-nous pour cette interruption. Dans notre cas, cela signifie que les mesures de la tension seront aussi exécutées dans l'interruption. L'avantage est que les mesures sont effectuées exactement toutes les 10 ms, ce qui permet d'éliminer les éventuelles interférences du 50 Hz. Pour un ATmega, 10 ms sont amplement suffisantes pour procéder à une évaluation complète durant l'interruption (**listage 4**). Comme avec l'Arduino, cela concerne le calcul de la moyenne glissante, l'évaluation des flancs d'impulsions, l'élimination des erreurs de mesure causées par des impulsions manquantes et la sortie des résultats sur le terminal et l'afficheur à cristaux liquides. Et voilà, nous avons un pulsomètre programmé en Bascom qui présente les mêmes caractéristiques que celui programmé dans l'EDI Arduino.

Codeur rotatif

Si vous avez déjà démonté une vieille souris dotée d'une boule, vous aurez certainement remarqué les barrières lumineuses qui permettent de mesurer la rotation. Le point décisif est qu'il faut non seulement calculer les impulsions, mais aussi détecter les

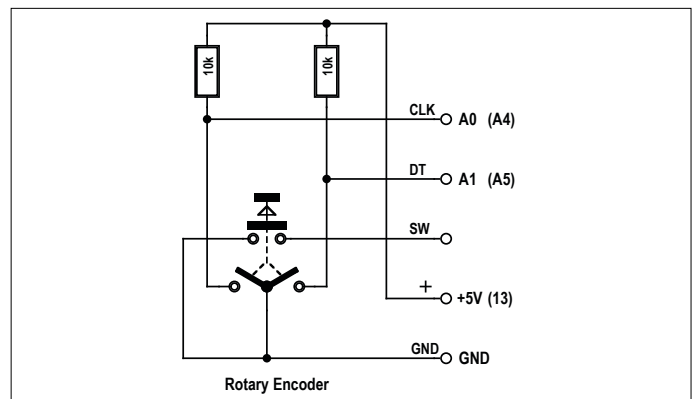


Figure 9. Codeur rotatif.

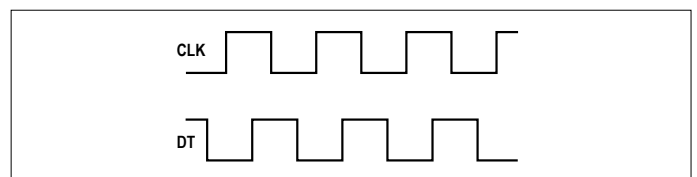


Figure 10. Signaux de commutation lors de la rotation.

inversions de rotation à l'aide des deux barrières lumineuses. Ces dernières sont disposées de telle manière qu'elles livrent des impulsions rectangulaires décalées de 90°.

C'est exactement ce type de signaux qui est émis par le codeur rotatif que l'on trouve dans le kit (**fig. 9**). En revanche, ce codeur rotatif n'utilise pas des barrières lumineuses, mais des contacts mécaniques. Il dispose néanmoins de crans mécaniques « audibles », ce qui permet de compter à chaque fois que le codeur tourne. Une rotation correspond à 20 crans. Au premier abord, cela peut sembler modeste, mais en fait comme

Listage 4. Pulsomètre Bascom

```

'-----
'Pulse.BAS  ADC2, LCD
'-----

...

Config Timer0 = Timer , Prescale = 1024
On Ovft0 Tim0_isr
Enable Timer0
Enable Interrupts

...

Do
Loop

Tim0_isr:
  Timer0 = 100      '10ms
  D = Getadc(2)
  Du = U / 40
  U = U - Du

  U = U + D
  Ticks = Ticks + 1
  U1 = D - Du
  If U1 > 0 And U2 < 0 Then
    T = Ticks * 10
    F = 60000 / T
    Ticks = 0
    If F > 45 Then
      Print T
      Print F
      Locate 1 , 1
      Lcd T
      Lcd " ms "
      Locate 2 , 1
      Lcd F
      Lcd " /min "
    End If
  End If
  U2 = U1
Return

```

l'axe n'a pas de butée de fin de course, le codeur peut tourner des centaines voire des milliers de fois, ce qui est bien pratique pour obtenir la précision requise pour le calcul d'une fréquence ou d'une tension. Certes, nous disposons d'un bouton-poussoir, mais nous n'avons pas besoin de l'utiliser pour ce montage. Même si les deux contacts se valent, l'un est cadencé par un signal d'horloge (CLK) tandis que l'autre se sert de la broche de données (DT). La **figure 10** montre les signaux de sortie lors d'une rotation. L'interprétation est facile : on attend un flanc descendant sur CLK et on regarde ce qui se passe sur DT, et en fonction du résultat on augmente ou on réduit la valeur du compteur.

Le programme donné en exemple (**listage 5**) utilise deux codeurs en même temps, ce qui permet de régler deux signaux de sortie analogiques PWM1 et PWM2 sur les sorties 9 et 10 de l'Arduino. En ajoutant un autre filtre passe-bas, ceci pourrait devenir une double source de tension réglable pour d'autres applications de mesure d'ordre général. Ce système permet,

p. ex., de régler le mélange de couleur de la LED rouge/verte du kit de capteurs et ce, de manière précise et reproductible. De plus, les valeurs réglées sont délivrées sur le traceur série et l'afficheur à cristaux liquides.

Certes, le codeur rotatif est doté de ses propres résistances de rappel vers le haut, mais nous préférons utiliser celles de l'Arduino. En effet, ceci permet d'exploiter un ou deux codeurs en même temps avec le même programme. Le premier codeur est raccordé aux entrées A0 et A1. Ce n'est pas un hasard puisque c'est là que se trouvent les touches S1 et S2 du *shield* d'extension. De cette manière, il n'y a aucun conflit, puisque tous les contacts sont reliés au fil de masse (GND). L'avantage est qu'en cas de besoin la valeur de sortie d1 peut être réglée à l'aide de ces touches. S1 livre le signal de cadence tandis que S2 donne le sens de rotation.

Le deuxième codeur est relié à A4 et A5. Les broches de port correspondantes sont doubles sur l'Arduino, elles se trouvent à la diagonale des lignes SDA et SCL du bus I²C. On y trouve

Listage 5. Analyse du codeur en langage C Arduino

```
//Encoder A0/A1 PWM1, A4/A5 PWM2

...

int clk1 = A0;
int dt1 = A1;
int clk2 = A4;
int dt2 = A5;
int pwm1 = 9;
int pwm2 = 10;

...

void loop() {
    new1 = digitalRead(clk1);
    if((new1==0) & (old1==1)){
        if (digitalRead(dt1)==0) d1++; else d1--;
        Serial.println (d1);
        if (d1 > 250) d1 = 250;
        if (d1 < 0) d1 = 0;

        analogWrite (pwm1, d1);
        lcd.setCursor(0, 0);
        lcd.print(d1 * 20);
        lcd.print(" mV ");
    }
    old1=new1;

    new2 = digitalRead(clk2);
    if((new2==0) & (old2==1)){
        if (digitalRead(dt2)==0) d2++; else d2--;
        Serial.println (d2);
        if (d2 > 250) d2 = 250;
        if (d2 < 0) d2 = 0;
        analogWrite (pwm2, d2);
        lcd.setCursor(0, 1);
        lcd.print(d2 * 20);
        lcd.print(" mV ");
    }
    old2=new2;
}
```

Listage 6. Analyse du codeur en Bascom

```
'Encoder.BAS C0/C1 PWM1a, C4/C5 PWM1b

...

Do
    New1 = Pinc.0
    If New1 = 0 And Old1 = 1 Then
        If Pinc.1 = 0 Then D1 = D1 + 1 Else D1 = D1 -1
        If D1 > 1023 Then D1 = 1023
        If D1 < 0 Then D1 = 0
        Pwm1a = D1
        Print D1
        Locate 1 , 1
        Lcd D1
    End If

    Old1 = New1

    New2 = Pinc.4
    If New2 = 0 And Old2 = 1 Then
        If Pinc.5 = 0 Then D2 = D2 + 1 Else D2 = D2 -1
        If D2 > 1023 Then D2 = 1023
        If D2 < 0 Then D2 = 0
        Pwm1b = D2
        Print D2
        Locate 2 , 1
        Lcd D2
    End If
    Old2 = New2
Loop
```

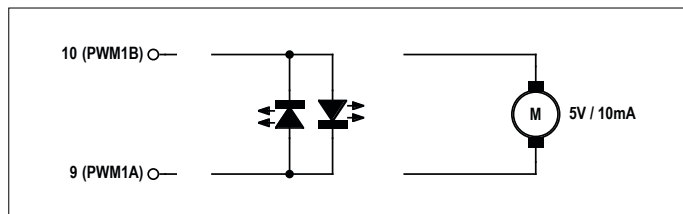



Figure 11. Étages de sortie montés en pont.

également à proximité une broche GND. Il ne manque plus qu'une connexion supplémentaire au 5 V pour les résistances de rappel vers le haut du codeur. Pas de problème, nous allons tout simplement mettre la broche 13 à l'état haut ; c'est la broche à laquelle est raccordée la LED de l'Arduino. Ça fonctionne à tous les coups : et puis, s'il nous manque une ligne GND ou VCC, on peut toujours programmer une broche de port en conséquence.

Codeur en Bascom

La version Bascom du programme présente à peu près la même structure. Comme les langages sont interchangeables, il suffit de reprendre le code source, de conserver le nom des variables et d'adapter le tout à la syntaxe spécifique à ce langage. Pour finir, on constate une seule véritable différence : en Bascom, on utilise les deux sorties MLI du *timer1* avec une résolution type de 10 bits, ce qui correspond à 1 023 pas écartés d'environ 5 mV.

Deux sorties MLI parfaitement réglables, voilà qui me donne envie de faire une petite expérience en liaison avec des étages de sortie montés en pont puisque les deux signaux à MLI proviennent du même *timer* et présentent des impulsions qui coïncident exactement et sont synchronisées par un flanc montant. Pourquoi ne pas se servir d'une LED verte et d'une LED rouge comme diodes antiparallèles entre ces deux sorties (**figure 11**). En réglant une tension moyenne de 2 500 mV p. ex. sur les deux canaux, les deux LED restent éteintes. Une LED commence à briller dès qu'un canal est en déséquilibre.

De la même façon, on pourrait brancher un moteur CC (de type magnétophone, courant de démarrage de 10 mA) pour régler le sens de rotation et le nombre de tours. Il est vrai qu'on devrait se servir d'une vraie commande de moteur. Mais le montage fonctionne quand même. Avec cette configuration, il n'y a pas de risque de tensions d'induction parce que les sorties se trouvent toujours à l'état basse impédance. Les sorties ne passent à l'état haute impédance que si l'on remet à zéro l'Arduino durant le fonctionnement. Par mesure de précaution, nous retirons donc le moteur durant le redémarrage... ◀

(160302 – version française : Pascal Duchesnes)

Liens

- [1] www.elektor.fr/arduino-sensor-kit
- [2] www.elektormagazine.fr/160152
- [3] www.elektormagazine.fr/160173
- [4] www.elektormagazine.fr/160210
- [5] www.elektormagazine.fr/160302
- [6] www.elektormagazine.fr/140009

IoT Design Contest

Curation is Creation

2017.5.1 ~ 8.31 | Total Prize of \$15,000

**For Makers,
By Makers**

IoT UCC Platform
cybermakerspace.com

WIZwiki_W7500

ARM mbed enabled

WiFi Shield

WizFi310

IOP (Internet Offload Processor), TCP/IP hardware implementation, is fast, safe and reliable to perform the data transfer by fully offloading processing of the entire TCP/IP stack without host CPU intervention.

Code source et fichiers Gerber : où sont-ils ?



Vous êtes nombreux à chercher les fichiers associés (code, circuit imprimé...) aux articles publiés dans le magazine, dans la série de livres « 3XX circuits » ou encore dans les compilations. Malheureusement, en dépit de notre vigilance, vous tombez parfois sur des liens morts. Ceux-ci sont un dommage collatéral de l'évolution constante des sites d'Elektor et nous vous remercions de nous les signaler. Nous les corrigerons aussitôt.

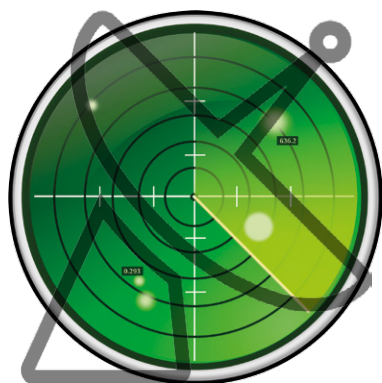
Le plus sûr chemin pour trouver un fichier lié à un article publié est de commencer par repérer les six premiers chiffres de la référence mentionnée en fin d'article dans le magazine. Prenons par ex. le « Chargeur de batteries à panneaux solaires » publié en juillet 2008. Sa référence est **080225-I**. Il suffit maintenant de saisir dans votre navigateur l'URL suivant : www.elektormagazine.fr/080225

Vous accéderez automatiquement à la page associée à cet article avec tous les téléchargements disponibles.

La rédaction à votre service.

corrélation numérique

détection de motif et de période



Robert Lacoste (Chaville)

Suite à la publication de mes articles sur le filtrage numérique (Elektor, janvier à mai 2016), un lecteur m'a écrit pour me demander si je pouvais dire deux mots sur les algorithmes de corrélation. Voici donc un petit article sur ce thème pour contenter nos lecteurs. Comme quoi il ne faut pas hésiter à contacter les auteurs d'Elektor !

Pas de panique ! Même si le traitement de signal peut être quelquefois un peu rébarbatif, je vous promets que je n'utiliserai ici aucune notion plus complexe que... des additions et des multiplications.

Corrélation, quésako ?

Ouvrons nos dictionnaires. En français une corrélation est « une relation existant entre deux notions dont l'une ne peut être pensée sans l'autre » (Larousse [1]). En traitement du signal, la notion équivalente est proche. De quoi s'agit-il ? Imaginez que vous avez un signal bruité, mais qui inclut des répétitions d'un signal connu. Chaque réplique est plus ou moins précise, a une amplitude variable, peut être n'importe où dans le signal et peut même être noyée dans du bruit (**figure 1**). Le signal à rechercher, appelons-le « motif », est connu, mais comment en localiser chaque occurrence, et estimer son amplitude ? Pas évident, n'est-ce pas ? En fait c'est facile grâce à l'opération mathématique également appelée « corrélation ».

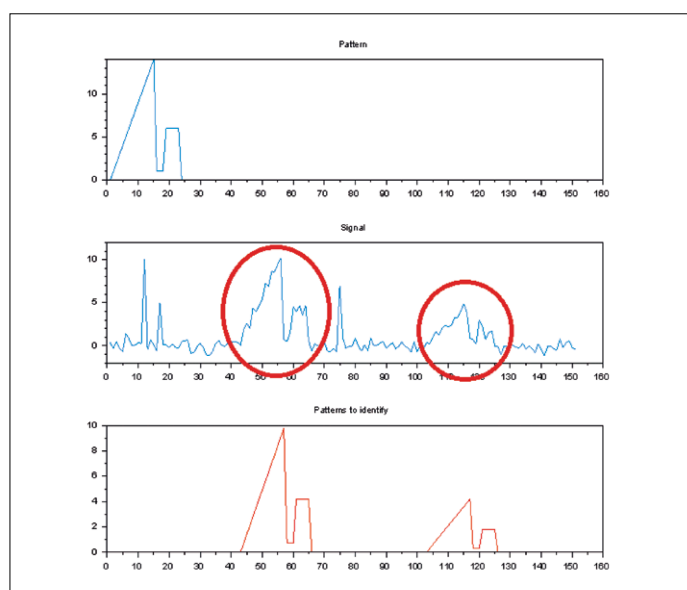


Figure 1. Un motif connu (haut) est répliqué deux fois dans un signal bruité, à différentes amplitudes (milieu). Le but du jeu est d'identifier ces deux répliques (bas)...

Je vais vous expliquer comment cela se calcule, mais voyons d'abord une application typique : le radar. Un radar envoie une impulsion, puis cherche à détecter l'impulsion réfléchiée par les obstacles. C'est facile si l'amplitude de l'écho est grande, mais beaucoup moins simple si le bruit de fond est élevé. Grâce à la corrélation, les radars modernes peuvent détecter un objet de petite taille à plusieurs dizaines de kilomètres, en envoyant non pas une simple impulsion, mais un motif plus complexe qu'il sera possible d'identifier dans l'écho par traitement numérique...

Un algorithme !

Entrons dans le vif du sujet. La bonne nouvelle est qu'un calcul de corrélation est simple à expliquer, et encore plus simple à programmer sur un microcontrôleur, un DSP ou un FPGA. Imaginez que vous avez un signal numérisé par un convertisseur analogique/numérique, sous la forme d'un tableau de N échantillons successifs. De même, un second tableau contient le motif recherché, de M échantillons. Le motif est plus court

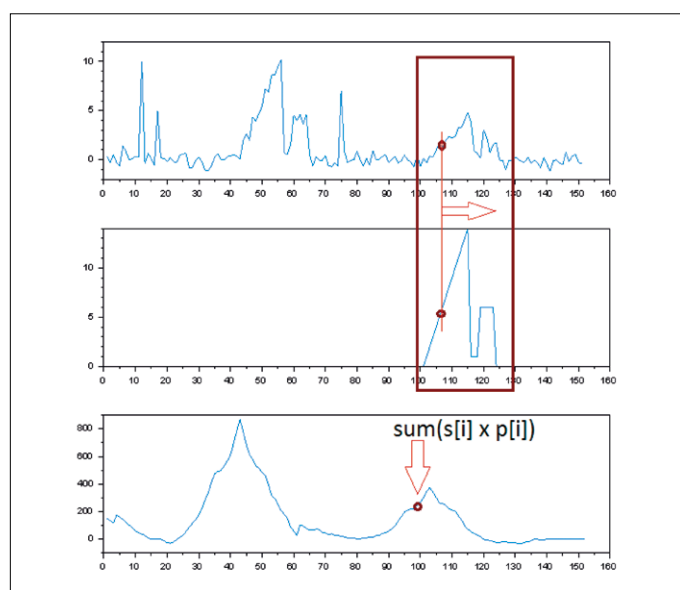


Figure 2. Une corrélation fonctionne en déplaçant progressivement le motif (milieu) par rapport au signal (haut). Les points des deux courbes sont alors multipliés et additionnés, ce qui donne un point sur la corrélation (bas).

que le signal complet, donc $M < N$. Commencez par positionner le tableau des M points du motif en face des M premiers points du signal. Pour chaque point, multipliez les valeurs correspondantes du motif et du signal, puis additionnez tous ces résultats. Vous avez le premier point de la corrélation. Ensuite, déplacez le motif d'un échantillon vers la droite et refaites le même calcul pour trouver le deuxième point de la corrélation. Continuez jusqu'à la fin du signal et vous avez la corrélation du motif et du signal, qui compte $N - M + 1$ points dans ce cas. Pour ceux qui préfèrent un pseudo-code (en supposant que les index des tableaux commencent à zéro comme en C), le voici :

```
pour i=0 à N-M faire
{
    convolution[i]=0
    pour j=0 à M-1 faire
    {
        convolution[i] =
            convolution[i] + motif[j] * signal[i+j]
    }
}
```

Très simple, non ? Cet algorithme est illustré sur la **figure 2**. Comment ce calcul répond-il à notre problème ? Très simple également : la valeur du signal de corrélation est d'autant plus élevée que le motif et le signal correspondent. Il suffit donc de rechercher les pics du signal de corrélation, leur position dans le temps indique les occurrences probables du motif, et leur amplitude est proportionnelle à l'amplitude de la réplique ! Regardez de nouveau la figure 2 pour vous en convaincre.

Quelques exemples

Bien sûr, dans la vie réelle c'est un peu plus complexe que dans ce premier exemple, car les signaux sont bruités, mais l'algorithme fonctionne en général bien... J'ai fait quelques simulations pour vous avec différents types de motifs et niveaux de bruit pour illustrer tout cela. J'ai utilisé l'outil de calcul numérique Scilab (voir [2]), qui a le bon goût d'être gratuit. Vous pourrez ainsi le télécharger et reproduire ces exemples vous-même.

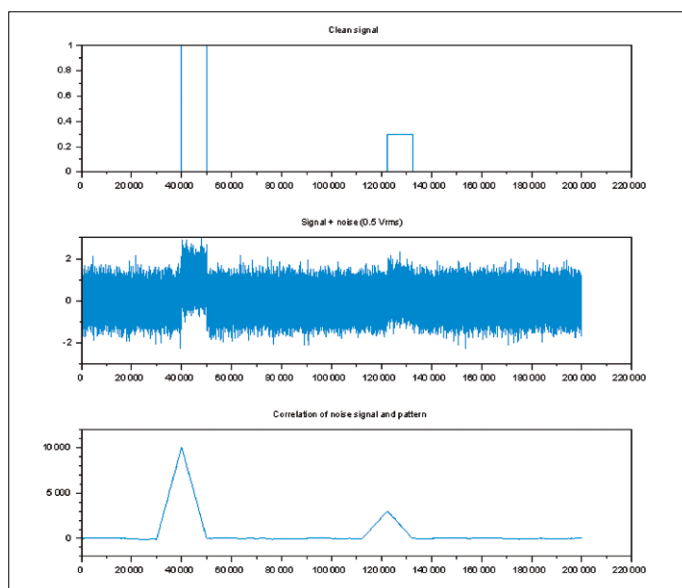


Figure 3. Une corrélation avec un motif rectangulaire donne un résultat en forme de triangle. Ici, le motif est assez long (10.000 échantillons), ce qui permet d'obtenir un très bon résultat même avec un niveau de bruit élevé.

Commençons par le cas simple d'un motif rectangulaire. Imaginons que le signal contienne deux copies du motif, avec un fort niveau de bruit. Le code source Scilab est le suivant :

```
// Motif à trouver, rectangulaire
PatternLength=10000;
Pattern=ones(1,PatternLength);

// Signal d'entrée, avec deux répliques du motif
SignalLength=200001;
Signal=[1:SignalLength]*0;
PatternPosition1=40000; PatternAmplitude1=1;
Signal(PatternPosition1:PatternPosition1+
    PatternLength-1)=Pattern*PatternAmplitude1;
PatternPosition2=122300; PatternAmplitude2=0.5;
Signal(PatternPosition2:PatternPosition2+
    PatternLength-1)=Pattern*PatternAmplitude2;

// Ajoutons du bruit gaussien
NoiseLevel=0.5;
NoisySignal=Signal+grand(1,SignalLength,"nor",0,
    NoiseLevel);

// Calculons la corrélation
// (Scilab a une fonction pour cela...)
Correlation = xcorr(NoisySignal, Pattern);
plot(Correlation);
```

Regardez le résultat de cette simulation sur la **figure 3** : malgré le bruit, le calcul de corrélation permet de détecter sans soucis les occurrences et les amplitudes du motif. Notez qu'avec un motif rectangulaire, la corrélation est un triangle (réfléchissez... c'est normal, car la corrélation est ici proportionnelle à l'aire de la zone commune entre le signal et le motif).

Une petite précision : sur la figure 3, le motif est long (10.000 échantillons). Que se passe-t-il s'il est dix fois plus court, avec le même niveau de bruit ? J'ai fait la simulation pour vous, et la **figure 4** donne le résultat. L'algorithme fonctionne toujours, mais la corrélation résultante est un peu plus bruitée. Ceci montre que, comme souvent en calcul numérique, plus on a d'échantillons, plus il est aisé d'extraire le signal du bruit. Donc une fréquence d'échantillonnage plus rapide est bénéfique pour le résultat, mais nécessite bien sûr plus de calculs. Regardons ce qui se passe avec des motifs plus complexes que des impulsions rectangulaires. La **figure 5** montre le résultat lorsque le motif correspond à une période d'un signal sinusoïdal. Le niveau de bruit est tellement élevé qu'un humain ne peut quasiment pas « voir » l'occurrence du motif dans le bruit, mais l'algorithme fonctionne parfaitement, pourvu que le motif soit assez long (10.000 échantillons ici aussi). En fait, les mathématiciens ont même démontré que le calcul de corrélation est le meilleur algorithme linéaire pour détecter un motif connu dans un signal, du moins si le bruit est ce qu'on appelle un bruit blanc. Pour cette raison, un calcul de corrélation entre un signal et un motif connu est appelé **filtre adapté** ou **filtre optimal**.

Une pincée d'aléatoire

Allons un cran plus loin. Plus le motif est complexe et plus l'algorithme fonctionne bien, car la probabilité d'avoir un motif complexe dans le bruit est d'autant plus faible. Vous voulez être impressionné ? Regardez la **figure 6**. Ici, le motif est court, seulement 200 échantillons, mais il s'agit d'un signal pseudo-aléatoire (une suite de 200 nombres pris au hasard). Cette

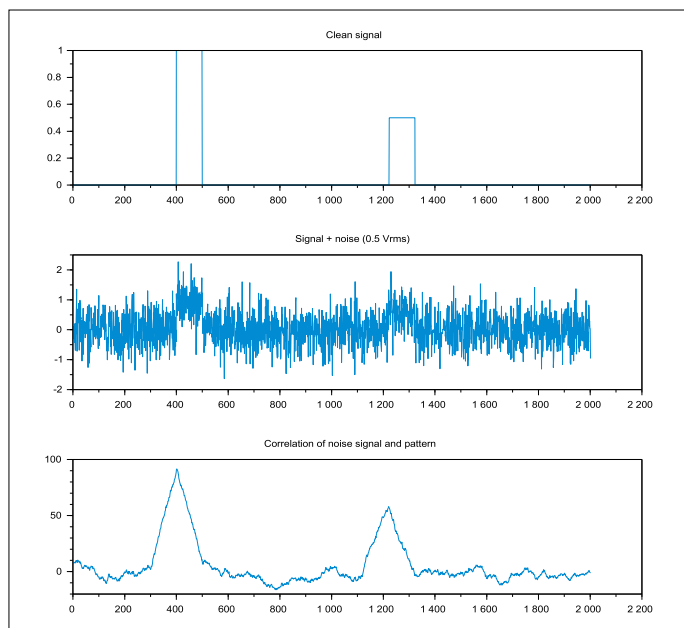


Figure 4. Le même exemple qu'à la figure 3, mais avec un motif plus court (100 échantillons) donne un résultat moins précis.

séquence est à nouveau répétée deux fois dans le signal à des amplitudes différentes, et un énorme bruit blanc est ajouté au résultat. Ce motif pseudo-aléatoire devient clairement invisible dans le bruit. Cependant le résultat de la corrélation est plus que limpide : chaque réplique du motif dans le signal est clairement identifiée par une impulsion très étroite sur la corrélation ! Si vous y réfléchissez, vous comprendrez facilement pourquoi. Comme le motif est pseudo-aléatoire, il n'y a une corrélation que lorsque le motif est précisément aligné avec le signal correspondant, le reste du bruit s'annule en moyenne. Ceci donne un positionnement très précis du motif dans le signal ainsi qu'une amélioration très efficace du rapport signal sur bruit.

Cette approche fournit en particulier un très bon moyen pour améliorer les performances d'un système de communication.

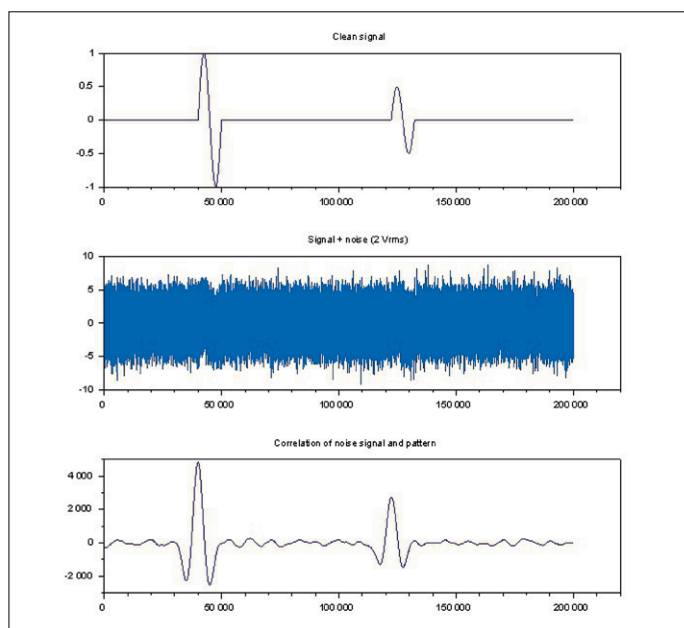


Figure 5. Un exemple avec un motif sinusoïdal.

Imaginez que vous avez un flux de bits à transmettre. Vous pouvez définir deux motifs pseudo-aléatoires différents, puis encoder chaque bit avec l'un ou l'autre de ces deux motifs (l'un pour les 0, l'autre pour les 1). Côté récepteur, faites une corrélation du signal reçu avec les deux motifs. Vous récupérerez les 0 et 1 du signal initial même si le niveau de bruit est très élevé. C'est une technique de modulation très connue et appelée **DSSS** (*Direct Sequence Spread Spectrum*). Elle est utilisée par Zigbee, le GPS, mais aussi le Wi-Fi dans sa variante 802.11b. Encore plus fort : cette même méthode peut être utilisée pour faire communiquer plusieurs paires de correspondants sur le même canal radio, sans qu'il y ait de mélange entre eux. Comment ? Il suffit de définir des paires de motifs pseudo-aléatoires différents pour chaque couple de correspondants ! Les autres communications seront vues comme du bruit puisqu'elles ne seront pas corrélées aux motifs recherchés. Cela vous rappelle quelque chose ? C'est très exactement ce qu'on appelle une modulation **CDMA** (*Code Division Multiple Access*), et votre ordiphone l'utilise s'il est en 3G... En passant, c'est aussi une bonne solution pour cacher une transmission à des récepteurs non autorisés, vous devinerez donc facilement pour quel type d'application ça a été inventé...

Quelle puissance de calcul ?

Un mot ici pour les lecteurs qui seraient inquiets de la puissance du processeur nécessaire pour calculer une corrélation. Supposons que le motif soit constitué de 100 échantillons ($M = 100$) et que vous vouliez traiter un signal en temps réel échantillonné à 10 kéch/s (10.000 échantillons par seconde). Le calcul d'une corrélation nécessite 100 multiplications et additions pour chaque échantillon du signal d'entrée, soit $100 \times 10.000 = 1$ million de multiplications par seconde et autant d'additions. Ceci serait difficile sur un 8051 âgé de 20 ans, mais c'est à la portée de tout processeur à 32 bits actuel. De plus la majorité des cœurs ARM (comme les Cortex M3) et bien sûr des DSP disposent de l'instruction MAC (*Multiply And Add*) qui permet de réaliser les deux opérations en un seul cycle.

Que faire si les contraintes sont plus sévères, par exemple si la longueur du motif ou le débit des échantillons sont supérieurs de plusieurs ordres ? Vous avez alors besoin d'un processeur beaucoup plus rapide ou d'un FPGA... ou d'un peu plus de math. En deux mots, pour ceux qui sont intéressés, il existe un moyen pour calculer une corrélation beaucoup plus efficace que l'algorithme présenté ci-dessus. On passe alors par deux FFT (*Fast Fourier Transform*). Cela nécessiterait un article un peu plus long, mais en bref (et avec mes excuses pour les lecteurs réticents aux mathématiques), une corrélation est quasi identique à une autre opération appelée **convolution** (il y a juste un retournement temporel du motif dans le calcul entre les deux). Et un algorithme de convolution rapide est possible dans le domaine fréquentiel. Respirez et répétez après moi : « la FFT d'une convolution est la multiplication terme à terme des FFT des deux signaux » (avec un conjugué complexe quelque part). Le fait est que l'utilisation de cette méthode réduit considérablement le nombre d'opérations tant que les signaux sont assez longs. Plus précisément, la complexité du calcul est réduite de N^2 à $3 \times N \times \log(N)$, ce qui est beaucoup plus faible si N est suffisamment élevé. Voir les références [3] à [7] pour plus d'informations sur ce sujet.

Et l'autocorrélation ?

Jusqu'à présent, nous avons discuté de la corrélation entre un motif connu et un signal. Mais on peut aussi calculer une cor-

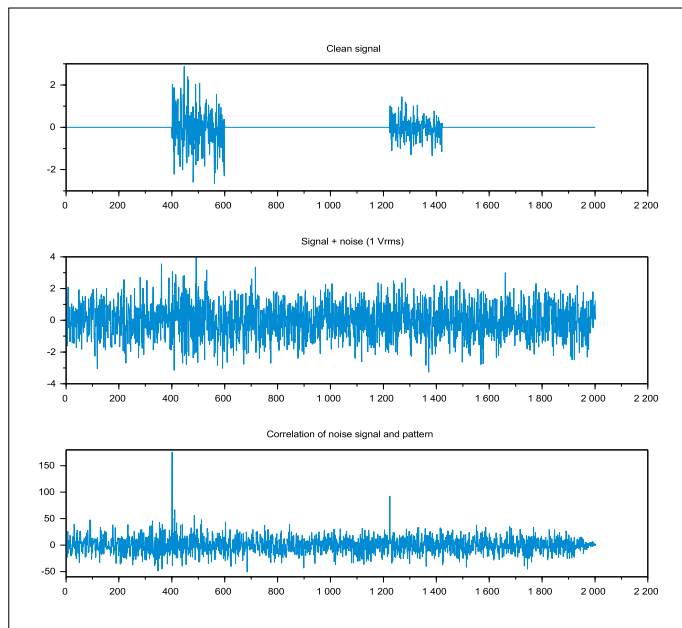


Figure 6. Les meilleurs résultats sont obtenus avec des motifs complexes, ici un signal pseudo-aléatoire. La corrélation fournit des impulsions très nettes quand il y a une correspondance.

relation d'un signal avec lui-même. Cette méthode, appelée autocorrélation, utilise donc le même algorithme, mais avec le signal lui-même comme motif. À quoi cela peut-il donc bien servir ? Encore une fois, gardez à l'esprit l'idée du motif glissant au-dessus du signal. Dans le cas de l'autocorrélation, le calcul identifiera si le signal est corrélé avec une copie de lui-même décalée dans le temps. Donc une autocorrélation permettra d'identifier la période du signal, c'est-à-dire l'instant où il se répète !

Regardez donc la **figure 7** où j'ai pris l'exemple d'un signal *wobulé*. C'est une suite de sinusoïdes de fréquence linéairement croissante. Les radaristes appellent chacun de ces segments un *chirp* (ce qui est vaguement ce que vous entendez si vous envoyez un tel signal sur un haut-parleur : *chiiiirp chiiiirp...*). Malgré le bruit, l'autocorrélation permet clairement de reconnaître la période du signal, c'est-à-dire la période de répétition des *chirps*.

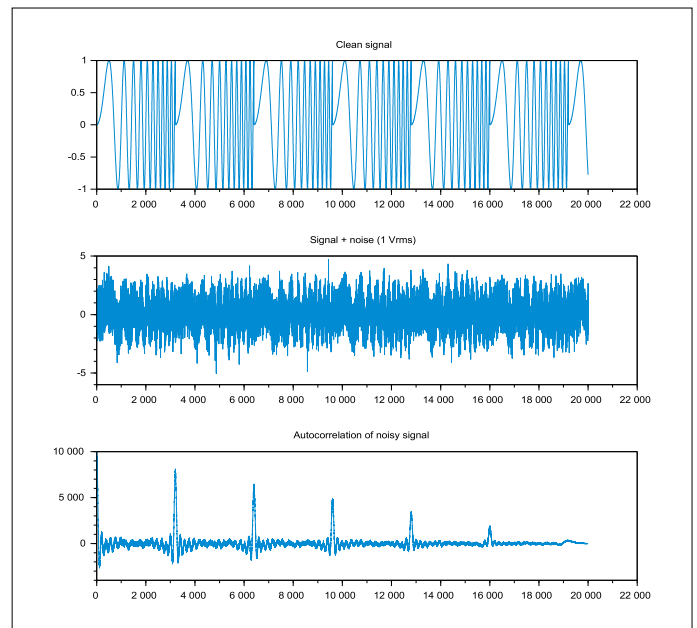


Figure 7. Un exemple d'autocorrélation qui permet de détecter les périodes du signal.

Pour conclure

J'ai fait de mon mieux pour éviter les mathématiques et vous faire comprendre comment un calcul de corrélation peut drastiquement améliorer la performance d'un produit, avec une complexité relativement faible. Retenez qu'il faut penser à une corrélation dès que vous avez besoin de savoir si un motif donné est présent dans un signal. Vous pouvez à ce stade relire mon article sur les filtres à réponses impulsionnelles finies [9], vous devriez retrouver des notions très proches de celles exposées ici... Comme quoi le monde est petit, en particulier dans le domaine du traitement du signal !

Comme d'habitude, je vous encourage à expérimenter. Téléchargez l'outil et reproduisez ces exemples. C'est la meilleure solution pour comprendre réellement ce qui se passe !

(160374)

Cet article a été publié dans la revue Circuit Cellar (n°299, juin 2015).

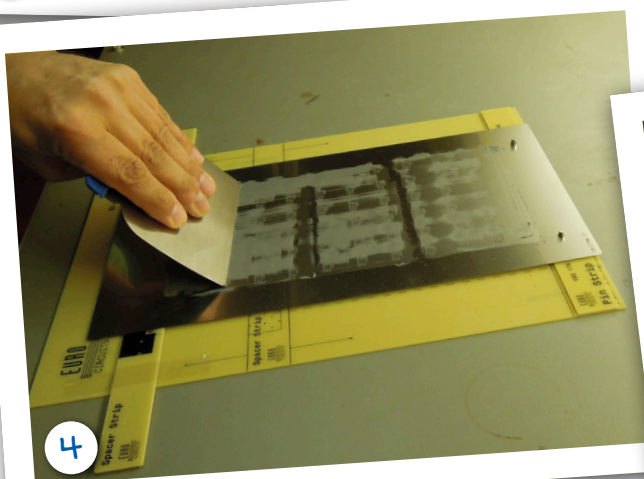
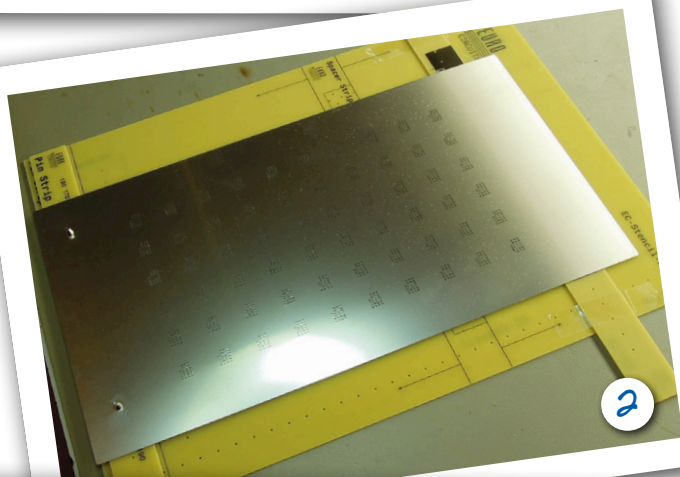
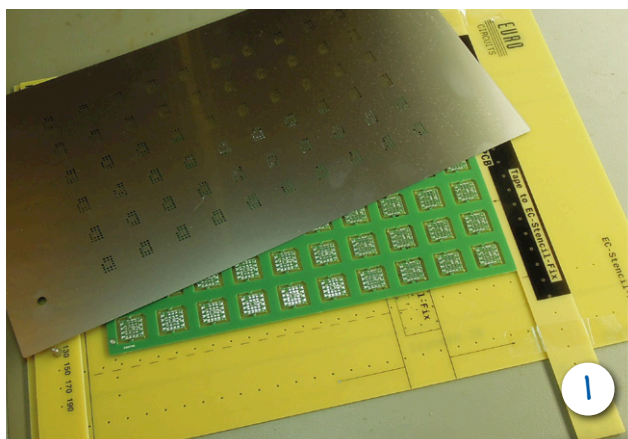
Liens et ressources

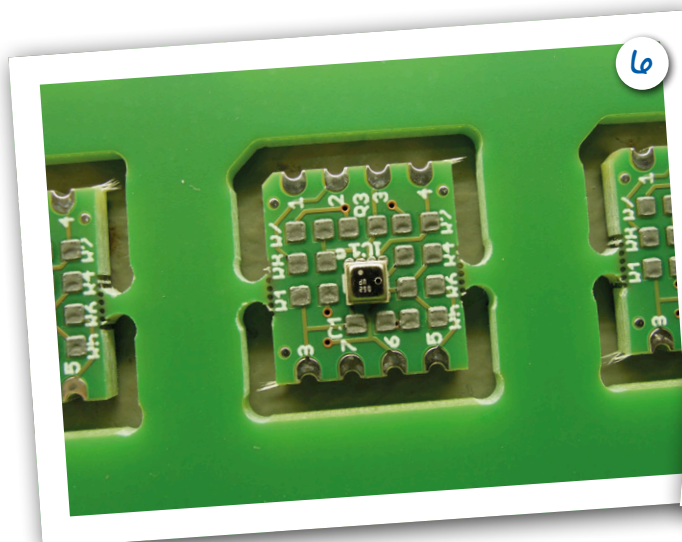
- [1] Définition de la corrélation : www.larousse.fr/dictionnaires/francais/correlation/19435
- [2] Outil de calcul numérique gratuit Scilab : www.scilab.org
- [3] *Numerical recipes in C: The art of scientific computing*, William H. Press & all, Chapter 13.2 *Correlation and auto-correlation using FFT*, ISBN 0-521-43108-5
- [4] *Understanding Digital Signal Processing (3rd Edition)*, Richard G. Lyons, ISBN 978-0137027415
- [5] *Signal processing and correlation techniques*, Peter T. Gough, University of Canterbury, Christchurch, New Zealand : http://pollux.dhcp.uia.mx/manuales/Filtros/UIA_correlation.pdf
- [6] *Digital Signal Processing: A Computer Science Perspective*, Chapter 9 *Correlation*, Jonathan Y. Stein, Editions John Wiley & Sons, Inc., ISBN 0-471-29546-9
- [7] *The Scientist and Engineer's Guide to Digital Signal Processing*, Steven W. Smith : <http://www.dspguide.com/ch7/3.htm>
- [8] Wikipedia, *cross-correlation* : <http://en.wikipedia.org/wiki/Cross-correlation>
- [9] *hors circuits - le filtrage numérique sans stress : les filtres FIR*, Elektor 01/2016 : www.elektormagazine.fr/150635

utilisation de *eC-stencil-fix* pour l'e-BoB avec capteur BME280 et d'autres cartes

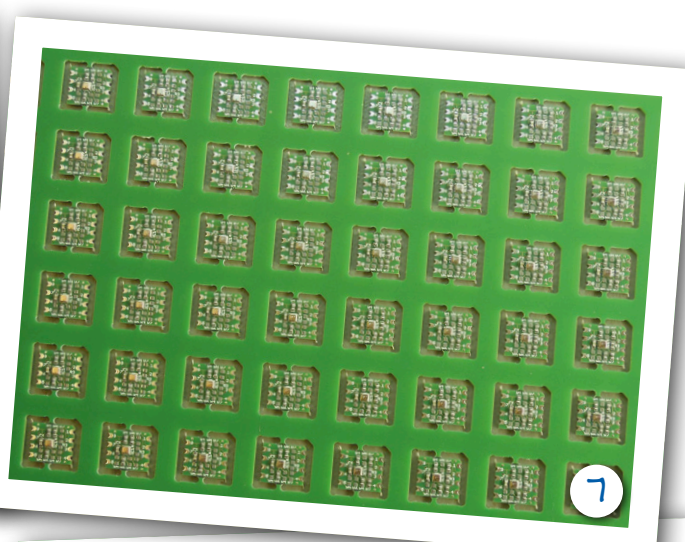
Thijs Beckers (rédaction des Pays-Bas)

Pour approvisionner notre *e-choppe* avec les produits conçus au labo d'Elektor, il est parfois plus rapide, ou plus simple, ou meilleur marché – bref, c'est une meilleure option ! – de les réaliser nous-mêmes, au labo. Ce fut le cas pour l'e-BoB avec le capteur BME280 (Elektor mars/avril 2017, p. 9 [1]). Comme cette réalisation s'y prêtait, nous avons sorti notre *eC-stencil-fix* [2], un dispositif qui facilite l'application de la pâte à braser sur les pastilles des CMS (**photo 1**). Il faut commencer par positionner correctement le circuit imprimé et le pochoir (**photo 2**) ; le circuit imprimé est maintenu en place par deux tenons. *Eurocircuits* a doté le pochoir de deux trous de fixation spéciaux qui permettent son positionnement exact. Deux réglettes dans le même matériau que le circuit imprimé (fournies avec l'*eC-stencil-fix*) permettent de maintenir le pochoir à une distance correcte par rapport à la plaque de base. Sur la photo 2, on voit qu'à gauche la réglette de gauche et le pochoir sont fixés avec deux goupilles en forme de champignon et qu'à droite, la réglette est simplement installée contre le circuit imprimé et arrimée avec de l'adhésif. Les goupilles traversent la réglette pour se planter dans la plaque de base. Il n'y a plus qu'à enduire le pochoir de suffisamment de pâte à souder (**photo 3**) avec l'outil approprié (une spatule convient aussi), de telle manière que les emplacements ad hoc du circuit imprimé en soient recouverts (**photo 4**). L'épaisseur du





6



7

pochoir est calculée pour que la bonne quantité de pâte à souder soit déposée sur le circuit (**photo 5**). On peut alors retirer le pochoir, avec moult précautions !

On place ensuite les composants sur le circuit imprimé (**photo 6**). Ici aussi il faut faire preuve de douceur, car nous travaillons sur un certain nombre de circuits à la fois, leur petite taille le permet (**photo 7**).

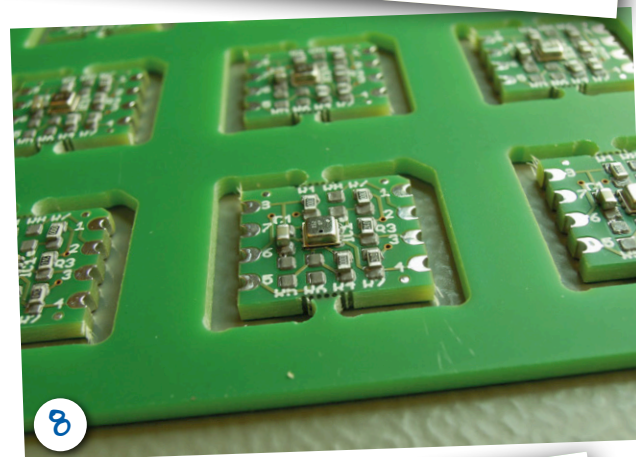
Nous avons réalisé deux versions du circuit, une pour le bus I²C et une pour le bus SPI (**photos 8 et 9**). Les composants diffèrent en fonction de la version, mais le circuit imprimé est le même.

Il ne reste plus qu'à « cuire » le circuit imprimé dans un four à refusion, et les biscuits sont prêts (**photo 10**). Après vérification (le nez sur le circuit), pour repérer d'éventuels circuits mal soudés qu'il faut retirer du lot, direction le stock : nos jeunes magasiniers (et magasinnières) seront à nouveau ravis de pouvoir servir les clients ! ◀

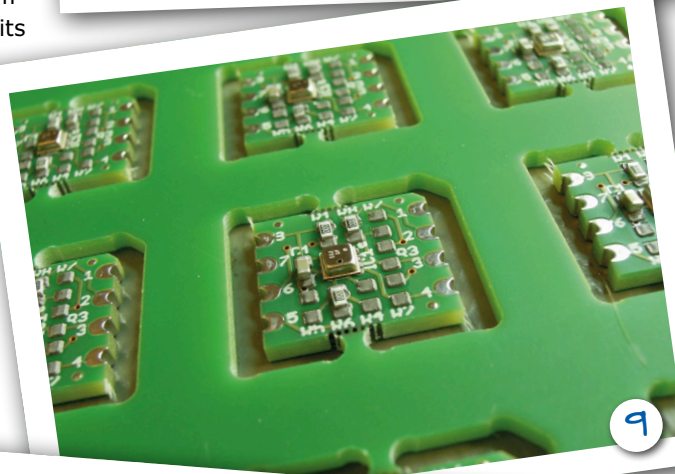
(160297 - version française : Jean-Louis Mehren)

Liens

- [1] www.elektormagazine.fr/160100
- [2] be.eurocircuits.com/shop/offtheshelf/product.aspx?ad=9565&ano=ec-stencil-fix&an=ec-stencil-fix&s=consumables



8



9

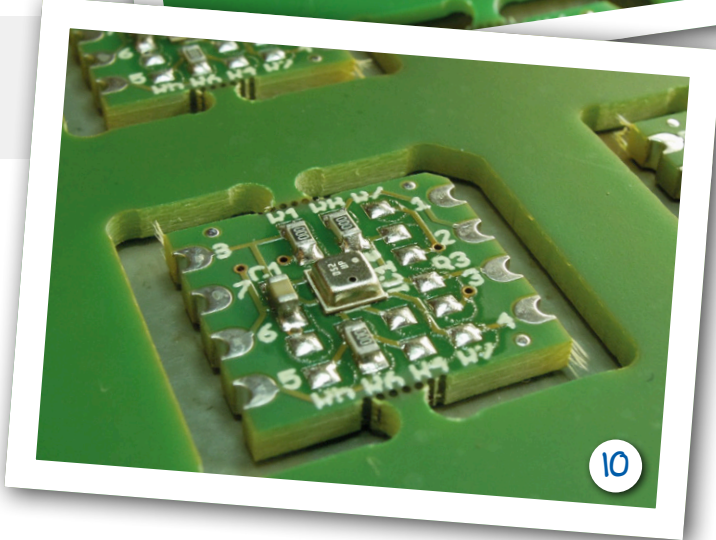
Vous voulez en apprendre plus sur la mise en œuvre d'eC-stencil-fix ?
Nous vous conseillons le petit film suivant sur YouTube : youtu.be/HBWtqZro_fg.

DANS L'E-CHOPPE

→ 160109-2
circuit imprimé du shield BME280

→ 160109-91
module BME280 (version I²C)

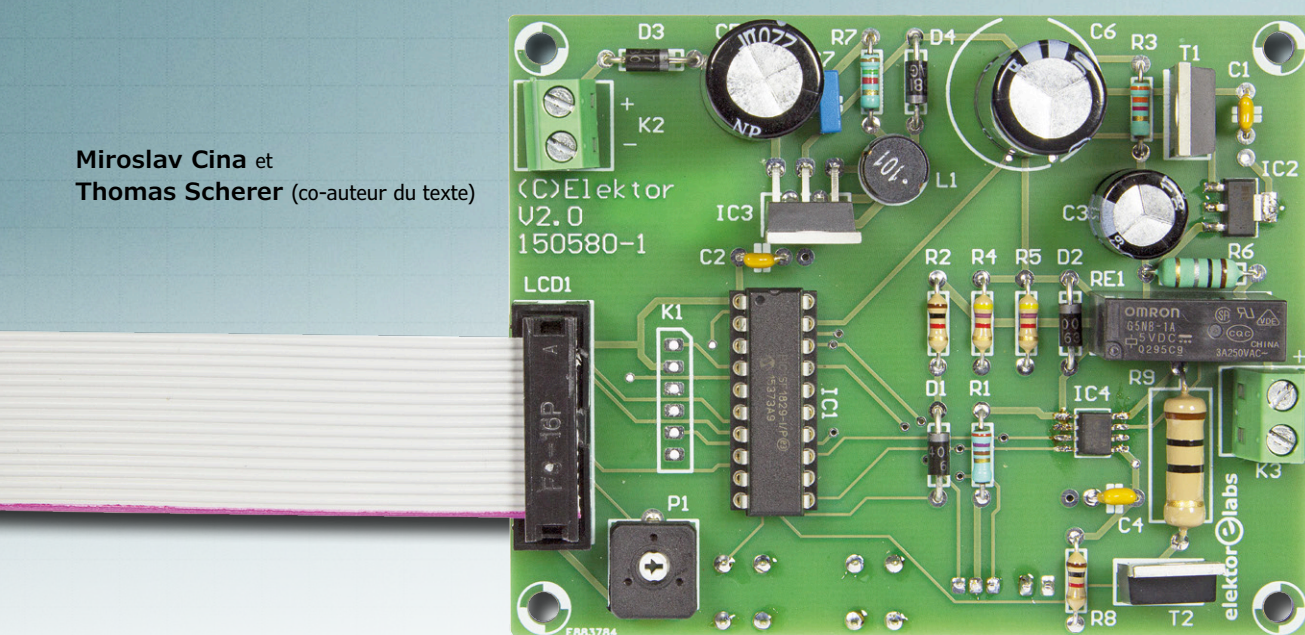
→ 160109-92
module BME280 (version SPI)



10



Miroslav Cina et
Thomas Scherer (co-auteur du texte)



Récemment, Miroslav trébucha sur un monceau de vieux accus pour ordinateurs portables. Devenus inaptes à l'alimentation de ces derniers, ils avaient été mis au rebut. Curieux, il les démonta. L'électronique se limitait à la partie congrue, et il s'avéra que nombre de cellules individuelles étaient encore en parfait état. Pour les utiliser dans des projets personnels, elles requéraient cependant un chargeur adapté. Aussitôt dit, aussitôt fait.

Caractéristiques

- Charge toutes sortes d'accus Li-ion de tension nominale de 3,6 V.
- Mode « formatage » pour essayer de sauver des cellules ayant subi une décharge profonde.
- Mode « décharge » pour déterminer la capacité réelle de l'accum.
- Toutes les informations pertinentes affichées sur l'écran LCD.

Les projets personnels ainsi que nombre d'appareils électroniques du commerce sont alimentés par piles ou accus NiMH. Or les piles se vident vite et les accus sont relativement lourds. Ni les unes ni les autres ne conviennent vraiment dans le cas d'appareils gros consommateurs d'énergie ou stockés à long terme. C'est pourquoi aujourd'hui une multitude d'appareils contiennent des accus à base de lithium, une technologie plus coûteuse et plus complexe. Si l'on trouve des chargeurs automatiques pour accus AA/AAA ou RC en technologie NiCd ou NiMH dans

quasiment n'importe quel supermarché, les chargeurs pour les accus au lithium sont plus rares. En effet les accus Li-ion sont normalement (cf. les ordinateurs portables ou les tournevis électriques) montés dans les appareils en usine, et donc dotés d'une électronique de charge intégrée ou accompagnés d'un chargeur adapté.

Le besoin en chargeurs universels est donc infime puisque tout simplement, on ne trouve pas d'accus au lithium à la caisse des supermarchés. Les cellules au lithium sont produites sous quelques

chargeur d'accus Li-ion

à courbes caractéristiques correctes



formes très réglementées. Destinées à être montées à demeure dans les appareils, elles comportent donc des fils à souder ou des borniers à vis. À cause de leur sensibilité, leur manipulation est ardue, voire dangereuse ; elles n'ont rien à faire dans les mains du grand public. Les types LiPo très courants, caractérisés par un rapport puissance/poids élevé, sont fragiles du point de vue mécanique et « aiment » alors prendre feu. De plus, une cellule NiMH a une tension nominale de 1,2 V. En technologie lithium, il faut s'attendre à une tension nominale de 3,6 V. Les types LiFePo4 plus stables avec leur tension nominale de 3,3 V ne sont pratiquement pas utilisés dans les perceuses, tournevis électriques, ordinateurs portables.

Accus Li-ion

Quel que soit votre projet, désosser de vieux blocs d'accus (**fig. 1**) pour en

extraire les cellules et les réutiliser, ou acheter des cellules Li-ion neuves, il faut un chargeur adéquat. Cela implique de tenir compte de nombreux facteurs, car la (re)charge est bien plus délicate qu'avec des accus NiMH. Il existe de nombreuses différences.

Commençons par un avertissement : il est interdit, contrairement à ce qui est le cas des cellules au plomb, NiCd ou NiMH, de mettre en série des cellules au lithium sans prendre de mesures supplémentaires. L'encadré « Équilibrage » vous dira pourquoi. Le chargeur décrit ici permet d'éviter ces complications, car il est optimisé pour charger une unique cellule au lithium de tension nominale de 3,6 V, valeur classique en technologie lithium, hors technologie LiFePo4.

Outre la tension nominale de 3,6 V, il existe quelques autres valeurs « typiques » à respecter lors de la construction d'un chargeur. En fonction

INFOS SUR LE PROJET



lithium accus
chargeur
recyclage d'accus



débutant
→ **connaissseur**
expert



env. 3 h



Câble FTDI USB/série
Interface pour programmeur PICkit (tous en option)



env. 40 €



Figure 1. Monticule d'accus d'ordinateurs portables mis au rancart, avec leurs cellules au grand jour. Nombre d'entre elles pourront continuer à être utilisées longtemps.



Figure 3. En utilisant un bloc secteur, Miroslav a construit (avec circuit de la figure 2) un premier chargeur Li-ion fonctionnel. Inconvénient : on ne voit pas ce qui se passe.

de leur type, les cellules connaissent une tension minimale de 2,75 à 3 V et une tension maximale de 4,2 V. Si pendant le stockage, la charge et la décharge, la tension de cellule quitte ce domaine, on risque des dommages. En cas de « recyclage » de cellules individuelles, l'obtention de la capacité maximale est secondaire. Pour de nombreuses applications, des cellules à capacité réduite conviennent aussi. On pourra donc aussi essayer de rendre réutilisables des cellules qui ont subi une décharge trop profonde. Pour ce faire, il faudra éviter de les confronter immédiatement avec un courant de pleine charge. Lorsqu'elles présentent une tension inférieure à 2,7 V, il faut commencer par les « formater » avec un courant de charge minimal de quelques mA seulement ; une fois les 2,7 V atteints, charge normale. Selon les circonstances, ce processus peut cependant prendre plusieurs heures. Si après

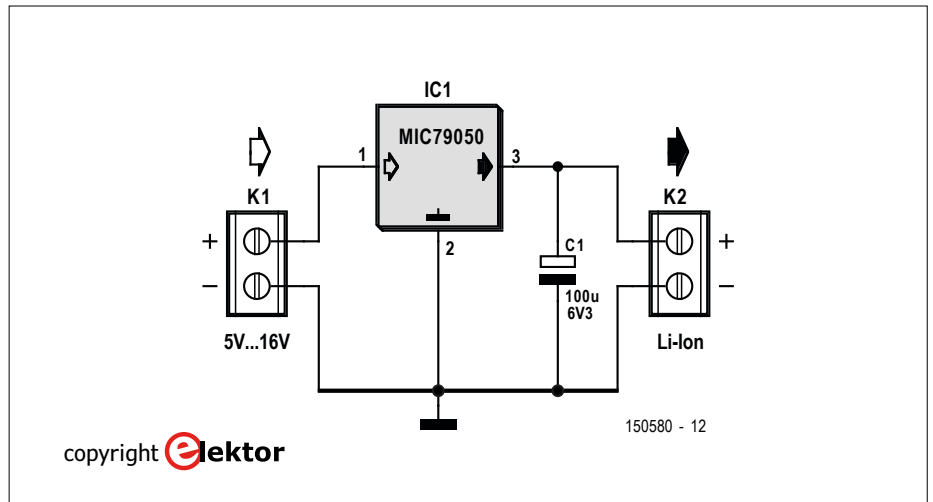


Figure 2. Le circuit minimaliste de ce chargeur pour accus au lithium se débrouille avec un unique circuit intégré tripode (+ condensateur électrochimique).

de nombreuses heures, une cellule ne veut toujours pas dépasser ce seuil, la seule option est un recyclage final, c.-à-d. l'élimination. Il en va de même pour les cellules qui restent stables à 0,0 V, voire qui présentent une polarité légèrement négative (ça arrive dans les montages en série). Dans ce dernier cas, il ne faut pas même tenter un formatage.

Régulation de charge

Dans le cas d'accus au lithium, l'un des facteurs importants est la valeur « C », pas celui de « Capacité », mais un facteur qui donne le courant en fonction de la capacité suivant la formule :

$C = \text{capacité en Ah} / 1 \text{ h}$.
Pour une cellule d'une capacité de 2 300 mAh, à un courant de charge maximal de 1 C correspond un courant maximal de 2,3 A. Une charge à un courant constant de 1 C est classique ; des cellules spéciales ou vieilles n'acceptent parfois que 0,5 C, voire légèrement moins. Des types prévus pour une charge rapide acceptent même jusqu'à 5 C – et atteignent donc leur pleine charge en 20 min. Pas de contre-indication pour une charge à un courant inférieur à l'intensité maximale.

Lors de la décharge aussi ce facteur joue un rôle, mais alors à des valeurs plus élevées. 2 C est quasiment standard et la cellule de type 18650B de Panasonic utilisée dans les voitures de Tesla atteint cette valeur (au cas où vous envisageriez de cannibaliser une Tesla ;-). Mais il y a mieux : les accus LiPo atteignent souvent 5, voire 10 C, et certaines cellules

spéciales peuvent même, brièvement, fournir de l'ordre de 20 C. Pour la charge cela ne joue qu'un rôle secondaire.

De ces caractéristiques découlent directement les exigences posées à un chargeur d'accus au lithium : à des tensions inférieures à 2,7 V, il doit charger avec quelques mA seulement puis, ce seuil atteint, avec un courant constant déterminé plus élevé jusqu'à ce soit atteinte la tension finale de 4,2 V. À partir de là, on se contente d'un maintien de la tension à 4,2 V constants, le courant de charge diminue automatiquement plus la cellule approche du « niveau plein ». Si le courant de charge résultant tombe alors en deçà d'une valeur minimale de, disons 0,05 C, la charge est arrêtée et l'accu est supposé être plein. Important aussi : il est fort possible que des accus endommagés aient été drainés à un point tel que le courant entraînant l'interruption de la charge ne soit jamais atteint. On a besoin dans ce cas-là d'un circuit de temporisation pour éviter une charge « éternelle ».

Solution intégrée

Comme les accus au lithium sont principalement incorporés dans des appareils, il n'est guère surprenant que l'industrie des semi-conducteurs ait dans son programme toutes sortes de contrôleurs de charge intégrés pour simplifier la tâche des équipementiers. L'une des solutions de charge intégrées les plus simples est indéniablement la puce tripode MIC79050S de Microchip. Un chargeur l'utilisant se résume au schéma de

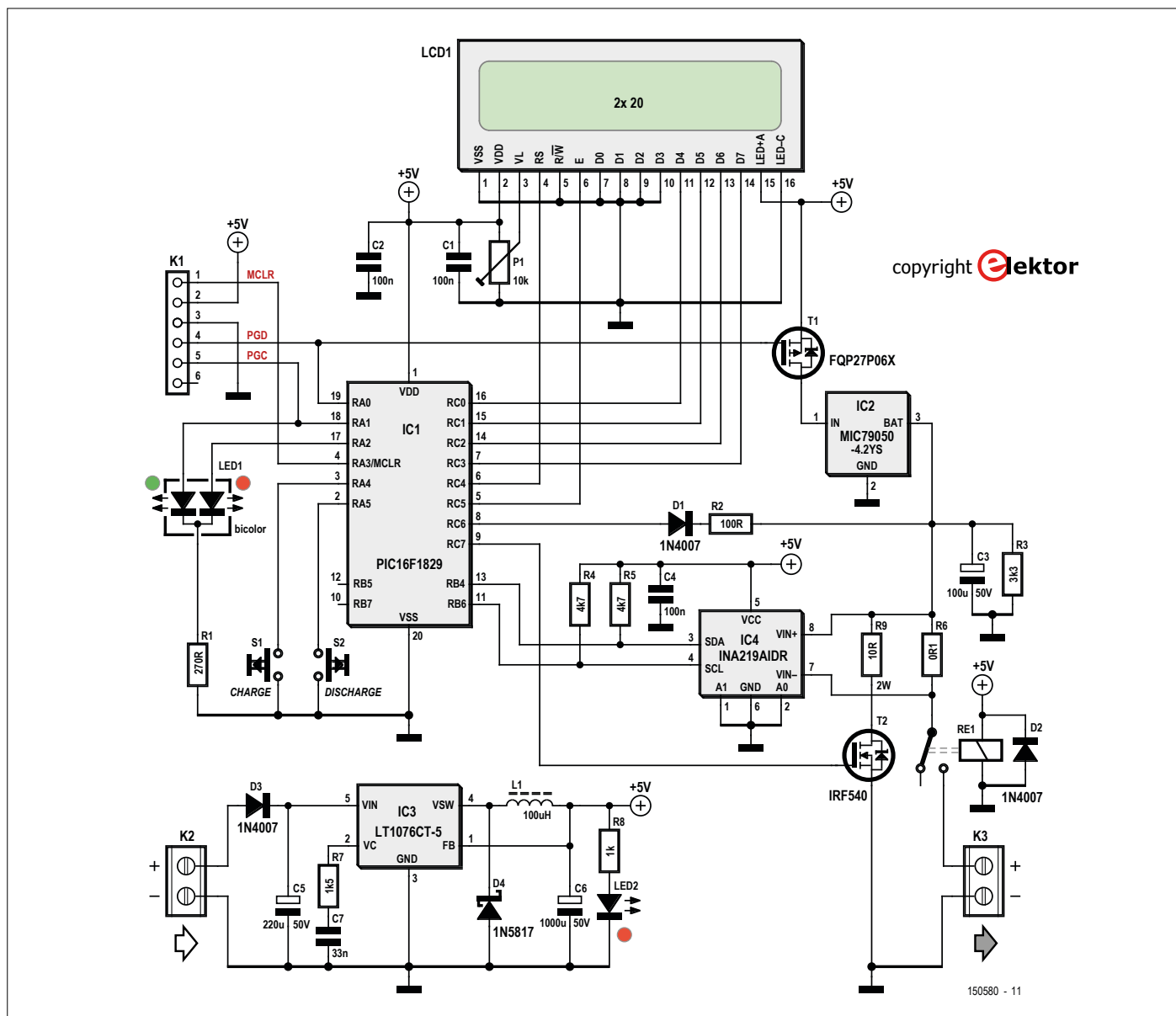


Figure 4. Avec un μC épaulé par un rien d'électronique, le régulateur de charge simple de la figure 2 devient un chargeur pour accu au lithium que l'on peut qualifier « de luxe ».

la **figure 2**. Ce régulateur de charge fournit à sa sortie une tension stabilisée de 4,2 V avec une tolérance (nécessaire) de 0,75%. Son courant de charge est de 0,5 A – pas très intense, mais il convient à la grande majorité des cellules. Une cellule courante de 2 à 3 Ah se charge donc en 4 à 6 h. Outre une protection thermique intégrée, le CI n'intègre pas grand-chose. Il est disponible en boîtier SOT223-3 facile à souder et ne requiert rien de plus qu'une tension d'entrée entre 5 et 16 V, sous 0,5 A. Il est recommandé de rester dans la plage de tension inférieure, sinon le régulateur linéaire dégage beaucoup d'énergie et chauffe donc. Miroslav a donc réalisé, à titre de test, un chargeur d'accu au lithium simple, constitué d'un

ancien bloc d'alimentation de portable Nokia (5,7 V / 0,8 A) et d'un morceau de carte conçue à cet effet (cf. **fig. 3**). Le test fut concluant.

Ce chargeur simplissime fonctionne, mais présente cependant un inconvénient majeur pour l'électronicien : à un moment quelconque, l'accu est probablement plein, mais on ne voit pas ce qui se passe alors. Pour cela, il faut un peu plus d'électronique, ce qui explique pourquoi Miroslav a développé un « vrai » chargeur.

Version de luxe

De nos jours, plus de luxe signifie l'utilisation d'un microcontrôleur (μC) pour épauler une fonction de base. C'est le cas ici : la **figure 4** montre le circuit

complet d'un chargeur confortable pour les cellules Li-ion commerciales. Le μC multifonction, un PIC16F1829, contrôle la charge proprement dite, mesure la tension et le flux de courant, pilote une LED d'état bicolore, interroge deux boutons et affiche toutes les données importantes sur un écran LCD.

Mais ce n'est pas tout : pour éviter des pertes d'énergie thermique inutiles, le circuit est doté d'un régulateur à découpage, IC3. Rassurez-vous, vous n'aurez pas à bobiner vous-même l'inductance requise, L1, elle est disponible toute faite. D3 fait office de protection contre une inversion de polarité. Comme il faut à IC3, pour une tension de sortie de 5 V, une tension d'entrée comprise entre 8 et 60 V, l'alimentation du circuit pourra

prendre la forme d'un bloc secteur, non régulé avec une capacité de charge de quelque 5 W. Sa tension de sortie doit être comprise entre 9 et 36 V. L'idéal : 12 V et 0,5 A. Pour D4, utilisez impérativement une diode Schottky rapide pouvant encaisser 1 A. La remplacer par une diode au silicium d'« occase » est une mauvaise idée. LED2 s'allume lorsque IC3 fournit le 5 V.

IC1 détecte l'état de S1 et S2, les deux boutons-poussoirs, via les broches d'E/S RA4 et RA5. Le schéma est si spartiate parce que le μ C a activé des résistances de polarisation haute internes à ses entrées. RA1 et RA2 pilotent la LED bicolore (cf. encadré « **LED bicolore et état** »). LCD1 est piloté en mode quartet classique et, selon son type, comporte deux lignes de 20 ou 24 caractères. P1 règle le contraste. Le rétroéclairage est relié directement au +5 V (broches 15 et 16). Si vous optez pour un autre afficheur que celui de la liste des composants, assurez-vous d'utiliser un modèle à résistance série intégrée. T1 est un transistor MOSFET de puissance à canal P ; il active et coupe l'alimentation du régulateur de charge proprement dit, IC2. La mesure du courant se fait, à travers un *shunt* de faible valeur (R6), à l'aide d'un circuit intégré spécialisé, un INA219. IC4 est relié à IC1 par I²C. IC4 est ici une version moins chère et moins précise, un INA219B. T2 sert à raccorder la résistance de charge R9 à l'accu pour le décharger. Grâce à une fonction de décharge activée avec S2, il

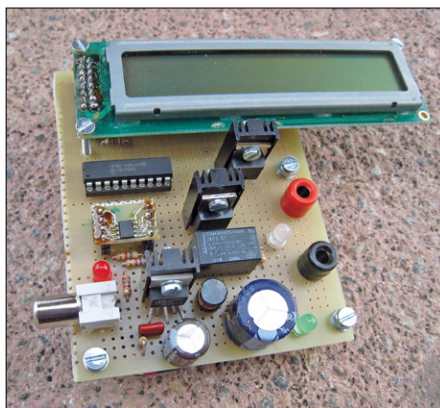


Figure 5. Photo du premier prototype du chargeur de luxe de Miroslav, sur platine à trous.

est possible de déterminer de façon automatique la capacité réelle d'un accu à tester (chargé à 100%).

Le « formatage » d'un accu ayant subi une décharge trop profonde se fait par un niveau haut sur la broche RC6 de IC1. R2 et la résistance interne de la broche du μ C limitent le courant ; D1 empêche tout reflux de courant. L'intensité du courant de formatage est fonction de la tension de la cellule. Exemple : à 2,3 V on a une tension directe de D1 et de la tension de cellule = 2 V sur R2. Il circule donc un courant maximal de 20 mA. Le μ C ne peut pas fournir plus de 25 mA.

À quoi sert RE1 vu qu'il n'est pas piloté par le μ C ? C'est simple : le relais ne colle qu'à condition que le chargeur soit alimenté. Si le bloc secteur est débranché, l'accu, s'il est présent, est déconnecté ce qui évite qu'il ne se « rétrodécharge » accidentellement via IC3 et IC4 ; on élimine ainsi de manière fiable tout risque de décharge profonde non intentionnelle.

Logiciel

Pas de microcontrôleur sans micrologiciel. Ses tâches sont : déterminer l'état actuel, mesurer le courant et la tension, interroger les boutons-poussoirs, piloter LED1, T1 et T2, compter le temps et afficher des informations pertinentes sur l'écran LCD. Le micrologiciel a été écrit en assembleur et est disponible gratuitement au téléchargement (code source et fichier hex en [1]). Jetons un coup d'œil aux parties

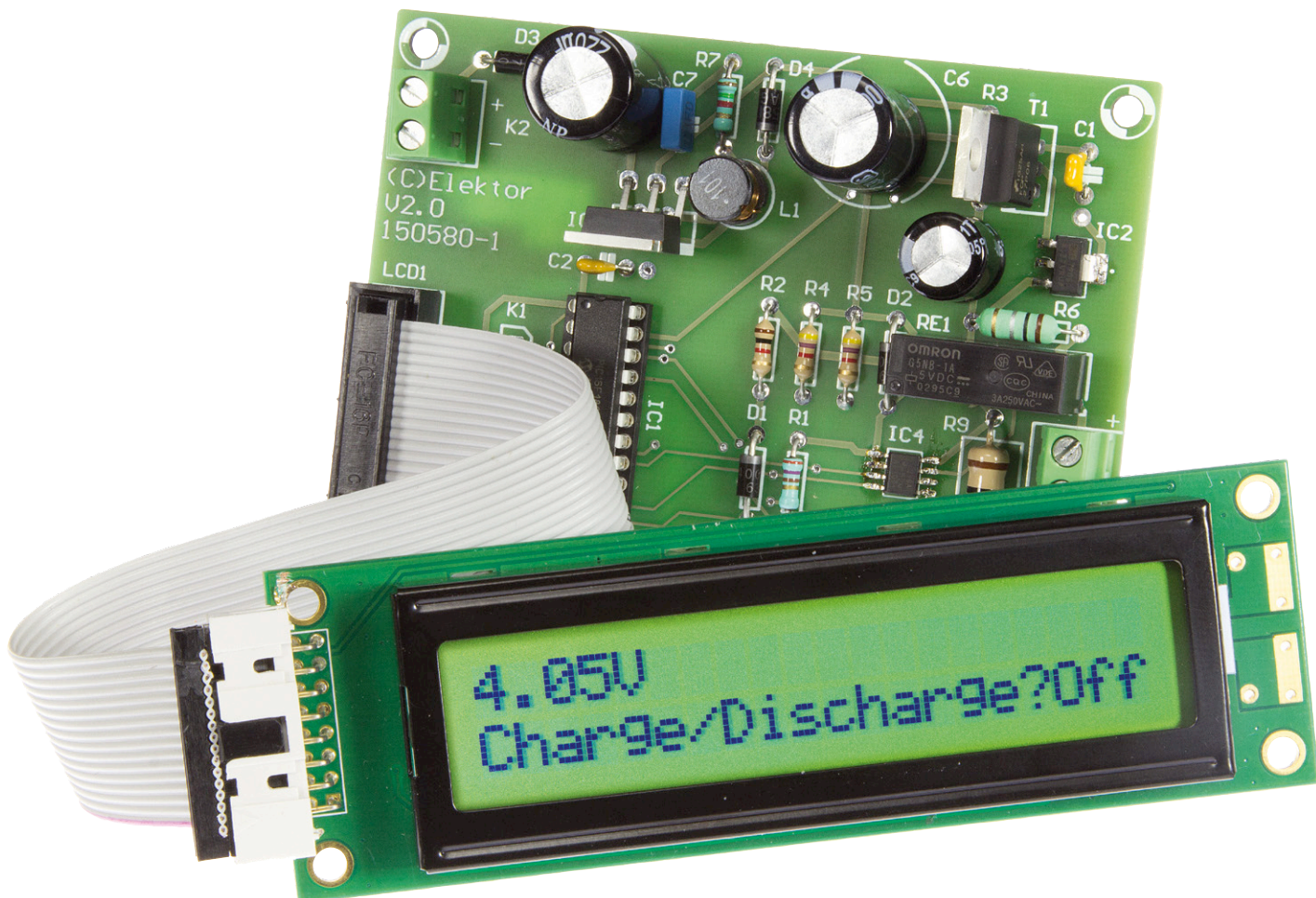


Équilibrage

Contrairement aux technologies d'accus classiques, le matériau des cellules au lithium est sensible aux surcharges et aux décharges profondes ; elles peuvent l'endommager de façon irréversible. Dans un circuit en série sans surveillance électronique explicite (précisément un équilibrage), la cellule de la capacité la plus faible se retrouvera, lors de la charge, en « dépassement de capacité » (tension de cellule supérieure à la valeur admissible) alors que la charge des autres cellules doit se poursuivre. Conséquence : outre une augmentation de la température et le risque d'incendie de nombreux modèles, une autre conséquence est la dégradation de la cellule d'accu concernée avec une nouvelle réduction de sa capacité. Idem pour la décharge : la cellule ayant la capacité la plus faible tombe, en l'absence

de surveillance, dans des plages de sous-tension critique, d'où perte de capacité. Après plusieurs cycles de charge/décharge, le problème empire rapidement.

D'où la règle impérative : ne jamais mettre en série des cellules au lithium sans ce que l'on appelle un équilibreur ou un circuit électronique spécialisé dans la gestion d'accus. Ceci est vrai tant pour la décharge que pour la charge. On trouve dans le commerce des équilibreurs pour différents nombres de cellules, diverses capacités et chimies de lithium. La thématique de l'équilibrage (*balancing*) est complexe et sort du cadre de cet article. De nombreuses applications ne requièrent qu'une seule cellule, de sorte que l'on peut passer outre la thématique de l'équilibrage.



► recyclage d'accus au lithium à l'aide d'un chargeur Li-ion universel

pertinentes :

d'abord le logiciel initialise le compteur de temps (par des interruptions et TIMER2) ; ensuite démarre la boucle principale.

La routine `charge_main` s'occupe du processus de charge. Dans la boucle principale, on a, via I²C, par l'intermédiaire de INA219, lecture et traitement des valeurs actuelles de courant et de tension. Si,

tout au début, on se trouve en présence d'une tension de cellule inférieure à 2,7 V, le processus lance d'abord le formatage. Pendant ce dernier, T1 reste bloqué. Si, au cours du formatage, la ten-

LED bicolore et état

Les états de fonctionnement du chargeur d'accus Li-ion sont visualisés par LED1, une LED bicolore. Liste des affectations :

- LED1 = **éteinte**. Le chargeur n'est pas actif. Il n'y a ni charge ni décharge de l'accu.
- LED1 = **orange** (R et V allumées simultanément). Processus de formatage en cours
- LED1 = **rouge**. Phase de charge. Le courant de charge est > 200 mA.
- LED1 = **verte**. Phase de charge. Le courant de charge est compris entre 80 et 200 mA.
- LED1 = **rouge clignotant**. Phase de décharge.

Il en résulte qu'avec un chargeur en fonction et un accu connecté, il ne se passe d'abord rien et seule LED2 s'allume. Après une action sur S1, la LED1 s'allume (tant que la tension de cellule est inférieure à 2,7 V) en orange. Au-delà de ces 2,7 V, elle vire au rouge si le courant est > 0,2 A, et passe au vert quand il tombe en deçà de cette intensité. Lorsque le courant tombe en dessous de 80 mA, la LED s'éteint et la charge est terminée.

Lors d'une action sur S2, la LED clignote jusqu'à ce que la tension de décharge finale de 3 V soit atteinte. Il est possible, à tout moment, d'interrompre tous les processus par une pression simultanée des deux boutons-poussoirs.

sion passe au-delà de 2,69 V, ce dernier est stoppé et, via T1 et MIC79050 on démarre le processus de charge normale. Pendant la charge normale, la tension et le courant sont mesurés périodiquement. La boucle principale se termine lorsque le courant de charge tombe en dessous de 80 mA ou lorsque le temps de charge total a dépassé 30 h. On peut modifier ces deux valeurs dans le micrologiciel. La valeur de la durée de charge maximale se trouve dans le sous-programme `mm02` à la ligne :

```
movlwD'030'
; maximal charging time (hours)
```

La valeur du courant minimal se trouve dans le sous-programme `curr_an` à la ligne :

```
xorlwD'080'
; keep charging above 080 mA
```

On peut aussi modifier le seuil pour la commutation du courant de charge élevé à un courant de charge faible dans le même sous-programme.

```
xorlwD'200'
; green LED off above 200 mA
```

Le sous-programme `discharge_main` s'occupe de la décharge qui s'interrompt lorsque la tension de l'accu tombe en dessous de 3 V. Lors de la décharge, le facteur pertinent pris en compte n'est



LISTE DE COMPOSANTS

Résistances :

(par défaut 5%, 1/4 W, 250 V)

R1 = 270 Ω
R2 = 100 Ω
R3 = 3,3 k Ω
R4, R5 = 4,7 k Ω
R6 = 100 m Ω , 1 W
R7 = 1,5 k Ω
R8 = 1 k Ω
R9 = 10 Ω , 2 W

Condensateurs :

C1, C2, C4 = 100 nF / 50 V, céramique, pas de 5,08 mm
C3 = 100 μ F / 50 V, condensateur électrochimique, vertical, pas de 3,5 mm, \varnothing 8 mm
C5 = 220 μ F / 50 V, condensateur électrochimique, vertical, pas de 5 mm, \varnothing 10 mm
C6 = 1000 μ F / 50 V, condensateur électrochimique, vertical, pas de 7,5 mm, \varnothing 16 mm
C7 = 33 nF / 50 V, céramique, pas de 5,08 mm

Inductance :

L1 = 100 μ H, 190 m Ω , 900 mA, verticale, MCSCH895-101KU

Semi-conducteurs :

D1 à D3 = 1N4007
D4 = 1N5817*
LED1 = bicolore, 5 mm, cathode commune
LED2 = rouge, 3 mm
T1 = FQP27P06X, canal P, 70 m Ω
T2 = IRF540NPBF, canal N, 44 m Ω
IC1 = PIC16F1829-I/P, programmé*
IC2 = MIC79050-4.2YS, TO223-3
IC3 = LT1076CT-5
IC4 = INA219AIDR*

Divers :

LCD1 = module LCD, avec rétroéclairage, MC22005A6W-GPTLY de Midas par ex.*
Embase mâle, 1x6, verticale, pas de 2,54 mm
2 borniers à vis encastables, à deux pôles, pas de 5,08 mm

2 boutons-poussoirs encastables, à contact travail, 6x6 mm
Relais, 5 V, unipolaire à contact travail, 5 A, Omron G5NB1A5DC par ex.

Support pour CI à 20 contacts, DIP, pour IC1
Circuit imprimé 150580-1 V2.0*

* cf. texte

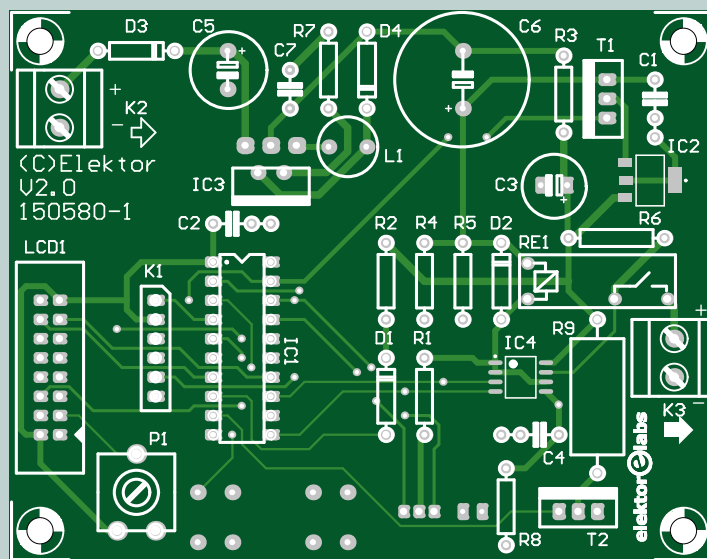
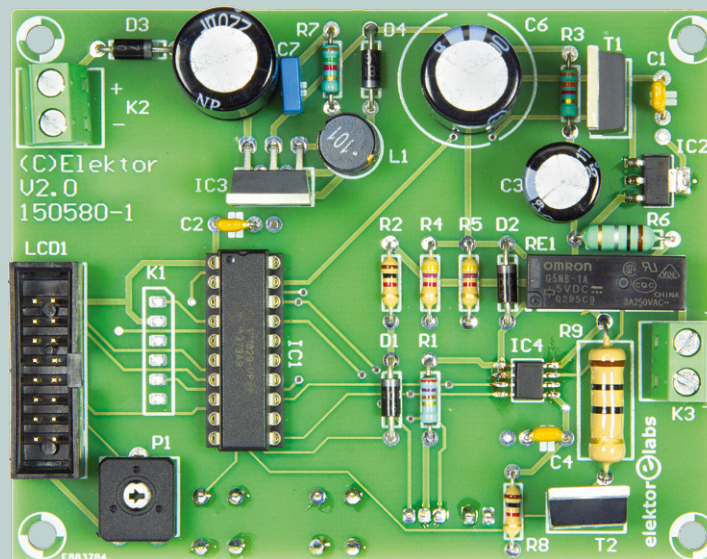


Figure 6. Grâce au circuit imprimé, la réalisation du chargeur d'accus au lithium de luxe devient un jeu d'enfant.



pas le temps, mais la quantité de courant fournie par l'accu = capacité en mAh. Il n'y a pas de durée d'attente (*timeout*). La mesure proprement dite se fait, en mA, dans le sous-programme `inc_q`.

Mise en œuvre

Après la **mise sous en fonction**, la version du micrologiciel s'affiche brièvement ; ensuite apparaissent les informations sur l'accu connecté. L'affichage reproduit ici est celui de l'afficheur à 2×20 caractères :

```
U = 3.10V
CHARGE? / DISCHARGE?
```

Maintenant on peut démarrer la charge avec S1 ou la décharge avec S2.

Pendant une charge normale (tension de cellule > 2,7 V), l'écran commence par afficher des informations du genre :

```
4.16V/+203MA/1736MAH
05:02:29 25
```

La première ligne donne les tension et courant actuels, ainsi que la quantité totale de courant chargé [en mAh]. La seconde ligne donne le temps déjà écoulé [en hh:mm:ss] et à droite les heures restant encore jusqu'à la fin du délai d'attente (25 h dans le cas présent).

Si le courant chute en dessous du **seuil de 200 mA**, l'affichage devient :

```
4.16V/+199MA/1758MAH
05:05:29-00:05:24 25
```

Au milieu de la seconde ligne apparaît le temps de charge écoulé à courant réduit. Lorsque la cellule est **complètement chargée**, la LED1 s'éteint et on peut avoir l'écran suivant :

```
4.16V 1807MAH
05:05:29-00:10:32 OFF
```

Sur la seconde ligne, on trouve maintenant au lieu du délai d'attente, l'information « Off ».

Si la tension de la cellule est trop faible pour démarrer une charge normale, le **processus de formatage** est lancé avec l'affichage suivant :

```
1.28V/+016MA/0000MAH
FORMAT 00:01:43 30
```

Dans la ligne du haut, en plus de la tension de la cellule et du courant, la quantité totale de courant sera très faible. La ligne du bas indique la durée du processus en cours et la durée maximale que pourrait encore prendre la charge.

Pour la **décharge** via S2, l'accu se décharge sur R9. Avec un accu bien chargé, le courant qui commence à circuler est de l'ordre de 360 mA. On lit donc quelque chose du genre :

```
3.72V/-364MA/0759MAH
00:46:10
```

Dans la ligne du haut, on trouve donc après les valeurs actuelles de tension et de courant (signe négatif = décharge), la quantité de courant consommée [en mA]. La ligne du bas donne le temps écoulé.

Montage et observations

La **figure 5** montre le prototype de Miroslav réalisé sur une platine à trous. L'utilisation de la platine (**fig. 6**) disponible dans l'e-choppe d'Elektor [1] simplifie bien les choses. Vu leur nature, tous les composants sont faciles à souder, même IC2 dans son boîtier CMS. IC1 est disponible préprogrammé (avec le micrologiciel pour un écran de 2×20 car.). Si vous souhaitez travailler avec un afficheur à 2×24 caractères, optez pour l'autre micrologiciel du téléchargement. Avant d'utiliser un LCD de substitution, assurez-vous qu'il soit doté d'un contrôleur compatible HD44780. Le résultat de votre travail devrait ressembler au prototype du labo d'Elektor (**fig. 7**).

Le branchement d'un accu demande une certaine prudence. En cas d'inversion de sa polarité, un accu chargé risque fort d'endommager IC4 vu que le circuit ne supporte, sur ses entrées de mesure, qu'une tension négative faible. Pour éviter tout risque de ce genre, intercalez le circuit de protection de la **figure 8** entre le chargeur et la cellule d'accu. En cas d'erreur de polarité lors du raccordement de l'accu, la diode conduit et le fusible « saute ». Pour une réaction immédiate, le fusible doit être du type à action rapide. Et pour qu'en cas d'erreur, la tension reste la plus faible possible il est souhaitable d'utiliser pour D une diode Schottky capable de supporter une intensité d'au moins 3 A.

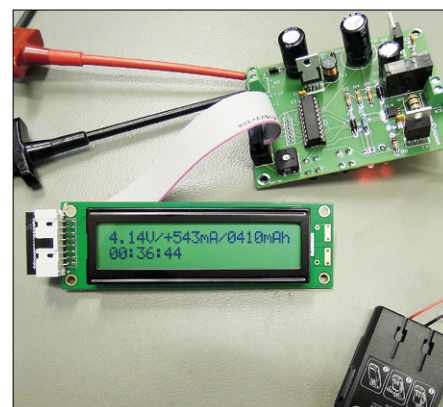


Figure 7. Le prototype du labo d'Elektor réalisé sur le circuit imprimé de la figure 6.

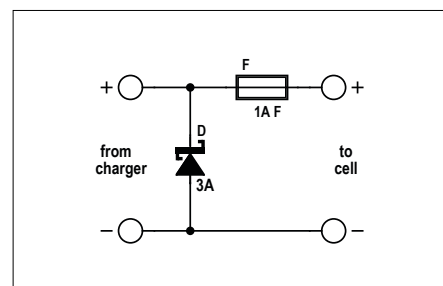


Figure 8. Ce circuit de protection simple prévient tout risque de dommages en cas d'inversion accidentelle de la polarité de l'accu.

Depuis que Miroslav a construit son chargeur Li-ion, il a déjà remplacé les accus Ni-MH d'un certain nombre d'appareils par des cellules au lithium en améliorant ainsi leurs caractéristiques de manière significative. Cela lui permet d'affirmer que le chargeur a vraiment fait ses preuves au quotidien. Pour des questions, commentaires et suggestions, il est à tout moment à l'écoute à l'adresse de courriel miroslav.cina@t-online.de. ❏

(150580 – version française : Guy Raedersdorf)

Lien

[1] www.elektor.fr/150580

DANS L'ÉCHOPPE

→ 150580-1
Circuit imprimé

→ 150580-41
Microcontrôleur programmé,
version pour 2×20 car.



questions d'éthique

avons-nous besoin d'une loi sur les robots ?

Les parlementaires européens demandent une réglementation européenne pour régir les aspects éthiques et juridiques des robots et de l'intelligence artificielle. Peut-on réglementer ces technologies émergentes alors qu'elles sont encore en développement ? Doivent-elles encore mûrir avant d'être examinées par les législateurs ?

Tessel Renzenbrink (Pays-Bas)

En février dernier, le Parlement européen (PE) a adopté une résolution qui appelle la Commission européenne (CE) à préparer un cadre législatif pour réglementer la robotique et l'intelligence artificielle (IA). Cette résolution compte 19 pages de recommandations qui détaillent ce qu'un tel cadre devrait couvrir. De nombreux domaines sont abordés notamment l'innovation, la standardisation, les principes éthiques, l'emploi, l'éducation et des règles de responsabilité. Voici dans le détail ce que le Parlement déclare à propos de ces sujets :

Innovation

L'Europe est à la pointe dans la recherche et le développement en IA et robotique. Pour garder sa position de chef de file, l'UE devrait augmenter l'enveloppe destinée à la recherche et promouvoir la science ouverte et l'innovation éthique responsable. Deuxièmement, une infrastructure numérique solide est la base du développement de ces technologies. L'UE devrait donc chercher à atteindre la connectivité omniprésente et l'accès universel et maintenir le principe de neutralité de l'internet. Enfin, l'interopérabilité entre systèmes est essentielle pour des flux de données en temps réel qui permettent aux robots et à l'IA de devenir plus flexibles et plus autonomes. Les standards ouverts et la transparence devraient être mis en avant pour éviter l'enfermement dans des systèmes propriétaires qui réduisent l'interopérabilité.

Standardisation

La standardisation et l'interopérabilité sont indispensables pour stimuler une saine compétition dans les domaines de la robotique et de l'IA. L'UE devrait donc se concentrer sur l'harmonisation internationale de normes techniques en collaboration avec les organes européens et internationaux de standardisation. Plus proche de la réalité européenne, un règlement valable dans toute l'UE devrait éviter une fragmentation du marché intérieur et empêcher le développement de systèmes incompatibles dans les différents États membres. Les *makers* et autres inventeurs seront heureux d'apprendre que le PE insiste sur l'importance de la rétro-ingénierie licite et appelle à la création de « zones d'accueil des bricoleurs » où des expériences avec des robots sont autorisées.

Principes éthiques

Les lois de la robotique d'Asimov ne peuvent être converties en langage machine, elles doivent donc être comprises et transmises aux concepteurs, ingénieurs, fabricants et opérateurs de robots, selon le PE. Il propose un code de conduite pour tous ceux qui sont actifs dans ce domaine. Ils devraient adhérer aux quatre principes suivants :

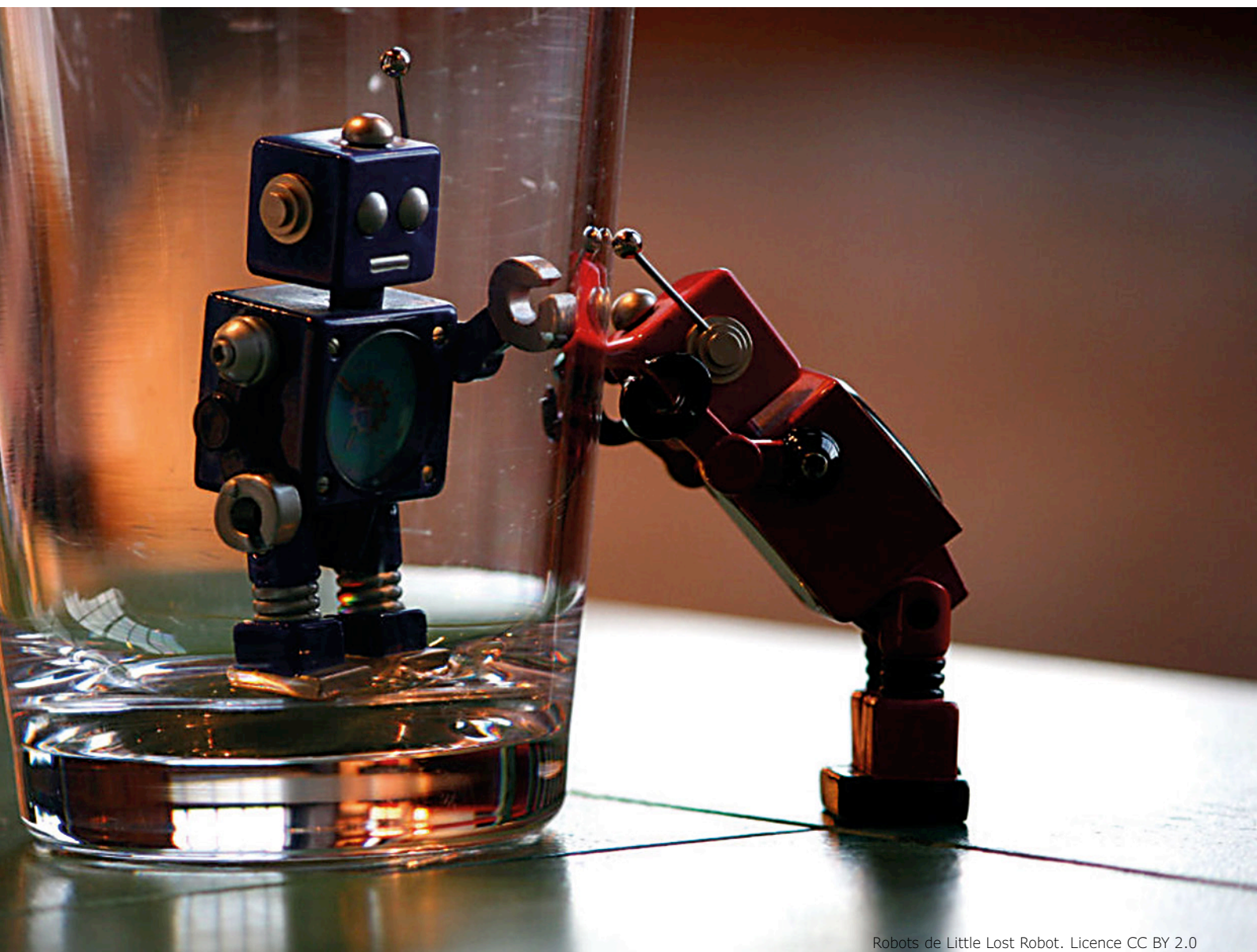
1. les robots doivent agir pour servir les intérêts des humains ;
2. les robots ne doivent pas blesser un humain ;
3. les humains devraient avoir la possibilité de prendre une décision éclairée et non forcée sur leur interaction avec les robots ;
4. les bénéfices de la robotique devraient être distribués de façon équitable, en particulier en ce qui concerne les robots utilisés dans les soins (services de santé et à domicile).

De plus, le PE appelle à considérer les risques potentiels auxquels la population et la société en général peuvent être exposées par l'IA et la robotique. Le développement et l'utilisation de robots doivent donc toujours respecter les valeurs européennes inscrites dans le traité sur l'Union européenne y compris la dignité humaine, l'égalité, la justice et l'équité, la non-discrimination, le consentement éclairé et la protection des données.

Emploi & éducation

L'aspect des robots qui a été le plus discuté dans le public est la perte d'emplois due à la robotisation. Si de nombreux emplois disparaissent sans substitution, nous devons repenser l'organisation des sociétés. L'emploi comme moyen de distribuer les richesses est une notion qui n'aurait plus de sens. Dans ce contexte, le PE demande à la CE de surveiller cette tendance et d'analyser les conséquences éventuelles pour la viabilité des systèmes de sécurité sociale des États membres.

En ce qui concerne l'éducation, le Parlement relève un manque grandissant de professionnels des TIC et que 90% des emplois demandent des connaissances de base en informatique. Il recommande à l'UE de développer des systèmes de formation et d'éducation flexibles pour aider les Européens à se former



Robots de Little Lost Robot. Licence CC BY 2.0

et se préparer à la numérisation et l'automatisation grandissantes de l'économie.

Responsabilité

Un autre sujet souvent abordé est la question de la responsabilité. Qui est responsable des dommages causés par un système s'il est autonome et autodidacte ? Le fabricant ? Le propriétaire ? Le robot lui-même ? Si ce problème n'est pas résolu, il peut entraver l'innovation, car les fabricants et les utilisateurs finaux manquent de sécurité juridique. Le PE demande à ce sujet à la CE de proposer un instrument législatif pour aborder la question de la responsabilité.

Finalement, le PE demande la création d'une agence européenne pour la robotique et l'IA qui servirait de centre d'expertise pour atteindre les buts mentionnés plus haut.

Débat public

La Commission n'est pas obligée de suivre la proposition du PE, mais elle doit indiquer ses raisons si elle ne le fait pas. Au moment de la rédaction de cet article, la CE n'a pas encore répondu. La CE acceptera-t-elle cette résolution ? La rejettera-t-elle ?

Les opposants déclarent qu'il est trop tôt pour une loi sur les robots. Ce domaine est en pleine émergence et nous en savons trop peu sur le développement de la robotique. Ils sont également opposés à l'association de l'IA et de la robotique qui sont deux domaines différents. En fait, il y a de vastes différences au sein même de la robotique. Un Roomba ne nécessite pas la même surveillance qu'un robot de soin pour personnes âgées. L'objection la plus importante est que trop de régulation va étouffer l'innovation.

Les partisans de cette mesure ne sont pas de cet avis. Ils avancent qu'un cadre commun va encourager l'innovation. Premièrement, parce que cela apportera une sécurité juridique pour les fabricants et utilisateurs finaux. Deuxièmement, parce que cela harmonisera les normes technologiques entre les États membres. La question de savoir si nous avons besoin maintenant d'une législation pour la robotique fait débat. Toutefois c'est le moment d'y penser. Plutôt que d'être submergé par la prochaine vague d'avancements technologiques, nous devons tenir un débat public sur la place à attribuer à la robotique et à l'IA. Il faut déterminer de manière démocratique comment notre société sera transformée par ces technologies émergentes. ◀

(160334 – version française : Thierry Destinobles)



bienvenue dans votre e-choppe

la rédaction recommande



TL866A Universal Programmer

Les microcontrôleurs modernes, équipés de mémoire flash, prennent en charge eux-mêmes leur programmation *in situ*. Pas besoin de programmeur externe. Pourtant, il n'y a pas si longtemps, tout amateur d'électronique éclairé rêvait d'un appareil pour programmer à la fois les PROM, les EPROM, les GAL et les microcontrôleurs. Plus il accepterait de circuits différents, plus il serait apprécié. Jadis, un tel merle blanc, si tant est qu'il existât, était hors de portée pour la plupart d'entre nous. Aujourd'hui, pour votre labo, il y a le programmeur

universel *MiniPro TL866A* pour une somme modique. Ce programmeur convient parfaitement aux amateurs pour l'étude et la mise au point de leurs prototypes, mais aucun des algorithmes de programmation n'est certifié. Pour la production commerciale, préférez un programmeur professionnel certifié.

Luc Lemmens (labo d'Elektor)



www.elektor.fr/universal-programmer

Vos favoris :

1. Microscope USB Andonstar
www.elektor.fr/andonstar-usb-microscope



2. Raspberry Pi 3
www.elektor.fr/raspberry-pi-3
3. DVD Elektor 2016
www.elektor.fr/dvd-2016
4. Horloge de sable Arduino
www.elektor.fr/horloge-de-sable-arduino
5. Carte de commande LEGO pour Raspberry Pi
www.elektor.fr/lego-rpi-board
6. 36 Expériences de Physique avec Arduino
www.elektor.fr/arduino-36-experiences

Livre BBC micro :bit



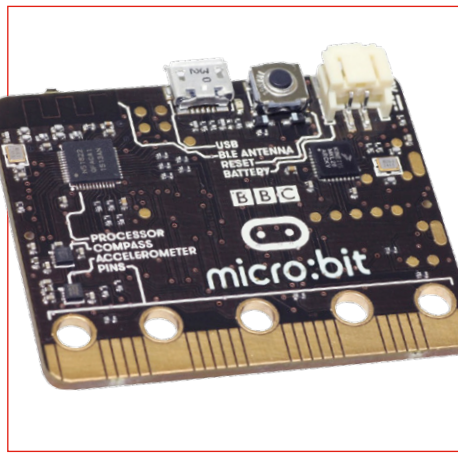
Ce livre (en anglais) permet de prendre en main la carte micro:bit conçue par la BBC et plusieurs acteurs de l'industrie électronique, et destinée à l'enseignement de la programmation. La carte animée par un ARM Cortex-M0+ (Nordic) est bien équipée : émetteur-récepteur BLE, matrice de 25 LED, accéléromètre, magnétomètre, deux boutons-poussoirs, connecteur d'extension pour accéder au bus I²C et à plusieurs broches GPIO, plages de connexion pour fiches banane.



Prix (membres) : 22,45 €

www.elektor.fr/dmicrobit-livre

BBC micro:bit

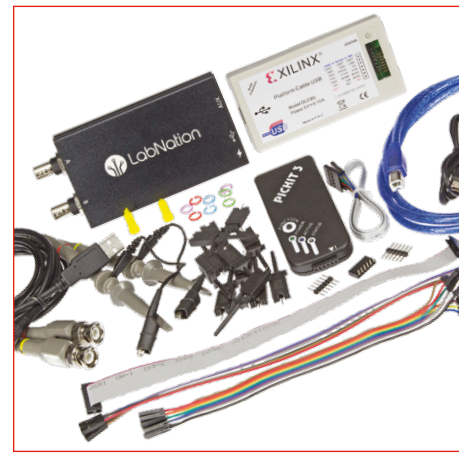


La carte BBC micro:bit est compacte et programmable dans différents langages, idéale pour faire ses débuts dans l'électronique programmée sans être intimidé. Elle combine capteurs (magnétomètre, accéléromètre), boutons, LED et Bluetooth. Chaque élément est programmable sans restriction grâce à un logiciel facile à utiliser, disponible sur un site consultable à partir d'un PC/téléphone. Concevez des jeux, connectez-vous à l'Internet des Objets...

Prix (membres) : 15,26 €

www.elektor.fr/bbc-micro-bit

SmartScope Maker Kit



Le SmartScope n'est pas qu'un excellent oscilloscope USB utilisable avec un ordiphone, une tablette ou un ordinateur, c'est aussi une formidable carte de développement FPGA. Ajoutez quelques connecteurs à la carte et vous voilà prêt à expérimenter, aidé par un logiciel facile à utiliser. Elektor propose un *Maker Kit* avec un SmartScope préconfiguré (version exclusive), deux programmeurs (JTAG et PICKIT3), les câbles nécessaires ainsi que deux sondes analogiques.



Prix (membres) : 269,10 €

www.elektor.fr/smartscope-maker-kit



maîtrisez les microcontrôleurs à l'aide d'Arduino 3^e édition augmentée

L'ambition de cet ouvrage est de vous faire entrer dans le monde Arduino puis de vous en faire ressortir victorieux pour vous emmener plus loin dans l'apprentissage de la programmation des microcontrôleurs. Il met la théorie en pratique sur une carte Arduino* avec l'environnement de programmation Arduino. Après ce parcours initiatique inédit, plaisant et ludique, vous programmerez vous-même n'importe quel microcontrôleur. Ce livre sera donc votre premier livre sur les microcontrôleurs avec une fin heureuse !

Tous les programmes présentés peuvent être téléchargés.

Points forts

- objectif double :
 - **Théorie** : apprentissage général de la programmation des microcontrôleurs
 - **Pratique** : montages sur carte Arduino avec l'environnement de développement Arduino
- réalisations très originales
- matériel peu coûteux ; logiciel gratuit, code source ouvert
- carte* d'expérimentation polyvalente créée par l'auteur
- **NOUVEAU** : carte AVR Playground*, mélange d'Arduino Uno et de carte de développement traditionnelle
- **NOUVEAU** : carte Elektor Uno R4* avec ATmega328PB et plus de périphériques que la carte Arduino Uno R3

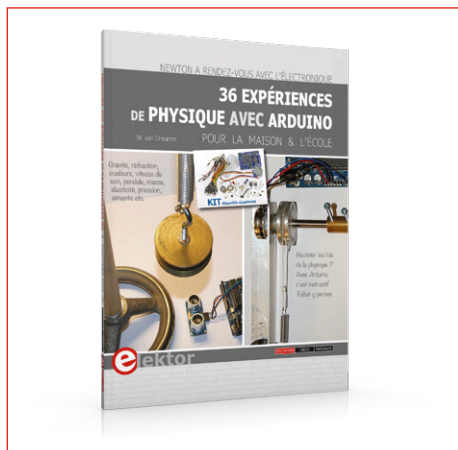
* à commander séparément du livre



Prix (membres) : € 36,50

www.elektor.fr/maitrisez-les-microcontrolleurs-3

36 Expériences de Physique avec Arduino



Ce livre n'est pas un manuel de physique : pas d'équations différentielles ni de courbes abstraites, mais des phénomènes physiques de la vie quotidienne. C'est une approche nouvelle et créative des leçons de physique grâce aux techniques modernes de mesure et de traitement des données. L'électronique utilisée (Arduino) est simple. Ajoutez-y le logiciel gratuit CoolTerm pour enregistrer les mesures et les retravailler ensuite sous Excel.



Prix (membres) : 22,41 €

www.elektor.fr/arduino-36-experiences

DVD Elektor 2016

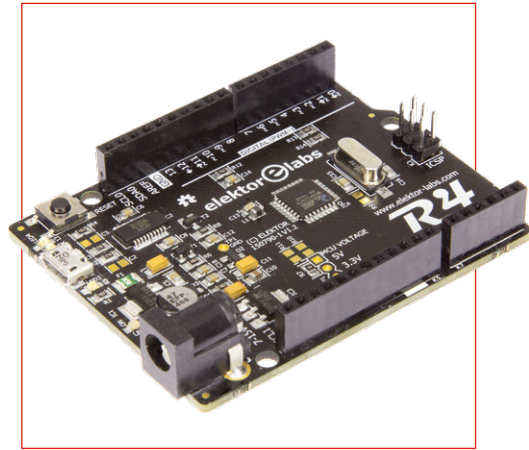


Ce DVD-ROM réunit tous les articles d'ELEKTOR parus au cours de l'année 2016, en français (mais aussi en anglais, allemand et néerlandais). Il contient le texte des articles ainsi que les schémas, et tous les dessins des circuits imprimés, sous forme de fichiers à haute résolution. Une fonction de recherche dans la table des matières vous permet de trouver immédiatement l'article souhaité.

Prix (non-membres) : 27,50 €

www.elektor.fr/dvd-2016

Elektor Uno R4



Quand un fabricant propose la version B d'un produit donné, les différences entre nouveau et ancien produits sont généralement peu significatives. Ce n'est pas le cas pour l'ATmega328P d'Atmel au cœur de l'Arduino Uno R3. La version B de ce µC a tellement de nouvelles fonctions que l'abandon de la version R3 est justifié : passez à l'Elektor Uno R4.



Prix (membres) : 24,95 €

www.elektor.fr/elektor-uno-r4

Hexadoku casse-tête pour elektorniciens

Votre magazine se termine toujours et encore par une grille de sudoku particulière. Éteignez le fer à souder, oubliez les écrans de PC et d'oscilloscope, rangez les pinces et les fils. Faites travailler votre matière grise d'une autre façon : attrapez un crayon pour remplir la grille d'hexadoku.

N'oubliez pas d'envoyer votre réponse pour tenter de gagner un bon d'achat.

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

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

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



Participez et gagnez !

Nous tirons au sort **cinq** des réponses internationales correctes reçues dans les délais ; leurs auteurs recevront 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 **23 mai 2017** à l'adresse **hexadoku@elektor.fr**

Les gagnants

La solution de la grille du numéro de mars/avril est **314AB**

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

Michael **Düren** (Allemagne), Denis **Moucharte** (Belgique), József **Nagy** (Hongrie), Eugene **Stemple** (États-Unis), Sake **van der Schaaf** (Pays-Bas)

Bravo à tous les participants et félicitations aux gagnants !

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

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

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.



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.

mai 2017

◇ Journée Régionale de l'Électronique

04/05 – Nantes
www.acsiel.fr/JRE

◇ MICM - matériaux innovants et chimie des matériaux

10 au 11/05 – Bordeaux
www.micm.events

◇ NAVEXPO - industrie maritime

10 au 12/05 – Lorient
www.navexpo.com

◇ Salon Iseramat – radioamateurs

13/05 – Tullins-Fures
<http://iseramat.org/salon/iseramat-2017>

◇ Innorobo - salon et congrès international de la robotique

16 au 18/05 – Paris
<http://innorobo.com/fr/accueil>



◇ Amiens Digital Forum – robotique, drones, objets connectés...

16/05 – Amiens
www.amiensdigital.com

◇ Hydrogaïa - conférences internationales et salon de l'eau

17 au 18/05 – Montpellier
www.hydrogaia-expo.com

◇ E-crime & cybersecurity France

18/05 – Paris
www.e-crimecongress.org/event/france

◇ Graphitec - chaîne graphique (de l'imprimé au multicanal)

30/05 au 01/06 – Paris
www.graphitec.com

◇ Semaine européenne du développement durable

30/05 au 05/06 – France et Europe
www.esdw.eu

◇ Journées COFREND 2017

30/05 au 01/06 – Strasbourg
www.cofrend.com

◇ Salon Mesure Solutions Expo2017

31/05 au 01/06 – Lyon
mesures-solutions-expo.fr



juin 2017

◇ Forum 5i

01/06 – Grenoble
www.forum5i.fr

◇ Le temps des hélices - meeting aérien

03 au 04/06 – La Ferté-Alais
www.ajbs.fr/presentation-du-meeting-2016



◇ MIDEM - forum international du disque, de l'édition musicale et vidéo

06 au 09/06 – Cannes
www.midem.com

◇ Smart Manufacturing Summit

07 au 08/06 – Paris
www.smart-manufacturing-summit.com

◇ Salon Only Light

13 au 15/06 – Lyon
onlylight-event.com

◇ Viva Technology

15 au 17/06 – Paris
vivatechnology.com

◇ ITS European Congress - services et systèmes de transport intelligent

19 au 22/06 – Strasbourg
strasbourg2017.itsineurope.com



52^e SALON INTERNATIONAL
DE L'AÉRONAUTIQUE ET DE L'ESPACE
PARIS LE BOURGET
19 - 25 JUIN 2017

◇ Salon de l'aéronautique et de l'espace du Bourget

19 au 25/06 – Paris
www.siae.fr

◇ Go numérique !

20/06 – Lyon
go-numerique.fr

◇ Parkopolis - stationnement et mobilité urbaine

21 au 22/06 – Paris
www.salonparkopolis.com

◇ Leti Innovation Days

28 au 29/06 – Grenoble
www.letidays.com/2017

◇ Nanotech

28 au 30/06 – Paris
www.nanotechfrance.com/fr

DÉCOUVREZ LA TECHNIQUE DE MESURE FLUKE À UN SUPER PRIX !

Multimètre numérique TRMS précis, convivial, sûr et fiable

- Mesure à valeur efficace vraie de la tension et du courant, 1000 V/10 A
- Précision de base 0,1 %
- LCD, résolution 6000 counts
- Sélection manuelle et automatique de la plage
- Mesure de la fréquence, de la capacité et de la résistance

FLUKE®



**ENSEMBLE COMPRENANT
POCHETTE & CIRCUITS DE MESURE**



au lieu de 283,99

FLUKE 175 PROMO

**ÉCONOMISEZ
44 %**

158,82

Pour mesurer précisément le courant de charge, la tension ou la résistance :
Pince ampèremétrique à valeur efficace vraie

- Mesure du courant alternatif jusqu'à 400 A
- Mesure de la tension continue ou alternative jusqu'à 600 V
- Mesure TRMS permettant des résultats de mesure précis avec des signaux non linéaires
- Mesure de résistance jusqu'à 40 kohms
- Mesure de la température, de la capacité et de la fréquence



**ENSEMBLE AVEC
HOLSTER & COFFRET**

au lieu de 311,01 FLUKE 325 PROMO

**ÉCONOMISEZ
38 %**

192,44

FLUKE®

Testeur de tension de la toute dernière technologie de mesure et de sécurité !

T150 VDE avec affichage LED & LCD

- Test de passage & mesure de la résistance
- Indicateur de champ de rotation pour 2 systèmes de phases
- Fonction pour contrôler des RCD
- Lampe de poche intégrée



FLUKE T150 VDE

**ASTUCE
PRIX**

125,88

FLUKE®

**ABONNEZ-VOUS DÈS MAINTENANT
AU NEWSLETTER ET GAGNEZ !**

Nous offrons tous les mois par tirage au sort un super cadeau technique aux abonnés de notre Newsletter !

PARTICIPER MAINTENANT ! ► <http://rch.it/Ay>



Prix du jour! Prix à la date du: 21. 3. 2017

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

Les langues de notre boutique:

MODES DE PAIEMENT INTERNATIONAUX:



ACHETER EN LIGNE TOUT SIMPLEMENT !



www.reichelt.fr

ASSISTANCE TÉLÉPHONIQUE: +33 97 518 03 04