

convertisseur USB-DMX compact



entièrement isolé

carte tampon pour le Raspberry Pi

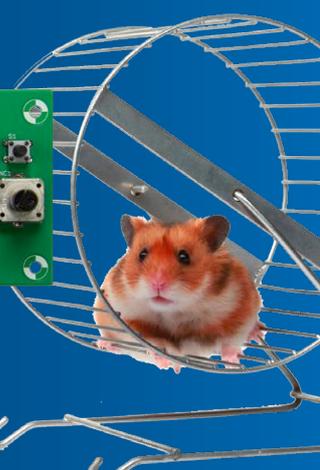


ne grillez plus vos entrées/sorties



run-O-mètre pour hamster

mesure et affichage de la vitesse
et de la distance parcourue



shield SDR comme instrument de mesure ✂️ ESP8266 dans le monde du modélisme ✂️ filtres *crossover* audio actifs simples et bon marché ✂️ synthèse numérique directe (DDS) sur Arduino ✂️ conception à ultra faible puissance ✂️ horloge à LED économe en énergie ✂️ la pratique des amplificateurs opérationnels ✂️ lampe magique : polychromie sans fil ✂️ lampemètre AVO ✂️ écrans couleur OLED

Des solutions pour les besoins analogiques simples ou complexes

Des composants performants pour gérer toutes les contraintes de conception

www.microchip.com/AnalogProducts

microchip
DIRECT
www.microchipdirect.com



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

Téléphone : (+33) 01 70 61 27 41
du lundi au vendredi de 10h à 13h

Télécopie : (+33) 01 70 61 27 42

www.elektor.fr | www.elektormagazine.fr

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

Publicité :

Margriet Debeij
Tél. : +49 (0)241 955 09 174
margriet.debeij@elektor.com

DROITS D'AUTEUR :

© 2018 Elektor International Media B.V.

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

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

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



La barrière de la langue, passez au-dessus !

Nombre d'articles d'Elektor sont des traductions. En effet, pour ses textes, la rédaction internationale pioche dans un pot commun de projets à publier en anglais, allemand, néerlandais ou français.

Pour les traduire, nous avons renoncé aux « traducteurs » en ligne comme Babelfish ou Google Translate. Le résultat est parfois rigolo, souvent décevant et les guillemets justifiés. Voici qu'un nouveau venu, DeepL, conçu par des spécialistes de l'intelligence artificielle, offre une qualité de traduction sans précédent.

Nous avons procédé à des essais comparatifs avec nos propres articles. Les traductions obtenues avec DeepL sont instantanées, imparfaites, mais plutôt compréhensibles. La plupart des écueils sont liés à la méconnaissance de la terminologie électronique : *time* traduit par « temps », quand il s'agit d'une durée ; *saved* traduit par « économisé », au lieu de sauvegardé etc. Le « piano framboisé » (pour Raspberry Pi, vous l'aurez deviné) nous a fait bien rire. Ce n'est pas la première application de l'intelligence artificielle qui conduise à des situations comiques (cf. mon édito du dernier numéro sur les commandes vocales), mais ce service-là redistribue les cartes et chamboule tout.

DeepL ne remplace pas (encore) un traducteur compétent, mais il offre ses services gratuitement. Dommage que parfois il oublie carrément certains mots, les adverbes notamment, ce qui fausse gravement le sens...

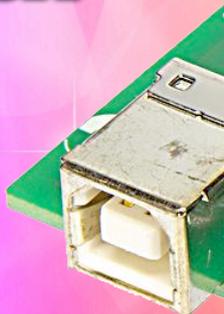
Pour l'instant, nos traducteurs-électroniciens ont et auront encore du pain sur la planche. J'en profite pour saluer leur travail de grande qualité, leur implication (ils ont toujours le schéma en main), leur disponibilité et leur patience (quand les corrections se suivent et se contredisent).

Mariline Thiebaut-Brodier

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 :	Pascal Godart, Yves Georges, Robert Grignard, Denis Lafourcade, Jean-Louis Mehren, Denis Meyer, Hervé Moreau, 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

convertisseur USB-DMX compact entièrement isolé

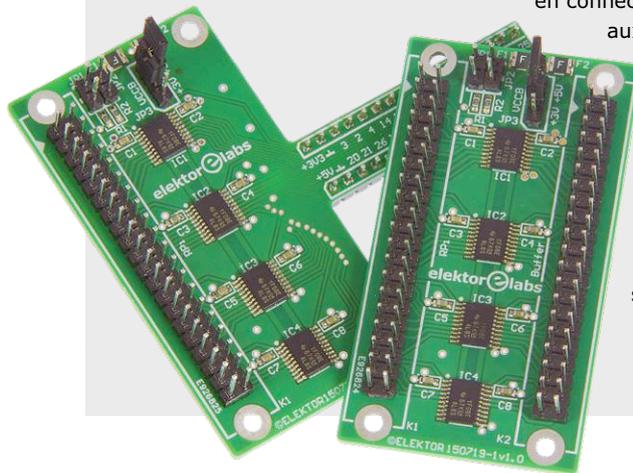


- 5 bientôt dans Elektor**
- 10 Q & R**
(presque) tout ce que vous avez toujours voulu savoir sur... quand et comment butiner de l'énergie
- 48 vol tous azimuts**
CC contre CA : qui va gagner ? * Outil de labo indispensable * Projet DJ Elektor * Histoire d'éléphant
- 64 agenda**
novembre 2018
- 70 référence de tension à « zener enfouie » LTZ1000**
drôle de composant n°39
- 76 mur des célébrités**
- 83 agenda**
décembre 2018
- 105 Rétronique**
lampemètre AVO toujours en forme
- 110 questions d'éthique**
l'intelligence artificielle à l'état sauvage
- 112 l'e-choppe d'Elektor**
- 114 Hexadoku**
casse-tête pour elektorniciens

12

carte tampon pour le Raspberry Pi ne grillez plus vos entrées/sorties

Lorsque l'on expérimente avec le Raspberry Pi en connectant divers matériels aux entrées/sorties via le connecteur GPIO, il peut arriver qu'une tension trop élevée ou une décharge électrostatique détruise le port. La carte est alors bonne à mettre au rebut. Avec la carte tampon décrite ici, plus de souci à se faire !



- 6 conception à ultra faible puissance**
trucs, astuces et achoppements
- 25 Elektor Board Games**
concours parrainé par Avnet Silica et EBV Elektronik
- 29 la pratique des amplificateurs opérationnels**
partie 1 : fondements
- 38 cartes de développement Arrow**
ouvertes et... offertes aux bonnes idées
- 52 conception de matériel avec (V)HDL (2)**
affichage de l'heure en binaire
- 62 filtres crossover audio actifs simples et bon marché**
à 3 voies, des 3^e et 4^e ordres
- 65 écrans couleur OLED**
pilotes par STM32
- 88 mise à jour pour le réveil à triple affichage**
version 2.6.1 du logiciel
- 98 hors-circuits avec Robert Lacoste**
techniques de filtrage de canal pour éviter le gaspillage de bande passante

29

la pratique des amplificateurs opérationnels partie 1 : fondements





26

run-O-mètre pour hamster

mesure et affichage de la vitesse
 et de la distance parcourue



Si vous avez un hamster, sa cage contient probablement une roue d'exercice dans laquelle il peut courir à tire-larigot. Les électriciens connaissent bien ce genre de cage d'écureuil, mais comme rotor de moteur à induction. C'est aussi un terrain d'expérimentation pour les électroniciens. Ce montage simple vous donnera une idée de la vitesse atteinte et de la distance parcourue par votre hamster.



93

- 32 lampe magique**
polychromie sans fil
- 42 ESP8266 dans le monde du modélisme**
télécommande par ordiphone simple et bon marché
- 58 horloge à LED économe en énergie**
affichage de l'heure à la demande
- 71 détecteur de court-circuit**
avec source de courant et sonde à effet Hall
- 78 synthèse numérique directe sur Arduino**
expériences intéressantes à moindres frais
- 85 variateurs pour LED (2)**
passer au numérique, c'est (parfois) ringard !
- 93 run-O-mètre pour hamster**
mesure et affichage de la vitesse et de la distance parcourue

 bientôt sur ces pages

Extrait du sommaire du prochain numéro :

- Alarme « ronflante » pour Arduino
- Android et C sur Raspberry Pi
- Communication sûre avec ATmega328
- Découpe de polystyrène
- Limitation de courant USB/Appareil de mesure
- Poste de soudage de CMS fait maison
- Principes de la commande de moteurs
- Récepteur FM-RDS pour Raspberry Pi
- Station-météo en kit

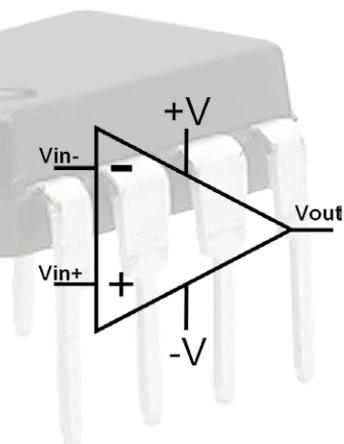
Etc.

Sous réserve de modification.

Le numéro de janvier-février 2019 paraîtra le 27 décembre 2018.

 sur la scène : les projets

- 12 carte tampon pour Raspberry Pi**
ne grillez plus vos entrées/sorties
- 18 shield SDR 2.0 d'Elektor (3)**
techniques de mesure
- 26 convertisseur USB-DMX compact**
entièrement isolé



ce n'est pas vraiment aussi simple. La plupart des applications comprennent tant des calculs que des transferts de données. Ceux-ci réclament généralement des délais fixes, par exemple pour une transmission sérielle à un débit binaire imposé. Dans ces cas-là, on peut économiser pas mal d'énergie en changeant l'horloge du système. Pendant les phases de communication, on remonte suffisamment la fréquence d'horloge ; lors des calculs, on fait travailler l'unité centrale à la plus haute fréquence possible, puis on la fait passer en mode de sommeil. Par rapport à une longue phase active à fréquence réduite (à la même tension d'alimentation), on économise ainsi de l'énergie.

Quel genre d'horloge faut-il choisir ? Si on utilise les modes de sommeil, l'oscillateur RC interne est tout indiqué, à condition qu'il soit assez précis. Dans le doute, mieux vaut comparer différents types, voire de la même famille de produits. Les oscillateurs RC modernes varient tout au plus de 2 % sur tous les domaines de tension et de température. Si l'instabilité vous donne des soucis, demandez au fabricant les spécifications nécessaires. Il est déjà arrivé que de nouvelles versions du même composant présentent plus de gigue que leurs prédécesseurs, sans que ce soit signalé explicitement dans la fiche technique. Ceci montre combien il est judicieux de laisser disponibles des composants en versions plus anciennes ! Si l'oscillateur RC n'a pas la stabilité voulue, un résonateur en céramique est indiqué en raison de sa rapidité de démarrage, plutôt qu'un quartz.

Périphériques

Il est très utile que les périphériques puissent communiquer entre eux. Cela ne fait pas qu'économiser l'énergie, mais facilite aussi la conception, relève les performances, raccourcit le code de programme et permet une chronométrie exacte et reproductible avec un fonctionnement synchrone entre plusieurs périphériques. Sans oublier que cela permet de satisfaire plus facilement aux critères de sécurité. Les mots-clés sont par exemple DMA, *event system* (système d'événement discret), *core independant peripherals* (CIP). L'unité centrale peut ainsi exécuter d'autres tâches sans interruption ou simplement entrer en mode de veille. Sur les μC modernes à faible puissance, il est indispensable d'avoir la possibilité d'éteindre les périphériques inutilisés. Ici aussi, il y a intérêt à comparer les fiches techniques (errata compris) de chaque composant et aussi à aller voir dans les notes d'application du fabricant.

On éteint en bloquant l'horloge ou en débranchant l'alimentation. La logique CMOS privée d'horloge, mais toujours reliée à Vcc, ne consomme plus que le minuscule courant de fuite. Malgré tout, il est préférable de l'isoler totalement de la tension d'alimentation.

Si le contrôleur est équipé de CCL (*customer configurable logic* ou logique configurable par l'utilisateur), il peut contribuer à l'économie d'énergie du fait que les fonctions incorporées dans le matériel s'exécutent plus vite et à moindre consommation.

Les broches d'entrée

Les entrées ouvertes, flottantes, sont bien sûr à proscrire, mais les mettre tout bêtement au niveau logique 0 ou 1 ne suffit pas à atteindre la consommation la plus basse. Il faut veiller à ce que les tensions d'entrée ne diffèrent pas de plus 0,5 V de la masse ou de Vcc.

Horloge en temps réel

De nombreux μC sont dotés d'une horloge en temps réel (HTR

ou RTC) à laquelle il faut juste brancher un quartz de montre. C'est bien sûr moins cher qu'une HTR externe, mais présente des inconvénients. La consommation est nettement plus élevée et la précision est moindre par rapport à celle d'un module, spécialement s'il est compensé en température. Les HTR les moins gourmandes à quartz intégré descendent à 40 nA, les compensées à environ 100 nA. Certaines possèdent un oscillateur RC supplémentaire à plus faible précision qui ne prend que 17 nA environ. Les modules à quartz intégré sont toujours plus précis que les anciennes versions à quartz externe, leur imprécision est bien spécifiée à 5 ppm, mais elle risque d'augmenter sérieusement lors du soudage. Avec les modules intégrés et un profil de température de soudage maintenu avec précision, ce risque ne se présente pas. La conception du circuit et de son tracé exige plus de soin avec un quartz externe, sinon d'autres déviations de fréquence, voire des difficultés de démarrage, menacent, en particulier avec les oscillateurs à plus haute impédance.

Les producteurs de quartz professionnels connaissent les caractéristiques des différents μC et fournissent de l'aide à la conception. N'essayez pas de mesurer quoi que ce soit sur un quartz de montre avec votre sonde standard !

Que penser des valeurs typiques ?

Ne tenez pas compte des « typical values » publiées dans les fiches techniques ! C'est plus de la publicité qu'autre chose, la plupart sont inutiles pour la production en série, partez toujours du scénario du pire, donc de la consommation maximale, souvent beaucoup plus élevée que la valeur typique. La formule « xx μA per MHz » ne correspond pas à la consommation du contrôleur à juste 1 MHz ! En règle générale, elle est nettement plus haute. L'information ne ressortira jamais que de l'étude minutieuse de la fiche technique, notes de bas de page comprises.

Mémoire

La plupart des μC ont des mémoires flash de programme que l'on peut actualiser à l'aide d'un chargeur d'amorçage. Si vous voulez de temps à autre enregistrer de petits ensembles de données, assurez-vous que le contrôleur sélectionné a une granularité des mémoires suffisante (format de page). Sinon, vous devrez écraser de nombreuses cellules, ce qui coûte de l'énergie, du temps d'écriture (donc encore de l'énergie) ainsi que de la durée de vie. On peut réécrire dans les EEPROM (NOR) flash de 10 à 100 fois. Des EEPROM externes peuvent apporter une solution, surtout si le μC n'en dispose pas. La technologie moderne a produit des CBRAM, *Conductive Bridging RAM*, qui s'écrivent avec 25 à 50 fois moins d'énergie que les EEPROM et en des temps beaucoup plus courts, résistent bien au rayonnement, ont des tensions d'alimentation moindres et réduisent les coûts. Le nombre de cycles d'écriture se situe aux alentours de cent mille et, contrairement aux flash et aux EEPROM, les CBRAM sont insensibles à la température. Il faut voir aussi la rétention des données : même à 150 °C de température en service continu, elles sont préservées pendant dix ans. Les capacités de mémoire vont actuellement jusqu'à 512 Ko, les types plus grands sont en cours de développement.

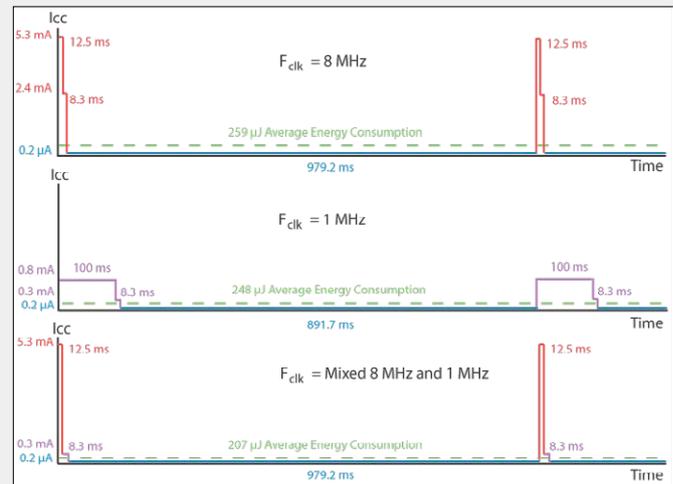
Condensateurs

Les condensateurs de découplage sont indispensables dans les circuits électroniques à μC , FPGA et consorts. Ils ne servent

Commutation d'horloge dynamique

Dans cette application, c'est un ATmega128 classique qui a été utilisé, vu que l'AVR Pico Power avec 128 Ko de flash n'était pas encore disponible. L'émetteur de la clé travaille pendant 98 % du temps en mode *power down* et, pour le test, est activé une fois par seconde par le bouton-poussoir. La tension d'alimentation est de 3 V. La communication cryptée AES s'opère par radio avec la station de base à un débit binaire de 9 600 bauds. Le traitement des données, chiffrement compris, prend environ 100 000 cycles d'horloge, 8 octets sont envoyés par transmission.

On démarre à 8 MHz, la fréquence d'horloge maximale de l'oscillateur RC. La quantité d'énergie totale par transmission est de 250 μ J. Puis on abaisse l'horloge à 1 MHz, l'économie d'énergie est alors significative pendant la durée de la communication au débit binaire fixé. Les opérations mathématiques aussi se déroulent alors à moindre consommation de courant, mais durent plus longtemps. Cela se traduit maintenant par une économie de 248 μ J, en tout 4 % par rapport au fonctionnement à 8 MHz. Mais si les calculs s'effectuent à 8 MHz et que l'on commute l'horloge à 1 MHz pour les communications, la consommation n'est plus que de 207 μ J, on économise donc 20 % d'énergie !



pas seulement au découplage entre différentes parties d'un circuit, ils sont aussi une réserve d'énergie. Les condensateurs standard suffisent aux applications à basse puissance, mais généralement pas aux appareils à ultra faible puissance avec des piles à très grande longévité. Les fuites considérables de courant des condensateurs électrolytiques et au tantale sont connues de tous, mais saviez-vous que les condensateurs multicouches à la céramique (MLCC) pouvaient avoir des fuites de l'ordre de 5 μ A ?

Tension d'alimentation

Dans les applications à basse puissance, on travaille sous des tensions plus faibles que celle du standard industriel de 5 V. Généralement la tension provient de piles ou d'accumulateurs. La question de savoir quelle est la bonne source d'énergie mérite un article à elle seule, mais voici déjà quelques conseils. Le taux de décharge spontanée des petites piles alcalines se situe autour de 5 à 10 μ A à température ambiante. Dans ce cas, pas besoin de vous battre pour quelque 100 nA dans le projet. Les fabricants de piles sont rarement enclins à fournir une documentation convenable, mais contactez-en un de bonne réputation et montrez-vous tenace.

La tension nominale de 1,5 V de la pile alcaline est trop basse pour la plupart des μ C, d'autant qu'elle faiblit à l'usage. Si l'espace disponible le permet, le plus simple est d'en mettre deux en série. Le convertisseur survolteur CC/CC est une solution de rechange. Il est moins encombrant et fournit une tension constante de 3 V, par exemple, bien stabilisée jusqu'à environ 0,7 V de tension de source, voire moins. Les pertes dans les convertisseurs modernes sont plus que compensées par une conception soignée des circuits et une meilleure utilisation des piles. D'autre part, leur utilisation n'est pas critique s'il y a un interrupteur allumé/éteint mécanique. En revanche, si c'est le changement de potentiel d'une ligne d'E/S, un temporisateur, une variation de température par thermostat mécanique, du bruit ou un signal radio qui réveille le système, alors le contrôleur ou une autre partie du circuit doit rester sous tension en

permanence. Dans ce cas, le convertisseur CC/CC doit présenter une consommation propre extrêmement faible, même à charge très basse.

Il est préférable que la partie sensible du circuit se trouve à une tension nettement plus basse que le contrôleur, tout en conservant une faible consommation propre – une combinaison rare ! On peut alors éteindre le convertisseur pendant le mode de sommeil.

Encore mieux, utiliser l'un des rares μ C à ultra faible puissance qui se contente de 1,2 V ou moins, disponibles depuis peu avec cœur Cortex M0+. Des types à moins d'un volt sont en développement.

Si l'appareil à concevoir doit de toute façon être alimenté sous une plus haute tension que celle que le contrôleur peut supporter, il reste à faire preuve de créativité. Par exemple, mettre quatre piles en série pour alimenter l'électronique ainsi qu'un moteur, des électroaimants et relais. La tension totale de 6 V de la série descend au fur et à mesure à 3,6 V. Alors pourquoi ne relierait-on pas l'électronique à la moitié de la série ? La tension y descend de 3 V à 1,8 V, mais reste dans le domaine acceptable pour de nombreux types standard. On peut alors se passer du convertisseur CC/CC ou du régulateur à faible chute de tension.

La résistance interne d'une pile bouton au lithium se situe à quelques dizaines d'ohms et monte avec la décharge progressive à plus de 100 Ω . C'est trop pour une charge par impulsions. Avec les accumulateurs, il faut éviter la décharge profonde et avec les cellules LiPo, c'est interdit.

Les liaisons radio

La mise en réseau par radio est beaucoup plus ancienne que l'Ido, pensons simplement aux horloges pilotées par DCF77 ou aux appareils auditifs. L'électronique portable aussi existe depuis des décennies. Pourtant, les équipements sportifs et médicaux, les enregistreurs, la domotique et l'éclairage sont des marchés en pleine expansion dans ce domaine.

La radio à longue portée

Le besoin en énergie d'une liaison radio dépend de la puissance d'émission, du débit des données, de la gamme de fréquences, de la sensibilité du récepteur et de l'antenne. On peut atteindre des portées de 5 à 30 km avec des émetteurs à basse puissance dans la gamme UHF sous le gigahertz pour de faibles débits et de courtes périodes d'émission. Dans ces conditions, une pile peut servir des années durant. On fait souvent appel à des standards populaires comme LoRa ou Sigfox pour les applications telles que les systèmes de téléalarme en cas de cambriolage, de vol ou de vandalisme dans des installations distantes, pour le contrôle de processus, la météorologie, l'acquisition de données, la localisation d'animaux, entre autres. Avec WSPR (*weak signal propagation reporter*) en ondes courtes, on atteint même des portées de 20 000 km, autant dire l'autre bout du monde, avec 100 mW de puissance d'émission, 1 000 km avec 10 mW ! Le débit de données est très faible, 1 bit/s, mais il suffit par exemple à transmettre quotidiennement le bulletin météo depuis le pôle Sud.

La radio à courte portée

Les systèmes d'accès sans clé, transpondeurs et certains ordinateurs de vélo sont des applications typiques de radio de proximité en grandes ondes. Il existe pour cette gamme GO des récepteurs extrêmement économes avec une consommation de 2 μ A en réception permanente, donc pas seulement pour de brèves périodes de fonctionnement. En revanche, Wi-Fi, BTLE, ZigBee sont des standards bien connus dans la gamme UHF en GHz à la limite supérieure du spectre radio.

Pour l'accès sans clé, on utilise les deux gammes de fréquences : grandes ondes pour réveiller une puce spéciale dans la voiture quand la clé s'en approche. Elle ne réagit qu'à une forme de modulation déterminée du signal en grandes ondes de la voiture. La puce éveille alors le trançeur UHF pour s'occuper de la communication proprement dite, bidirectionnelle et sécurisée avec le véhicule.

Il y a aussi une autre combinaison de ce type de composant. Un récepteur à diode démodule le signal radio à haute fréquence à double modulation. Si le signal correspond au domaine de réception du récepteur de réveil, celui-ci active le reste du circuit pour que l'échange de données puisse avoir lieu. Même si le récepteur est en écoute permanente, son courant de veille n'est que de 2 μ A environ. C'est à peu près un pour mille de la

consommation habituelle d'un module BTLE (Bluetooth à basse consommation) ! Avec des protocoles bien ficelés, la procédure pourrait s'utiliser aussi pour BTLE, Wi-Fi et d'autres standards.

NFC (*Near Field Communication*)

La communication en champ proche (CCP) est aussi intéressante pour les produits à puissance ultra faible. Les balises NFC passives existent depuis longtemps. On peut les utiliser comme d'autres transpondeurs, mais le lecteur NFC de l'ordiphone omniprésent peut aussi l'exploiter. Certaines de ces balises contiennent une technique de chiffrement qui les rend infalsifiables, pour empêcher les copies illégales (anticlonaage) et toute manipulation illicite. Les composants d'interface CCP permettent à n'importe quel appareil de communiquer en champ proche, par exemple comme interface de service. Ils sont bon marché, on peut les implémenter facilement sans autorisation ni connaissance particulière de la HF, puisqu'ils sont passifs.

La puce et éventuellement le contrôleur attendant sont alimentés par un module intégré de glanage d'énergie, celle prélevée dans le champ du lecteur CCP suffit à reprogrammer un μ C par chargeur d'amorçage. Beaucoup de ces composants contiennent d'ailleurs leur propre EEPROM. Il existe aussi des μ C avec interface CCP intégrée.

Le glanage d'énergie s'inscrit parfaitement dans nombre d'applications Ido pour lesquelles l'économie d'énergie est une préoccupation récurrente. Un capteur utilisé occasionnellement peut rester sans pile pour autant qu'il n'ait pas de fonction d'enregistrement. Ainsi, on peut encore lire avec un ordiphone ou un lecteur après des décennies un capteur d'humidité enfoui dans un mur ou un plancher. Les petits écrans de papier électronique peuvent aussi être alimentés par le champ CCP lors de l'écriture, ils n'ont pas besoin de pile non plus.

Les balises à capteur de puissance ultra faible enregistrent les valeurs de température, d'accélération, de rayonnement et autres paramètres, souvent pendant des années dans des mémoires non volatiles et fournissent les données avec les horodatages de la HTR. Des balises de ce genre contiennent déjà la plupart des composants nécessaires sur une puce, à côté de l'interface CCP, une HTR, une EEPROM, un convertisseur A/N, un capteur de température et une interface analogique pour des capteurs externes. ◀

(180440-04 - version française : Robert Grignard)

Bibliographie

- [1] « Ultra-low Input Voltage DC-DC Converter for Micro Energy Harvesting », D. Grgić, T. Ungan, M. Kostić and L. M. Reindl, University of Freiburg, Technical Faculty, Department of Microsystems Engineering, Germany
- [2] « A Guide to Choosing the Right Ultra - Low IQ Low Dropout Linear Voltage Regulators », Onsemi AppNote AND9089/D
- [3] « PicoPower Basics », Atmel Application Note AVR4013
- [4] « Ultra Low Power Bioelectronics », R. Sarpeshkar, Cambridge, 3rd edition 2013, ISBN 978-0-521-85727-7
- [5] « Low Level Measurements Handbook », Keithley, 6th edition : www.keithley.com
- [6] « New Series for Low Power Application », Epson, Training Paper
- [7] « Hardware and Firmware Issues in Using Ultra-Low Power MCUs », J. Ganssle, Rev. 4, March 2016
- [8] « Innovative Techniques for Extremely Low Power Consumption with 8-bit Microcontrollers », A.M. Holberg, A. Saetre, Atmel White Paper
- [9] « Conception de circuits à faible consommation avec les AVR », A. Riedenauer, Elektor 03/2010 : www.elektormagazine.fr/090157



(presque) tout ce que vous avez toujours voulu savoir sur... quand et comment butiner de l'énergie

Réponses de **Robert Lacoste**

Q *Le butinage énergétique (« energy harvesting » pour les Anglo-saxons) est très à la mode. Que recouvre exactement ce terme ?*

R Je ne suis pas certain qu'il existe une définition officielle, mais ce terme désigne en pratique l'ensemble des techniques permettant d'alimenter un petit système électronique en utilisant des sources d'énergie opportunistes. Cela veut dire des sources d'énergie qui sont « naturellement » présentes autour de l'objet, et peuvent être utilisées pour récupérer assez d'énergie pour son fonctionnement.

Q *De quelles sources d'énergie parle-t-on ?*

R Évidemment, la première est le soleil. Son « butinage » n'est pas nouveau : les calculatrices solaires existent depuis la fin des années 70... Cependant beaucoup d'autres sources peuvent être envisagées : la chaleur, les mouvements mécaniques, les vibrations, les flux de fluides, les réactions électrochimiques, les rayonnements électromagnétiques, etc. Nous sommes entourés d'énergie, la difficulté est de la récupérer...

Q *Imaginons que je veuille concevoir un produit électronique alimenté par butinage énergétique. Quelle doit être ma première étape ?*

R De manière un peu schématique, toutes ces sources opportunistes ne peuvent fournir qu'une très faible quantité d'énergie (à part un grand panneau solaire à l'extérieur...). C'est une lapalissade, mais il faut donc commencer par optimiser le plus possible le circuit électronique du produit pour réduire drastiquement sa consommation. Le moindre micro-ampère risque en effet de compter... N'hésitez pas à jeter un œil à l'article que j'ai publié il y a quelque



photo : shutterstock.com

temps sur ce thème [1].

La conception électronique du produit doit aussi prendre en considération une caractéristique commune à beaucoup de sources opportunistes : elles sont souvent aléatoires. Imaginez que vous réalisez un pisteur de vélo alimenté par la rotation d'une des roues. Accepteriez-vous qu'il s'arrête de travailler si le vélo est volé et transporté dans une camionnette ? Évidemment que non, il faut donc prévoir soit une réserve d'énergie (batterie ou supercondensateur), soit une seconde source d'alimentation (une pile ?). De même si votre projet est un thermomètre alimenté par les champs électromagnétiques ambiants, admettriez-vous qu'il n'effectue des mesures que lorsque le Wi-Fi de votre voisin est actif ? Ou, plus subtil, peut-être qu'il mettrait à jour sa température plus ou moins souvent en fonction de l'énergie disponible ? Ce type de fonctionnement « adaptatif » est très utile pour faciliter l'utilisation de sources fluctuantes.

Q *Et ensuite ?*

R La seconde étape est d'estimer de combien d'énergie on pourra disposer en butinant. En général, ce n'est

pas très simple, mais il est indispensable d'avoir rapidement un ordre de grandeur afin de vérifier si le projet est réaliste ou pas. Si votre produit consomme 1 mA et que la source peut fournir 1 μ A de temps en temps, mieux vaut passer à autre chose ! À ce stade des maquettes de faisabilité peuvent être utiles.

Q *Prenons le cas d'une cellule solaire. Comment savoir quelle quantité d'énergie sera disponible ?*

R Tout d'abord il y a deux cas très différents : les panneaux à l'extérieur et ceux à l'intérieur. Pour l'extérieur, évidemment l'ensoleillement dépend beaucoup de la région et de la saison. La bonne nouvelle est qu'il existe d'excellents sites web permettant de connaître l'ensoleillement moyen pour chaque mois de l'année et pour un lieu donné en Europe entre autres [2]. Avec ces données et un peu de sueur, vous saurez combien de Wh/m² et par jour sont disponibles pendant les mois les plus critiques. D'autres sites permettent même de déterminer la probabilité que l'ensoleillement soit inférieur à un seuil défini un certain jour, ce qui permet si besoin de calculer la capacité d'une batterie tampon [3].

Imaginons que votre circuit électronique a besoin de 3 V et 1 mA en moyenne, soit 72 mW.h par jour, que vous êtes à Paris, et que vous avez prévu une batterie tampon suffisante pour n'avoir à vous préoccuper que de l'énergie moyenne sur quelques jours. Via ces sites web, vous trouverez que l'ensoleillement à Paris varie de 6000 Wh/m²/j en juin à 793 Wh/m²/j en décembre, c'est bien sûr ce dernier chiffre qu'il faut utiliser pour le dimensionnement. Les panneaux solaires sont toujours spécifiés pour 1000 W/m², vous devez donc dénicher un panneau qui fournit au moins 72 mW.h × 1000 W/m² / 793 W.h/m², ce qui donne 90 mW. Vous prenez une marge de sécurité en multipliant ce résultat par trois ou quatre... Consultez des catalogues, ce sont des panneaux de l'ordre de 10 × 10 cm, c'est donc envisageable.

Il vous reste ensuite à concevoir le circuit électronique pour exploiter au mieux ce panneau, les composants de type MPPT (*maximum power point tracking*) sont vos amis (intéressez-vous au circuit intégré de gestion d'alimentation ADP5091 d'Analog Devices par exemple). La technique MPPT, littéralement « poursuite du point de puissance maximale », s'applique aux générateurs électriques non linéaires (photovoltaïque ou éolien par exemple). La tension aux bornes du générateur est adaptée en permanence pour fournir le maximum de puissance à la charge (batterie).

Q Et le solaire en intérieur ?

R Là c'est une autre paire de manches, car les niveaux d'ensoleillement n'ont rien à voir. Les cellules photovoltaïques optimisées pour l'intérieur sont différentes, et les puissances disponibles sont très faibles. Par exemple un panneau à hautes performances de 5×2 cm (Sanyo AM-1805) permet d'espérer à l'intérieur de l'ordre de 4 µA sous 2,6 V dans une pièce peu éclairée (50 lux).

Q Comment récupérer l'énergie d'un objet qui bouge ou qui vibre ?

R Il y a trois catégories primordiales de solutions. La première est d'avoir recours à un convertisseur électromagnétique, c'est un grand mot pour dire qu'on fait bouger un aimant dans une bobine. La dynamo de votre vélo est un bon exemple. On trouve des interrupteurs basés sur ce principe : ils utilisent

l'énergie apportée par un doigt pour produire assez d'énergie afin d'émettre un message radio. La seconde est d'utiliser l'effet piézoélectrique : pressez un bout de quartz et de l'énergie en sortira, c'est le principe de l'allume-gaz. Seule difficulté : l'effet piézo crée de très faibles courants, mais à haute tension, d'où le besoin de composants spécifiques (cf. l'alimentation de nanopuissance à butinage énergétique LTC3588-2). Enfin on peut se servir de l'effet électrostatique : faire varier périodiquement l'espacement entre deux électrodes d'un condensateur chargé fournit un peu d'énergie.

Q Quid de la chaleur ?

R On peut exploiter l'effet Seebeck ou thermoélectrique (inverse de l'effet Peltier). Une différence de température entre les deux faces d'un module Peltier produit une tension à ses bornes. Attention, c'est bien une différence de température : il faut une source chaude et une source froide. Le rendement et la tension de sortie dépendent directement de cette différence. Avec quelques degrés d'écart, on n'obtient guère plus que quelques dizaines de millivolts avec des modules classiques, qui nécessitent des circuits électroniques très spécifiques pour les utiliser. Fort heureusement les fabricants de circuits intégrés ont des solutions fantastiques à proposer : le LTC3109 par ex. est un convertisseur DC/DC qui démarre avec seulement 30 mV...

Q Elektor a publié il y a quelque temps une lampe de chevet alimentée par la chaleur d'une bougie [4], c'est un bel exemple de butinage énergétique, non ?

R C'est un superbe projet, mais honnêtement il ne répond pas à la définition du butinage énergétique puisque la bougie est mise là exprès : ce n'est pas une source opportuniste !

Par contre c'est un très bel exemple d'optimisation énergétique parce que l'énergie thermique de la flamme de la bougie, normalement perdue, est utilisée pour créer de la lumière et amplifier ainsi la lumière naturelle de la bougie.

Q Avec tout ça, verra-t-on bientôt la fin des piles et des batteries ?

R Soyons francs, le caractère aléatoire des sources de butinage impose souvent la présence d'une petite pile auxiliaire ou au moins d'un accumulateur d'énergie. Il y a bien sûr des cas d'usage où le butinage suffit, mais il faut pour cela avoir une source « maîtrisée ». C'est généralement le cas pour les applications industrielles, moins pour les produits domestiques.

Vous voulez une anecdote qui m'est arrivée plusieurs fois ? On part d'une idée de produit en se disant « chouette, on va butiner de l'énergie ». Après calcul, on se rend compte que l'énergie disponible est très faible, disons quelques µW. On passe alors beaucoup de temps à optimiser la consommation énergétique du produit. On finit, heureux, avec un montage ne consommant que quelques µA. Puis on se dit qu'il faut quand même une petite pile bouton, car la source peut ne pas être présente pendant quelque temps. Alors on constate, atterrés, qu'en fait cette petite pile bouton suffit pour alimenter toute seule le produit pendant des années vu qu'on a bien réduit sa consommation ! ◀

(180441-01)



@ WWW.ELEKTOR.FR

→ Kit de la lampe de lecture à élément Peltier

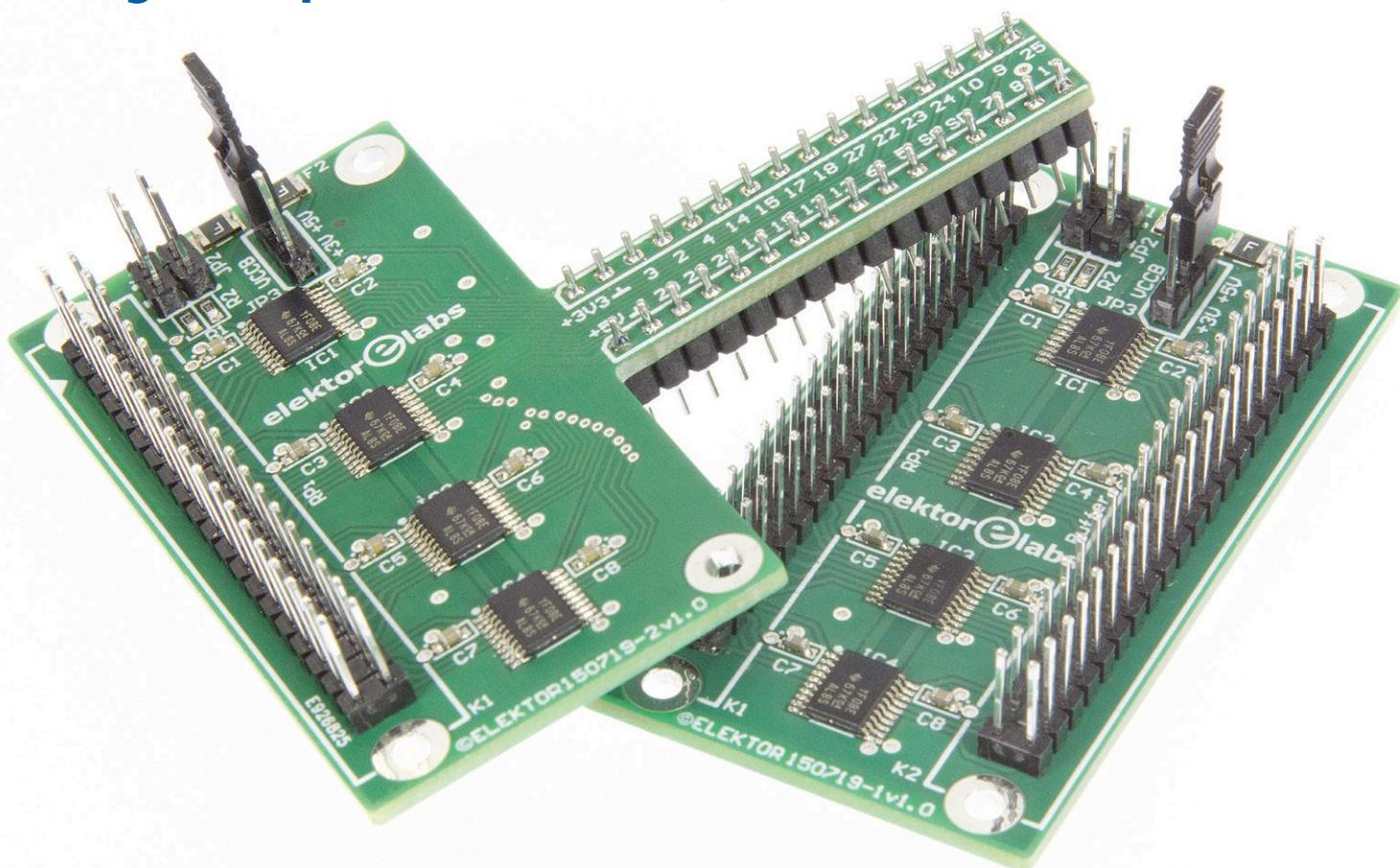
www.elektor.fr/peltier-lamp-1

Weblinks

- [1] « Hors circuits - Consommer moins avec du calcul et de la méthode », Elektor 01/2018 : www.elektormagazine.fr/160633
- [2] Système d'information géographique photovoltaïque : http://re.jrc.ec.europa.eu/pvg_tools/fr/tools.html
- [3] Base de données européenne sur la lumière du jour et les radiations solaires : www.satel-light.com/indexeS.htm
- [4] « de la bougie aux LED, modules Peltier convertis en générateur thermoélectrique », Elektor 09-10/2017 : www.elektormagazine.fr/160441

carte tampon pour le Raspberry Pi

ne grillez plus vos entrées/sorties



INFOS SUR LE PROJET



Raspberry Pi
HAT
T-Board



débutant
→ **connaisseur**
expert



30 min.



soudage de CMS



env. 35 €

Guy Weiler

Lorsque l'on expérimente avec le Raspberry Pi en connectant divers matériels aux entrées/sorties via le connecteur GPIO, il peut arriver qu'une tension trop élevée ou une décharge électrostatique détruise le port. La carte est alors bonne à mettre au rebut. Avec la carte tampon décrite ici, plus de souci à se faire !

Les étudiants sont des tueurs de semi-conducteurs ! Après avoir assisté, impuissant, à la mort de trois Raspberry Pi, l'auteur a décidé de concevoir une carte pour en protéger les entrées/sorties. Après quelques recherches, il est tombé sur la page de Brian Dorey [1], qui utilise des convertisseurs de niveau TXS0108E

de Texas Instruments, dont la mise en œuvre est très simple. Le projet initial de Guy Weiler, l'auteur de cet article [2], se limitait à 20 broches du port GPIO (qui possède plusieurs broches de masse). Sous 3,3 V, le montage fonctionnait parfaitement avec les bus 1-Wire, I²C et RS-232 ; sous 5 V, les signaux étaient

entachés de bruit. Le labo a repris le projet pour l'améliorer et éliminer ses défauts [3].

Le circuit

Nous avons profité de la révision pour réaliser deux versions. La première se contente de dupliquer le port GPIO, avec les mêmes connexions (figure 1, circuit imprimé 150719-1). La seconde version permet le raccordement direct de la carte tampon à une platine d'expérimentation

Caractéristiques

- Tampons bidirectionnels des broches du GPIO du RPI
- Convient pour des tensions de 1,2 à 5,5 V (voir texte)
- Version « T-board » pour raccordement à une platine d'expérimentation
- Protection contre les décharges électrostatiques

(figure 2, circuit imprimé 150719-2) ; elle a la forme d'une carte T-board, bien connue des lecteurs et n'encombrant pas trop la platine d'expérimentation (figure 3).

Les deux versions reposent sur le même circuit, bâti autour de quatre TXS0108E, des convertisseurs de niveau bidirectionnels sur 8 bits, utilisables dans des montages push-pull ou à collecteur ou

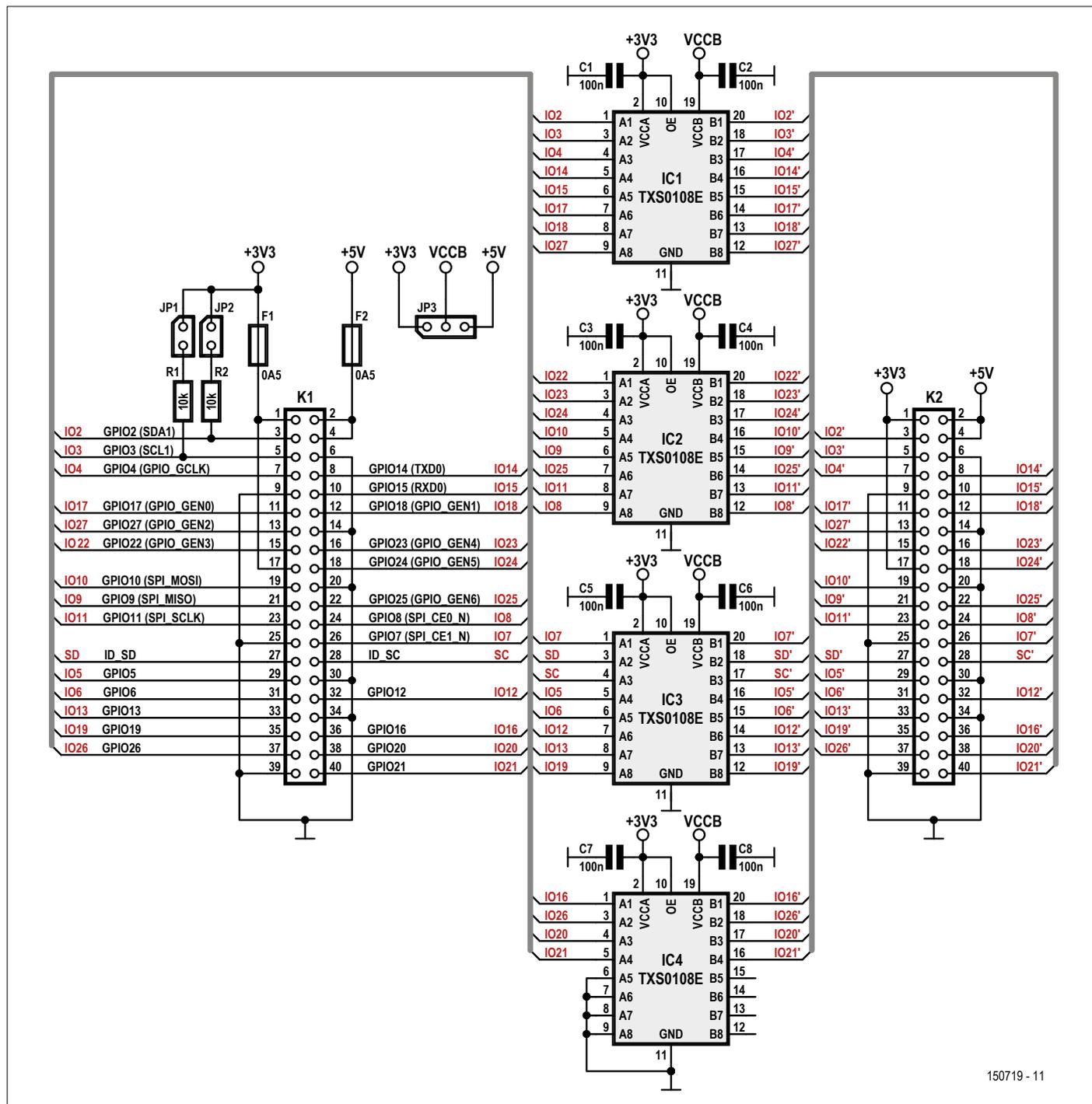


Figure 1. Le circuit est bâti autour de convertisseurs de niveau bidirectionnels sur 8 bits.

drain ouvert. La vitesse de transfert peut atteindre 60 Mbps et le circuit est protégé contre les décharges électrostatiques ; les signaux entre 1,2 et 3,6 V sur le port A peuvent être convertis en signaux de 1,65 à 5,5 V sur le port B. Non seulement les 26 broches du GPIO sont toutes tamponnées, mais les signaux SD et SC le sont aussi ; ils sont utilisés pour adresser la mémoire EEPROM qui identifie les cartes d'extension HAT (*hardware attached on top*) raccordées au RPi.

Les résistances de rappel du GPIO du RPi, si activées, sont sans effet et peuvent même être gênantes. Le TSXS0108E possède des résistances de tirage sur toutes les broches d'E/S, vers V_{CCA} sur le port A et V_{CCB} sur le port B ; la valeur des résistances est de 40 k Ω pour les sorties à l'état bas, 4 k Ω pour les sorties à l'état haut. Les sorties des tampons sont en fait des circuits à drain ouvert. Si on connecte par ex. une LED entre une sortie et la masse, la résistance série de la

LED et la résistance de tirage du tampon constituent un diviseur de tension ; avec une charge sur la sortie, le niveau haut n'est pas de 3,3 ou 5 V, mais moins, ce dont il faut tenir compte.

Les condensateurs C1 à C8 veillent au découplage de l'alimentation. Quant aux fusibles réarmables à coefficient de température positif (*Poly-Fuse®*) F1 et F2, ils protègent les alimentations 3,3 V et 5 V du RPi. On peut aussi connecter des résistances de tirage additionnelles

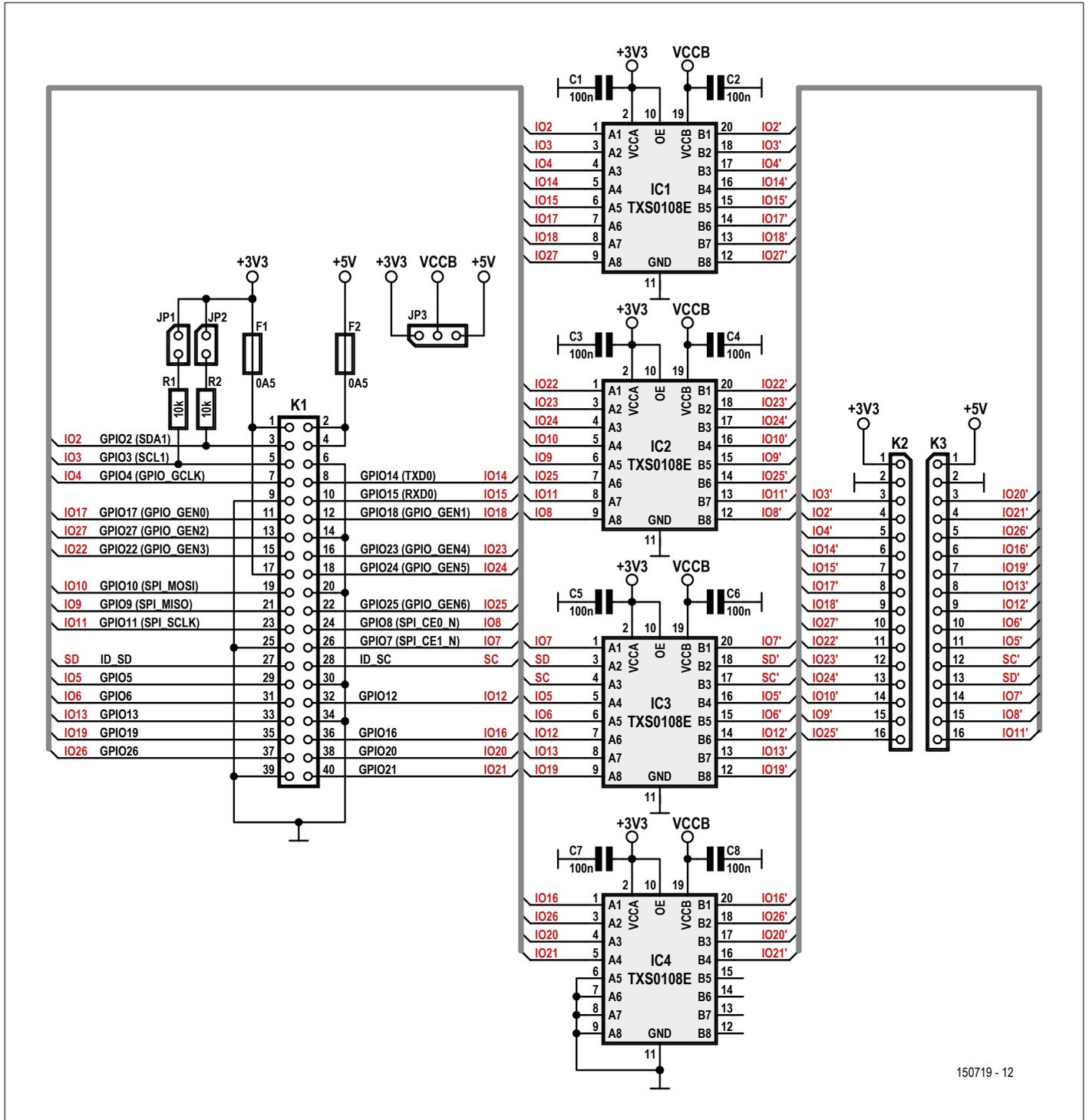


Figure 2. La version T-board offre les mêmes connexions, mais est plus compacte.

de 10 kΩ (R1 et R2) sur les lignes I²C avec les connecteurs JP1 et JP2 et leurs cavaliers. Enfin, le connecteur JP3 permet, avec son cavalier, la sélection du niveau haut des E/S (tamponnées) des ports B : 3,3 V ou 5 V.

Raccordement au RPi

Si K1 est un connecteur mâle standard de 2×20 broches, le raccordement au RPi se fera avec un câble plat à 40 brins muni à ses extrémités d'un connecteur femelle. On peut aussi utiliser un connecteur femelle extra haut de 2×20 broches (*stacking header*) et enficher le circuit directement sur le RPi, comme pour la plupart des cartes d'extension HAT. Cela permet en tous cas de rajouter une autre carte par-dessus la carte tampon. Ce n'est sans doute pas très judicieux avec la version T-board, car la platine d'expérimentation devrait être exactement à la bonne hauteur.

Les sorties de la carte standard (réf. 150719-1) peuvent aussi être raccordées à un circuit externe avec un câble à 40 brins muni de deux connecteurs à 2×20 broches ou une barrette à 40 contacts.

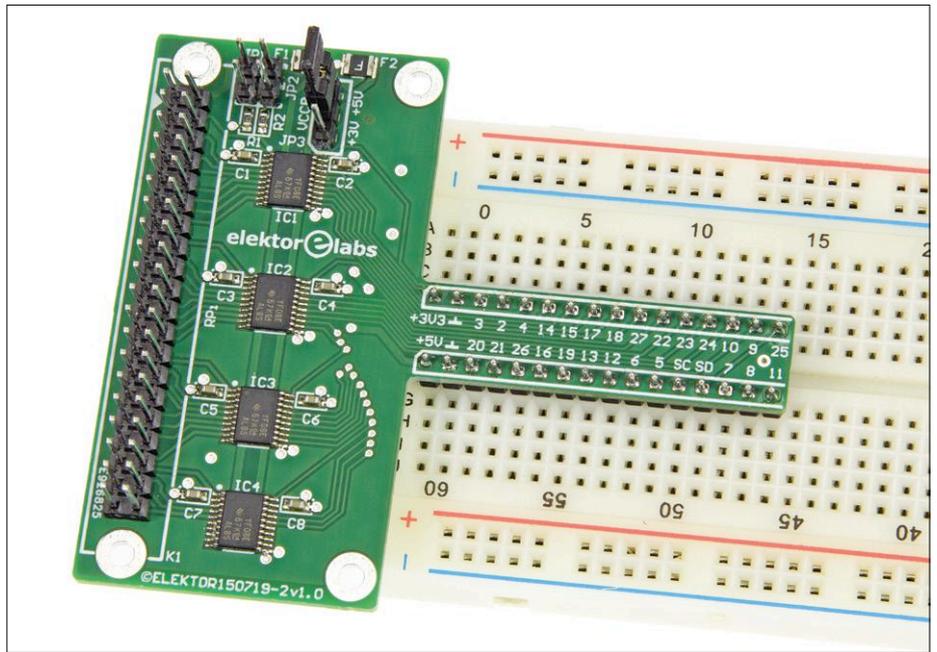


Figure 3. La forme en T permet de minimiser l'encombrement sur une platine d'expérimentation.

Pour le raccordement de la carte T-board à la platine d'expérimentation, nous avons choisi une barrette à 32 picots mâle-mâle du fabricant

Harwin, il faut la couper en deux (K2 et K3). Il faut souder les picots les plus épais sur le circuit imprimé ; les plus fins sont enfichés dans la platine d'ex-

Publicité

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.

Survolez 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.

Elektor
Elektor en kiosque

Dans quel magasin acheter cet article près de chez moi ?

Indiquez votre code postal

TROUVER

N°	Nom du point de vente	Adresse	Statut	Horaires de vente
11	AUCHAN	ROUTE DE MONTLUCON - CHATELAINQUE	✓ Disponible	02 64 62 61 61
12	CULTURA	ALLÉE DU COMMERCE - ST MAUR	✓ Disponible	02 64 62 61 60
13	CARRE D'AS	1 PLACE GARVOT - DODDS	✓ Disponible	02 64 34 17 53
14	CARRÉFOUR	ROUTE DE TOULOUSE - BOSSUILL	⚠ Ne sera pas en stock	02 64 31 54 32
15	BIAG PRESSE	C/COMMERCIAL DE TERRE NEUVE - DONBAT	✓ Disponible	04 70 01 99 96
16	BIAG PRESSE	1 RUE HENRI DORVILLE - DONBAT	✓ Disponible	04 70 01 99 92
17	MAISON DE LA PRESSE	6 PLACE FOURNIER - LINDOGES	✓ Disponible	02 64 32 62 66
18	CARRÉFOUR MARNIET	AVENUE GUSTAVE EIFFEL - DODDS	✓ Disponible	02 64 32 62 60
19	INTERMARCHÉ	44 RUE BERNARD VENTURAUD - LINDOGES	✗ Epuisé	02 64 32 70 40
20	RELAIS	K/BNCP LINDOGES VERTICALE - LINDOGES	✓ Disponible	02 64 77 38 72



LISTE DES COMPOSANTS

Version standard (circuit imprimé réf. 150719-1)

Résistances

R1, R2 = 10 k Ω , CMS 0603

Condensateurs

C1 à C8 = 100 nF, 50 V, 10%, X7R, CMS 0603

Semi-conducteurs

IC1-IC4 = TXS0108EPWR, CMS TSSOP-20

Divers

K1 = connecteur mâle à 2×20 broches, vertical, au pas de 2,54 mm, ou connecteur femelle GPIO à 40 broches extra haut
K2 = connecteur mâle à 2×20 broches, vertical, au pas de 2,54 mm
JP1, JP2 = connecteur mâle à 2 broches, vertical, au pas de 2,54 mm, avec cavalier
JP3 = connecteur mâle à 3 broches, vertical,

au pas de 2,54 mm, avec cavalier
F1, F2 = fusible réarmable CTP, CMS (1210L050YR Littelfuse)
Circuit imprimé, réf. 150719-1 (e-choppe [6])

Version T-board (circuit imprimé réf. 150719-2)

Résistances

R1, R2 = 10 k Ω , CMS 0603

Condensateurs

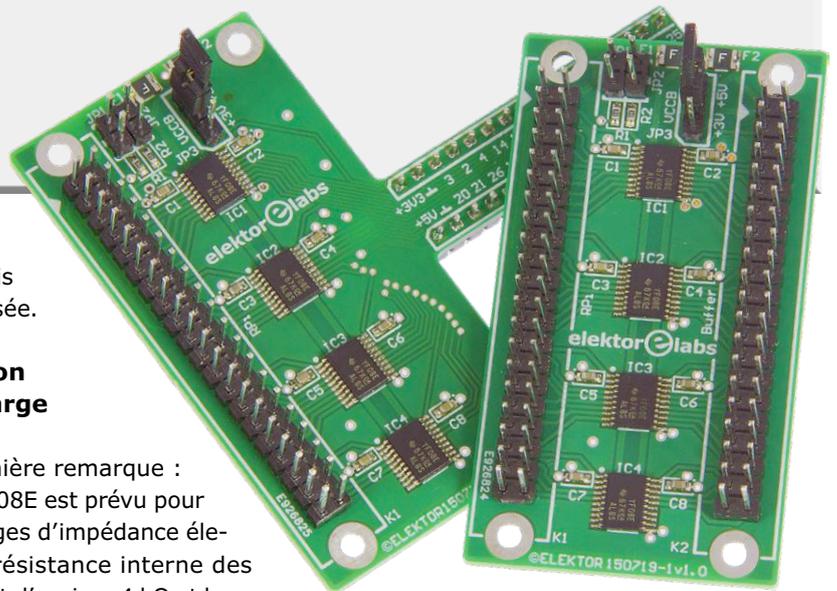
C1 à C8 = 100 nF, 50 V, 10%, X7R, CMS 0603

Semi-conducteurs

IC1-IC4 = TXS0108EPWR, CMS TSSOP-20

Divers

K1 = connecteur mâle à 2×20 broches, vertical, au pas de 2,54 mm, ou connecteur femelle GPIO à 40 broches extra haut
K2-K3 = barrette de 32 picots mâle-mâle, verticale, au pas de 2,54 mm
JP1, JP2 = connecteur mâle à 2 broches, vertical, au pas de 2,54 mm, avec cavalier
JP3 = connecteur mâle à 3 broches, vertical, au pas de 2,54 mm, avec cavalier
F1, F2 = fusible réarmable CTP, CMS (1210L050YR Littelfuse)
Circuit imprimé, réf. 150719-2 (e-choppe [7])



périmentation (ce qui évite de détériorer ses contacts).

Programmes de test

Deux programmes écrits en Python [5] permettent de tester la carte : un avec toutes les broches du GPIO configurées en sortie ([Check_all_GPIOs_as_output.py](#)), et l'autre – vous l'aurez deviné – avec les broches configurées en entrée ([Check_all_GPIOs_as_input.py](#)). Pour le test des sorties, on n'a besoin que de huit LED à faible consommation, le programme testant les sorties par groupe de 8.

La valeur des résistances série pour les LED est de 1,8 k Ω , mais ce n'est pas critique. Le but est de protéger les LED en cas de connexion directe à V_{cc} . Avec les sorties à drain ouvert du TXS0108E, la tension aux bornes d'une LED rouge avec sa résistance série est d'environ 2,6 V lorsque la tension sélectionnée avec JP3 est de 5 V.

Si les LED s'allument normalement, tout fonctionne comme prévu, la carte peut

désormais être utilisée.

Attention à la charge

Une dernière remarque : le TSX0108E est prévu pour des charges d'impédance élevée. La résistance interne des sorties est d'environ 4 k Ω et le courant ne pourra excéder quelques centaines de μ A. Si l'impédance de la charge est trop faible, le circuit peut devenir instable et le signal de sortie oscillera. Il faut en tenir compte lors du raccordement de charges externes à la carte ; pour de plus amples explications, n'hé-

sitez pas à consulter la fiche de caractéristiques du circuit [4] et d'autres documents sur la page du produit chez TI.

(180430-04 –
version française : Jean-Louis Mehren)



@WWW.ELEKTOR.FR

→ Circuit imprimé de la version standard (réf. 150719-1) :

www.elektor.fr/raspi-pcb-150719-1

→ Circuit imprimé de la version T-board (réf. 150719-2) :

www.elektor.fr/raspi-pcb-150719-2

Liens

- [1] Page web de Brian Dorey : www.briandorey.com/post/raspberry-pi-gpio-protection
- [2] Projet original de l'auteur sur le site du labo d'Elektor : www.elektormagazine.fr/labs/raspi-buffer-board
- [3] Page de l'article sur le site du labo d'Elektor : www.elektormagazine.fr/labs/buffer-boards-for-raspberry-pi-23
- [4] Documentation technique du TXS0108E : www.ti.com/product/TXS0108E/technicaldocuments
- [5] Page de l'article : www.elektormagazine.fr/180430-04
- [6] Circuit imprimé de la carte standard : www.elektor.fr/raspi-buffer-150719-1
- [7] Circuit imprimé de la carte en version T-board : www.elektor.fr/raspi-buffer-150719-2



electronica
fast forward
powered by elektor

the startup platform

PROPULSEZ VOTRE **PROJET NOVATEUR**

SUR

**LE MARCHÉ
INTERNATIONAL !**

● Participez à l'édition 2018
du concours

du 13 au 16 novembre 2018
à Munich

Les prix du concours electronica fast forward sont offerts par

Informations détaillées :
www.elektormagazine.com/e-ffwd



electronica



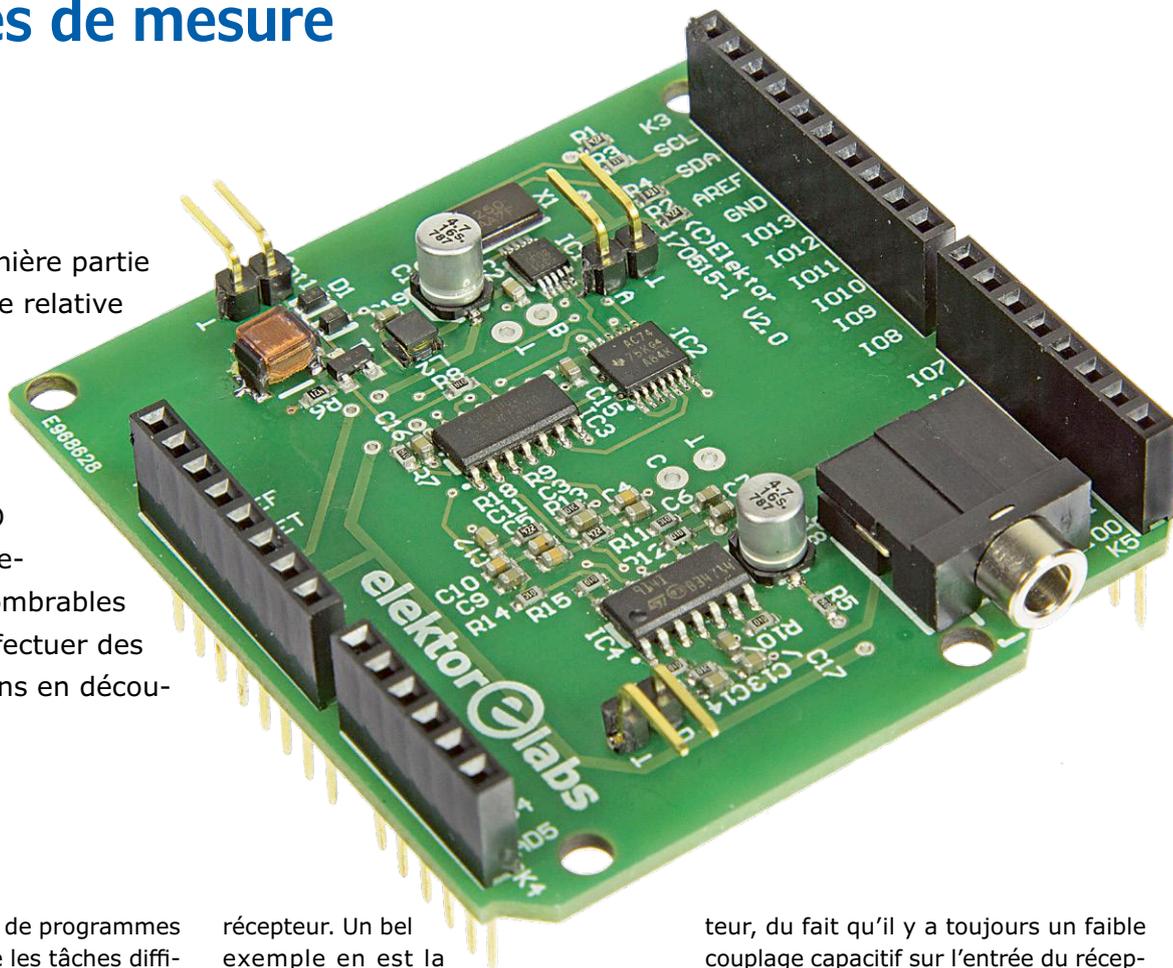
elektor

shield SDR 2.0 d'Elektor (3)

techniques de mesure

Burkhard Kainka

Venons-en à la dernière partie de notre petite série relative au *shield* SDR 2.0 d'Elektor [1]. Un récepteur à accord automatique et deux sorties VFO à piloter individuellement offrent d'innombrables possibilités pour effectuer des mesures. Nous allons en découvrir quelques-unes.



Avec le développement de programmes Arduino simples, même les tâches difficiles deviennent réalisables. Pas besoin d'acheter à tout coup des instruments de mesure onéreux, vous programmez simplement ce dont vous avez besoin.

Sous-porteuse

Dès qu'on branche un des canaux supplémentaires, un faible signal arrive au

récepteur. Un bel exemple en est la fréquence pré-réglée du canal A. Le milieu du domaine WSPR de seulement 200 Hz dans la bande des 40 m est à 7 040,100 kHz (fig. 1). Pour un accord précis, la sous-porteuse peut aider à l'orientation. Sans aucune liaison filaire, la sous-porteuse choisie est visible dans le récep-

teur, du fait qu'il y a toujours un faible couplage capacitif sur l'entrée du récepteur. À cause de cela, on n'atteint plus l'atténuation complète des bandes latérales. Le signal se manifeste éventuellement à +12 kHz et un peu moins fort à -12 kHz aussi. C'est dû à ce que le couplage n'a pas lieu rien qu'à l'entrée, mais aussi à d'autres endroits du récepteur, comme sur la sortie du mélangeur. On peut aussi recevoir des harmoniques. Comme ce sont des signaux rectangulaires symétriques qui sont produits, les harmoniques impairs dominent. La sortie à 1 MHz contient donc aussi du 3 MHz, 5 MHz, 7 MHz, etc. Normalement, on reçoit toutes les porteuses non modulées avec le réglage du récepteur sur la bande latérale supérieure (USB) ou l'inférieure (LSB). Pour deux porteuses proches l'une de l'autre, AM est aussi utilisable. Par exemple, on règle le canal A sur 20 000 kHz et le B sur 20 001 kHz. Sur 20 MHz, on reçoit alors un signal

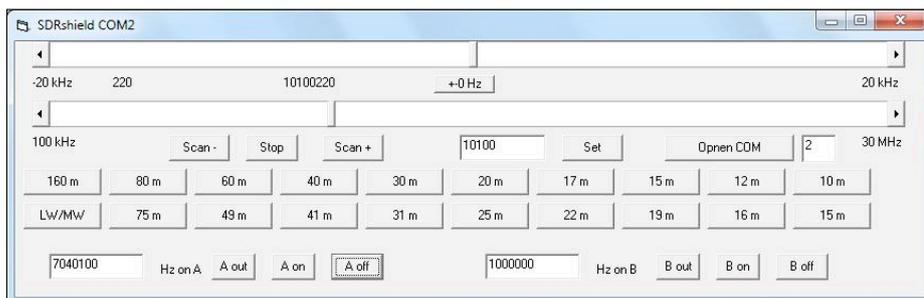


Figure 1. Réglage de la fréquence du VFO et des autres sorties.

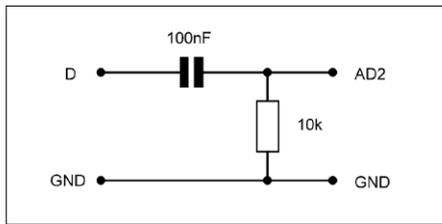


Figure 2. Mesure du niveau du signal.

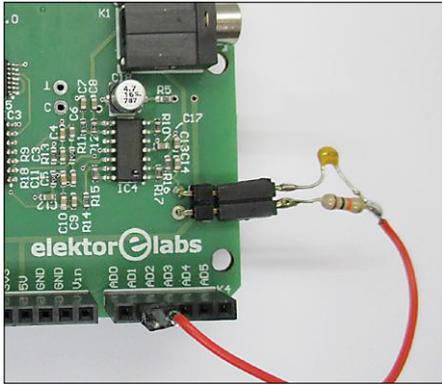


Figure 3. Liaison BF vers l'entrée de conversion A/N.

AM modulé à 1 kHz. Des amusements apparemment inutiles tels que ceux-ci peuvent aider à mieux comprendre le contexte et les possibilités du matériel, mais aussi ses limites. Dans les cas difficiles, vous pouvez entrevoir des problèmes inattendus. Et on peut utiliser les harmoniques d'amplitude connue comme signaux de référence pour étalonner le S-mètre (cf. ci-dessous).

Mesures sur systèmes à microcontrôleur

Le système composé de récepteurs et de générateurs peut se révéler utile dans de nombreuses situations. Voici un exemple tiré de la vie courante dans un labo d'électronique. Un système à deux contrôleurs présentait une panne de transmission sur l'interface série entre les contrôleurs. Il est logique de commencer par mesurer les fréquences d'horloge. Pas question d'y brancher un fréquencemètre, il perturberait chacun des oscillateurs. Alors, on écoute les signaux couplés très faiblement au SDR, avec un fil d'antenne placé dans les environs. L'oscillateur à céramique à 12 MHz a pu être reçu directement, il travaillait dans les tolérances prévues. Le second contrôleur, muni d'un quartz à 40 MHz a été perçu par mélange avec un har-

Listage 1. Mesure de l'occupation de bande HF (croquis *RFplotdB*).

```
void uac(void)
{
  long uac =0;
  for (int i=0; i <= 200; i++){
    uac = uac+ analogRead(A2);
  }
  //if (uac > 400) uac = 400;
  u=20 * log10(uac);
  Serial.println(u);
}

void loop(void)
{
  for (int i=1; i <= 498; i++){
    int f= i * 50;
    si5351.set_freq((f+5)*400000, SI5351_PLL_FIXED, SI5351_CLK1);
    if (i== 1) delay (500);
    delay(10);
    uac();
  }
  while ( (digitalRead(A1) == 1));
}
```

monique à 13,333 MHz. Une fréquence parfaitement normale aussi. Pourtant, des mesures à l'oscilloscope ont montré que le débit binaire du plus rapide des contrôleurs était trop élevé de 5 %, ce qui dénote une erreur du logiciel. Une preuve de plus a été fournie par l'injection d'un signal d'horloge externe à l'oscillateur céramique à 12 MHz du second contrôleur, en provenance de la sortie A du *shield* SDR. Il est ainsi apparu que la fréquence d'horloge devait passer à au moins 12,3 MHz pour que la transmission série fonctionne impeccablement. Le *shield* SDR a ainsi aidé à analyser le problème.

Affichage de l'occupation des bandes

Les conditions de travail en ondes courtes varient d'heure en heure. Il est donc intéressant de savoir à quel moment chaque bande est fort exploitée et offre de bonnes conditions de propagation. On utilise pour cela un récepteur à accord automatique et on mesure la force des signaux. Pour ce faire, on n'a pas besoin d'un décodage complet des signaux IQ, il suffit de mesurer la tension du signal sur l'une des sorties. C'est ce que fait Arduino tout seul, pour peu que les signaux ne soient pas extrême-

ment faibles. Le signal à basse fréquence sur la broche D est porté à un potentiel de masse moyen et envoyé directement au canal ADC2 du convertisseur A/N (**figures 2 et 3**). Comme ce convertisseur ne mesure que des tensions positives, il opère en pratique un redressement du signal. La valeur moyenne sur 200 mesures est une valeur fiable de la tension du signal.

Le programme *RFplotB* (**listage 1**, téléchargement de tous les programmes sur [2]) contient une boucle de réception pour aller de 0 à 25 MHz. On prend les moyennes des tensions des signaux à la sortie du *shield*, on en calcule les logarithmes pour les présenter en dB, par l'intermédiaire du traceur de courbe

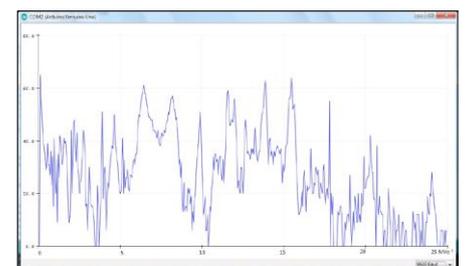


Figure 4. Le spectre capté par un dipôle en plein air.

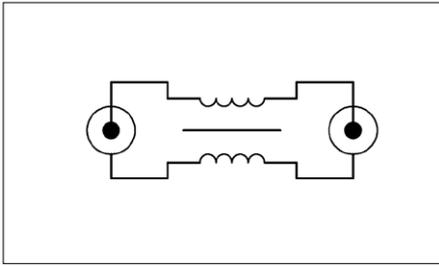


Figure 5. Le principe du symétriseur d'antenne ou *balun*.



Figure 6. Du câble coaxial bobiné sur un noyau de ferrite.

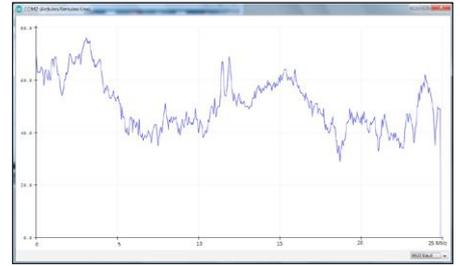


Figure 7. Les signaux issus d'une antenne intérieure.

sériel de l'EDI Arduino. Comme la largeur de la fenêtre peut contenir 500 valeurs, le domaine de fréquence jusqu'à 25 MHz permet une résolution de 50 kHz. Cela représente aussi à peu près la largeur de bande du récepteur sans logiciel SDR en aval.

La **figure 4** représente l'occupation de la bande vers minuit captée par un dipôle de deux fois 7,5 m de long, placé en plein

air. En complément, j'ai installé un *balun* (symétriseur/asymétriseur) pour empêcher le bruit ambiant d'atteindre l'entrée du récepteur. Il est composé d'un anneau de ferrite porteur de plusieurs spires de câble coaxial fin (**figures 5 et 6**). Cette bobine arrête les parasites captés par le blindage extérieur (le manteau) du coaxial. Preuve de l'efficacité du dispositif : une dynamique de 60 dB atteinte.

Par comparaison, la **figure 7** montre une mesure sur une antenne intérieure constituée d'une boucle de 30 m de fil. On reconnaît par endroits des niveaux de signal similaires, mais nulle part dans le spectre le bruit ne descend sous les 30 dB. Les mesures coïncident avec les résultats réels de la réception sur plusieurs fréquences, mais un bruit de fond trop élevé est partout perturbant.

Listage 2. Mesure d'un quadripôle jusqu'à 25 MHz (croquis *RFplotU*).

```
void uac(void)
{
  long uac = 0;
  for (int i=0; i <= 200; i++){
    uac = uac + analogRead(A2);
  }
  u = uac/50;
  Serial.println (u);
}

void loop(void)
{
  Serial.println (0);
  for (int i=1; i <= 500; i++){
    int f= i * 50;
    //Serial.print (f); Serial.print (" ");
    si5351.set_freq(f*100000, SI5351_PLL_FIXED, SI5351_CLK0);
    delay(10);
    si5351.set_freq((f+5)*400000, SI5351_PLL_FIXED, SI5351_CLK1);
    if (i== 1) delay (500);
    delay(10);
    lcd.setCursor(0, 0);
    lcd.print (f);
    lcd.print (" ");
    delay(20);
    uac();
  }
  while ( (digitalRead(A1) == 1));
}
```

Mesures sur quadripôles

Pour analyser des dipôles ou des quadripôles passifs sur une large gamme de fréquences, on utilise un oscillateur accordable accompagné d'un récepteur. C'est facile à réaliser avec le *shield* SDR. La boucle de réception disponible pour aller de 0 à 25 MHz est prolongée d'une sortie de VFO qui suit la fréquence de réception à un écart constant de 5 kHz (**listage 2**). Le récepteur est utilisé comme mélangeur direct dont le signal à 5 kHz en sortie indique l'amplitude du signal reçu.

Pour des mesures sur de simples quadripôles, on se sert d'un diviseur de tension qui amène l'amplitude du signal dans le domaine permis à l'entrée du récepteur, environ 200 mV. Je l'ai composé de résistances de 1 k Ω et 51 Ω (**figures 8 et 9**) ce qui procure au générateur une impédance de sortie voisine de 50 Ω . Il faut aussi boucler chaque réseau à tester au moyen d'une résistance de charge appropriée. Dans l'exemple présenté, on mesure un filtre passe-bas pour émetteur d'ondes courtes avec une impédance de 50 Ω .

Le résultat des mesures est à la **figure 10**, il montre une fréquence limite de 13 MHz environ. L'atténuation est nettement supérieure à dix fois (20 dB) aux environs de 18 MHz. La légère remontée du signal de sortie à partir de 20 MHz est attribuable à l'inductance et la chute de tension sur les conducteurs de masse qui,

dans ce cas-ci, sont des câbles nus. De nombreuses mesures ont montré que, dans cette gamme de fréquences, une longueur de fil de quelques centimètres peut déjà conduire à des erreurs de mesure, à moins de n'utiliser que des coaxiaux à l'impédance caractéristique correcte et un boîtier blindé.

Observons à l'oscilloscope le signal BF de sortie du récepteur : c'est un signal triangulaire. Quand on mélange deux signaux rectangulaires, on obtient toujours un triangle du fait que les deux signaux défilent avec un déphasage mutuel constant. On peut aussi considérer que le signal de sortie contient des harmoniques qui sont appliqués au récepteur par mélange d'harmoniques. Dès qu'on utilise un filtre passe-bas suffisamment efficace, c'est un signal BF sinusoïdal qui sort. Utiliser des signaux rectangulaires conduit à des erreurs de mesure à cause des harmoniques. En pratique, on peut souvent négliger l'influence néfaste.

Étalonnage des niveaux de réception

Le S-mètre d'un récepteur d'ondes courtes doit afficher S9 pour 50 µV sur l'entrée d'antenne. L'écart entre deux nombres correspond à 6 dB, d'où S8 = 25 µV, S7 = 12,5 µV, S6 = 6 µV, S5 = 3 µV, etc. Cependant, la plupart des S-mètres ne sont pas très précis. Pour les étalonner, on utilise un signal HF fiable dont le niveau est connu. En reliant directement l'atténuateur de la sortie A à l'entrée d'antenne avec une résistance de 50 Ω en parallèle (un atténuateur comme à la figure 8), on obtient un signal d'un niveau défini. On procède ainsi pour étalonner un S-mètre ou encore ajuster la sensibilité du SDR sur le réglage de la carte son.

Le signal rectangulaire sur la sortie A a toujours une tension de 3,3 V_{crête-crête} donc une amplitude de 1,65 V. L'atténuateur avec 1 kΩ et 25 Ω introduit un facteur de 41, donc 41,2 mV. C'est encore considérablement trop comme signal d'antenne normal, cela correspond à S9 + 60 dB.

Or, cela favorise considérablement la réception des harmoniques les plus faibles. Selon Fourier, un signal carré d'amplitude A contient le fondamental f₀ et tous les harmoniques impairs (f₀ × k) avec l'amplitude 4 A / (pi × k). La fréquence fondamentale a dans ce cas une amplitude sinusoïdale de 52,5 mV,

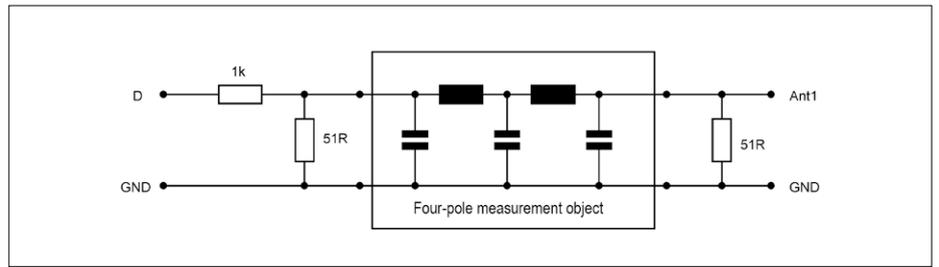


Figure 8. Mesure sur un quadripôle, un filtre passe-bas.

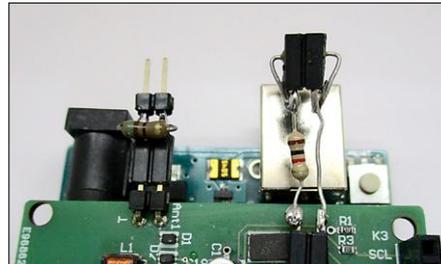


Figure 9. Prise et résistance pour la mesure d'un quadripôle.

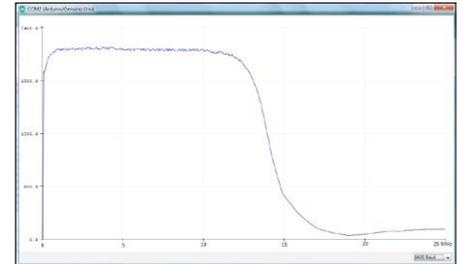


Figure 10. Courbe caractéristique d'un filtre passe-bas.

ce qui correspond à une tension efficace de 37,1 mV. Mais pour le millième harmonique, il ne reste que 37,1 µV. Plus précisément, il faut chercher le signal à f₀ × 999 ou celui à f₀ × 1 001, puisque tous les harmoniques pairs d'un signal carré sont nuls.

Concrètement, on règle la fréquence pour la sortie A sur 10 kHz et on trouve à 10 010 kHz un signal de 37 µV. Ou bien on cherche un harmonique à 7 MHz, plus précisément à 7 430 kHz, et on arrive là à 50 µV, soit exactement S9. Si l'on règle le VFO du récepteur sur 7 000 kHz, on voit directement des signaux relativement forts à -10 kHz et +10 kHz. Ce sont des signaux de la fréquence fondamentale à 10 kHz qui ont traversé le mélangeur. Dès qu'on décale la fréquence du récepteur de 1 kHz vers 7 001 kHz, on

retrouve le signal à +9 kHz (harmonique sur 7 010 kHz) et sur -11 kHz (harmonique sur 6 990 kHz).

De même, on trouve d'autres harmoniques toujours à distance de 20 kHz. Tous ces signaux ont une tension de 50 µV, ils correspondent donc à S9. Vous pouvez ainsi régler convenablement la sensibilité d'entrée du SDR. Le signal S9 pourrait bien se situer à -10 dB. Un signal de force S1 serait alors 8 × 6 dB = 48 dB plus bas, aux environs de -60 dB. C'est à peu près là que se situe le bruit de fond du récepteur sans antenne connectée.

Le signal riche en harmoniques convient très bien aux bandes pour radioamateurs. Quand on reçoit du S9 à 7 MHz, il y a encore du S8 à 14 MHz et du S7 à 28 MHz. La mesure sur le shield SDR confirme ces niveaux, ce qui prouve

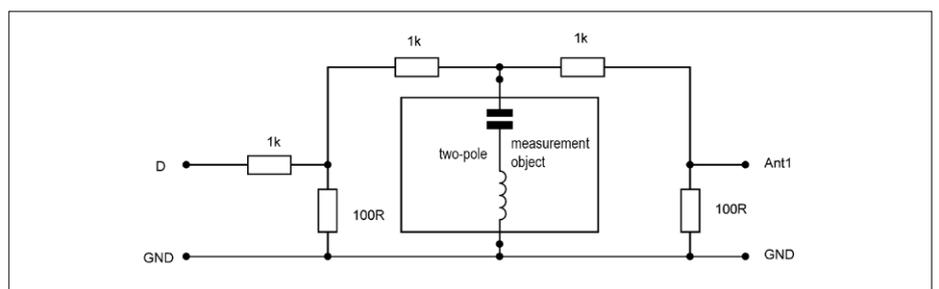


Figure 11. Mesure d'un dipôle.

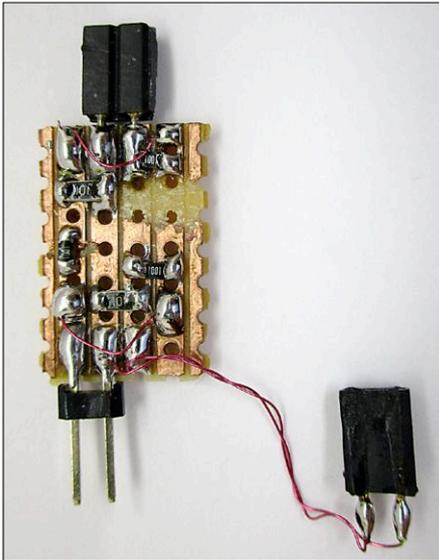


Figure 12. Construction avec des CMS pour réduire l'inductance parasite.

encore une fois que la sensibilité du récepteur est constante sur la totalité du domaine jusqu'à 30 MHz.

Mesures sur dipôles

Nous mesurons ici la caractéristique de fréquence d'une impédance. Le schéma simple de raccordement selon la **figure 11** met en œuvre un diviseur de tension à haute résistance avec 1 k Ω et le réseau à tester. Un autre diviseur de tension conduit à l'entrée du récepteur. Le circuit de la **figure 12** a été

construit avec des résistances CMS de la plus faible inductance possible. Avec ce circuit simple, on ne peut pas faire la différence entre résistance réelle et réactance. Ce que nous mesurons, c'est l'impédance en ohms. Une résistance mesurée de 100 Ω peut ainsi être réelle, ohmique, ou par exemple être une réactance capacitive.

La tension mesurée est convertie en résistance du réseau en test et fournie de manière sérielle (**listage 3**). En outre, le LCD affiche en permanence la fréquence actuelle. Le bouton 2 du *shield* LCD lance une nouvelle opération. L'ensemble des valeurs d'impédance est stocké dans le tableau $d[i]$, de sorte que l'on puisse mesurer directement à la base d'une antenne, par exemple, même sans connexion au PC. Le bouton S1 déclenche la sortie des données stockées.

Pour des mesures autonomes, il faut brancher une alimentation externe. De plus, il convient de ponter la ligne *Reset* d'Arduino au +5 V pour la bloquer. Quand on rebranche la carte au PC et qu'on démarre le traceur graphique sériel, celui-ci tente de lancer une réinitialisation qui effacerait les données de la RAM. Avec la ligne de RàZ bloquée, les données sont protégées, on pourra les envoyer au traceur avec le bouton S1.

La **figure 13** montre la courbe de l'impédance d'un filtre en pi composé de deux condensateurs variables et d'une bobine. On y voit une résonance série avec la plus petite impédance à 7 MHz et une résonance parallèle avec un maximum d'impédance à 11 MHz.

La **figure 14** représente la courbe d'impédance d'une antenne intérieure de fortune, confectionnée avec 30 m de fil de cuivre émaillé de seulement 0,2 mm d'épaisseur, fixée aux murs pour entourer deux pièces. On y reconnaît clairement une résonance juste en dessous de 10 MHz, comme on peut s'y attendre avec une antenne quad d'une ampleur de toute la longueur d'onde. Une deuxième résonance est identifiable à environ 18 MHz. Avec un filtre adaptateur en pi, les radioamateurs l'utilisent comme antenne émettrice, mais comme antenne réceptrice, elle pêche par un bruit de fond trop élevé (cf. figure 7).

Pont de mesure d'ondes stationnaires

Le câble coaxial entre l'antenne et un récepteur ou un émetteur a généralement une impédance de 50 Ω . S'il est

Listage 3. Mise en œuvre de la mesure d'un dipôle (fragments du croquis *RFplotR*).

```
void uac(void)
{
  long uac = 0;
  for (int i=0; i <= 200; i++){
    uac = uac+ analogRead(A2);
  }
  if (uac>10000) uac=10000;
  r = uac*600/(14000-uac);
  // Serial.print (uac);
  // Serial.print (" ");
  Serial.println (r);
}

void loop(void)
{
  for (int i=1; i <= 500; i++){
    int f= i * 50;
    // Serial.print (f); Serial.print (" ");
    si5351.set_freq(f*100000, SI5351_PLL_FIXED, SI5351_CLK0);
    delay(10);
    si5351.set_freq((f+5)*400000, SI5351_PLL_FIXED, SI5351_CLK1);
    if (i== 1) delay (500);
    delay(10);
    lcd.setCursor(0, 0);
    lcd.print (f);
    lcd.print (" ");
    delay (19);
    uac();
    d[i] = r;
  }
  while ( (digitalRead(A0) == 1));
  for (int i=1; i <= 500; i++){
    Serial.println (d[i]);
  }
  while ( (digitalRead(A1) == 1));
}
```

terminé par une résistance ohmique de 50Ω , il n'y a pas de réflexion, toute la puissance injectée dans le câble ne circule que dans une direction. En revanche, si le câble est ouvert ou en court-circuit, il y a réflexion totale. L'onde de retour se superpose à celle qui entre, produisant des ondes stationnaires. Le câble se comporte alors plutôt comme un circuit oscillant dans lequel l'énergie va et vient et se transforme complètement en chaleur, perdue dans la résistance du câble. Une antenne dipôle idéale présente une résistance d'environ 50Ω à sa fréquence propre. Il y a alors accord parfait et très peu de pertes. Sur une antenne moins parfaite, il y a une déviation de la résistance nominale, souvent accompagnée d'une réactance capacitive ou inductive. Il se produit une réflexion partielle avec ondes stationnaires et de grandes pertes dans le câble. Ce n'est pas grave pour un récepteur, du fait que sa sensibilité est souvent plus grande que nécessaire. Mais les émetteurs dépendent d'une impédance de raccordement correcte. Pour y parvenir, on mesure le taux d'ondes stationnaires (TOS) pour avoir une indication sur la dérive par rapport à l'impédance caractéristique de 50Ω . La solution : un simple pont de mesure (**figures 15 et 16**). Quand l'objet à mesurer fait exactement 50Ω , il n'y a pas de tension sur le pont. Le petit transformateur à large bande dans le pont transmet une différence entre branches du pont quand se produit une dérive par rapport à une impédance réelle de 50Ω . En cas de court-circuit ou d'ouverture, on mesure une tension élevée, la même que pour un taux d'ondes stationnaires infiniment grand. Cependant, la résistance interne du générateur provoque un déséquilibre qui peut être compensé par des résistances plus petites dans la partie supérieure du diviseur de tension. J'avoue que le transformateur à noyau annulaire utilisé provenait d'un fond de tiroir et n'était pas destiné à cette gamme de fréquences, il a pourtant fonctionné impeccablement. Une mesure sur une résistance de 50Ω doit toujours donner un résultat nul, ce qui correspond à un TOS de 1,0. Avec 100Ω ou 25Ω , on trouve un taux de 2. Des mesures sur des résistances réelles (avec des fils très courts !) peuvent démontrer la précision du pont de mesure. De petites erreurs sont possibles à cause du transformateur utilisé

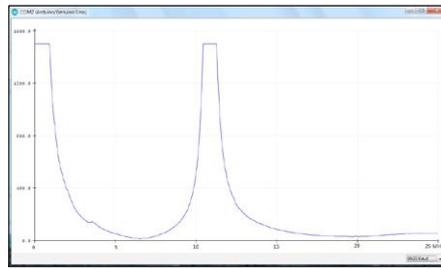


Figure 13. Courbe d'impédance avec résonance série et parallèle.

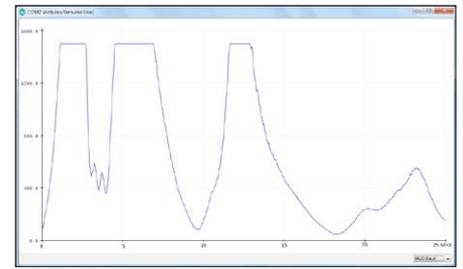


Figure 14. Courbe d'impédance d'une boucle d'antenne.

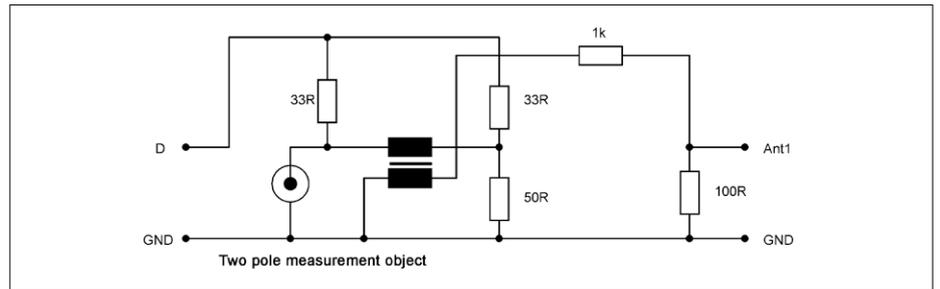


Figure 15. Le pont de mesure des ondes stationnaires.

ou s'il y a des fils trop longs dans le pont. Mais on peut les négliger, ce qui importe, c'est de trouver la largeur de bande de l'antenne et de déterminer à quelle fréquence se situe le meilleur TOS à la résonance. En principe, une simple mesure d'impédance suffirait, mais le pont de mesure des ondes stationnaires détecte mieux les impédances indésirables. Le logiciel du **listage 4** ne diffère guère de la simple mesure de réponse en fréquence.

La mesure de la **figure 17** montre le résultat avec un câble d'antenne ouvert. Aux points où le TOS est le plus bas, on est en présence d'une résistance réelle. D'ailleurs, le câble est ici en résonance

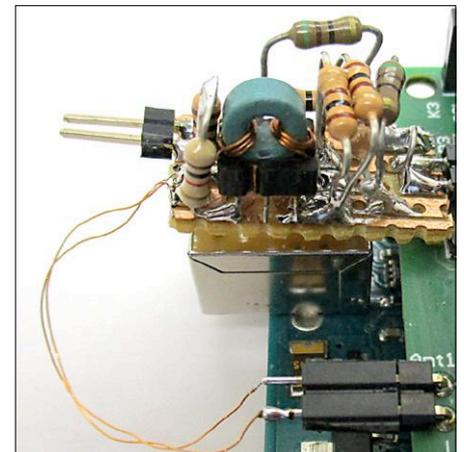


Figure 16. Pont de mesure avec transformateur à noyau ferrite annulaire.

Listage 4. Évaluation du TOS, le taux d'ondes stationnaires (croquis RFplotSWR).

```
void uac(void)
{
    long uac =0;
    for (int i=0; i <= 200; i++){
        uac = uac+ analogRead(A2);
    }
    snr = uac/50;
    if (snr > 400) snr = 400;
    Serial.println (snr);
}
```

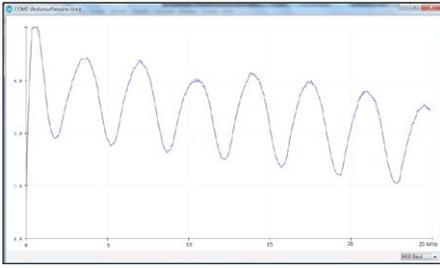


Figure 17. Mesure des ondes stationnaires sur un câble à sortie ouverte.

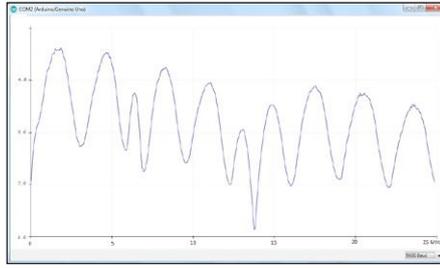


Figure 18. Résonance à 14 MHz.

bien. C'est pourquoi on obtient une meilleure marge au bruit au point de résonance.

De nombreux appareils de mesure présentés ici méritent des améliorations, en particulier dans leur construction et la connectique. Mais ils montrent la possibilité de réaliser des mesures utiles à peu de frais. Le *shield* SDR est devenu un auxiliaire pratique dans bien des aspects de la technique HF. ◀

(180284-03 -

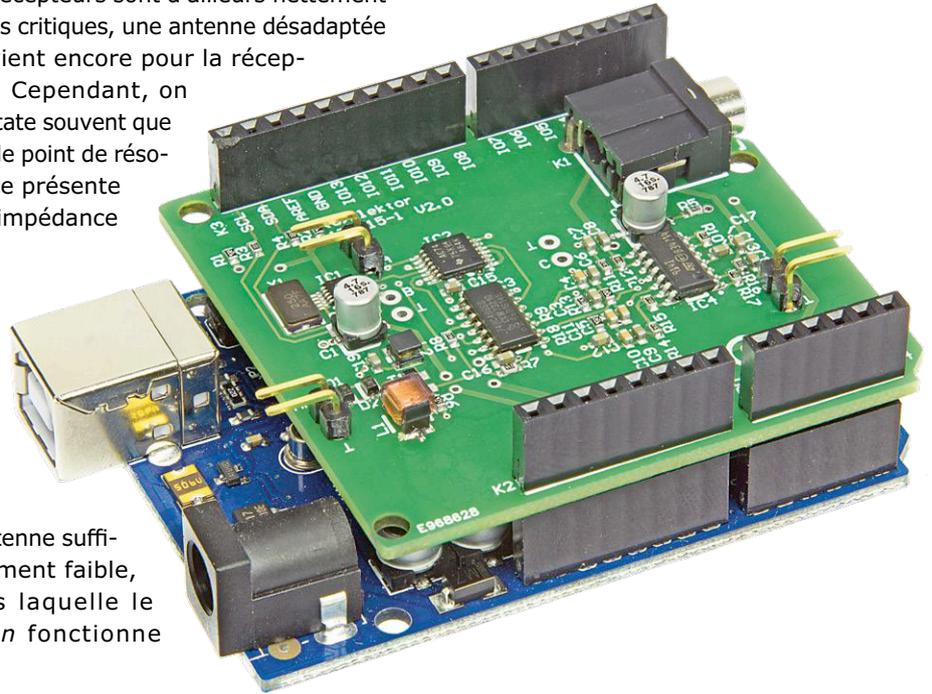
version française : Robert Grignard)

sur un multiple impair d'un quart de longueur d'onde. La totalité de l'énergie introduite dans le câble est transformée en chaleur. L'amélioration du TOS visible dans les fréquences plus hautes, jusqu'à environ 2,0, est uniquement due à l'augmentation de l'affaiblissement avec la montée en fréquence. Il convient d'en tenir compte pour juger correctement une antenne connectée.

La **figure 18** montre une antenne dipôle de $2 \times 7,5$ m avec son câble de descente. Il y a au pied de l'antenne un transformateur symétriseur avec rapport de transformation variable (**figures 19** et **20**) pour la faire fonctionner en dehors de sa résonance propre à 10 MHz. Avec le rapport de transformation adapté, il a été possible de travailler à 14 MHz avec un TOS de 1,2. Sur 7 MHz aussi, on a atteint un TOS d'à peu près 2. Cela a permis de couvrir les bandes radioamateur de 40 m

et de 20 m avec la même antenne. Les récepteurs sont d'ailleurs nettement moins critiques, une antenne désadaptée convient encore pour la réception. Cependant, on constate souvent que seul le point de résonance présente une impédance

d'antenne suffisamment faible, dans laquelle le *balun* fonctionne



Liens

- [1] *Shield* SDR 2.0 d'Elektor (1) : www.elektormagazine.fr/160577
- [2] Croquis à télécharger : www.elektormagazine.fr/180284-B-03



@WWW.ELEKTOR.FR

→ *Shield* SDR 2.0 d'Elektor
www.elektor.fr/sdr-shield-2

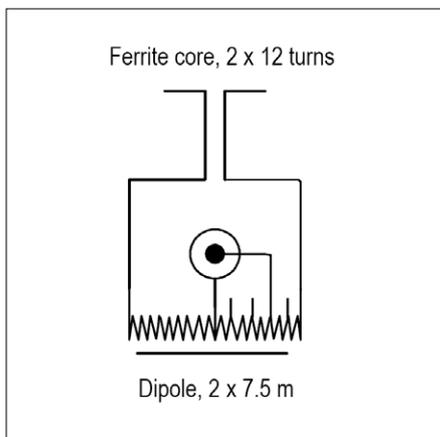


Figure 19. Symétriseur et adaptation d'impédance.

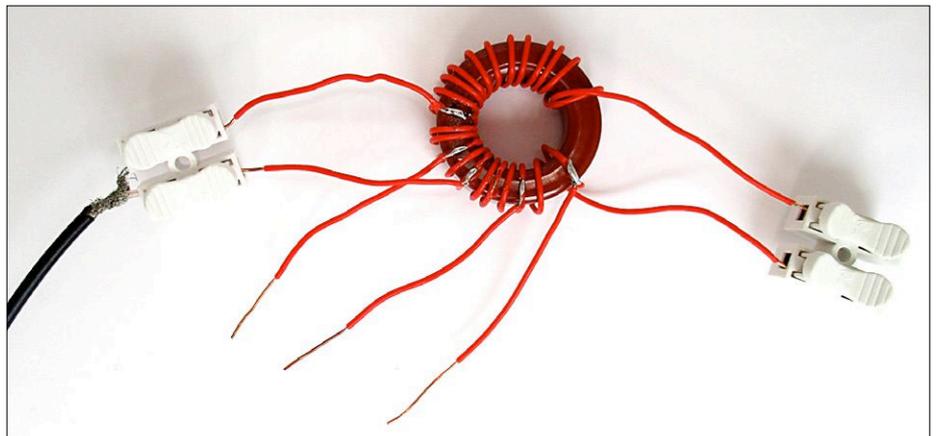


Figure 20. Adaptateur et symétriseur à transformateur à noyau de poudre de fer.

Participez au concours

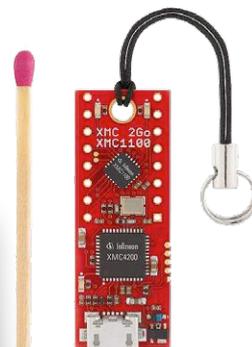
elektor board games

concours parrainé par AVNET SILICA
et EBV ELEKTRONIK

Premier prix*

Composants électroniques pour un montant de 5.000 €

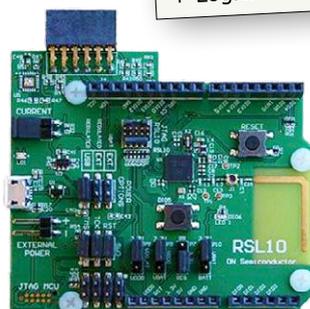
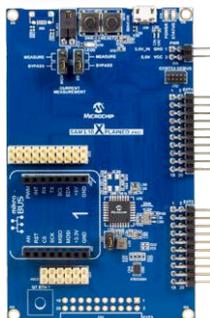
- + Assistance pour transformer votre idée en produit commercialisable
- + Logiciel Product Planner de Dragon Innovation
- + Présentation de votre projet lors du salon *electronica 2018*



Deuxième prix*

Composants électroniques pour un montant de 2.500 €

- + Assistance pour transformer votre idée en produit commercialisable
- + Logiciel Product Planner de Dragon Innovation



Troisième prix*

- Assistance pour transformer votre idée en produit commercialisable
- + Logiciel Product Planner de Dragon Innovation



* Avnet Silica et EBV Elektronik sélectionnent **chacun** un gagnant pour les premier, deuxième et troisième prix. Il y a donc **six** prix décernés au total.

Comment participer ?

- Choisissez une carte parmi les six kits proposés.
- Complétez le formulaire.
- Attendez l'acceptation de votre proposition de projet.
- Passez à l'action !

Bon concours « Board Games 2018 » à tous !

Avnet Silica, EBV Elektronik et l'équipe du labo d'Elektor

Lien

- [1] [Concours « Board Games » d'Elektor :
www.elektormagazine.fr/widgets/board-games](http://www.elektormagazine.fr/widgets/board-games)

À propos d' Avnet EMEA

Avnet EMEA est leader de la distribution de composants en Europe, Moyen-Orient et Afrique (EMEA). Grâce à ses divisions opérationnelles indépendantes, Avnet EMEA épaula ses clients depuis la conception des produits jusqu'à la fabrication en grandes séries et la gestion de la fin de vie des produits. Avnet Silica et EBV Elektronik distribuent les produits des fabricants de semi-conducteurs majeurs ; leurs ingénieurs sont aux côtés des clients pour transformer des idées en produits capables de changer la donne sur le marché.

convertisseur USB-DMX compact entièrement isolé

Jochem Brouwers (stagiaire au labo d'Elektor)

Le prix élevé des convertisseurs USB vers DMX512 correctement isolés m'a conduit à concevoir un équivalent « à construire soi-même », évidemment moins coûteux mais offrant la même garantie d'isolement.



Il existe de nombreux convertisseurs bon marché destinés à la commande des systèmes d'éclairage de salles de concert, théâtres et autres discothèques. Ces convertisseurs souffrent malheureusement d'un grave défaut : ils ne sont pas isolés de la dangereuse alimentation CA du système. Un court-circuit dans n'importe lequel des appareils DMX risque donc de griller le port du PC de commande. Gage de **sécurité électrique**, ce projet devrait renvoyer dans les coulisses bon nombre des convertisseurs d'entrée de gamme.

Éclairons d'abord votre lanterne

Le circuit (**fig. 1**) est relativement simple. Il repose sur l'interface *Open DMX USB* d'Enttec, avec ceci en plus qu'il fournit un isolement complet. Voyons comment en examinant le schéma, par ailleurs l'une des dernières belles œuvres de notre dessinateur Mart Schroyen, parti en retraite au mois d'août dernier après plus de 40 années passées chez Elektor. Vous avez certainement identifié l'inter-

face USB-UART série sous les traits de cette bonne vieille puce FT232RL (*IC1*). Elle prend en charge à elle seule tout le protocole USB, autrement dit aucun matériel USB particulier n'est nécessaire. *IC2* (WPMIB9200501S) est un convertisseur CC/CC isolé – notez la diagonale sur son symbole, de même que l'identification « USB » sur la masse côté entrée et l'absence d'identification sur la masse côté sortie (côté DMX).

IC4 est un circuit de surveillance de microprocesseur, un ADM-809SARTZ-REEL7 d'Analog Devices. Il fournit un signal RESET actif à l'état logique bas lors des mises sous et hors tension, ainsi que lors des baisses de tension. À la mise sous tension, un temporisateur interne active RESET durant 240 ms.

IC3 est le cœur du circuit. Il s'agit d'un ADM2483BRWZ, un transcepteur RS-485 semi-duplex isolé, lui aussi d'Analog Device. Il fait partie de la famille iCoupler® qui offre des solutions d'isolement

bon marché pour les systèmes numériques. Le fabricant le décrit comme un CI destiné à la communication bidirectionnelle sur les lignes de transmission équilibrées à bus multipoint. Notons qu'il est conforme aux normes ANSI EIA/TIA-485-A et ISO 8482:1987(E). Il combine en un unique boîtier un isolateur à 3 canaux, un pilote de ligne différentielle à 3 états, ainsi qu'un récepteur à entrée différentielle. Son côté logique (numérique) s'alimente en 5 V ou 3 V, son côté bus uniquement en 5 V.

Sa fiche technique nous apprend également que les temps de montée et de descente sont limités de façon à réduire les réflexions sur les lignes refermées avec une mauvaise terminaison. Cette régulation limite la vitesse de transmission à 500 kbauds. Son impédance d'entrée est de 96 kΩ, ce qui autorise jusqu'à 256 (!) émetteurs-récepteurs sur le bus. L'état haut de la broche 5 active le pilote, l'état bas le désactive. Les sorties différentielles du pilote et les entrées différentielles du récepteur sont reliées pour former un port d'E/S différentiel. La désactivation du pilote, ou VDD1 = 0 V, ou encore VDD2 = 0 V, impose une charge minimale sur le bus. Enfin, la mise à l'état haut de la broche 4 désactive le récepteur.

Spécifications

- Isolement CA complet entre appareils DMX et ordinateur
- Transcepteur RS-485 isolé ADM2483 de la série iCoupler®
- Boîtier IP40 compact
- Prise en charge des 512 canaux

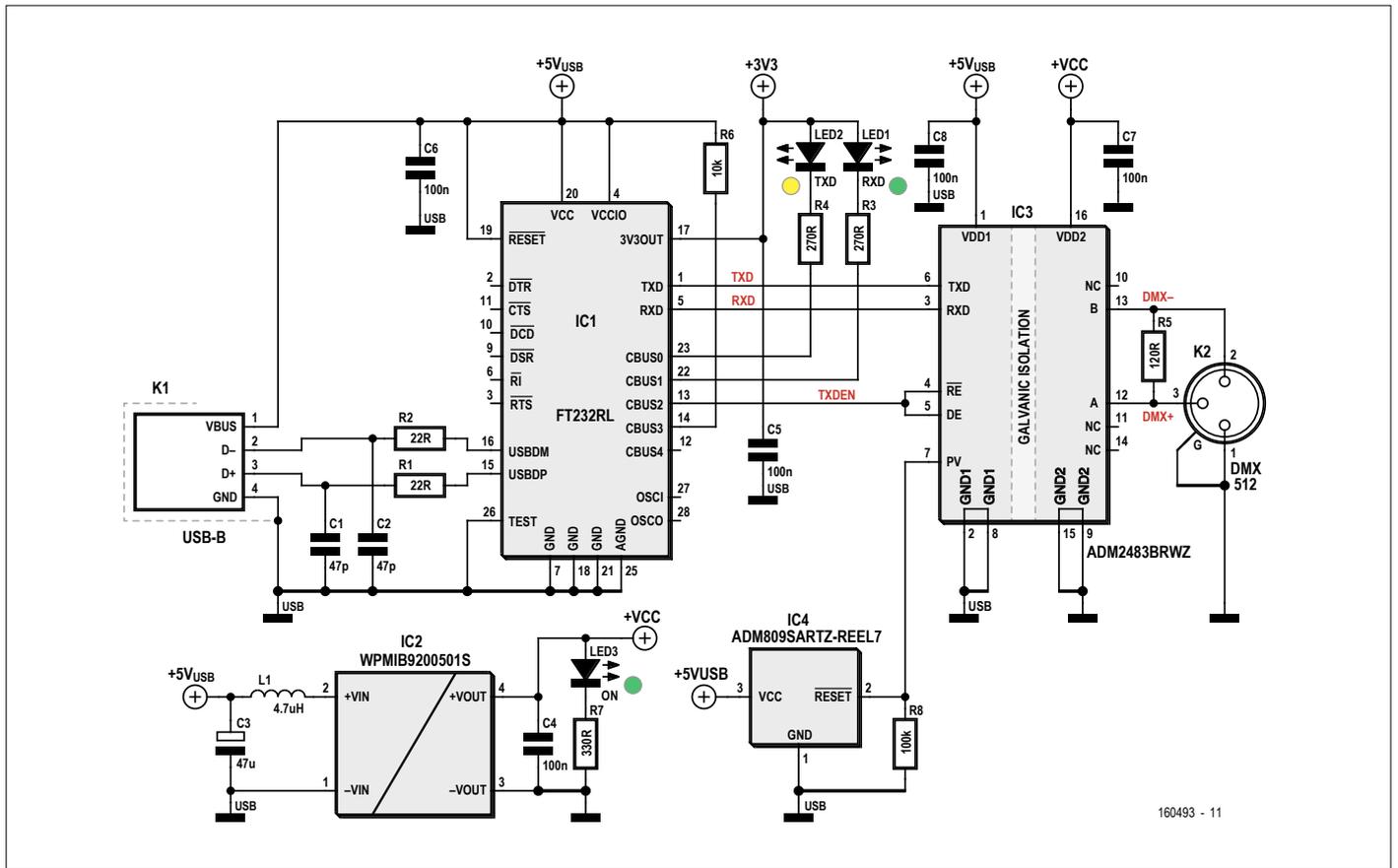


Figure 1. Le convertisseur comprend quatre circuits intégrés dont deux, le convertisseur CC/CC (IC2) et le transcepteur RS-485 (IC3), assurent l'isolement galvanique total entre les récepteurs DMX d'un côté, votre précieux ordinateur et votre non moins précieuse santé de l'autre.



LISTE DES COMPOSANTS

Résistances

Toutes 5 %, 0,1 W, 150 V
 R1, R2 = 22 Ω
 R3, R4 = 270 Ω
 R5 = 120 Ω
 R6 = 10 kΩ
 R7 = 330 Ω
 R8 = 100 kΩ

Condensateurs

C1, C2 = 47 pF, 50 V, C0G/NP0, 0805
 C3 = 47 μF, 25 V, ESR = 0,36 Ω, CMS, électrolytique à l'aluminium
 C4, C5, C6, C7, C8 = 100 nF, 50 V, X7R, 0805

Semi-conducteurs

LED1 = verte, 50 mcd, 2,1 V @20 mA
 LED2 = jaune, 50 mcd, 2 V @20 mA
 LED3 = verte, 3 mm
 IC1 = FT232RL
 IC2 = WPMIB9200501S
 IC3 = ADM2483BRWZ
 IC4 = ADM809SARTZ

Divers

K1 = connecteur USB-B

K2 = NC3FAAH2, prise XLR à 3 voies, contacts dorés, série AA
 Boîtier en ABS, 40×85×81 mm, IP40, p. ex. Bopla modèle 26085000

Vis à tête cylindrique Phillips M3×5 mm
 Rondelles M3
 Circuit imprimé, réf. e-choppe 160493-1

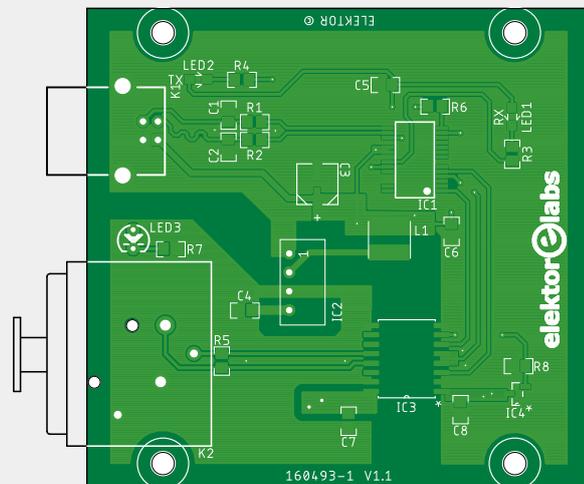


Figure 2. Le circuit imprimé a été conçu avec comme objectif l'isolement total des lignes DMX et USB, mais nous souhaitons également que la carte soit compacte.

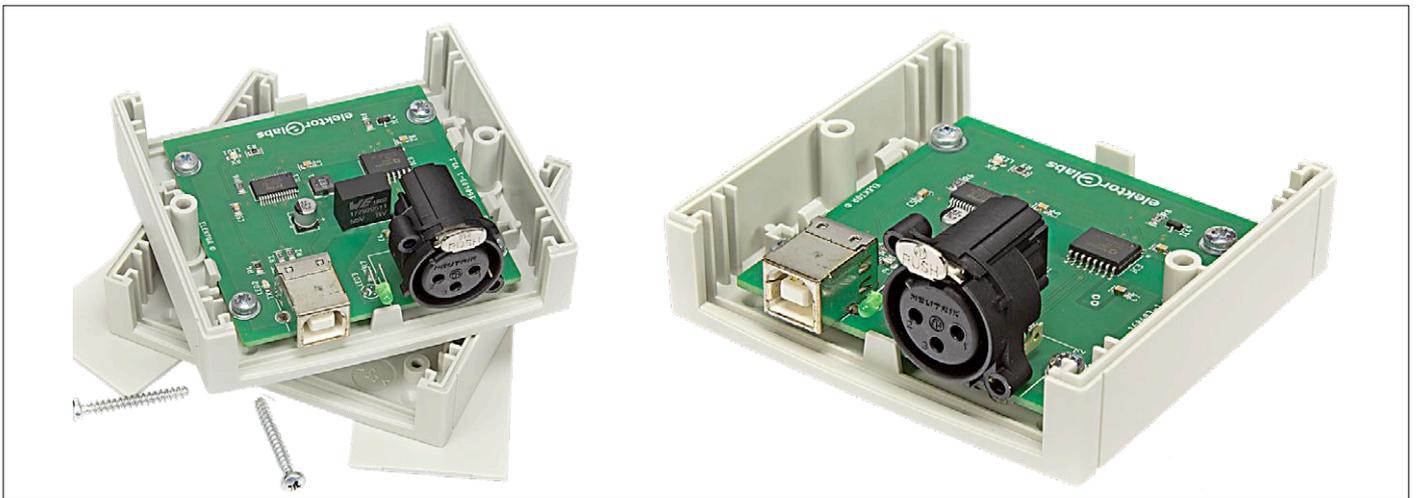


Figure 3. La carte épouse parfaitement les dimensions du modèle de boîtier choisi.

Le système DMX qui éblouira votre public se relie au connecteur XLR (K2). Votre PC se connecte au connecteur USB-B (K1). Notez que les masses USB et DMX d'IC3 sont maintenues séparées, autrement dit qu'elles possèdent leurs propres (double) broches. Relevons ici aussi la nature des masses d'IC3 : une notée USB, l'autre sans identification particulière (masse DMX).

Le reste du circuit comprend les indispensables condensateurs de découplage, une LED témoin de mise sous tension (LED3), et deux LED témoins de l'activité TXD/RXD (LED2/LED1), toutes trois avec leurs résistances-talons.

Circuit imprimé

Le convertisseur étant en grande partie composé de CMS, sa construction ne s'avérera sans doute pas des plus simples. Pour vous aider, nous mettons à votre disposition le tracé des pistes [1] ainsi que la sérigraphie de la carte (fig. 2) et la liste des composants. Vous pouvez bien sûr gagner du temps en vous procurant le circuit imprimé non assemblé.

Comme le montre la sérigraphie du circuit, nous avons accordé une attention particulière à tous les aspects de l'isolement galvanique. Les pistes « responsables » de cet isolement sont toutes séparées d'une distance qui permet d'atteindre ou de dépasser le degré de sécurité requis. Par ailleurs la carte a été spécialement dimensionnée pour être logée dans un boîtier Bopla 26085000 en ABS (fig. 3).

Sous le jeu des projecteurs

Nous avons testé le convertisseur sous Windows avec le logiciel *Freestyler DMX* (fig. 4). Nous l'avons également testé

▶ Ce projet surpasse bon nombre de « convertisseurs » d'entrée de gamme.

sous MacOS avec *Lightkey* et *QLC+*. Les 512 canaux peuvent réellement être tous utilisés, là encore un avantage par rapport aux convertisseurs bas de gamme. Nous avons effectué divers tests, notamment avec des scanners et projecteurs à LED RVB. Et avec succès : rien n'a grillé, rien n'a fumé ! ◀

(180356-03 – version française : Hervé Moreau)



@ WWW.ELEKTOR.FR

→ Circuit imprimé nu du convertisseur USB-DMX

www.elektor.fr/usb-dmx-160493-1

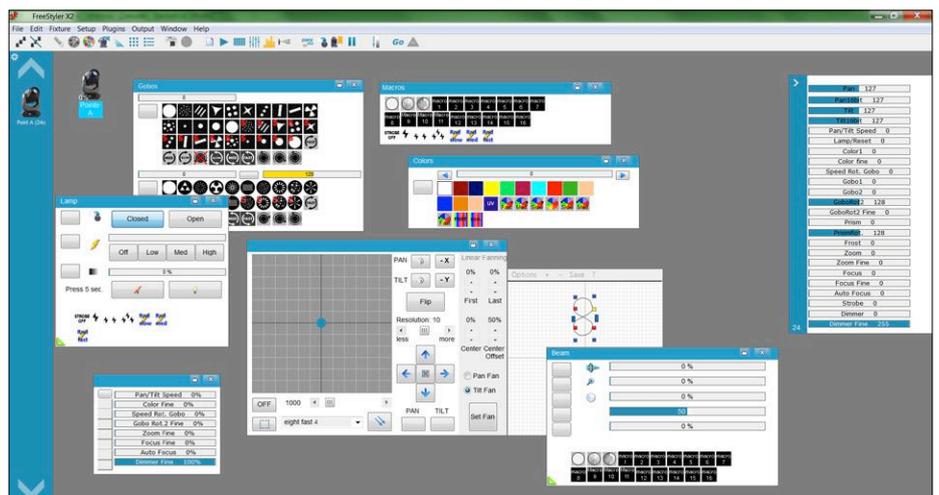


Figure 4. Freestyler DMX est un logiciel gratuit de commande d'éclairage utilisé par de nombreux DJ, VJ et éclairagistes. Freestyler et notre convertisseur font parfaitement la paire.

Lien

[1] Page de l'article : www.elektormagazine.fr/180356-03



la pratique des amplificateurs opérationnels

partie 1 : fondements

Burkhard Kainka

On trouve des amplificateurs opérationnels partout dans l'électronique. Ils sont utilisés pour toutes sortes de fonctions dans des circuits très divers. Il en existe d'innombrables types et familles présentant des avantages et inconvénients spécifiques. Pour y voir clair, il faut examiner cela de plus près.

Les amplificateurs opérationnels (ou ampli-op, pour le petit nom) sont des circuits intégrés à transistors bipolaires, JFET ou CMOS. Le nom de ce composant vient de son utilisation originelle dans les calculateurs. Grâce à ces amplificateurs, les calculateurs analogiques pouvaient en effet effectuer des opérations arithmétiques : additions, multiplications, etc.

Circuits fondamentaux à ampli-op

L'ampli-op amplifie la différence de potentiel présente sur ses deux entrées (gain en tension). L'ampli-op idéal a un gain en tension infini ; en pratique, il atteint 100 000 environ. Dans la plupart des cas, une contre-réaction détermine le comportement du circuit. La **figure 1** montre un ampli-op avec contre-réaction. Le gain en tension du montage est très exactement de 1, mais le gain en courant est considérable. L'entrée est à haute impédance tandis que la sortie peut être chargée par une faible impédance. Ce circuit est l'archétype de l'ampli tampon.

Suiveur de tension

On peut considérer cette fonction comme une boucle de régulation. L'ampli-op compare en permanence la consigne (sur l'entrée +, non-inverseuse) à la tension obtenue effectivement (sortie, entrée -). Toute différence minimale est automatiquement annulée (régulée). En pratique, on observe une petite

différence constante, appelée tension de décalage (ou offset) de l'ordre des mV. Sur certains types comme le LM741, il est possible de compenser (annuler) cet offset.

Autrefois, il fallait impérativement alimenter les ampli-op par un double rail de tension, souvent +15 V et -15 V. La plage de tension d'entrée et de sortie avoisinait généralement -10 V à +10 V. Cette caractéristique date des calculateurs analogiques, les ampli-op actuels se contentent de tensions nettement plus faibles.

Amplificateur non-inverseur

Un ampli-op peut également être utilisé pour amplifier un signal d'un facteur donné. À cet effet, la contre-réaction comporte un diviseur de tension (**fig. 2**). La tension de sortie s'établit automatiquement de sorte que la tension de l'entrée inverseuse (-) et celle de la non-inverseuse (+) soient égales. Tout écart minime se traduit par une forte modification de la tension de sortie qui est alors compensée par la contre-réaction. La différence des tensions d'entrée (leur décalage/offset) ne change pratiquement pas en raison du gain élevé.

Pour concevoir un circuit, on considère le cas d'un ampli-op idéal, sans tension de décalage. L'amplificateur opérationnel idéal a en outre une impédance d'entrée infinie, une impédance de sortie nulle et une bande passante infinie. Dans de

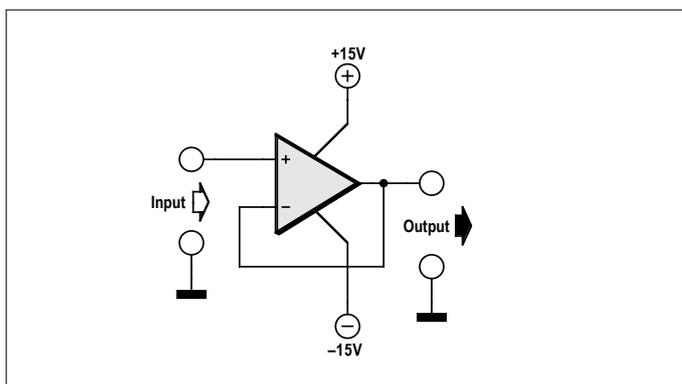


Fig. 1. Ampli-op : l'amplificateur tampon.

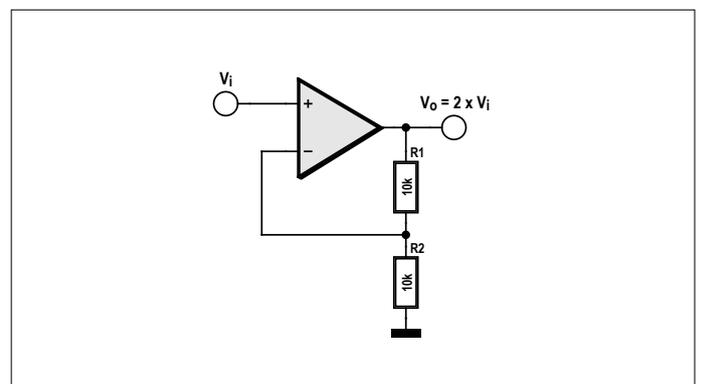


Fig. 2. Circuit amplificateur de tension de gain 2.

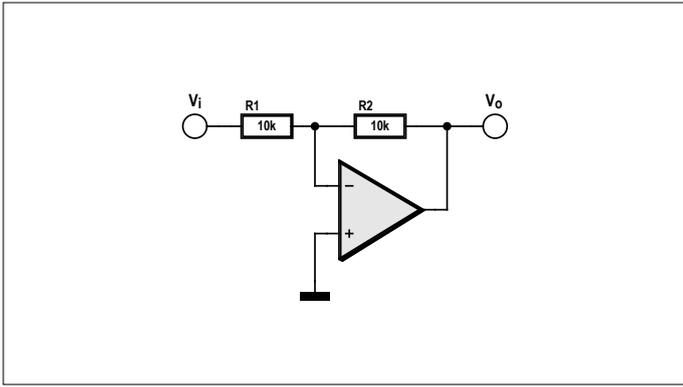


Fig. 3. Amplificateur inverseur.

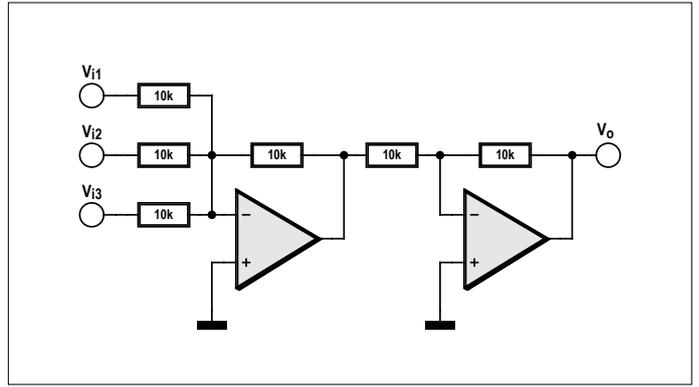


Fig. 4. Circuit additionneur de trois tensions.

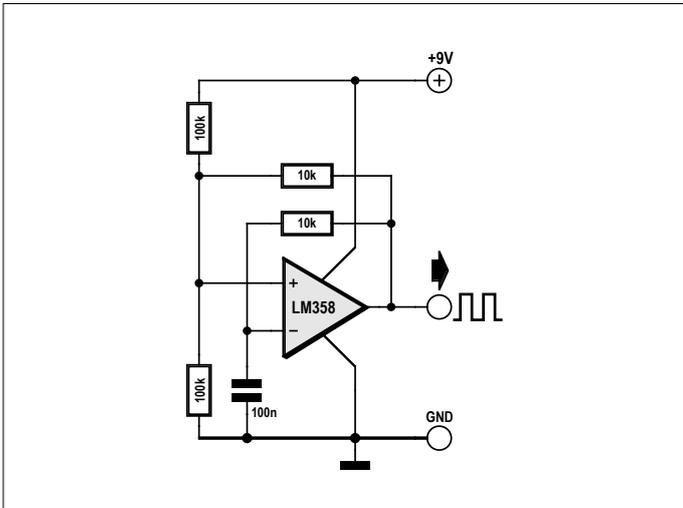


Fig. 5. Générateur de signaux carrés.

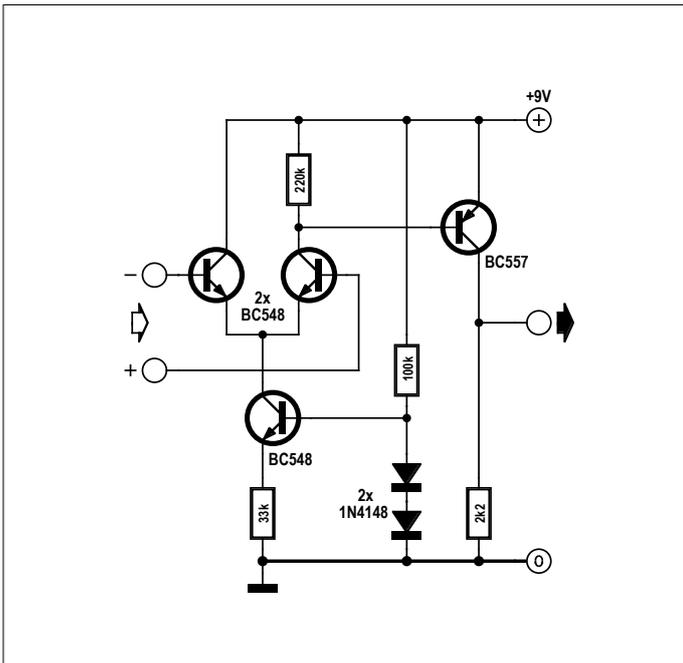


Fig. 6. Ampli-op à composants discrets.

nombreuses applications, l'ampli-op standard se comporte en pratique comme un ampli-op idéal.

Par principe, le choix de valeurs de résistance appropriées permet d'atteindre n'importe quelle valeur de gain. Le facteur d'amplification (gain) correspond au facteur de division du diviseur de la boucle de contre-réaction (facteur 2 pour la fig. 2). Avec un ampli-op, on peut atteindre l'amplification précise souhaitée comme on le faisait dans un calculateur analogique. Ce circuit permet donc une opération de calcul, dans notre cas, une multiplication par un facteur constant A :

$$A = (R1 + R2) / R2$$

Avec un gain très élevé, il faut prendre garde que le décalage de l'ampli-op est amplifié du même facteur. Sur certains types d'ampli-op (par ex. « le vénérable » LM741), il existe des entrées spécifiques pour compenser ce décalage. D'autres types comme l'OP07 sont ajustés en usine et spécifiés pour un décalage de quelques microvolts seulement.

Amplificateur inverseur

Un ampli-op peut être utilisé pour inverser strictement un signal d'entrée (fig. 3). Pour cela, l'entrée non-inverseuse de l'ampli-op est mise à la masse. La tension sur l'entrée inverseuse prend d'elle-même une valeur nulle. Si l'on utilise deux résistances égales, en appliquant une tension d'entrée de +1 V, on obtient une tension de sortie de -1 V, de sorte que la tension sur l'entrée inverseuse soit égale à zéro. En modifiant le rapport des résistances, on peut obtenir un facteur d'amplification A quelconque. On retiendra que :

$$A = -R2/R1$$

Amplificateur additionneur

Souvent, il faut faire la somme de différentes tensions d'entrée. Pour réaliser cela, il suffit de prévoir plusieurs résistances d'entrée (fig. 4). L'inversion peut être annulée à l'aide d'un second inverseur placé en aval.

Générateur

Dans les circuits purement amplificateurs, il n'y a qu'une contre-réaction, les circuits générateurs ont également besoin d'une rétroaction. La figure 5 montre un générateur de signaux carrés avec rétroaction sur l'entrée +. En même temps, la

contre-réaction est freinée par une cellule RC qui détermine la fréquence de sortie.

La vie intérieure d'un ampli-op

En interne, chaque ampli-op est constitué d'un amplificateur différentiel et d'un étage de sortie. Il est possible de concevoir un ampli maison équivalent à partir de transistors discrets (fig. 6). Il est utile de s'y intéresser si l'on souhaite mieux comprendre les caractéristiques, les avantages et les inconvénients des ampli-op.

Nous pouvons effectuer des mesures de tensions internes et externes dans une application réelle avec contre-réaction pour recueillir des données sur le comportement de l'ampli-op maison (fig. 7). La source de courant des deux transistors d'entrée fournit un courant de 14 µA qui se répartit presque également entre lesdits transistors. Le courant de repos d'entrée est de 60 nA (à comparer avec le 741 : 10 nA). La tension de décalage est de 5 mV (741 : 1 mV). En outre, nous avons mesuré l'amplification à vide. Pour cela nous avons appliqué une tension d'entrée de 2 Vc-à-c. Nous obtenons à la sortie également 2 Vc-à-c, mais en opposition de phase. Sur l'entrée inverseuse, nous avons pu mesurer 50 mVc-à-c. Nous pouvons en déduire une amplification à vide de 80. Ce n'est pas encore énorme, le simple 741 affiche déjà 100 000. Mais l'ampli-op maison est déjà meilleur sur un point : il délivre une tension de sortie de rail à rail qui atteint 9 Vc-à-c.

La plupart des fabricants publient dans leurs fiches techniques un schéma simplifié du circuit interne qui donne accès à des informations importantes pour l'utilisation de l'amplificateur. La figure 8 donne le schéma interne de l'ampli-op standard LM741. On y reconnaît des similitudes avec l'ampli-op maison aux entrées à transistors NPN.

Le schéma interne montre un amplificateur d'entrée différentiel à transistors NPN. Le courant d'émetteur est ajustable de l'extérieur pour améliorer la symétrie et ainsi annuler la tension de décalage. Un miroir de courant transfère le signal sur un étage intermédiaire d'amplification qui pilote un étage de sortie inverseur de phase. Un unique condensateur permet de réduire à 1 MHz le produit amplification interne × bande passante. En conséquence, si l'amplification à vide jusqu'à environ 10 Hz avoisine les 100 000, à 1 MHz elle n'est plus que de 1. La figure 9 montre la décroissance de l'amplification en échelle logarithmique. Cette réduction de l'amplification à haute fréquence est indispensable pour obtenir une stabilité suffisante en toutes circonstances. Cependant elle révèle également qu'un ampli-op standard ne peut pas être utilisé à des fréquences très élevées.

Dans la deuxième partie de cette mini-série, nous ferons connaissance avec un autre type d'amplificateur opérationnel et ses applications particulières (ampli-op à étages d'entrée à transistors à effet de champ). ◀

(180036-04 - version française : Yves Georges)

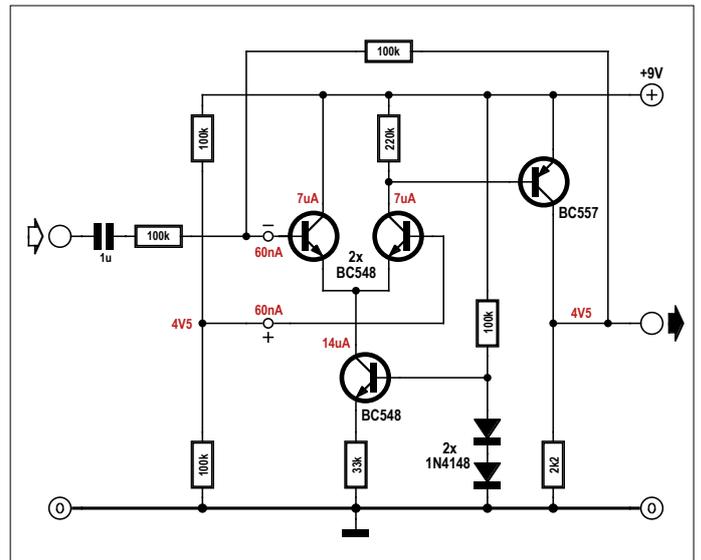


Fig. 7. Test d'un amplificateur inverseur.

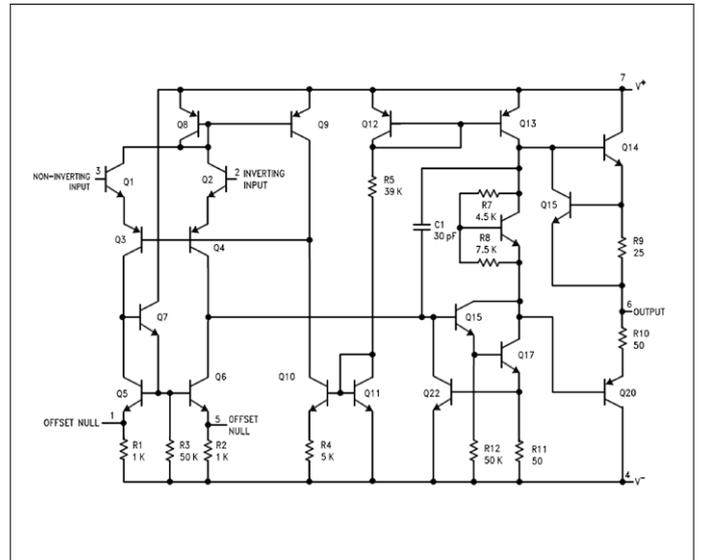


Fig. 8. Schéma interne du LM741 (source : Texas Instruments).

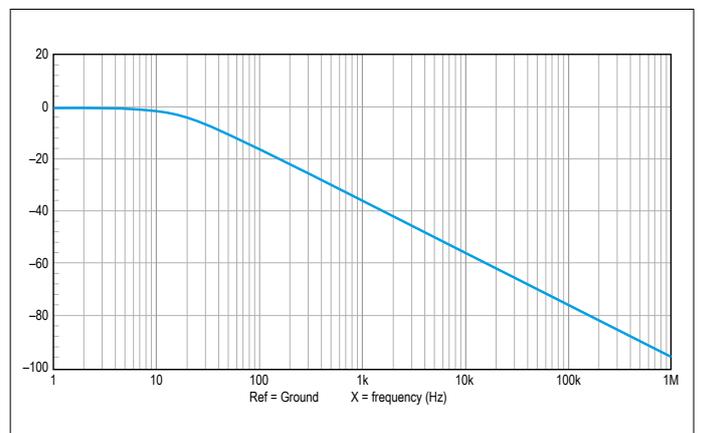


Fig. 9. Courbe de réponse de l'amplification à vide.

lampe magique

polychromie sans fil

Roel Arits (Pays-Bas)

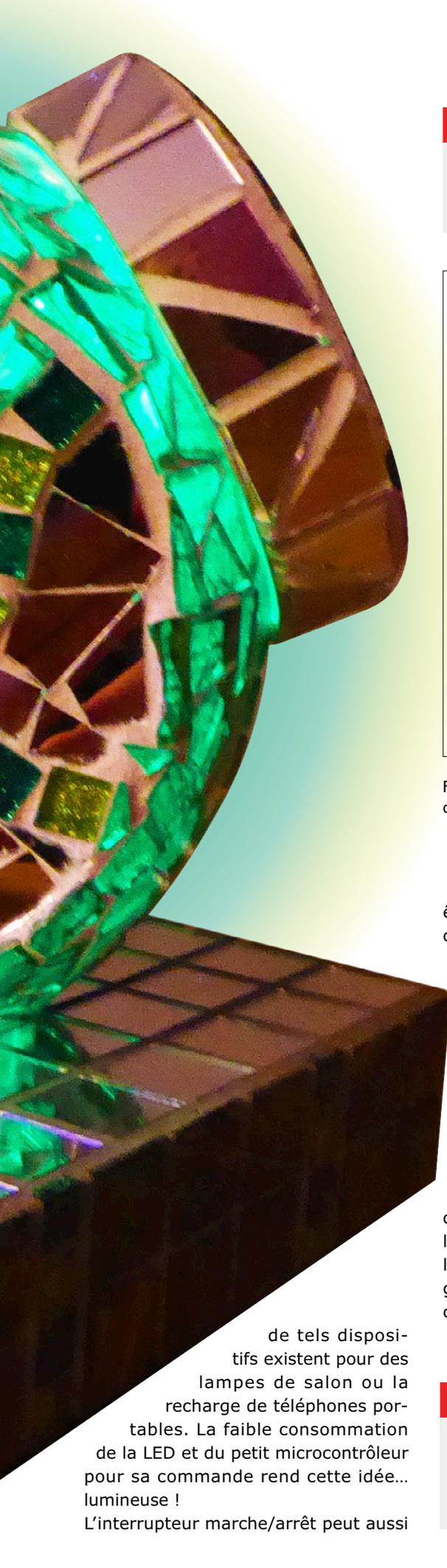
Quoi de plus passionnant que de s'associer à un autre amateur pour travailler sur un projet commun ? Unir les passe-temps de deux conjoints peut-être... Comme la passion de l'auteur pour les circuits d'éclairage à LED et celle de son épouse pour les réalisations avec des tesselles de verre. Le résultat : une très belle sculpture de verre lumineuse.

La combinaison de LED RVB et de tesselles en verre transparent donne des effets lumineux intéressants. Cependant travailler avec du verre est un véritable défi pour un électronicien : le forage des trous pour les câbles d'alimentation et l'interrupteur marche/arrêt n'est pas du tout évident. On peut utiliser des piles, mais il faut les remplacer régulièrement,

ce qui n'est pas une bonne idée non plus (surtout pour la nature...).

Il faut donc une autre approche, vite trouvée avec un peu d'imagination. Il est possible d'alimenter sans fil la LED RVB et l'électronique connexe qui se trouvent à l'intérieur de la lampe ;





Caractéristiques

- Variations polychromes : un générateur pseudo-aléatoire commande une LED RVB
- Marche/arrêt par interrupteur à capteur capacitif
- Alimentation sans fil à l'aide d'un oscillateur de puissance

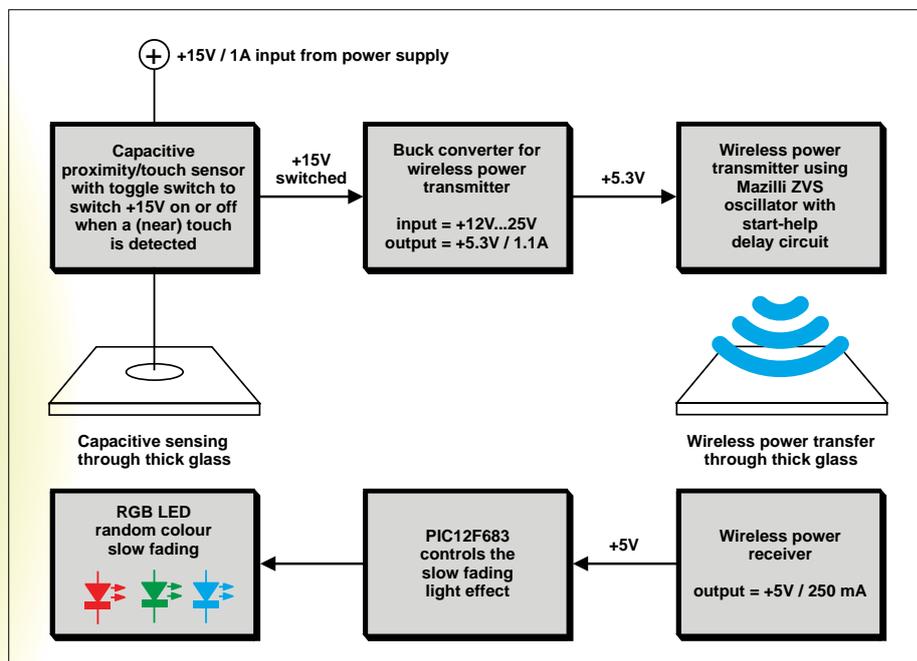


Figure 1. Le schéma de principe comporte nombre de blocs, mais l'électronique est moins compliquée qu'il n'y paraît.

être remplacé par un circuit à capteur capacitif, dont la consommation au repos est négligeable. La distance entre les électrodes – le capteur lui-même et un doigt – est tout au plus de quelques centimètres, ça ne devrait donc pas poser de problème.

Schéma de principe

Nous avons tous les éléments pour établir un schéma de principe (**figure 1**) : un interrupteur à bascule marche/arrêt basé sur un capteur capacitif qui commande l'alimentation de l'émetteur qui lui transmet l'énergie à la lampe ; un récepteur pour cette énergie ; une LED RVB avec son circuit de commande.

L'alimentation de 15 V, avec un courant maximal de 1 A, est fournie par un adaptateur secteur standard ; le seul trou à percer dans la base de la lampe est celui pour le connecteur de l'alimentation. Un dispositif à capteur capacitif commande cette alimentation via un MOSFET à faible résistance à l'état passant $R_{DS(on)}$. Un régulateur à découpage Buck abaisse la tension à 5,3 V pour alimenter l'émetteur. L'émetteur de puissance repose sur un oscillateur de puissance Mazilli à commutation au passage par zéro de la tension pour la commande d'un transformateur *flyback* (voir également *Mazilli ZVS Fly-back Driver*). Nous y avons ajouté un retard à l'allumage, parfois nécessaire. Le récepteur est un simple circuit accordé

de tels dispositifs existent pour des lampes de salon ou la recharge de téléphones portables. La faible consommation de la LED et du petit microcontrôleur pour sa commande rend cette idée... lumineuse !
L'interrupteur marche/arrêt peut aussi

Spécifications

- Courant maximal d'environ 150 mA sous 15 V (oscillateur et LED en service)
- Courant de ± 94 mA pour l'émetteur
- Courant de ± 60 mA pour la LED et le PIC
- Courant de ± 5 μ A au repos (oscillateur et LED éteints)

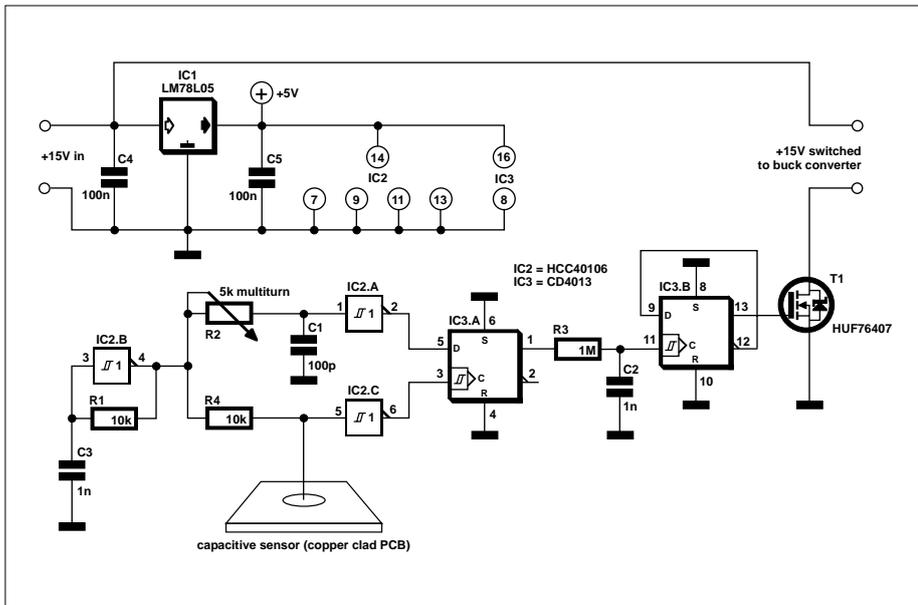


Figure 2. L'interrupteur à capteur capacitif, basé sur la « mesure » d'une différence de capacité.

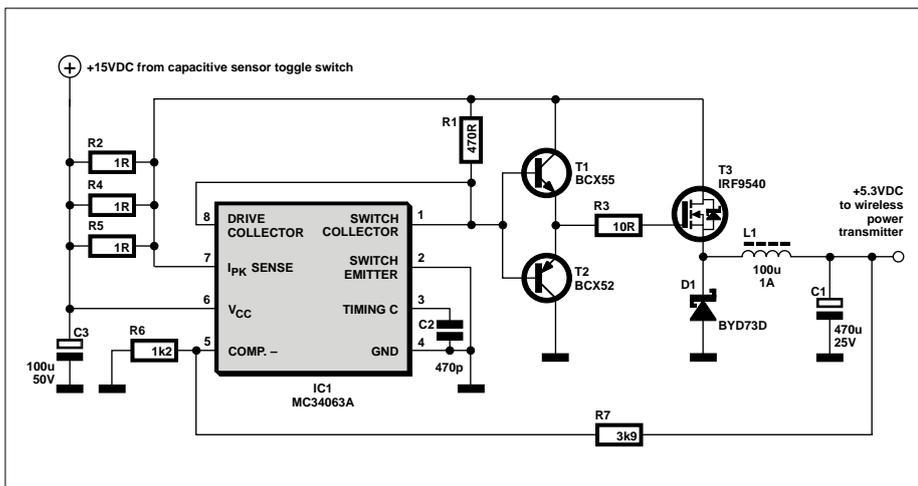


Figure 3. Une application standard du MC34063 : 15 V à l'entrée, une tension stable de 5,3 V à la sortie.

LC avec un redresseur, suivi d'un régulateur linéaire à très faible tension de déchet. Ce dernier fournit la tension de 5 V nécessaire au microcontrôleur PIC et à la LED RVB (ou à trois LED séparées).

Interrupteur à capteur capacitif

Nous allons maintenant examiner les divers blocs en détail, en commençant par l'interrupteur à capteur capacitif. Deux exigences : il doit être suffisamment sensible, mais aussi stable. Par exemple, il doit être insensible aux changements de température.

Le cœur du circuit (**figure 2**) est un

oscillateur à relaxation, pour lequel on utilise l'un des six inverseurs à bascule de Schmitt (IC2.B) d'un classique circuit CMOS, un 40106. Inutile de s'attarder sur le fonctionnement de ce circuit élémentaire ; avec les valeurs choisies pour R1 et C3, la fréquence du signal rectangulaire est d'environ 100 kHz et le rapport cyclique de 50%.

Le signal de sortie d'IC2.B attaque deux circuits intégrateurs RC. L'un est constitué de R2 et C1, l'autre de R4 et d'un condensateur variable. Ce dernier n'est autre que notre capteur capacitif, dont l'électrode est constituée d'un simple morceau de circuit imprimé.

La capacité du capteur varie en fonction de son environnement, et par conséquent la constante de temps du circuit RC ; la capacité augmente si on approche la main du capteur, donc la constante aussi. Ceci entraîne un retard du signal – proportionnel à l'augmentation de la capacité – dans cette partie du circuit.

On pourrait tout simplement comparer les deux signaux, tamponnés par des inverseurs du 40106, avec une fonction OU exclusif (XOR), mais ce n'est pas une très bonne idée (souvenez-vous de vos cours d'électronique numérique !).

On utilise donc une bascule D (IC3.A) qui recopie la valeur à l'entrée (D) sur sa sortie à chaque front d'horloge. Pour notre circuit, la sortie passera à l'état haut si le signal d'horloge est retardé par rapport au signal d'entrée, sinon elle restera à l'état bas.

Le signal de sortie de la bascule est à nouveau intégré par le circuit R3C2, dont la constante de temps est beaucoup plus grande que la période des signaux.

La sortie de la bascule IC3.B change d'état lorsque la tension aux bornes de C2 est supérieure au seuil de l'entrée d'horloge CLK, donc si la constante de temps formée par R4 et le capteur capacitif augmente, CQFD !

La sortie Q d'IC3.B attaque le MOSFET T1, qui commute l'alimentation du circuit convertisseur Buck.

Convertisseur Buck

Afin d'abaisser la tension de 15 V à 5,3 V, suffisante pour le PIC et la LED (ainsi que pour l'émetteur), nous utilisons une application standard du régulateur à découpage MC34063, dont la fiche de caractéristiques est en [2].

Le schéma de cette partie du circuit est en **figure 3** ; la différence avec l'application standard d'ON Semiconductor se trouve dans le circuit de sortie (T1, T2 et FET1). Le MC34063 devient en effet assez chaud lorsqu'il doit délivrer 1 A en continu ; T1 et T2 ont donc été ajoutés pour la commande du MOSFET de sortie. Ceci permet d'éviter un échauffement du circuit, inévitable si l'alimentation est dans un boîtier fermé et fonctionne en continu : en électronique de puissance, mieux vaut trop froid que trop chaud.

Alimentation sans fil, l'émetteur

La transmission d'énergie sans fil est relativement facile à comprendre : d'un côté nous avons un émetteur dont l'an-

tenne est une bobine, et de l'autre un récepteur, aussi avec une bobine. Si ces deux bobines sont couplées, nous avons un transformateur, qu'il y ait un noyau en ferrite ou non. Si la distance entre les bobines est faible, quelques cm tout au plus, et que la fréquence est relativement élevée, l'absence de noyau n'est pas gênante ; le fonctionnement est encore amélioré avec des circuits résonnants aux primaire et secondaire.

Nous avons juste besoin d'un oscillateur qui permette une transmission d'énergie entre bobine émettrice et bobine réceptrice. Nous avons opté pour un oscillateur de puissance Mazilli à commutation au zéro de tension, qui n'exige pas de gros radiateurs pour les transistors de puissance. Le schéma complet de l'émetteur est en **figure 4**.

En **figure 5** nous avons redessiné l'oscillateur seul pour mieux en expliquer le fonctionnement. La partie la plus importante est le circuit LC constitué du condensateur et de la bobine à prise médiane, ce qui permet l'obtention d'un signal sinusoïdal.

Considérons que c'est le transistor T1 qui conduit en premier à la mise en route (les caractéristiques des deux transistors ne sont jamais tout à fait identiques, et l'un des deux sera plus rapide que l'autre). Le courant dans la moitié correspondante de la bobine sera plus élevé, et la tension sur la grille de T2 plus faible ; celui-ci va se bloquer.

La tension au point A sera proche de 0 V, tandis que celle au point B atteindra un maximum, puis redescendra vers 0 V après une demi-période du cycle du circuit LC. Lorsque la tension en B s'approche de 0 V, c'est T1 qui se bloque, et la tension au point A grimpe. T2 devient conducteur, la tension en B reste nulle et T1 bloqué. Après une nouvelle demi-période, la tension en A redescend et le cycle recommence. L'oscillateur fait ce pour quoi il a été conçu et continue d'osciller. L1 évite les appels de courant trop forts.

L'intérêt de ce montage est la commutation des MOSFET au passage par (presque) zéro de la tension. Les pertes sont ainsi minimisées, et même pour des puissances relativement élevées la taille des radiateurs peut être réduite. Dans notre circuit, il n'en faut même pas !

Lors des expérimentations avec ce circuit très simple, l'auteur a constaté que le démarrage de l'oscillateur n'était pas toujours garanti ; un des transistors conduit

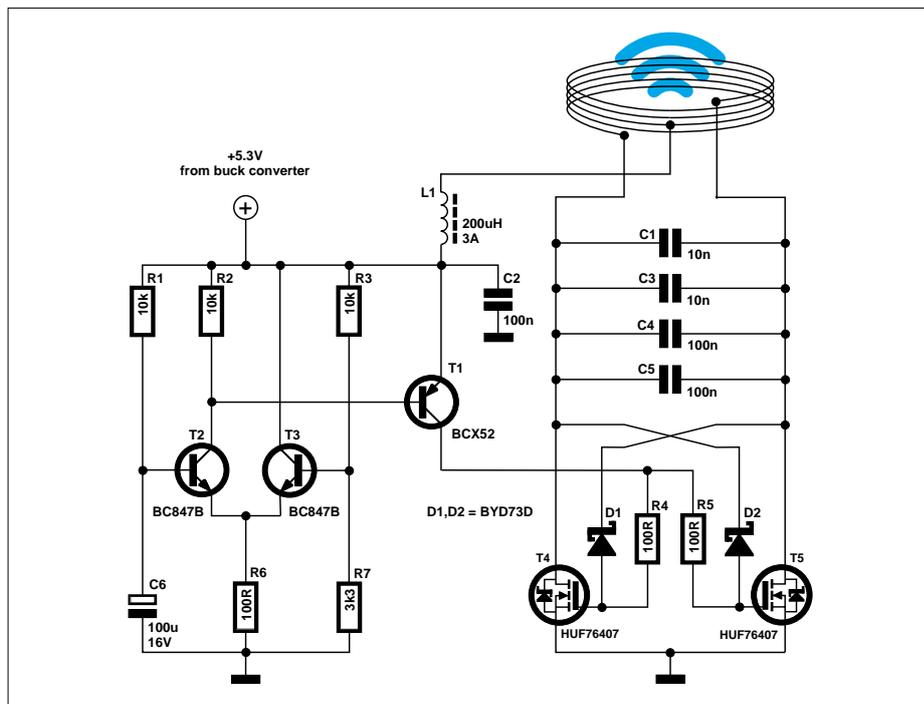


Figure 4. L'émetteur est un oscillateur de puissance Mazilli à commutation au passage par zéro de la tension, avec un circuit de retard à l'allumage.

et l'autre reste bloqué. Il en a résulté une odeur « chaude » que les amateurs d'électronique connaissent bien : un semi-conducteur vient de rendre l'âme... La cause du problème semble venir de l'adaptateur secteur, dont la tension met un certain temps à s'établir.

Deux solutions pour y remédier : trouver un adaptateur secteur plus rapide – ce n'est pas très satisfaisant – ou modifier le circuit pour qu'il s'accommode d'une alimentation un peu lente à la détente. L'auteur retarde la conduction des MOSFET en appliquant la tension sur les résistances de grille environ 250 ms plus tard que sur les drains.

Le circuit de retard est bâti autour des transistors T1 à T3. Le couple T2-T3 forme un amplificateur différentiel. À la mise sous tension, C6 se charge lentement via R1. Lorsque la tension à ses bornes atteint la tension de seuil fixée par le diviseur R3/R7, T2 conduit. La tension sur la base de T1 baisse et le transistor conduit (il s'agit d'un PNP) ; la tension des grilles des MOSFET, alimentées via R4 et R5, grimpe et ils conduisent à leur tour.

La bobine comporte 14 spires serrées d'un diamètre de 57 mm et constituées d'un fil de cuivre émaillé de 1 mm de diamètre ; la hauteur de la bobine est de 16 mm et son inductance d'environ 16 µH. En combinaison avec les

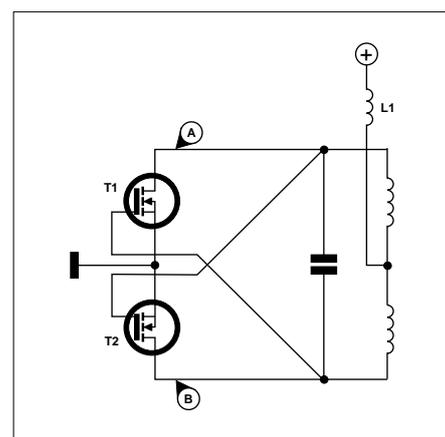


Figure 5. Schéma simplifié de l'oscillateur, à des fins didactiques.

condensateurs C4 et C5 (200 nF) – et les autres éléments du circuit ! – la fréquence de résonance est de 81 kHz. Les condensateurs C1 et C3 permettent d'ajuster cette fréquence de résonance, bien que cela n'ait pas grande importance ; ce qui compte c'est que les fréquences de l'émetteur et du récepteur soient les mêmes.

Alimentation sans fil, le récepteur

Le schéma du récepteur est en **figure 6**. Nous y voyons un circuit oscillant (bobine

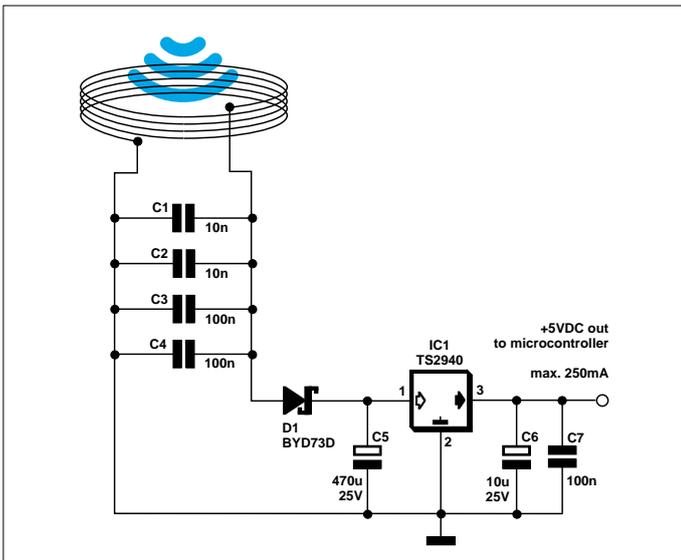


Figure 6. Le récepteur : il n'y a pas plus simple !

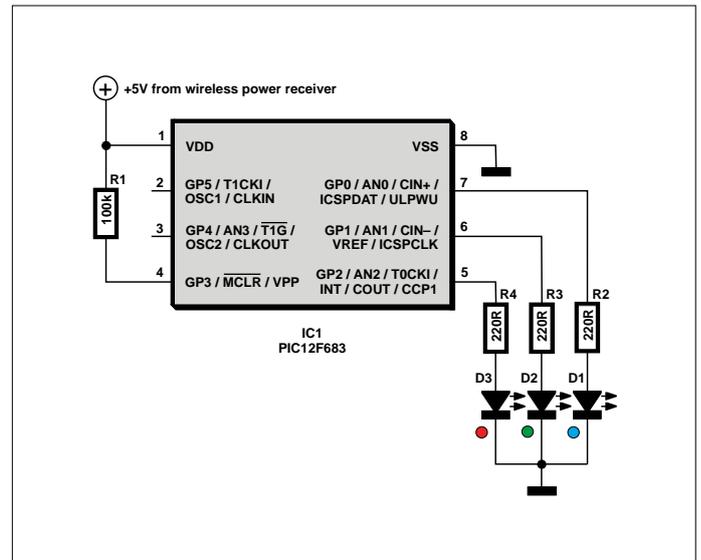


Figure 7. Et la lumière fut : une LED RVB commandée par un PIC.

et condensateurs C1 à C4), identique à celui de l'émetteur, mais la bobine n'a pas de prise médiane. Les condensateurs C1 et C2, dont la valeur peut être modifiée, permettent d'ajuster la fréquence de résonance (81 kHz) si néces-

saire. La diode D1 redresse le signal reçu par le circuit accordé. Ce circuit n'est rien d'autre qu'un récepteur à cristal, utilisé depuis le début du XX^e siècle [3]. Un circuit accordé, dont le condensateur est variable, une diode

et un écouteur piézoélectrique, c'est tout ce qu'il fallait pour écouter les stations de radiodiffusion proches émettant dans la bande des Ondes Moyennes (525 à 1 605 kHz) en modulation d'amplitude. Le redresseur est suivi d'un condensateur intégrateur (C5) et d'un régulateur linéaire à très faible tension de déchet IC1 (0,6 V pour un courant de 1 A), pour l'alimentation stable du microcontrôleur et de la LED.

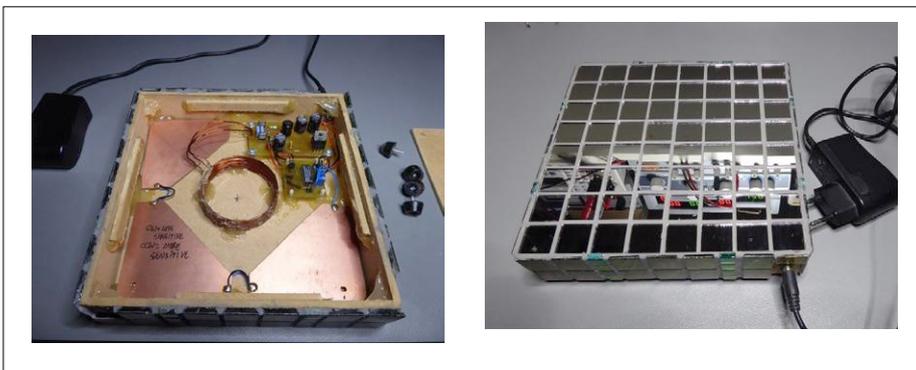


Figure 8. Le capteur capacitif est constitué de trois morceaux de circuit imprimé, de forme triangulaire.

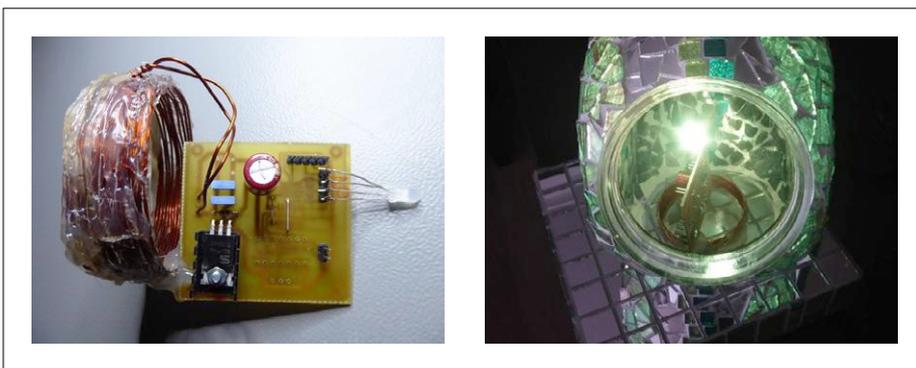


Figure 9. La bobine du récepteur et le circuit imprimé de l'électronique, sans fixations.

Et pour finir, le contrôleur

La description de la source lumineuse, c'est-à-dire le microcontrôleur et la LED, peut être réduite à sa plus simple expression, tout comme le circuit d'ailleurs... La **figure 7** montre un PIC12F683 à 8 broches, alimenté par les 5 V du récepteur, et trois LED (séparées ou assemblées en une LED RVB) avec leur résistance en série. Le démarrage du microgiciel est automatique à la mise sous tension. Il s'agit d'un générateur pseudo-aléatoire de signaux MLI très lents ; la luminosité des LED va augmenter et diminuer de manière imprévisible. Le logiciel tient aussi compte de la caractéristique logarithmique de notre vue (voir à ce sujet l'article sur les variateurs analogiques pour LED dans le numéro de septembre/octobre 2018), afin que nous percevions des variations linéaires de la luminosité. Une table de correspondance est utilisée pour ce faire. Le programme est disponible sur la page de l'article [4].

Montage et réglages

La **figure 8** montre la base de la lampe, ouverte et fermée. On y voit deux cartes (celle de l'interrupteur à capteur capacitif et celle de l'oscillateur de puissance) ainsi que la bobine de l'émetteur. Le capteur est facile à reconnaître : il est constitué de trois morceaux de forme triangulaire de circuit imprimé connectés entre eux. Le seul trou est pour le connecteur de l'adaptateur secteur.

Le capteur est très sensible et réagit à la moindre variation de capacité. Le raccordement de la masse d'une sonde d'oscilloscope au circuit avec une pince crocodile suffit à modifier la capacité et à déclencher l'interrupteur. Il en va de même si vous connectez une alimentation de laboratoire. Il est donc important de régler la valeur du potentiomètre R2 en situation réelle, sans aucun instrument de mesure connecté.

On peut raccorder une LED à la sortie d'IC3.A pendant les essais, cela permet de visualiser le résultat des réglages.

Si par la suite vous déplacez la lampe, par ex. près d'un mur ou sur un meuble d'un autre matériau, il est pratiquement certain que vous devrez procéder à un nouvel ajustage de R2. Peut-être même aussi lorsque vous connecterez l'adaptateur secteur...

La **figure 9** montre le circuit imprimé du récepteur et du contrôleur, démonté et placé dans la lampe. Le PIC est câblé du côté soudures. Il n'y a pas de fixations, le circuit est « porté » par la bobine. Le

microcontrôleur peut être programmé en circuit via les broches ICSPCLK et ICSPDAT ; nous vous renvoyons pour ce faire à la fiche de caractéristiques [5].

Les condensateurs de 10 nF des émetteur et récepteur permettent de syntoniser les deux circuits accordés. L'auteur est parvenu à « extraire » un courant de 250 mA du récepteur avant que la tension commence à s'effondrer.

Les deux bobines se trouvaient l'une au-dessus de l'autre à quelques cm de distance. Un ajustage plus précis des fréquences de résonance permettrait sans doute un plus gros transfert d'énergie.

En **figure 10**, la lampe terminée sur l'établi de l'auteur. Il est à noter que ce projet ne date pas d'hier : cette « lampe magique » a déjà quelques années de bons et loyaux services à son actif !

Au début, il semblait que la lampe s'allumait parfois spontanément pendant la nuit ; avant de tout démonter, l'auteur s'est aperçu que c'est son chat qui – par curiosité – déclenchait le circuit. C'est dire si le capteur capacitif est sensible... ◀

(170463-04 -

version française : Jean-Louis Mehren)



Figure 10. Le résultat final.



@WWW.ELEKTOR.FR

→ Programmeur universel TL866A

www.elektor.fr/tl866a-universal-programmer

Liens

[1] Oscillateur à relaxation : https://fr.wikipedia.org/wiki/Oscillation_de_relaxation

[2] Feuille de caractéristiques du régulateur MC34063 : www.onsemi.com/pub/Collateral/MC34063A-D.pdf

[3] Récepteur à cristal : https://fr.wikipedia.org/wiki/R%C3%A9cepteur_%C3%A0_cristal

[4] Page de l'article : www.elektormagazine.fr/170463-04

[5] Feuille de caractéristiques du PIC12F683 : ww1.microchip.com/downloads/en/DeviceDoc/41211D_.pdf

cartes de développement Arrow

ouvertes et... offertes aux bonnes idées

Clemens Valens (labo d'Elektor)

La réussite commerciale d'un nouveau produit dépend d'au moins deux critères : être le premier de sa catégorie sur le marché, et reposer sur des technologies dernier cri. Comment y parvenir ? Arrow, fournisseur mondial de composants électroniques, propose aux concepteurs des outils qui facilitent la création de nouveaux produits et accélèrent leur mise sur le marché.

Pour vous aider à mettre au point rapidement des prototypes fonctionnels, Arrow a créé une vaste famille de cartes de développement couvrant toutes sortes d'applications. Arrow peut également mettre en avant votre projet sur la plateforme de financement participatif Indiegogo. Cet article ne présente que quelques-unes des cartes Arrow les plus populaires. Gardez à

l'esprit qu'il en existe beaucoup d'autres – cherchez « Arrow Development Tools » sur le site du fabricant pour en découvrir toute la palette. Notez que vous pouvez obtenir ces cartes gratuitement et bénéficier d'autres soutiens si vous soumettez un projet viable à Arrow. Pour plus d'informations, consultez www.elektormagazine.fr/arrow-dev-boards

ARIS Edge S1 et Aris Edge S3

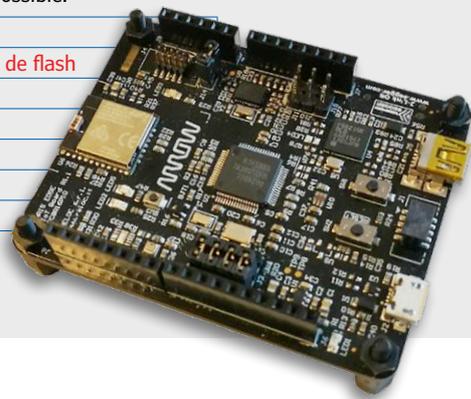
Les cartes *ARIS Edge S1* et *ARIS Edge S3* ont été conçues pour servir de nœuds périphériques dans un réseau de l'Internet des Objets (IdO). Elles reposent sur un contrôleur Synergy de Renesas (S1 pour Edge S1, S3A3 pour Edge S3). Les nombreuses fonctions dont elles sont dotées les rendent idéales pour les applications à capteurs intelligents et l'IdO. Elles peuvent communiquer avec d'autres dispositifs ou avec un nuage via un module radio multiprotocole prenant en charge *Bluetooth Low Energy* (BLE), Zigbee et Thread. Elles disposent de capteurs de température, d'humidité relative, de pression, de luminosité, enfin de mouvement grâce à une centrale inertielle à 9 degrés de liberté (DDL). La fusion de données est possible.

Brain Edge S1	Synergy S1 ARM Cortex-M0+ de Renesas, 32 MHz, 16 Ko de RAM & 128 Ko de flash
Brain Edge S3	Synergy S3A3 ARM Cortex-M4 de Renesas avec FPU, 48 MHz, 96 Ko de RAM & 512 Ko de flash
Connectivité	USB, radio multiprotocole (BLE, Thread et Zigbee)
Capteurs	Humidité, température, centrale inertielle à 9 DDL
Extension	Connecteurs pour shields Arduino, connecteur I ² C, écran tactile résistif LCD
Programmation	JTAG, JLink (+ JTAG pour module radio)
Alimentation	USB, pile bouton

Plus d'informations

ARIS Edge S1 : www.arrow.com/en/products/aris-edge/arrow-development-tools

ARIS Edge S3 : www.arrow.com/en/products/aris-edge-s3/arrow-development-tools



ARIS IoT Board (BLE) et ARIS Gateway (mesh)

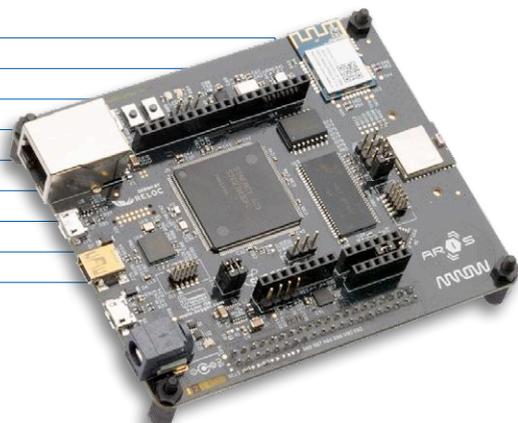
Les cartes *ARIS IoT Board* et *ARIS Gateway* sont des plateformes pour l'Internet des Objets prêtes à l'emploi. Elles reposent sur le système de développement Synergy de Renesas et sur un microcontrôleur Synergy S7 de Renesas avec cœur ARM Cortex M4 cadencé à 240 MHz. Elles sont adaptées aux applications de l'IdO telles que les hubs, les passerelles et les maillages sans fil, et peuvent aussi servir de nœuds périphériques. Elles communiquent avec le nuage et d'autres dispositifs par Wi-Fi ou Ethernet 10/100. La carte *ARIS IoT* dispose d'un module *Bluetooth Low Energy* (BLE 4.1/4.2), la carte *ARIS Gateway* d'un module pour maillage sans fil. Les deux ont une balise NFC, un chargeur de démarrage crypté et une fonction de mise à jour sans fil du micrologiciel. Elles disposent d'un gyroscope à 3 axes et de capteurs de température et d'humidité.

Processeur	Synergy S7 Cortex-M4 de Renesas avec FPU, RAM de 640 Ko & flash de 4 Mo
Mémoire embarquée	SDRAM de 32 Mo, mémoire flash QSPI de 64 Mo, micro-SD
Connectivité	USB, Ethernet 10/100, Wi-Fi b/g/n, balise NFC Forum Type 2
Carte IoT	Bluetooth 4.1/4.2, Bluetooth 5 advertising
Carte Gateway	Radio 2,4 GHz IEEE 802.15.4, Zigbee et Thread
Capteurs	Humidité, température, gyroscope à 2 axes, accéléromètre à 3 axes
Extension	Connecteurs pour shields Arduino, connecteur I ² C, écran tactile résistif LCD
Alimentation	Embase jack, mini & micro USB

Plus d'informations

ARIS IoT Board : www.arrow.com/en/products/aris/arrow-development-tools

ARIS IoT Gateway : www.arrow.com/en/products/aris-gateway/arrow-development-tools



Lion (LoRa) et Fox (Sigfox)

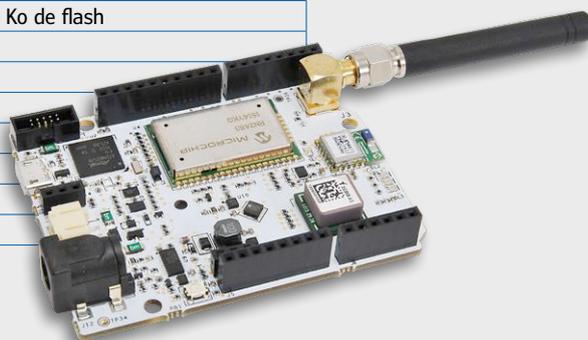
Les cartes *EveryLion* (LoRa) et *Fox* (SigFox) reposent sur des microcontrôleurs à très basse consommation ARM Cortex-M0+ SAMD21 à 32 bits de Microchip. *Lion* possède un module Lora, *Fox* un module SigFox. Toutes deux sont équipées de modules GPS et *Bluetooth Low energy* (BLE) et sont protégées par une puce d'authentification cryptée. Les deux ont la taille d'une carte Arduino Uno et peuvent être programmées et déboguées depuis l'EDI Arduino, ou via Atmel Studio grâce à la présence d'une interface *Serial Wire Debug* (SWD).

Processeur	Microchip SAMD21 ARM Cortex-M0+, 48 MHz, 32 Ko de RAM & 256 Ko de flash
Mémoire embarquée	32 Ko d'EEPROM
Connectivité	USB, Bluetooth BLE
Carte Lion	LoRa
Carte Fox	Sigfox
Capteurs	GPS
Extension	Connecteurs pour shield Arduino Uno
Alimentation	Embase jack, USB, pile Li-Po (>= 700 mAh)

Plus d'informations

MC27561-Lion : www.arrow.com/en/products/mc27561-lion/arrow-development-tools

MCS7561-Fox : www.arrow.com/en/products/mcs7561-fox/arrow-development-tools



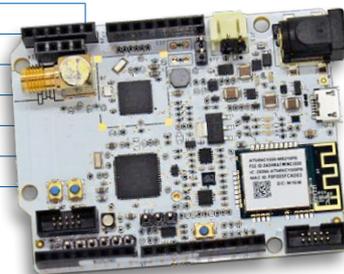
Tiger

La carte *Tiger* est équipée de deux processeurs : un SAMD21 de Microchip, et un KW41 de NXP. La fonction première du KW41 est d'autoriser la communication radio multiprotocole (Bluetooth BLE 4.2, Thread et Zigbee), mais il peut aussi servir à programmer le contrôleur principal de la carte. Grâce à un « arbitre » intégré, le SAMD 21 et le KW41 peuvent interagir avec le module Wi-Fi ou communiquer via le port série. Comme ses cousines Lion et Fox, la *Tiger* a les dimensions d'une Arduino Uno et se programme facilement et rapidement depuis l'EDI Arduino, ou depuis Atmel Studio puisque la carte dispose d'un port SWD.

Processeur 1	SAMD21J18A-64 ARM Cortex-M0+ de Microchip, 48 MHz, RAM de 32 Ko & flash de 256 Ko
Processeur 2	KW41 ARM Cortex-M0+ de NXP, 48 MHz, RAM de 128 Ko & flash de 512 Ko
Mémoire embarquée	EEPROM I ² C de 32 Ko, EEPROM SPI de 4 Mbits
Connectivité	USB, Wi-Fi, radio multiprotocole (BLE 4.2, Thread & Zigbee)
Extension	Connecteurs pour shield Arduino Uno, connecteur FTDI
Alimentation	Embase jack, USB, pile Li-Po (>= 700 mAh)

Plus d'informations

MC27561-Tiger : www.arrow.com/en/products/mc27561-tiger/arrow-development-tools



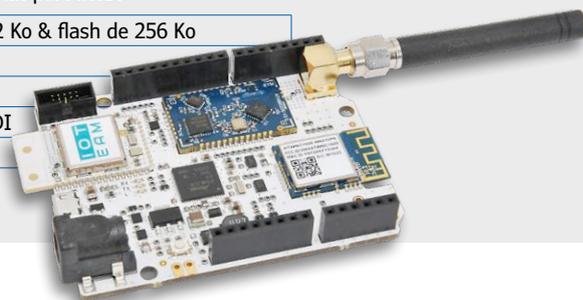
Dragonfly

La carte *Dragonfly* est compatible avec la carte Arduino Zero et permet des communications sans fil aussi bien à courte qu'à longue portée par le biais de trois modules – Dusty, Sigfox et Wi-Fi. Elle peut servir d'interface entre le nuage et l'écosystème Dusty (basé sur SmartMesh IP), soit directement, soit à travers une passerelle Wi-Fi SmartMesh IP. La carte *Dragonfly* a les dimensions d'une Arduino Zero et se programme facilement et rapidement depuis l'EDI Arduino. D'autres chaînes de compilation ARM sont utilisables grâce à la présence commode d'un port SWD.

Processeur	SAMD21J18B-MU ARM Cortex-M0+ de Microchip, 48 MHz, RAM de 32 Ko & flash de 256 Ko
Mémoire embarquée	EEPROM I ² C de 32 Ko
Connectivité	USB, Wi-Fi b/g/n, Dusty (SmartMesh IP), Sigfox
Extension	Connecteurs pour shields Arduino Zero, port console compatible FTDI
Alimentation	Embase jack, USB, pile 3 V (p. ex. 2 AA)

Plus d'informations

MC27561-Dragonfly : www.arrow.com/en/products/mc27561-dragonfly/arrow-development-tools



Dustino

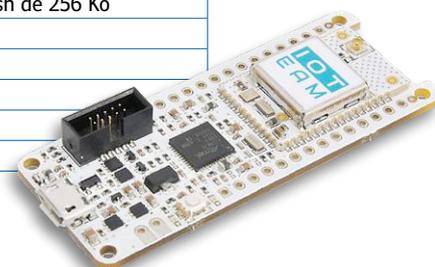
Dustino est une petite carte conçue pour les applications en réseau sans fil reposant sur Dusty (SmartMesh IP). Elle est compatible avec la carte Arduino MKR1000 et en a les dimensions. Deux modèles sont disponibles : *Dustino ITM-DOPA-B-01* et son antenne imprimée, ou *Dustino ITM-DOUF-B-01* et son connecteur U.FL. Les fonctions de maillage sans fil de la puce SmartMesh IP sont synonymes de 99,999 % de fiabilité pour les applications industrielles de l'IdO, et de plus de 10 ans de longévité de la pile. *Dustino* est reconnue par l'EDI Arduino mais peut aussi être programmée et déboguée avec d'autres chaînes de compilation ARM via son port SWD.

Processeur	SAMD21G18A-48 ARM Cortex-M0+ de Microchip, 32 MHz, RAM de 32 Ko & flash de 256 Ko
Connectivité	USB, Dusty (SmartMesh IP)
Dustino ITM-DOPA-B-01	Antenne imprimée
Dustino ITM-DOUF-B-01	Connecteur U.FL
Extension	Connecteurs pour Arduino MKR1000
Alimentation	Embase jack, USB, pile 3 V (p. ex. 2 AA)

Plus d'informations

Dustino ITM-DOPA-B-01 : www.arrow.com/en/products/itm-dopa-b-01/arrow-development-tools

Dustino ITM-DOUF-B-01 : www.arrow.com/en/products/itm-douf-b-01/arrow-development-tool



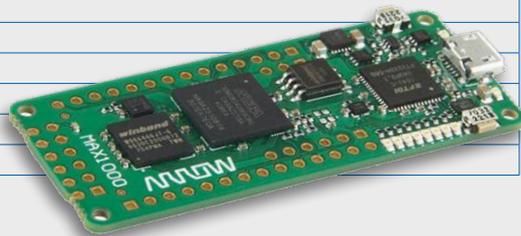
MAX1000

Les FPGA aussi ont leur place dans l'IdO et le prototypage rapide, comme le prouve la carte *MAX1000*. Basée sur un FPGA MAX 10 d'Intel, elle offre 8000 éléments logiques, 378 kbits de RAM, 1376 kbits de mémoire flash, un multiplicateur 18 x 18 bits (ou 2 multiplicateurs 9 x 9 bits) et un CA/N. Elle peut faire tourner un processeur softcore Nios II à 32 bits et donne accès à un port I²C ou SPI, ce qui la rend accessible aux débutants en FPGA, sans exclure les experts. La *MAX1000* est prise en charge par le logiciel gratuit *Quartus Prime Lite*. Elle se programme par JTAG ou via le programmeur intégré USB Blaster.

Processeur	MAX10 10M08SAU169C8G d'Intel, 8k éléments logiques, RAM de 4,75 Ko, flash de 172 Ko (adapté au processeur softcore Nios II à 32 bits)
Mémoire embarquée	SDRAM de 64 Mbits, flash de 64 Mbits
Connectivité	USB
Capteur	Accéléromètre à 3 axes
Extension	Connecteurs pour Arduino MKR, PMOD, E/S utilisateur
Programmation	JTAG, USB Blaster
Alimentation	USB

Plus d'informations

MAX1000 : www.arrow.com/en/products/max1000/arrow-development-tools



CYC1000

La carte *CYC1000* ressemble à s'y méprendre à la *MAX1000* mais en diffère notablement. Elle repose en effet sur un FPGA Cyclone 10 d'Intel offrant 24 000 éléments logiques, 594 kbits de RAM et un multiplicateur 18 x 18 bits (ou 2 multiplicateurs 9 x 9 bits). Une suite complète de blocs de propriété intellectuelle accélère le traitement numérique du signal et donne toute sa puissance à la carte. Elle peut bien sûr faire tourner un processeur softcore Nios II à 32 bits et donne accès à un port I²C ou SPI. Elle est prise en charge par le logiciel gratuit *Quartus Prime Lite* et se programme par JTAG ou via le programmeur intégré USB d'Arrow.

Processeur	Cyclone 10 LP 10CL025YU256C8G d'Intel, 24k éléments logiques, RAM de 74,25 Ko (adapté au processeur softcore Nios II à 32 bits)
Mémoire embarquée	SDRAM de 64 Mbits, flash de 16 Mbits
Connectivité	USB
Capteur	Accéléromètre à 3 axes
Extension	Connecteurs pour Arduino MKR, PMOD, E/S utilisateur
Programmation	JTAG, USB Blaster
Alimentation	USB

Plus d'informations

CYC1000 : www.arrow.com/en/products/cyc1000/arrow-development-tools



RSL1000

Conçue pour l'Internet des Objets, la carte *RSL1000* repose sur le SoC RSL10 de ON Semiconductor, un système sur puce radio multiprotocole et Bluetooth 5.0. Sa taille identique à celle de la famille MKR d'Arduino la rend idéale pour la conception de dispositifs de l'IdO servant de nœuds périphériques, pour l'informatique vestimentaire ou pour exploiter les caractéristiques de Bluetooth 5.0.

Processeur	SoC radio multiprotocole RSL10 de ON Semiconductor, ARM Cortex-M3 core+ LPDSP32 core, RAM de 88 Ko, flash de 384 Ko
Connectivité	USB, Bluetooth 5.0
Capteurs	
Extension	Connecteurs Arduino MKR
Programmation	J-Link OB
Alimentation	USB

Plus d'informations

RSL1000 : www.arrow.com/rsl1000



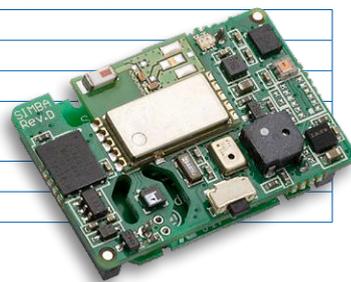
Simba-Pro

Si vous avez besoin d'une toute petite carte à faible consommation, la *Simba-Pro* fera votre bonheur. Basée sur un ARM Cortex-M4 à 32 bits et faible consommation de ST, cette carte mère de la famille des systèmes embarqués *SensiBLE* est équipée d'un module Bluetooth BLE, de 8 capteurs, d'une LED rouge/vert, d'un bouton utilisateur et d'un buzzer. Sa mémoire flash AT25XE041B permet de stocker 4 Mbits de données. Elle peut être alimentée par une pile bouton de 2 V et communiquer avec la plateforme en nuage *IBM Watson IoT*. Comme elle ne mesure que 20 x 30 mm, elle se glisse presque partout.

Processeur	STM32L476RG ARM Cortex-M4 de STmicroelectronics, 80 MHz, SRAM de 128 Ko, flash de 1 Mo
Mémoire embarquée	Flash de 4 Mbits
Connectivité	Bluetooth BLE 4.1, port USB
Capteurs	Température, humidité, accéléromètre à 3 axes, gyroscope à 3 axes, magnétomètre, pression, microphone, lumière (couleur, IR, ambiante)
Extension	Connecteur à 13 broches, connecteur à 14 broches
Alimentation	4,5 à 7 V CC, pile de 2 à 3,6 V (CR2025 ; CR2032)

Plus d'informations

Simba-Pro : www.arrow.com/en/products/simba-pro/sensiedge



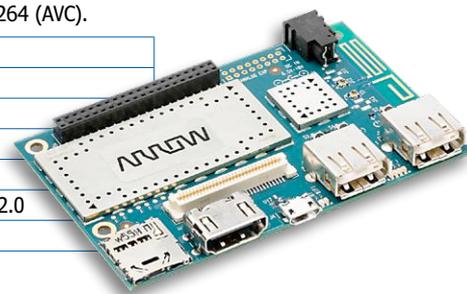
Dragonboard 410c

La carte *DragonBoard 410c* loge un processeur Snapdragon 410 de Qualcomm à modes 32 et 64 bits, ainsi qu'un quadricœur ARM Cortex-A53 cadencé jusqu'à 1,2 GHz par cœur. Elle prend en charge les systèmes d'exploitation Android 5.1, Linux et *Windows 10 IoT core*. Elle est conforme à la spécification *96Boards Consumer Edition* et embarque des modules Wi-Fi, Bluetooth et GPS. Son GPU et son processeur d'images intégré prennent en charge les caméras offrant jusqu'à 13 Mpx de définition, lisent des vidéos au format HD 1080p et enregistrent au format H.264 (AVC).

Processeur	Snapdragon 410 de Qualcomm à quadricœur ARM Cortex-A53
Mémoire embarquée	LPDDR3 533 MHz de 1 Go, e.MMC 4.51 de 8 Go, SD 3.0 (UHS-I)
Connectivité	USB, Wi-Fi b/g/n, Bluetooth 4.1
Multimédia	HDMI (audio & vidéo), prise en charge des caméras (13 Mpx)
Capteurs	GPS
Extension	Connecteur Low Speed (LS) à 40 broches & High Speed (HS) à 60 broches, USB-A, USB 2.0
Alimentation	6,5 – 18 (12) V CC

Plus d'informations

Dragonboard 410c : www.arrow.com/en/products/dragonboard410ciotsdk/arrow-development-tools



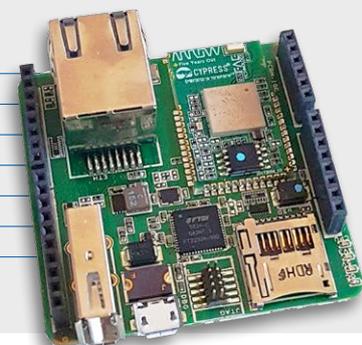
Quadro IoT Wi-Fi Kit

De la taille d'une Arduino Uno, la carte *Quadro* comporte une petite carte imprimée (18 x 30 mm) équipée d'un compact système sur puce (SoC) de Murata. Ce SoC est décliné en deux versions, *1GC* et *1GC-Imp05*. Tous deux logent un système CYW43907 de Cypress équipé d'un ARM Cortex-R4, d'une SRAM de 2 Mo et d'un contrôleur Wi-Fi/Ethernet de 2,4 GHz et 5 GHz. On y trouve aussi un module cryptographique et un circuit RF. Le SoC *1GC-Imp05* est compatible avec la plateforme sécurisée *Electric Imp* pour l'IdO. La petite carte imprimée est un module certifié de communication et de traitement, prêt pour la production.

Processeur	CYW43907 de Cypress à ARM Cortex-R4, 320 MHz, RAM de 2 Mo
Mémoire embarquée	Flash de 128 Mo, carte micro-SD
Connectivité	USB, Ethernet 10/100, Wi-Fi bande a/b/g/n
Extension	Connecteurs pour shield Arduino
Programmation	JTAG
Alimentation	USB

Plus d'informations

Quadro IoT Wi-fi Kit : www.arrow.com/en/products/sh-pcbm-1gc/arrow-development-tools



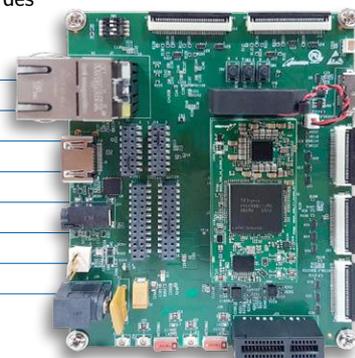
EIC-Q820-210

Le kit de développement *EIC-Q820-210* repose sur le système sur module (SoM) Eragon 820, lui-même construit autour d'un SoC Snapdragon 820 de Qualcomm. La carte permet le déploiement rapide d'applications nécessitant de la vidéo 4K, des modules de connexion, des graphismes de qualité, de la puissance de traitement et une consommation efficace. Cette puissante carte adaptée aux caméras 4K, aux écrans et autres drones, prend en charge Android Lollipop 5.1.1, Linux (Debian 8.0) et Windows 10.

Processeur	Snapdragon 820 de Qualcomm, CPU 64 bits + GPU + DSP
Mémoire embarquée	LPDDR4 de 4 Go, UFS de 32 Go / eMMC de 16 Go
Connectivité	USB, Ethernet 10/100/1000, Wi-Fi a/b/g/n, Bluetooth 4.1 + BLE
Capteurs	GPS, accéléromètre, gyroscope, magnétomètre
Extension	Audio, caméra, PCIe WiGig, PCIe SATA, HDMI 4K, connecteur d'extension <i>low speed</i>
Programmation	JTAG
Alimentation	Embase jack 12 V CC

Plus d'informations

EIC-Q820-210 : www.arrow.com/en/products/eic-q820-210/einforchips-limited



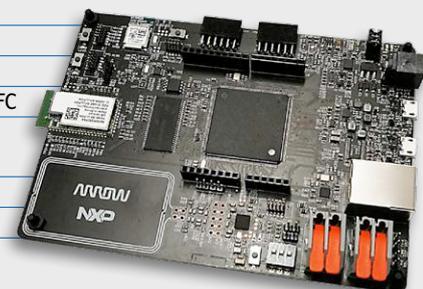
Carte HMI Arrow NXP IoT, dite HANI

Conçue pour les interfaces homme-machine (HMI), cette carte prend en charge de nombreuses tailles d'écrans couleur, du petit afficheur QVGA (240 x 320) de 3,5" aux écrans de 7" de 800 x 400 pixels. Sa connectivité comprend un module sans fil multiprotocole (à puce KW41Z compatible BLE 4.2, Zigbee et Thread), un module Wi-Fi, un lecteur NFC, 2 ports CAN et des interfaces Ethernet et USB. Son vaste lot de capteurs embarqués permet des interactions utilisateur telles que les commandes gestuelles, donc la création de dispositifs intelligents connectés.

Processeur	LPC54618 de NXP à ARM Cortex-M4, 180 MHz, SRAM de 200 Ko, flash de 512 Ko
Mémoire embarquée	Flash de 16 Mbits
Connectivité	Ethernet 10/100, CAN, USB, radio multiprotocole (BLE, Thread et Zigbee), Wi-Fi b/g/n, NFC
Capteurs	Température, accéléromètre à 3 axes, gyroscope à 3 axes, pression biocompatible, luminosité ambiante
Extension	Connecteurs pour shield Arduino
Alimentation	Embase jack, USB

Plus d'informations

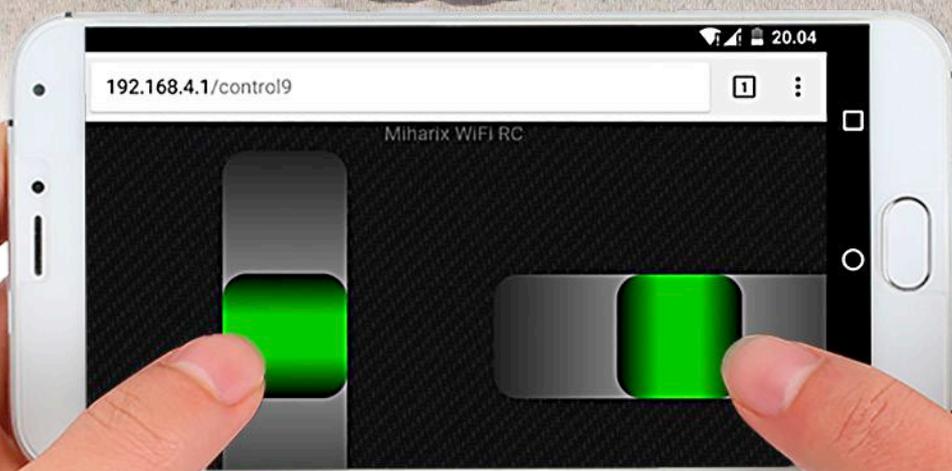
HANI : www.arrow.com/en/campaigns/arrow-hani-board



ESP8266 dans le monde du modélisme télécommande par ordiphone simple et bon marché

Miha Kocar

Même dans leur version la plus simple, les télécommandes des modèles réduits terrestres, aériens ou même navals, ne sont pas forcément à la portée des petites bourses. Mais si l'on utilise le Wi-Fi comme support de communication, on fait déjà l'économie de l'émetteur, étant donné que n'importe quel ordiphone fera l'affaire. Et le récepteur est extrêmement simple et bon marché !



Je suis professeur dans une école des faubourgs de Ljubljana (Slovénie) où je donne depuis quelques années un cours de réalisation de modèles réduits. C'est là que j'ai eu l'idée de concevoir une télécommande Wi-Fi parce que je me heurte fréquemment au même problème avec les futurs élèves :

L'élève : Moi aussi, je veux construire quelque chose.

Moi : Volontiers ! Tu penses à quoi ? Ici, nous construisons des avions, des bateaux à moteur, des voiliers, des voitures...

L'élève (aux yeux grand ouverts) : Est-ce qu'on peut aussi les télécommander ?

Moi : Bien entendu ! Tiens, en voici quelques-uns qui le sont.

L'élève (aux yeux encore plus grands et brillants) : Waouh ! Je peux construire tout ça ici ? (puis avec un air soucieux)

Mais ça doit coûter un max, non ?

Moi (devinant la cause du problème) : Le coût du matériel n'est pas très élevé ; on peut même prendre du matériel dans la réserve de l'école. La seule chose vraiment coûteuse, c'est la télécommande, il faut bien compter dans les cent euros.

L'élève (devenu triste) : ...mes parents n'ont pas autant d'argent.

Moi : Pas de problème, tu construis quelque chose sans télécommande ; on pourra toujours l'ajouter par la suite.

L'élève (maussade) : Bof, c'est d'un ennui ! Moi, ça ne m'intéresse que si c'est télécommandé !

Et encore un modéliste potentiel de perdu ! Il me fallait changer cela...

Contexte

L'ironie de cette situation est que tous les enfants possèdent déjà des ordiphones. En tant qu'électronicien amateur, j'ai donc imaginé une solution moins chère : télécommander des modèles réduits avec un ordiphone. Mais toutes mes recherches m'ont conduit invariablement à des solutions nécessitant une appli spécifique qui ne fonctionnait pas avec tous les ordiphones. Par ailleurs, la société chinoise Espressif proposait avec l'ESP8266 [1] une puce Wi-Fi déjà populaire et avant tout bon marché. Alors je me suis mis à expérimenter avec ce module à microcontrôleur à 32 bits adapté aux capteurs et actionneurs. À ma grande surprise, il suffisait d'un circuit externe extrêmement simple pour commander les servomoteurs montés dans les modèles réduits. Je ne pouvais pas croire qu'il n'existait pas de carte chinoise spécialement conçue pour télécommander des servomoteurs. J'ai pourtant trouvé une quantité de modules avec la puce et une antenne imprimée ainsi que nombre de cartes de développement Wi-Fi en tous genres pour quelques euros sur Ebay et AliExpress. Mes critères étaient clairs : la meilleure compatibilité possible avec les ordiphones, des connexions compatibles avec le matériel de modélisme, un poids plume, une capacité d'extension, une simplicité de programmation et d'utilisation et enfin, mais pas le moins important, un prix bas. Pour limiter autant que possible le poids, il ne serait pas mal d'avoir un programmeur externe plutôt qu'embarqué.

Modules, cartes et davantage

Pour toutes ces raisons, mon premier choix s'est porté sur « Witty Cloud » (**fig. 1**), une petite carte de développement



Figure 1. La carte de développement Wi-Fi Witty Cloud, très bon marché, est constituée d'un empilement de deux circuits imprimés. La partie inférieure est le programmeur.

ESP8266 bon marché. On la trouve partout à petit prix et elle est en deux parties. La partie inférieure consiste simplement en une carte de programmation USB. Sur la partie supérieure se trouve le module ESP8266 avec tout l'attirail HF, une LED RVB, une photorésistance, un bouton-poussoir et une prise micro-USB. C'est donc le matériel parfait pour le modéliste en herbe qui veut faire ses premiers pas en électronique. C'est approprié pour l'expérimentation, mais incomplet pour l'utilisation sur des modèles réduits (pour la connexion de servomoteurs). Pour concevoir ma propre carte-mère, mieux adaptée (**fig. 2**), j'ai choisi le module ESP8266 de type ESP-14 (**fig. 3**). Cette carte est aussi en grande partie compatible avec la variante ESP-12 plus facile à trouver. Outre le module, ma petite carte comprend un régulateur de tension de 3,3 V, deux résistances de 10 k Ω , un condensateur de filtrage de 1 μ F et une diode

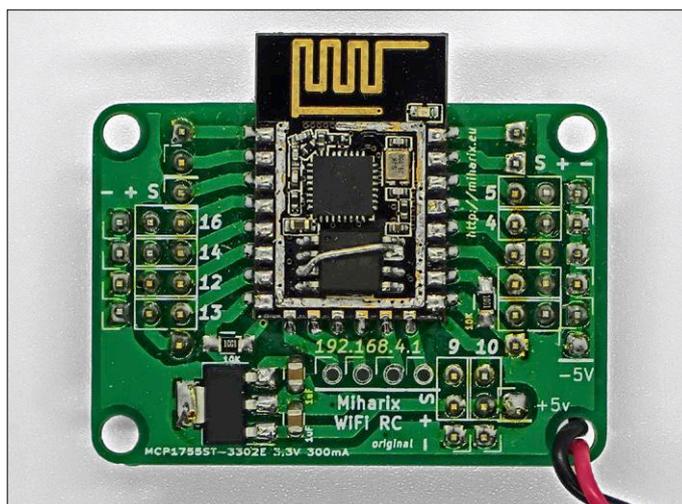


Figure 2. Circuit imprimé conçu par l'auteur : le module ESP8266, avec ses entrailles à l'air, est modifié pour gérer onze servomoteurs.



Figure 3. Le module ESP8266 de type ESP-14 ne contient pratiquement tout le nécessaire pour une télécommande Wi-Fi, y compris une antenne imprimée.

Schottky (court-circuitable) de protection contre l'inversion de polarité. Par rapport à celles de Witty Cloud, les broches TX0, RX0, REST, GPIO0, GND et VCC sont décalées vers l'intérieur de 1/10», soit 2,54 mm. Il faut donc un adaptateur pour la programmation avec le programmeur de Witty Cloud.

Un coup d'œil au schéma (fig. 4) montre que cette solution est en principe apte à piloter jusqu'à onze servomoteurs. Mais si l'on veut vraiment aller jusque-là, il faut légèrement modifier le module. C'est faisable, mais presque impossible pour le débutant – il faut enlever le couvercle du module et intervenir à l'intérieur sur des parties très fines (la figure 4 montre le module ESP8266 ouvert et modifié). Pour cette raison, mon logiciel ne le prévoit même pas ; tout est limité à six servomoteurs. La carte présente donc des connexions pour six servomoteurs, ce qui devrait être largement suffisant pour la télécommande de voitures miniatures ou de robots simples de niveau scolaire. Comme le lecteur traditionnel d'Elektor est tout de même un peu plus âgé, la modification pour la configura-

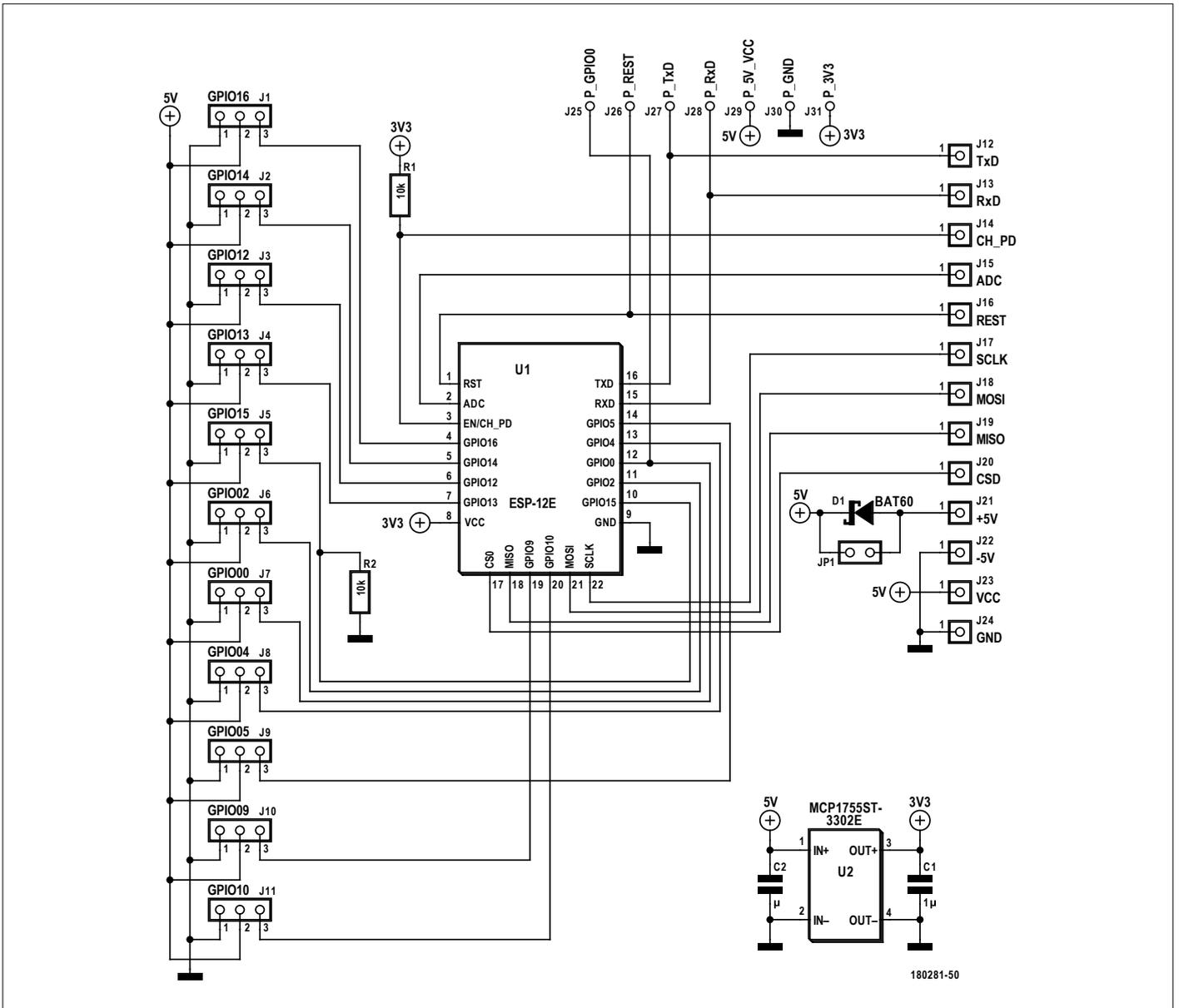


Figure 4. Le schéma de la carte-mère conçue par l'auteur (fig. 2) est très simple, car aucun convertisseur de niveau n'est nécessaire pour attaquer les servomoteurs.

tion complète est décrite dans l'encadré « **Onze servos** » (à vos risques et périls...).

Les modélistes savent que les servomoteurs, les récepteurs radio, etc. sont alimentés sous 5 V ou même 6 V. Or, l'ESP8266 est une puce sous 3,3 V, avec des entrées/sorties en 3,3 V aussi. Il est donc raisonnable de penser qu'une sorte de convertisseur de niveau va s'avérer indispensable pour attaquer les servomoteurs. En fait, il n'en est rien. Dans toute ma documentation de modélisme, je n'ai pas trouvé un seul servomoteur qui ne fonctionnerait pas sous les 3,3 V de l'ESP8266. C'est pourquoi le circuit est tellement simple.

Programmation

Tout d'abord une indication : si une carte Witty Cloud neuve semble ne pas être programmable, c'est peut-être à cause d'un câble USB défectueux. Alors commencez par le remplacer et essayez à nouveau.

Comme environnement de programmation, on va se servir du bon vieux EDI Arduino. Je recommande une installation portable. Pour cela, on crée simplement un répertoire baptisé « portable » dans le répertoire d'installation d'Arduino [2]. Après l'installation, on va à [Fichier->Préférences](#) et dans le champ URL de gestionnaires de cartes supplémentaires, on saisit l'URL indiquée en [3]. Ensuite on cherche « ESP8266 » sous [Croquis -> Inclure une bibliothèque -> Gérer les bibliothèques](#) et on installe *ESP8266*, *ESP8266HTTPClient*, *DNSServer*, *ESP8266WebServer*, *ESP8266Wi-Fi* et *Servo(esp8266)*. Tout est alors prêt pour la programmation d'une interface simple pour des modèles réduits lents ou immobiles. Pourquoi seulement des objets lents ? Parce que leur commande passe par l'appel de quelques liens, ce qui peut quelquefois prendre du temps. La structure du code de cette commande simple est la suivante : au début du code source apparaissent les directives `include` pour les bibliothèques *ESP8266Wi-Fi.h*, *Wi-FiClient.h*, *ESP8266WebServer.h* et *Servo.h*. Les deux premières bibliothèques prennent en charge la communication Wi-Fi, *ESP8266WebServer.h* gère un point d'accès Wi-Fi et *Servo.h* sert à produire les signaux de modulation de largeur d'impulsion pour les servomoteurs. On peut ensuite déclarer le mot de passe et le SSID (nom du point d'accès Wi-Fi) sous forme de `const char *`. La longueur du mot de passe devrait dépasser huit caractères. S'il est plus court, même si la compilation et le téléversement se passent bien, ça ne fonctionnera pas. Il faut une variable globale pour chaque servomoteur piloté. Avec l'instruction `ESP8266WebServer server(80)`, on démarre sur l'ESP8266 un serveur web sur le port 80.

Nous en arrivons aux fonctions qui mettent le servomoteur dans une position donnée : `servo_x.write(degrés)` ou `servo_x.writeMicroseconds(μs)`. Ces fonctions doivent être suivies de l'instruction `server.send(200, "text/html", respondX)`. Le paramètre `respondX` doit contenir le code HTML du message à afficher sur l'écran de l'ordiphone dans son navigateur web. Dans la partie `setup`, les instructions `servo_x.attach(broche)` et `servo_x.write(degrés)` positionnent le servomoteur. Les instructions `Wi-Fi.softAP(ssid, password)` et `IPAddress myIP = Wi-Fi.softAPIP()` activent le point d'accès Wi-Fi.

Il ne reste plus qu'à créer un lien correspondant à chaque position du servomoteur et à le lier à la fonction correspondante : Exemple 1 : `server.on("/", Root)` exécute la fonction `Root()` à l'appel de l'adresse `http://192.168.4.1/`

Exemple 2 : `server.on("/L5", L50)` exécute la fonction `L50()`

Onze servos

Pour les courageux, voici la modification nécessaire pour les onze sorties servomoteur possibles :

- Chauffer avec un pistolet à souder à air chaud le couvercle métallique du module ESP8266.
- Puis essayer de retirer ce couvercle (avec un tournevis, mais avec douceur).
- Dessouder les broches 3 et 7 de la puce flash interne et les tordre vers le haut.
- Raccorder ces deux broches (3 et 7) à la broche 8 (VCC).

C'est terminé. La **figure 2** montre ce à quoi cela peut ressembler. Toutefois il n'y a sur ce circuit des broches de masse et d'alimentation que pour six servomoteurs, mais il y a des broches de commande pour les cinq autres.

à l'appel de l'adresse `http://192.168.4.1/15`

Tout cela est suivi de l'instruction `server.begin()` qui démarre le serveur web.

Enfin, dans la partie `loop()`, on a l'instruction `server.handleClient()` pour que toute requête HTTP du client (ordiphone) soit traitée aussi rapidement que possible.

Interfaces

Avec cette interface simple, on arrive à joindre un appareil mobile en commençant par établir une connexion avec le point d'accès de l'ESP8266 puis en appelant l'adresse IP correspondante sur son navigateur web favori, par exemple « `http://192.168.4.1` ».

Outre l'interface logicielle rudimentaire décrite ici, il existe une solution plus avancée disponible sur GitHub [4]. Cette solution étendue ne se contente pas de liens simples, mais utilise des canaux *WebSockets* pour la commande des servomoteurs. Le tout repose sur un site web comprenant divers éléments et fichiers de types *.html*, *.css*, *.jpg* et *.js*. Les *WebSockets* réduisent considérablement le délai entre la commande et le mouvement des servomoteurs, et rendent la télécommande utilisable aussi pour des maquettes plus rapides, telles que des voitures et des bateaux. Cette interface étendue utilise l'adresse standard `http://192.168.4.1`. Mais on peut procéder à divers réglages par `http://192.168.4.1/setup` (ou `/admin`, `/root`, `/config`, `/edit`, `/administrator`, `/uredi`, `/nastavi`). On peut spécifier le SSID, le mot de passe, la création d'un point d'accès Wi-Fi ou la connexion à un réseau Wi-Fi existant, ainsi qu'une temporisation de sécurité. Pour chaque servomoteur, on peut régler individuellement les paramètres *trim center*, débattement max/min, position de sécurité, expo, vitesse, retournement et, (pour l'instant), neuf modes d'interface différents. Il s'y ajoute la possibilité de restaurer les paramètres à leur valeur standard par cavalier ou réinitialisation, ce qui est bien utile en cas de saisie incorrecte du SSID ou du mot de passe, ou de connexion à un réseau Wi-Fi existant. On peut aussi attribuer à chaque puce sa propre SSID.

Le logiciel est structuré de telle sorte qu'il est très facile de l'utiliser pour la commande d'autres objets. Par souci d'évi-

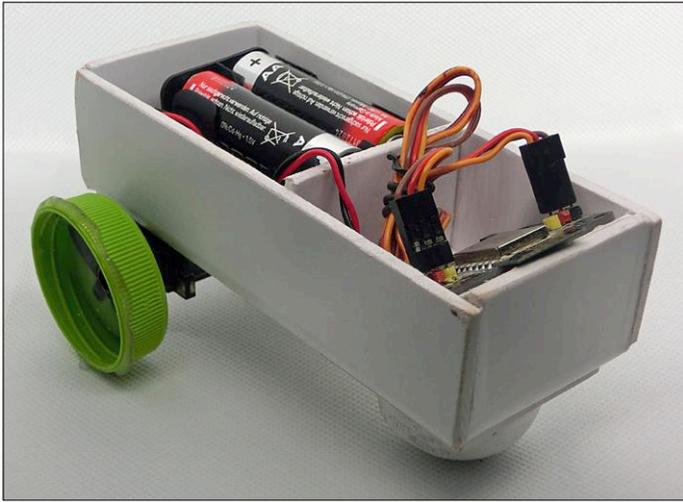


Figure 5. Un robot très simple construit par un élève de l'auteur.

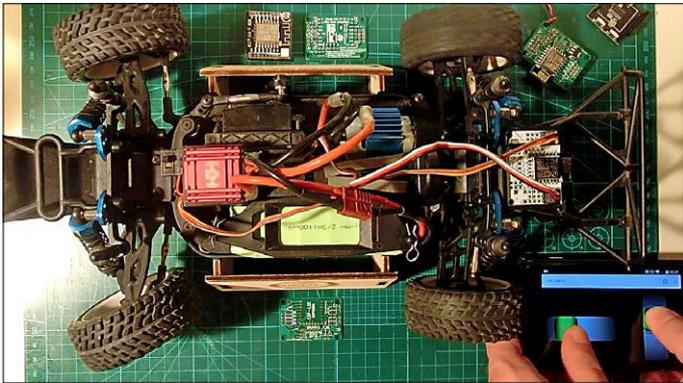


Figure 6. Cette maquette de voiture plus complexe est aussi une réalisation scolaire pilotée à la télécommande Wi-Fi.

ter les redondances, on se contentera de ne décrire que les particularités :

- Temporisation de sécurité (*fail save timeout*) : si la puce ne reçoit aucune nouvelle commande au bout de X secondes, on admet que la communication est rompue. Les servomoteurs se mettent alors dans la position de sécurité.
- Expo : une fonction très utilisée pour convertir les mouvements linéaires du manche en mouvements non linéaires des servomoteurs. On arrive fréquemment à améliorer ainsi la précision dans le domaine central du manche.
- Connexion à un réseau Wi-Fi existant : c'est possible avec un ESP8266. L'adresse IP est alors celle du serveur DHCP local.
- Mode : appel direct de diverses interfaces par « /controlX » après l'adresse où 'X' représente le numéro de l'interface.

Fonctions de manipulation des fichiers

Pour téléverser vers la puce les fichiers avec les extensions *.html*, *.css*, *.jpg* ou *.js*, Arduino a besoin du greffon SPIFFS (*Serial Peripheral Interface Flash File System*). Cela permet à la puce d'utiliser sa mémoire flash comme mémoire de masse pour

y créer, lire et effacer des fichiers et des répertoires. L'installation du greffon s'effectue par simple extraction dans */Arduino/tools/*. Dans le répertoire du projet Arduino, on crée un nouveau répertoire « data ». Après un redémarrage d'Arduino, on trouve sous *Outils* un nouvel élément nommé *ESP8266 Sketch Data Upload* dont on se sert pour copier le contenu complet du répertoire « data » dans la mémoire flash de l'ESP8266. Remarques : pendant le téléversement, le moniteur série doit être obligatoirement fermé. Le téléversement peut durer plusieurs minutes.

Dans cette version du logiciel, le code Arduino contient trois autres directives *include* :

- *FS.h* – nécessaire pour l'accès SPIFFS.
- *WebSocketsServer.h* – serveur socket du côté Arduino.
- *ArduinoJson.h* – facilite la manipulation du fichier *json*.

Le fichier *json* contient la configuration individuelle qui est directement ou indirectement accessible par le navigateur web (*http://192.168.4.1/config.json*). Cela simplifie la réalisation d'une configuration de base standard, la mise en mémoire d'une configuration temporaire (jusqu'à la prochaine réinitialisation ou au prochain redémarrage) ainsi que la conception d'une structure web claire.

Mais comment faire pour lire du côté Arduino avec SPIFFS le contenu de la mémoire ? Après les instructions *include*, on complète la fonction *config* avec *SPIFFS.begin()*. Après la séquence d'instructions *File MonFichier = SPIFFS.open("/Data.txt", "r")*, on peut alors lire le contenu du fichier *Data.txt* avec *MonFichier.read()* comme s'il s'agissait d'une entrée série normale. La lecture se prolonge tant que *MonFichier.available()* retourne la valeur *true*. Il faut encore fermer le fichier avec *MonFichier.close()*.

L'écriture fonctionne de la même manière. Les instructions s'appellent ici *File MonFichier = SPIFFS.open("/Data.txt", "w")* et *MonFichier.write()* et le fichier est fermé à la fin. La lecture du fichier *json* s'effectue de la manière suivante : on ouvre le fichier en lecture avec *File configFile = SPIFFS.open("/config.json", "r")* et on en obtient la taille avec *size_t size = configFile.size()*. Avec *std::unique_ptr buf(new char[size])*, on crée un tampon temporaire qu'on remplit de données avec *configFile.readBytes(buf.get(), size)*. On crée le tampon pour l'analyseur JSON avec *StaticJsonBuffer<1024> jsonBuffer* qu'on analyse avec *JsonObject& json = jsonBuffer.parseObject(buf.get())*. On peut alors lire le contenu du fichier *json* simplement comme un tableau (exemple : *RCmode = json["Mode"]*). Enregistrer des données *json* se fait de la même manière, sans analyseur, bien sûr, mais à la fin intervient l'instruction *json.prettyPrintTo(configFile)*.

On envoie le contenu d'un fichier au navigateur avec *streamFile*. Pour cela, on ouvre le fichier et on l'envoie avec *size_t sent = server.streamFile(file, contentType)*, avec le type MIME correct dans le paramètre *contentType*. Pour des fichiers *.html*, on aura donc « text/html ».

Canaux WebSockets

Les WebSockets fonctionnent de la manière suivante : après l'inclusion, on déclare avec *WebSocketsServer websocket = WebSocketsServer(81)* la disponibilité du serveur WebSocket sur le port 81. Ensuite il faut exécuter un *websocket*.

`begin()` dans `setup()` pour démarrer le serveur et lier la fonction `WebSocketEvent` aux événements `WebSocket` avec `WebSocket.onEvent(WebSocketEvent)`. Après l'exécution de `void WebSocketEvent(uint8_t num, WStype_t type, uint8_t * payload, size_t length)`, on peut alors déterminer l'adresse IP du client avec `IPAddress ip = WebSocket.remoteIP(num)` et envoyer des données au client avec `WebSocket.sendTXT(num, "Hello!")` et les lire dans un tableau tel que `payload[0]`. Dans ce cas, le client est le navigateur de l'ordiphone qui exécute le code Javascript.

Dans le script `.js` apparaît au début du code :

```
var connection = new WebSocket('ws://' + location.hostname + ':81/', ['arduino'])
```

pour l'établissement de la connexion.

Si l'on veut envoyer un paquet de données, on utilise :

```
connection.send('Hello Server!')
```

On peut lire les données reçues avec : `connection.onmessage = function (e) { console.log('Server: ', e.data) }` où le résultat apparaît dans le listage de la console du navigateur seulement dans le mode « Développement web » (à activer avec `Ctrl+Shift+K` dans le cas de Firefox).

Certaines des interfaces disponibles contiennent du code qui émule un manche. Ce code a été élaboré à l'aide de « nipple.js » par « yoannmoinet ».

Le code source du projet est disponible au téléchargement sous [5].

Conclusion

Comme exemples de ce que les élèves de l'auteur ont réalisé avec cette télécommande par ordiphone, la **figure 5** montre un robot ultrasimple : deux roues motorisées, les piles et l'électronique ; la **figure 6** montre une maquette de voiture déjà plus élaborée.

Pour finir, mentionnons qu'une telle télécommande Wi-Fi a une portée limitée à 80-90 m environ. Cette solution est certes très économique, mais un ordiphone ne remplacera jamais un vrai manche. La **figure 7** montre la page de configuration sur un ordiphone et la **figure 8**, une copie d'écran d'une interface utilisateur (comme exemple de l'une des neuf variantes réalisées à ce jour), ici avec deux curseurs, l'un horizontal et l'autre vertical. Il n'y a pratiquement pas de limite à l'imagination dans la conception de telles interfaces.

Si l'auteur décide de déménager sur GitLab, il vous en informera sur son site web [5]. Vous trouverez sur ce site des informations complémentaires (y compris en langue anglaise), des fichiers de conception du circuit imprimé, du logiciel et des clips vidéo montrant de quoi il retourne, malgré la langue slovène. S'il vous faut davantage d'informations générales sur le matériel ou le logiciel, vous croulerez sous les références que vous fournira Google quand vous aurez saisi « ESP8266 » ou « Witty Cloud ». ◀

(180281-03 – version française : Helmut Müller)

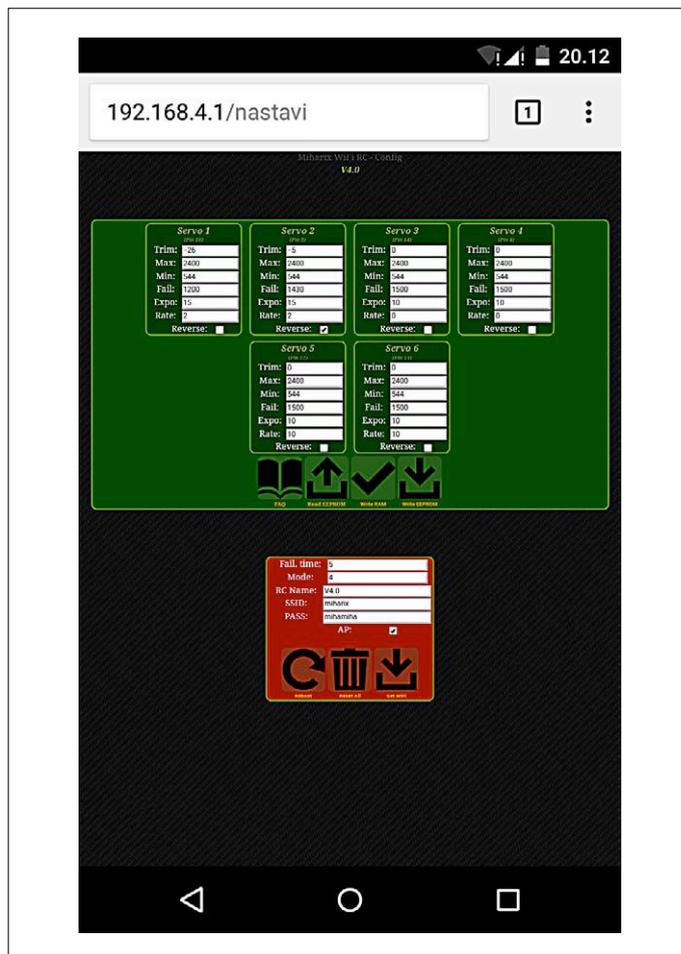


Figure 7. Copie d'écran de la page de configuration de la télécommande Wi-Fi à ESP8266.

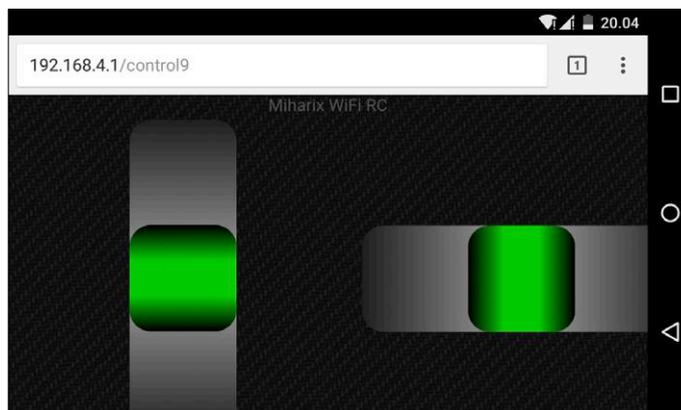


Figure 8. Copie d'écran de l'une des neuf interfaces réalisées à ce jour pour la télécommande à ESP8266. On distingue un curseur horizontal et un curseur vertical.

Liens

- [1] ESP8266 : <https://bit.ly/2zIKRd7>
- [2] Installation portable : www.arduino.cc/en/Guide/PortableIDE
- [3] Gestionnaire de carte : http://arduino.esp8266.com/stable/package_esp8266com_index.json
- [4] Dépôt GitHub : <https://github.com/miharix/miharix-wifi-rc>
- [5] Site de l'auteur : <http://wifi-rc.miharix.eu>



vol tous azimuts

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

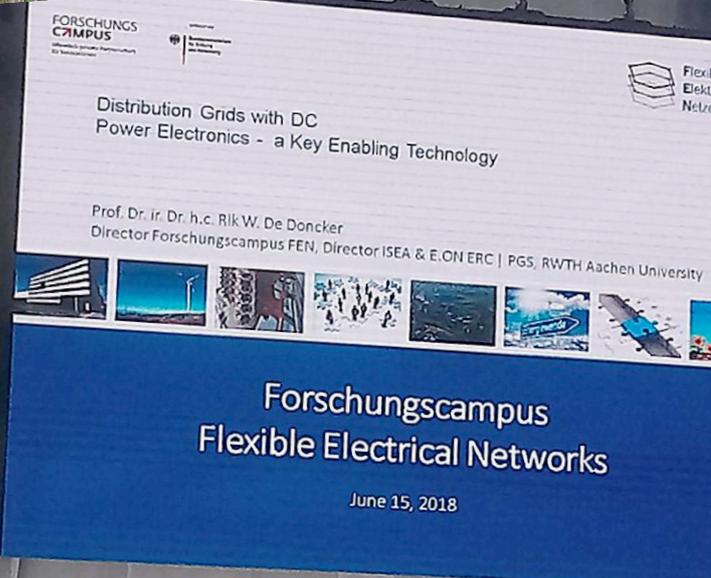
Qui a raison ?



CC (à gauche) contre CA (à droite, ou bien était-ce l'inverse ?) : qui va gagner ?

Luc Lemmens, reporter d'un jour pour le labo d'Elektor

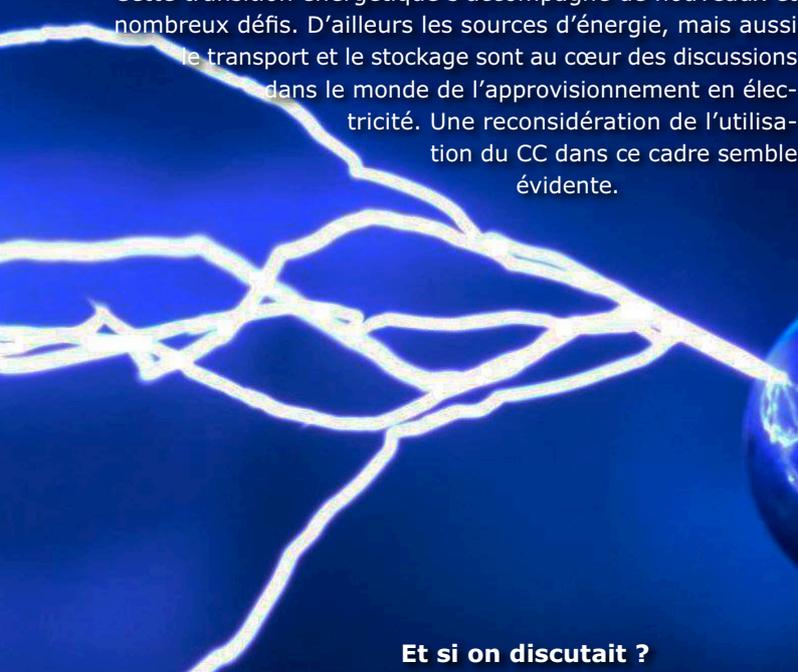
Il y a quelque 120 ans déjà que la soi-disant « guerre des courants » s'est terminée sur la victoire du système à courant alternatif (CA) de Tesla et de Westinghouse. Edison, avec son système de distribution d'énergie électrique à tension continue (CC), l'a perdue. Depuis lors, notre réseau d'énergie se compose en grande partie de lignes véhiculant du courant alternatif. À l'époque d'Edison et de Tesla, c'était une bonne idée, mais est-ce encore le cas ? La circuiterie interne de la plupart des appareils (ménagers) électriques exige du courant continu. Ces appareils contiennent donc presque tous une alimentation électrique qui convertit la tension alternative du réseau en tension continue, cette conversion donne lieu à une perte d'énergie. De plus, ce sont souvent ces « fameux » modules d'alimentation qui tombent en panne les premiers. Les condensateurs électrolytiques qu'ils renferment ont généralement une durée de vie relativement courte. L'alimentation CA d'une lampe à LED par ex. trépassé souvent bien avant les LED, qui elles restent bien plus longtemps en état de briller. Un réseau en CC aurait certainement des avantages. Les systèmes de production d'énergie à partir de sources alternatives (comme le soleil) débitent généralement du courant continu. La plupart des installations photovoltaïques conver-





tissent le CC produit en CA, pour pouvoir l'envoyer sur le réseau. Cette étape aussi pourrait être supprimée.

Les combustibles fossiles, qui occupent une grande place dans notre production d'énergie, s'épuisent. En outre, des exigences et des objectifs environnementaux plus stricts (notamment en termes d'émissions de CO₂) nous obligent, de plus en plus, à utiliser des sources d'énergie renouvelables et plus propres. Cette transition énergétique s'accompagne de nouveaux et nombreux défis. D'ailleurs les sources d'énergie, mais aussi le transport et le stockage sont au cœur des discussions dans le monde de l'approvisionnement en électricité. Une reconsidération de l'utilisation du CC dans ce cadre semble évidente.



Et si on discutait ?

Le 15 juin 2018, la *Stichting Gelijkspanning Nederland* (fondation « Courant Continu » des Pays-Bas) a organisé le « DC Congres Groot Gelijk 2018 ». Si on traduit, le congrès est intitulé : *le CC a raison*. Les institutions KIEN Innovatiemeesters, CityTec et TKI Urban Energy sont à l'origine de ce congrès qui a eu lieu dans la ville de Nieuwegein (province d'Utrecht). « Une journée d'inspiration sur le rôle du courant continu dans la transition énergétique », comme l'annonçait l'organisation. Cette journée a vu passer un groupe varié de chercheurs, concepteurs, responsables de l'élaboration des politiques et des plans, et responsables d'exécution.

Trois orateurs-vedettes ont animé une grande partie de la matinée. L'après-midi était réservé à des sessions plus courtes, au cours desquelles divers aspects et développements dans le domaine du CC ont été abordés.

Orateurs-vedettes

Le premier orateur, Doekle Terpstra, président d'Uneto VNI (organisation patronale), a souligné les défis que pose la transition énergétique aux installateurs.

Selon lui, les systèmes à CC y joueront certainement un rôle essentiel. Il a également mis en garde contre la pénurie de professionnels, notant qu'avec les technologies actuelles et futures, les technologies de l'information joueront un rôle de plus en plus important et que la demande de personnel technique hautement qualifié – y compris dans les techniques d'installation – ne fera que croître. Actuellement, le système d'enseignement néerlandais (idem pour la France sans doute) ne sait pas répondre à cette demande, en ce qui concerne les installations en CC surtout, ne serait-ce que par manque d'enseignants.

Le second orateur était Ed Buddenbaum du ministère de l'Économie et du Climat, secrétaire du *Top Sector Energy*. Il a parlé de la vision d'avenir et de la politique énergétique du gouvernement néerlandais, qui visent, entre autres, à limiter les émissions de CO₂. La réduction de l'utilisation des combustibles fossiles, y compris pour la production d'électricité, est une priorité. À l'en croire, le CC peut certainement jouer un rôle à cet égard, mais ce n'est pas une fin en soi.

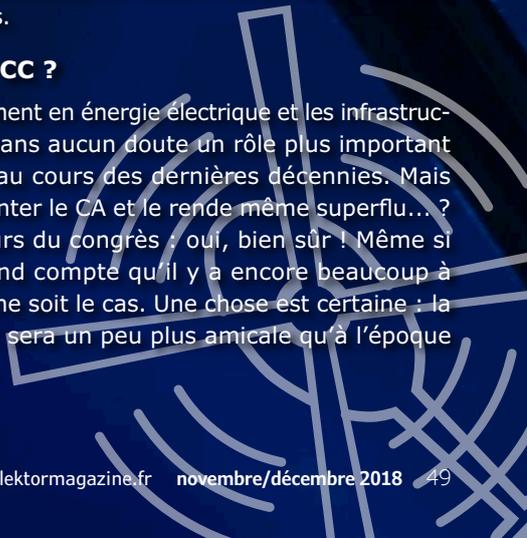
Le dernier intervenant, Rik W. De Doncker (directeur de l'*Institute for Power Electronics and Electrical Drives* de l'Université RWTH d'Aix-la-Chapelle) prévoit un avenir en or pour le courant continu où la tension alternative jouera un rôle mineur dans l'approvisionnement énergétique. Les énergies éolienne et solaire sont déjà largement disponibles, mais leur rendement est bien sûr lié à la météo et aux saisons. Aux heures de pointe, la surproduction peut être stockée pour ensuite être réinjectée sur le réseau en période de faible production. Le CC est la solution la plus évidente, en particulier pour le stockage de l'énergie (par exemple dans des batteries ou sous forme d'hydrogène).

Sentiments mitigés

Les avantages de la tension continue sont certes nombreux, mais malheureusement, lors du congrès, ses inconvénients de la tension continue ont été ignorés : corrosion par électrolyse ; exigences plus élevées auxquelles doit satisfaire le matériau de commutation par exemple. Et, inconvénient majeur dans la situation actuelle : l'absence de spécifications et de normes pour les installations CC. L'histoire du CA est parfaitement décrite aux Pays-Bas et est bien encadrée par les normes NEN 1010 ; pour le CC, il n'y a que la première version du « code pratique néerlandais » NPR 9090 (un ensemble d'explications et d'instructions pour sécuriser les installations CC). Notons que les Pays-Bas sont leader mondial en la matière. Cependant l'absence de réglementation a aussi des avantages ; la réglementation stricte que connaît le réseau de CA limite souvent les projets novateurs.

L'avenir est-il au CC ?

Dans l'approvisionnement en énergie électrique et les infrastructures, le CC jouera sans aucun doute un rôle plus important que celui qu'il a eu au cours des dernières décennies. Mais qu'il vienne à supplanter le CA et le rende même superflu... ? Pour les organisateurs du congrès : oui, bien sûr ! Même si tout le monde se rend compte qu'il y a encore beaucoup à faire avant que cela ne soit le cas. Une chose est certaine : la confrontation CA/CC sera un peu plus amicale qu'à l'époque d'Edison et de Tesla.



Histoire d'éléphant

Dans cette rubrique, nous vous parlons de la « guerre des courants ». Au cours de cette guerre, Thomas Edison a utilisé une arme létale pour prouver le danger du courant alternatif : l'électrocution d'animaux. La cruauté d'Edison a été rendue célèbre par l'électrocution d'un animal gigantesque : l'éléphante de cirque baptisée Topsy. La société de production cinématographique d'Edison a filmé cette exécution, mais, à en croire les historiens, Edison n'y était même pas. En effet, ce jour-là, le 4 janvier 1903, Edison avait perdu la guerre des courants depuis une dizaine d'années déjà. En réalité, l'exécution de Topsy avait été le seul moyen que ses propriétaires Frederick Thompson et Elmer Dundy avaient trouvé pour s'en débarrasser. Comme Topsy traînait une très mauvaise réputation, personne n'en voulait plus. Pour l'éliminer, on n'a laissé peu de place au hasard : empoisonnement, étranglement avec une machine à vapeur et, pour en finir, électrocution...



▶ Les molécules sont les composants électroniques de demain

Des molécules individuelles peuvent être reliées entre elles avec des nanoparticules métalliques et ainsi transporter une charge dans un circuit.

<https://phys.org/news/2018-07-future-electronics-chemical.html>

(180293-03 – version française : Guy Raedersdorf)

conception de matériel avec (V)HDL (2)

affichage de l'heure en binaire

Jörg Zollmann

La première partie de notre série sur VHDL nous a embarqués dans un tour d'horizon. À partir d'un exemple tout simple (une LED clignotante), nous avons découvert les différentes étapes du développement et les outils nécessaires à la conception d'un circuit CPLD/FPGA. Dans cette deuxième partie, nous nous focaliserons sur la description du matériel avec VHDL.

Tableau 1. Affectation des broches.		
Broche du connecteur K2 (haut gauche)	Nom du signal	Broche CPLD
2	hrs_r[4]	Pin_68
4	hrs_r[3]	Pin_69
6	hrs_r[2]	Pin_70
7	hrs_r[1]	Pin_71
8	hrs_r[0]	Pin_72
9	min_r[5]	Pin_73
10	min_r[4]	Pin_74
11	min_r[3]	Pin_75
12	min_r[2]	Pin_76
13	min_r[1]	Pin_77
14	min_r[0]	Pin_81
15	sec_r[5]	Pin_82
16	sec_r[4]	Pin_83
17	sec_r[3]	Pin_84
18	sec_r[2]	Pin_85
19	sec_r[1]	Pin_86
20	sec_r[0]	Pin_87

Dans cet article, nous allons construire pas à pas un afficheur numérique de l'heure. Pour que cet afficheur ne puisse être lu que par des amateurs d'électronique chevronnés, nous allons représenter les chiffres en format binaire. Pour l'affichage des heures, nous utiliserons des valeurs sur cinq bits, pour les minutes et les secondes, des valeurs sur six bits. On se contentera ici d'une simple simulation ; bien entendu, vous pouvez amener les valeurs des registres sur des broches de sortie du CPLD et y raccorder une rangée de LED (voir **tableau 1**). Pour cet article, nous reprenons notre carte CPLD au format DIL (voir l'encadré « @www.elektor.fr »). On peut télécharger le logiciel du projet sur la page web de cet article [2]. Le tout est organisé de la même façon que le projet décrit dans la première partie de cette série [3].

Diviseurs de fréquence

Pour la base de temps de notre horloge, nous utilisons le signal à 40 MHz de la carte CPLD. Pour une horloge, il faut plusieurs compteurs : un compteur pour les secondes incrémenté une fois par seconde, un pour les minutes et un pour les heures. Une première idée serait de produire pour chacun de ces compteurs des impulsions d'horloge à la fréquence appropriée et d'utiliser les compteurs pour compter ces impulsions. Le compteur des secondes serait ainsi alimenté avec un signal à 1 Hz, celui des heures à la cadence de 0,278 mHz ($1/3600 \text{ s}^{-1}$), etc. Cette approche est fonctionnellement correcte, mais présente l'inconvénient de ne pas donner un circuit totalement synchrone. Il vaut mieux alimenter toutes les parties du circuit avec le même signal d'horloge et n'exécuter les calculs (ici l'incrémenta-

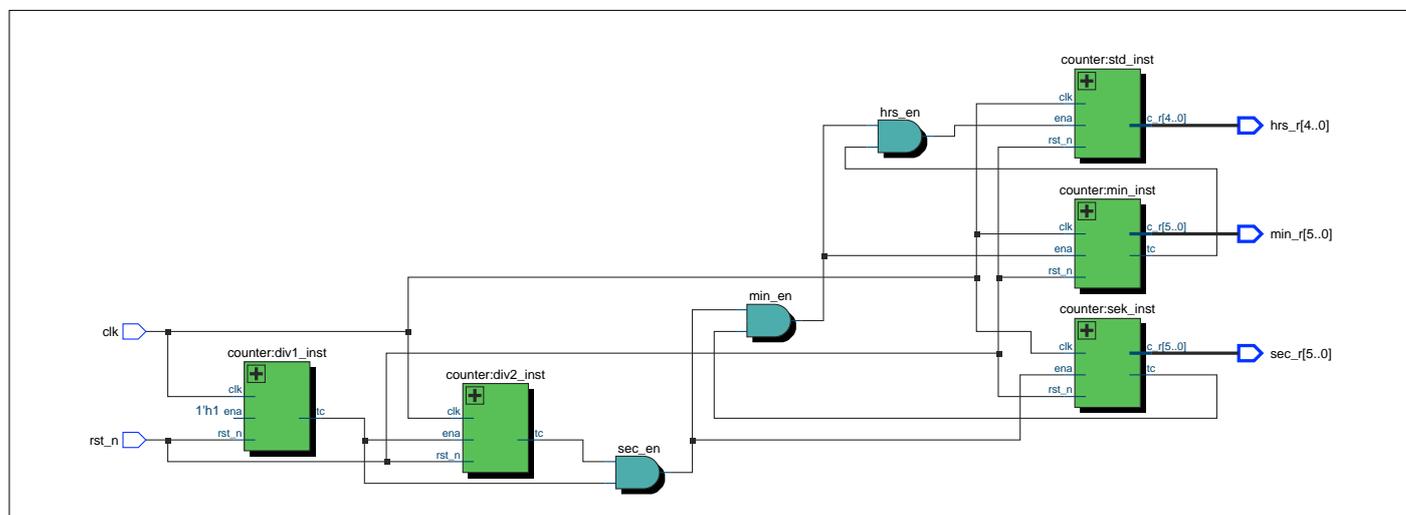


Figure 1. Synoptique de l'entité de niveau supérieur.

Listage 1. Compteur paramétrable.

```
Library ieee; -- On utilise la bibliothèque ieee
use ieee.std_logic_1164.all; -- dans laquelle on va chercher le paquet std_logic_1164
use ieee.numeric_std.all; -- dans le paquet numeric est défini entre autres le type unsigned

entity counter is -- Déclaration d'entité
    -- un structure generic est valable pour l'entité et l'architecture
    -- les 'generic' peuvent être écrasées dans la generic map au moment de l'instanciation
generic(c_width:natural:=5;
c_max_val:natural:=10;
reset_val:unsigned);
port( -- Liste des ports
clk:in std_logic;
rst_n:in std_logic;
ena:in boolean;
c_r:out unsigned(c_width-1downto0);
tc:out boolean
);
end counter; -- Fin de la déclaration d'entité

architecture rtl of counter is -- Déclaration d'architecture
begin

tc<=true when c_r=c_max_val else false; -- C'est toujours lorsque le compteur a atteint
-- sa valeur maximale que tc =true
-- sinon false

up_counter:process (clk,rst_n) is
begin
if (rst_n= '0')then
c_r<=reset_val; -- Après l'initialisation, le compteur démarre
-- avec la valeur reset_val définie dans la structure generic
elsif rising_edge(clk)then
if (ena)then
if (c_r=c_max_val)then
c_r<=(others=>'0'); -- Le compteur ne doit compter que jusqu'à sa valeur maximale
-- puis recommencer à zéro

else
c_r<= c_r+1; -- Tant que le compteur n'a pas atteint sa valeur maximale
-- il est incrémenté de 1 à chaque impulsion (si ena est actif)
end if;
end if;
end process up_counter;
end rtl;
```

tion) que lorsque l'impulsion de validation (*enable pulse*, *ena*) apparaît. Pour notre horloge, cela signifie que le compteur des secondes reçoit une impulsion de validation toutes les secondes, celui des minutes toutes les minutes, celui des heures toutes les heures. Pour un cadencement à 40 MHz, chaque impulsion ne dure qu'une période (25 ns). Pour produire les différentes impulsions, nous pouvons aussi utiliser un compteur qui, outre la valeur de comptage, contient un signal indiquant que la valeur maximale est atteinte (*tc* = *Terminal Count*). Un principe fondamental de VHDL est la conception hiérarchique du circuit. C'est pourquoi, dans notre exemple,

nous allons modéliser le fonctionnement de notre horloge au moyen de plusieurs instances de compteurs. La **figure 1** montre le synoptique du niveau hiérarchique supérieur (*Top-Level Entity*), produit par le RTL-Viewer de Quartus. On peut choisir l'entité (*entity*), ou fichier, que Quartus doit prendre comme sommet (*Top-Level*) dans la fenêtre *Project Navigator* (menu déroulant *Files*, clic droit sur le fichier désiré, *Set as Top-Level Entity*). La figure montre que chaque instance de compteur reçoit la même impulsion de cadencement et que ses entrées de validation reçoivent une combinaison logique des sorties *Terminal Count*.

Compteurs

L'horloge se compose donc de cinq instances différentes de compteurs. Les deux instances *div1_inst* et *div2_inst* servent à produire l'impulsion de validation à la fréquence de 1 Hz. Pour cela, le compteur *div1_inst* compte de 0 à 9999 et indique qu'il est arrivé à la valeur maximale toutes les 0,25 ms. Ce signal est utilisé pour valider l'entrée du compteur *div2_inst* qui incrémente donc son contenu toutes les 0,25 ms. Après 4000 impulsions [soit 1 s], ce compteur atteint sa valeur maximale. Les autres compteurs forment l'horloge proprement dite en comptant les secondes, les minutes et les heures qui fournissent

Listage 2. Banc d'essai pour tester les fonctions log2 et ceil.

```
Library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use ieee.math_real.log2;
use ieee.math_real.ceil;

entity testbench is
end entity testbench;

architecture tb of testbench is
begin
process is
begin
report "Comment fonctionnent log2 et ceil?";
for i in 1 to 9 loop
report "i= "&to_string(i)&
" log2= "&to_string(log2(REAL(i)))&" ceil(log2(i)): "
&to_string(INTEGER(ceil(log2(REAL(i)))));
end loop;
wait;
end process;

end architecture tb;
```

l'heure effective. Comme le contenu des deux compteurs *div* n'est pas significatif, on affecte explicitement au moment de leur instanciation l'attribut *open* à la

sortie *c_r* dans la *port map*. Le fait que chaque compteur puisse avoir une valeur maximale différente est pris en compte en VHDL au moyen de para-

mètres de type *generic*. Ces paramètres, valables pour l'entité et l'instance de l'élaboration, se comportent comme des constantes, mais peuvent être écrasés

Listage 3. Paquet pour la configuration des compteurs.

```
Library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

library work;
-- EN-TETE DU PAQUET =====
Package count_pkg is

type t_cnt_config is record
  t_CNT_div1:natural;
  t_CNT_div2:natural;
t_CNT_SEC:natural;
t_CNT_MIN:natural;
t_CNT_HRS:natural;
end record;

-- Constantes pour la synthèse
-- une constante de type 'record' t_cnt_config
Constant p_cnt_config_syn:t_cnt_config:=(
10000, -- T=10000*25ns enable Pulse toutes les 0,25 ms
4000, -- T=4000 *0.25ms enable Pulse toutes les secondes
60-1, -- 60 secondes --> 0-59
60-1, -- 60 minutes
24-1 -- 24 heures
);
-- Constantes pour la simulation
Constant p_cnt_config_sim:t_cnt_config:=(
10,
4,
60-1,
```

```

60-1,
24-1
);
end package count_pkg;
-- CORPS DU PAQUET =====
package body count_pkg is
-- Le paquet ne contient que des constantes pour la simulation.
-- Le corps du paquet est vide.
-- Ici se trouvent par exemple les définitions des fonctions.
end package body;

```

depuis des niveaux hiérarchiques plus élevés (nous y reviendrons plus tard). Le **listage 1** montre les définitions d'entité et d'architecture du compteur générique abondamment commentées. En VHDL, les opérations arithmétiques (dans le cas de notre compteur l'addition de 1) ne sont pas définies pour tous les types de données, mais seulement pour les types numériques (*integer*, *real*, *signed*, *unsigned*). Les types *unsigned* et *signed* sont définis dans la bibliothèque *numeric_std* et utilisent en interne le type *std_logic_vector*. C'est pourquoi nous prenons pour la valeur du compteur un signal de type *unsigned* et incluons les deux bibliothèques *numeric_*

std et *std_logic_1164*. Le compteur possède trois paramètres qui déterminent la valeur maximale, la valeur initiale (voir ci-dessous) et la largeur du registre en bits. Le signal de sortie *tc* et l'entrée de validation *ena* sont du type *boolean*. La détermination de la largeur des compteurs s'effectue au niveau hiérarchique immédiatement supérieur. Pour cela, on utilise deux fonctions de la bibliothèque *math_real*. Il est bien entendu possible d'écrire ses propres fonctions en VHDL, mais nous n'en aurons besoin que plus tard. Les deux fonctions *log2* et *ceil* permettent de déterminer le minimum de bits ou de registres indispensables pour pouvoir compter jusqu'à la valeur maxi-

male souhaitée. Le banc de test (*test bench*) du **listage 2** précise la fonction de ces deux fonctions. Ce banc de test est simulable avec la commande `do sim_tb.do`. Comme autre possibilité de se familiariser et de tester des fonctions inconnues, l'auteur suggère l'application web libre Eda-Playground [3]. Eda-Playground est un EDI basé web qui permet d'une manière simple d'écrire et de simuler du code HDL. Il n'est nul besoin d'installer un programme, ni d'écrire des scripts *do/.tcl*.

Réduction du temps de simulation

Dans Modelsim, la simulation de l'hor-

Listage 4. Extrait de l'entité du niveau supérieur.

```

entity watch is
generic(
g_cnt_config:t_cnt_config:=p_cnt_config_syn
-- Les valeurs par défaut de g_cnt_config sont les valeurs pour la synthèse.
-- Dans le banc de test, elles sont écrasées dans la generic map avec
-- g_cnt_config => p_cnt_config_sim.
);
port(--
...
)
);
end watch;
architecture rtl of watch is

constant c_CNT_div1:natural:= g_cnt_config.t_CNT_div1;
-- La valeur est assignée à la constante.
-- Ceci est effectué à la compilation (car il s'agit d'une constante).
...
begin
...
div1_inst:entity work.counter
generic map(
c_width=> div1_width,
c_max_val=> c_CNT_div1,-- La valeur maximale du compteur est maintenant différente
-- selon qu'on exécute la synthèse ou la compilation.
reset_val=>to_unsigned(0,div1_width)
)
port map(
...
);
...

```

Listage 5. Script Tcl pour créer le fichier *time_pkg.vhd*.

```
set filename ../src/time_pkg.vhd

# Heure courante
set hour [clock format [clock seconds]-format"%H"]
set min [clock format [clock seconds]-format"%M"]
set sec [clock format [clock seconds]-format"%S"]

# Le fichier existe-t-il ?
if {[file is file$filename]}{
puts "$filename exists"
puts "--- On efface et on recrée le fichier"
file delete $filename
} else {
puts "create $filename"
}

# Ouverture du fichier
if [catch{open"$filename"w"} f]{
puts stderr$f
} else {
puts $f"-----"
puts $f"-- AUTOMATICALLY GENERATED FILE DON'T CHANGE "
puts $f"-----"
puts $f"library IEEE; "
puts $f"use IEEE.STD_LOGIC_1164.all;"
puts $f"use IEEE.NUMERIC_STD.all; "

puts $f"library work;"
puts $f" package time_pkg is "

puts $f"constant c_hour: unsigned:= to_unsigned($hour,5);"
puts $f"constant c_min: unsigned:= to_unsigned($min,6);"
puts $f"constant c_sec: unsigned:= to_unsigned($sec,6);"

puts $f"end package time_pkg;"
puts $f"package body time_pkg is "
puts $f"end package body;"
}
# Fermeture du fichier
close $f
```

loge prend un certain temps à cause des valeurs de débordement réglées pour les compteurs. Pour ne tester que le mode de fonctionnement des compteurs, nous pouvons prendre des valeurs de débordement différentes de celles de la réalisation matérielle ultérieure. Pour ne pas avoir à adapter le code source avant chaque simulation, on a recours à un paquet (*package*) VHDL. Les paquets rassemblent les déclarations globales utilisées à plusieurs endroits. Dans les paquets, il est possible de déclarer des fonctions, des constantes et des types de données.

Le **listage 3** montre le paquet `count_pkg.vhd` utilisé pour l'horloge. Un paquet se compose d'une partie déclarative (*header*) et d'un corps (*body*). La partie déclarative constitue l'interface du

paquet avec l'extérieur et est comparable à un fichier *.h* de la programmation en C. Le corps contient en principe la définition des fonctions ou des procédures déclarées dans la partie déclarative. Dans notre paquet, il n'y a pas de déclaration de fonctions, son corps est donc vide. À la place, il y a une déclaration d'enregistrement (*record*) VHDL et la définition de deux constantes. Avec un enregistrement, on peut réaliser des combinaisons logiques de types de données et donc créer des types de données abstraits. Les enregistrements sont comparables aux `struct` de la programmation en C. L'accès aux éléments/champs d'un enregistrement s'effectue par leurs noms. Lorsqu'on utilise un enregistrement dans un processus et qu'on assigne des valeurs à certains de ses

éléments, il faut assigner des valeurs à tous les éléments de cet enregistrement. Dans notre cas, l'enregistrement baptisé `t_cnt_config` contient les cinq valeurs maximales des instances des compteurs. Les constantes s'appellent `p_cnt_config_syn` et `p_cnt_config_sim`. Ces noms ne servent qu'à distinguer les fonctions, ils peuvent être choisis arbitrairement (un choix des noms significatif et cohérent pour l'ensemble du projet s'est avéré très utile). Comme mentionné plus haut, les valeurs maximales des différentes instances de compteurs sont spécifiées par des constantes définies dans le paquet. Le **listage 4** montre un extrait de l'entité de niveau supérieur (*Top-Level-Entity*). Cette entité possède aussi un paramètre générique du type `t_cnt_config`, auquel est attribuée une valeur par défaut qui

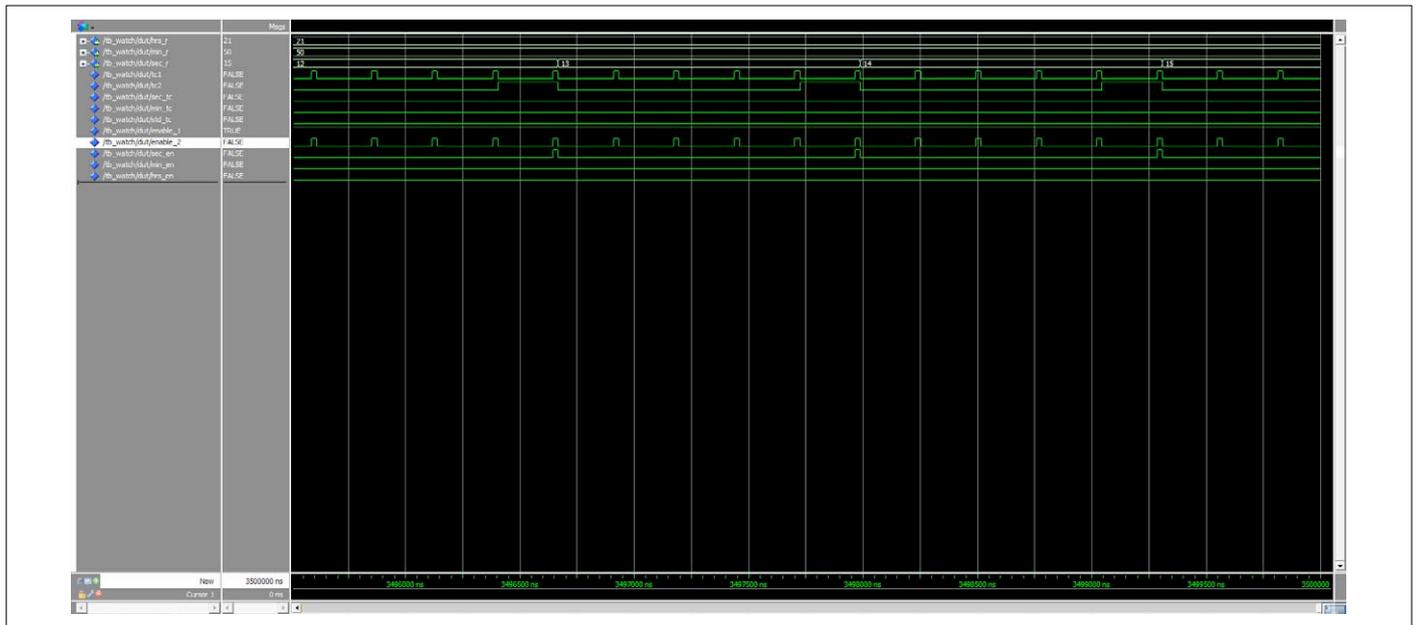


Figure 2. Simulation.

contient la configuration des compteurs nécessaire pour la synthèse. Lors de l'instanciation de l'horloge sur le banc de test, le jeu de paramètres est finalement écrasé par un autre, de sorte que lors de la simulation, le signal de validation est produit chaque milliseconde au lieu de chaque seconde.

Mise à l'heure de l'horloge

Il reste encore la question de savoir comment donner aux compteurs des heures, minutes et secondes des valeurs initiales. Si les compteurs doivent ultérieurement afficher l'heure et ne pas servir que de chronomètres, on doit au moins pouvoir les initialiser à une valeur quelconque. On y arrive en passant comme constantes génériques les valeurs de démarrage (ou d'initialisation) de la même manière que les valeurs maximales, à partir d'un paquet séparé. Nous pouvons ainsi modifier l'heure dans un fichier central avant toute nouvelle passe du compilateur. Cette méthode susceptible de provoquer des erreurs peut être automatisée avec une entrée dans le fichier *qsf* de l'horloge ou un script *tcl*. Quartus offre à l'utilisateur la possibilité d'exécuter des scripts intermédiaires à différents moments de la compilation. Dans le fichier *Setting*, la ligne `set_global_assignment -name PRE_FLOW_SCRIPT_FILE «quartus_sh:time.tcl»` impose que le fichier baptisé *time.tcl* soit exécuté au début d'une nouvelle passe. Le **listage 5** montre le script *tcl*. Ce script crée un fichier nommé *time_pkg.vhd* dans lequel il écrit l'heure cou-

rante dans trois constantes VHDL distinctes (*c_hour*, *c_min*, *c_sec*), sous la forme d'un paquet VHDL.

Banc de test

Le banc de test nécessaire à la simulation est réduit au minimum et n'est constitué que de l'instanciation de l'horloge et d'un autre processus qui active le signal d'initialisation au démarrage puis le désactive. Comme dans la première partie de cette série, la simulation est démarrée lorsqu'est exécutée la commande `do sim_watch.tcl` dans la fenêtre *Modelsim-Transcript*. La **figure 2** montre les registres de l'horloge (*hrs_r*, *min_r*, *sec_r*) ainsi que les signaux *enable* et *tc*. Comme le projet complet ne peut être traduit que si le fichier *time_pkg.vhd* est disponible, la synthèse doit avoir été exécutée au moins une fois pour que la simulation fonctionne.

vhd est disponible, la synthèse doit avoir été exécutée au moins une fois pour que la simulation fonctionne.

On continue ?

Cette horloge est un exemple de petit projet CPLD surtout destiné à encourager l'expérimentation et la réalisation personnelle. Dans la prochaine partie, nous élargirons nos connaissances de base et nous nous intéresserons aux automates. En outre, nous donnerons à notre horloge un meilleur affichage en écrivant un pilote d'afficheur en VHDL. Pour approfondir vos connaissances en VHDL, lancez-vous dans toute une série de tutoriels en ligne [4][5][6]. ◀

(180285-B-04 -
version française : Helmut Müller)

Liens

- [1] « Conception de matériel avec (V)HDL (1) », *Elektor* 09-10/2018 : www.elektormagazine.fr/180285-03
- [2] Page de l'article : www.elektormagazine.fr/180285-B-04
- [3] EDI sur le web : www.edaplayground.com/
- [4] Tutoriel VHDL (allemand) : <https://de.wikibooks.org/wiki/VHDL-Tutorium>
- [5] Tutoriel VHDL (anglais) : www.nandland.com/vhdl/tutorials/index.html
- [6] Tutoriel VHDL (anglais) : www.fpga4fun.com/HDLtutorials.html



@WWW.ELEKTOR.FR

→ Carte CPLD au format DIL

www.elektor.fr/cpld-breakout-board-160425-91

horloge à LED économe en énergie

affichage de l'heure à la demande

Miroslav Cina

Les horloges à affichage à cristaux liquides (LCD) sans rétroéclairage sont difficiles à lire au crépuscule et la nuit, même sous éclairage artificiel. Malheureusement, le rétroéclairage ramène la consommation électrique d'un LCD à celle d'un afficheur à LED. Une pendule murale à pile peut néanmoins s'accommoder d'un afficheur à LED si elle n'affiche quelque chose que si on le lui demande. Voilà qui est facile à réaliser avec un détecteur de mouvement passif à infrarouge (PIR).

Les horloges murales ou autres, équipées d'un afficheur à LED pour améliorer leur lisibilité, ont besoin d'un bloc d'alimentation et d'une prise de courant si l'on ne

veut pas passer son temps à changer les piles. Mais il arrive souvent qu'on veuille installer une horloge à un endroit trop éloigné d'une prise de courant. Un affi-

chage LCD à faible consommation n'est pas la bonne solution : il ne permet pas de lire l'heure dans l'obscurité. Les afficheurs à 7 segments à LED produisent

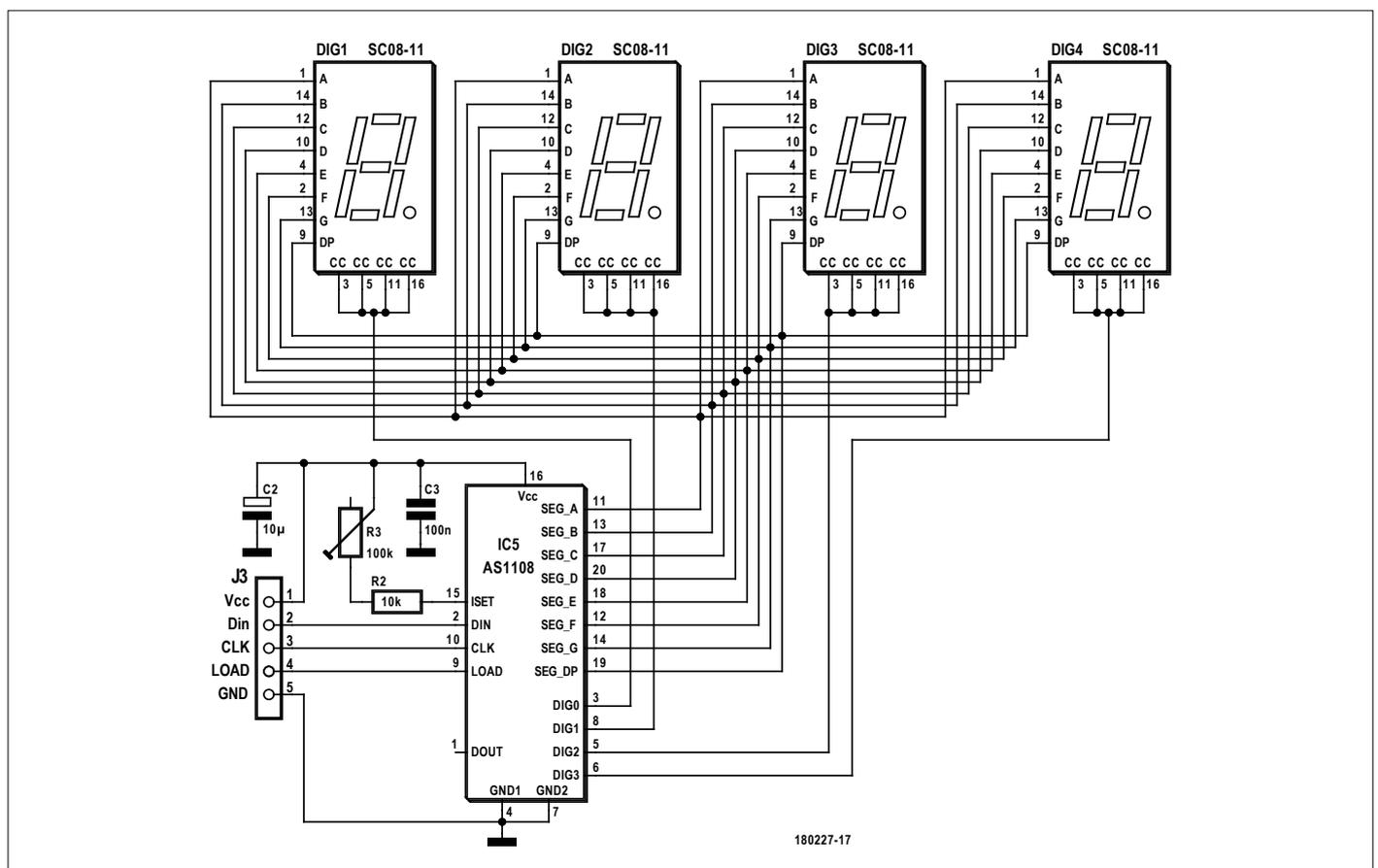


Figure 1. Le schéma du module d'affichage est très simple : un pilote et quatre afficheurs à 7 segments, voilà tout.

de la lumière et un excellent contraste, mais consomment malheureusement trop d'énergie. Alors, que faire ?

Affichage à LED économe en énergie

Il existe plusieurs moyens de réduire la consommation électrique des afficheurs à LED. On peut par exemple limiter le courant, de sorte que les segments à LED ne s'illuminent que faiblement. Pas de problème dans l'obscurité, mais, à la lumière, on n'y verrait malheureusement plus rien. Dans le même ordre d'idées, on pourrait n'activer l'affichage que quand on a besoin de lire l'heure. Mais réaliser cela au moyen d'un bouton serait peu pratique et d'une ergonomie contestable. J'ai décidé de le remplacer par un détecteur de mouvement PIR. Si l'on passe devant l'horloge (ou si on lève la main), l'heure ne s'affiche qu'à la demande et que pendant quelques secondes. Ainsi, la consommation moyenne s'établit à une valeur très faible.

Considérations préliminaires

Pour mon horloge PIR, l'affichage des heures et des minutes m'a paru suffisant. Il ne me fallait donc que quatre afficheurs à 7 segments, ce qui contribuait aussi à une basse consommation. Comme il y avait beaucoup de contre-indications à l'alimentation par le secteur, l'horloge devait se contenter de piles ou de batteries. De plus, le fonctionnement de l'horloge ne devait pas s'interrompre au moment du changement de piles, pour qu'on n'ait pas à la remettre à l'heure à chaque fois. Comme déjà mentionné, il suffisait de n'afficher l'heure qu'en présence d'une personne.

Mon horloge idéale s'offre le luxe d'une alimentation par une batterie Li-Ion, ce qui rend toute opération de changement de piles superflue, mais exige l'intégration d'un circuit de charge de la batterie, facilement réalisable aujourd'hui sous la forme d'un circuit intégré. Pour que l'horloge ne s'arrête pas quand la batterie est déchargée, l'heure doit être gérée par une puce d'horloge en temps réel (*Real Time Clock*, RTC) accompagnée de sa pile de sauvegarde. Le détecteur de présence est réalisé au moyen d'un détecteur de mouvement PIR.

Matériel

Le circuit est très raisonnable. Il ne comporte en effet qu'une puce RTC de Maxim, un microcontrôleur, un pilote

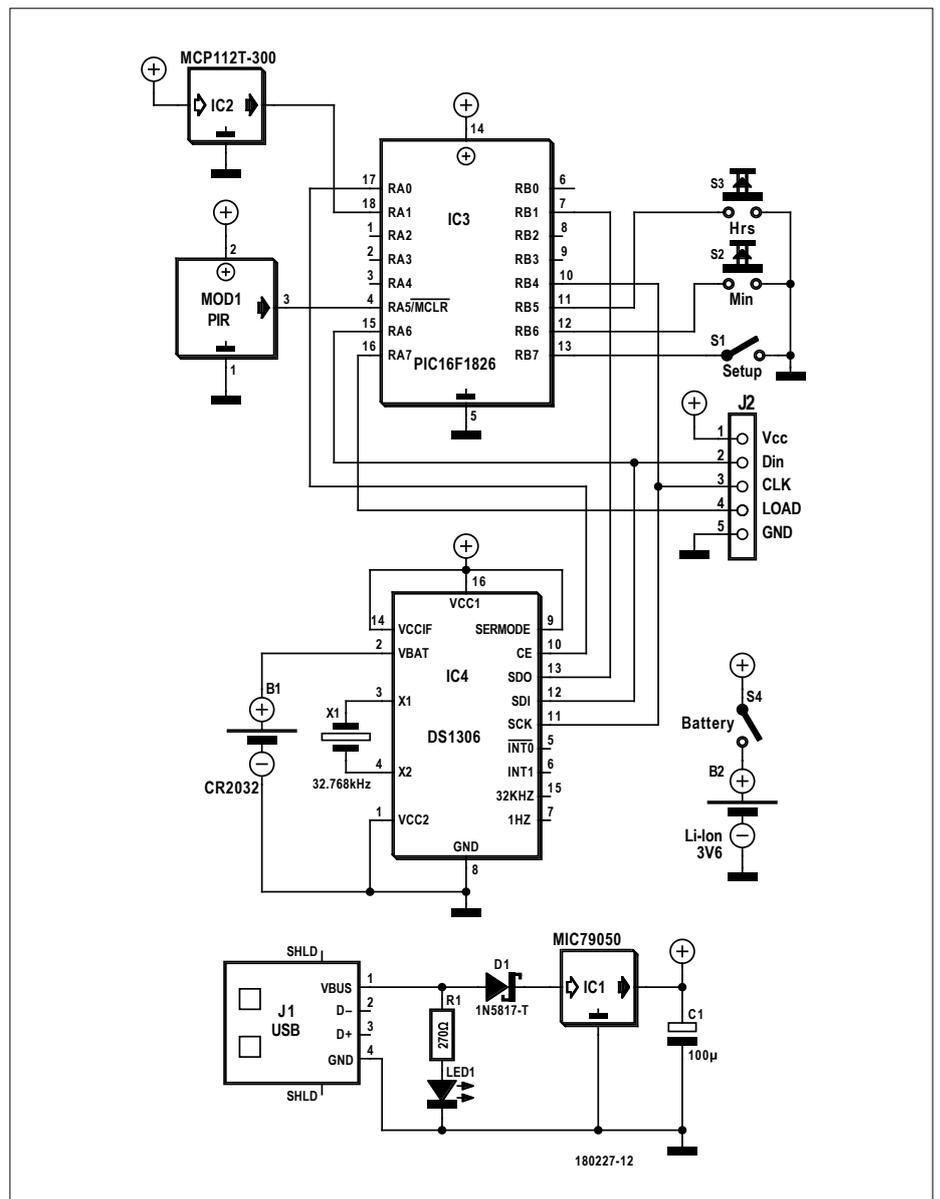


Figure 2. Malgré les quatre puces, l'horloge elle-même n'est guère plus compliquée.

d'afficheur, quatre afficheurs à 7 segments et un détecteur de mouvement PIR. S'y ajoutent une cellule Li-Ion avec son contrôleur de charge ainsi qu'une surveillance de tension pour éviter une décharge profonde de la batterie.

Le matériel de l'horloge est divisé en un module d'affichage avec son pilote sur une face et l'horloge proprement dite et le reste des fonctions sur l'autre face.

Module d'affichage

Le schéma du module d'affichage (fig. 1) est relativement simple. Les quatre afficheurs à 7 segments du type SC08-11 sont commandés par le pilote AS1108 d'AMS dans une configuration matricielle. En principe, tous les afficheurs à 7 segments à cathode commune conviennent.

Les deux condensateurs C2 et C3 empêchent un effondrement de la tension provoqué par des impulsions de courant ou des parasites, susceptible d'entraîner une réinitialisation involontaire d'IC5. La luminosité de l'afficheur dépend de la résistance formée par R2 en série avec le potentiomètre d'ajustage R3. La valeur minimale de R3 correspond à la luminosité maximale. On règle R3 sur la valeur de luminosité subjectivement la plus confortable.

Module principal

Toutes les autres fonctions sont comprises dans le module principal. Malgré les quatre puces, son schéma (fig. 2) reste relativement simple.

Le cœur de l'horloge réside dans la puce

MSSP

Un module MSSP peut être configuré dans une quantité de modes différents. On peut, par exemple, le transformer en un port I²C ou SPI. Sur les PIC, les deux modes peuvent fonctionner en maître ou en esclave.

Beaucoup de microcontrôleurs n'offrent qu'un module SSP, sans le 'M', qui ne peut être facilement configuré qu'en esclave. Le 'M' (=master) économise du temps et de la mémoire si l'on veut faire fonctionner le port en mode maître.

À l'origine, le bus SPI a été conçu par Motorola à l'occasion de l'introduction de l'unité centrale légendaire 68K. Il s'agit d'un port sériel travaillant en *full duplex*.

Cela complique un peu les choses, car pour la communication bidirectionnelle, outre la ligne d'horloge (SCLK) et des deux lignes de données MOSI (*Master Output, Slave Input*) / SDO (*Serial Data Out*) et MISO (*Master Input, Slave Output*) / SDI (*Serial Data In*), il faut encore au moins une ligne de sélection de puce CS (*Chip Select*) et une ligne de masse. En fait, chaque esclave nécessite sa propre ligne CS.

Par rapport à la technologie à collecteur ouvert de l'I²C, ce besoin accru en ressources et l'utilisation d'étages pilotes push-pull permettent des vitesses de transfert de données nettement plus élevées. L'inconvénient est la longueur limitée des lignes, qui est de l'ordre de celle du RS232. Avec des vitesses jusqu'à 80 Mbit/s, SPI est particulièrement indiqué pour la connexion de puces à haut débit comme les convertisseurs analogiques-numériques ou la mémoire flash externe. Quelques microcontrôleurs supportent même les deux protocoles en même temps (par exemple MSSP1 en maître I²C et MSSP2 en maître SPI), ce qui leur donne une grande souplesse d'utilisation.

Dans une application comme cette horloge, la vitesse de transfert des données n'a toutefois guère d'importance.

mots (kilowords) de 14 bits. Il s'y ajoute 256 octets de SRAM et 256 octets d'EEPROM (non utilisés ici). IC3 possède un port MSSP (*Master Synchronous Serial Port*), configuré en maître SPI, qui lui permet de non seulement lire l'heure sur IC4, mais aussi de commander le pilote IC5 des afficheurs. IC3 dispose encore d'un nombre suffisant d'entrées/sorties libres pour réaliser des extensions personnelles.

IC2 est un détecteur de tension MCP112T-300 de Microchip, dont la sortie passe au niveau bas quand la tension à son entrée tombe en dessous de 3,0 V. Cela indique à IC3 que la batterie est déchargée et qu'il peut le signaler afin de la protéger contre une décharge profonde. La tension d'une pile neuve B1 peut aller jusqu'à 3,2 V. IC4 peut alors avoir basculé sur B1 bien avant que IC2 se manifeste. IC2 n'est donc pas absolument indispensable. On peut s'en passer à condition de relier uniquement les broches 1 et 2.

Le détecteur PIR utilisé est le PIR 555-28027 facile à trouver dans le commerce. La seule chose importante est qu'il fonctionne sur une plage de 3 à 5 V et que sa sortie passe au niveau haut en cas de détection de mouvement.

Enfin, IC1 est la puce de gestion de la charge de la batterie Li-Ion. Elle charge la batterie au moyen d'un courant limité à 500 mA jusqu'à une tension de 4,2 V. La chute de tension dans IC1 est typiquement de 380 mV pour un courant de 500 mA, on peut donc l'alimenter avec un bloc d'alimentation 5 V du commerce branché sur la prise USB J1. D1 protège contre l'inversion de polarité et LED1 s'allume dès que l'alimentation est branchée sur le secteur. Le tampon C1 diminue encore l'impédance dynamique, déjà faible, de B2.

Tableau 1. Sous-programmes SPI.

Sous-programme	Fonction,	Utilisation
spi_init_m0	Initialisation MSSP SPI Master Mode 0	SPI Mode 0 pour la communication avec le pilote d'affichage AS1108
spi_init_m1	Initialisation MSSP SPI Master Mode 1	SPI Mode 1 [TS1] pour la communication avec la puce RTC DS1306
spi_send	Émission et réception par le port SPI	Communication SPI : émission et réception (simultanées) d'un octet

d'horloge en temps réel DS1306 (IC4). Elle compte le temps et comprend, outre la date, quelques fonctions supplémentaires non utilisées ici. La fréquence de base typique de 32 768 Hz est fournie par le quartz d'horloge X1. La pile de sauvegarde B1, une cellule bouton au lithium CR2032 standard du commerce, est raccordée à la broche 2 d'IC4. Sa logique d'intervention est simple : quand la tension de la batterie B2 sur la broche d'alimentation 16 d'IC4 est supérieure à celle de B1 sur la broche 2, le fonctionnement de l'horloge est normal. Si UB2 < UB1, B2 est réputée déchargée

et B1 alimente IC4, et seulement IC4 qui bloque sa communication SPI. B2 continue à alimenter le reste du circuit, mais le contrôleur IC3 constate le passage en mode sauvegarde et déclenche un signal optique de niveau bas de la batterie.

Encore une remarque sur la communication SPI d'IC4 : cette puce utilise un signal de sélection CS inhabituel, presque inversé, à savoir qu'elle est active quand CS est au niveau haut.

Le microcontrôleur utilisé est le PIC16F1826 de Microchip, qui possède une mémoire flash de 2 kW. Ces kW ne sont pas des kilowatts, mais des kilo-

Microgiciel

Le microgiciel est entièrement écrit en assembleur et plutôt compact : il occupe moins de 400 mots de la mémoire. Comme le matériel, le logiciel est divisé en deux parties : le fichier `91_SPI_com_16F1826.inc` contient toutes les fonctions relatives à la communication SPI, le fichier `01_PIR_clock_v1p09.asm` contient l'horloge proprement dite ainsi que toutes les autres fonctions. Les fichiers source téléchargeables depuis la page web du projet [1] sont modifiables avec n'importe quel éditeur de texte et ont été compilés avec MPASM (composant

de l'EDI MPLAB X) dans la version 5.65. Le fichier Hex produit est bien entendu également disponible sous [1].

Communication SPI

Les routines SPI peuvent aussi être utilisées dans d'autres projets, c'est pourquoi elles sont détaillées ci-dessous. On trouvera dans le **tableau 1** des informations sur les trois fonctions contenues dans le fichier d'inclusion `91_SPI_comm_16F1826.inc`. Elles ont été écrites pour le module MSSP du PIC16F1826. Moyennant quelques adaptations, elles sont portables vers n'importe quel microcontrôleur PIC à 8 bits avec module MSSP.

Les fonctions de commande des signaux CS de sélection de puce (pour les puces AS1108 et DS1306) sont très simples et adaptées au circuit réel et aux puces utilisées. Il s'agit des fonctions `spi_cs0_en` et `spi_cs1_en`, qui activent respectivement CS0 pour le pilote AS1108 et CS1 pour la puce RTC. Les fonctions `spi_cs0_dis` et `spi_cs1_dis` désactivent respectivement CS0 et CS1. Toutefois, les deux signaux CS sont toujours désactivés ensemble.

Programme principal

Même le programme principal reste simple. Après l'initialisation, il entre dans la boucle principale (`main_loop`), où il exécute périodiquement les opérations suivantes :

1. Interrogation du capteur PIR :
 - Pas de détection de mouvement : couper l'affichage LED et retourner au début
 - Mouvement détecté : continuer en 2.
2. Lire l'heure sur IC4 (fonction `rtc_spi_read`).
3. Les données lues sont converties en un format affichable.
4. Les données converties sont envoyées au pilote des afficheurs IC5.
5. En cas de mise à l'heure, appeler les fonctions correspondantes.

Après la mise à l'heure, la nouvelle heure est envoyée à la puce IC4 par la fonction `rtc_spi_write`.

Outre la boucle principale et la mise à

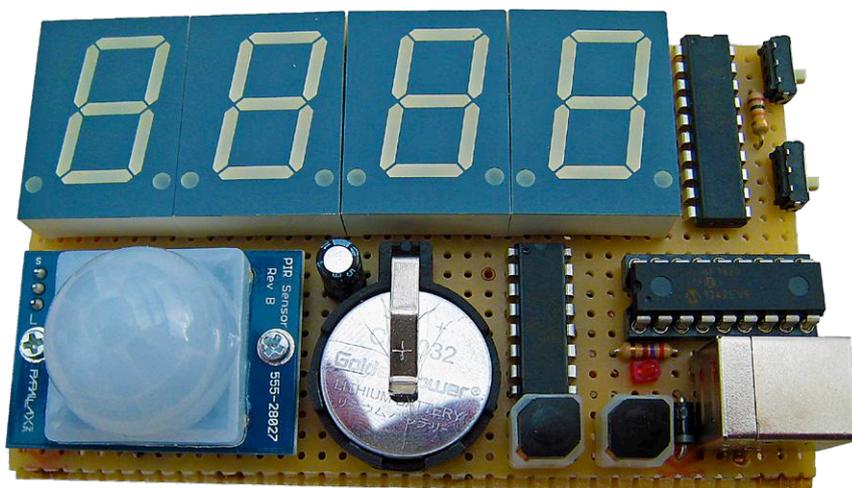


Figure 3. Le prototype de l'auteur, réalisé sur des cartes à trous.

l'heure, il y a encore les fonctions pour le pilote d'affichage. Elles sont facilement identifiables par leur préfixe `AMS_`.

Utilisation

Il y a trois touches S1, S2 et S3, qui suffisent pour le réglage de l'heure. Pour mettre à l'heure, on commence par appuyer sur la touche S1 (*setup*, sur la broche RB7 d'IC3). Tous les points décimaux des afficheurs s'allument pour indiquer le mode mise à l'heure.

Presser sur S2 (*min*, sur la broche RB6 d'IC3) avec S1 toujours enfoncé permet une avance rapide des minutes, qui reboucle à 00 après 59. On règle de même les heures avec S3 (*hrs*, sur la broche RB5 d'IC3). Dès qu'on relâche S1, le réglage est terminé et les secondes (non visibles) sont remises à zéro. Avec S4, on peut couper la batterie, l'horloge continuant à fonctionner. Voilà, c'est tout.

L'horloge affiche l'heure au format « 24 heures ». Au besoin, on peut modifier le code pour passer au format « 12 heures ».

Quand, par IC2, le microcontrôleur détecte une tension trop faible, il allume le dernier point décimal, indiquant qu'il est temps de recharger la batterie B2. Si l'on néglige cet avertissement, IC4 finit par passer en mode sauvegarde, il n'y

a plus d'affichage de l'heure, mais uniquement du texte « ---- » ce qui devrait finir par attirer l'attention.

Conclusion

Mon horloge (**fig. 3**) est installée pour l'instant dans la salle de bains, où tous les membres de la famille ont de bonnes raisons d'avoir à surveiller l'heure en semaine. Grâce au détecteur PIR et à l'affichage à la demande, deux batteries Li-Ion au format 18650 branchées en parallèle tiennent trois bons mois entre deux recharges, effectuées par branchement de quelques heures sur le secteur ou au moyen d'une batterie externe (*power bank*).

Il y a d'autres possibilités de réduire encore la consommation d'énergie. On pourrait par exemple connecter un capteur de lumière et régler par logiciel l'intensité de l'affichage en fonction de la luminosité ambiante.

Pour toutes les questions, suggestions d'améliorations et propositions, vous pouvez me joindre par courriel à l'adresse miroslav.cina@t-online.de. ◀

(180277-03 - version française : Helmut Müller)

Lien

[1] Page de l'article : www.elektormagazine.fr/180277-03

filtres crossover audio actifs simples et bon marché

à 3 voies, des 3^e et 4^e ordres

Jac Hettema (Pays-Bas)

Au fil des ans, nous avons présenté de nombreux filtres crossover actifs ; et sans exception ou presque, ils comportaient tous beaucoup de composants, souvent coûteux. Le filtre présenté ici est une exception rare puisque ses composants sont peu nombreux et « normaux », c'est-à-dire ni flanqués d'une tolérance infinitésimale ni fondus dans de l'unobtainium™.

Ce sont là les avantages, entre autres, que l'on peut tirer de l'utilisation du *filtre à variable d'état*. Malgré cela, la solution la plus couramment adoptée pour un filtre crossover audio est un filtre du 2^d ordre passe-haut (HP) ou passe-bas (LP) constitué

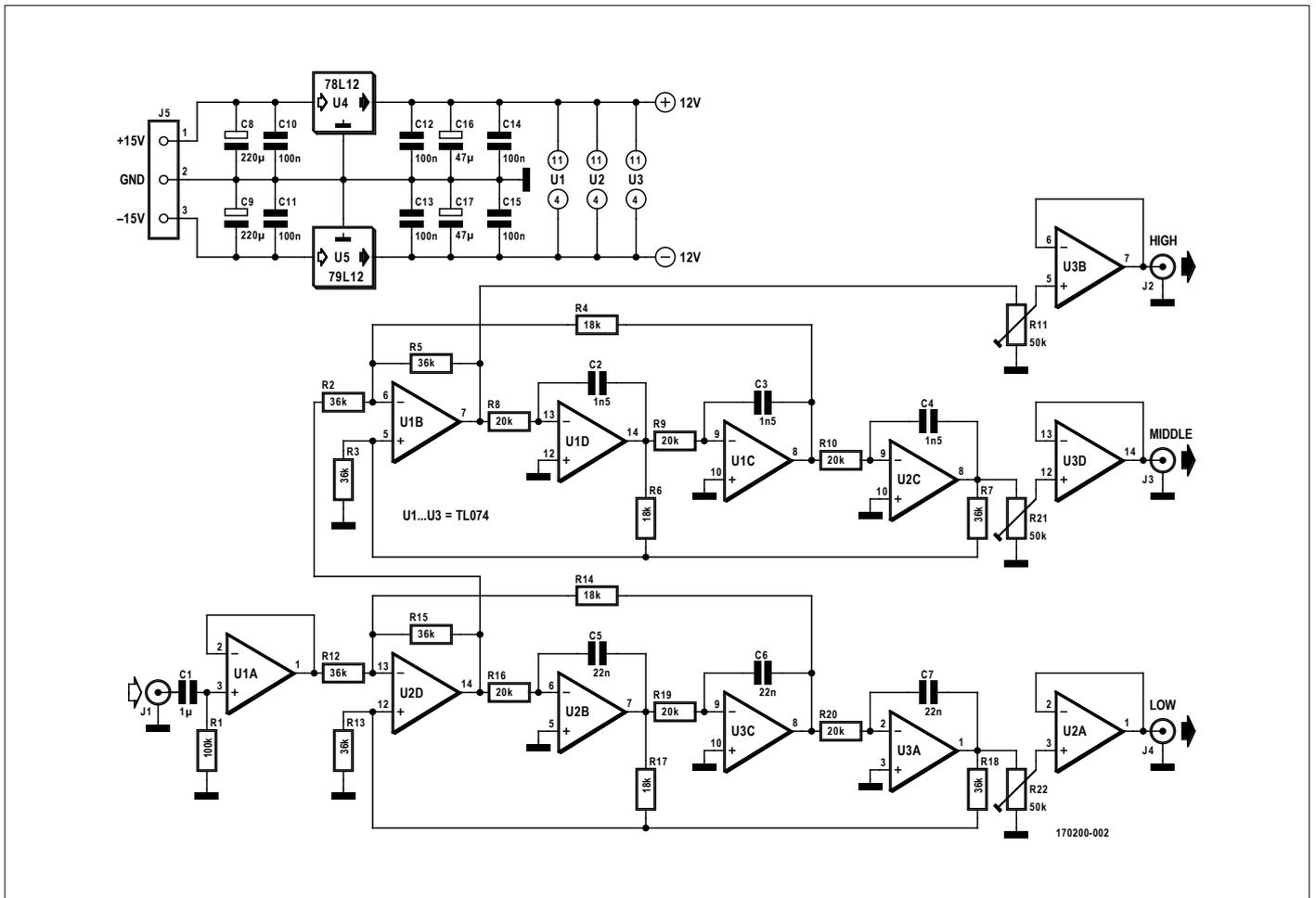


Figure 1. Schéma élémentaire d'un filtre crossover actif à trois voies du 3^e ordre.

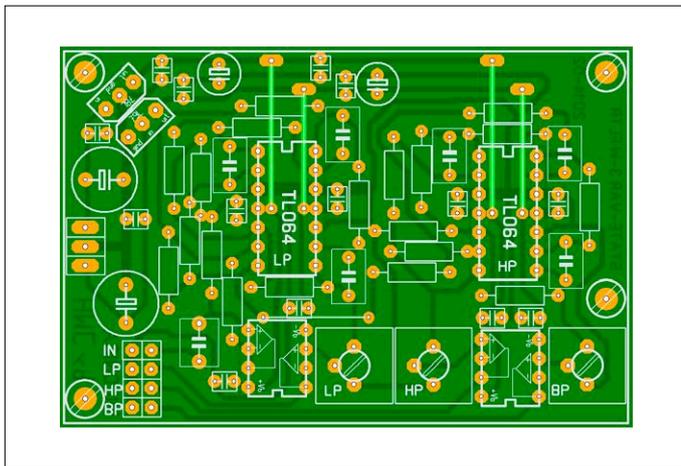


Figure 2. Circuit imprimé pour le filtre crossover à trois voies du 3^e ordre (dessin de l'auteur).

de deux résistances, deux condensateurs et d'un ampli-op ou dispositif actif similaire. Ce type de structure crée deux cellules RC dans le circuit avec leurs inévitables interactions et effets croisés.

Une excellente alternative

Le filtre à variable d'état consiste en un amplificateur sommateur suivi par une série d'intégrateurs. La sortie de chaque intégrateur est renvoyée vers l'amplificateur sommateur avec une phase et un niveau de signal adaptés.

Le nombre d'intégrateurs dans le circuit détermine l'ordre du filtre, tandis que le degré de contre-réaction détermine sa caractéristique. Des réponses de types Bessel et Butterworth sont possibles, on trouvera les calculs nécessaires dans la littérature sur le web [1] & [2] et dans le livre de Douglas Self intitulé *The Design of Active Crossovers*.

Dans le filtre à variable d'état, toutes les cellules RC sont séparées par un ampli-op « dédié » pour qu'elles ne puissent pas se perturber mutuellement. Ceci permet d'employer des condensateurs ordinaires (mais de bonne qualité) et d'élaborer des cellules RC correctes avec des résistances à 1% – largement disponibles de nos jours et beaucoup moins chères que des condensateurs à tolérance de 1%.

De plus, un unique filtre permet de séparer les hautes et basses fréquences, ce qui au global réduit le nombre de composants nécessaires. Par ailleurs, avec un filtre crossover, on peut déjà réaliser un système audio à double ampli ; avec deux filtres, on peut concevoir un filtre à trois voies.

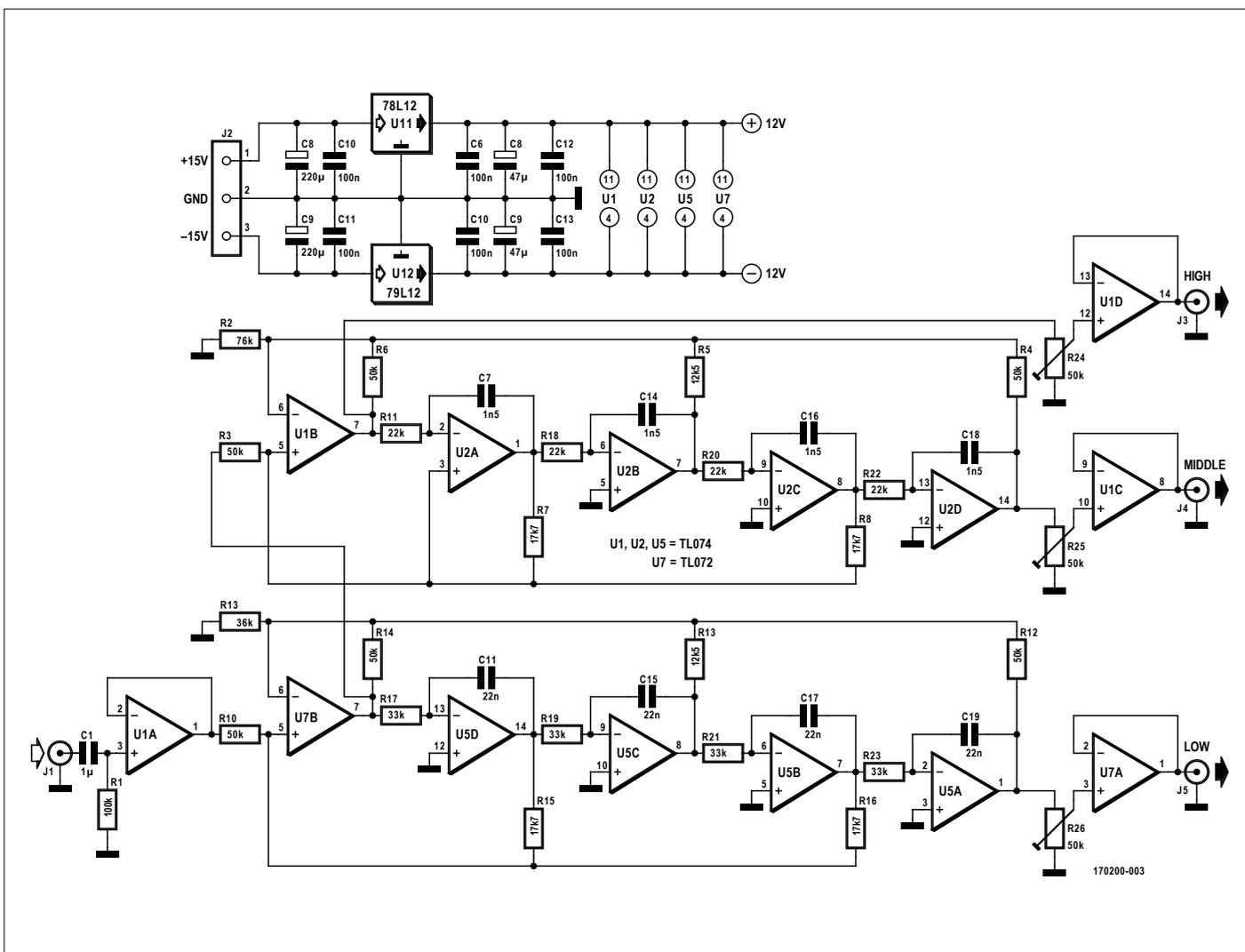


Figure 3. Schéma élémentaire d'un filtre crossover actif à trois voies du 4^e ordre.

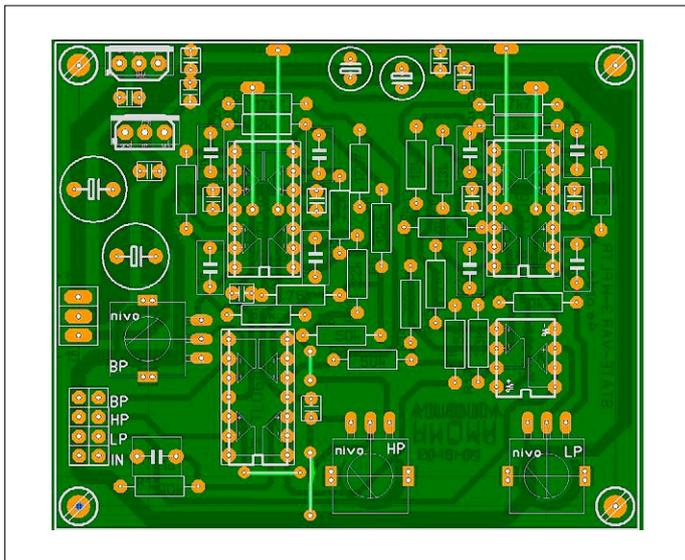


Figure 4. Circuit imprimé pour le filtre crossover à trois voies du 4^e ordre (dessin de l'auteur).

Les fréquences de coupure sont déterminées par les constantes de temps de l'intégrateur ainsi :

$$f = 1 / (2 \pi R C)$$

Il est important que les constantes de temps ($R \times C$) des cellules soient égales – on peut compenser l'effet d'un condensateur hors tolérance par l'utilisation de résistances à 1%.

Schémas et circuits imprimés

Le schéma de la **figure 1** montre un filtre crossover actif à trois voies du 3^e ordre. La **figure 2** montre le dessin du circuit imprimé pour ce circuit, le fichier de conception réalisé avec Sprint-Layout 5 est téléchargeable en [3].

Comme le montre le schéma de la **figure 3**, la variante du 4^e ordre (« plus raide ») diffère peu du filtre du 3^e ordre. Il en est de même pour le dessin du circuit imprimé à la **figure 4**. ◀

(170200-03 – version française : Denis Lafourcade)

Liens

- [1] « A Fourth-Order State Variable Filter for Linkwitz-Riley Active Crossover Designs », Dennis A. Bohn : www.rane.com/pdf/linriley.pdf
- [2] « State Variable Filters », Rod Elliott (ESP) : <http://sound.whsites.net/articles/state-variable.htm>
- [3] Page de l'article : www.elektormagazine.fr/170200-03

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

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.

novembre 2018

- ◇ **IBS + SmartCity + SmartGrid**
06 au 07/11 - Paris
www.ibs-event.com & www.smartgrid-smartcity.com
- ◇ **SATIS & SCREEN4ALL**
06 au 07/11 - Paris
www.satis-expo.com
- ◇ **AERO SPACE DAYS**
07 au 08/11 - Paris
www.asddays.com
- ◇ **Salon de la photo**
08 au 12/11 - Paris
www.lesalondelaphoto.com
- ◇ **EPOQU'AUTO**
(autos & motos anciennes)
09 au 11/11 - Lyon
www.epoquauto.com
- ◇ **Salon du Made In France**
10 au 12/11 - Paris
www.mifexpo.fr
- ◇ **Connected Week**
17 au 25/11 - Angers
<http://connectedweek-angers.fr>
- ◇ **Semaine Européenne de la Réduction des Déchets**
17 au 25/11
www.serd.ademe.fr & www.ewwr.eu
- ◇ **BE 4.0** (salon des Industries du futur)
20 au 21/11 - Mulhouse
www.industriesdulfutur.eu
- ◇ **Educatec/Educatrice**
21 au 23/11 - Paris
www.educatec-educatrice.com
- ◇ **Maker Faire Paris**
22 au 25/11 - Paris
<https://paris.makerfaire.com>
- ◇ **Assises nationales de l'éolien terrestre**
22/11 - Paris
www.assises-nationales-eolien.fr
- ◇ **Colloque IoT**
22/11 - Angers
www.eseo.fr/iotcolloque/index.html
- ◇ **JTSE** (Journées Techniques du Spectacle et de l'Événement)
27 au 28/11 - Paris



écrans couleur OLED

pilotés par STM32

Tam Hanna

Un écran couleur OLED apporte une valeur ajoutée certaine à un projet ; les puces intégrées pour sa commande sont accessibles par SPI. Les contrôleurs à 32 bits bon marché, bibliothèques avancées et environnements de développement conviviaux en facilitent grandement la mise en œuvre, comme le démontre cet article.

Le choix d'un écran monochrome OLED s'impose généralement dès qu'il s'agit pour un Arduino classique à 8 bits d'afficher des données. L'écran ne requiert que peu de mémoire, et est pilotable même avec de « petits » processeurs.

Mais le monde continue de bouger autour de la galaxie Arduino, c'est ainsi que le prix des contrôleurs à 32 bits, en particulier ceux de ST, a fortement diminué. Vous aimez la programmation de haut niveau ? Les puces ST32 sont faites pour vous, maintenant qu'elles sont disponibles à des prix comparables à ceux des gros PIC.

De plus, le développement continu du matériel a permis de réduire le coût des écrans organiques. Solomon Systech a ajouté le pilote SSD1351 au contrôleur OLED SSD1306 – une

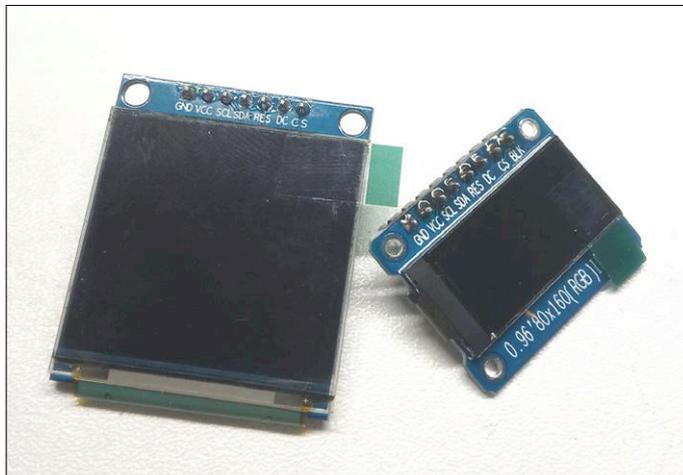
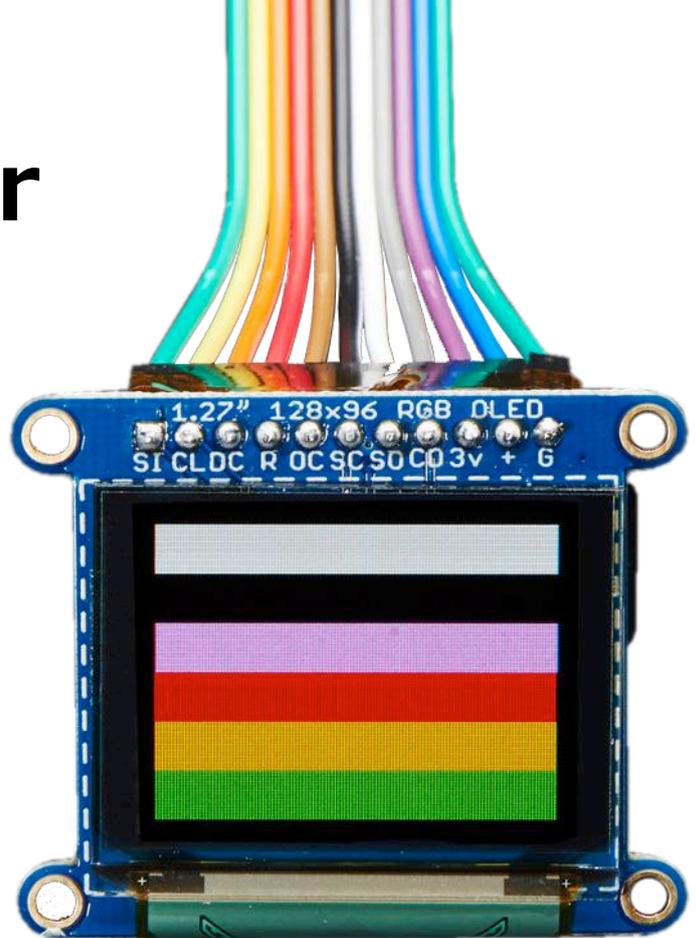


Figure 1. Disponibles à peu de frais chez AliExpress : écrans organiques en couleur (source : tam.hanna sur Instagram).



version améliorée qui peut prendre en charge des écrans couleur d'une taille allant jusqu'à 128x128 pixels.

Les sociétés de vente par correspondance chinoises ont réagi et proposent des afficheurs organiques, sous forme de cartes d'évaluation prêtes à l'emploi (voir **figure 1**). Un écran de 128x128 pixels coûte moins de 20 € ; assurez-vous de choisir un écran avec un pilote SSD1351.

Préparation des outils logiciels

D'un point de vue technique, le pilote SSD1351 (voir synoptique, **figure 2**), est similaire à son prédécesseur. Il dispose d'une interface SPI pour la communication.

La question de l'environnement de développement est des plus intéressantes. ST a racheté Atollic il y a quelques mois et considère l'environnement de développement intégré « TrueSTUDIO », autrefois extrêmement coûteux, désormais comme choix préférentiel. Il convient d'utiliser CUBE en tant que générateur de code. Il y a quelque temps, ST a signalé à l'auteur qu'au lieu d'utiliser le greffon CUBE, il vaudrait mieux, pour une meilleure stabilité, utiliser une version autonome de TrueSTUDIO ainsi qu'une version autonome de CUBE installée séparément. N'utilisez que les dernières versions des logiciels. Cet article a été créé sous Ubuntu 14.04 – veuillez impérativement à ne pas utiliser de versions obsolètes, et installez immédiatement toute mise à jour disponible.

Vous pouvez utiliser n'importe quel Nucleo comme carte d'évaluation. Nous utiliserons un STM32L476 pour les essais qui suivent : une carte bon marché et pourtant puissante, disponible dans l'e-choppe d'Elektor (voir l'encadré « @www.elektor.fr »). Les projets commencent dans CUBE ; sous Ubuntu, le logiciel peut être lancé avec la commande suivante (l'utilisation de sudo n'est requise que pour les mises à jour ; il est recommandé d'exécuter CUBE en tant qu'utilisateur normal) :

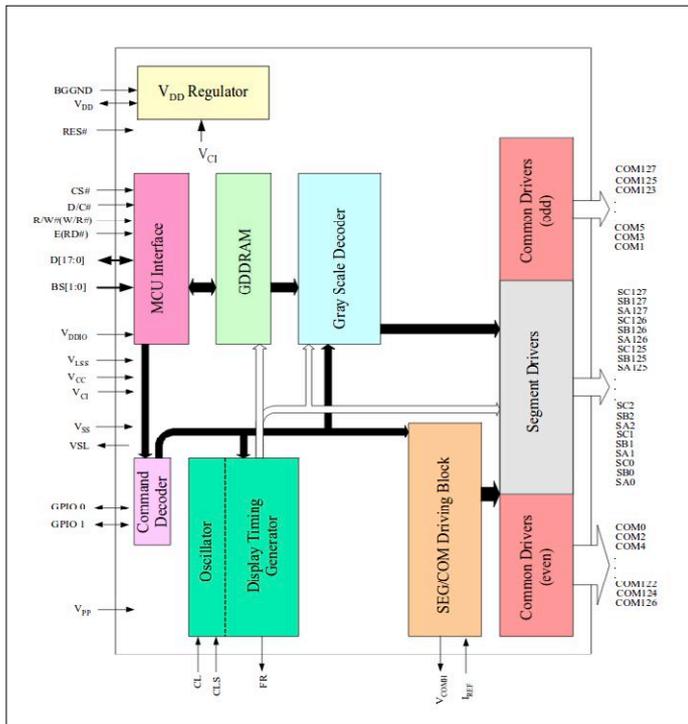


Figure 2. La structure interne du pilote SSD1351 rappelle celle de ses prédécesseurs.

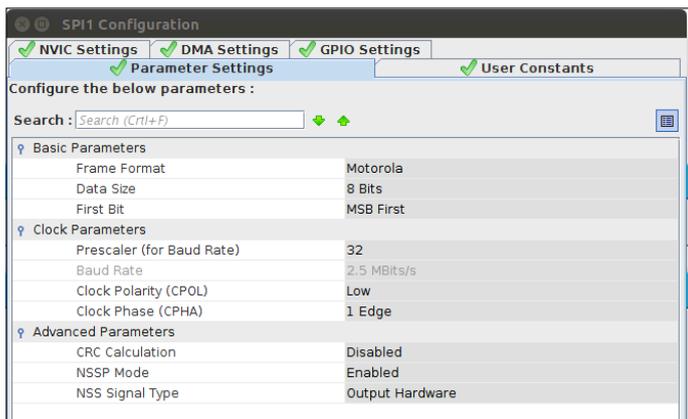


Figure 3. Ces réglages sont assez rapides pour notre afficheur.

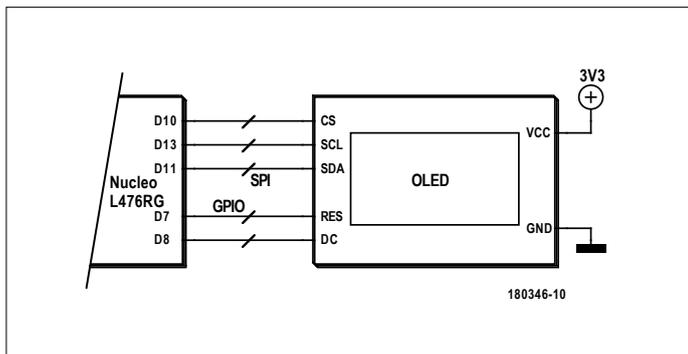


Figure 4. La connexion à l'écran couleur OLED n'est pas particulièrement difficile. Il est possible de le configurer par I2C ou SPI.

Cliquez ensuite sur [New Project](#) et accédez à la rubrique [Board Selector](#). Recherchez la chaîne « NUCLEO-L476RG » ; la saisie d'un nom de CPU provoque l'affichage d'une carte inadéquate. Confirmez l'initialisation de tous les périphériques avec leurs valeurs par défaut de sorte à disposer d'une configuration exécutable immédiatement.

Nous avons besoin d'abord d'une interface SPI chargée de la sortie des données. Sur la gauche, à la rubrique [Pinout](#), se trouve un groupe de périphériques qui peut être déployé, à l'instar d'une arborescence classique d'un programme Windows. STM équipe ses cartes Nucleo d'un groupe de connecteurs rappelant la connectique de l'Arduino. Malheureusement, le « port SPI principal » est bloqué. C'est pourquoi il faut d'abord cliquer sur la broche PB3 puis sur [Reset State](#) pour effacer la configuration. ST prévoit cette broche pour la programmation JTAG, inutile si vous travaillez avec le programmeur ST-Link.

Dans l'arborescence *Auth*, le symbole X disparaîtra alors devant la ligne SPI1. Déployez celle-ci et sélectionnez [Transmit Only Master](#) dans la boîte de sélection *mode*. Dans le champ [Hardware NSS](#), cliquez sur [Hardware NSS Output Signal](#) : nous souhaitons que le moteur SPI s'occupe de la production du signal CS. Assurez-vous ensuite que les broches ont été correctement affectées ; les périphériques STM sont très flexibles et peuvent être l'objet d'affectations erronées. C'est fâcheux, en particulier parce que dans ce cas vous ne pourriez pas accéder aux signaux sur les broches spécifiées par le standard Arduino. La broche PA5 est affectée à la LED verte, il faut la redéfinir si vous voulez utiliser le brochage de l'Arduino.

À l'étape suivante, nous avons besoin de deux broches GPIO classiques : d'abord, *Reset* pour gérer le contrôleur dans son ensemble, puis *DC* pour connecter l'émetteur-récepteur SPI à la mémoire de l'écran ou à la mémoire de paramètres.

Dans CUBE, la broche Arduino D8 correspond à la broche PA9 : cliquez dessus, et sélectionnez l'option [GPIO_Output](#). La broche D7 correspond à PA8 : là aussi il faut un clic droit, puis choisissez [GPIO_Output](#). À l'étape suivante, cliquez sur l'onglet [Configuration](#), puis sur le bouton [GPIO-Pin](#) dans la rubrique [System](#) et activez l'onglet [GPIO](#). Dans le champ [User Label](#), vous pouvez alors attribuer des noms conviviaux aux broches, noms sous lesquels celles-ci seront plus tard accessibles dans votre code.

Par défaut, ST configure de manière « agressive » le moteur SPI de la puce. Dans l'onglet [Configuration](#), cliquez sur SPI1 et définissez les paramètres comme indiqué à la **figure 3**. Cliquez sur [Project Generate Code](#) pour produire le code. Sous [Project Name](#), choisissez « ElektorOLED » ; dans la rubrique [Project Location](#), vous devez sélectionner le répertoire de travail de l'installation Atollic. Sur le poste de travail de l'auteur, il se trouve sous `/home/tamhan/Atollic/TrueSTUDIO/STM32_workspace_9.0/`. Dans le champ [Toolchain / DIE](#), sélectionnez [TrueSTUDIO](#). Après avoir cliqué sur OK, le programme télécharge environ 700 Mo de code source puis crée le projet.

Et c'est parti !

Notre prochaine étape est la construction du circuit illustré à la **figure 4**. Comme d'habitude chez Elektor, le code est téléchargeable gratuitement depuis la page de l'article [1].

Démarrez ensuite TrueSTUDIO et cliquez sur [File Open Projects from File System](#). Accédez au projet et chargez-le comme d'habitude. Le fichier `.ioc` permet de « recharger » le projet dans CUBE, une tâche qui ne nous intéresse pas pour le moment. Avant de commencer la programmation proprement dite, consultez la fiche technique du contrôleur. Solomon Systech est assez flou à ce sujet ; vous en trouverez une version sous [2], actualisée à la mise sous presse de ce numéro. Même s'il est compliqué à manipuler, CUBE reste l'un des meilleurs générateurs de code dans le monde de l'embarqué. La « ré-écriture » de projets après modifications dans le fichier `.ioc` se déroule sans problème dans la pratique, dès lors que vous placez votre propre code uniquement dans les secteurs du code utilisateur. À propos des points d'entrée, deux sont disponibles, ils ressemblent à ceci :

```
int main(void) {
    . . .
    /* Infinite loop */
    /* USER CODE BEGIN WHILE */
    while (1)
    {
        /* USER CODE END WHILE */

        /* USER CODE BEGIN 3 */
    }
    /* USER CODE END 3 */
}
```

ST laisse la boucle `while`, pour son début et sa fin, sous l'entière responsabilité du développeur. Puis nous enchaînons par « l'initialisation ». Il s'avère pertinent d'effectuer une réinitialisation complète après le premier démarrage de l'affichage. À cet effet, la broche `Reset` est mise au niveau bas, nous attendons 500 ms, et laissons ensuite 500 ms à l'écran pour s'initialiser avec succès :

```
/* Infinite loop */
/* USER CODE BEGIN WHILE */

HAL_GPIO_WritePin(DISPLAY_RESET_GPIO_Port, DISPLAY_
    RESET_Pin, GPIO_PIN_SET);
HAL_Delay(500);
HAL_GPIO_WritePin(DISPLAY_RESET_GPIO_Port, DISPLAY_
    RESET_Pin, GPIO_PIN_RESET);
HAL_Delay(500);
HAL_GPIO_WritePin(DISPLAY_RESET_GPIO_Port, DISPLAY_
    RESET_Pin, GPIO_PIN_SET);
HAL_Delay(500);
```

Nous avons besoin ensuite de deux fonctions, chacune écrivant un octet dans le contrôleur d'écran. Dans un premier temps, nous utilisons l'API `HAL_GPIO_WritePin`, qui permet d'interagir avec les GPIO sous CUBE. Les valeurs `DISPLAY_DC_GPIO_Port` et `DISPLAY_DC_Pin` sont toutes les deux des constantes résultant de noms conviviaux saisis lors de l'assignation des broches :

```
/* USER CODE BEGIN 0 */

void writeCommand(char _q) {
    HAL_GPIO_WritePin(DISPLAY_DC_GPIO_Port, DISPLAY_DC_
```

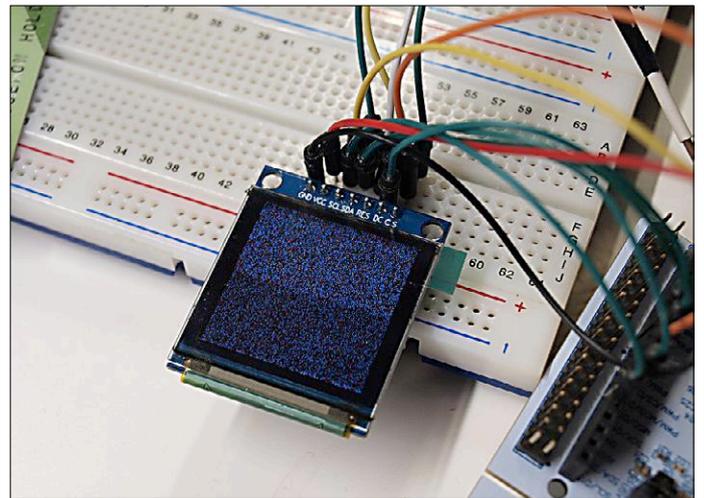


Figure 5. Le contenu de l'écran est aléatoire, car la mémoire n'est pas initialisée.

```
Pin, GPIO_PIN_RESET);
HAL_SPI_Transmit(&hspl1,&q,1,100);
}
```

```
void writeData(char _q) {
    HAL_GPIO_WritePin(DISPLAY_DC_GPIO_Port, DISPLAY_DC_
        Pin, GPIO_PIN_SET);
    HAL_SPI_Transmit(&hspl1,&q,1,100);
}
/* USER CODE END 0 */
```

La transmission des données est effectuée avec `HAL_SPI_Transmit`. La fonction, qui est aussi un modèle établi dans le monde ST, fait référence à un objet de port créé par l'une des méthodes de CUBE. Celle-ci est suivie d'une référence à un champ de caractères, de la quantité de données à transférer, et d'un délai maximal d'attente (qui ne joue pas ici un rôle majeur). Les fonctions créées par le développeur doivent se trouver dans la zone `USER CODE BEGIN 0`.

Configuration pas-à-pas

Si vous prévoyez de mettre en œuvre une puce complexe, il est recommandé dans un premier temps de rechercher une implémentation de référence. La société Adafruit, qui fournit un pilote prêt à l'emploi sous [3], est à ce titre une source particulièrement appréciée.

On y trouve entre autres un groupe de constantes qui décrivent les différents registres de la puce, transférez-le dans `main.h`. L'initialisation proprement dite commence à l'étape suivante :

```
int main(void)
{
    . . .
    HAL_Delay(500);

    writeCommand(SSD1351_CMD_COMMANDLOCK);
    // set command lock
    writeData(0x12);
    writeCommand(SSD1351_CMD_COMMANDLOCK);
    // set command lock
    writeData(0xB1);
```

Write Data			Data bus																		
Bus width	Color Depth	Input order	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	
8 bits/Serial	65k	1st	X	X	X	X	X	X	X	X	X	X	X	C ₄	C ₃	C ₂	C ₁	C ₀	B ₅	B ₄	B ₃
		2nd	X	X	X	X	X	X	X	X	X	X	X	B ₂	B ₁	B ₀	A ₄	A ₃	A ₂	A ₁	A ₀
8 bits/Serial	262k	1st	X	X	X	X	X	X	X	X	X	X	X	X	C ₅	C ₄	C ₃	C ₂	C ₁	C ₀	
		2nd	X	X	X	X	X	X	X	X	X	X	X	X	B ₅	B ₄	B ₃	B ₂	B ₁	B ₀	
		3rd	X	X	X	X	X	X	X	X	X	X	X	X	X	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀
16 bits	65k		X	X	C ₄	C ₃	C ₂	C ₁	C ₀	B ₅	B ₄	B ₃	B ₂	B ₁	B ₀	A ₄	A ₃	A ₂	A ₁	A ₀	
16 bits	262k format 1	1st	X	X	X	X	X	X	X	X	X	X	X	X	C ₅	C ₄	C ₃	C ₂	C ₁	C ₀	
		2nd	X	X	X	X	B ₅	B ₄	B ₃	B ₂	B ₁	B ₀	X	X	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀	
16 bits	262k format 2	1st	X	X	X	X	C ₁₅	C ₁₄	C ₁₃	C ₁₂	C ₁₁	C ₁₀	X	X	B ₁₅	B ₁₄	B ₁₃	B ₁₂	B ₁₁	B ₁₀	
		2nd	X	X	X	X	A ₁₅	A ₁₄	A ₁₃	A ₁₂	A ₁₁	A ₁₀	X	X	C ₂₅	C ₂₄	C ₂₃	C ₂₂	C ₂₁	C ₂₀	
		3rd	X	X	X	X	B ₂₅	B ₂₄	B ₂₃	B ₂₂	B ₂₁	B ₂₀	X	X	A ₂₅	A ₂₄	A ₂₃	A ₂₂	A ₂₁	A ₂₀	
18 bits	262k		C ₅	C ₄	C ₃	C ₂	C ₁	C ₀	B ₅	B ₄	B ₃	B ₂	B ₁	B ₀	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀	

Figure 6. Le SSD1351 est très flexible. Ce tableau montre les différentes combinaisons « largeur de bus et profondeur de couleur » possibles.

```

writeCommand(SSD1351_CMD_DISPLAYOFF); // 0xAE
...
writeCommand(SSD1351_CMD_PRECHARGE2);
writeData(0x01);
writeCommand(SSD1351_CMD_DISPLAYON);

writeData(0xFF);
}
HAL_Delay(10);
/* USER CODE END WHILE */
/* USER CODE BEGIN 3 */
}

```

Cet extrait de code définit un ensemble de paramètres pour agir sur la configuration de la puce. Finalement, il envoie la commande `SSD1351_CMD_DISPLAYON`, qui amène le contrôleur à activer l'afficheur. De telles commandes sont utiles, car elles permettent, entre autres, d'allumer et d'éteindre l'écran sans transistors ou autre attention particulière.

Vous pouvez à présent exécuter le programme. La **figure 5** montre le résultat sur l'écran.

Et maintenant, on passe à la couleur !

Certes l'affichage de motifs « aléatoires » est amusant, mais en pratique, cela n'a pas beaucoup d'intérêt. Il est préférable d'afficher sur l'écran des informations pertinentes. Selon sa fiche technique, le SSD1351 connaît deux modes de résolution pour la couleur : *262k color* procure une meilleure résolution des couleurs, mais requiert 18 bits par pixel (6:6:6) alors que *65k color* fonctionne en (5:6:5) avec seulement 16 bits par pixel. En outre, le contrôleur connaît une bonne douzaine de formats différents qui peuvent être sélectionnés via les registres. Dans les étapes suivantes, nous travaillerons avec le type *16bits/65k*, le tableau de la **figure 6** répertorie d'autres combinaisons (largeur de bus et profondeur de couleur) possibles.

Commençons par l'affichage d'une constante :

```

while (1) {
writeCommand(SSD1351_CMD_SETCOLUMN);
writeData(0x00);
writeData(0x7F);
writeCommand(SSD1351_CMD_SETROW);
writeData(0x00);
writeData(0x7F);
writeCommand(SSD1351_CMD_WRITERAM);

for (uint16_t i=0; i < 128*128; i++) {
writeData(0x00);

```

Ce code n'est certainement pas des plus efficaces ; il parcourt la totalité de la mémoire d'image et produit deux pixels pour chaque valeur lue. En pratique, nous pourrions nous affranchir de la commutation de la broche CS, mais à ce stade nous renonçons à l'effort de modifier le programme. Après exécution du programme, l'écran affiche une couleur unie.

Ensuite, nous devons créer la mémoire d'écran réelle (les Anglo-saxons préfèrent le terme de *framebuffer*). Les ressources en mémoire de notre contrôleur n'étant pas illimitées, nous créons dans un premier temps une zone mémoire dédiée :

```

#pragma pack(2)
typedef struct {
unsigned int rField : 5;
unsigned int gField : 6;
unsigned int bField : 5;
} colorquaine;

```

L'expérience montre que l'exploitation directe du contenu des zones binaires ne fonctionne pas toujours. Le recours à l'arithmétique binaire est une solution plus rapide – et pas trop fastidieuse si l'écran n'est pas rafraîchi fréquemment :

```

char makeLower(colorquaine _x){
char work = _x.rField<<3;
char temp = _x.gField>>3;
return work+temp;
}
char makeHigher(colorquaine _x){
char work= (_x.gField & 0b111)<<5;
return work + _x.bField;
}

```

Les fonctions `makeLower` et `makeHigher` ont pour but de convertir les trois valeurs de couleur en un format compatible avec

le contrôleur. Analysons `makeHigher` : dans un premier temps, nous appliquons un masque sur la valeur de `.gField` afin d'isoler les bits significatifs (à savoir les trois derniers bits). Ensuite on décale ces bits vers la droite et on leur ajoute la valeur de la couleur bleue.

Il ne reste plus qu'à afficher les informations :

```
for (uint16_t i=0; i < 128*128; i++) {
    writeData(makeLower(myDisplayStorage[i]));
    writeData(makeHigher(myDisplayStorage[i]));
}
```

À ce stade, vous pouvez faire à peu près tout ce que vous voulez : utilisez votre algorithme graphique préféré et confiez-le au `framebuffer`. Si vous souhaitez économiser de la mémoire, vous pouvez aussi utiliser l'affichage couleur palettisé.

Consommation...

Dès son introduction sur le marché des appareils mobiles, l'OLED a été considéré comme le meilleur élève en termes de basse consommation. Il est donc logique de mesurer cette consommation.

Deux options se présentent : utiliser le cavalier décrit par Dogan Ibrahim dans son livre (en anglais) sur le STM32 (voir l'encadré « @www.elektor.fr ») ou monter un multimètre. Amateur de solutions radicales, l'auteur a opté pour cette dernière et

Cartes d'interface coûteuses !

Vous souhaitez vous lancer dans une grande série ? Alors songez à développer vous-mêmes les cartes-filles, car les sociétés de vente par correspondance s'octroient là des marges juteuses.

a utilisé un Solartron 7150. Comme le montre le **tableau 1**, la consommation de l'OLED dépend du contexte d'utilisation.

Conclusion

En alimentant l'afficheur en 3,3 V, celui-ci reste pâlichon : il est donc impossible d'utiliser un modèle chinois à l'extérieur. En revanche il peut trouver sa place à l'intérieur.

À l'époque des assistants numériques personnels (PDA), les utilisateurs estimaient déjà que les modèles dotés d'un écran en couleur étaient plus séduisants que leurs homologues monochromes. Même si vous travaillez habituellement avec des écrans haut de gamme (chers), n'oubliez pas de jeter un œil aux nouveaux venus meilleur marché. ◀

(180346-03 – version française : Xavier Pfaff)

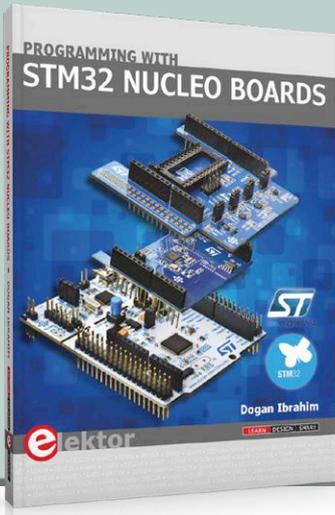
Tableau 1 : consommation

Couleur affichée, état	Consommation
<code>myDisplayStorage[i].rField=0;</code> <code>myDisplayStorage[i].gField=0;</code> <code>myDisplayStorage[i].bField=0;</code>	4,2 mA
<code>myDisplayStorage[i].rField=0b11111;</code> <code>myDisplayStorage[i].gField=0;</code> <code>myDisplayStorage[i].bField=0;</code>	67 mA
<code>myDisplayStorage[i].rField=0;</code> <code>myDisplayStorage[i].gField=0b111111;</code> <code>myDisplayStorage[i].bField=0;</code>	57 mA
<code>myDisplayStorage[i].rField=0;</code> <code>myDisplayStorage[i].gField=0;</code> <code>myDisplayStorage[i].bField=0b111111;</code>	102 mA
<code>myDisplayStorage[i].rField=0b111111;</code> <code>myDisplayStorage[i].gField=0b111111;</code> <code>myDisplayStorage[i].bField=0b111111;</code>	112 mA
DISPLAYOFF	1,3 mA

@WWW.ELEKTOR.FR

→ Carte STM32 Nucleo L476RG :
www.elektor.fr/stm32-nucleo-l476rg-board

→ Livre (en anglais) « Programming with STM32 Nucleo Boards » + carte STM32 Nucleo L476RG offerte :
www.elektor.fr/programming-with-stm32-nucleo-boards



Liens

- [1] Page de l'article : www.elektormagazine.fr/180346-03
- [2] Fiche technique du pilote d'OLED SSD1351 : www.newhavendisplay.com/app_notes/SSD1351.pdf
- [3] Configuration OLED : https://github.com/adafruit/Adafruit-SSD1351-library/blob/master/Adafruit_SSD1351.h

référence de tension à « zener enfouie » LTZ1000 drôle de composant n°39

Neil Gruending (Canada)

Les références de tension à barrière de potentiel, précises et faciles à utiliser, sont souvent une solution de choix. Mais quand la stabilité de la référence dans le temps et en température est plus importante que l'exactitude de la tension de sortie, plus vraiment. La solution se trouve alors dans les entrailles du célèbre voltmètre HP 3458A : la référence de tension LTZ1000 de Linear Technology. Examinons de plus près cette référence unique à « zener enfouie ».

La LTZ1000 (**fig. 1** [1]) est un composant étonnamment simple qui ne contient que le strict minimum de composants analogiques pour créer la référence. La diode zener de référence est reliée à la broche 3 et Q1 s'occupe de sa compensation en température. Q2 sert de capteur de température pour le filament de chauffage relié aux broches 1 et 2. Comme nous le savons tous, la tension aux bornes d'une diode zener change avec la température, c'est pourquoi un circuit de régulation externe sert à pour maintenir la température du filament un peu au-dessus de la température ambiante. Pour améliorer encore plus la stabilité, les concepteurs ont utilisé un boîtier régulé en température et ils se sont assurés que celle-ci soit uniforme partout dans le boîtier.

Le résultat est une référence de 7 V ultra-stable à propos de laquelle le concepteur Carl Nelson a dit : « Le but est de réaliser un composant tellement bon qu'on ne peut pas le mesurer » [2]. Il parlait du bruit et de la dérive. En fait, seules une pile étalon ou une référence à jonction Josephson ont une meilleure stabilité ; c'est pourquoi cette référence est utilisée dans les voltmètres haut de gamme tels que le HP 3458A et les étalons de transfert de tension comme le Fluke 7001.

Toutefois, cette performance a un prix : les effets thermiques pouvant affecter la LTZ1000 sont partout. Par exemple, les broches en Kovar™ forment avec le cuivre d'un circuit imprimé une jonction de type thermocouple qui peut produire des différences de potentiel allant jusqu'à 35 $\mu\text{V}/^\circ\text{C}$. Il est donc impératif de maintenir toutes les broches à la même température. Il faut même prendre en compte le type de résistances utilisées dans le circuit externe, car celles-ci peuvent également créer des thermocouples indésirables. Voilà pourquoi on utilise typiquement des résistances bobinées ou des résistances spéciales

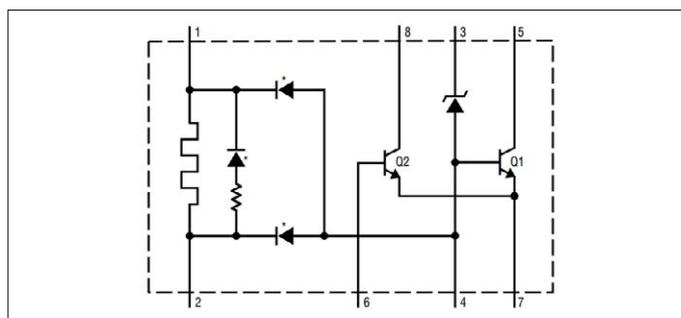


Figure 1. Schéma interne de la référence de tension LTZ1000.

à haute stabilité scellées hermétiquement.

Pour celles et ceux qui veulent expérimenter, la LTZ1000 est toujours fabriquée et vous pourrez aussi mettre la main sur des références de tension d'un HP 3458 d'occasion (**fig. 2**) [3]. Beaucoup ont passé de nombreuses heures à construire et tester divers circuits [4], mais le vrai défi reste de les caractériser ! ◀

(180464-03 – version française : Kévin Petit)

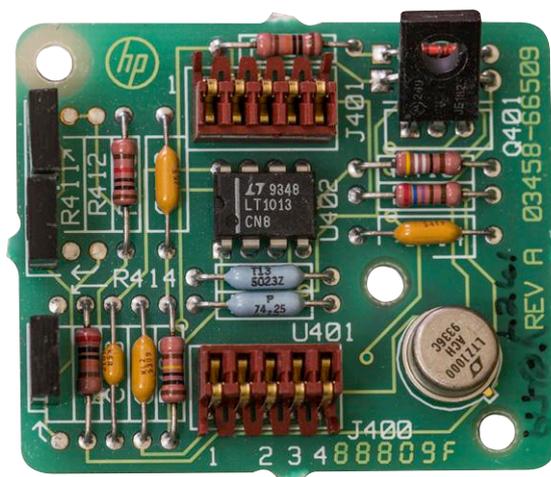


Figure 2. Référence de tension d'un voltmètre HP 3458.

Liens

- [1] Schéma de la référence LTZ1000 : www.analog.com/media/en/technical-documentation/data-sheets/1000afe.pdf
- [2] Interview de Carl Nelson : www.electronicdesign.com/analog/interview-analog-guru-carl-nelson
- [3] Référence de tension d'un voltmètre HP 3458 : https://xdevs.com/doc/HP_Agilent_Keysight/3458A/img/a9_top.jpg
- [4] Circuits et tests de la référence LTZ1000 : www.eevblog.com/forum/projects/

détecteur de court-circuit avec source de courant et sonde à effet Hall



Josef Tausch (Allemagne)

L'appareil localise un court-circuit en injectant des impulsions de courant de test dans la ligne suspecte et en détectant le champ magnétique produit.

Quand il y a un court-circuit dans un circuit ou un câble, il n'est pas si facile de le localiser. L'idée la plus évidente est de suivre à la trace le champ magnétique produit par le courant de court-circuit. Seulement, on peut oublier cette idée si, surchargée par le court-circuit, la source de courant est momentanément

ou définitivement hors service et que plus aucun courant ne circule. Un appareil de test doit donc être en mesure non seulement de détecter un champ magnétique et d'afficher le signal, mais aussi de produire son propre courant de test qui doit parcourir le conducteur de manière stable, même en cas de court-circuit.

Le détecteur de court-circuit travaille selon deux modes de fonctionnement : en courant continu et en courant pulsé. Dans le mode courant continu, on peut déjà détecter un champ magnétique par un petit déplacement du capteur. En mode courant pulsé, le champ magnétique produit ne peut plus être confondu

avec d'autres champs (parasites) comme le champ magnétique terrestre. L'appareil de contrôle se compose donc de deux parties indépendantes dans des boîtiers distincts : une source de courant qui fait circuler le courant dans la ligne en court-circuit et une sonde qui détecte le champ magnétique et révèle son existence de manière visuelle et acoustique. Chacune des deux parties dispose de sa propre alimentation constituée de batteries lithium-ion et équipée d'une puce de gestion de leur recharge. À condition qu'il s'agisse d'un « véritable » court-circuit, donc à faible résistance, l'appareil permet de localiser un court-circuit sans avoir à démonter les composants. La sonde permet non seulement de localiser des courts-circuits, mais aussi de mettre en évidence des champs magnétiques quelconques, comme ceux de relais, d'électroaimants et d'autres composants du même type.

Ordres de grandeur

Avant de pouvoir se lancer dans la conception concrète de l'électronique, on doit savoir à quelles valeurs du champ magnétique on a affaire. Le problème se situe du côté des sondes à effet Hall, dont la sensibilité est de l'ordre de 10 à 100 mV/mT.

L'intensité d'un champ magnétique H est mesurée par son induction magnétique B exprimée en Tesla (T). Pour une idée des ordres de grandeur, voici quelques exemples : le champ magnétique terrestre est, sous nos latitudes (encore) de 50 μ T environ, la valeur supérieure des champs parasites créés par le secteur 50 Hz est de l'ordre de 100 μ T.

On peut comparer ces valeurs avec l'induction magnétique B créée par un courant d'intensité I dans un conducteur rectiligne :

$$B = \mu_0 \times H = \mu_0 \times I / 2 \pi r$$

avec μ_0 , la perméabilité magnétique et r , la distance au conducteur.

Le calcul montre qu'un courant de 250 mA crée un champ dont l'intensité ne dépasse pas 16,7 μ T à une distance de 3 mm. L'intensité du champ magnétique terrestre est donc trois fois plus élevée, sans parler des champs parasites à 50 Hz ! Il importe donc d'utiliser une sonde aussi sensible que possible et de la rapprocher du conducteur aussi près que possible. En même temps, il faut que

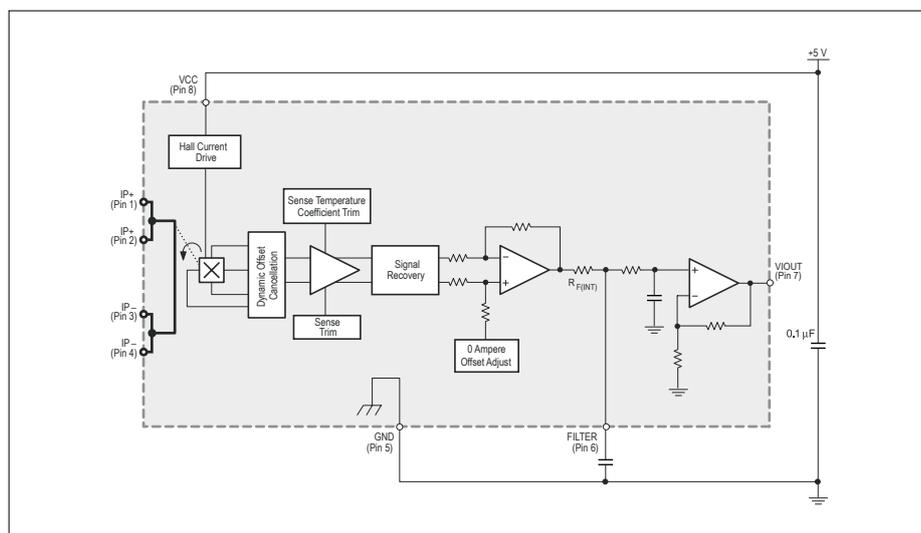


Figure 1. La puce de capteur à effet Hall amputée (source : Allegro MicroSystems).

le courant de test soit aussi élevé que possible, à la limite de ce qui est acceptable pour un appareil alimenté par batterie. Ce n'est qu'ainsi que le signal utile ne sera pas complètement noyé dans le « bruit ».

La partie détection

La première tâche lors de la conception fut donc de choisir une sonde à effet Hall adéquate. L'auteur a testé un certain nombre de ces sondes et s'est finalement arrêté à l'ACS712-05 de Allegro Microsystems [1]. Comme le montre la **figure 1**, cette puce possède huit broches parmi lesquelles les broches 1-2 et 3-4 sont réservées au conducteur parcouru par le courant, de sorte que le champ magnétique est produit directement à l'intérieur de la puce. Cette configuration n'est pas adéquate pour notre appareil, étant donné que nous ne voulons pas ouvrir la ligne en court-circuit. Pour que la sonde puisse être mise au plus près du conducteur, l'auteur a même coupé et limé les broches inutiles !

Le même fabricant propose des sondes à effet Hall en boîtier SIP3, comme la A1389LUA-9T, avec une sensibilité relativement élevée de 90 mV/mT. Cela évite d'avoir à limer les broches inutiles de la puce.

Finalement, la **figure 2** montre le schéma complet, en deux parties, de l'appareil, avec le détecteur en haut et la source de courant en bas.

La tension de sortie de la sonde Allegro Microsystems, sur la broche 7, vaut +2,5 V en l'absence de champ magnétique et augmente ou diminue selon le

sens du courant. Comme la variation de tension (lissée par R1 et C7) n'est que de quelques millivolts (dans notre exemple de 250 mA à une distance de 3 mm), elle doit être fortement amplifiée, puis comparée à une tension de seuil réglable par P1 (sensibilité). Cette tâche est prise en charge par le double comparateur classique LM393 (IC2), qui délivre en sortie de IC2.B une impulsion négative qui allume une LED et déclenche aussi un signal acoustique, pour qu'il ne soit pas nécessaire, au cours de la recherche, d'avoir constamment un œil sur la LED. Cet indicateur acoustique est constitué d'un simple circuit temporisateur 555 (IC3) et d'un traducteur piézoacoustique passif. La fréquence est réglable par P2. L'indication optique et acoustique existe aussi sur la source de courant, avec un rôle très important. Elle assure qu'en mode pulsé, on a bien affaire à un champ magnétique effectif et non à un quelconque signal parasite. Les LED et les vibreurs doivent alors se manifester simultanément ou en alternance. Le dernier cas indique un sens du courant inversé dans le conducteur de retour.

La partie sonde est alimentée en 5 V. Comme la batterie Li-Ion ne fournit que 3,7 V, un convertisseur élévateur de tension MCP1640 de Microchip [2] fournit cette tension d'alimentation, fixée par le diviseur R16 à R18 selon :

$$V_{OUT} = [(R16/R17) || (R18) + 1] * 1,21 \text{ V}$$

Il existe d'ailleurs plusieurs variantes de ce convertisseur, n'importe laquelle fera l'affaire.

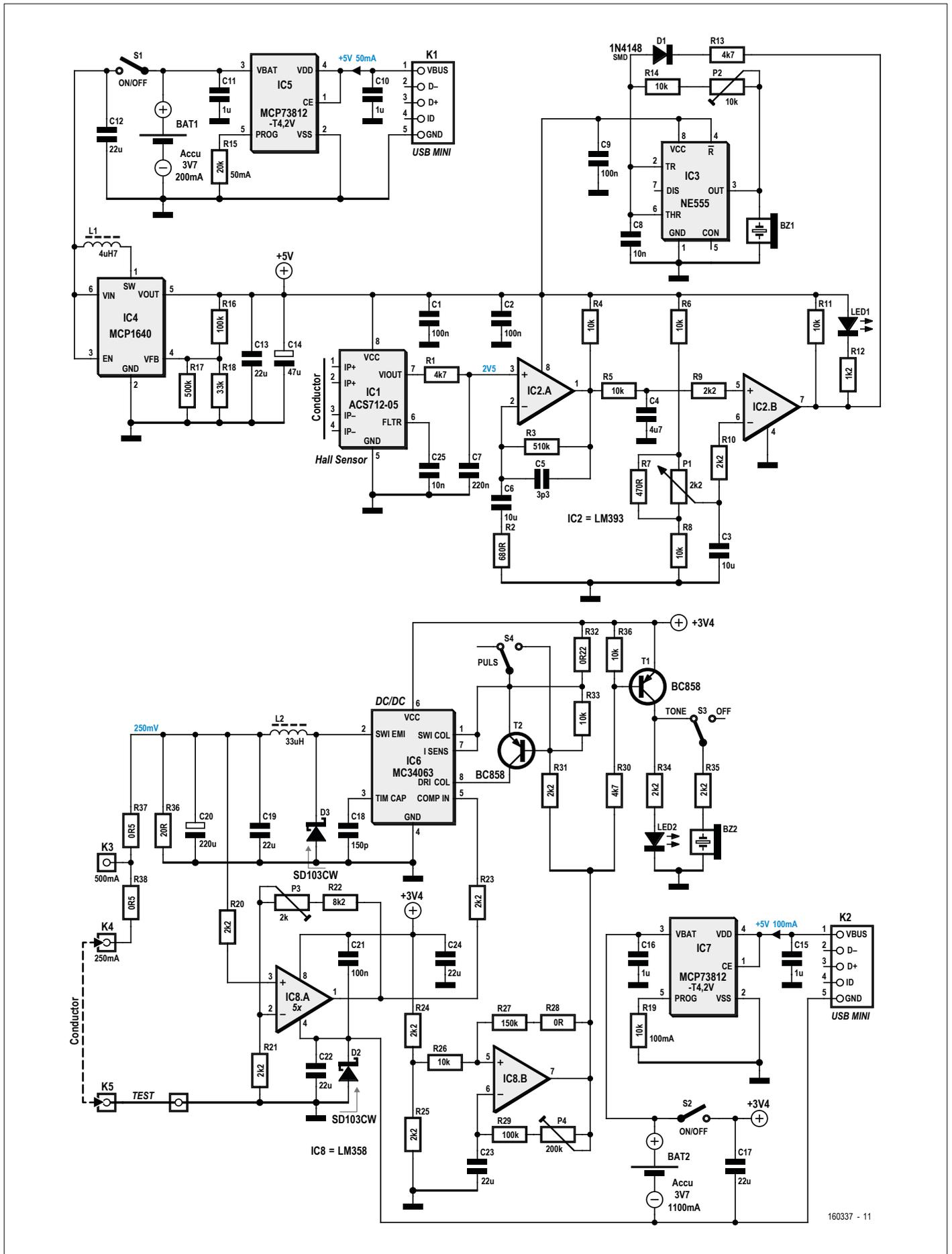


Figure 2. Les schémas du détecteur de court-circuit (en haut) et de la source de courant (en bas).

La partie détection est conçue comme une sonde de mesure et montée dans son propre boîtier constitué de profilés en L en plastique du commerce (25 mm × 25 mm). La sonde à effet Hall se trouve à la pointe, qui peut être promenée le long du conducteur. Cette partie est alimentée par un accumulateur rechargeable par l'intermédiaire d'une prise mini-USB.

La source de courant

Dans l'exemple ci-dessus, le courant à travers le conducteur n'a pas été fixé à 250 mA tout à fait par hasard. Il semble que cette valeur soit un bon compromis entre la valeur du champ magnétique au niveau de la sonde et l'intensité du courant qu'on peut raisonnablement espérer d'une source alimentée par batterie (ou tolérer dans le conducteur). Si la sonde s'avère incapable de détecter un signal significatif, on aura deux possibilités : élever le courant à 500 mA ou/et passer du mode continu au mode pulsé. Le mode pulsé doit surtout permettre d'éviter la confusion avec des champs magnétiques parasites.

Voyons comment ces tâches sont accomplies par l'électronique. L'énergie est fournie par une batterie Li-Ion d'une tension de 3,7 V qui alimente un convertisseur de tension continue (IC6) du type MC34063 [3]. Ici, cette puce est configurée en abaisseur de tension, dont la sortie est réglée à 250 mV.

Il n'en faut pas plus, cette tension n'étant pas déterminante

pour la valeur du champ magnétique. Normalement, la tension de sortie est fixée par le rapport de deux résistances entre l'entrée du comparateur interne et la tension de sortie et la masse. Ici, notre intention est différente, alors la manière dont les choses se passent est un peu particulière.

L'ampli op IC8.A, un LM358, amplifie la sortie du convertisseur de 250 mV à 1,25 V et transmet cette tension au comparateur interne du convertisseur à travers la résistance R23. Cette tension correspond à la référence interne de 1,25 V, de sorte que cette puce stabilise cette tension de sortie. La deuxième moitié de IC8 est montée en oscillateur, dont la fréquence, de l'ordre de 1 Hz, est réglable par P4. La sortie de l'ampli op est connectée à deux transistors de commutation PNP. T1 pilote à la fréquence de 1 Hz une LED et un vibreur

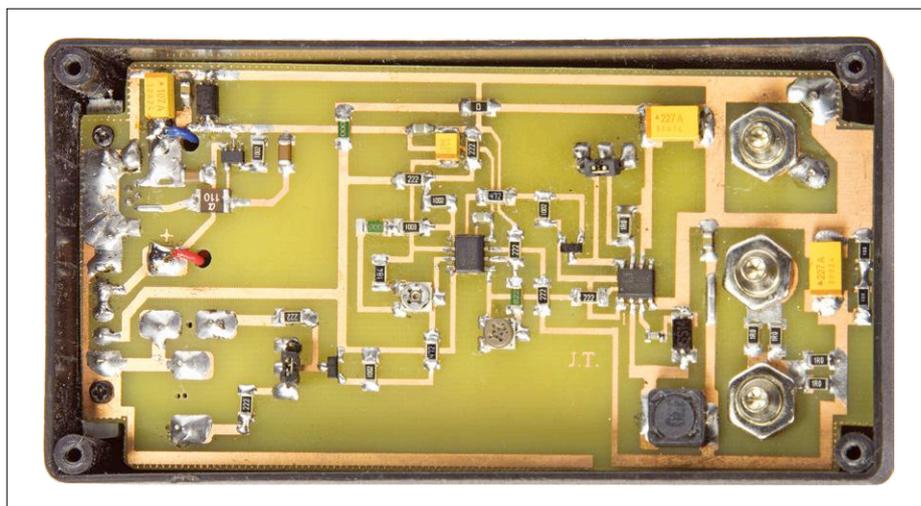


Figure 3. La source de courant est installée dans un boîtier standard du commerce, où il reste de la place pour une batterie plate.

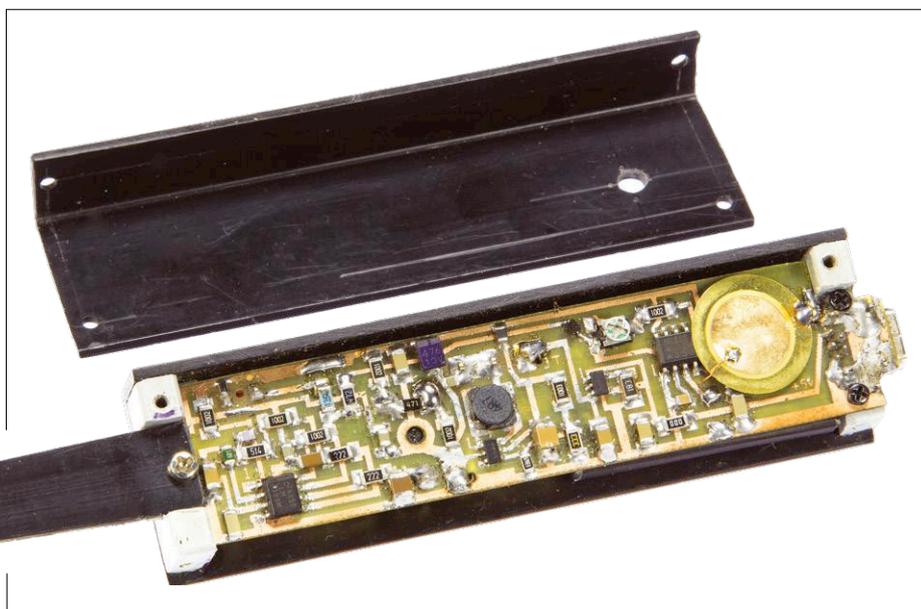


Figure 4. Le boîtier pour la partie sonde est réalisé avec des profilés standard en plastique du commerce.

piézo-électrique (qu'on peut heureusement faire taire avec S3).

Mais le plus important est que le transistor T2 est activé à travers R31. Ainsi le collecteur du transistor interne pilote de IC6 n'est relié à la tension d'entrée

que pendant la durée de l'impulsion. Hors impulsion, ce transistor reste « en l'air », de sorte que le convertisseur ne peut plus fournir de courant de sortie. S4 permet de basculer entre les modes permanent et pulsé.

Liens

- [1] www.allegromicro.com/en/Sample-And-Buy/Contact-Sales.aspx#germany
- [2] www.microchip.com/wwwproducts/en/en547080
- [3] www.onsemi.com/pub/Collateral/MC34063A-D.PDF
- [4] www.microchip.com/wwwproducts/en/MCP73812
- [5] www.elektormagazine.fr/160337

Pour que le LM358 puisse fonctionner comme amplificateur, sa tension d'alimentation est légèrement surélevée par rapport à la masse. Le pôle négatif de la batterie n'est pas connecté à la masse en direct, mais à travers une diode Schottky D2, dont la petite chute de tension de 0,3 V est suffisante. On peut s'en passer en utilisant un double amplificateur rail-à-rail.

À la sortie du convertisseur IC6, on dispose donc d'une tension continue à relier à la masse au travers de deux résistances, ce qui nous donne une source de courant pur jus. Selon la loi d'Ohm, on obtient 250 mA en reliant le conducteur à tester au côté ouvert de R38 (K4) et 500 mA quand on le relie au point commun aux deux résistances R37 et R38 (K3). L'autre extrémité du conducteur doit bien entendu être reliée à la masse en K5 au moyen d'un petit fil d'environ 10 mm de long, dont le champ magnétique permettra de tester la sonde.

La charge des batteries

Les deux parties de l'appareil sont alimentées par des batteries Li-Ion. La grosse batterie (3,7 V/4,1 Wh) de la source de courant provient d'ordiphones réformés Galaxy-3 de Samsung et est facile à se procurer pour quelques euros sur l'internet sous la référence EB504465VUCSTD. La petite batterie (3,7 V/200 mAh) pour la partie sonde, référence D1A083382009, appartenait à l'origine à un mini-caméscope PMDV85. L'auteur l'a découverte comme surplus bon marché chez Pollin-Elektronik. Le modèle précis importe peu, ce qui compte, c'est la tension et la compatibilité des dimensions avec le boîtier choisi.

Pour la charge des deux batteries, on utilise le même régulateur du type MCP73812 [4] de Microchip (IC5 et IC7). Ces puces sont spécialement conçues pour la charge de batteries Li-Ion. La courbe de charge est fixée, il n'y a donc rien dont on ait à se soucier. Le processus de charge ne s'arrête pas automatiquement quand la batterie atteint la pleine charge, mais seulement lorsqu'on débranche la tension d'alimentation du chargeur. La résistance connectée à la broche 5 de la puce détermine le courant de charge. Le besoin en énergie de la source de courant dépend bien entendu de son débit. Avec un courant de test permanent de 500 mA sous 250 mV, la batterie fournit 148 mA. Avec R19 = 10 k Ω , le courant de charge sous 5 V est fixé à 100 mA, compatible sans problème avec un port USB.

Pour la partie sonde, qui consomme sensiblement moins d'énergie (25 mA environ), on peut réduire le courant de charge à 50 mA en prenant R15 = 20 k Ω . Ici aussi, l'alimentation provient d'un port USB.

Réalisation et mise en œuvre

L'auteur a conçu un dessin de circuit imprimé pour chaque partie, disponibles sur la page du projet [5]. Pour la partie source de courant, il a utilisé un boîtier standard en matière plastique (**fig. 3**). Pour la partie sonde, il faut réaliser soi-même un boîtier adapté. Comme déjà mentionné, l'auteur s'est servi de profilés du commerce en plastique (**fig. 4**). Il est également possible de modifier un produit standard (comme le boîtier LP1 de Teko) pour qu'il puisse contenir le circuit imprimé. ◀

(160337 – version française : Helmut Müller)

La plus vaste sélection de composants électroniques



MOUSER
ELECTRONICS

mouser.fr

COMMANDEZ EN TOUTE CONFIANCE

Distributeur agréé de semi-conducteurs et
composants électroniques pour les ingénieurs d'étude.

Elektor International Media est fier de vous présenter son mur des célébrités 2018 ! Nous travaillons en partenariat avec les plus grandes sociétés d'électronique – des distributeurs de composants et instruments jusqu'aux fabricants – pour promouvoir des produits et des services passionnants qui stimulent l'innovation de haute technologie. Les entreprises inscrites sur le mur des célébrités de cette année offrent aux ingénieurs électroniciens, aux *makers* et aux étudiants, un large éventail d'outils et de solutions indispensables pour concevoir les produits électroniques de demain.

AVNET[®] SILICA

www.avnet.com

 **EBV**Elektronik
I An Avnet Company I

www.ebv.com

ARROW

www.arrow.com

Honeywell

THE POWER OF **CONNECTED**

www.honeywell.com

KCS Elektronica
assemblage
Research & Development

www.kcs.tv


TRINAMIC
MOTION CONTROL

www.trinamic.com



www.icape-group.com

 **KEYSIGHT**
TECHNOLOGIES

www.keysight.com

**BECKER
MÜLLER** 
www.becker-mueller.de

www.becker-mueller.de



www.eurocircuits.com

 **kurtz ersa**

www.kurtzera.com


Bürklin
DIE GANZE ELEKTRONIK

www.buerklin.com

 **HAMMOND
MANUFACTURING[®]**

www.hammondmfg.com

 **CONRAD**

www.conrad.com


CDPStudio

www.cdpstudio.com

HEILIND
Performance. Trust. Innovation.

www.heilind.com

LOETRONIC

www.loetronic.com

MUR DES CÉLÉBRITÉS

2018



www.microchip.com



www.picotech.com



www.schaffner.com



www.mouser.com



www.pi-top.com



www.schurter.com



www.multi-circuit-boards.eu



www.radiocontrolli.com



www.siglent.com



www.ni.com



www.recom-power.com



www.labcenter.com



www.peak-system.com



www.reichelt.com



www.wiznet.io



www.peaktech.de



www.lpkf.com



www.we-online.com



www.phoenixcontact.com



www.rs-online.com



www.unitronic.de

(PUBLICITÉ)

synthèse numérique directe sur Arduino expériences intéressantes à moindres frais

Roland Stiglmayr

Le principe de la synthèse numérique directe (DDS, *Direct Digital Synthesis*) est désormais exploité dans la plupart des appareils de télécommunications et de mesure à haute fréquence et audio. La SND est devenue une puissante alternative aux synthétiseurs à boucle à verrouillage de phase (PLL, *Phase Locked Loop*) parce que sa résolution en fréquence est extrêmement élevée et parce qu'il est possible de programmer la forme de sa courbe ainsi que modifier sa fréquence quasiment instantanément.

Le mode de fonctionnement de la SND ou d'un générateur SND a été abordé dans un article de 2015 d'Elektor [1]. Cet article a certainement éveillé la curiosité de nombreux lecteurs et aussi l'envie de réaliser des essais personnels pour mieux comprendre cette technique. L'idéal serait d'utiliser une plateforme simple d'expérimentation, à monter soi-même sans grands moyens. La carte électronique à tout faire Arduino, ne répond-elle pas précisément à ce besoin ?

La réponse est oui : nous l'avons utilisée pour réaliser un montage fonctionnel. Un Arduino Uno et un simple filtre passe-bas font l'affaire. Le filtre passe-bas est constitué de quelques composants que l'on peut souder en l'air et brancher directement sur la barrette de connexion d'un Arduino (**fig. 1**). Si vous souhaitez un montage SND plus élaboré, construisez un simple *shield* sur une plaque pastillée (**fig. 2**).

Le but premier de ce projet n'est pas de réaliser un générateur SND, mais avant tout d'explorer et de comprendre cette technique. C'est pourquoi il est très utile de disposer d'un oscilloscope pour bien en comprendre le mécanisme. Il va de soi

que le signal de sortie (sinusoidal, de 1,9 Hz à 31 kHz), peut s'écouter sur un amplificateur BF. Mais il sera difficile d'en tirer des conclusions vraiment détaillées.

Que permet l'expérimentation avec un Arduino ?

Avec la SND, on peut synthétiser différentes formes de signaux de fréquence pratiquement quelconque. On peut faire varier la fréquence, la phase et l'amplitude du signal de sortie quasi instantanément. C'est pourquoi la SND est idéale pour la conception de générateur de fonctions, oscillateurs locaux, oscillateurs sin/cos pour modulateurs I/Q, mais également pour créer directement diverses modulations numériques.

La carte Arduino produit un signal sinusoidal de fréquence réglable de 1,9 Hz à 31 kHz. Il est possible d'augmenter ou de diminuer la fréquence par pas de 1,9 Hz. Un pas d'une octave est également prévu. La fréquence est modifiée au moyen de trois boutons. Des expériences avancées donnent des exemples de modulation numérique.

Les impatientes peuvent déjà télécharger le croquis [DDS_SIN](#),

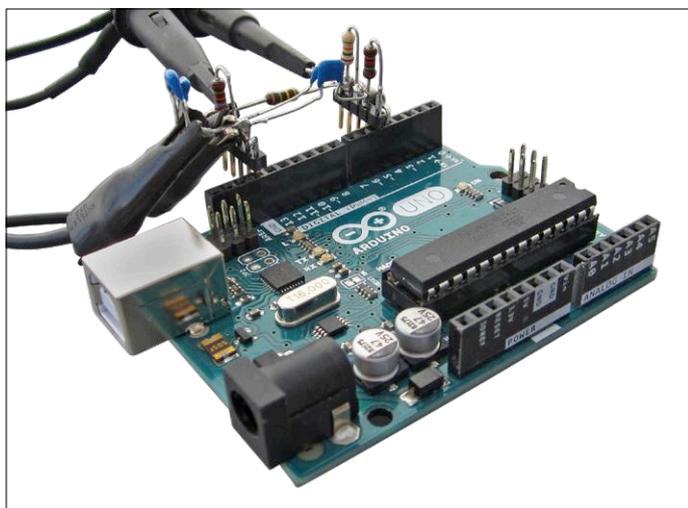


Figure 1. Montage volant du filtre passe-bas. Le point de mesure Ph_PI découplé par la résistance permet de déclencher l'oscilloscope.

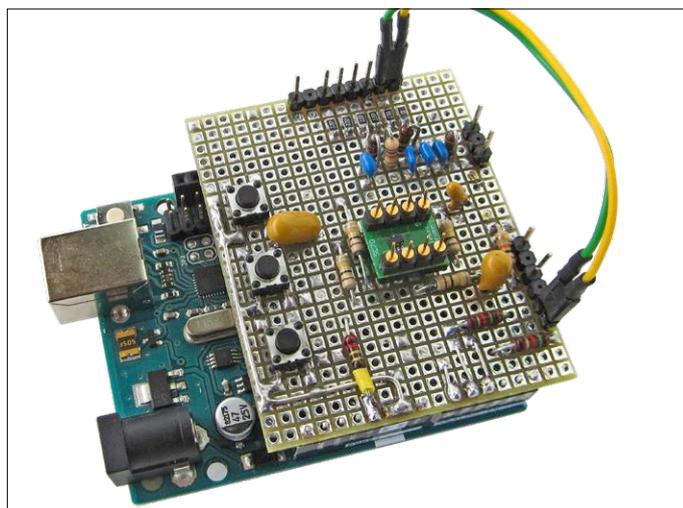


Figure 2. Réalisation personnelle de la carte d'extension avec filtre passe-bas, boutons et transformateur d'impédance équipé d'un ampli-op.

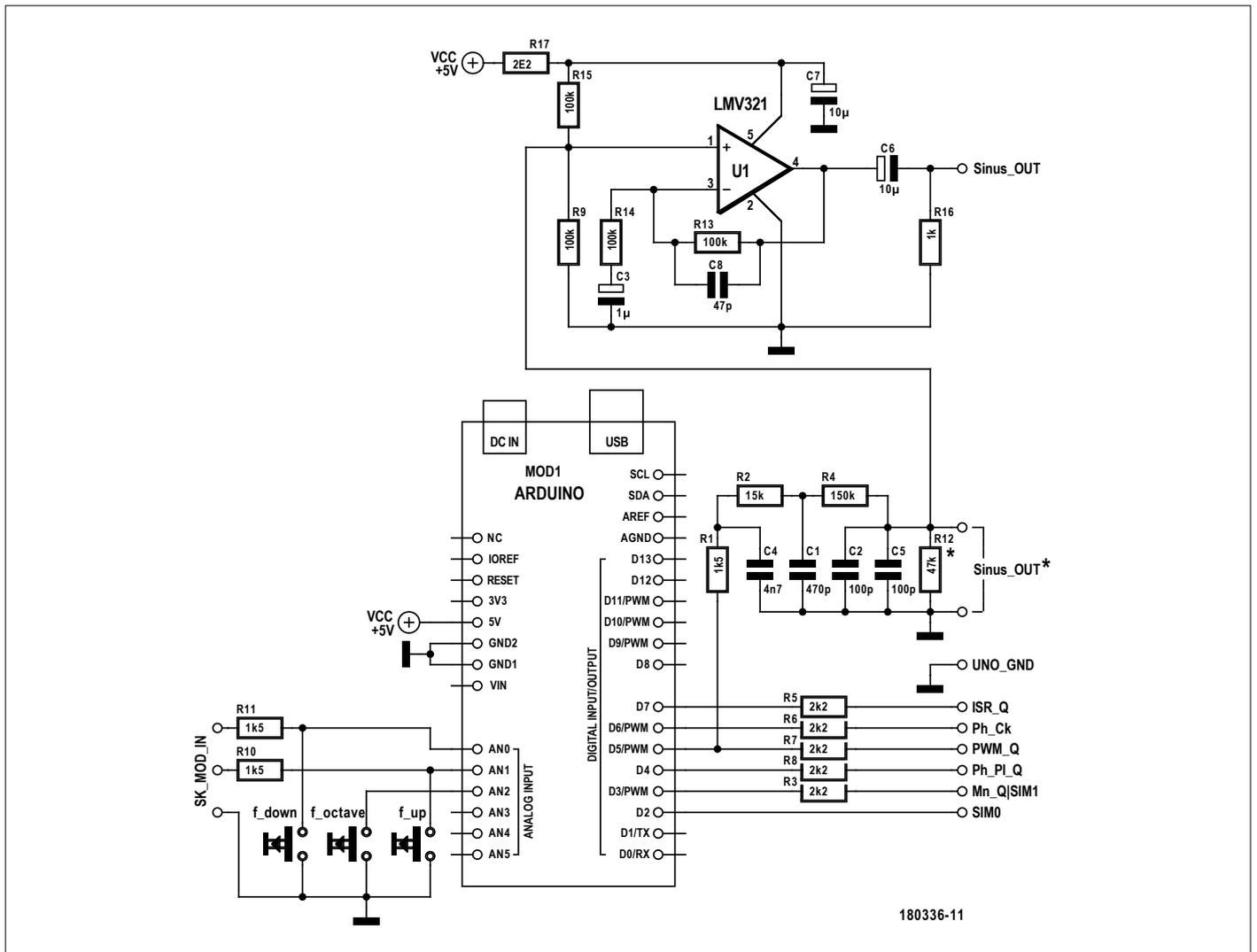


Figure 3. Le petit circuit pour l'Arduino Uno peut être réalisé sous forme d'un montage volant ou d'une carte d'extension sur une plaque d'essai pastillée.

construire le filtre passe-bas selon le schéma (fig. 3) et commencer les premières mesures.

Cependant, comme une bonne compréhension de la théorie exige de mener sérieusement les expériences et essais, mieux vaut d'abord se pencher sur le principe de la SND décrit ci-après en termes simples.

Fonctionnement de la SND

L'accumulateur de phase est au cœur de la SND. À chaque instant, il calcule la phase instantanée du signal à synthétiser. À cet effet, à chaque cycle, il ajoute un incrément de phase déterminé. Lorsque l'accumulateur déborde, la retenue est perdue et il n'est pas remis à zéro. Il prend donc une valeur qui n'est pas nécessairement nulle (sans passer par les valeurs intermédiaires que la résolution autoriserait). L'amplitude des valeurs de l'accumulateur de phase couvre un domaine de phase de 2π , de sorte que chaque dépassement marque l'écoulement d'une période du signal de sortie. Le temps nécessaire pour cela représente la période T_{out} . De cette manière, la période ne dépend que de la valeur de l'incrément de phase et d'une constante égale au quotient de la capacité de l'accu de phase (l'amplitude de sa plage de valeurs) et de la fréquence d'horloge f_{DDS} .

$$(phase_{accu_max} / phase_{increment}) * t_{ck,DDS} = T_{out}$$

soit,

$$(phase_{increment} / (phase_{accu_max} * t_{ck,DDS})) = f_{out}$$

En raison de l'arithmétique entière, le plus petit pas de phase possible est égal à 1. On en déduit la fréquence la plus basse f_{min} :

$$f_{min} = (1 / (phase_{accu_max} * t_{ck,DDS}))$$

Si nous définissons l'incrément de phase comme un multiple entier du pas élémentaire de phase, on peut calculer la fréquence de sortie :

$$f_{out} = f_{min} * n \quad (\text{avec } n \text{ entier et } \leq (phase_{accu_max}/2))$$

f_{min} est ainsi la constante appelée résolution de fréquence du système. Cette relation est remarquable, car la fréquence de sortie ne dépend linéairement que d'un seul facteur. Prenons un exemple, à savoir synthétiser un signal à une fréquence de 2 kHz : un accumulateur de phase à 16 bits

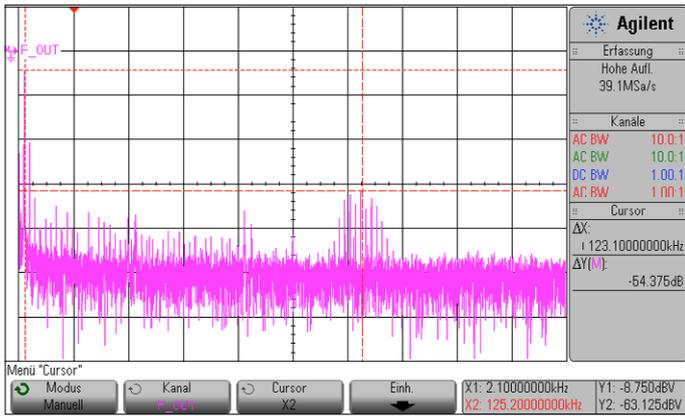


Figure 4. Spectre du signal de sortie à 1953 Hz.
 $f_{mid} = 100 \text{ kHz}$, 20 kHz/div, 20 dB/div.

déborde au bout de 2^{16} périodes d'horloge. Avec une fréquence d'horloge de 125 kHz pour la SND, f_{min} est égale à $125 \text{ kHz} / 65536 = 1,90734 \text{ Hz}$.
 Nous en déduisons l'incrément de phase pour 2 kHz :

$$\text{phaseincrement}_{2\text{kHz}} = 2 \text{ kHz} / 1,90734 \text{ Hz} = 1048,58$$

Cela fait apparaître un problème inhérent à la SND : puisque l'incrément de phase est représenté par un nombre entier, la fréquence de 2 kHz ne peut être exacte qu'à 1 Hz près. Pour résoudre ce problème, on utilise dans les circuits intégrés SND un accumulateur de phase ayant un très grand nombre de bits. Jusqu'à maintenant, nous n'avons considéré que les valeurs de phase du signal à synthétiser. Il nous manque en effet l'amplitude du signal correspondant à chaque valeur de phase. Comment obtient-on les amplitudes correspondant aux valeurs successives de la phase ? Ces valeurs sont enregistrées dans un tableau dit LUT (*Look Up Table*). En raison de la très grande quantité de mémoire nécessaire, on n'associe pas une amplitude à chaque valeur de la phase, mais seulement à chaque $2^{\text{ième}}$ valeur. Le nombre d'entrées restantes de la table est donc le maximum divisé par 2^n . Le facteur n est ainsi choisi de sorte que les amplitudes correspondant aux phases restantes ne fassent pas de grands sauts de valeur. Il faut donc tenir compte de la résolution de l'amplitude. Avec Excel, il est facile de construire une telle table et de l'optimiser [1].

Voici un exemple pour mieux comprendre : si la résolution de l'amplitude est de 7 bits (soit 128 valeurs discrètes) et que la forme d'onde à obtenir est sinusoïdale, la table devra contenir un minimum de 128 entrées. Cela permet d'avoir une résolution suffisante aux endroits où la pente est la plus élevée, soit aux passages par zéro. Avec un accumulateur de phase à 16 bits, on obtient un facteur de division de 2^9 .

La valeur ainsi réduite de l'accumulateur de phase représente maintenant un index qui renvoie à l'entrée correspondante de la table. Les amplitudes ainsi pointées sont transmises à un convertisseur numérique/analogique (CN/A), qui au final produit la tension de sortie. Habituellement, les CN/A sont réalisés à l'aide de réseaux R2R. Pour simplifier le circuit, il est aussi possible d'utiliser un modulateur de largeur d'impulsion (MLI). Dans tous les cas, il faut mettre un filtre passe-bas en aval pour atténuer les résidus de fréquence d'horloge, les fréquences latérales ainsi que les harmoniques de rang élevé.

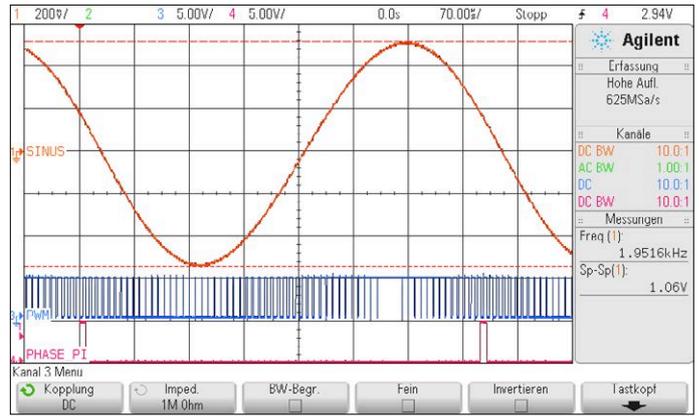


Figure 5. Représentation dans le domaine temporel du signal de sortie de la figure 4.

Mise en pratique à l'aide du logiciel

Laissons là la théorie... Passons à la pratique avec un Arduino Uno et un logiciel simple. Si ce n'est déjà fait, il est temps de charger le croquis `DDS_SIN [1]` dans l'Uno.

Comme pour chaque projet, se pose la question de savoir quels blocs de fonction sont nécessaires et comment les réaliser. Il faut bien évidemment une horloge, qui déclenche l'addition de l'incrément de phase pour obtenir la valeur courante de l'accumulateur de phase. À cet effet, le `Timer0` appelle à intervalle régulier la routine d'interruption `Timer0-Interrupt-Service-Routine (ISR)`. L'ISR réalise l'addition, produit l'index qui pointe sur la LUT, lit l'amplitude correspondante et la transmet au CN/A. La LUT est définie comme variable de type `Byte-Array` (tableau d'octets) dans les déclarations. Une MLI réalise la conversion numérique/analogique.

Nous aurions bien aimé que l'Uno ne soit pas aussi limité en vitesse. Notre but est en effet de produire un signal de sortie utilisable jusqu'à 12 kHz au moins. Si nous voulons au minimum huit points d'amplitude par période, la fréquence d'horloge minimale est de 96 kHz. Et comme à fréquence constante du signal, la distance entre ce dernier et les fréquences latérales augmente avec la fréquence d'horloge, plus celle-ci est élevée, mieux c'est. L'index de `Timer0` est initialisé de sorte à revenir à zéro au bout de 128 coups d'horloge système. De cette manière, l'ISR est appelée toutes les 8 μs ce qui correspond à une horloge SND de 125 kHz. Afin que la MLI travaille en synchronisme avec chaque amplitude produite, elle doit fonctionner au même rythme. Cela ne correspond pas à l'initialisation standard du `Timer` en mode MLI qui nécessite 256 impulsions (fréquence résultante de 62,5 kHz). Ce problème peut être contourné par une astuce. La MLI est également réalisée avec le `Timer0`. Le compteur compte seulement jusqu'à 128, de sorte que seulement 128 périodes élémentaires sont produites. Par conséquent, la résolution de la MLI se réduit à 7 bits. Afin de ne pas ralentir inutilement l'exécution du programme, nous utilisons un minimum d'instructions de la bibliothèque Arduino. Pour pouvoir mieux interpréter le déroulement du programme sur un oscilloscope, nous produisons quelques signaux d'état décrits plus en détail dans les commentaires du programme. Les résistances sur les signaux d'état atténuent les parasites qu'une charge capacitive pourrait produire au point de mesure. Le signal « `Ph_PI` » qui apparaît un peu avant le passage négatif par zéro est une particularité de la SND. « `Ph_PI` » n'intervient

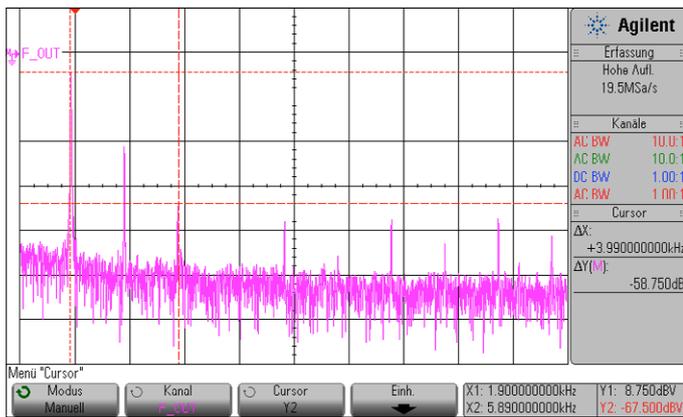


Figure 6. Spectre du signal de sortie de la figure 4.
 $f_{mid} = 10 \text{ kHz}$, 2 kHz/div, 20 dB/div.

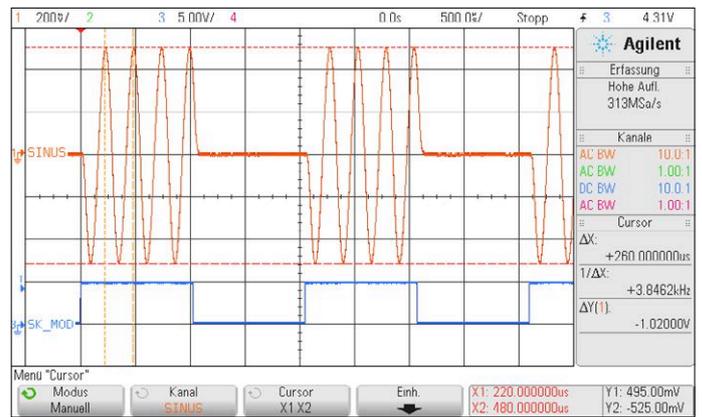


Figure 7. Signal de sortie en modulation ASK.
 $f_0 = 3906 \text{ Hz}$, vitesse = 1 kbit/s.

effectivement dans chaque période que pour les fréquences les plus basses. Cela tient à l'arithmétique entière : en cas de dépassement, l'accumulateur de phase reprend les additions avec une valeur initiale qui change constamment au lieu d'être forcée à zéro. Ainsi, cette phase précise n'est pas produite à chaque période.

Avec passe-bas et convertisseur d'impédance

Un générateur SND comporte toujours un filtre passe-bas de sortie. En particulier avec une MLI, le filtre passe-bas influence considérablement la qualité du signal de sortie. Ici, il faut régler la fréquence de coupure du filtre passe-bas à 12 kHz pour respecter une décade d'écart avec sa fréquence d'horloge. Avec un filtre d'ordre 3, l'atténuation est par définition de 60 dB par décade c'est-à-dire peu ou prou la valeur de l'atténuation de la fréquence d'horloge et des fréquences latérales. Le passe-bas pourrait être un filtre actif construit à partir d'un ampli-op, mais ce serait trop coûteux. Un autre inconvénient serait également les valeurs non standard à respecter pour les éléments du réseau de contre-réaction.

Il est plus simple d'adopter un réseau RC passif. En utilisant trois filtres passe-bas passifs en cascade, on obtient un filtre passe-bas d'ordre 3. Toutefois cela ne fonctionne que si les cellules RC sont découplées entre elles. C'est le cas si l'impédance d'entrée d'une cellule RC est très supérieure à l'impédance de sortie de la cellule immédiatement précédente. La fréquence de coupure de chaque cellule est calculée de sorte que chaque cellule de l'ensemble du filtre procure une atténuation de 1 dB. Pour les trois cellules, l'atténuation est alors de 3 dB. En utilisant la fonction de transfert on en déduit

$$f_{\text{coupure, cellule RC}} = 1,98 * f_{\text{coupure, sortie}}$$

Avec les éléments calculés indiqués sur le schéma, la fréquence de coupure de chaque cellule est de 22,6 kHz. Pour réduire la valeur de la résistance de sortie du filtre, la dernière cellule comporte une seconde résistance, R12, connectée à la masse (GND). Si elle a bien l'effet recherché, elle réduit cependant l'amplitude de sortie.

L'analyse spectrale en sortie du filtre passe (fig. 4) démontre l'efficacité du circuit. La figure 5 représente le signal dans le domaine temporel. Les 125 kHz de la MLI et les fréquences latérales autour de la fréquence MLI sont à 54,3 dB au

dessous du signal synthétisé à 1953 Hz.

Si vous souhaitez utiliser le générateur pour des mesures ou alimenter un amplificateur BF, mieux vaut utiliser un ampli-op comme transformateur d'impédance. Il faut remplacer la résistance de sortie du filtre R12 évoquée ci-dessus par les deux résistances R15 et R9 qui fixent le point de travail en continu à $V_{cc}/2$. L'ampli-op travaille comme amplificateur non inverseur couplé en continu. Un condensateur de sortie permet de découpler le signal. Sans R14/C3, le gain en tension A vaut +1 et avec R14/C3 A=+2.

Les croquis

Il existe plusieurs croquis Arduino pour réaliser les différents types de modulation. La page du projet [1] permet de les télécharger.

Le générateur sinusoïdal DDS_SIN

Le programme principal interroge le clavier. Selon la touche activée, la variable de l'incrément de phase est incrémentée, décrétementée ou doublée (touche « octave »). Lors de l'actualisation de la fréquence, la nouvelle valeur est transmise par l'UART au moniteur série de l'IDE. Les étapes du programme sont décrites pas à pas en commentaire.

La figure 5 montre le signal de sortie sinusoïdal « SINUS » prélevé à la sortie du filtre passe-bas, le signal « Ph_PI » qui déclenche l'oscilloscope et le signal d'entrée « MLI » du filtre passe-bas. Les figures 4 et 6 montrent le spectre associé à la fig. 5. Sur la figure 6, nous observons les harmoniques et la distance entre la troisième harmonique et la fondamentale, ainsi que le 1^{er} harmonique.

Modulation par déplacement d'amplitude avec DDS_ASK

Avec la modulation ASK (*Amplitude Shift Keying*), l'amplitude de la fréquence porteuse est augmentée ou diminuée, selon l'état logique de l'entrée de modulation. C'est très facile à réaliser en SND. Pour chaque état de l'information d'entrée, nous utilisons une table contenant les valeurs successives de l'amplitude. L'index pointe sur la valeur correspondante de la table en fonction de l'information d'entrée.

En alternant l'amplitude entre 0 % et 100 %, nous obtenons une forme particulière d'ASK, appelée « ON OFF Keying » (OOK). Cette OOK est illustrée dans cette expérience. Le programme fixe l'incrément de phase en fonction de l'état d'entrée, soit

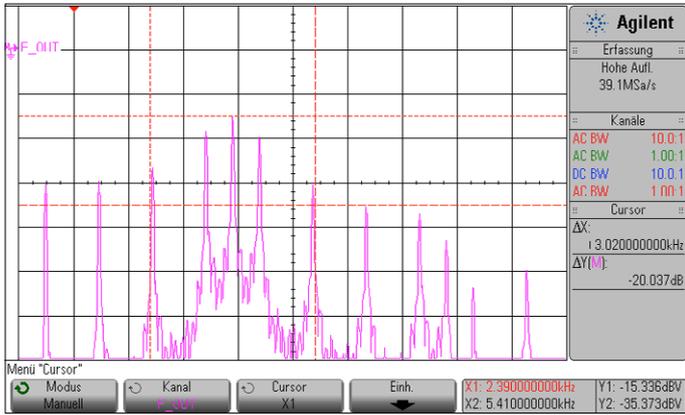


Figure 8. Spectre du signal de sortie en modulation ASK de la figure 7.
 $f_{mid} = 5 \text{ kHz}$, 1 kHz/div, 10 dB/div.

à 3906,25 Hz, soit à 0 Hz. Le changement ne s'effectue qu'au passage par zéro de la sinusoïde. D'autres informations sont données en commentaire du croquis.

La **figure 7** montre le signal ASK et la **figure 8** le spectre correspondant. Ici SIM0 sur PD2 commande l'entrée SK_MOD0. Avec le signal de simulation interne SIM0, la fréquence de modulation est de 500 Hz, la vitesse de transmission correspondante est de 1 kbit/s. Comme pour toutes les autres mesures, les conditions sont idéales, car le signal de modulation est synchronisé sur la phase de la porteuse.

Le spectre est intéressant dans la mesure où il contient exactement la moitié du spectre produit avec la modulation 2FSK. Il contient la fréquence porteuse f_0 et les bandes latérales avec $(f_0 \pm f_{mod})$, $(f_0 \pm 3*f_{mod})$ et ainsi de suite. Il apparaît donc immédiatement que la largeur des bandes latérales dépend de la fréquence de modulation et par conséquent de la vitesse de transmission des données. Les curseurs de fréquence sont positionnés de façon à inclure environ 90 % de la puissance spectrale dans ce signal non limité en largeur de bande. L'étalement des fréquences résultantes représente la bande passante nécessaire pour le canal de transmission. La modulation OOK est par exemple exploitée par la norme AISG, qui spécifie la commande des antennes des stations de base en téléphonie mobile.

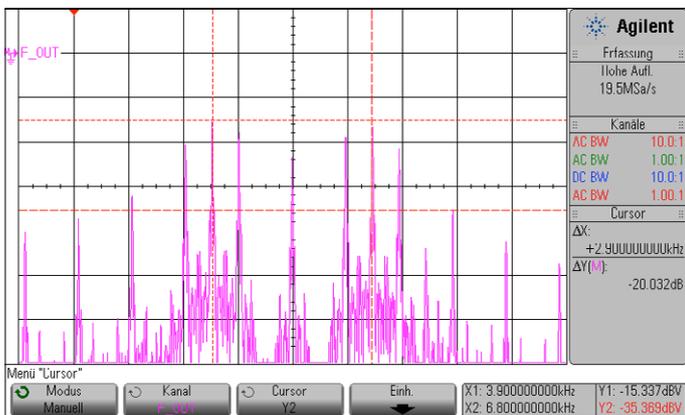


Figure 10. Spectre du signal de sortie 2FSK de la figure 9.
 $f_{mid} = 5,37 \text{ kHz}$, 1 kHz/div, 10 dB/div.

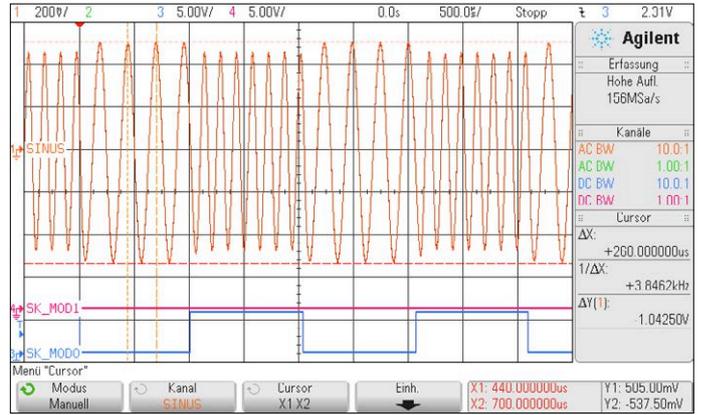


Figure 9. Signal de sortie en modulation 2FSK.
 $f_1 = 3906 \text{ Hz}$, $f_2 = 6836 \text{ Hz}$, vitesse = 1 kbit/s.

Modulation par déplacement de fréquence avec DDS_FSK

Avec la modulation FSK (*Frequency Shift Keying*), chaque état logique des informations numériques d'entrée correspond à une fréquence discrète. Pour un signal d'entrée à 1 bit, l'état bas (LOW) produit une fréquence, l'état haut (HIGH) en produit une autre. Si l'entrée utilise n bits, il faut alors 2^n fréquences pour représenter les 2^n états possibles. Chaque état des n bits représente ce que l'on appelle un symbole. À chaque pas, un symbole à n bits est transmis. Si les fréquences extrêmes sont constantes, la bande passante nécessaire pour le canal de transmission ne dépend pas du nombre de symboles.

Dans le programme, pour l'information d'entrée de chaque symbole à 2 bits, nous déclarons une valeur prédéfinie pour la variable d'incrément de la phase. Cela permet d'attribuer une fréquence propre à chacun des quatre états logiques. Pour activer ces quatre fréquences, il suffit d'actionner f_{down} et f_{up} .

Pour simplifier l'expérimentation, nous créons un signal de pilotage à 2 bits, SIM0 et SIM1, appliqué comme signal d'entrée sur SK_MOD_IN. En raison du découplage des résistances des boutons, il est possible de réduire le signal 4FSK à un signal 2FSK en actionnant f_{up} ou f_{down} .

La **figure 9** montre le signal 2FSK et la **figure 10** le spectre correspondant. La **figure 11** montre le signal 4FSK et la **figure 12** le spectre correspondant. La ligne spectrale située exactement

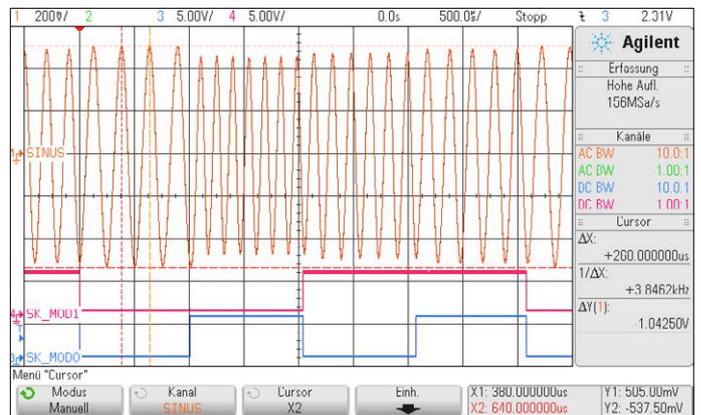


Figure 11. Signal de sortie en modulation 4FSK.
 $f_1 = 3906 \text{ Hz}$, $f_2 = 6836 \text{ Hz}$, $f_3 = 5859 \text{ Hz}$, $f_4 = 4883 \text{ Hz}$, vitesse = 2 kbit/s.

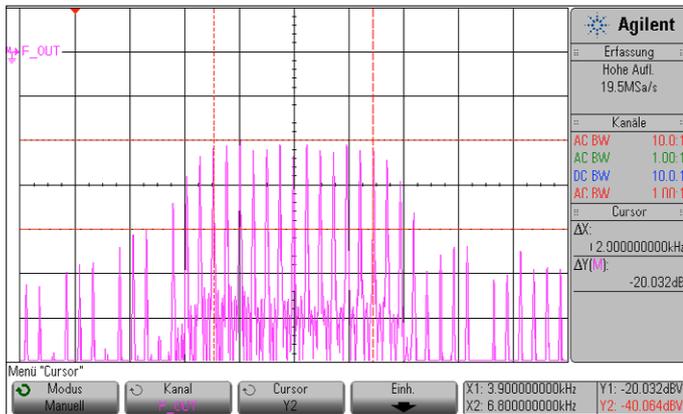


Figure 12. Spectre du signal de sortie 4FSK de la figure 11.
 $f_{mid} = 5,37 \text{ kHz}$, 1 kHz/div, 10 dB/div.

au milieu de la bande provient du recouvrement des bandes latérales. En choisissant judicieusement les fréquences individuelles en fonction de la fréquence de modulation, il est possible de réduire ces recouvrements qui peuvent conduire à des défauts. Sur les graphiques représentant le signal dans le domaine temporel, nous voyons immédiatement qu'il n'y a pas de discontinuité du signal comme ce serait le cas avec une commutation de deux oscillateurs sans précaution. C'est l'une des grandes qualités de la SND. La modulation FSK est utilisée entre autres par les modems des télécopieurs ainsi que les téléphones DECT.

Modulation par déplacement de phase avec DDS_PSK

Avec la modulation PSK (*Phase Shift Keying*), pour chaque état logique des informations numériques d'entrée, le signal de sortie est affecté d'une valeur de décalage de phase par rapport à la porteuse. Par conséquent, à chaque changement d'état d'entrée correspond une valeur de phase du signal de sortie. Avec un symbole à 1 bit, le décalage de phase est de 0° à l'état bas (LOW), de $+180^\circ$ à l'état haut (HIGH). Comme un décalage de phase de -45° et $+135^\circ$ donne le même résultat, « DDS_PSK » utilise ces valeurs de sorte que le même logiciel permet de réaliser les modulations 2PSK et 4PSK. Avec les symboles à 2 bits, le décalage de phase prend les valeurs 45° , 135° , 225° ou 315° ($\hat{=}$ -45°).

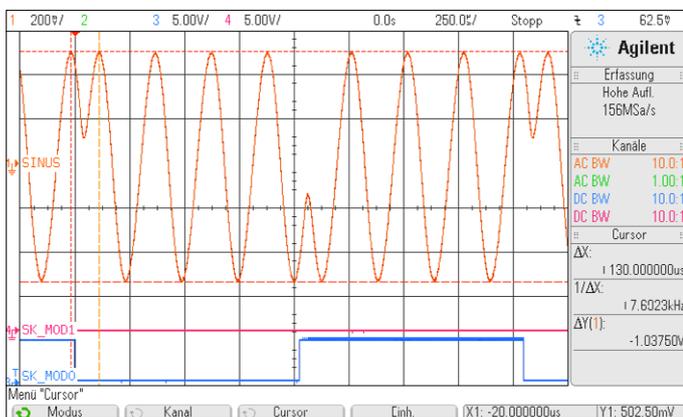


Figure 13. Signal de sortie en modulation 2PSK. Distance des curseurs = $130 \mu\text{s} \hat{=} 180^\circ$ de saut de phase, vitesse = 1 kbit/s.



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.

décembre 2018

♦ **Aeromart summit + Space industry summit**
 04 au 06/12 - Toulouse
<http://toulouse.bciaerospace.com>

♦ **BioFit**
 04 au 05/12 - Lille
www.biofit-event.com

♦ **Paris Open Source Summit**
 05 au 06/12 - Paris
www.opensourcesummit.paris/?lg=fr

♦ **SIA CESA 5.0**
 05 au 06/12 - Paris
www.sia.fr/evenements/111-cesa-5-0

♦ **SIMI** (salon de l'industrie immobilière)
 05 au 07/12 - Paris
www.salonsimi.com

♦ **NAUTIC**
 06 au 16/12 - Paris
www.salonnaviqueparis.com

♦ **Embedded-SEC18** (sécurité des systèmes embarqués)
 07/12 - Paris

♦ **Energaïa** (forum des énergies renouvelables)
 11 au 12/12 - Montpellier
<http://energaia.fr>

♦ **Supply Chain Event**
 11 au 12/12 - Paris
www.supplychain-event.com

♦ **Toymania**
 16/12 - Paris
www.toymania.org

♦ **Assises de l'embarqué**
 19/12 - Paris
<https://assises.embedded-france.org>

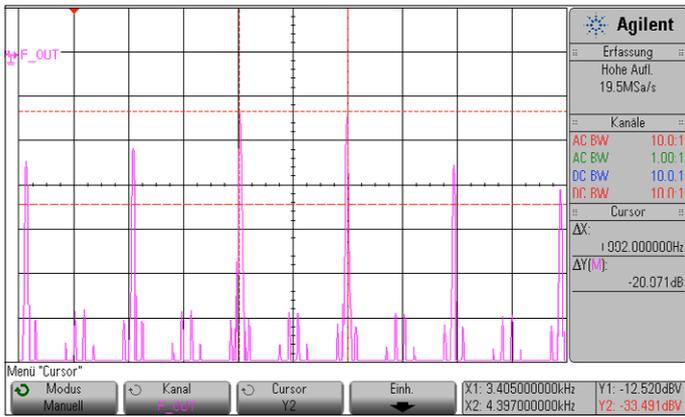


Figure 14. Spectre du signal de sortie 2PSK de la figure 13.
 $f_{mid} = 3900 \text{ kHz}$, 500 kHz/div, 20 dB/div.

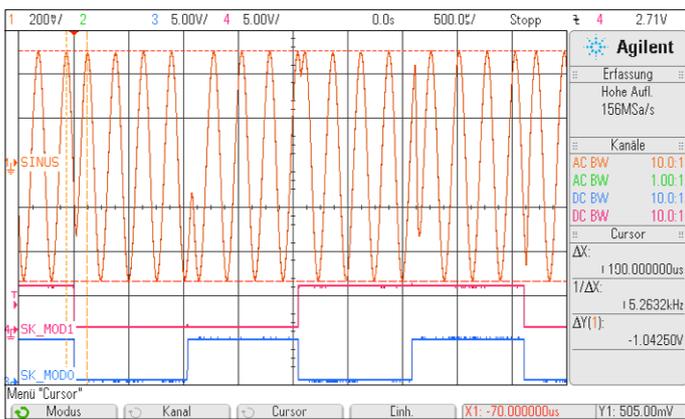


Figure 15. Signal de sortie en modulation 4PSK. Distance des curseurs = $190 \mu\text{s} \triangleq +90^\circ$; $+90^\circ$, 180° , -90° , 180° de sauts de phase.
 Vitesse = 2 kbit/s.

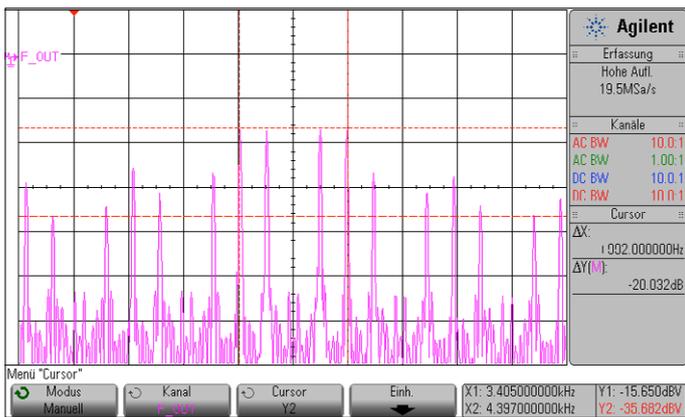


Figure 16. Spectre du signal de sortie 4PSK de la figure 15.
 $f_{mid} = 3900 \text{ kHz}$, 500 kHz/div, 20 dB/div.

Parce que la fréquence doit rester stable, à la différence du procédé FSK de modulation, la valeur de l'incrément de phase est constante. Le décalage de phase codé par l'état des données d'entrée est au contraire additionné par l'accumulateur de phase pour calculer l'index qui pointe sur la LUT. La valeur de l'accumulateur de phase contient comme auparavant la valeur momentanée de la phase zéro de la porteuse. Les décalages correspondants à chaque symbole sont enregistrés dans une table. Pour les tests, nous appliquons à nouveau les signaux SIM0 et SIM1 sur SK_MOD_IN. La **figure 13** montre le signal de sortie avec 2PSK pour des sauts de phase de 180° ; la **figure 14** montre le spectre de fréquence correspondant. La fréquence porteuse n'est pas présente ; cela vient du fait que dans la période d'observation, les intervalles pour 0 et 1 sont de même longueur. Les bandes latérales sont à $(f_0 \pm f_{mod})$, $(f_0 \pm 3f_{mod})$ et ainsi de suite.

La **figure 15** illustre le signal 4PSK (également dénommé QPSK) avec des sauts de 90° et 180° . Le chronogramme représentant les sauts de phase dépend de l'instant de changement de symbole par rapport à la phase de la porteuse. La **figure 16** montre le spectre correspondant. Là encore, la porteuse est absente. Comme avec la modulation 2PSK, cela est dû au chronogramme de la modulation et à ses conséquences temporelles sur la suite des valeurs de phase. Pour chaque symbole, l'amplitude peut être représentée par un vecteur avec un angle égal au décalage de phase. Si pendant la période d'observation, la somme des multiples vecteurs individuels reste nulle, la porteuse est supprimée.

L'analyse du signal de sortie en modulation 2PSK comme en 4PSK montre que la reconstitution, c.-à-d. la démodulation du signal numérique n'est pas possible sans synthèse locale de la porteuse. Dans les procédés de modulation PSK différentiels, qui codent les informations numériques par des transitions d'état, ce n'est pas indispensable.

En pratique, la modulation PSK est réalisée avec des modulateurs I/Q, pour lesquels la limitation de bande passante du signal de modulation conduit à pouvoir réduire de façon simple la largeur de bande occupée.

La modulation PSK, qui utilise également une combinaison de modulation d'amplitude, s'appelle QAM. C'est le procédé dominant en téléphonie mobile ainsi qu'en télévision et radio numériques.

En résumé

La SND est une technique extrêmement puissante qui trouve sa place dans de nombreuses applications. La transposition du procédé sur Arduino ne donne certes pas des résultats de grande qualité, mais permet à moindres frais de l'expérimenter et de s'y initier. Nous recommandons à ceux qui souhaiteraient aller plus loin dans cette technique de suivre le tutoriel sur le SND d'Analog Devices [2]. ◀

(180336-04 - version française : Yves Georges)

Liens

- [1] « synthèse numérique directe », Elektor 07-08/2015 : www.elektormagazine.fr/150315
- [2] Page de l'article : www.elektormagazine.fr/180336-04
- [3] Tutoriel SND (en anglais) : www.analog.com/media/en/training-seminars/design-handbooks/Technical-Tutorial-DDS/technical-tutorial-DDS.pdf

variateurs pour LED (2)

passer au numérique, c'est (parfois) ringard !

Roel Arits (Pays-Bas)

Dans le cadre d'un concours du labo [1], Roel Arits nous a proposé de nombreux circuits de variateurs analogiques. Deux d'entre eux ont été présentés dans le précédent numéro [2]. Ce second article met en lumière deux autres expériences de Roel.

Caractéristiques

- Diverses possibilités de commande analogique de LED
- Composants standard
- Invite à l'expérimentation...

Rappelons que si nous voulons que la variation – croissante ou décroissante – de la luminosité d'une LED soit linéaire, la variation du courant devra quant à elle être exponentielle. C'est aussi valable pour une lampe à incandescence, mais c'est hors de propos ici. Ceci résulte de la caractéristique plus ou moins logarithmique de notre vue ; cette loi avait déjà été établie au 19^e siècle par Weber et Fechner [3].

Inutile de nous étendre à nouveau là-dessus, contentons-nous de dire que le courant de commande des LED devra varier de manière exponentielle pour que nous percevions une variation linéaire de la luminosité.

Gyrateur

Tous les chemins mènent à Rome, et il y en a aussi de très nombreux pour concevoir un variateur analogique pour LED. La variante décrite ici utilise un oscillateur à relaxation bâti autour d'un comparateur configuré en bascule de Schmitt (figure 1).

Le comparateur IC1 est au cœur du circuit. Une fraction de sa tension de sortie est ramenée sur l'entrée non-inverseuse via la résistance R7, ce qui crée une hystérésis de part et d'autre du seuil de déclenchement fixé par le diviseur de tension R6/R9 ; nous obtenons ainsi une bascule de Schmitt. L'amplitude de la tension de sortie dépend de la valeur de R7.

La capacité des condensateurs C2 et C3 en parallèle détermine la constante de temps de l'oscillateur. Les diodes D1 et D3 garantissent que les temps de charge et décharge sont indépendants l'un de l'autre.

La décharge des condensateurs se fait via R5, elle est à peu près exponentielle, ce que nous souhaitons obtenir.

Nous voulons aussi que la charge des condensateurs soit exponentielle. Nous avons donc besoin d'un circuit qui four-

nisse un courant de charge exponentiel à partir d'une tension constante.

Si nous montons une bobine ordinaire en série avec une résistance, la variation du courant sera logarithmique lorsque nous appliquerons une tension constante. Nous observons un phénomène identique si nous remplaçons la bobine par un gyrateur, un circuit électronique qui simule le fonctionnement d'une bobine. L'avantage du gyrateur est qu'il est facile à modifier pour que la tension présente

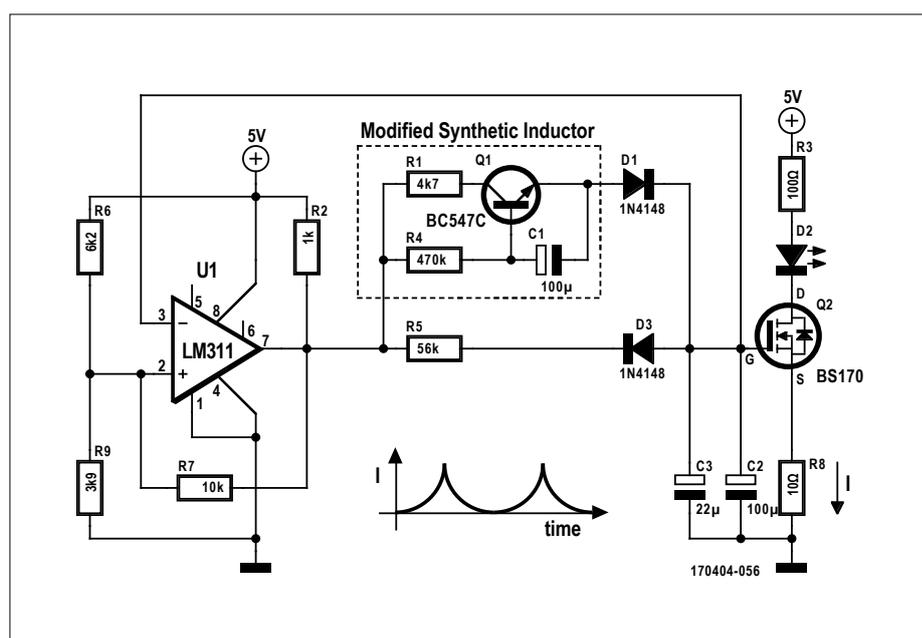


Figure 1. Le schéma d'un variateur avec un gyrateur modifié.

une caractéristique exponentielle, c'est ce que l'auteur a réalisé ici.

Le transistor T1, les résistances R1 et R2, et le condensateur C1 simulent le fonctionnement d'une bobine ; la différence provient de R1, nous y reviendrons plus loin.

La **figure 2** montre le gyrateur seul. Nous ne détaillerons pas son fonctionnement ici, le lecteur intéressé trouvera facilement des explications dans les ouvrages spécialisés. Toutefois, il est possible de décrire le fonctionnement du gyrateur de manière qualitative dans le domaine fréquentiel, à basses et hautes fréquences.

Si le signal appliqué sur l'entrée est à très basse fréquence ou continu, l'impédance du condensateur C sera très élevée. Comme la base du transistor est alimentée via R2, celui-ci sera conducteur. Vu de l'entrée, le dispositif peut être considéré en première approximation comme un court-circuit, ce qui serait aussi le cas d'une bobine.

Avec un signal à haute fréquence à l'entrée, l'impédance du condensateur est faible et il « vole » littéralement le courant de base du transistor. Celui-ci conduit moins et le circuit présente une impédance élevée ; encore une fois le comportement d'une bobine.

La résistance R1, en série avec le collecteur de Q1 (voir figure 1), cause une diminution de la tension sur le collecteur du transistor lorsque le courant I_c (et I_e) augmente. La charge des condensateurs C2//C3 sera alors approximativement exponentielle.

Le reste du circuit est classique ; le MOSFET T2 est un convertisseur tension-courant qui permet d'alimenter la LED (maximum de 5 mA avec la valeur de R8 choisie), et la résistance R3 impose une limitation supplémentaire du courant.

Avec les valeurs sélectionnées pour les composants, les temps de montée et de descente sont d'environ 1,2 s. L'augmentation et la diminution de la luminosité de la LED seront relativement symétriques. Les résistances R6 et R9 peuvent être remplacées par un potentiomètre de 10 k Ω , dont le curseur est connecté à R7 et à l'entrée non-inverseuse du comparateur. Ceci permet de régler la luminosité minimale de la LED.

Il est conseillé de choisir un type de LED à haute efficacité lumineuse, qui permettra un meilleur éclairage avec les faibles courants en jeu.

Le circuit a un sérieux inconvénient (c'eut

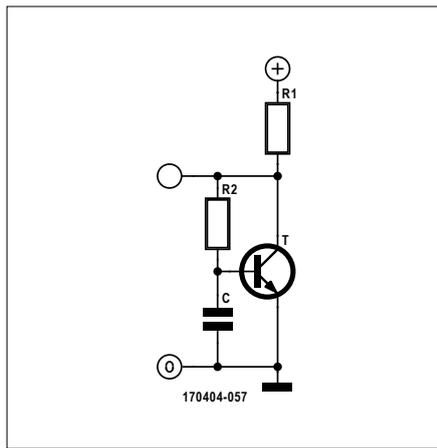


Figure 2. Le schéma du gyrateur seul, pour mieux en saisir le fonctionnement.

été trop beau...) : les temps de montée et de descente ne peuvent pas être modifiés facilement. Il faut en effet changer simultanément les valeurs du couple C2-C3, de C1, et éventuellement de R5.

La **figure 3** montre le prototype de l'auteur sur une platine d'expérimentation.

Cerise sur le gâteau

Et pour finir en beauté, un variateur mécanique ! On utilise ici un potentiomètre motorisé et un relais électromagnétique. L'effet est absolument rétro, on aurait même envie de dire *steampunk*... Dans les amplis bon marché actuels, plus question de trouver des potentiomètres pour régler le volume : on utilise un variateur électronique. C'est moins cher, plus simple à réaliser, et on peut y ajouter une télécommande pour un

faible surcoût.

Les amplis plus chers, et certainement ceux haut de gamme, utilisent toujours des potentiomètres de haute qualité, afin de minimiser la distorsion et le bruit. Si on veut une télécommande, on n'a guère le choix : il faut des potentiomètres motorisés. Une des marques les plus connues est Alps ; leurs modèles sont d'excellente qualité et ils ne sont pas trop chers.

Inutile de démonter votre ampli pour lui emprunter son potentiomètre motorisé, un modèle de récupération est amplement suffisant pour la réalisation d'un variateur pour LED. Celui que l'auteur a trouvé sur un ampli mis au rebut n'est pas un modèle étanche dernier cri, mais un simple potentiomètre couplé à un moteur avec un réducteur. Un couplage souple entre l'axe du potentiomètre et le réducteur permet aussi de régler le volume manuellement.

Les potentiomètres que l'on trouve dans les amplis sont en général des modèles à variation logarithmique, la caractéristique de sensibilité de notre ouïe l'étant aussi (il n'y a pas que notre vue qui suit la loi de Weber-Fechner, mais bien tous nos sens). Si nous utilisons un tel potentiomètre pour faire varier la luminosité d'une LED et que la vitesse de rotation du moteur est constante, nous obtiendrons automatiquement la variation exponentielle du courant qui nous permettra de percevoir une variation linéaire de la luminosité.

Si vous voulez expérimenter avec ce montage, mais que vous ne disposez que d'un potentiomètre linéaire, ne vous

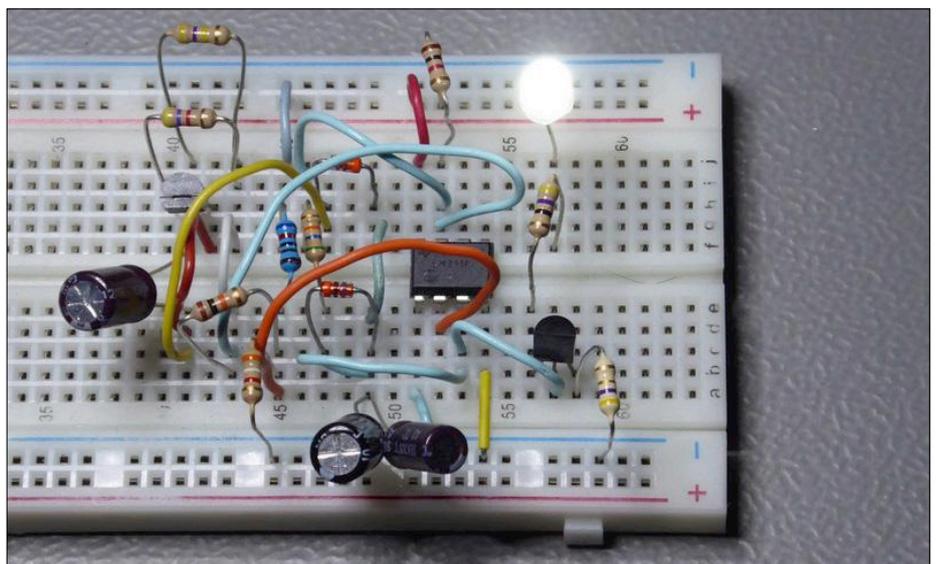


Figure 3. Le prototype de l'auteur.

désespérez pas pour autant : en mettant une résistance en parallèle avec une des deux moitiés du potentiomètre, on obtient une caractéristique plus ou moins logarithmique. Ceci permet également de modifier la caractéristique d'un potentiomètre logarithmique (R4 sur le schéma de la **figure 4**).

Il nous faut maintenant un circuit qui fait tourner le moteur à vitesse constante dans un sens puis dans l'autre, ce qui n'est pas spécialement compliqué à réaliser. On utilise à nouveau une bascule de Schmitt bâtie autour d'un comparateur (IC1 sur la figure 4). La tension sur le curseur du potentiomètre motorisé est comparée à la tension sur l'entrée non-inverseuse du comparateur, réglée à l'aide de P3. La réaction entre la sortie et l'entrée non-inverseuse du comparateur via R6 crée une hystérésis, ce qui nous donne deux seuils de basculement. La tension de sortie du comparateur fixe le sens de rotation du moteur, qui est modifié via le relais : le moteur tourne dans un sens jusqu'à ce qu'un des seuils de basculement soit atteint, puis dans l'autre jusqu'à ce que le second seuil le soit à son tour. Il s'agit donc d'un oscillateur mécanique. La luminosité de la LED est proportionnelle à la tension sur le curseur du potentiomètre P2.

La **figure 5** montre le prototype de l'auteur sur une platine d'expérimentation ; on voit bien que le réducteur du moteur n'est pas spécialement protégé ! L'oscillogramme de la **figure 6** montre la belle caractéristique exponentielle obtenue avec le circuit. ◀

(170404-B-04 -

version française : Jean-Louis Mehren)

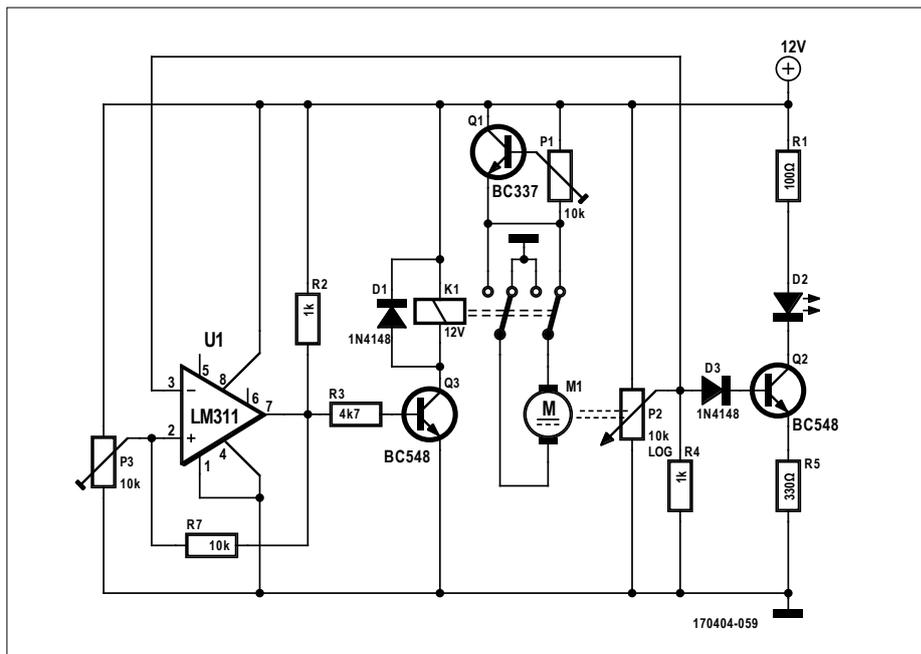


Figure 4. Le cœur du variateur mécanique est un potentiomètre motorisé.

Avertissement

Les circuits décrits dans cet article sont des preuves de concept (*proofs of concept*). Ils sont conçus pour de très basses tensions, de l'ordre de 5 à 12 V, et pour la commande de quelques LED. Ils **ne sont pas du tout adaptés** à la commande de lampes à LED, et **encore moins** à un raccordement au réseau électrique !

Liens

- [1] Gagnants du concours de circuits de fondu lumineux à LED : www.elektormagazine.fr/160558
- [2] Variateurs pour LED (1), Elektor 09-10/2018 : www.elektormagazine.fr/170404-03
- [3] Loi de Weber-Fechner : https://fr.wikipedia.org/wiki/Loi_de_Weber-Fechner

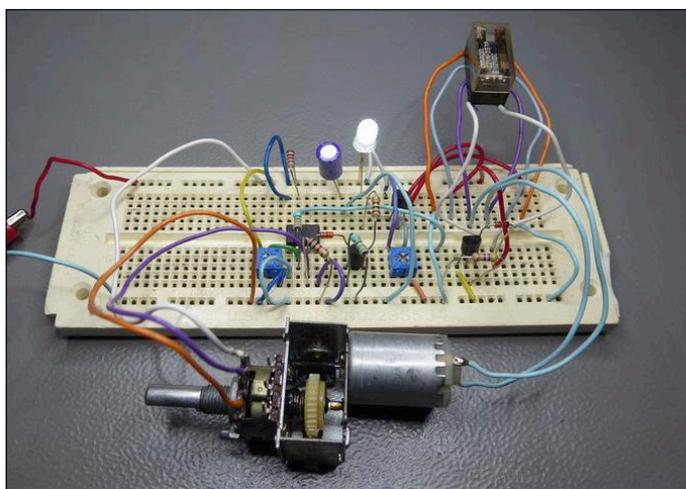


Figure 5. Un potentiomètre motorisé « suranné » donne un look vintage au variateur mécanique.

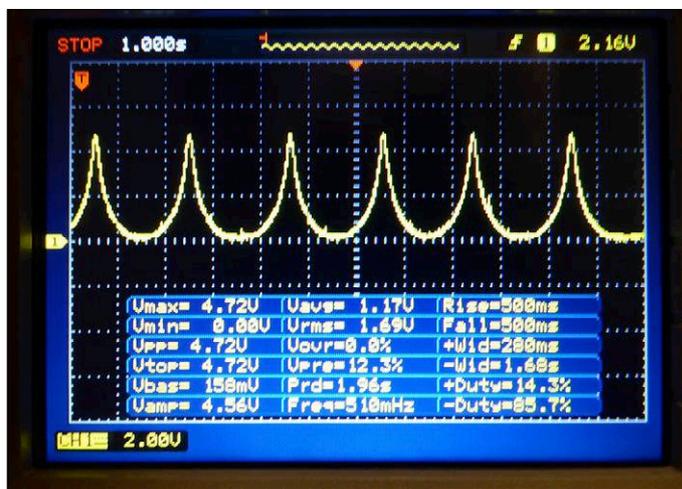
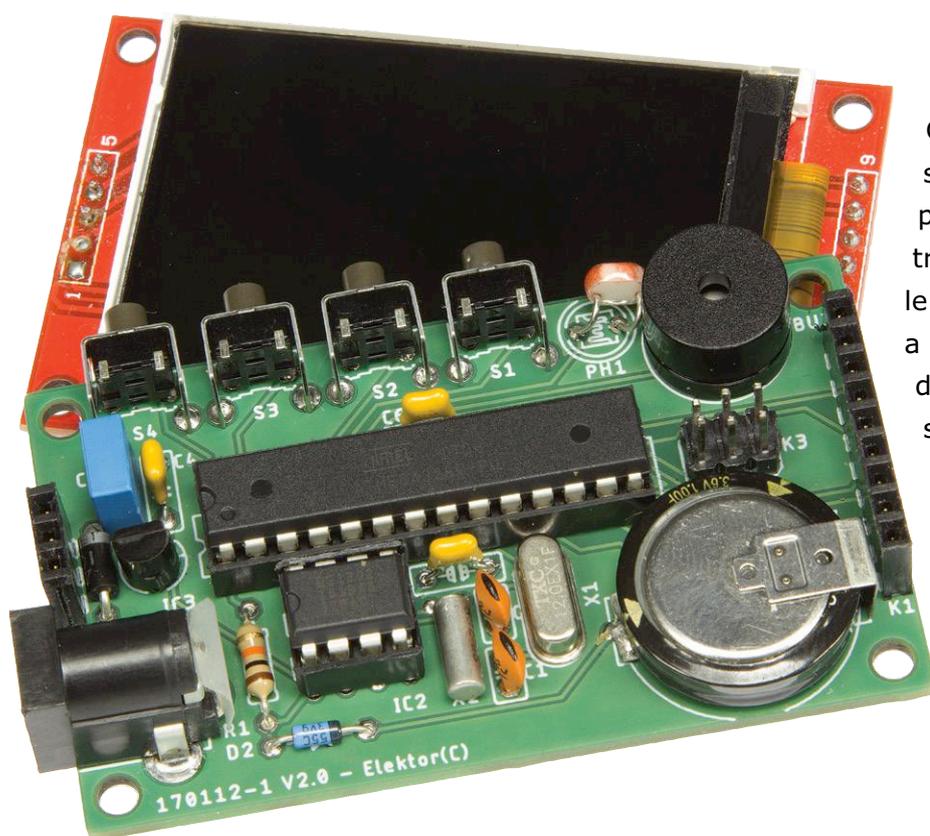


Figure 6. Oscillogramme de la tension sur le curseur du potentiomètre motorisé.

mise à jour pour le réveil à triple affichage

version 2.6.1 du logiciel

Mathias Claussen et Thijs Beckers (labo d'Elektor)



Certains lecteurs ont exprimé le souhait de pouvoir programmer plusieurs alarmes sur le « réveil à triple affichage » (projet publié dans le numéro de mai/juin 2018 [1]). Qui a envie d'être réveillé tôt le matin le dimanche ou un jour de congé ? « Il suffit de modifier le logiciel pour y ajouter ces alarmes, ce n'est pas sorcier ! », nous direz-vous. C'est vite dit ! Si vous avez déjà essayé de modifier un code existant – et écrit par quelqu'un d'autre – vous le savez bien, c'est loin d'être évident...

Quoi de neuf ?

On peut désormais programmer neuf alarmes sur le réveil. Pourquoi neuf ? C'est joli sur l'écran, et c'est plus qu'assez pour le programme d'une semaine entière. On peut aussi désactiver la variation automatique de la luminosité de l'écran, et régler trois niveaux fixes de luminosité.

Il y a maintenant deux menus pour les réglages : l'un pour l'heure et le rétroéclairage (**fig. 1**), l'autre pour les alarmes (**fig. 2**). Le bouton S1 active le premier menu. L'ajout de l'objet `Backlight` per-

met de sélectionner la luminosité variable automatique ou un niveau fixe ; ce menu permet en outre de calibrer la photorésistance (LDR) qui détecte la luminosité ambiante. Le bouton S2 active le menu des alarmes.

Nous avons profité de la révision du programme pour en optimiser certaines parties du code : l'affichage, l'emploi de la mémoire vive (RAM) et le temps de calcul. Si les détails ne vous intéressent pas, vous savez maintenant ce qui a été modifié et vous n'avez pas besoin d'aller plus loin, sinon lisez la suite de l'article.

Rétroéclairage et boutons

L'utilisation de `Timer1` pour le rétroéclairage a été expliquée en détail dans le numéro de mai/juin. Il est possible de simplifier la procédure en utilisant la bibliothèque `TimerOne` plutôt que d'écrire directement dans les registres de `Timer1`. La programmation de `Timer1` et la production du signal MLI (`PWM`) pour le rétroéclairage sont facilitées :

```
• Timer1.initialize (1000);  
  // initialise Timer1 et fixe la  
  période à 1000 µs
```

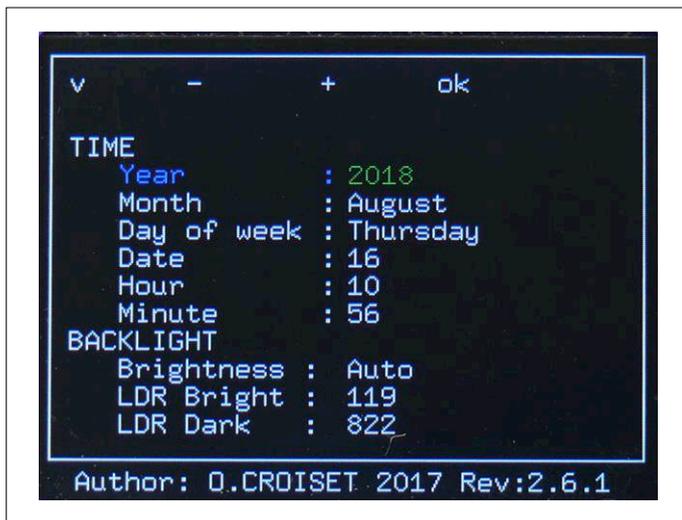


Figure 1. Le réglage du réveil passe désormais par deux menus. Le premier permet le réglage de l'heure et de la luminosité de l'écran.

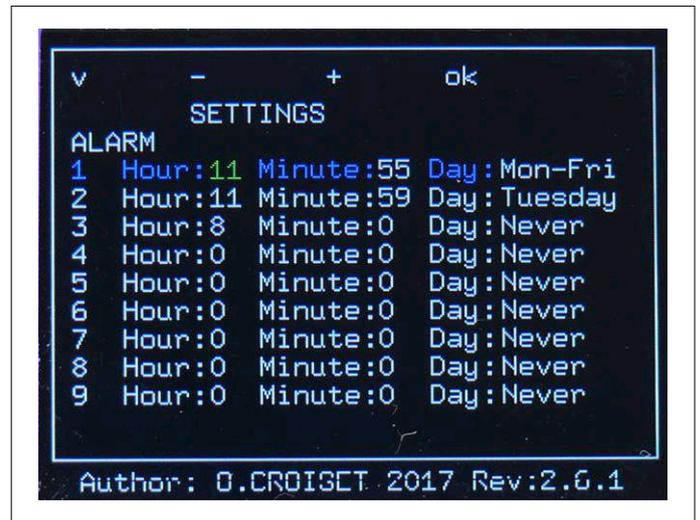


Figure 2. Le second menu sert à régler les neuf alarmes.

- `Timer1.pwm(9, 512);`
// sortie MLI sur la broche 9,
rapport cyclique de 50%
- `Timer1.attachInterrupt(Callback);`
// interruption de dépassement de
Timer1

`Timer1.initialize` spécifie la période en microsecondes et le code de la bibliothèque la convertit en valeur adaptée aux registres du microcontrôleur. Le rapport cyclique peut varier entre 0 et 100%, soit une valeur numérique de 0 à 1023 ; 512 correspond à un rapport cyclique de 50%. La troisième ligne permet la gestion d'une interruption en cas de dépassement, c'est-à-dire lorsque `Timer1` atteint la fin d'une période. La fonction `Callback()` est alors appelée. Comme cet appel a lieu 1 000 fois par seconde, la fréquence du signal MLI est de 1 000 Hz. Il est évident qu'à cette fréquence l'œil humain ne perçoit aucun scintillement de l'écran. Le buzzer du réveil est activé pendant une seconde, puis éteint pendant la même période. La fonction `Callback()`, appelée lors de chaque interruption de `Timer1`, veille à la mise en route et à l'arrêt stable du buzzer toutes les secondes.

Temporisation des alarmes

Le bouton S2 active le menu des alarmes. Ce bouton sert aussi au rappel de l'alarme, 5 min après son arrêt. Chaque alarme peut être programmée individuellement pour un jour de la semaine et une heure. En activant *Mon-Fri*, l'alarme se répétera chaque jour du lundi au vendredi ; *Never* désactive l'alarme.

Pour déterminer quand une alarme doit être déclenchée, le programme calcule le temps restant avant le déclenchement des différentes alarmes programmées. L'alarme la plus proche dans le temps est affichée sur l'écran comme étant la prochaine. Cela semble trivial, mais il en va tout autrement dans la pratique. Prenons un exemple pour illustrer nos propos : considérons que nous sommes un vendredi à 11h56 et que les alarmes du tableau ci-après ont été programmées.

Alarme	Heures	Minutes	Jours
1	11	55	Mon-Fri
2	11	59	Tue
3	08	00	Never

L'alarme 1 se déclenche chaque jour, du lundi au vendredi, il faut donc savoir quel jour on est. Si on est samedi ou dimanche, on peut supposer que l'alarme est pour le lundi suivant, sinon c'est pour le jour même.

Vendredi, 11h56 : l'alarme est pour aujourd'hui (en réalité elle doit déjà avoir été déclenchée). Le programme décompte les jours (0 dans le cas présent), les heures (0 aussi), puis les minutes (-1). Ce dernier résultat signifie que l'alarme a déjà été déclenchée et qu'il faut corriger en conséquence.

On retire 60 min (1 h) aux heures et on l'ajoute aux minutes, ce qui nous donne 59 min. Le compteur d'heures était à 0, on obtient -1 h. On doit corriger cela et on retire 24 h (1 jour) du compte de jours. Résultat : 23 h, ce qui rend les

heures à nouveau positives, mais pas les jours (-1 jour). Cela se complique un peu : si on est lundi, mardi, mercredi ou jeudi, on remplace simplement le -1 par 0, mais si on est vendredi on doit ajouter 3 jours. On arrive ainsi à 2 jours, 23 h et 59 min avant un nouveau déclenchement de l'alarme. Pour faciliter les calculs, on convertit le tout en minutes, ce qui nous donne 4 319 min.

C'est plus facile pour l'alarme 2, qui n'est active qu'un seul jour de la semaine. On calcule d'abord le nombre de jours entiers : nous sommes vendredi, l'alarme est prévue pour mardi, soit -3 jours. L'alarme ne se déclenchant qu'une fois par semaine on ajoute 7 jours, le déclenchement est donc dans 4 jours. L'alarme est à 11h, même heure que maintenant, la différence est nulle ; elle est à 59 min, soit 3 min plus tard que maintenant. Résultat final : 4 jours, 0 h et 3 min, soit 5 763 min.

C'est encore plus simple pour la troisième alarme, qui ne doit jamais se déclencher. On considère 65 535 min, ce qui est plus long qu'une semaine et donc invalide. On obtient alors le résultat suivant.

- Alarme 1 : 4 319 min.
- Alarme 2 : 5 763 min.
- Alarme 3 : 65 535 min

L'alarme 1 est donc bien celle qui doit se déclencher en premier.

Optimisation

Le code du microcontrôleur AVR a besoin d'environ 30 Ko de la mémoire flash, sur



Figure 3. L'affichage semble simple pour nous... mais pas pour le microcontrôleur !



Figure 4. L'affichage des secondes est gourmand en capacité de calcul.

les 32 disponibles, et de 1 192 octets de mémoire vive sur les 2 000 à ne pas dépasser. Une modification du programme pour rendre ces valeurs moins critiques ne fera donc pas de mal. En examinant le croquis, on note l'utilisation suivante des variables :

```
double-x1Trait;
// internal circle Cinqmin x
double-y1Trait;
// internal circle Cinqmin y

float x1precH = 120;
float x2precH = 120;

int myYear;
int myMonth;
```

C'est facile pour nous de travailler avec des nombres entiers simples ou doubles, ou réels en virgule flottante, mais ce n'est pas le plus efficace pour le microcontrôleur (consommation de RAM). Les architectures modernes de microprocesseurs traitent les entiers doubles sur 64 bits et les réels en virgule flottante sur 32 bits. Les compilateurs AVR quant à eux « nivellent par les bas », et l'utilisation de l'un ou l'autre type de donnée n'a guère d'influence sur la précision. On déclare souvent beaucoup de variables de type entier (`int`). Ces variables sont codées sur deux octets et peuvent prendre une valeur comprise entre -32 868 et +32 767. Il ne faut pas réfléchir plus avant pour comprendre qu'une optimisation du code permettra de gagner non seulement en temps de calcul, mais aussi en utilisation des mémoires flash et vive.

Lors de l'écriture d'un programme pour

un microcontrôleur, on essaie toujours de respecter la taille des registres, qui est de 8 bits (un octet) pour les AVR. On peut dans le cas des AVR ajouter, soustraire ou multiplier (uniquement pour les ATmega) deux octets en un cycle d'horloge. Si les variables sont codées sur deux octets ou plus, le compilateur doit tout ramener à des calculs sur un seul octet ; les opérations avec des entiers doubles ou des réels en virgule flottante deviennent vite très gourmandes en nombre de cycles d'horloge. Il en faut des milliers pour des fonctions trigonométriques !

Limiter l'utilisation des fonctions trigonométriques

La bibliothèque `ucglib` facilite l'écriture du code pour piloter l'écran, mais l'exécution du code consomme beaucoup de ressources, et le processus de dessin sur l'écran n'est pas des plus rapides. Le calcul de la position des aiguilles ou des segments indiquant les secondes (**fig. 3**) exige de nombreux appels aux fonctions trigonométriques ; les coins des deux aiguilles demandent à eux seuls le calcul de huit positions, une par coin ! Quand on sait qu'un AVR a besoin de quelque 1 650 cycles d'horloge pour un seul sinus ou cosinus...

Examinons par exemple la fonction `SecondeSecteur`, qui dessine le cercle des secondes (en bas à gauche sur la **figure 4**), avec un segment toutes les 5 s. L'examen du code montre que la fonction `sin()` est appelée quatre fois, et la fonction `cos()` aussi :

```
x_ext_Sec = ext_radius *
cos(angleSec); x_ext_Sec = x_
```

```
ext_Sec + xcenter ;
y_ext_Sec = ext_radius *
sin(angleSec); y_ext_Sec = y_
ext_Sec + ycenter ;
x_int_Sec = int_radius *
cos(angleSec); x_int_Sec = x_
int_Sec + xcenter ;
y_int_Sec = int_radius *
sin(angleSec); y_int_Sec = y_
int_Sec + ycenter ;
x_ext_SecPrec = ext_radius *
cos(angleSecPrec); x_ext_
SecPrec = x_ext_SecPrec +
xcenter ;
y_ext_SecPrec = ext_radius *
sin(angleSecPrec); y_ext_
SecPrec = y_ext_SecPrec +
ycenter ;
x_int_SecPrec = int_radius *
cos(angleSecPrec); x_int_
SecPrec = x_int_SecPrec +
xcenter ;
y_int_SecPrec = int_radius *
sin(angleSecPrec); y_int_
SecPrec = y_int_SecPrec +
ycenter ;
```

Les variables sont toutes de type entier, et la multiplication de deux entiers dure 20 cycles. On arrive donc à un total de $(1\ 650 \times 8) + (20 \times 8) = 13\ 360$ cycles, et c'est sans compter d'autres calculs ! Avec une fréquence d'horloge de 12 MHz, cela prend 1,11 ms ; et il faut calculer la position de douze segments toutes les 5 s... On peut pourtant passer de quelques milliers de cycles d'horloge à moins de 100 : il suffit d'effectuer les calculs à l'avance et d'entreposer les résultats dans une table de correspondance.

Certes, cela ne représente pas une économie spectaculaire en temps de calcul, mais ça illustre parfaitement les possibilités d'optimisation. Nous avons utilisé la même méthode pour la fonction `AiguilleSecondes`.

Pour les aiguilles indiquant les heures et les minutes, il nous faudrait une table avec $(8 \times 60) + (8 \times 12 \times 60) = 6\,240$ valeurs (l'aiguille des minutes change de position toutes les minutes). Là, la taille de la table deviendrait excessive...

Surtout pas de doublons

Les fonctions pour le dessin des divers éléments de l'écran peuvent être exécutées séparément, mais il est possible de combiner la production d'éléments semblables. C'est ce que nous avons fait pour les fonctions `SegmentA()` à `SegmentG()`. Une seule fonction suffit, seule la couleur des segments doit être paramétrée. Cependant n'allez pas croire que c'est toujours simple, sinon l'auteur l'aurait fait directement. Certaines modifications demandent une réécriture du code en profondeur.

Nous avons aussi modifié `Cadran2()`, et remplacé la structure à branchements multiples pour les heures et minutes par une fonction d'aide. La taille du code en est nettement réduite.

Il faut éviter les répétitions inutiles, comme redessiner un chiffre déjà affiché. Pourquoi recalculer la position de l'aiguille des heures à chaque cycle si elle ne change que toutes les quelques minutes ? Deux solutions sont possibles pour la commande de cette fonction : travailler avec des variables globales ou des variables statiques. Nous avons choisi la seconde : si l'heure effective diffère de l'heure affichée, seule la partie à modifier est recalculée ; c'est beaucoup plus efficace et satisfaisant.

Déboguer sans déboguer

Bien entendu, il faut tester les modifications, mais l'EDI de l'Arduino ne permet pas vraiment le débogage. On peut brancher des LED (avec une résistance

en série) sur les broches d'E/S du microcontrôleur et en vérifier l'état à certaines étapes du programme.

On peut aussi utiliser l'interface USART, si la broche 3 (Tx) du microcontrôleur n'est pas utilisée, et transmettre un texte via `Serial.print()`. C'est a priori pratique, mais une vitesse de transmission de 9 600 bauds ralentit visiblement le réveil. Avec une fréquence d'horloge de 12 MHz, on peut grimper à 250 000 bauds, ce qui est nettement meilleur, et qui reste dans les limites acceptables par les circuits convertisseurs USB-série CH340 ou CH341.

Nous avons finalement utilisé une combinaison USART et broche d'E/S avec un analyseur logique. Les modifications de l'état de la broche étaient ainsi visibles, de même que le comportement temporel. Les bogues peuvent être localisés avec un listing obtenu via `printf()`.

Si le débogage n'est plus nécessaire, on peut supprimer chaque `printf()` du programme, ou utiliser le préprocesseur :

```
#define DEBUGPRINT( X ) Serial.  
    print( X )  
//#define DEBUGPRINT( X ) do{ }  
    while( 1 ==0)
```

Pour activer le débogage, on utilise la première ligne : `DEBUGPRINT("Test")` permet d'imprimer le message "Test". Pour arrêter le débogage, il suffit d'utiliser la seconde ligne ; la boucle `do-while` veille à ce que rien ne soit envoyé vers l'USART. L'optimisation pas à pas rend difficile l'uti-

lisation d'un outil professionnel comme l'AVR Dragon. Lors du processus pour rendre le programme le plus petit possible, le compilateur s'assure seulement que le résultat des fonctions est correct. Il prend lui-même des mesures et nous ignorons comment il arrive à ce résultat. Notre programme peut être totalement chamboulé et certaines fonctions peuvent même être supprimées.

Conclusion

L'utilisation du réveil est désormais plus simple, et les possibilités d'alarmes devraient satisfaire les plus exigeants d'entre vous. L'option d'étalonnage de la photorésistance s'est avérée nécessaire à cause de la dispersion des caractéristiques du composant : le réglage de la luminosité n'était pas correct avec certains exemplaires.

Nous avons levé le voile sur certains aspects de la programmation du microcontrôleur, et cela vous servira sans doute dans d'autres applications. Le nouveau logiciel est disponible sur la page du projet [2].

(170112-B-04 -

version française : Jean-Louis Mehren)



@WWW.ELEKTOR.FR

- Circuit imprimé, réf 170112-1
www.elektor.fr/pcb-170112-1
- µC programmé, réf. 170112-41
www.elektor.fr/microcontroller-170112-41
- Écran TFT de 2,2", SPI, 240×320 pixels
www.elektor.fr/spi-tft-display-240x320
- Kit avec circuit imprimé nu, écran TFT, µC programmé et tous les autres composants
www.elektor.fr/3-way-alarm-kit

Liens

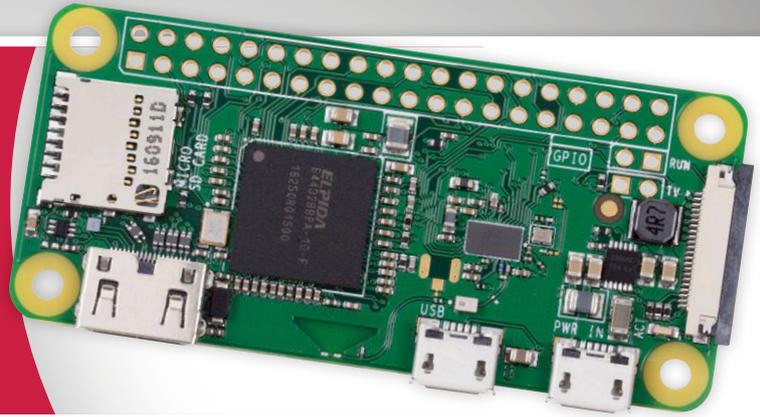
[1] Réveil à triple affichage, Elektor 05-06/2018 : www.elektormagazine.fr/160590

[2] Page de cet article : www.elektormagazine.fr/170112-B-04

[3] Page du projet au labo : www.elektormagazine.fr/labs/3-displays-alarm-clock-with-tft-screen-1

ABONNEZ-VOUS ET RECEVEZ

RPI ZERO W GRATUIT



SEULEMENT
54,95 €
PAR AN
(6 NUMÉROS)

TOUS LES 2 MOIS, LES DERNIÈRES NOUVELLES DU RASPBERRY PI ET LES MEILLEURS PROJETS !

Souscrivez dès maintenant un abonnement d'un an au magazine MagPi, nous vous offrons :

- Six numéros du magazine MagPi
- Une carte Raspberry Pi Zero W
- Un boîtier avec trois couvercles différents
- Un connecteur pour module de caméra
- Un câble HDMI/mini-HDMI et un câble micro-USB/USB OTG

Vos avantages :

- Prix au numéro réduit
- Chaque numéro directement dans votre boîte aux lettres
- Tous les numéros disponibles sous forme numérique (PDF)
- Cadeau de bienvenue d'une valeur de 22,95 €
- Découverte de chaque nouveau numéro avant sa sortie en kiosque



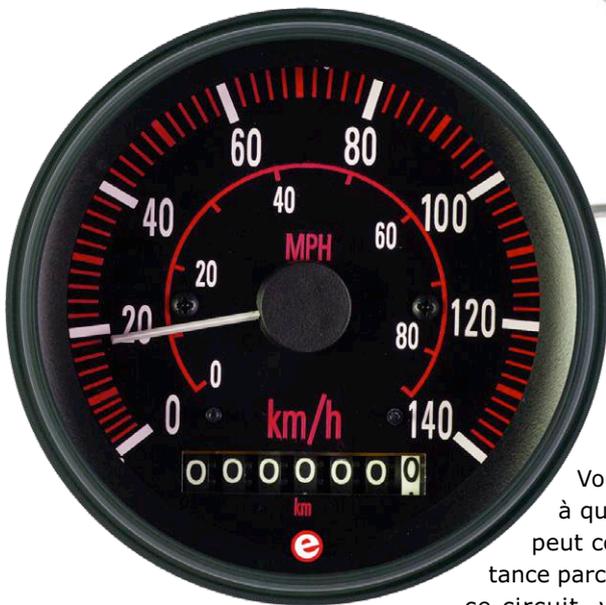
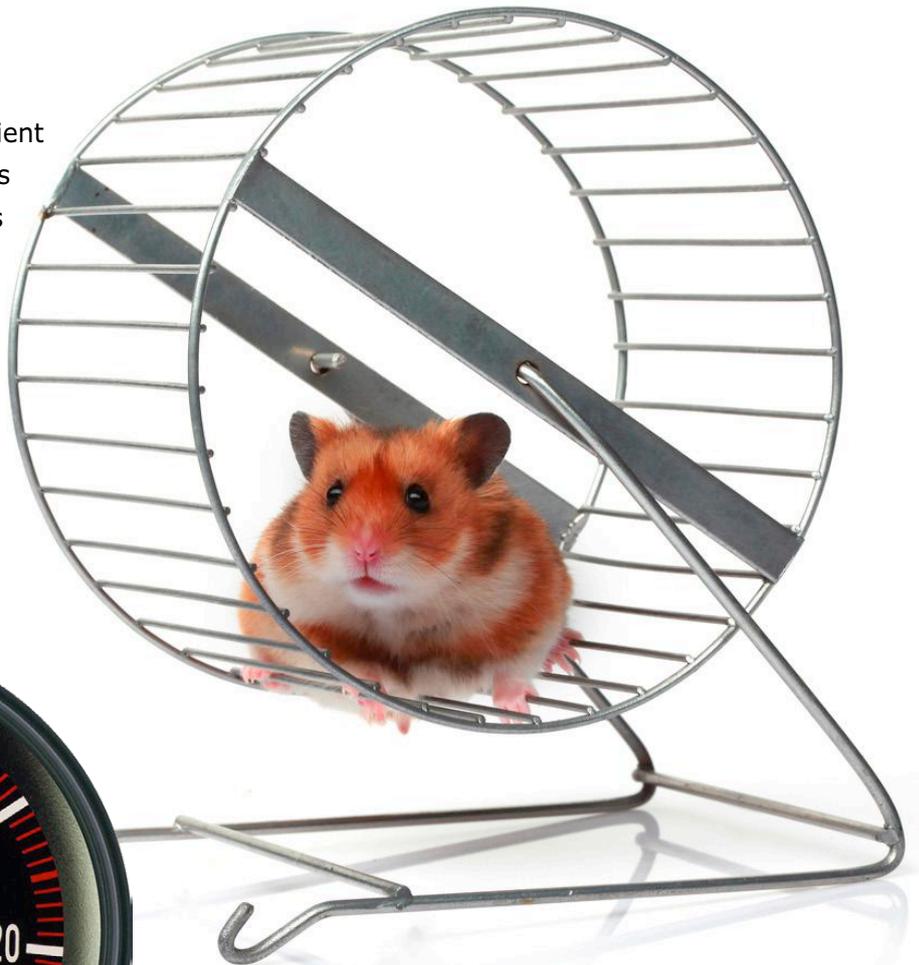
ABONNEZ-VOUS : WWW.MAGPI.FR

run-O-mètre pour hamster

mesure et affichage de la vitesse et de la distance parcourue

Willem Tak

Si vous avez un hamster, sa cage contient probablement une roue d'exercice dans laquelle il peut courir à tire-larigot. Les électriciens connaissent bien ce genre de cage d'écureuil, mais comme rotor de moteur à induction. C'est aussi un terrain d'expérimentation pour les électroniciens. Ce montage simple vous donnera une idée de la vitesse atteinte et de la distance parcourue par votre hamster.



Caractéristiques

- Enregistre le chemin parcouru et la vitesse maximale
- Rétro-éclairage réglable
- Alimentation entre 8 et 15 V=
- L'EEPROM peut stocker 15 mesures
- Maniement simple par codeur rotatif

Vous êtes-vous déjà demandé à quelle vitesse votre hamster peut courir ? Ou bien quelle distance parcourt-il dans sa roue ? Avec ce circuit, vous pouvez mesurer sa vitesse maximale, ainsi que la distance totale parcourue pendant toute la durée de la mesure. Cette durée est également sauvegardée, ce qui permet de calculer la vitesse moyenne.

Le projet

L'idée est de ne pas trop modifier la cage ou la roue du hamster. C'est pourquoi nous fixons deux capteurs de réflexion

IR et un bout de matériau réfléchissant (feuille d'alu) sur la roue pour en compter les tours. Deux LED IR fournissent le rayonnement nécessaire. Sur le prototype, nous n'avions utilisé qu'un capteur, mais nous en avons vite ajouté un second pour éviter les erreurs quand le hamster ne fait que se balancer. Avec deux capteurs, seules les rotations complètes sont totalisées.

Le matériel

Le matériel nécessaire est représenté dans le schéma de la **figure 1**. La puce IC3 communique les données au run-

INFOS SUR LE PROJET



mesure enregistrement

PIC18F26K22

LED IR



débutant

connaissseur

expert



env. 1 h



matériel de soudage de composants traversants



env. 40 €

O-mètre basé sur un microcontrôleur PIC18F26K22. Dans son EEPROM, il n'y a de la place que pour quinze mesures. Chacune d'entre elles contient la durée de la mesure, la distance totale et la vitesse de pointe. Le connecteur K3 est compatible avec l'interface de programmation Pickit de Microchip et peut servir à programmer le microcontrôleur. L'initialisation du PIC est lancée par le poussoir S1. Les réglages et les données s'affichent sur un écran LCD standard à 2 lignes de 16 caractères. On en règle le contraste avec P2 et le microcontrôleur allume et éteint le rétroéclairage via le transistor T2. Le bouton rotatif avec bouton-poussoir intégré ENC1 sert à changer l'affichage et les réglages.

Les LED IR en parallèle sont alimentées entre les broches 4 et 5 de K2 (VCC sur les anodes et la tension de commande sur les cathodes), intensité réglée par P1. Le transistor T1 conduit le courant pour

soulager le PIC.

Pour le prototype au labo d'Elektor, nous avons utilisé deux capteurs de réflexion TCRT5000-IR pour détecter la rotation de la roue, mais à peu près toute autre combinaison de LED IR et détecteur conviendrait (figure 5). Le signal produit par le PIC est carré, modulé à 38 kHz, donc un capteur de télécommande TSOP4838 (ou tout autre type à 38 kHz) de concert avec une LED IR (par exemple une CQY99) le fera aussi bien. Cette dernière solution est un peu plus volumineuse et plus compliquée à installer, mais la modulation rend les capteurs beaucoup moins sensibles aux autres sources d'infrarouge. Les émetteurs du TCRT5000 sont raccordés à la masse (broche 3 de K2). Le collecteur d'un capteur va sur la broche 1 de K2, l'autre sur la broche 2. Les amplificateurs opérationnels IC4.A et IC4.B conditionnent les signaux de sortie des détecteurs pour les triggers de Schmitt

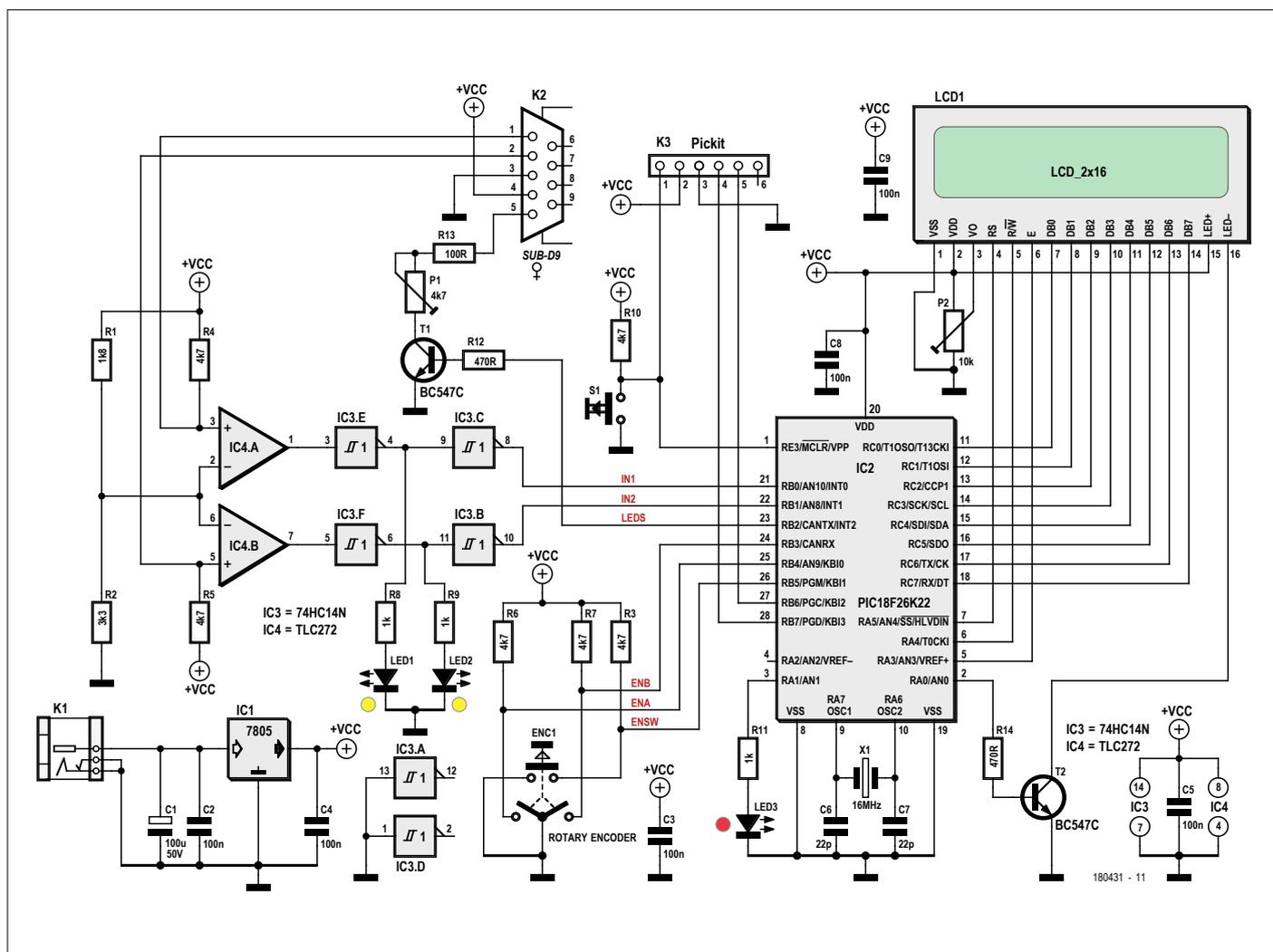


Figure 1. Le microcontrôleur PIC commande le LCD et s'occupe des mesures. On manie ENC1 pour les réglages.

de IC3 et ensuite le microcontrôleur. La LED3 clignote lors de la mise sous tension, puis s'allume et s'éteint après chaque tour de la roue. Les deux autres (LED1 et LED2) donnent l'état des capteurs. Elles s'allument lors de la réception du rayonnement IR réfléchi. Cette indication est également transmise au LCD dans l'option « Sensorstatus » (nous y reviendrons), mais les LED réagissent plus vite que l'écran, relativement lent. La stabilisation de la tension d'alimentation est assurée par IC1, un 7805 en boîtier TO-220. On peut ainsi utiliser un adaptateur secteur à partir de 8 V=. Question dissipation dans le 7805, mieux vaut ne pas dépasser 15 V.

Maniement

La structure du menu du run-O-mètre pour hamster est à la **figure 2**. On manipule le circuit par une combinaison de rotations (lignes rouges) et de pressions (lignes bleues) sur le bouton du codeur optique. La ligne verte est une transition automatique (retardée).

Si le LCD est en service, il indique **Hamster Run-O-Meter**. Quand on appuie sur le bouton du codeur, le premier pas du menu s'affiche : **Start measuring**. Vous pouvez redémarrer la mesure en poussant à nouveau sur le bouton ou bien tourner le codeur pour atteindre les autres propositions du menu principal. Dans le menu **Controls** se trouvent deux possibilités de réglage et une de test. Le premier réglage consiste à donner le diamètre de la roue, entre 12 et 35 cm. Une pression sur le bouton confirme le réglage qui est alors inscrit dans l'EEPROM. Le deuxième réglage est la sélection de l'éclairage d'écran **Measurement Backlight on/off**. Comme les hamsters préfèrent courir la nuit, il peut être ennuyeux de laisser le rétroéclairage du LCD allumé toute la nuit si la mesure se poursuit. Cette option éteint automatiquement l'éclairage du LCD quand une mesure commence et le rallume quand elle se termine. Tourner le codeur permet de choisir **on** ou **off**. Ce réglage est enregistré dans l'EEPROM quand on appuie sur le bouton du codeur. À propos, dans le menu, avec l'option **Switch backlight**, on peut éteindre complètement l'éclairage de fond ; seul le LCD bleu que nous avons utilisé est très difficile à lire.

L'écran montre ensuite **Sensor status** sur la première ligne et deux fois **on/off** sur la seconde ligne (une indication par capteur). Si un capteur ne reçoit pas

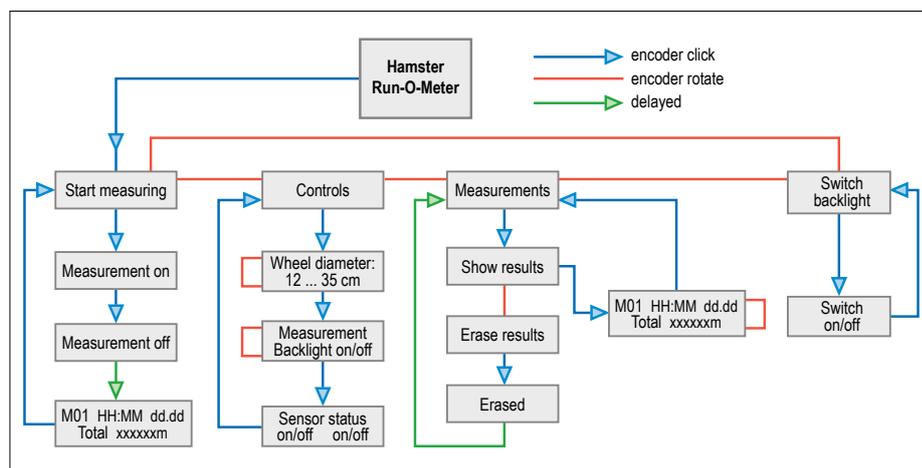


Figure 2. La structure du menu montre bien comment se servir du Run-O-mètre pour hamster.

de rayonnement en retour, c'est **off** qui s'affiche ; s'il en reçoit, le texte est **on**. Vous pouvez de cette manière savoir si chaque capteur travaille bien en faisant tourner la roue vous-même.

Les mesures

L'option suivante du menu est **Measurements**. Après avoir poussé sur le bouton, tourner le codeur donne sur le LCD **Show results** ou **Erase results**. Si vous poussez encore sur le codeur pendant que **Show results** est à l'écran, vous pouvez lire un maximum de quinze mesures enregistrées. Voici les données qui s'affichent :

- le numéro de la mesure, **M_x** (x va de 1 à 15)
- la durée de la mesure au format hh:mm (minimum 00:01 et maximum 23:59)
- la vitesse maximale en xx.xx km/h (les vitesses < 0,50 km/h ne sont pas enregistrées)
- la distance totale parcourue en mètres (**Total xxxxxx m**).

En tournant le codeur, le LCD parcourt les quinze mesures et en poussant dessus, vous retournez au menu **Measurements**. Si vous poussez sur le bouton alors que **Erase results** est affiché, toutes les mesures s'effacent.

La dernière option du menu principal est **Switch backlight** qui permet d'allumer et d'éteindre l'éclairage d'arrière-plan.

Câblage du circuit imprimé

Comme tous les composants sont du type traversant, le soudage n'est pas trop ardu (cf. **figure 3**). Commencez

par souder les résistances, puis les supports de puces, les condensateurs, etc. Le connecteur K3 pour brancher l'interface de programmation n'est nécessaire que si vous n'avez pas le PIC préprogrammé ou si vous voulez expérimenter avec votre version du micrologiciel. Il est assez pratique de souder ce connecteur sur l'envers de circuit imprimé.

Pour le LCD, soudez un support à 16 contacts côté composants du circuit imprimé, l'embase à 16 contacts vient à l'arrière de l'afficheur. Fixez les quatre entretoises de 10 mm sur le circuit imprimé, insérez avec précaution l'écran dans l'embase et vérifiez que le LCD est bien aligné sur les entretoises (peut-être bien que non). Pour éviter d'endommager l'écran, ajoutez des rondelles avant le vissage pour que le LCD repose convenablement sur les entretoises.

Installation mécanique

Collez une petite bande (assez grande quand même pour réfléchir le faisceau IR vers le détecteur) de feuille d'aluminium ou d'un matériau réfléchissant à l'extérieur de la roue, de manière à ce que les capteurs puissent être placés juste hors de la cage (par définition, les rongeurs rongent... et les câbles n'y résistent pas). Placez les capteurs de préférence à environ 180° l'un de l'autre, à distance d'un demi-tour de la roue (cf. **figure 4**). Le montage des capteurs dépend du type utilisé. Première solution : chaque capteur se compose d'une LED IR (p.ex. une TSUS5400 ou une CQY99, de préférence avec réflecteur) et d'un récepteur IR TSOP4838. Montez-les l'un à côté de l'autre sur un bout de platine perforée.



LISTE DES COMPOSANTS

Résistances :

R1 = 1,8 k Ω
R2 = 3,3 k Ω
R3, R4, R5, R6, R7, R10 = 4,7 k Ω
R8, R9, R11 = 1 k Ω
R12, R14 = 470 Ω
R13 = 100 Ω
P1 = 4,7 k Ω , potentiomètre ajustable horizontal
P2 = 10 k Ω , potentiomètre ajustable horizontal

Condensateurs :

C1 = 100 μ F/50 V, au pas de 3,5 mm
C2, C3, C4, C5, C8, C9 = 100 nF/50 V, X7R, au pas de 5,08 mm
C6, C7 = 22 pF/50 V, COG/NP0, au pas de 2,5 mm

Semi-conducteurs :

IC1 = MC7805, 5 V, 1 A
IC2 = μ C à 8 bits PIC18F26K22, programmé (réf. 180341-1)

IC3 = 74HCT14, inverseurs à trigger de Schmitt
IC4 = TLC272CP, double amplificateur opérationnel LinCMOS
IC5, IC6 * = TCRT5000, capteur IR à réflexion (PAS sur la carte)
IC5, IC6 * = TSOP4838, récepteur pour télécommande à 38 kHz (PAS sur la carte)
LED1, LED2 = LED jaune, 3 mm
LED3 = LED rouge, 3 mm
LED4, LED5* = LED IR TSUS5400 (PAS sur la carte)
T1, T2 = BC547C, 45 V, 100 mA, 500 mW, hfe=400

Divers :

LCD1 = LCD de 2x16 caractères avec rétroéclairage (réf. 120061-74)
LCD1' = support à 1x16 contacts, vertical
LCD'' = embase à 1x16 broches, verticale
S1 = bouton-poussoir, 24 V, 50 mA, 6x6 mm
ENC1 = EC12E2424407, codeur rotatif

incrémental avec bouton-poussoir, 12 mm, vertical
X1 = quartz 16 MHz
K1 = connecteur d'alimentation, diam. = 1,95 mm encartable
K2 = connecteur sub-D à 9 contacts, encartable, femelle
K3* = embase à 6 broches, au pas de 2,5 mm
Support de puce DIP 28, étroit (IC2)
Support de puce DIP 14 (IC3)
Support de puce DIP 8 (IC4)
Connecteur sub-D à 9 broches, mâle, avec fixation de câble
Entretoises à vis M3 x 10 mm
Vis M3 x 5 mm
Rondelles M3
Écrous M3
Circuit imprimé, réf. 180431-1

* = cf. texte

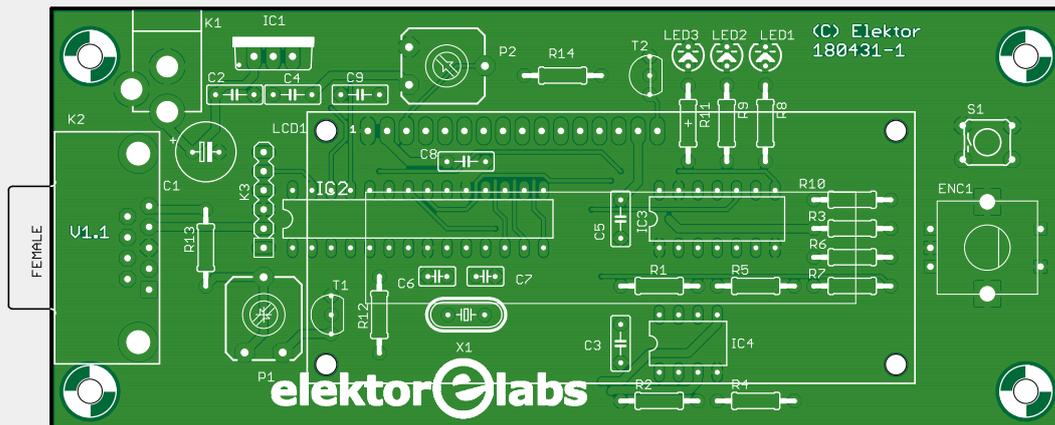


Figure 3. Avec ce prototype, vous verrez en détail comment câbler ce montage.

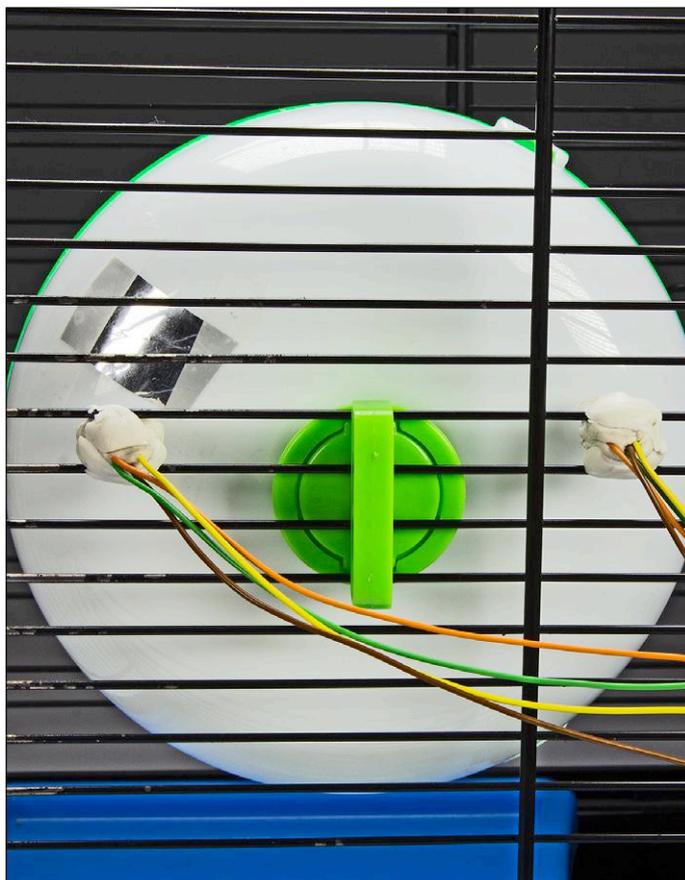


Figure 4. Installer les capteurs et les bandes réfléchissantes de cette manière sur la roue permet d'atteindre la meilleure efficacité.

Inclinez-les légèrement l'un vers l'autre pour que le détecteur puisse bien capter le faisceau IR réfléchi. Il faut construire un mécanisme hors de la cage pour supporter ces circuits de capteurs. Évidemment, les faisceaux IR ne doivent pas être réfléchis par les barreaux de la cage. Un conseil : tenez compte de la sensibilité du TSOP4838 aux infrarouges émis par les télécommandes, donc ne zappez pas pendant les mesures.

La deuxième solution plus simple demande un peu de savoir-faire : il s'agit de monter les composants directement sur les barreaux de la cage. Dans ce cas, on utilise deux capteurs combinés aux TCRT5000. Leur sortie se fait sur le collecteur d'un phototransistor. C'est pourquoi nous avons eu recours aux amplificateurs opérationnels (IC4) et aux inverseurs à triggers de Schmitt (IC3) qui fournissent de vrais signaux numériques.

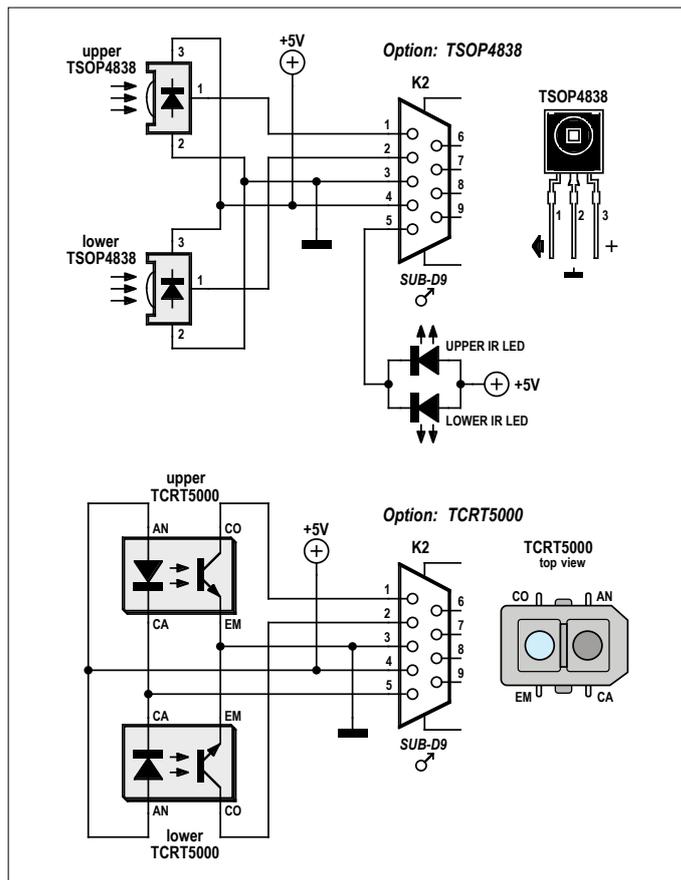


Figure 5. Le câblage dépendra du type de capteur choisi.

Selon notre expérience, la seconde méthode (TCRT5000) est moins fiable que celle à TSOP4838. La **figure 5** montre le schéma de câblage des deux formules.

Conclusion

Quelle que soit l'option choisie pour les capteurs, vous devrez expérimenter. Mécaniquement, on peut régler pas mal de choses en jouant sur l'emplacement et l'angle des capteurs, et surtout sur la distance à la roue. Électriquement, on peut par exemple faire varier le courant dans les LED avec le potentiomètre P1 pour améliorer les résultats.

Le logiciel pour le microcontrôleur, comme d'habitude, est gratuit et disponible au téléchargement sur la page de ce projet [1].

(180431-04 - version française : Robert Grignard)

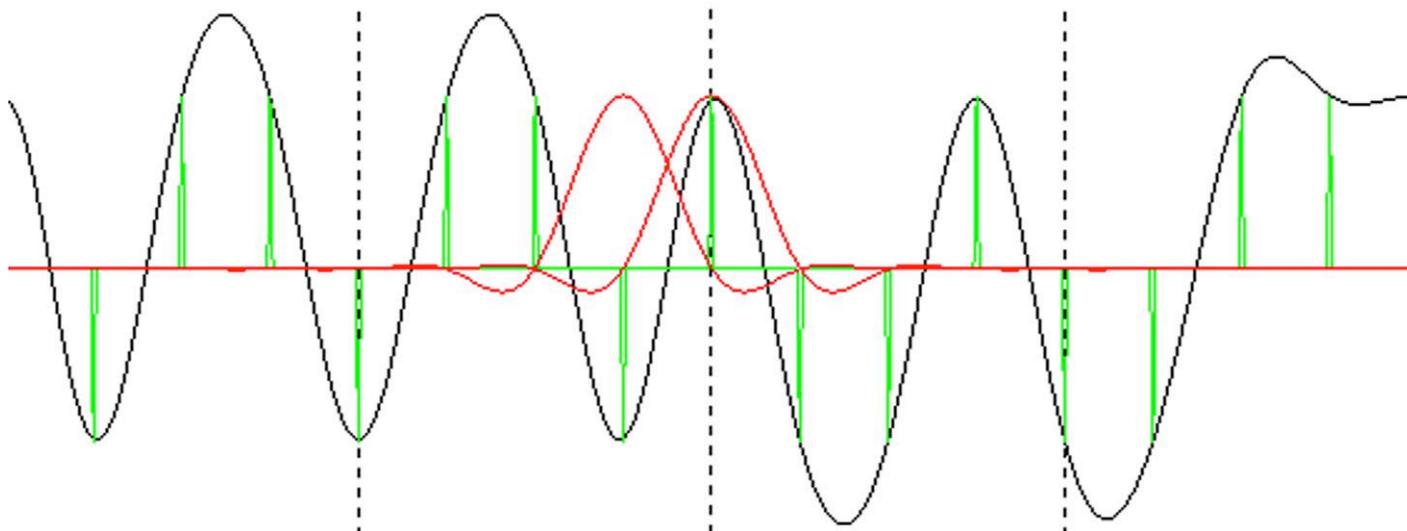
@WWW.ELEKTOR.FR

- Circuit imprimé, réf. 180431-1
www.elektor.fr/pcb-180431-1
- µC PIC PIC18F26K22-I/SP programmé, réf. 180431-41
www.elektor.fr/hamster-microcontroller
- LCD à 2×16 caractères avec rétroéclairage, réf. 120061-74 :
www.elektor.fr/lcd-120061-74

Lien

[1] Page de l'article : www.elektormagazine.fr/180431-04

techniques de filtrage de canal pour éviter le gaspillage de bande passante



Robert Lacoste (Chaville)

Cet article est une introduction au vaste sujet du filtrage de canal. L'idée est de filtrer un signal binaire entre sa source et l'émetteur pour réduire significativement la bande passante occupée sans augmenter le taux d'erreur. Vous découvrirez ici différentes techniques de filtrage plus ou moins efficaces.

Dans le précédent article [1], je vous ai présenté les principales techniques utilisées pour le « codage en ligne ». Pour mémoire, il s'agissait de choisir une bonne méthode pour transformer des '1' et des '0' en une tension analogique, appelée pompeusement « signal en bande de base », de manière à exploiter au mieux la bande passante disponible et à faciliter la vie du récepteur. Cette tension peut ensuite être envoyée à un correspondant éloigné directement sur une paire de fils ou en modulant une porteuse radiofréquence dans le cas de liaisons sans fil.

Pour simplifier, j'avais pris l'exemple d'une communication filaire ordinaire. Restons donc sur le même exemple et imaginons que vous ayez besoin de transmettre un million de bits par

seconde (1 Mbps) entre deux appareils distants. Toujours pour faire simple, supposons que vous décidiez d'utiliser la méthode de codage de canal la plus élémentaire baptisée NRZ. Pour mémoire, NRZ (*Non Return to Zero*) signifie qu'on envoie les bits tels qu'ils sont : une tension donnée pour représenter les '1' (par ex. 5 V), une autre tension pour les '0' (par ex. 0 V). C'est exactement ce que fait l'UART de votre microcontrôleur préféré (**figure 1a**). Cela fonctionne très bien, les milliers de projets utilisant des UART publiés dans cette revue en sont la preuve.

Pourquoi filtrer ?

Si cela fonctionne bien, pourquoi lire cet article où je vais vous expliquer qu'il est judicieux d'ajouter un filtre entre la sor-

tie de l'UART et le transmetteur ? Simplement pour une question d'efficacité spectrale. En d'autres mots, sans filtre, vous utiliseriez beaucoup trop de MHz pour votre transmission à 1 Mbps (je vous montre pourquoi ci-après). Vous me direz que ce n'est pas forcément un souci. Par exemple pourquoi s'en inquiéter si votre canal de transmission a beaucoup de capacité excédentaire ? Si vous utilisez un câble Ethernet de catégorie 6 (donné pour 1 Gbps) pour transmettre ces pauvres 1 Mbps sur quelques mètres, un filtrage sera inutile. C'est la même chose pour une communication entre deux puces sur un circuit imprimé. Par contre, dans la vraie vie, vous aurez souvent besoin de réduire le plus possible la bande passante nécessaire. Pourquoi ? Les raisons sont nombreuses : pour

utiliser un câble moins cher, pour des questions de compatibilité électromagnétique, pour des problèmes normatifs (largeurs des canaux radio), pour augmenter autant que possible le débit sans changer le tuyau de communication, etc.

De là on cherche souvent à n'utiliser que la largeur de bande nécessaire. Pour cela, on ajoute un filtre entre la source de données (par ex. l'UART) et l'émetteur (**figure 1b**). Le but de ce filtre est de réduire la bande passante occupée par le signal, en rabotant les angles, donc les fréquences hautes. Toute la difficulté est d'y parvenir sans affecter les caractéristiques du système en termes de débit binaire et de taux d'erreurs. Comment concevoir un tel filtre ? C'est ce que je vais essayer de vous expliquer ici.

Un petit coup de transformée de Fourier

Bon, vous avez maintenant l'habitude des allers-retours entre les domaines temporels et fréquentiels et vous savez que la transformée de Fourier est votre amie pour passer de l'un à l'autre. Pour vous rafraîchir la mémoire, ou pour les nouveaux lecteurs, jetez donc un œil à la **figure 2**. Les deux premiers graphes (**figure 2a**) montrent un signal sinusoïdal, ici à 500 kHz, et sa transformée de Fourier (c'est-à-dire son spectre en fréquence). Rien de très étonnant, il n'y a qu'une seule raie sur le spectre, qui correspond à la fréquence de 500 kHz. Qu'en est-il de la transformée de Fourier d'un signal carré de 500 kHz ? Regardez donc la **figure 2b**, son spectre comprend des raies à 500 kHz (la fréquence fondamentale), mais aussi à toutes les fréquences harmoniques impaires : 1500 kHz, 2500 kHz, etc., avec des amplitudes décroissantes. Comme d'habitude, tous ces exemples ont été réalisés à l'aide de petits scripts Scilab [2], téléchargeables depuis le site d'Elektor [3]. N'hésitez donc pas à les étudier et vous les approprier.

Passons maintenant à un exemple plus intéressant : quelle est la transformée de Fourier d'un seul bit de donnée transmis en NRZ et à 1 Mbps ? Ce signal n'est rien de plus qu'une impulsion de 1 µs de long. Sa transformée de Fourier a la forme étrange montrée sur la **figure 2c**. Vous reconnaissez cette forme, non ? Il s'agit de la fonction sinus cardinal, ou $\sin(x)/x$, dont j'ai parlé une dizaine de fois dans cette rubrique. Regardez-la encore une fois de près. Son spectre

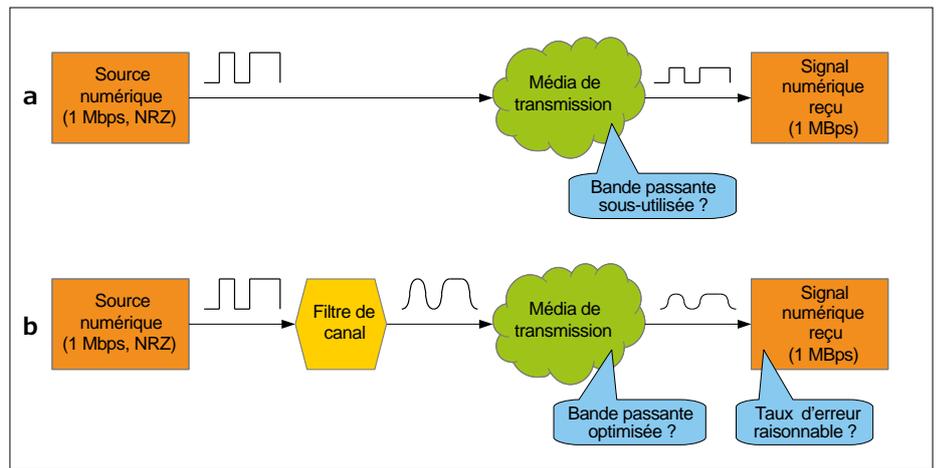


Figure 1. a) Transmettre directement un signal numérique brut, ici NRZ à 1 Mbps, est un gaspillage de bande passante. b) L'utilisation d'un filtre réduit la bande passante requise pour le même débit, mais avec un risque d'augmentation des erreurs de transmission.

en fréquence s'étend jusqu'à l'infini, et est formé de lobes successifs. Le premier, centré autour de 0 Hz, est deux fois plus large que les autres. La fonction s'annule entre chaque lobe, c'est-à-dire aux fréquences multiples de 1 MHz. Je vous encourage à jeter encore un œil à la figure 2 et à comparer ces trois paires de courbes, vous devriez y trouver des ressemblances intéressantes.

NRZ sans filtrage ?

Maintenant quel est le spectre d'un flux aléatoire de bits transmis en NRZ

et à 1 Mbps ? Si le signal est un '1' suivi d'une longue chaîne de '0', alors le spectre fréquentiel est identique à celui de l'impulsion de 1 µs que nous venons d'examiner, à savoir un $\sin(x)/x$ avec des zéros à 1 MHz, 2 MHz, etc., comme le montre la **figure 3a**. Dans le cas d'une séquence aléatoire, il y aura des chaînes de '1' et de '0' de toutes les longueurs, et tous les spectres correspondants s'additionneront les uns aux autres. La transformée de Fourier qui en résulte est assez complexe (**figure 3b**), mais sa forme générale est toujours une

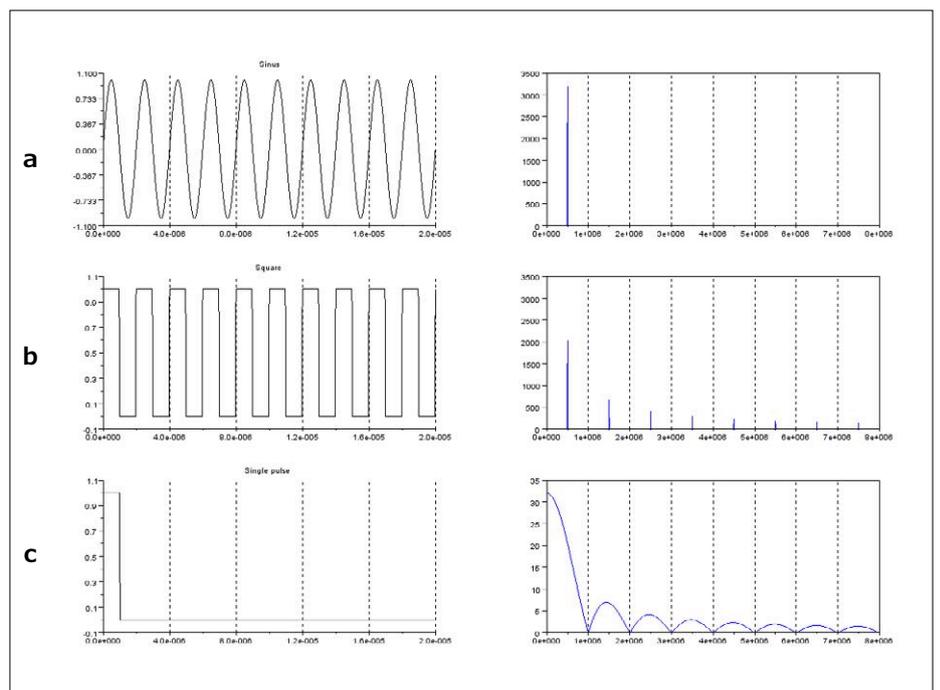


Figure 2. a) La transformée de Fourier d'une onde sinusoïdale est une raie unique à la fréquence de la porteuse. b) Un signal carré a des composantes spectrales à toutes les harmoniques impaires. c) Le spectre d'une impulsion unique est la fonction $\sin(x)/x$.

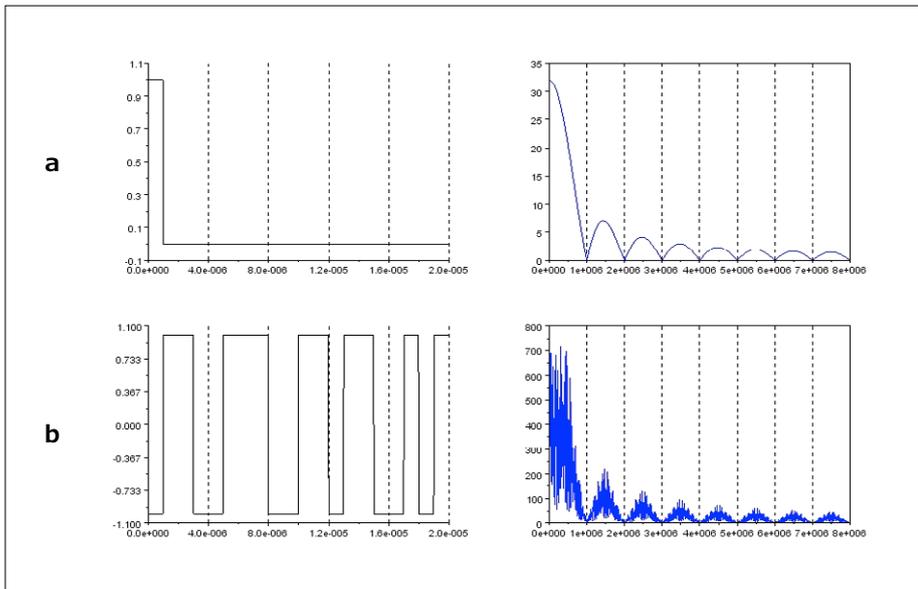


Figure 3. a) Le spectre d'une impulsion de 1 bit est un $\sin(x)/x$. b) Le spectre d'une séquence binaire aléatoire est plus complexe, mais a toujours la même forme en $\sin(x)/x$, avec des nuls à chaque multiple de la fréquence d'horloge (ici tous les MHz).

fonction $\sin(x)/x$. Plus important, ce spectre, comme celui d'une seule impulsion à 1 Mbps, est nul à chaque multiple de 1 MHz.

Pour en revenir à notre problème de limitation de la bande passante, ce spectre montre pourquoi un filtre est nécessaire :

le spectre en fréquence d'un signal NRZ non filtré contient des composantes jusqu'à des fréquences théoriquement infinies ! Bien sûr, dans la vie réelle, cet « infini » sera limité, mais l'occupation de la bande passante sera toujours beaucoup plus grande que nécessaire.

Filtres passe-bas ?

Vous avez donc compris qu'un filtre sera utile entre la source de signal binaire et le transmetteur (ou le modulateur pour un système radio). Mais quelles doivent être ses caractéristiques ? Tout d'abord, il doit atténuer le plus possible les hautes fréquences. Mais il doit aussi permettre au récepteur de décoder le signal sans erreurs, ou plus exactement sans plus d'erreurs que tolérable.

Pour limiter les hautes fréquences, vous aurez évidemment besoin d'un filtre passe-bas. Pourquoi ne pas utiliser un filtre classique, par ex. un filtre de Butterworth ? J'ai réalisé une simulation pour vous, avec un Butterworth d'ordre 2 et de fréquence de coupure variable (**figure 4**). Ces graphes méritent une petite explication. La colonne de gauche montre le spectre du signal sans filtrage (en haut), avec filtre passe-bas de fréquence de coupure à 3 dB à 310 kHz (milieu) ou 410 kHz (en bas). La réponse du filtre apparaît superposée en rouge. Les graphes de la colonne du milieu montrent quelques bits du signal filtré, c'est-à-dire dans le domaine temporel, comme si vous utilisiez un oscilloscope. Enfin, la colonne la plus à droite est ce qu'on appelle le **diagramme de l'œil** du signal reçu. Cela peut paraître impres-

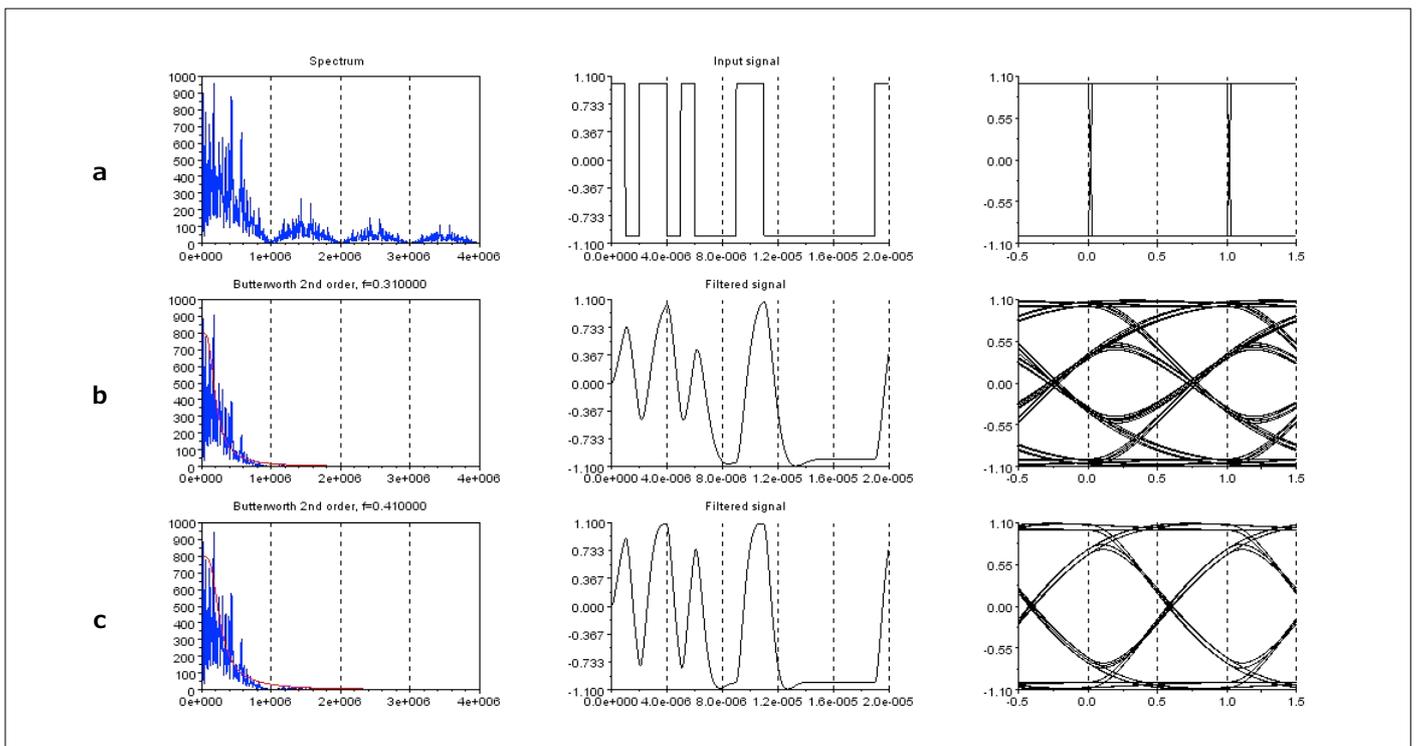


Figure 4. a) Le spectre d'un signal aléatoire sans filtrage, et son diagramme de l'œil à droite. b) Avec un filtre Butterworth à 310 kHz, le signal est très déformé et le diagramme de l'œil est quasiment fermé. c) Avec une fréquence de coupure plus élevée (410 kHz), l'ISI est tolérable, et le spectre a des composantes visibles jusque vers 2 MHz.

sionnant, mais c'est une notion très simple : imaginez que vous avez un oscilloscope. Déclenchez-le sur n'importe quel front de montée ou de descente du signal d'entrée, mettez à l'échelle « la chose » pour afficher deux bits sur la largeur de l'écran, l'un des deux étant au centre de l'écran. Enfin accumulez beaucoup de courbes sur l'écran, c'est-à-dire nombre de bits aléatoires. On obtient le diagramme de l'œil. Ce diagramme fournit en un clin d'œil (c'est le cas de le dire...) une représentation de la difficulté que le récepteur aura à « récupérer les bits » sans erreur. En effet, plus les courbes représentant les '1' restent éloignées des courbes représentant les '0', plus cette réception sera facile, par ex. avec un comparateur de tension. On dit alors que l'œil est « ouvert ». Si les trajectoires des bits successifs ne se superposent pas, cela signifie qu'il y a une sorte d'effet mémoire : la forme d'onde pour un bit donné varie en fonction des bits précédents. Ce phénomène est appelé **interférence inter-symboles** (*Inter-Symbol Interference*, ou ISI, en anglais) et rend la réception plus compliquée. Évidemment cet effet mémoire est accentué par le filtrage. Regardez à nouveau ces simulations de filtre Butterworth. La première ligne

(**figure 4a**), sans filtrage, a un œil complètement ouvert. La deuxième ligne (**figure 4b**), avec une fréquence de coupure à 3 dB de 310 kHz, montre un spectre en fréquence très réduit à partir de 1 MHz, mais au prix d'une ISI très élevée : le diagramme de l'œil est presque fermé, ce qui veut dire que des erreurs de réception apparaîtront dès qu'il y aura un peu de bruit ou de parasites. La troisième ligne (**figure 4c**), avec un filtre à 410 kHz, semble un juste compromis. Son niveau d'ISI est raisonnable, même s'il est encore très visible (les trajectoires successives ne passent pas par le même point), et le spectre est très réduit à partir de 2 MHz.

Et les filtres gaussiens ?

L'utilisation d'un filtre passe-bas même rudimentaire apporte donc une amélioration notable quant à l'occupation de la bande passante. L'exemple précédent montre qu'on peut transmettre les mêmes 1 Mbps avec une bande passante drastiquement réduite, de plusieurs dizaines de MHz sans filtrage à 2 MHz environ. Mais existe-t-il une meilleure solution ?

La réponse est évidemment oui. La plus classique d'entre elles est d'utiliser un filtre gaussien. Qu'est-ce que c'est encore

que cette chose ? Simplement un filtre qui a une réponse en fréquence en forme de demi-gaussienne (ou demi-cloche) : il n'a pas de fréquence de coupure précise, mais il abaisse progressivement les hautes fréquences, avec une forme très douce, et s'approche très vite de zéro. Enfin la gaussienne a une propriété remarquable : sa transformée de Fourier est... aussi une gaussienne. Cela implique que sa réponse dans le domaine temporel a les mêmes qualités.

Allez, une petite simulation vous permettra de comprendre son intérêt (**figure 5**). Si l'on prend un filtre gaussien très large (**figure 5a**), sa réponse impulsionnelle est une gaussienne très étroite et il n'y a presque pas d'interférences inter-symboles, le diagramme de l'œil est complètement ouvert. Inversement, un filtre étroit dans le domaine fréquentiel a une longue réponse impulsionnelle et donc un niveau élevé d'ISI (**figure 5c**). La valeur intermédiaire choisie pour la figure 5b est proche de l'optimum, avec une ISI presque nulle et une occupation raisonnable de la bande passante d'environ 2 MHz. Si vous comparez cette simulation à celle du filtre de Butterworth, vous verrez que la bande passante globale utilisée est similaire, 2 MHz dans les deux cas, mais le filtre gaussien four-

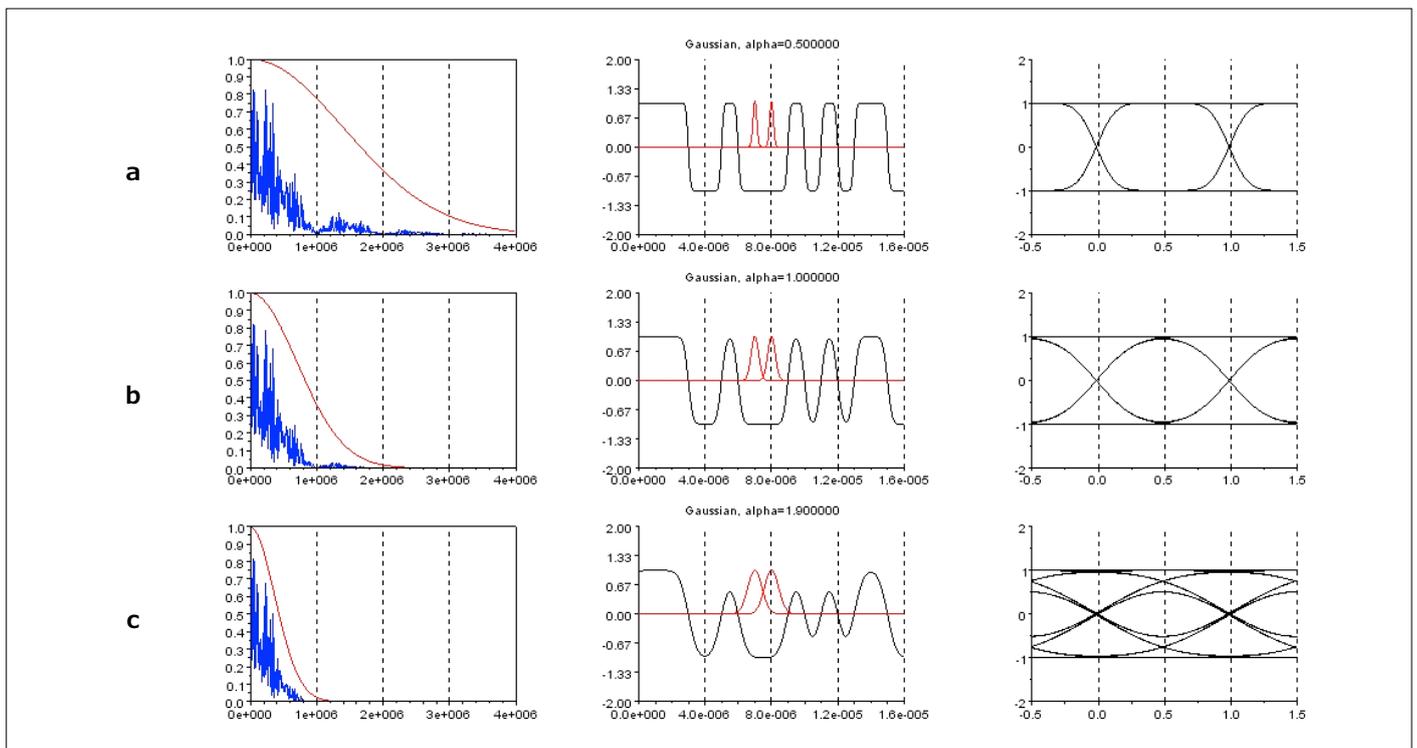


Figure 5. a) Un exemple de filtre gaussien assez large, avec un signal parfait. b) Ici le filtre gaussien est nettement plus resserré, mais l'ISI reste nulle. c) Avec un filtre trop étroit, de l'ISI réapparaît.

nit un signal reçu beaucoup plus propre. C'est pourquoi ces filtres gaussiens sont très couramment utilisés. Par exemple le 'G' dans des acronymes comme GMSK ou GFSK signifie « gaussien », car ces techniques de modulation utilisent un filtre gaussien. Vous les avez tous utilisées, car elles sont au cœur du GSM comme du Bluetooth.

Nyquist à la rescousse

Un filtre gaussien est déjà excellent, mais bien sûr les mathématiciens ont étudié comment obtenir la meilleure efficacité pour l'occupation de la bande passante. En 1928, Harry Nyquist a publié un célèbre article intitulé « Certains facteurs influençant le débit des transmissions télégraphiques ». Dans ce texte, il a prouvé un résultat fondamental (qui sera d'ailleurs étendu par Claude Shannon plus tard) : la largeur de bande minimale pour transmettre R symboles par seconde sans interférence inter-symboles est égale à $R/2$ hertz. D'ailleurs, vous connaissez sûrement cette loi de Nyquist sous sa forme inverse : un signal de bande passante B hertz doit être échantillonné avec au moins $2B$ échantillons par seconde.

Ainsi, dans notre exemple, comme nous avons 1 Mbps à transmettre, nous devrions pouvoir trouver une solution en utilisant seulement 500 kHz de

bande passante, et sans ISI. Ce serait une énorme amélioration par rapport aux 2 MHz que nous avons obtenus laborieusement avec les filtres précédents. Heureusement, Nyquist a également expliqué comment faire : théoriquement, le filtre optimal est un filtre passe-bas infiniment raide, coupant exactement à $R/2$, c'est-à-dire ici à 500 kHz !

Un peu d'explication pour ceux que la théorie ne rebute pas (les autres, sautez ce paragraphe). Vous vous souvenez de la fonction $\sin(x)/x$? Cette fonction passe par zéro périodiquement (tous les T), mais à un lobe principal deux fois plus large ($2T$). Que se passe-t-il si on prend deux fonctions $\sin(x)/x$, qu'on les décale d'exactly T secondes et qu'on en fait la somme ? Lorsque la première atteint son maximum, l'autre est nulle, et vice versa. Par conséquent, il n'y aura pas d'ISI à cet instant précis. C'est l'idée, mais comment la mettre en œuvre ? Rappelez-vous que la fonction $\sin(x)/x$ est la transformée de Fourier d'une impulsion unitaire. Réciproquement, cela implique que la réponse impulsionnelle d'un filtre parfaitement raide est aussi un $\sin(x)/x$, et une telle réponse impulsionnelle permet d'obtenir ce que l'on cherche. Il suffit de l'ajuster pour que $T = 1/2$ Mbps, c'est-à-dire en fait une fréquence de coupure de 500 kHz. CQFD. Une fois de plus une

petite simulation Scilab vous aidera à comprendre le pourquoi de ce résultat très étonnant (**figure 6**). Ici, par simplicité les bits d'entrée ne sont pas codés en NRZ, mais sous forme d'impulsions positives ou négatives. Lorsqu'une telle impulsion passe à travers un filtre, la sortie est par définition la réponse impulsionnelle du filtre. J'ai pris l'exemple de trois bits successifs, respectivement '1', '1' et '0' (figure 6, colonne 1). Les réponses impulsionnelles qui en résultent sont indiquées dans la deuxième colonne. Vous voyez qu'il s'agit de $\sin(x)/x$, et que les formes d'onde sont exactement nulles aux points de décision, sauf une. De là, la somme de ces trois réponses impulsionnelles fournit un signal exactement égal au signal d'entrée aux instants d'échantillonnage (figure 6, à droite). Pour ceux qui sont intéressés par des protocoles radio un peu plus avancés, tout ceci est étroitement lié aux protocoles OFDM, mais cela nécessiterait un autre article... Revenons à nos moutons et à un signal binaire aléatoire de 1 Mbps codé en NRZ et passant au travers d'un filtre infiniment raide. J'ai effectué la simulation pour vous (**figure 7**). Les différentes lignes vous montrent le résultat de la simulation avec une fréquence de coupure légèrement variable : 460 kHz en haut, précisément 500 kHz au milieu et 540 kHz en bas. Si le filtre n'est pas exac-

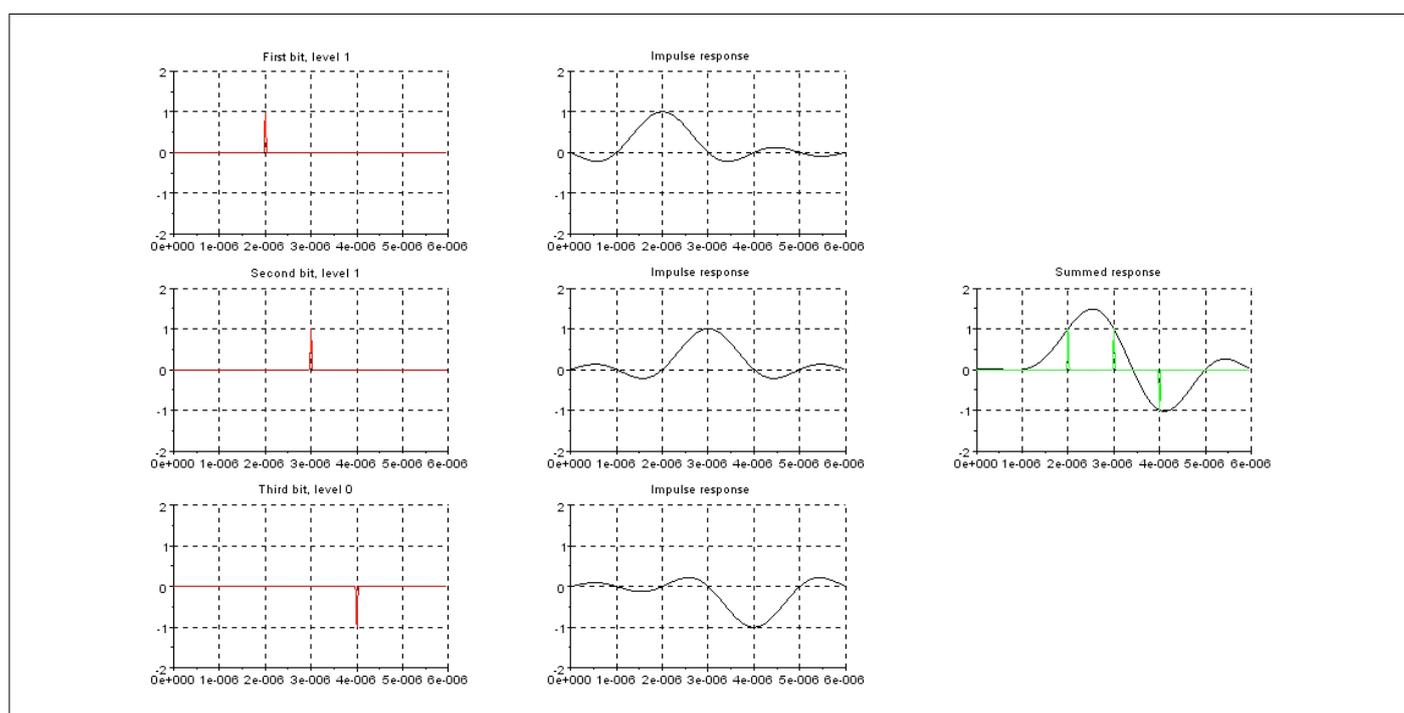


Figure 6. Un filtre de Nyquist a des performances optimales. Ici, trois bits sont codés sous forme d'impulsions positives ou négatives (première colonne). Le filtre crée trois fonctions $\sin(x)/x$ décalées dans le temps (colonne centrale). La somme de ces trois signaux n'a aucune ISI (à droite).

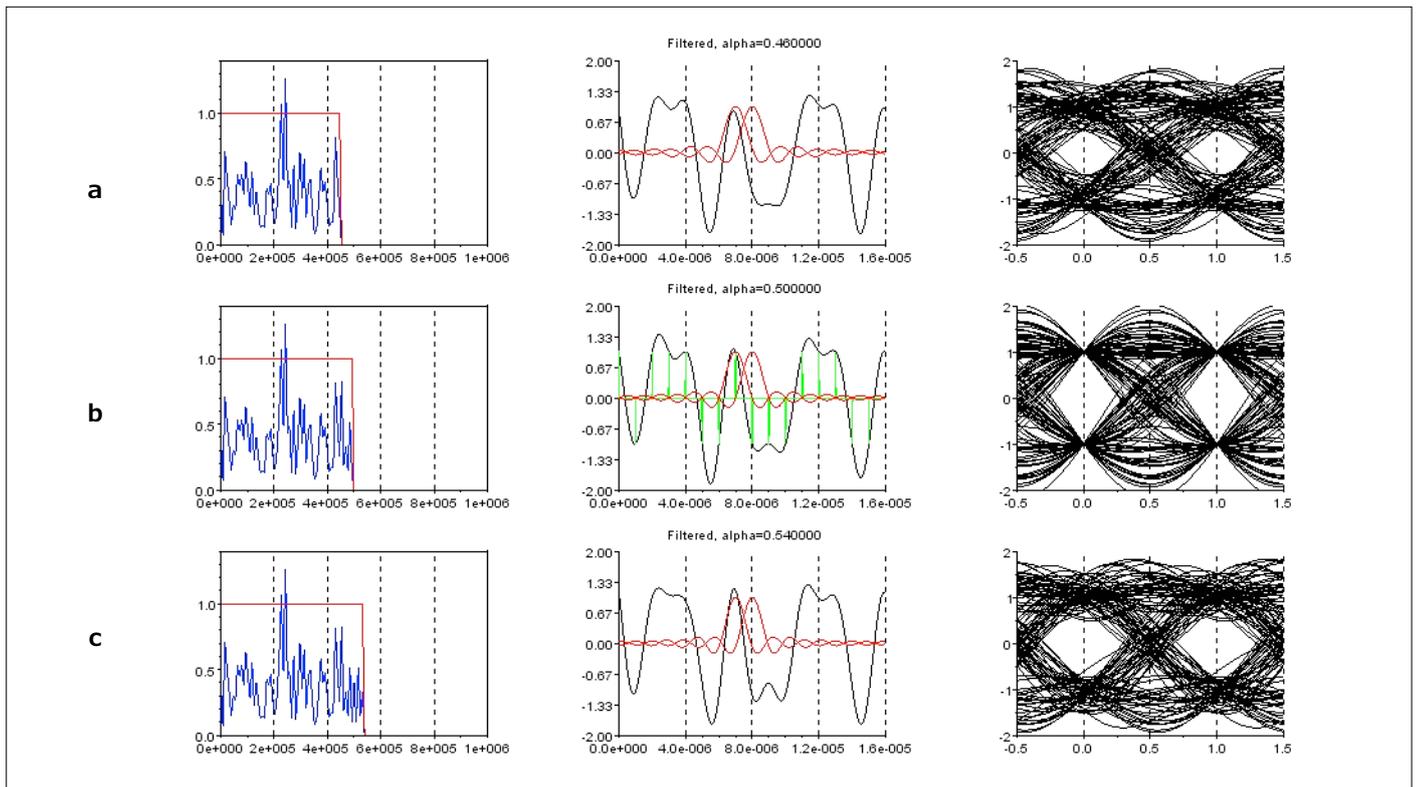


Figure 7. Un filtre de Nyquist n'a pas d'ISI si sa fréquence de coupure est précisément de 500 kHz pour 1 Mbps (b, au milieu). Des fréquences de coupure légèrement inférieures (en haut) ou supérieures (en bas) dégradent considérablement la réponse.

tement réglé, une très forte ISI est présente (**figures 7a** et **7c**). Par contre, s'il est précisément réglé à la moitié du débit binaire, ici 1 Mbps/2 = 500 kHz, l'ISI disparaît comme par magie (**figure 7b**). C'est clairement visible sur le diagramme de l'œil : toutes les trajectoires semblent aléatoires, mais passent exactement par le même point à l'instant précis où il faut décider si c'est un '1' ou un '0'. Dans la colonne du milieu, j'ai ajouté (en rouge) la réponse impulsionnelle simulée, pour deux bits successifs, ainsi qu'une copie du signal d'entrée (en vert). Comme prévu, le signal filtré est exactement

égal au signal d'entrée aux instants de décision.

Filtres à cosinus surélevé

Magique non ? Un tel filtre, dit de Nyquist, fournit une efficacité optimale en ce qui concerne la bande passante et cela sans ISI. Super, mais malheureusement personne ne peut concevoir un filtre parfaitement raide. Cette solution est donc l'optimum théorique, mais n'est souvent pas réalisable. Telle la cavalerie, les mathématiciens sont revenus pour nous aider : « Vous ne voulez pas de notre filtre parfaitement

raide ? Pourquoi pas un filtre à peu près similaire, c'est-à-dire toujours avec une fréquence de coupure de $R/2$, mais avec une pente un peu moins droite et permettant toujours zéro ISI ? ». Cette solution s'appelle le filtre à cosinus surélevé (*raised cosine filter* en anglais). Vous trouverez sa définition exacte dans la littérature, et en particulier dans le magnifique ouvrage écrit par Bernard Sklar [4]. En résumé, les filtres à cosinus surélevés sont une famille de filtres paramétrés par une valeur notée w . Si w est proche de 0,5, le filtre est proche d'un filtre de Nyquist.

Liens

- [1] « modulations numériques : codage des données (NRZ, Manchester &) », Elektor 09-10/2018 : www.elektormagazine.fr/180370-01
- [2] Logiciel de simulation Scilab : www.scilab.org
- [3] Page de l'article (scripts Scilab à télécharger) : www.elektormagazine.fr/180509-01
- [4] « Digital communications – Fundamentals and Applications », 2nd edition, Bernard Sklar, Prentice Hall, ISBN 0-13-084788-7
- [5] « Pulse shaping » (mise en forme d'impulsions), Wikipedia : http://en.wikipedia.org/wiki/Pulse_shaping
- [6] « Inter Symbol Interference(ISI) and Root-raised Cosine (RRC) filtering », Charan Langton : <http://complextoreal.com/wp-content/uploads/2013/01/isi.pdf>
- [7] Puce d'émetteur-récepteur ISM ADF7021 à bande étroite et à haute performance, Analog Devices Inc. : www.analog.com/en/rfif-components/rfif-transceivers/adf7021/products/product.html

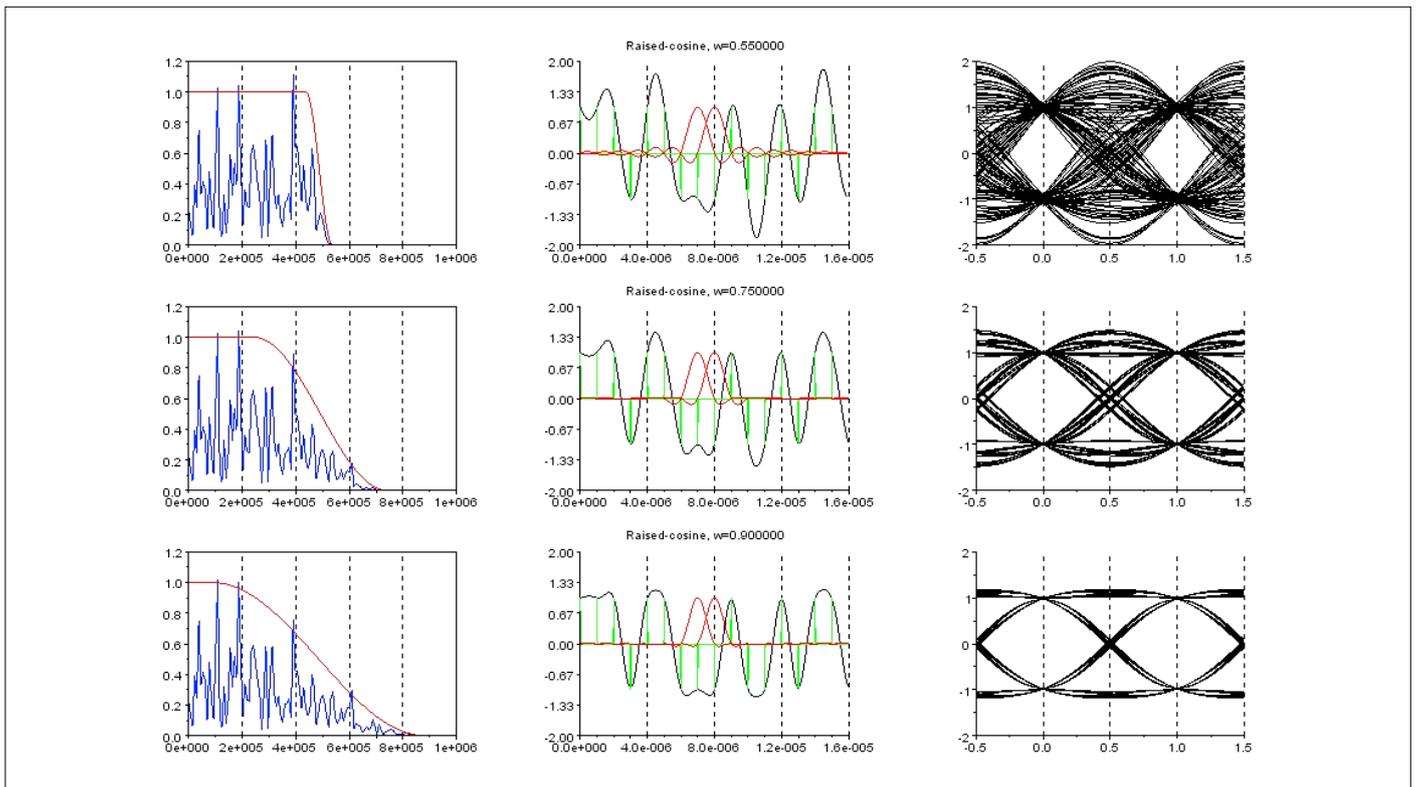


Figure 8. Un filtre à cosinus surélevé est proche d'un filtre de Nyquist, mais avec une caractéristique de coupe moins raide, ici avec une réponse nulle à partir de 550 kHz (en haut), 750 kHz (au milieu) et 900 kHz (en bas). Dans tous les cas, l'ISI est nulle aux temps d'échantillonnage.

S'il est augmenté, le profil du filtre est un peu relaxé, et son spectre est plus large. Les filtres à cosinus surélevés ont deux avantages. Tout d'abord, le spectre du signal filtré est borné : si vous sélectionnez $w = 0,8$, le signal filtré n'aura théoriquement aucune énergie supérieure à 0,8 fois la fréquence de Nyquist. C'est un gros avantage par rapport aux filtres passe-bas classiques qui laissent toujours passer un peu de signal dans les hautes fréquences. Deuxièmement, l'ISI est toujours strictement nulle, quelle que soit la valeur sélectionnée pour w . J'ai procédé à une simulation pour vous, avec différentes valeurs de w (**figure 8**). Une fois de plus, les fichiers sources Scilab pour toutes ces simulations sont sur le

site web d'Elektor alors n'hésitez pas à jouer avec eux.

Pour conclure

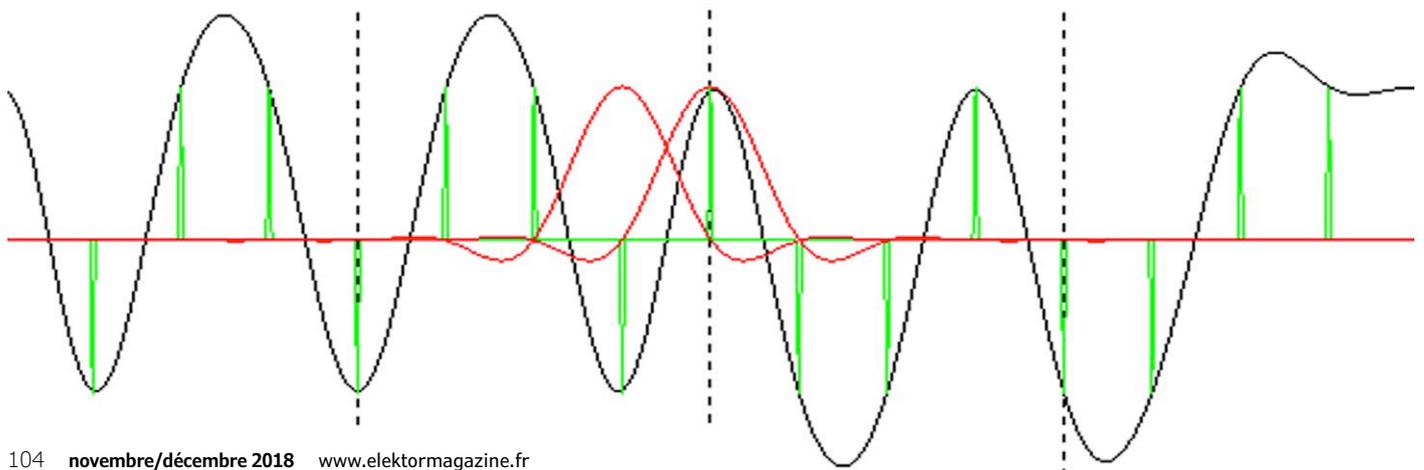
Un article aussi court ne peut bien sûr être qu'une introduction au vaste sujet du filtrage de canal. J'espère que vous en avez saisi l'idée : transmettre un signal numérique brut sur n'importe quel support est un gaspillage de bande passante. Un filtre peut réduire significativement la bande occupée sans augmenter fortement le taux d'erreur. Ce filtre doit par contre être bien conçu afin de minimiser l'interférence entre symboles (ISI). La solution idéale, à savoir le filtre de Nyquist, permet de limiter le spectre à la moitié du débit binaire transmis (500 kHz pour 1 Mbps), mais ce filtre n'est qu'un

rêve de mathématiciens. Les filtres à cosinus surélevés et les filtres gaussiens sont deux classes de filtres réalisables qui fournissent un rapport complexité/performance plus qu'intéressant.

Encore une fois, tout ce que je vous ai présenté ici s'applique quel que soit le canal de communication, qu'il s'agisse d'une paire de fils, d'une liaison sans fil en modulation d'amplitude, de fréquence et/ou de phase, de modems à courant porteur ou même de liaisons optiques. Dans le prochain article, je basculerai un peu plus dans le domaine de l'électronique radio en vous parlant de mélangeurs. ◀

(180509-01)

Cet article a été publié dans la revue *Circuit Cellar* (n°285, avril 2014).



lampemètre AVO

toujours en forme

Martin Beusekamp

La Collection d'études historiques de la Faculté de génie électrique, mathématique et informatique (EWI) de l'université de Twente (Pays-Bas) [1] rassemble plus de 1 100 appareils de l'histoire des domaines de spécialisation de la faculté : instruments de mesure électroniques, ordinateurs, calculateurs, appareillage de télécommunication, etc. Ils sont gérés par des volontaires, principalement des retraités de la faculté. Cet article de rétronique ravive le souvenir des lampemètres de la firme AVO, dont certains font partie de la collection en question.

Un lampemètre, ce n'est rien de plus ni rien de moins qu'un certain nombre d'alimentations, un assemblage considérable de commutateurs et de potentiomètres, un tableau garni de tout un échantillonnage de supports de tubes (à l'époque, on parlait de lampes de radio), et un appareil de mesure, le tout rassemblé dans un seul boîtier (cf. photo ci-dessus).

En général, un lampemètre doit pouvoir fournir au tube à tester quatre tensions différentes (cf. encadré « **Rien que pour les tubes chauffés en tension ?** ») : la tension de filament, celle d'anode, celle de grille-écran et celle (négative) de grille de commande. Elles ne sont pas toujours nécessaires, une diode n'a pas de grille, une triode, pas de grille-écran, un tube à cathode froide n'a pas de filament.

Dans le lampemètre AVO, les quatre tensions proviennent de trois alimentations distinctes, chacune des trois avec un transformateur secteur individuel. Ces alimentations méritent une attention particulière, elles sont essentielles dans un lampemètre.

Faut-il une tension de filament ?

La tension de filament d'un lampemètre AVO provient d'une source alternative capable de délivrer beaucoup de courant. Un tube AZ4, par exemple, une double diode fréquente dans l'alimentation des radios à tubes du milieu du siècle dernier, demande 2,3 A de courant de chauffage sous une tension de 4 V. Certains tubes industriels étaient même plus gourmands. La cathode d'un tube électronique est réalisée dans un matériau qui, chauffé, libère facilement beaucoup d'électrons, c'est



Un lampemètre AVO dans toute sa splendeur. Remarquez le manuel d'accompagnement dans l'étagère sous l'appareil. Photo : Gerrit Busscher.

l'émission thermique, dit effet Edison. On distingue le chauffage direct, où le courant circule dans la cathode elle-même, et le chauffage indirect, produit par un filament séparé à proximité immédiate de la cathode.

Pour faire passer un courant de chauffage et produire l'incandescence, il faut une tension (cf. encadré « **Rien que pour les tubes chauffés en tension ?** »). Et il y a presque autant de tensions différentes que de tubes existants. Le lampemètre AVO doit donc disposer de pas moins de 121 tensions de filament différentes. Un commutateur à onze positions procure une première sélection pour des tensions entre 0 et 110 V \sim . Un autre commutateur à onze positions assure la sélection fine en augmentant la tension de filament de 0 à 7,5 V \sim .

La plus basse tension de filament que le lampemètre AVO peut délivrer est 0,625 V \sim , la plus haute est 117,5 V \sim , à peu près la tension du secteur aux États-Unis dans la première décennie après la Seconde Guerre. Il y avait en effet beaucoup de tubes dont le filament était alimenté directement par le secteur, par exemple les types 117Z3 et 117Z6.

Deux bonnes tensions

Avec un nuage d'électrons libres autour de la cathode, nous ne sommes pas fort avancés. Sauf si nous attirons ces électrons vers l'anode en lui appliquant une assez haute tension continue. Les électrons chargés négativement se déplacent de la cathode vers l'anode, le courant continu va dans l'autre sens, le sens conventionnel. Impossible de renverser la ten-

dance, comme il n'y a pas de nuage d'électrons libres autour de l'anode, ils ne peuvent pas aller dans le sens inverse. Le tube électronique est bien une diode !

Sur le lampemètre AVO il y a un commutateur à dix-sept positions pour la tension d'anode entre 12,6 V et 400 V. Pas de réglage fin, il n'est pas indispensable. Si vous voulez connaître les caractéristiques d'un tube à la tension d'anode de 275 V, une tension qui n'est pas prévue, mesurez-les à 250 V et à 300 V et prenez la moyenne des deux.

La tension d'anode relativement haute qui attire les électrons libres de la cathode leur confère une grande vitesse. Certains d'entre eux rebondissent même et peuvent perturber le fonctionnement du tube. C'est pourquoi on a fabriqué des tubes avec plus d'une grille (tétrodes, pentodes, etc.). On ajoute ainsi une grille-écran entre la cathode et l'anode sur laquelle on applique un potentiel élevé. La grille-écran capture bien quelques électrons au passage, mais la plupart ont une énergie suffisante pour traverser les mailles de l'écran. Presque tous les électrons qui ont rebondi sont récupérés par la grille-écran. L'alimentation de l'anode sert aussi pour la grille-écran, sa tension se règle en dix-sept pas de 12,6 V à 300 V. Pas de réglage fin non plus.

Une tension, mais pas de courant

La grille la plus importante dans un tube électronique est la grille de commande, la première à partir de la cathode. La grille de commande (notée grid 1 ou g1) est mise à un potentiel négatif par rapport à la cathode et freine le flux d'électrons. Une tension fortement négative peut d'ailleurs l'arrêter totalement. C'est précisément parce que la grille de commande est négative qu'elle repousse les électrons, il n'y a donc pas de courant de grille, exception faite du minuscule courant de charge et décharge de la capacité de la grille lors des changements de tension. Drôle d'idée, une alimentation qui doit fournir une tension réglable, mais pas de courant. Pourtant, le lampemètre AVO peut mesurer un courant de grille jusqu'à 100 μ A. C'est qu'un courant de grille peut apparaître si le vide ne règne plus dans le tube et il faut pouvoir détecter cette anomalie, le courant mesuré indique que de l'air s'y est introduit.

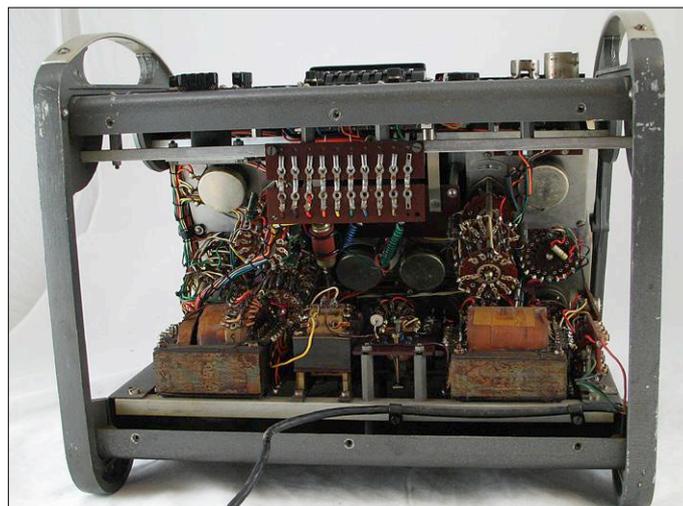


Figure 1. Les entrailles d'un lampemètre AVO. Photo : Gerrit Busscher.

La tension de grille de commande provient de la troisième alimentation. La plage va de 0 à -100 V et, en plus d'un commutateur pas à pas, il y a bien un réglage fin par potentiomètre pour offrir n'importe quelle tension souhaitée.

Où sont les condensateurs électrolytiques ?

La **figure 1** dévoile l'intérieur du lampemètre AVO. On ne peut pas rater les trois transformateurs des alimentations distinctes évoquées, ainsi que les nombreux commutateurs pour les réglages pas-à-pas et les potentiomètres de réglage fin. Mais où se cachent donc les condensateurs électrolytiques pour lisser les différentes tensions continues ?

La réponse est simple : il n'y en a pas. Pour réduire les coûts, les tensions d'anode, de grille-écran et de grille de commande dans un lampemètre AVO classique sont des tensions alternatives à 50 Hz, redressées à simple phase, mais ni aplanies, ni stabilisées ! Dans les modèles Mark III et antérieurs, la tension d'anode n'était même pas redressée, on y envoyait comme tension d'anode l'onde pleine, les deux alternances !

Rien que pour les tubes chauffés en tension ?

Les lampemètres AVO ne peuvent fournir qu'une **tension** de filament au tube à tester. Ils peuvent tester tous les tubes européens, notamment ceux dont le numéro de type commence par A (4 V), D (1,4 V) ou E (6,3 V), pour n'en citer que quelques-uns.

Mais il existe aussi des tubes destinés à recevoir un **courant** de chauffage, tels que ceux dont le type commence par P (300 mA) ou U (100 mA). Tout compte fait, dans la plupart des téléviseurs, les filaments des tubes étaient alimentés en série et reliés à la tension du secteur directement ou via une résistance ou un condensateur. Ceux-là, on ne peut pas les tester ? Bien sûr que si ! Dans tous les tubes, le filament, une fois à température, n'est jamais qu'une résistance ordinaire. Une tension de chauffage fait circuler un courant et

à l'inverse, un courant de chauffage provoque une chute de tension sur le filament. C'est pourquoi on trouvait aussi dans ces téléviseurs des tubes comme les ECC82 (12AU7) et EF80 (6BX6), prévus pour une tension de 6,3 V, mais en même temps pour un courant de 300 mA, comme ceux de la série P.

Prenez par exemple la célèbre penthode de puissance PL500 avec son courant de chauffage de 300 mA. Les guides de tubes (et à présent internet) renseignent que le courant de chauffage produit une chute de tension de 27 V aux bornes du filament. Sur le lampemètre, nous choisissons donc la tension de 27 V. Nous profitons ainsi de ce que le testeur ne peut vérifier qu'un seul tube à la fois, sans avoir à nous demander si le filament est alimenté en série ou en parallèle.

Ces lampemètres AVO sont basés sur le fait que l'alternance négative ne modifie en rien le fonctionnement du tube qui travaille en redresseur. Dans le schéma très simplifié de la **figure 2**, on voit que pour le modèle Mark III, la tension d'anode AV est prélevée directement du transformateur, elle est donc alternative à double alternance. Le sens des diodes indique que la grille-écran reçoit l'alternance positive SV et la grille de commande l'alternance négative V_g . Toutes les tensions appliquées au tube varient donc constamment, mais leur moyenne reflète la valeur réglée. L'inertie de l'instrument fait en sorte que là aussi, c'est la valeur moyenne de la grandeur mesurée qui s'affiche.

Seulement voilà, comme pour une mesure sur un tube, ce n'est pas la valeur moyenne qui compte, mais bien la valeur **efficace** de la grandeur à mesurer, par exemple le courant anodique, on en a tenu compte dans l'étalonnage de l'instrument de mesure. Par rapport à la valeur de pointe V_p , la valeur moyenne d'une alternance de sinus est toujours :

$$2 / \pi (V_p) = 0,636 V_p$$

et la valeur efficace est :

$$1/2 \sqrt{2} (V_p) = 0,707 V_p.$$

Il est alors facile pour le rapport $0,707 / 0,636 = 1,110$ de compenser la mesure.

Cette méthode de mesure avec des sinusoïdes au lieu de tensions continues a permis à AVO d'en faire ses choux gras. Outre une économie de composants dans ses appareils, en faisant breveter l'idée en Grande-Bretagne (n° 606707 [2]), AVO a encore gagné de l'argent.

Pente

Faire varier le courant anodique par la tension sur la grille de commande (cf. encadré « **Une tension, pas de courant** ») décrit exactement le fonctionnement du tube en amplificateur. La caractéristique la plus importante d'un tube est donc sa pente

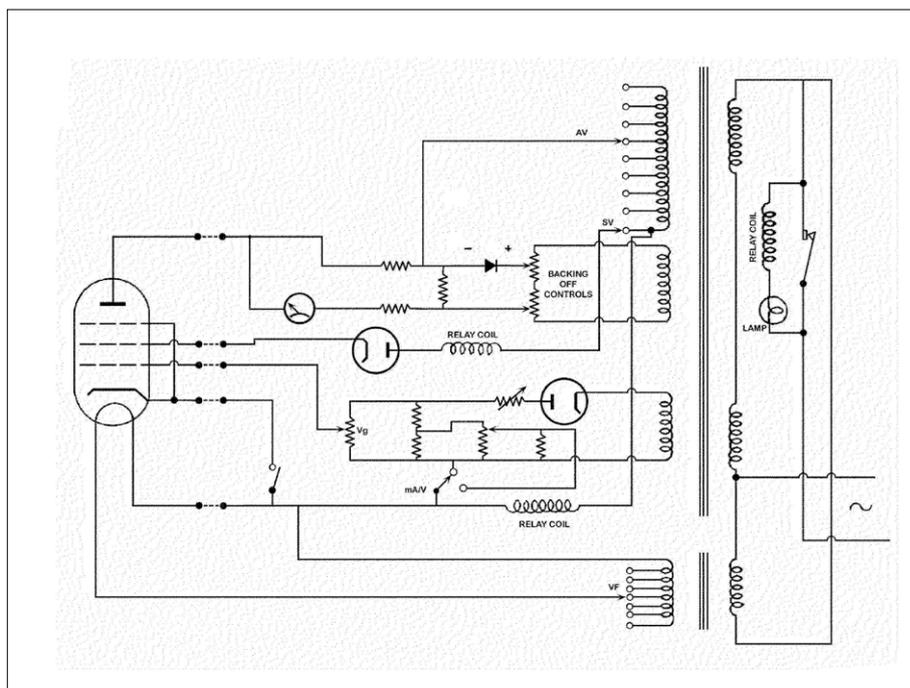


Figure 2. Schéma très simplifié d'un lampemètre AVO.

ou transconductance, le changement de courant d'anode en milliampères (mA) pour un changement de tension de grille en volts (V). La pente s'exprime donc en mA/V et son nom dérive de l'inclinaison, la pente de la courbe du courant d'anode (axe vertical) en fonction de la tension de grille (axe horizontal). Tous les lampemètres AVO peuvent bien sûr mesurer la pente d'un tube.

De plus, la précision de la mesure entre en jeu. Disons qu'un tube à un certain point de réglage conduit un courant d'anode de 50 mA et que nous voulons mesurer sa pente en petits signaux, donc pour une petite variation de la tension de grille. Si la variation de courant anodique est par exemple de 1 mA, nous n'aurons qu'une variation de 2 % du courant, donc peu de précision. C'est pour cela que le lampemètre AVO nous offre la possibilité, autour du point de réglage en continu, de demander un courant de « baking-off » (cf. schéma, figure 2) qui compense le courant d'anode au niveau du point de réglage. Ce faisant, on peut commuter l'ampèremètre de l'appareil sur une gamme de plus haute sensibilité et mesurer la pente du tube avec une plus grande précision.

Liens

- [1] Collection d'études historiques de la Faculté de génie électrique, mathématique et informatique (EWI) de l'université de Twente (Pays-Bas) : <http://studieverzameling.utwente.nl/>
- [2] Brevet n°606707 déposé au Royaume-Uni par AVO : <https://frank.pocnet.net/instruments/AVO/HR/ValveTesters/patent/AVOpatent.pdf>
- [3] Guide de maintenance du lampemètre Mk4 d'AVO : https://frank.pocnet.net/instruments/AVO/HR/ValveTesters/manuals/VCM_Mk-IVservice.pdf
- [4] Guide de l'utilisateur du lampemètre Mk4 d'AVO : https://frank.pocnet.net/instruments/AVO/HR/ValveTesters/manuals/VCM_Mk-IVuser.pdf



Figure 3. Des supports adéquats à tous les tubes en usage. Photo : Gerrit Busscher.



Figure 4. Le carrousel de commutateurs à molette. Chaque tube a son code unique. Photo : Gerrit Busscher.

Supports de tube à gogo

Sur le dessus du lampemètre AVO, on voit un grand nombre de supports de tube différents, sur le modèle Mark IV par exemple, il y en a dix-neuf (**figure 3**). Sur chacun d'eux, tous les contacts numéro 1 sont interconnectés, idem pour les contacts numéro 2, etc. En outre, on dispose aussi d'un fil muni d'un genre de pince crocodile pour les tubes dont l'ampoule est surmontée d'un contact.

Nous devons encore répartir les différentes tensions sur les broches adéquates du tube à tester. Cela se fait avec le carrousel de commutateurs à molette au milieu des supports de tube (**figure 4**). Voyez l'encadré « **Le manuel toujours nécessaire** » pour savoir comment s'y prendre au juste.

Oscillations parasites

Chaque lampemètre AVO contient plusieurs dizaines de mètres de fil de câblage pour relier tous les organes cités, alimenta-

tions, commutateurs et supports de tube de manière convenable. Inutile de dire que tous ces fils ont leur inductance et leur capacité parasites propres par rapport aux autres fils. La tendance à l'oscillation du tube sous investigation est bien présente.

Les lampemètres AVO ne mesurent pourtant pas de caractéristiques en haute fréquence. Donc, on peut équiper tout le câblage de perles de ferrite pour affaiblir les oscillations parasites. Le câblage entre les différents supports de tube et autour du carrousel de commutateurs à molette notamment est effectivement doté de perles de ferrite. Notez bien que le manuel d'entretien [3] avertit littéralement qu'après toute réparation, il faut remettre tout le câblage dans sa position d'origine.

Très solide

Autant les multimètres AVO sont solides, autant les lampemètres AVO le sont aussi. La Collection d'études historiques

Le manuel toujours nécessaire !

La plupart des lecteurs de ce magazine sont certainement capables de manipuler un multimètre ou une alimentation sans avoir étudié au préalable le mode d'emploi (pas toujours très clair, d'ailleurs). Et avec un générateur de fonctions ou un oscilloscope, ce sera souvent pareil, sans aide écrite. Qu'en est-il d'un lampemètre AVO ? N'y pensez pas ! Pour pouvoir tester le tube, il faut absolument consulter le manuel. Il faut toujours s'occuper d'envoyer toutes les tensions (filament, tension d'anode, de grille-écran, négative pour la grille de commande) sur la bonne broche du support de tube. Et les numéros des broches sont aussi différents d'un tube à l'autre.

C'est ce que résout le « carrousel » de commutateurs à molette au-dessus du lampemètre AVO (figure 4). Avant de mettre sous tension le testeur, la série d'interrupteurs doit être placée sur le bon code numérique, 141 230 651 pour un EF80, par exemple. Ce code se trouve dans le manuel, que vous devez donc avoir toujours à portée de main. Vous

n'appliqueriez jamais une tension d'anode de 250 V, par exemple, sur le filament. C'est pourquoi, sur les premières séries de lampemètres AVO, le manuel était attaché à l'appareil par une chaînette. D'ailleurs, sur la photo du chapeau de l'article, on remarque bien le manuel glissé sur un support sous l'appareil.

De plus, les lampemètres AVO sont protégés en interne contre les dommages dus à des courants trop forts en provenance de l'alimentation pour les tensions d'anode, de grille-écran et de grille de commande. Au-dessus d'une certaine valeur critique de courant, un relais s'enclenche et débranche le testeur de tube, voyez la bobine du relais dans la figure 2, le courant d'anode est limité par des résistances. Du reste, des erreurs de manipulation peuvent sonner le glas du tube en cours de test, tout comme vous pourriez, avec un traceur de courbes, démolir une diode ou un transistor si vous le soumettez à des tensions, des courants ou des puissances qui ne lui font pas de du bien.

EST^D 2004

www.elektor.tv



Rétronique est une rubrique mensuelle sur les pages glorieuses et jaunies de l'électronique, avec occasionnellement des montages de légende décrits dans Elektor.

Si vous avez des suggestions de sujets à traiter, merci de les télégraphier à redaction@elektor.fr

EWI en possède trois, dont un portable (**figure 5**), lequel a malheureusement l'instrument de mesure cassé. La révision des deux modèles de table, un Mark III et un Mark IV, a été simple. Nettoyer, lubrifier les commutateurs et potentiomètres, remplacer le cordon secteur et remettre en marche !

Il n'a même pas été nécessaire d'interposer un transformateur variable pour la tension secteur de manière à faire monter lentement et progressivement, des heures durant, la tension sur les condensateurs électrolytiques pour les reformer après une longue période d'inactivité. Et pour cause, les appareils n'en ont jamais eu !

À tous les utilisateurs

Au temps des « lampes de radio », on avait l'habitude de soumettre les appareils, notamment professionnels, à une inspection périodique au cours de laquelle on remplaçait les tubes chancelants. Pour les révisions rapides, le panneau de commande d'un lampemètre AVO présente une section rouge et une verte pour une indication simple, bon ou mauvais, du tube testé (cf. photo d'en-tête et figure 4). Pour les concepteurs et les développeurs qui veulent scruter davantage les caractéristiques d'un tube, le manuel d'utilisation [4] propose un choix de mesures réalisables sur l'appareil, y compris la détection



Figure 5. Il existe aussi une version portable.

AVO ? Qu'en savons-nous encore ?

La firme britannique AVO Ltd. a été fondée en 1923 sous un autre nom. Mais elle est sans aucun doute mieux connue pour ses multimètres universels, solides comme le roc, avec leurs lanières en cuir et les boîtiers en bois de chêne au départ, puis en bakélite (**figure 6**). Ce n'est pas pour rien que le nom AVOMeter™ fait référence à « Ampères, Volts & Ohms (mètre) ». Précisément ce que nous avons envie de mesurer !

Primitivement située sur Vauxhall Bridge Road au cœur de Londres, à environ un kilomètre de Buckingham Palace, la firme AVO a produit les célèbres multimètres à deux boutons rotatifs et un écran en forme de rein pendant 85 ans, entre 1923 et 2008. Actuellement, AVO fait partie de Megger Group Ltd. Mais AVO est aussi très connue pour ses lampemètres.



Figure 6. Très solides, ces multimètres AVO ! (Un exemple des années 1960)

des fuites indésirables entre électrodes, tant sur les cathodes chaudes que les froides.

Rien d'étonnant à ce que ces classiques lampemètres AVO aient été des instruments de mesure populaires pendant des décennies. Et le sont encore, en témoignent les prix d'environ mille euros pour des exemplaires d'occasion sur eBay. ◀

(180463-04 - version française : Robert Grignard)



@WWW.ELEKTOR.FR

→ Livre « Retronics » (en anglais)

www.elektor.fr/retronics

→ E-Book « Retronics » (en anglais)

www.elektor.fr/retronics-eb



QUESTIONS D'ÉTHIQUE

l'intelligence artificielle à l'état sauvage

Tessel Renzenbrink (Pays-Bas)

L'intelligence artificielle (IA) affole l'imagination parce que ce terme est immanquablement associé à la science-fiction. On pense tout de suite à Terminator et autres Robots Overlords. Ce genre de spéculations détourne en fait l'attention de la réalité des choses. À quoi donc, dans la pratique, ressemble l'application de l'IA ? L'AI Expo, qui s'est tenue à Amsterdam les 27 et 28 juin 2018, nous a permis de jeter un coup d'œil en coulisses. Ce salon s'adresse aux entreprises qui s'intéressent à « la mise en œuvre pratique de l'Intelligence Artificielle et des Mégadonnées (*Big Data*) » dans leurs activités commerciales [1]. Les conférenciers, les exposants et les visiteurs partagèrent des anecdotes sur l'utilisation de l'IA au travail. Il est particulièrement frappant de voir que les mégadonnées à fournir à la « bête » IA sont extrêmement difficiles à collecter. Sur la base de ce qui fut dit, il aurait peut-être été plus juste d'appeler ce salon celui du « défi de la rationalisation des données ».

Julio Peironcely, le responsable de l'équipe Data Science de l'aéroport Schiphol (Pays-Bas), a illustré ce problème avec un exemple pratique (la numérisation de cet aéroport). L'objectif de son département est d'utiliser les données pour rendre l'aéroport plus sûr, plus efficace et plus convivial. Un projet consiste à optimiser le temps de rotation (« *turnaround time* »), le temps nécessaire pour, après son arrivée, préparer un avion pour le départ suivant. Ce processus comporte plusieurs activités : déchargement de la cargaison, débarquement des passagers, ravitaillement en carburant et en restauration.

L'homme aux jumelles

Pour l'instant, il n'y a pas de données en temps réel disponibles sur le processus de rotation. De ce fait, la tour de contrôle ne sait pas avec précision quand la porte d'embarquement pourra à nouveau être ouverte. M. Peironcely montre l'image d'un homme dans une tour de contrôle (photo prise sur un aéroport américain) qui vérifie visuellement aux jumelles si un avion occupe une certaine porte. La situation à l'aéroport Schiphol (Amsterdam) est un peu plus moderne, la tour de contrôle peut y vérifier l'état de la porte d'embarquement à l'aide d'images

envoyées par des caméras. M. Peironcely et son équipe se sont dit qu'il devait être possible, avec des données, de faire bien plus efficace.

« Mais nous ne sommes pas les propriétaires des données », remarqua M. Peironcely. « Elles sont entre les mains de KLM, de la compagnie cargo ou d'autres compagnies de l'écosystème. » L'équipe a donc décidé d'utiliser le flux vidéo des caméras pour cartographier l'activité. L'idée consistait à utiliser les objets de la vidéo en tant qu'indicateurs. Un véhicule de restauration ou un camion-citerne présent identifie l'activité en cours. Mais les choses étaient loin d'être aussi simples.

Du concept du chat de Google à l'étudiant

Les caméras non plus n'appartiennent pas à Schiphol, mais à la sécurité. Il s'est écoulé quatre mois avant que l'équipe n'obtienne les autorisations nécessaires pour utiliser le flux vidéo. Ensuite, il a fallu identifier chaque objet sur chaque image de la vidéo. D'après M. Peironcely, les exemples de reconnaissance d'images de Google (« concept du chat ») sont très impressionnants. Mais si on lance un système de reconnaissance d'images à l'assaut d'une vidéo brute,

cela ne marche tout simplement pas. Ils ont donc embauché un étudiant qui a étiqueté manuellement tous les objets pendant deux semaines.

Les étiquettes manuelles ont ensuite été utilisées pour former un réseau neuronal. L'ordinateur a appris à reconnaître des modèles à partir des échantillons. Sur la base de ces modèles, l'ordinateur a fait des prédictions sur de nouvelles images. « Nous avons vérifié manuellement la qualité des prédictions », a confirmé M. Peironcely. Les prédictions erronées ont été réintroduites dans le modèle pour poursuivre l'entraînement du réseau neuronal.

La coopération homme-machine a finalement abouti à une représentation de tous les objets sur une ligne de temps. « Vous obtenez alors un graphique monstre indiquant le début et la fin de chaque activité », a dit M. Peironcely. « Si vous avez beaucoup de ces données, il vous est possible de voir quelles activités pourraient être effectuées avec une meilleure efficacité. »

Pour appliquer l'IA dans la pratique, il faut des données, une pléthore de données utiles et correctes. M. Peironcely montre que l'obtention de données existantes peut être difficile, car appartenant à d'autres parties. La création de

données est elle aussi un processus fastidieux. Et c'est loin d'être la seule difficulté à laquelle l'équipe de Schiphol Data Science a été confrontée. Même une fois le développement d'une application terminé, il reste des obstacles. M. Peironcely ajoute que pour la mise en œuvre d'une application, plusieurs parties doivent être prêtes à participer : compagnies aériennes, bagagistes, sécurité. Pour la plupart d'entre elles, la sécurité est une priorité absolue. Elles sont donc réticentes à modifier des procédures qui marchent bien. C'est pourquoi il faut beaucoup parler, bien expliquer les choses et être patient.

Des ordinateurs d'un autre âge dans le monde médical

Un débat d'experts sur les applications de l'IA dans les soins de santé a révélé un tableau similaire. Théoriquement, l'IA peut se traduire par d'importantes améliorations, mais la pratique est rebelle. Milan Petković, directeur du département Data Science chez Philips Healthcare, aimerait que l'accent soit mis sur la pré-

Lien

[1] <https://www.ai-expo.net/europe/collateral/>

vention plutôt que sur la guérison. L'IA pourrait bien aider dans ce cadre, par exemple en surveillant, à l'aide de capteurs portables, des fonctions corporelles telles que le pouls et la pression artérielle ou en réalisant des études de fond à long terme sur les effets des habitudes alimentaires sur la santé. « Mais cela exige beaucoup de données, et constitue un défi », a confirmé M. Petković.

Le partage des données dans le secteur médical est difficile. Il s'agit de données sensibles et personnelles qui ne peuvent pas être simplement partagées. En outre, les sources de données sont fragmentées. Pour M. Petković, le paysage est décentralisé : « Il y a beaucoup de parties différentes, de sorte que les données se retrouvent dans des silos. » Si déjà ces parties acceptent de partager leurs données, il reste des obstacles tech-

niques. « L'interopérabilité est un défi », déclare sans ambages M. Petković. Dans le secteur des soins de santé, il existe de nombreux systèmes TIC démodés. Il est impossible de les relier les uns aux autres d'un coup de baguette magique. De plus, ces systèmes anciens ne répondent souvent pas aux normes actuelles de protection de la vie privée et de sécurité, ce qui signifie qu'ils ne peuvent être ouverts à des tiers.

Les idées de M. Petković – tout comme l'exemple pratique de M. Peironcely – montrent clairement que l'application de l'IA aux systèmes et processus existants est loin d'être gagnée. ◀

(180462-04 -

version française : Guy Raedersdorf)



Aperçu du processus de rotation à l'aéroport de Gardermoen, Oslo. Photo : Trond Kvitvik (licence Creative Commons BY 3.0).



bienvenue dans votre e-choppe

la rédaction recommande



YDLidar X4

Le système YDLidar X4 est un détecteur Lidar extrêmement peu coûteux, essentiellement destiné aux applications de robotique. Un dispositif Lidar (*Light Detection and Ranging* - en français, détection et estimation de la distance par laser) utilise un laser impulsionnel pour mesurer la distance par rapport à des objets. Le signal laser est réfléchi par la surface de l'objet,

et la mesure du temps nécessaire à la lumière pour atteindre le capteur permet de déterminer la distance. Le détecteur YDLidar X4 utilise une tête rotative qui contient à la fois le laser et le récepteur. Le système peut ainsi « voir » à 360° autour de lui, mais seulement en deux dimensions, cela va de soi.

Thijs Beckers (rédacteur chez Elektor)



www.elektor.fr/ydlidar-x4

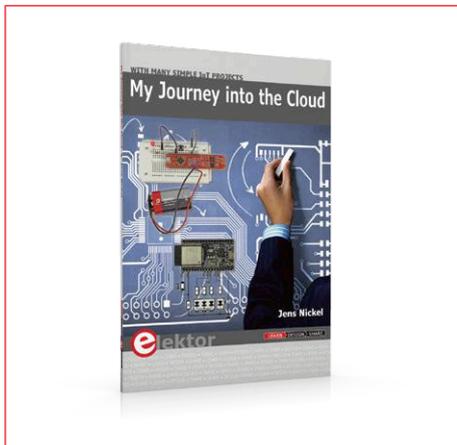
Vos favoris :

1. RasPad – tablette Raspberry Pi
www.elektor.fr/raspad



2. Testeur intelligent de CMS MS8911 (LCR-mètre) de Mastech
www.elektor.fr/ms8911
3. Walabot Creator Pack – capteur 3D programmable
www.elektor.fr/walabot
4. Raspberry Pi 3 (modèle B+)
www.elektor.fr/raspberry-pi-3-model-b-plus
5. Kit de l'horloge à tubes VFD avec ESP32
www.elektor.fr/vfd-horloge
6. The Official ESP32 Book (livre en anglais)
www.elektor.fr/esp32-book

Mon voyage dans le nuage

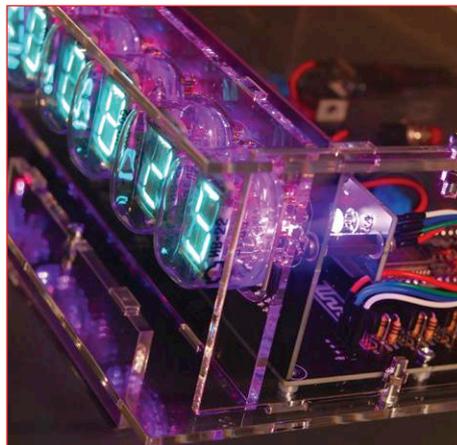


Dans une série d'articles en ligne, notre rédacteur en chef allemand a proposé plusieurs projets de démonstration (commande de luminaire dans un réseau domestique, envoi de données dans le nuage à partir d'une carte dotée de capteurs...) qui abordent différents thèmes : TCP/IP, MQTT, contrôle par smartphone, accès Wi-Fi, connexion à un service cloud, programmation orientée objet... Ce livre (en anglais) rassemble les 24 premiers épisodes de cette série sur l'IdO.

Prix (membres) : 13,46 €

www.elektor.fr/journey-into-cloud

Horloge à tubes VFD et liaison Wi-Fi par ESP32

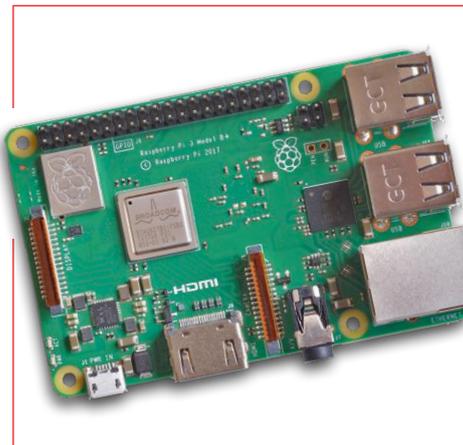


Cette horloge à l'allure vintage affiche l'heure et de la date sur six tubes russes, des VFD type IV-22 et deux indicateurs VFD DM160 (traits séparateurs entre heures, minutes et secondes). Pour sa mise à l'heure, elle se connecte par Wi-Fi à un serveur NTP de l'internet. La structure de l'horloge est modulaire : module DevKitC ESP32 d'Espressif compatible Arduino, carte d'affichage et carte d'éclairage RVB d'arrière-plan.

Prix (membres) : 152,96 €

www.elektor.fr/vfd-horloge

Raspberry Pi 3 (modèle B+)



Par rapport à celles de son prédécesseur, la vitesse et la puissance de ce Raspberry Pi 3 B+ ont gagné globalement 15 %. Cette amélioration significative du RPi 3 B+ n'entrave nullement sa compatibilité (électrique et mécanique) avec les modèles RPi 3 B, 2 et B+.

Prix (membres) : 35,50 €

www.elektor.fr/rpi3b-plus

**nouveau**

Android App Development

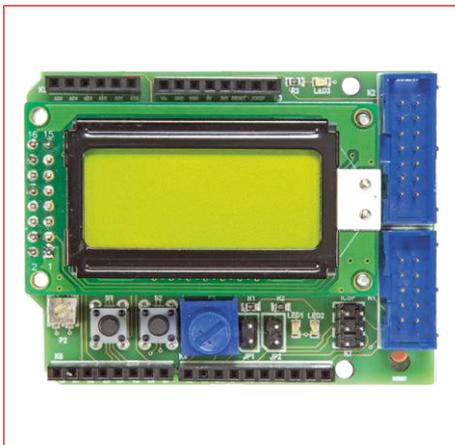
With Basic for Android – B4A

Les étudiants et les bricoleurs de tout poil sont amenés à concevoir des applis pour les appareils mobiles sous Android en utilisant le langage de programmation *Basic For Android* (B4A) et son environnement de développement intégré. Dans ce livre (en anglais), Dogan Ibrahim présente des projets simples pour s'initier à la syntaxe et à la programmation de B4A.

Les électroniciens apprécieront ce livre parce qu'il décrit la communication entre un appareil mobile Android et différentes plateformes (comme Raspberry Pi, Arduino et module ESP32) via une liaison Wi-Fi ou des SMS. Certains des projets montrent comment envoyer des données depuis Raspberry Pi, Arduino ou ESP32 vers un téléphone mobile Android et les y afficher.

**Prix (membres) : 24,95 €****www.elektor.fr/android-app-development**

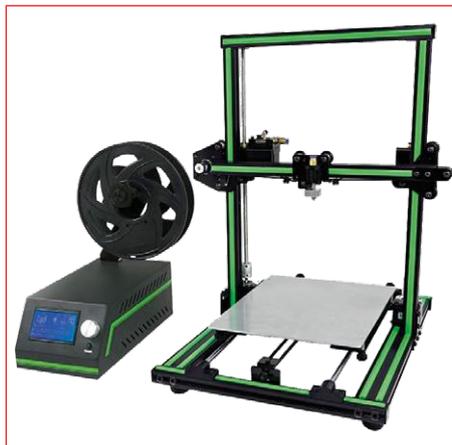
Shield d'expérimentation Arduino version 2.0



Le *shield* proposé en 2014 améliorait la carte Arduino avec des LED, des boutons-poussoirs, un écran LCD et un accès à l'interface UART et aux broches du microcontrôleur. Ce *shield* a suscité – et suscite encore – beaucoup d'intérêt ; une révision s'imposait donc ! La version 2.0 de ce *shield* permet maintenant de connecter n'importe quel écran LCD standard avec connexion à 2x8 trous.

**Prix (membres) : 29,25 €****www.elektor.fr/160593-91**

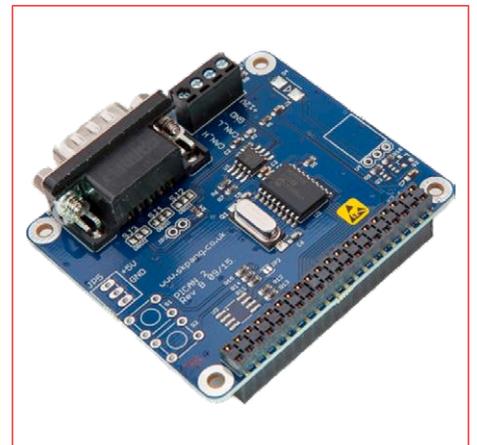
Anet E10 : nouvelle imprimante 3D en kit



Pas envie, ni le temps d'assembler des pièces détachées ? L'imprimante 3D Anet E10, en kit et assemblée en un ¼ h, est idéale pour vous. Elle est silencieuse, robuste (châssis en aluminium), rapide, précise et elle imprime des objets de grande taille. Vidéo sur le déballage et le montage : il suffit d'assembler le socle (muni de son plateau chauffant) et le portique, fournis montés l'un et l'autre. Pas de soudures, mais des connecteurs étiquetés.

**Prix (membres) : 289,00 €****www.elektor.fr/anet-e10**

PiCan 2 – Carte de bus CAN pour RPi



Cette carte PiCAN 2 « ouvre la porte » du bus CAN au Raspberry Pi. Cette carte renferme un contrôleur CAN MCP2515 de Microchip avec un émetteur-récepteur CAN MCP2551. Le raccordement s'effectue via une prise DB9 ou un bornier à vis à trois voies. Le pilote SocketCAN est facile à installer. La carte est programmable en C ou Python.

**Prix (membres) : 35,96 €****www.elektor.fr/pi-can2**

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 !

Où envoyer votre réponse ?

Envoyez votre réponse (les chiffres sur fond grisé) avec vos coordonnées par courriel, avant le **29 novembre 2018** à l'adresse **hexadoku@elektor.fr**

Les gagnants

La solution de la grille du numéro de septembre/octobre 2018 est **D83E1**.

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

Manfred **Häfner** (Allemagne), Olli **Hakala** (Finlande), Alexandr **Papazyan** (Russie), Axel **Stefanini** (France), Brian **Wood** (Royaume-Uni).

Bravo à tous les participants et félicitations aux gagnants !

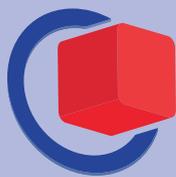
		3		F	9				6	A	5		C		
F							7		8			E	6	9	4
		5	8	1			B		9					7	A
	B	C	D				7			F					3
							C	5	1		4		7		
		9				0	E	3		D	2			1	F
D		0	5	3	7		2		6				4		
	1		4		8					0				6	
	4				D					1		6		2	
		D				8		F		C	7	0	A		E
8	0			7	E		4	9	B					C	
		A		0		2	1	D							
5					1			6					2	B	3
2	A					3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

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

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.

2ew19P
E-Code pour l'entrée gratuite
▶ embedded-world.de/voucher

Nuremberg, Allemagne
26 – 28.2.2019



embeddedworld

Exhibition & Conference

... it's a smarter world

AUJOURD'HUI, DEMAIN ET AU-DELÀ

Une adresse unique regroupant toute la gamme des systèmes embarqués : venez découvrir plus de 1 000 entreprises et laissez-vous inspirer par les dernières tendances, les nouveautés, des conférenciers renommés et des présentations captivantes.

Toujours actuel :

▶ embedded-world.de

Partenaires médias

Markt&Technik
Die unabhängige Wochenzeitung für Elektronik

DESIGN & ELEKTRONIK
KNOW-HOW FÜR ENTWICKLER

Elektronik
Fachmedium für industrielle Anwender und Entwickler

Elektronik automotive
Fachmedium für professionelle Automobilelektronik

Computer & AUTOMATION
Fachmedium der Automatisierungstechnik

SmarterWorld
Solutions for a Smarter World

MEDIZIN+elektronik
Fachmedium für Elektronik in der Medizintechnik

elektroniknet.de

Organisateur du salon

NürnbergMesse GmbH

T +49 9 11 8606-49 12

F +49 9 11 8606-49 13

visitorservice@nuernbergmesse.de

Organisateur des congrès

WEKA FACHMEDIEN GmbH

T +49 89 2 55 56-13 49

F +49 89 2 55 56-03 49

info@embedded-world.eu

NÜRNBERG MESSE

- Plus de 45 ans d'expérience
- Grande disponibilité et livraison garantie
- Plus de 100 000 produits

POUR AMATEURS ET PROFESSIONNELS – INDISPENSABLE POUR L'ÉLECTRONIQUE

Robuste multimètre numérique avec éclairage LED

Parfaitement conçu pour un emploi quotidien, à l'extérieur, pour la maintenance, le BTP et l'industrie, mais aussi pour les bricoleurs avertis.

- Eclairage automatique des touches de fonction et du bouton rotatif
- Puissante lampe LED pour éclairer le poste de travail
- Affichage des mesures à 3 5/6 digits (soit 6 000 counts)
- Afficheur LCD avec mesure relative et diagramme à barres 42 segments
- Contrôleur de continuité avec ronfleur et test des diodes

Référence : PEAKTECH 3443

NOUVEAU **80,69** (€ 67,24)



PeakTech®



UTILISABLE DANS L'OBSCURITÉ



30 MODES DE MESURE AUTOMATIQUES ET FONCTION FFT



PeakTech®

Oscilloscope numérique compact à mémoire

Impressionnant rapport qualité-prix : la technologie moderne et la commande simplifiée permettent un grand nombre d'applications, à l'école, au travail ou à la maison.

- 20 MHz, 2 canaux
- Taux d'échantillonnage : 100 Méch/s par canal
- Afficheur couleur TFT 17,5 cm (7") haute définition avec 800 x 480 pixels
- Connexion USB pour transmission des données en temps réel
- Connecteur USB pour support de données externe USB
- Design pratique du boîtier plat avec poignée

Référence : PEAKTECH 1335

NOUVEAU **301,66** (€ 251,38)



Référence :	€	Largeur de bande	Canaux	Taux d'échantillonnage
PEAKTECH 1335	301,66	20 MHz	2	100 Méch/s par canal
PEAKTECH 1336	372,25	50 MHz	2	500 Méch/s par canal (non illustré)
PEAKTECH 1337	452,92	100 MHz	2	1 Géch/s par canal (non illustré)

Prix du jour! Prix à la date du: 24. 9. 2018

Les langues de notre boutique:

Les réglementations légales en matière de résiliation sont applicables. Tous les prix sont indiqués en € TVA légale incluse, frais d'envoi pour l'ensemble du panier en sus. Seules nos CGV sont applicables (sur le site <https://rch.it/CG-FR> ou sur demande). Semblables aux illustrations. Sous réserve de coquilles, d'erreurs et de modifications de prix.
reichelt elektronik GmbH & Co. KG, Elektronikring 1, 26452 Sande (Allemagne), tél. +33 97 518 03 04

www.reichelt.fr
ASSISTANCE TÉLÉPHONIQUE: +33 97 518 03 04