

station météo à ESP32

lecture des
données des capteurs
à distance (Wi-Fi)



poste de soudage
de CMS compact

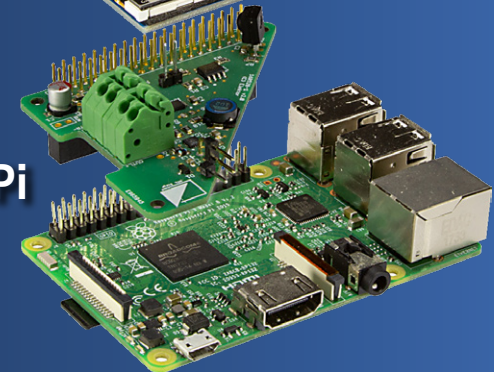


à régulation
de température précise et rapide

radio FM
avec
RDS



HAT pour
Raspberry Pi



feu de phare 2.0 ✕ B.A.-BA des mélangeurs de fréquences radio ✕ *shield* anti-ronflement ✕ Android Things prend son envol ✕ *shield* cryptographie à masque jetable ✕ se lancer dans Verilog ✕ télécommande/télémessure en semi-duplex à 2,4 GHz ✕ la commande des moteurs électriques ✕ speed-o-mètre pour coureurs ✕ Q & R : les boîtiers ✕ carte Pobjuino ✕ trigger de Schmitt à amplis op



FORUMLED europe

THE INTERNATIONAL CONGRESS
FOR LED TECHNOLOGY

NEW DATES
NEW PLACE

13 & 14
FEBRUARY 2019
EUREXPO - LYON

ALL ABOUT LED TECHNOLOGY AND ITS APPLICATIONS

10TH EDITION OF THE CONGRESS FORUMLED EUROPE

- + Market trends
- + Standards & labels
- + Technological innovations
- + Smart lighting & IoT
- + Human Centric Lighting
- + LED Technology & Health
- + Display
- + Horticulture
- + Embedded lighting...

**From the electronic
component to the lighting
source and its applications**

ASK FOR YOUR BADGE

on www.lighting-days.com
code to enter PMLDP

Preferential price
before 31 December 2018

Event organized as part of the **lightingdays**, the Lighting and LED Technology Rendez-vous !



CONTACT: Caroline ROQUES
+33 (0)4 78 176 184 | caroline.roques@gl-events.com

www.lighting-days.com
@LightingDays19

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 :

© 2019 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.



Tous au recyclage

Au cours des dernières semaines, notre équipe francophone a participé à différents salons. C'est une belle expérience, toujours enrichissante, d'y rencontrer nos lecteurs avec leurs compliments, mais aussi leurs critiques.

La Cité des Sciences et de l'Industrie a été le théâtre (grandiose) de notre dernière apparition publique en novembre dernier, à l'occasion de la Maker Faire Paris. Cet événement regroupe des bricoleurs de tout poil venus présenter et partager leurs créations. Cette réunion de famille des *makers* compte aussi bien des adeptes de Raspberry Pi et d'Arduino que des forgerons, des menuisiers ou des couturiers...

L'un des thèmes de cet événement de belle ampleur était l'économie circulaire. Certains projets étaient de parfaits exemples de recyclage, voire de surcyclage (*up cycling*) : un sommier à lattes converti en solide remorque de vélo, des bouchons transformés en filament d'imprimante 3D... Que pouvait bien faire là une grande enseigne française spécialisée dans l'électroménager et le multimédia ? Ne sont-ce pas des appareils par définition neufs qu'elle devrait vendre ? En fait, depuis deux ans, cette société récupère les appareils usagés ou cassés, les répare et les remet en vente, mais seulement sur l'internet. Aujourd'hui, comme les habitudes changent et que les consommateurs sont prêts à acheter des appareils reconditionnés, elle souhaite communiquer davantage sur cette forme inédite de (re)distribution.

L'innovation en matière de récupération et de recyclage concerne aussi le labo d'Elektor et les électroniciens. Avec la pénurie (organisée) de composants électroniques, nous allons devoir nous y mettre aussi (voir à ce sujet l'article « bruits de labo » de ce numéro).

Vous avez sûrement des montages qui dorment au fond de vos tiroirs. Et si vous les sortiez pour leur donner une seconde vie ?

Bonne et heureuse année 2019 !

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 Duchesnes, Pascal Godart, Yves Georges, Robert Grignard, Denis Lafourcade, Jean-Louis Mehren, Denis Meyer, Hervé Moreau, Helmut Müller, Kévin Petit, Xavier Pfaff
Service de la clientèle :	Cindy Tijssen
Graphistes :	Giel Dols, Mart Schroijsen, Patrick Wielders
Elektor en ligne :	Daniëlle Mertens

station météo à ESP32

lecture des données des

- 5 bientôt dans Elektor
- 20 **Fast Forward Award 2018 : cérémonie de remise des prix au salon electronica !**
ils inventent aussi notre futur
- 34 **vol tous azimuts**
Obsolescence programmée ?
Un faux procès !
- 79 **agenda**
janvier 2019
- 80 **Q & R**
(presque) tout ce que vous avez toujours
voulu savoir sur... les boîtiers
- 97 **agenda**
février 2019
- 99 **offres d'emploi**
- 106 **Rétronique**
mise en conformité et régulation de la
tension du secteur
plus question de péter les plombs !
- 110 **questions d'éthique**
système de fichiers interplanétaire
une nouvelle Toile mondiale à l'horizon...
- 112 **l'e-choppe d'Elektor**
- 114 **hexadoku**
casse-tête pour elektorniciens

poste de soudage de CMS compact

14

à régulation de température
précise et rapide



Le succès du poste de soudage de CMS à Platino publié en 2015 avait largement dépassé le cadre du lectorat d'Elektor. Imprégnés comme nous le sommes de l'esprit d'ingénieur, un projet qui fonctionnait aussitôt assemblé ne pouvait susciter chez nous que l'envie de l'améliorer. D'où ce nouveau poste de soudage, entièrement repensé mais toujours aussi compact.

en coulisse

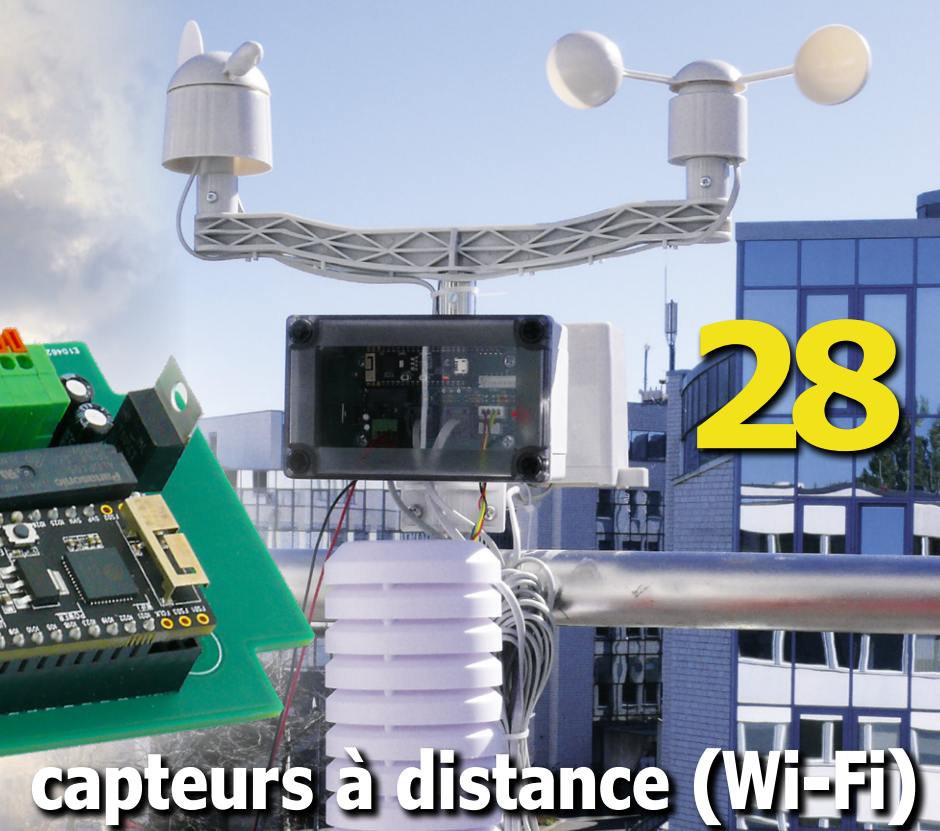
- 6 **la commande des moteurs électriques**
par contacteur, MLI ou pont complet
- 22 **projet 2.0**
corrections, mises à jour et
courrier des lecteurs
- 33 **bruits de labo**
obsolète ou « attribué » ?
- 38 **trucs et astuces**
adaptateur pour tester des CMS
- 68 **Android Things prend son envol**
grâce au langage C++
- 74 **trigger de Schmitt à amplis op**
calculs et circuits
- 82 **embrouillamini d'ampli**
ça marche, mais pas comme on le pense !
- 94 **se lancer dans Verilog**
les fondamentaux du langage de description
de matériel
- 100 **hors-circuits de R. Lacoste**
B.A.-BA des
mélangeurs de fréquences radio
tout ce qu'il faut savoir pour bien les choisir

6

la commande des moteurs électriques

par contacteur,
MLI ou
pont complet

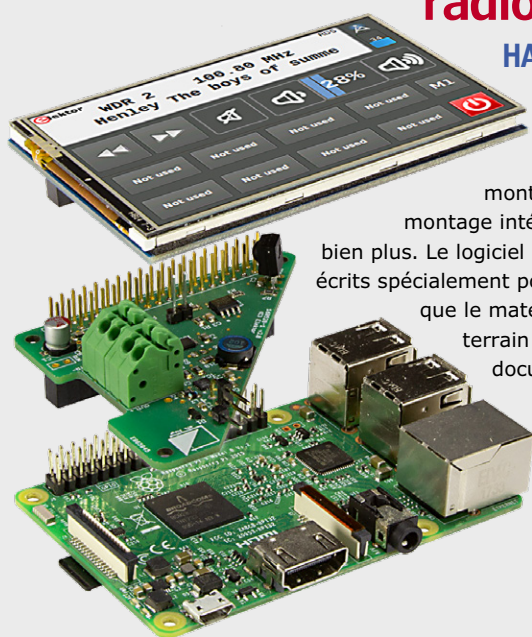




capteurs à distance (Wi-Fi)

radio FM avec RDS HAT pour Raspberry Pi

Construire une radio FM avec un nano-ordinateur Raspberry Pi et un tricorne fait maison monté en impériale (HAT), voilà un montage intéressant. Toutefois ce projet offre bien plus. Le logiciel (pilotes et interface graphique écrits spécialement pour cette application) ainsi que le matériel constituent un formidable terrain d'expérimentation assorti d'une documentation fouillée.



86

- 28 station météo à ESP32**
lecture des données
des capteurs à distance (Wi-Fi)
- 40 shield anti-ronflement**
pour des nuits enfin reposantes
- 45 speed-o-mètre pour coureurs**
avec totalisateur et enregistreur
- 48 shield cryptographe à masque jetable**
misez sur la transparence,
gagnez en sécurité
- 56 télécommande/télémessure
en semi-duplex à 2,4 GHz**
bon marché et simple
grâce aux modules prêts à l'emploi
- 62 carte PobDuino**
Flowcode et Arduino alliés contre
l'obsolescence
- 86 radio FM avec RDS**
HAT pour Raspberry Pi

 **bientôt sur ces pages**

Extrait du sommaire du prochain numéro :

- Baristor
 - Conception matérielle avec (V)HDL (3)
 - Commande pour la découpe de polystyrène
 - Générateur de fonctions simple
 - Kits à microcontrôleurs pour les débutants
 - La pratique des amplificateurs opérationnels (2^e partie)
 - Le projet IOTA et Raspberry Pi
 - Microprocesseur fait maison
 - Règle Raspberry Pi
 - Spectromètre audio
 - Surveillance de la pollution de l'air
- Etc.

Sous réserve de modification.

Le numéro de mars-avril 2019 paraîtra le 21 février 2019.

sur la scène : les projets

- 13 diviseur de fréquence avec facteur de division entier réglable**
pour trois fois rien (de CMOS)
- 14 poste de soudage de CMS compact**
à régulation de température
précise et rapide
- 24 feu de phare 2.0**
à signature lumineuse réglable





la commande des moteurs électriques

par contacteur, MLI ou pont complet

Thomas Scherer (Allemagne)

Pour convertir le courant électrique en énergie mécanique, on a le plus souvent recours au bon vieux moteur électrique. C'est parfait pour les électroniciens, car ce genre de moteur contient beaucoup moins d'éléments mécaniques « déroutants »

qu'une machine à vapeur ou un moteur Otto. Toutefois rien de nouveau pour les électroniciens ! Il y a longtemps que les moteurs électriques ont été inventés et que leur technologie ne subit plus de bouleversements. Mais des variétés de moteurs électriques, il en existe des quantités, et chacune demande à être commandée de manière adaptée à son utilisation.

À la différence des moteurs thermiques où quelque chose de chaud met des pièces en mouvement avec beaucoup de bruit et de mauvaises odeurs, par l'intermédiaire d'éléments comme des vilebrequins, il y a chez les moteurs électriques une bien plus grande diversité. Les moteurs électriques sont le plus souvent couplés directement aux arbres de transmission, ce qui

non seulement simplifie la mécanique et l'usage, mais diminue aussi le poids et le prix. Cette simplicité améliore la durabilité et rend possible une amélioration de la précision du pilotage. Mais on la paye avec l'inconvénient d'avoir quasiment toujours à traîner un câble électrique, ou alors une batterie, ce qui rend la chose plus lourde, plus chère et plus compliquée.

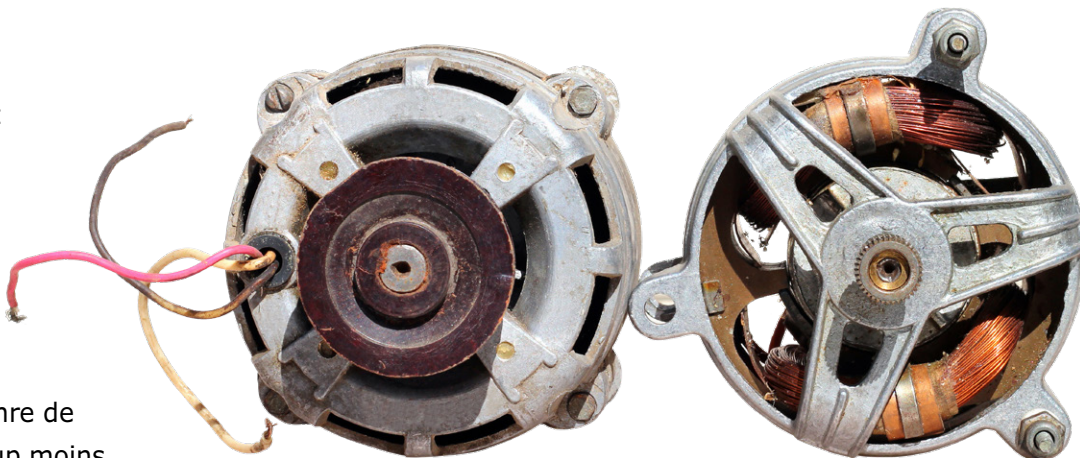


Figure 1. Moteur Jedlik de 1827 (illustration : Wikimedia Commons [2]).

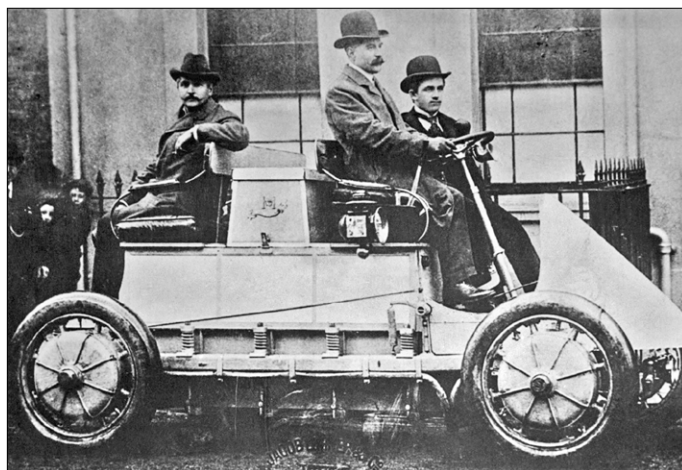


Figure 2. La légendaire Lohner-Porsche, photographiée vers 1900. Propulsion par quatre moteurs moyens de 1,5 kW chacun (photo : Michael Hereward Westbrook dans « The Electric Car » [3]).

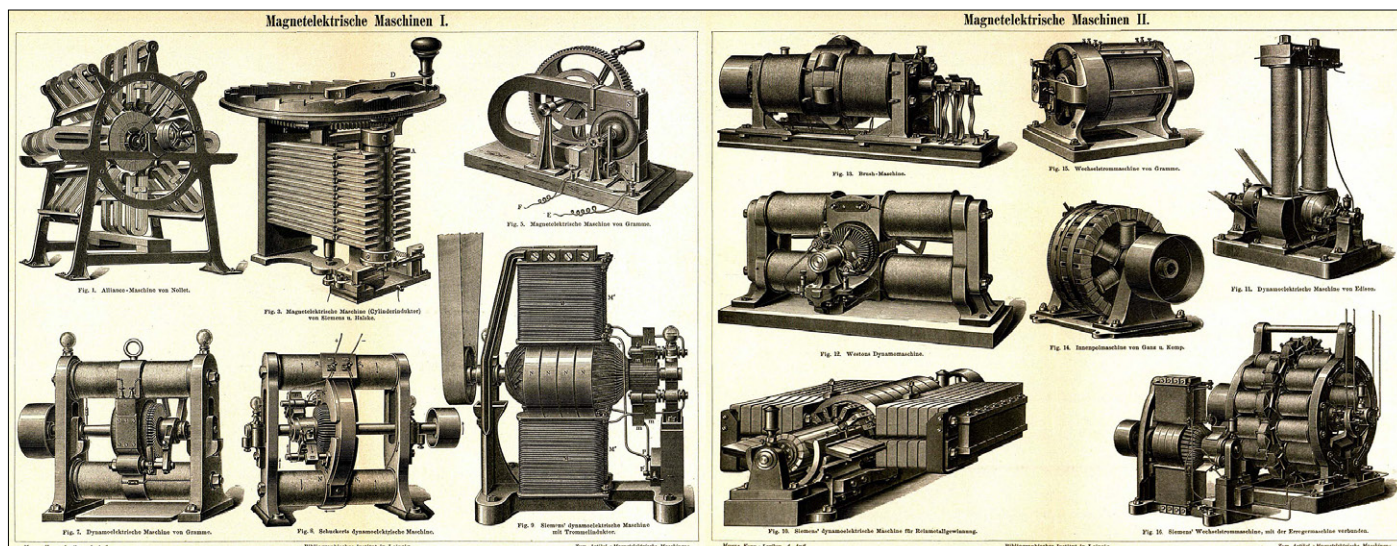


Figure 3. Déjà en 1885, le dictionnaire de conversation Meyer (4^e édition) présentait différents modèles de moteurs [4][5].

Multiplicité

Par sa découverte, il y a près de 200 ans, qu'un conducteur parcouru par un courant influence une boussole, le Danois Hans-Christian Ørsted donna le coup d'envoi de l'électrotechnique en général et du développement rapide des moteurs électriques en particulier (**fig. 1**). À peine vingt ans plus tard, on pouvait voir à Saint-Petersbourg un bateau à propulsion électrique naviguer sur la Neva. Et chaque *Homo technicus* sait qu'au début de l'histoire de l'automobile, il y avait plus de voitures électriques que de voitures à essence (**fig. 2**). Mais il

fallut attendre le développement des réseaux électriques par Edison (General Electric) et Tesla (Westinghouse) aux États-Unis ainsi que par AEG dans l'Empire allemand de jadis, pour que les moteurs électriques trouvent leur emploi dans les applications les plus diverses. Les chevaux, l'hydraulique et l'éolien passèrent au second plan et on se mit à envisager des possibilités de motorisation dont il n'était même pas question de rêver auparavant.

À la fin du 19^e siècle, les bricoleurs et les ingénieurs s'en donnèrent à cœur joie et concurent les types de moteurs électriques

Tableau 1. Types de moteurs

Désignation	Puissance	Caractéristiques typiques/particulières
Moteur asynchrone triphasé	W - MW	Rotor passif (pas de collecteur) qui suit le champ du stator, commande par contacteur ou variateur de vitesse, fort courant de démarrage (commutation étoile/triangle)
Moteur synchrone triphasé	W - MW	Excitation nécessaire (aimant permanent ou électroaimant), rotor synchrone avec le champ tournant, variateur de vitesse, champ tournant autopiloté par électronique, générateur, machine à pôles internes et externes
Moteur pas-à-pas	<1 W - 1 kW	Variante du moteur synchrone avec rotation du rotor d'un angle donné, positionnement précis, 24 à 400 pas par tour
Moteur BLDC	W - MW	Moteur à courant continu sans balais = variante du moteur synchrone à commutation électronique, pour entraînements, très fiable, souvent avec aimant permanent
Moteur à condensateur	W - <1 kW	Moteur asynchrone/synchrone monophasé, champ tournant produit par condensateur de déphasage et bobine secondaire, faible couple de démarrage, simple, nécessite peu d'entretien
Moteur à enroulements en court-circuit	W - <1 kW	Moteur asynchrone/synchrone monophasé, champ tournant produit par enroulements en court-circuit sur pôles divisés, faible couple de démarrage, fonctionnement silencieux, nécessite peu d'entretien, faible rendement
Moteur à courant continu	W - kW	Excitation permanente ou électrique, champ alternatif produit par collecteur mécanique, beaucoup de variantes, série ou parallèle, fort couple de démarrage
Moteur série monophasé	W - kW	Comme le moteur à courant continu avec des tôles pour limiter les pertes par courants de Foucault, vitesse de rotation élevée, usure des balais, CEM, fort couple de démarrage
Moteur linéaire	W - kW	Pas de rotor, mouvement linéaire, couple élevé, beaucoup de variantes, positionnement, disques durs, train Transrapid
Servomoteur	W	Combinaison de moteur électrique à commande et capteur d'angle de rotation, modélisme, commande par MLI, trois fils

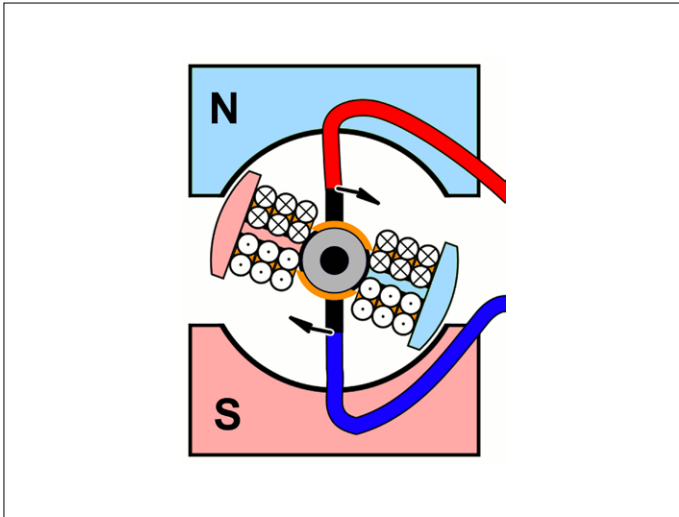


Figure 4. Principe de fonctionnement et structure d'un moteur électrique simple à excitation permanente. Sur le stator, le pôle nord est en haut et le pôle sud est en bas. Le rotor est un électroaimant alimenté par l'intermédiaire d'un collecteur en anneau de cuivre et des balais en graphite (d'après Michael Frey, CC 3.0 [6]).

les plus divers. L'expression « Guerre des courants » [1] vous dit-elle quelque chose ? En 1890, il y eut un affrontement entre General Electric et Westinghouse, c'est-à-dire entre leurs grands inventeurs respectifs Edison et Tesla, sur la question fondamentale : courant continu ou courant alternatif ? Il y avait déjà à cette époque des moteurs électriques qui non seulement se distinguaient par leur type et leur puissance (**fig. 3**), mais aussi par leur spécialisation en moteurs continus ou alternatifs et, pour ces derniers, selon le nombre de phases du courant. On disposait donc d'un moteur particulier pour chaque utilisation. La multiplicité des désignations avec toutes les différences qui s'y attachent est particulièrement déroutante pour l'électronicien sans connaissances particulières dans le domaine des moteurs électriques. Le **tableau 1** ne présente que les catégories de moteurs les plus générales.

Principes

Comme dans cet article il est question de pilotage et de commande, je ne m'étendrai pas sur les mystères du mode de fonctionnement de chaque type de moteur. Dans son principe, un moteur électrique n'est rien d'autre qu'un assemblage d'un ou plusieurs électroaimants, dont le champ magnétique exerce une force sur un aimant permanent, un autre électroaimant ou sur une pièce en fer de forme appropriée, pour obtenir un mouvement mécanique. De l'énergie électrique est alors convertie en énergie mécanique.

Cette conversion n'est jamais totale, car non seulement une partie de l'énergie est perdue dans des effets mécaniques comme le frottement, mais il y a des effets électriques et électromagnétiques qui grignotent le rendement, à commencer par la résistance ohmique des enroulements, source de chaleur, plutôt que d'énergie mécanique. Cela est encore augmenté par l'effet de peau en cas de courant alternatif si celui-ci est à haute fréquence. Cet effet est nul en continu, très modéré et sans véritable conséquence pratique à 50 ou 60 Hz, mais à 30 kHz, la profondeur de pénétration du courant est déjà limitée à 0,4 mm. Il s'y ajoute des phénomènes comme les pertes

par courant de Foucault dans les pièces en fer du moteur si elles sont soumises à des champs magnétiques variables. Enfin, il y a des effets secondaires, comme ce qui se passe sur les collecteurs et les balais, pour les moteurs qui en sont équipés. Enfin, les pertes dans les organes de commande jouent un rôle non négligeable. Dans le cas de commutateurs électromécaniques (contacteurs et relais), il s'agit des pertes thermiques dans les bobines, ainsi que dans les résistances des contacts et du câblage qui sont habituellement très faibles. Dans une commande à semi-conducteurs, qui s'impose s'il s'agit de commande à modulation de largeur d'impulsion (MLI), la consommation d'énergie est le fait de l'électronique de commande avec son alimentation (la part la plus faible) et des chutes de tension dans les semi-conducteurs commutés, ainsi que des pertes dues à la commutation rapide, obligatoire en MLI. Les MOSFET de puissance modernes sont assez bien adaptés pour des puissances de moteurs faibles et moyennes (jusqu'à quelques kilowatts), car leur résistance à l'état conducteur est de l'ordre de quelques milliohms. Mais pour de plus grosses puissances, comme dans le domaine de l'automobile, où il faut des tensions plus élevées, la résistance de canal à l'état conducteur augmente sur les MOSFET au silicium, ce qui a accru l'emploi de semi-conducteurs au carbure de silicium.

Le carbure de silicium supporte des champs plus intenses et permet donc des couches plus minces. Les SiC-MOSFET sont donc particulièrement bien adaptés aux tensions plus élevées, offrent des fréquences de commutation plus rapides et des résistances de conduction plus basses, donc des pertes moindres. De plus, ils peuvent fonctionner à des températures plus élevées, ce qui améliore la fiabilité. Mais comme souvent dans la vie, ces avantages se paient un peu plus cher.

Modes de fonctionnement

Comme déjà mentionné, la description détaillée du mode de fonctionnement de chaque type de moteur nous entraînerait trop loin. Toutefois, pour le thème « commande et pilotage », il est utile d'avoir en tête le principe de fonctionnement d'un moteur électrique.

La **figure 4** présente la structure de base d'un moteur électrique simplifié. À l'extérieur, c'est-à-dire en haut (nord) et en bas (sud) de la figure se trouvent les pôles d'un aimant permanent. Comme dans cette variante, il ne bouge pas, on l'appelle le stator (ou bien encore carcasse ou inducteur). La partie intérieure qui tourne est le rotor (ou induit), dont le noyau en fer est entouré d'une bobine en fil de cuivre isolé. Le rotor est conçu comme un électroaimant tournant. Cette structure est dite à rotor intérieur mobile. La structure opposée est tout à fait concevable : une partie intérieure fixe entourée d'un stator mobile, c'est la structure du moteur moyeu, utilisé pour les véhicules électriques.

Si l'on fait circuler un courant continu dans le bobinage, le pôle nord bleu du rotor sera attiré par le pôle sud rouge du stator et, de même, le pôle sud du rotor trouvera le pôle nord du stator irrésistible. Il se produira alors une secousse, le rotor tournera de 60° dans le sens des aiguilles d'une montre puis restera immobile, figé.

Comme cela n'aurait guère d'intérêt, le courant de l'enroulement du rotor est amené par des balais (en noir sur la figure, le plus souvent en graphite, mais sous la forme de lames de ressort en métal précieux pour les plus petits moteurs) frottant sur un anneau (en orange sur la figure) divisé en deux.



Figure 5. Le thyristor de grande puissance (4,2 kV pour 2,4 kA) a un diamètre de 11 cm (photo : Johnny.m76 / CC 3.0 [7]).

Cette structure, appelée collecteur, commute le courant dans la bobine en fonction de la position du rotor, de sorte que la polarité de l'électroaimant s'inverse au passage à la verticale, créant un couple permanent dans l'un des sens de rotation. Ce principe est aussi simple que génial. Et on peut en tirer des tas de variantes. On peut intervertir non seulement l'intérieur et l'extérieur, mais aussi l'électroaimant et l'aimant permanent. De plus, on peut remplacer l'aimant permanent par un second électroaimant. Dans l'exemple de la figure 4, le stator aurait alors lui aussi un enroulement. Et déjà on voit apparaître un éventail de possibilités : le second enroulement peut être mis en série avec le premier, mais aussi en parallèle. Dans le premier cas, on obtient un moteur série, appelé aussi non sans raison moteur universel. On le trouve presque partout, dans les perceuses jusque dans les aspirateurs ainsi que dans des applications industrielles. Universel, il l'est aussi parce qu'on peut l'alimenter indifféremment en courant continu ou alternatif. L'inconvénient du collecteur est le frottement. Les balais s'usent, ainsi qu'à plus long terme les surfaces de contact du collecteur. C'est pourquoi on a conçu des moteurs qui utilisent le champ tournant produit par un réseau multiphasé et qui peuvent se passer de collecteur. Il y a encore d'autres types qui créent un champ tournant à partir du monophasé au moyen de spires en court-circuit. Et de nos jours, on peut remplacer les collecteurs par des commutateurs électroniques inusables pilotés soit par des capteurs de position situés dans le moteur, soit par la force contre-électromotrice. Avec des onduleurs, on peut faire varier dans certaines limites la vitesse des moteurs synchrones qui serait sinon rigidement liée à la fréquence du secteur. Toutes ces variantes sont à l'origine d'une multiplicité de types de moteurs grossièrement répartie en catégories dans le tableau 1.

Tout aussi intéressant : presque tous les types de moteurs peuvent fonctionner en générateurs et restituer de l'énergie électrique à partir de l'énergie mécanique. Quelquefois, cela ne demande que des adaptations mineures. Dans certains cas, c'est utilisable : on peut par exemple freiner des moteurs en dissipant l'énergie cinétique résiduelle après coupure de l'alimentation sous forme de chaleur (éventuellement dans la bobine elle-même). Dans d'autres cas, cela peut s'avérer

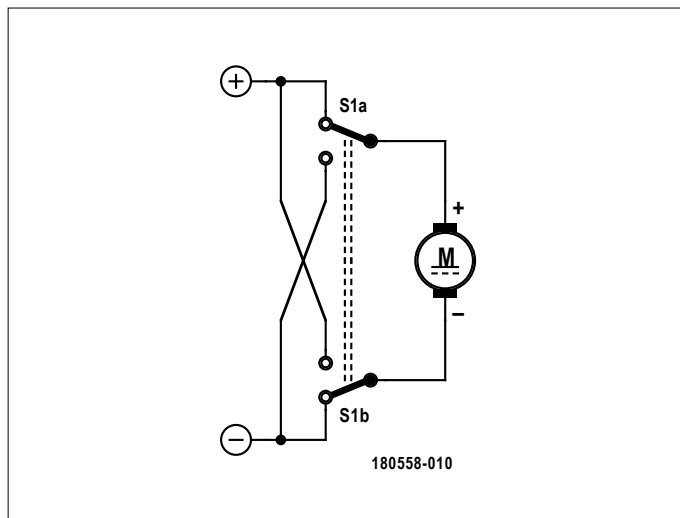


Figure 6. On obtient le changement de sens de rotation d'un moteur à courant continu au moyen de deux inverseurs couplés.

néfaste, si la rotation du moteur produit un effet de générateur susceptible d'activer l'électronique de commande.

Commandes

La multiplicité des types de moteurs électriques entraîne celle des types de commandes. Un simple interrupteur ne répond que très rarement aux exigences et aux possibilités des différents moteurs électriques et donc aux applications envisagées. Alors passons en revue quelques principes de commandes par complexité croissante. Les servomoteurs de modélisme et les moteurs pas-à-pas ont été laissés de côté, car ils sont moins appropriés aux entraînements et constituent un autre thème.

Interrupteurs

Les moteurs de petite et moyenne puissance pour les entraînements sont souvent commutés en tout ou rien. Dans le cas le plus simple, on utilise un interrupteur mécanique, qui peut être remplacé par un relais ou un contacteur (relais de moyenne ou forte puissance utilisé en courant alternatif), ce qui rend le moteur commutable à distance. En outre, il existe une alternative électronique sous la forme de relais statiques à semi-conducteurs pour les petites et moyennes puissances, ainsi que des petits et des gros thyristors (**fig. 5**) qui permettent non seulement de commuter des moteurs allant jusqu'à de nombreux kilowatts, mais de les contrôler en vitesse de rotation et en puissance fournie. Mais les interrupteurs offrent encore d'autres possibilités, comme on va le voir.

Sens de rotation par interrupteurs

Une des solutions les plus simples pour inverser le sens de rotation d'un moteur à courant continu utilise un unique commutateur à double inverseur (**fig. 6**). On peut bien entendu le réaliser avec n'importe quel type d'interrupteur, donc aussi avec des semi-conducteurs. Mais attention : si l'on inverse le courant pendant que le moteur tourne, peuvent apparaître des courants élevés et de fortes pointes de tension ainsi que des dégâts sur les parties mécaniques ou les paliers, dus à d'énormes couples de rotation. Les conséquences sont alors pires qu'un simple court-circuit des bobines. Que quelque chose fonctionne dans son principe est loin de signifier que c'est bon

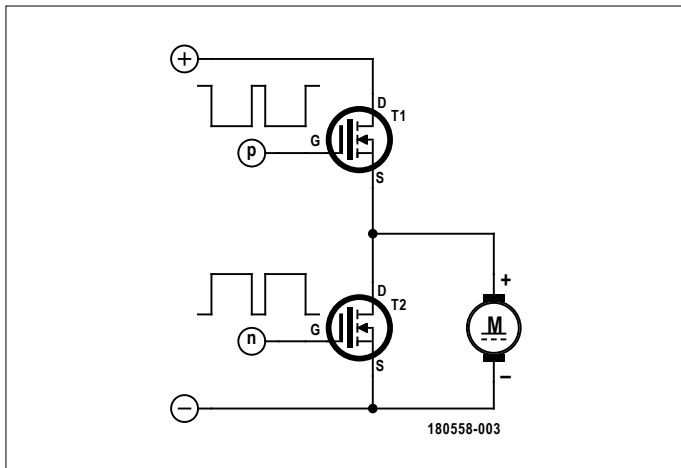


Figure 7. On règle la vitesse de rotation d'un moteur à courant continu au moyen d'un demi-pont de deux interrupteurs statiques pilotés par des signaux MLI en opposition de phase.

ou tout simplement utilisable. Sauf pour les tout petits moteurs, un tel commutateur devrait avoir une position centrale ou tierce dans laquelle le moteur serait d'abord freiné.

Il est bien entendu possible d'inverser le sens de rotation des moteurs monophasés qui possèdent deux enroulements séparés. Dans ce cas, l'inversion par commutateur de l'un d'entre eux suffit. Sur beaucoup de moteurs triphasés, il suffit de permuter deux phases. Notez que certains moteurs à courant alternatif ont un sens de rotation privilégié, leur collecteur étant décalé de quelques degrés par rapport à sa position théorique pour améliorer certaines caractéristiques techniques, dont le rendement.

Vitesse de rotation par MLI

Sur les moteurs à courant continu, outre le sens de rotation, il est possible de piloter la vitesse de rotation. La solution la plus simple, mais désastreuse du point de vue énergétique est une résistance série réglable, une espèce de potentiomètre de puissance. Mais son inutile dissipation d'énergie et d'autres inconvénients rendent une commande électronique « numérique » bien préférable. Il suffit d'un demi-pont avec deux semi-conducteurs à commutation rapide, typiquement des MOSFET à canal N, comme représentés sur la **figure 7**. Ces commutateurs sont commandés par des signaux MLI en opposition de phase, de sorte qu'ils ne conduisent jamais simultanément. La tension moyenne s'obtient par le produit « tension d'alimentation $\times$ rapport cyclique ». Par exemple, si l'on alimente le moteur de la figure 7 en 12 V, on obtient avec le signal MLI représenté et un rapport cyclique de 33% (signal « p ») une tension moyenne de 4 V aux bornes du moteur. Si la fréquence de commande est suffisamment élevée, l'inductance du moteur et son inertie mécanique filtrent efficacement les variations rectangulaires de la tension.

Si l'on se demande pourquoi ce procédé n'est pas simplement un contrôle de la puissance, mais effectivement un contrôle de vitesse, il suffit de réfléchir au principe de fonctionnement d'un moteur électrique (asynchrone). Même la vitesse à vide ne s'obtient pas simplement par combinaison de la puissance fournie et des pertes mécaniques. La force contre-électromotrice (FCEM), c'est-à-dire la tension induite par la rotation est

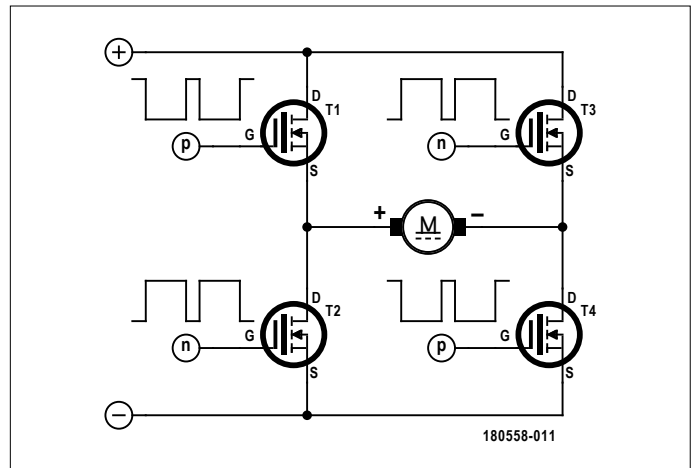


Figure 8. Avec un pont complet, pratiquement le double du circuit de la figure 7, on peut commander à la fois la vitesse et le sens de rotation au moyen de signaux MLI.

proportionnelle à la vitesse de rotation. À vide, elle n'augmente donc pas indéfiniment, mais s'établit à une valeur tout juste inférieure à la tension appliquée. Le quotient de cette petite différence par la résistance effective des enroulements donne le courant précis nécessaire pour produire le couple de compensation des pertes par frottement ou autre. Si l'on charge le moteur, la vitesse de rotation diminue, la différence entre la tension d'alimentation et la FCEM augmente, et donc le courant. Le courant est proportionnel au couple demandé et la puissance fournie est égale au produit du couple et de la vitesse de rotation.

Ces relations montrent aussi qu'il faut éviter de mettre brutalement un moteur qui tourne en court-circuit : sa résistance est très petite et la FCEM encore élevée ; le courant produit et le couple qui s'ensuit peuvent être passablement destructeurs. Ceci est également valable si au moyen d'un réglage de vitesse comme sur la figure 7, on réduit brusquement la vitesse de consigne. La nécessaire élimination de l'excès d'énergie cinétique impose des contraintes aux transistors et peut être la cause de pannes. À l'inverse, une accélération rapide au démarrage provoque des appels de courant qu'il faut prendre en considération.

Pont complet

Le sens et la vitesse de rotation peuvent être contrôlés au moyen d'un pont en H ou complet comme sur la **figure 8**. En principe, c'est juste un doublement du demi-pont de la figure 7. Le demi-pont de droite est habituellement aussi commandé par des signaux MLI en opposition de phase, avec toutefois une petite différence subtile : le signal MLI positif « p » est appliqué au commutateur T4 et le signal « n » à T3. Le moteur se trouve dans le bras horizontal du pont. Le moteur est à l'arrêt quand le rapport cyclique est précisément égal à 50%, car la différence entre les valeurs moyennes des deux demi-ponts est alors nulle. Si le rapport cyclique dépasse 50%, le moteur tourne dans un sens, s'il lui est inférieur, le moteur tourne dans l'autre sens.

En principe, une commande complète avec 2×2 signaux MLI pour attaquer les quatre transistors est un luxe. Il suffirait d'une combinaison plus simple où, comme sur la figure 7,

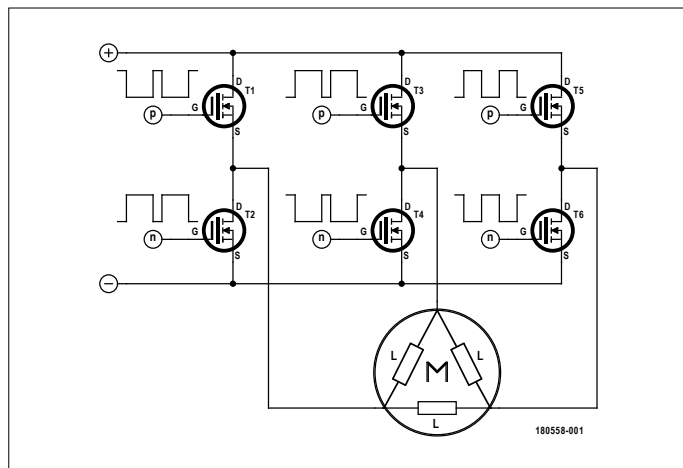


Figure 9. Il suffit de trois demi-ponts pour commander les moteurs BLDC en courant triphasé d'amplitude réglable.

seuls les transistors de gauche seraient commandés et où la moitié droite serait commandée de manière statique. En d'autres termes : la moitié gauche contrôlerait la vitesse de 0 à 100% et la moitié droite le sens de rotation. Dans la logique de commande, si l'on veut passer de 100% de marche avant à 100% de marche arrière, il faudrait commencer par prendre soin de ramener (lentement) le rapport cyclique du signal « p » de 100% à zéro avec T3 passant et T4 bloqué, et alors seulement bloquer T3 et ouvrir T4 tout en faisant passer le rapport cyclique du signal « p » à 100% d'un seul coup pour commencer avec 0 V aux bornes du moteur.

Commande de moteur à courant continu sans balais

Il est aussi possible de contrôler la vitesse et le sens de rotation de moteurs polyphasés par MLI. Un cas plus classique est le moteur à courant continu sans balais (*brushless direct current, BLDC*) utilisé dans nombre d'entraînements modernes (du vélo à la voiture électriques en passant par le gyropode et bien d'autres applications). Dans la version simple, il comporte trois phases, avec toutefois seulement trois bornes. Contrairement aux modèles industriels où toutes les extrémités des enroulements sont accessibles séparément, on n'a pas le choix entre les couplages triangle ou étoile, ce qui n'est d'ailleurs pas nécessaire dans le cas d'une commande entièrement électronique où les forts courants de démarrage sont limités par MLI. L'équipement supplémentaire nécessaire reste très limité : par rapport à la figure 8, la **figure 9** ne présente qu'un demi-pont de plus. Le diable se cache dans la logique de commande : comme le champ tournant nécessaire doit être produit par MLI et qu'un moteur BLDC est dans son principe un moteur synchrone, il faut commuter les tensions sur les trois bornes du moteur en fonction de la position du rotor et de telle sorte que les signaux MLI des trois demi-ponts soient toujours déphasés de 120°. Cette commutation électronique nécessite la connaissance précise de la position angulaire instantanée du rotor. Il y a deux méthodes d'acquisition de cette information : soit équiper le moteur de capteurs à effet Hall (au moins trois), soit exploiter la FCEM pendant les pauses de commutation des demi-ponts. Cette information permet à la commande de produire des signaux MLI assurant une rotation correcte du

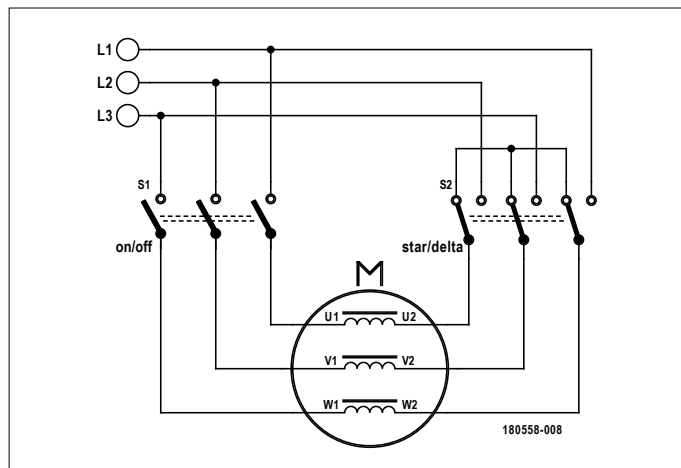


Figure 10. Pour limiter les courants de démarrage des moteurs synchrones de moyenne puissance, on les démarre en couplage étoile ; ensuite on commute sur le couplage en triangle quand une vitesse suffisante est atteinte. Ici la variante avec deux contacteurs.

champ et donc du moteur. Le contrôle par capteurs ou FCEM confère au moteur un comportement asynchrone.

Un tel moteur BLDC peut être alimenté par des tensions alternatives triphasées simulées par MLI et qui, dans le plus simple des cas ont une forme rectangulaire. Les tensions sinusoïdales par MLI ne causent pas non plus de problème. Les tensions alternatives trapézoïdales permettent d'atteindre des rendements élevés en même temps qu'une faible ondulation du couple moteur.

Commutation triangle-étoile

Pour être complet : les moteurs triphasés courants de moyenne et forte puissances ont souvent des courants de démarrage élevés. Pour réduire ces courants, on peut commencer par utiliser des résistances en série et ne connecter directement le moteur au réseau qu'une fois atteinte une vitesse suffisante. Cette méthode est effectivement utilisée dans l'industrie – il existe des résistances série à l'eau réglables (leur nom indique leur matériau constitutif). Pour des puissances inférieures à 10 kW, il suffit d'exploiter le fait qu'en couplage étoile, le moteur ne consomme que le tiers environ de la puissance en couplage triangle. On le fait donc démarrer en étoile et on le commute en triangle quand on dépasse 75% de la vitesse nominale.

La **figure 10** en montre une réalisation avec des contacteurs. L'inverseur triple S2 doit être obligatoirement du type ouverture avant fermeture sous peine d'explosion. Un autre contacteur avec deux inverseurs pourrait par permutation (correcte) de deux connexions (par ex. U1 et U2) inverser le sens de rotation du moteur.

En principe, ce genre de commutateur étoile-triangle est aujourd'hui largement obsolète dans l'industrie, car il est possible de démarrer les moteurs en douceur au moyen d'onduleurs à fréquence variable.

L'électronique

Il a déjà été mentionné que dans les installations de forte puissance, les commandes de moteurs utilisent surtout des MOSFET comme semi-conducteurs de puissance. Il est alors intéressant de voir comment est conçue la commande d'un demi-pont par des signaux MLI en opposition de phase. Tout



Figure 11. Un choix au hasard parmi les petites cartes à contrôleur pour moteurs de faible puissance, proposées sur eBay.

d'abord, il est recommandé de choisir une fréquence de base du signal MLI suffisamment élevée pour se situer au-delà de l'audible humain et animal, pour que le moteur n'ait pas tendance à se comporter en haut-parleur. D'un autre côté, les pertes par commutation augmentent avec la fréquence. Le compromis se trouve dans la zone de 30 à 60 kHz.

Comme dans un demi-pont les deux transistors ne doivent jamais conduire simultanément, il doit y avoir un temps mort entre le blocage de l'un et la mise en conduction de l'autre. Ce temps mort dépend des caractéristiques des transistors et de la fréquence du signal MLI. Les valeurs courantes vont de 0,1 à 2 μ s. Pour une fréquence de MLI de par ex. 50 kHz, un temps mort de 0,5 μ s signifierait que les 2,5% supérieurs de la puissance maximale sont inaccessibles, ce qui est habituellement acceptable.

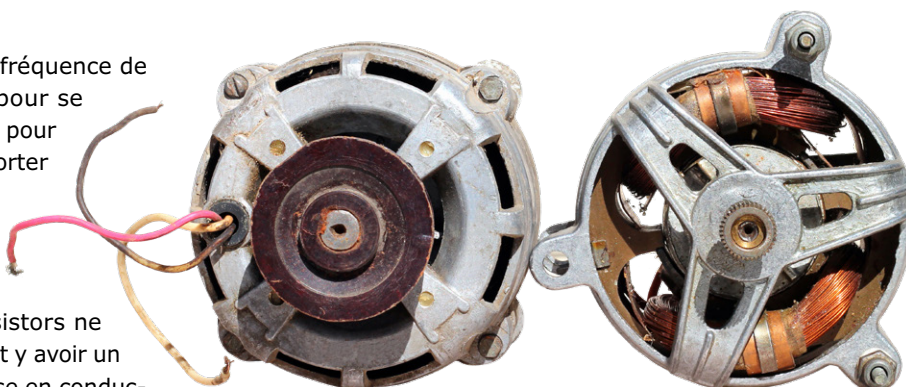
Les signaux MLI en opposition de phase adéquats, y compris les temps morts appropriés peuvent être produits par un microcontrôleur, quasiment « à la main » si on le désire. Si le nombre de broches est suffisant, on peut même produire par logiciel les six signaux MLI pour trois demi-ponts, on a ainsi un contrôle complet. Mais cette souplesse a aussi des inconvénients. Si un bogue plante le microcontrôleur, il y a de fortes chances non seulement que l'électronique prenne congé bruyamment, mais qu'il y ait de lourds dégâts matériels, voire corporels. Il est donc recommandé d'employer au moins des pilotes spéciaux pour demi-ponts à MOSFET. Ces circuits contiennent tout le nécessaire pour le pilotage d'un couple de MOSFET à canal N, y compris la génération des signaux de commande en opposition de phase incluant le temps mort (réglable) et des raffinements comme le circuit élévateur de tension (*bootstrap*) pour la commande des grilles des FET du côté supérieur (T1, T3 et T5 sur la figure 9). À l'entrée, un seul signal MLI monophasé ordinaire suffit. Certains pilotes gèrent jusqu'à trois demi-ponts.

Les signaux de commande de ces puces peuvent être pris en charge par un petit microcontrôleur qui a ainsi moins de choses à faire. Pour des raisons de sécurité, il est préférable d'éviter

de confier à ce genre de microcontrôleurs, même de puissance de calcul suffisante, des tâches qui vont au-delà de la commande du moteur proprement dite avec les déclenchements de sécurité, le démarrage en douceur, etc. Ou alors, en utilisant des puces spécialisées dans la commande de moteurs, on privilégie la sécurité aux dépens de la souplesse. Pour des petits moteurs de quelques dizaines de watts, il existe même des solutions complètes qui intègrent la commande, les pilotes et les transistors de puissance. Il n'y a plus qu'à suivre la feuille de caractéristiques. ;-)

Si l'on veut expérimenter avec des solutions complètes, on peut aller sur eBay et faire une recherche avec les mots clés « motor controller » et l'on trouvera des milliers de cartes et d'équipements les plus variés à des prix allant d'un à plusieurs milliers d'euros, depuis la conduite de moteurs monophasés de petite puissance (**fig. 11**) jusqu'aux applications industrielles (presque tout est d'origine chinoise). ◀

(180558-04 – version française : Helmut Müller)



@ WWW.ELEKTOR.FR

→ **PiCar-V pour Raspberry Pi de SunFounder**
www.elektor.fr/sunfounder-picar-v-for-raspberry-pi

→ **MotoPi de JOY-iT** (carte de commande de moteur pour Raspberry Pi)
www.elektor.fr/joy-it-motopi-motor-control-for-raspberry-pi

→ **Motorino de JOY-iT** (carte de commande de moteur pour Arduino)
www.elektor.fr/joy-it-motorino-motor-control-for-arduino

Liens

- [1] Guerre des courants : https://fr.wikipedia.org/wiki/Guerre_des_courants
- [2] Moteur Jedlik : www.jedliktarsasag.hu
- [3] Lohner-Porsche : <https://tinyurl.com/y8kx2ldr>
- [4] Dictionnaire, image de gauche : <https://tinyurl.com/y7mn24fz>
- [5] Dictionnaire, image de droite : <https://tinyurl.com/yc2pyp6k>
- [6] Principe du moteur : <https://tinyurl.com/yb98ug89>
- [7] Thyristor : <https://tinyurl.com/ychbckro>

diviseur de fréquence avec facteur de division entier réglable pour trois fois rien (de CMOS)

Michael A. Shustov (Russie) et Andrey M. Shustov (Allemagne)

La conception d'un diviseur de fréquence avec un facteur de division réglable est souvent considérée comme un exercice complexe. Pour relever ce « défi », nous avons créé un circuit qui, bien que rudimentaire, permet pourtant de diviser la fréquence d'un signal numérique par un facteur n .

Le circuit illustré à la **figure 1** se compose essentiellement d'une porte CMOS XOR de type 4070B (IC1.A) et d'un tampon non-inverseur de type 4050B (IC2.A), avec un réseau R-C réglable entre les deux, et une contre-réaction. Le diviseur (facteur de division ; facteur d'échelle) n est une valeur entière comprise entre 2 et 1.000, voire plus.

Le facteur de division souhaité est réglable avec la résistance ajustable P1 (préréglage ; potentiomètre). Avec P1 effectivement à 0Ω , le diviseur n vaut 2. Régler le potentiomètre sur $10 \text{ k}\Omega$ donne $n = 94$. Si on augmentait la résistance à $75 \text{ k}\Omega$, cela donnerait $n = 1.000$. Dans la plage $9 < n < 30$, la relation entre la résistance effectivement réglée sur P1 et le facteur de division résultant est presque linéaire et s'exprime par une formule :

$$n = \text{ROUND} (P1 + 0,2) / 0,133$$

où P1 est en kilo-ohms ($\text{k}\Omega$), et où ROUND indique une opération d'arrondi à la valeur entière la plus proche.

La **figure 2** montre les signaux d'entrée et de sortie du diviseur obtenus avec les paramètres suivants :

- VDD (tension d'alimentation de IC1, IC2) = 10 V
- Amplitude de f_{in} en entrée : 10 V
- P1 réglé sur $2,4 \text{ k}\Omega$

Pour obtenir d'autres facteurs de division, on peut connecter plusieurs diviseurs identiques en série (« mise en cascade »). Chaque diviseur de fréquence connecté doit être réalisé avec ses propres porte et tampon contenus respectivement dans les circuits intégrés CD4070BP et CD4050BP. Notez qu'il faut utiliser des puces CMOS tamponnées (suffixe B). Il ne faut pas non plus laisser l'entrée f_0 flottante ni aucune des entrées inutilisées des circuits intégrés CMOS du circuit. ◀

(180559-04 | 160514 – version française : Denis Lafourcade)

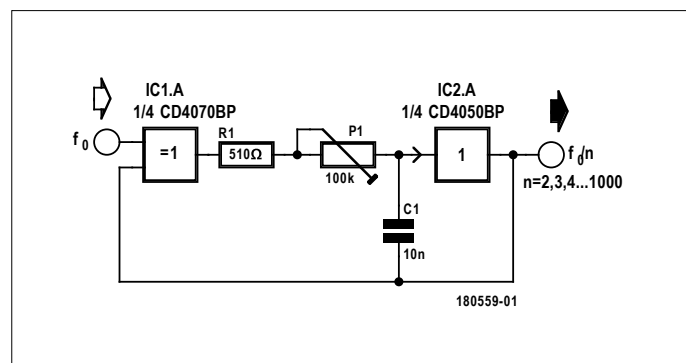


Figure 1. Schéma du diviseur de fréquence réglable. L'alimentation VDD des circuits intégrés n'est pas indiquée et est supposée être de 10 V_{CC} .

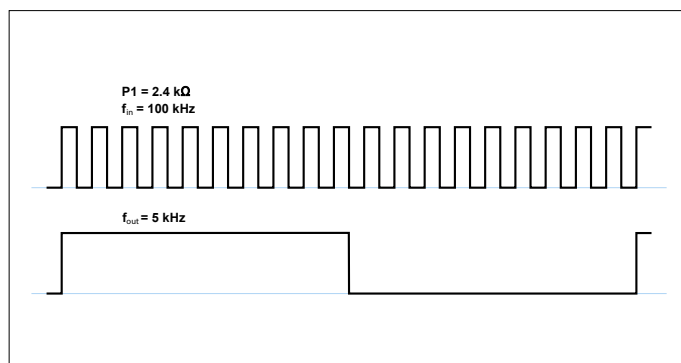


Figure 2. Exemple de signaux d'entrée et de sortie du diviseur de fréquence configuré pour un diviseur $n = 20$.

poste de soudage de CMS compact

**à régulation de température
précise et rapide**

Sunil Malekar (labo d'Elektor, Inde)

Le succès du poste de soudage de CMS à Platino publié en 2015 avait largement dépassé le cadre du lectorat d'Elektor. Imprégnés comme nous le sommes de l'esprit d'ingénieur, un projet qui fonctionnait aussitôt assemblé ne pouvait susciter chez nous que l'envie de l'améliorer.

D'où ce nouveau poste de soudage, entièrement repensé mais toujours aussi compact.



Le poste de soudage de CMS à construire soi-même de juillet/août 2015 [1] utilisait les têtes de soudage de la série RT de Weller et leur bien pratique jack stéréo de 3,5 mm. Ce nouveau projet repose également sur les têtes avec élément de chauffe et capteur de température intégrés. Il reste à fournir assez d'énergie pour chauffer la pointe et à réguler avec précision sa température.

Élémentaire mon cher Weller

Même si, et on peut le comprendre, Weller ne dévoile rien du mécanisme interne de sa pointe RT1 (**fig. 1**), celle-ci contient à n'en pas douter un dispositif résistif qui chauffe lorsque qu'un courant le traverse, la chaleur dégagée étant proportionnelle à l'intensité. De notre côté, il nous suffirait donc de réguler ce courant pour atteindre et maintenir constante une température donnée. Comment faire ? Rien de plus simple avec le capteur de température intégré : on envoie une certaine quantité de courant, puis on mesure la température ; si elle est plus élevée que la valeur de consigne, on réduit le courant ; si elle est plus basse, on envoie plus de courant.

De nos jours, à une époque où le fonctionnement du moindre bidule (et de certains humains) repose sur des 0 et des 1, on implante généralement une telle boucle de régulation au moyen d'un microcontrôleur et d'un signal MLI (modulation de largeur d'impulsion). Ici la valeur moyenne du signal MLI détermine l'intensité moyenne du courant qui traverse l'élément de chauffe (**fig. 2**).

Votez MLI

La commande par MLI est intéressante pour plusieurs raisons. D'abord parce que s'il y a bien une chose que les microcontrôleurs font mieux que les autres (à part se détraquer aux moments les plus inopportuns), c'est produire des signaux MLI, au besoin sur plusieurs voies et avec une précision de 16 bits ou plus. Ensuite parce que les signaux MLI commutent rapidement les transistors (de puissance) entre les états conducteur et bloqué, sans traîner dans la région intermédiaire énergivore (où aiment à opérer les amplificateurs analogiques). Grâce à la MLI, les transistors de puissance

peuvent donc commuter des courants forts sans surchauffer, et même des courants encore plus intenses dans le cas des MOSFET à résistance ON très faible. Ce sont ces deux propriétés qui nous ont permis de concevoir un régulateur de température simple.

Problèmes de voisinage

Relativisons toutefois le qualificatif « simple » pour ce qui est de la mesure précise de la température de la pointe. Car contrairement à ce que recommandent les bonnes pratiques de conception en matière de mesure de précision, les fils du capteur qui acheminent le signal des petites variations de température sont placés très près des fils de l'élément chauffant, qui eux délivrent une tension élevée. Pire, le capteur et cet élément ont un fil commun. Même si à n'en pas douter cette conception se justifie, elle nous complique la vie puisqu'elle nous oblige à appliquer quelque filtrage au signal de sortie du capteur, et à soigneusement planifier l'instant de sa lecture.

INFOS SUR LE PROJET

capteurs

mesure

soudage

débutant

→ connaisseur

expert

env. 3 h

outils standard
(y compris pour le soudage)

env. 60 €



Figure 1. Chacun des nombreux modèles de la série RT répond à un besoin particulier, comme ici la pointe RT1 et son aiguille à souder de 0,2 mm.

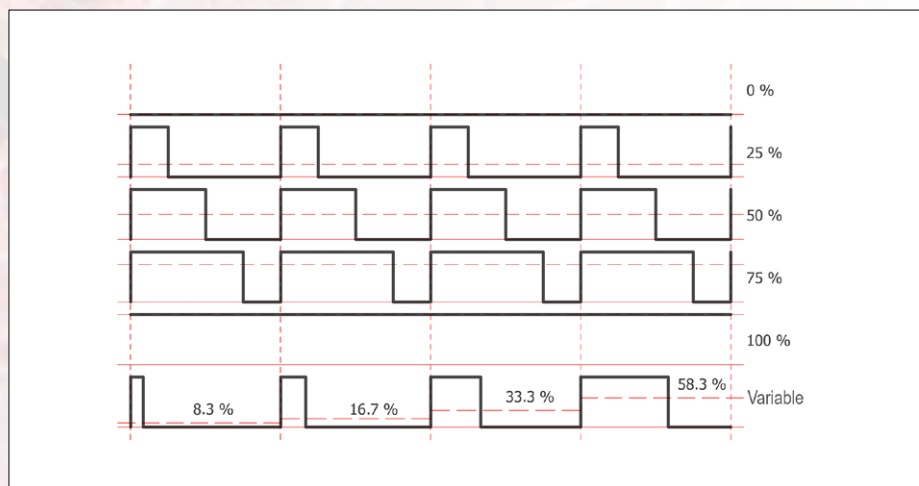


Figure 2. Le lien entre un signal MLI produit numériquement et les valeurs analogiques moyennes.

Caractéristiques

- Réglage de température entre +50 °C et +450 °C
- Tête de soudage interchangeable de la série RT
- Réglage précis de la température par codeur rotatif
- Enregistrement de la température de consigne
- Alimentation externe de 12 V CC / 2 A

Tout le monde à l'abri

Ce qui a été dit ci-dessus valait également pour le poste de soudage à Platino [1]. Incidemment, si nous utilisons ici aussi un codeur rotatif, l'afficheur

est cette fois-ci à OLED. Ce qu'il y a de nouveau dans ce projet, c'est que nous mesurons le courant de sortie pour contrôler la puissance thermique maximale que la tête de soudage est autorisée à produire. Cette limitation protège la pointe des courants potentiellement dangereux, et met les petits composants à l'abri des apports de chaleur excessifs.

Le circuit

Les explications précédentes vont nous aider à comprendre le circuit (**fig. 3**). Pour changer, entrons par la sortie, le connecteur K3, là où se raccorde la pointe RT1. La broche 1 délivre la tension de chauffage, la broche 2 le signal du capteur de température, et la broche 3 est la masse (GND).

Commandé par le signal MLI, le MOSFET T1 connecte périodiquement l'élément chauffant à l'alimentation de chauffage V_{IN} (typiquement 12 V). La grille d'un MOSFET tel que T1 se comporte presque comme un condensateur, d'où la nécessaire présence d'un étage push-pull (T2/T4) pour assurer des flancs raides aux impulsions de commutation, et ainsi une mise en conduction et un blocage du FET aussi rapide que possible. Et ceci simplement en chargeant et déchargeant la capacité de grille, sans se compliquer outre mesure pour optimiser l'ensemble. T2 et T4 sont attaqués par T3 car le microcontrôleur est incapable de commuter les 12 V de V_{IN} . Le courant de chauffe produit une petite différence de potentiel aux bornes de

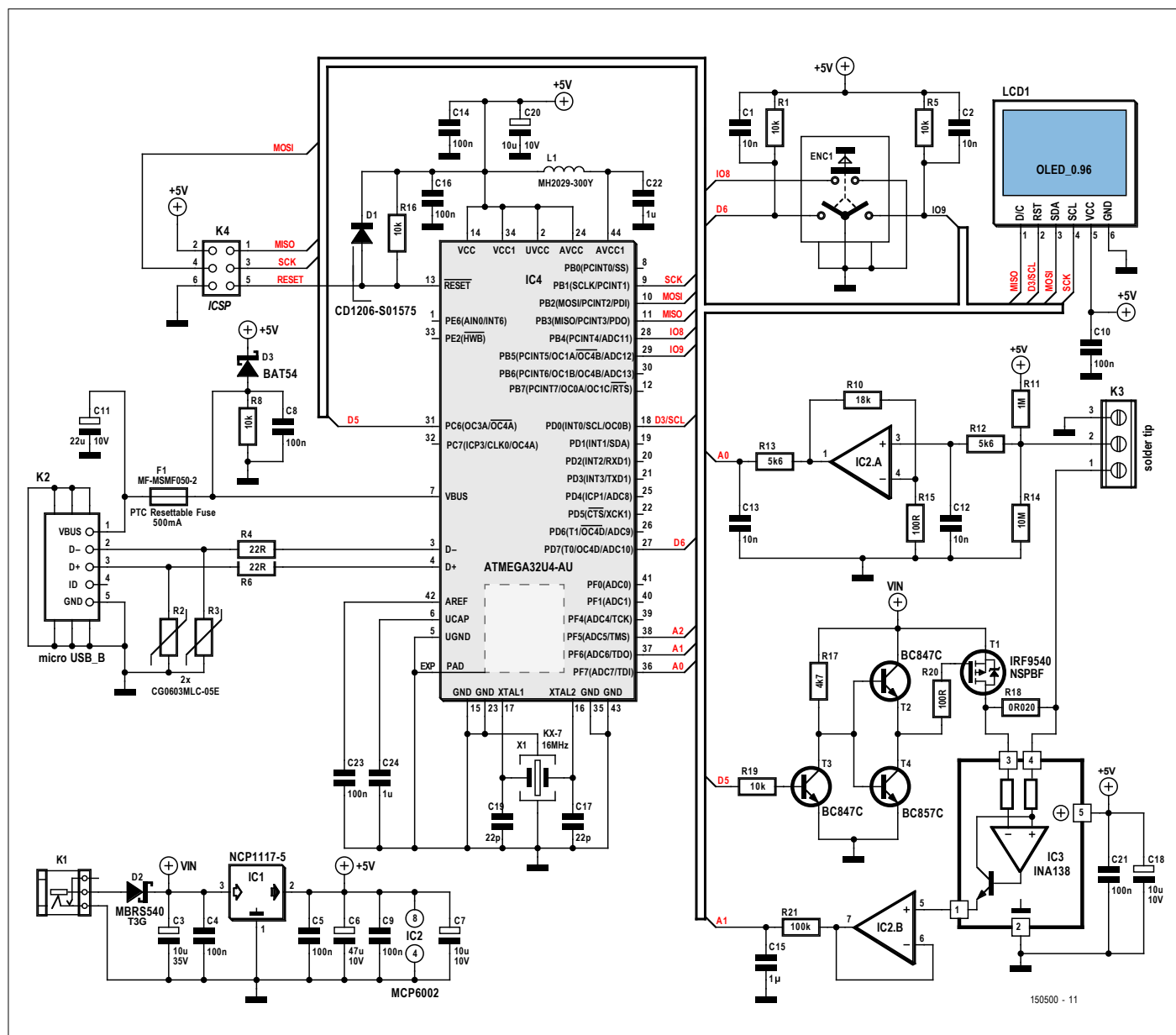


Figure 3. Le schéma du poste de soudage combine de façon ingénieuse électronique analogique et numérique.



LISTE DES COMPOSANTS

Résistances

Défaut : CMS 0805, 100 mW, 1 %
 R1,R5,R7,R8,R16,R19 = 10 kΩ
 R2,R3 = CG0603MLC-05E, anti-décharge électrostatique, compatible USB, CMS 0603
 R4,R6 = 22 Ω
 R9 = 2,7 kΩ
 R10 = 18 kΩ
 R11 = 1 MΩ
 R12,R13 = 5,6 kΩ
 R14 = 10 MΩ
 R15,R20 = 100 Ω
 R17 = 4,7 kΩ
 R18 = 0,02 Ω, 2 W, CMS, résistance ampèremétrique
 R21 = 100 kΩ
 R22 = 560 kΩ

Condensateurs

Tous CMS 0805
 C1,C2,C12,C13 = 10 nF
 C3 = 10 µF 35 V, boîtier C
 C4,C5,C8,C9,C10,C14,C16, C21,C23 = 100 nF
 C6 = 47 µF 10 V, boîtier A
 C7,C18,C20 = 10 µF 10 V, boîtier A
 C11 = 22 µF 10 V, boîtier A
 C15,C22,C24 = 1 µF
 C17,C19 = 22 pF

Inductance

L1 = ferrite, 3 A, 30 Ω @100 MHz, 0805

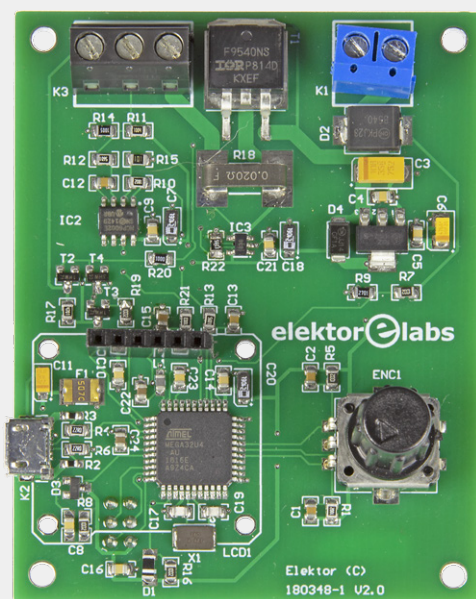
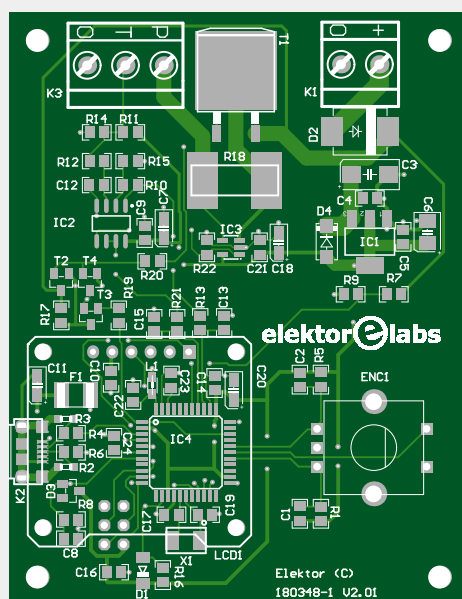
Semi-conducteurs

D1 = CD1206-S01575, CMS 1206
 D2 = MBR540T3G
 D3 = BAT54
 D4 = S1A
 T1 = IRF9540NSPBF
 T2,T3 = BC847
 T4 = BC857
 IC1 = NCP117ST50T3G, SOT-223-3
 IC2 = MCP6002-E/SN
 IC3 = INA138NA/250G4
 IC4 = ATmega32U4-AU

Divers

ENC1 = codeur rotatif avec bouton, Bourns, type PEC11R-4225F-N0024
 F1 = fusible CTP, 15 V, I_{maintien} = 0,5 A, 4,5 × 3,2 mm

K1 = bornier à 2 vis, au pas de 5,08 mm
 K2 = connecteur micro-USB de type B
 K3 = bornier à 3 vis, au pas de 5,08 mm
 K4 = barrette mâle à 6 broches (2×3), au pas de 2,54 mm
 LCD1 = afficheur OLED, 0,96", 128×64, interface SPI
 X1 = quartz de 16 MHz
 Barrette mâle à 6 broches, 1 rangée pour LCD1, au pas de 2,54 mm
 Barrette femelle à 6 contacts pour LCD1, au pas de 2,54 mm
 Boîtier (proposé) : type 102 de Teko, 110×70×46 mm
 Circuit imprimé, réf. 180348-1, www.elektor.fr



R18. Cette tension est copiée et amplifiée par le circuit qui entoure IC3.B, IC2.B et R21, ce qui permet au convertisseur analogique-numérique (CA/N) du microcontrôleur de l'échantillonner via la ligne A1. Le condensateur C15 filtre le bruit. Notez que la version actuelle du micrologiciel de l'ATmega limite l'intensité à 1,5 A.

L'amplificateur à faible bruit IC2.A accroît la tension de sortie du capteur de température. Les réseaux R12-C12 et R13-C13 forment un filtre passe-bas du second ordre chargé d'éliminer le bruit « inhérent à la ligne ». Le signal de la ligne A0 est échantillonné par le contrôleur IC4. Les résistances R11 et R14 permettent de détecter que le capteur est déconnecté, donc de couper l'alimentation de

la pointe. On évite ainsi de la détruire par surchauffe.

Circuit numérique

LCD1 est un afficheur OLED de 0,96" dont le câblage vous étonne peut-être. Rien de surprenant à cela, ce type d'écran peut fonctionner en mode SPI ou I²C selon le placement de certains de ses cavaliers et résistances. C'est le mode SPI qui est utilisé ici, or le marquage des afficheurs OLED fait souvent référence aux signaux I²C, ce qui explique le schéma. Si vous achetez ce type d'afficheur, veillez à ce qu'il s'agisse bien de la version SPI. ENC1 est le codeur rotatif servant à régler la température de consigne. Il nécessite trois entrées numériques car il est doté d'un bouton.

IC4, le microcontrôleur, fournit la puissance de calcul nécessaire ainsi que la circuiterie de production de la MLI. Il s'agit d'un ATmega32Ua à interface USB intégrée. La connexion à cette interface se fait via le connecteur USB K2 monté à gauche du contrôleur. R2 et R3 protègent ses lignes contre les décharges électrostatiques, et F1 est un fusible réarmable. Ce port USB est en fin de compte robuste, vous pouvez sans grand risque y relier un ordinateur. Le poste de soudage est tout aussi sûr et, comme nous le verrons plus loin, l'EDI Arduino le reconnaît comme carte Arduino Micro ou Leonardo.

Alimentation

Le bon fonctionnement du poste nécessite une alimentation stabilisée de 12 V

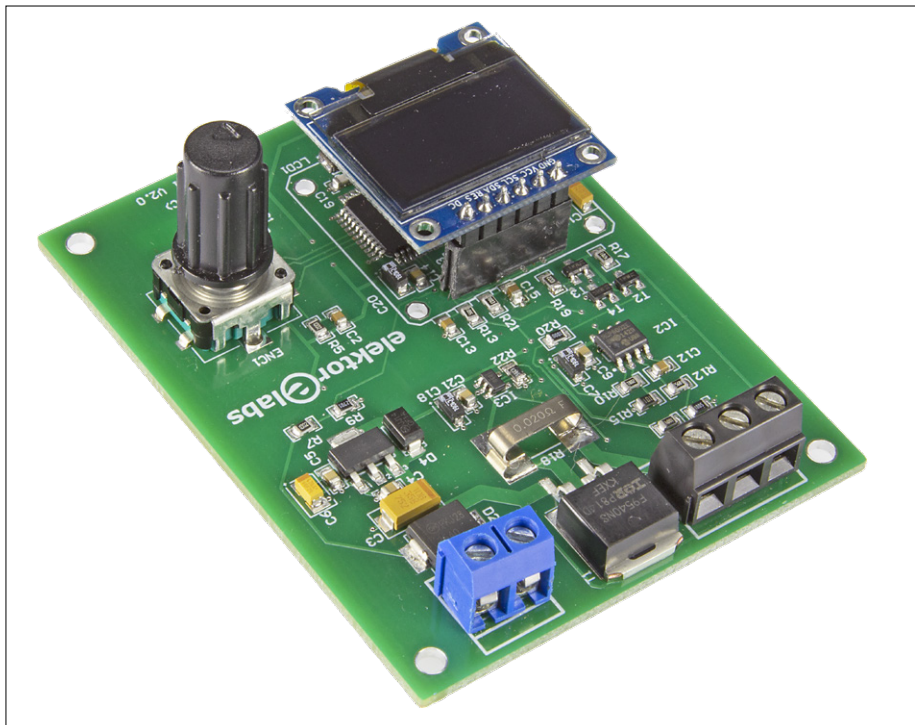


Figure 4. Face cuivre d'un prototype de la carte avec l'afficheur et le codeur rotatif assemblés.

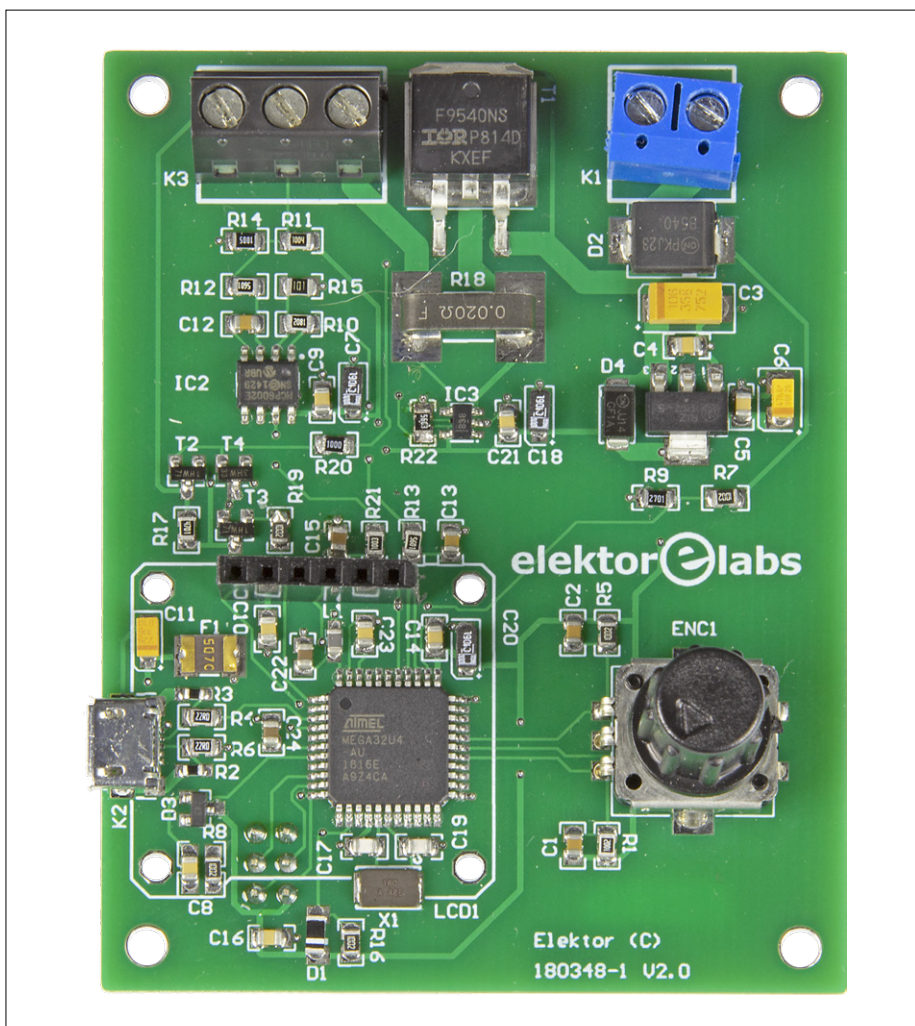


Figure 5. Face composants de la carte avec tous les CMS montés.

et 2 A. Si vous êtes tenté par un bloc secteur de 12 V pour son côté pratique et bon marché, vérifiez la régulation de sa tension de sortie et l'intensité de son courant de sortie avant de l'utiliser, car la précision de la mesure pâtira de toute faiblesse ou instabilité (bruit) de la tension délivrée. Le régulateur de tension IC1 alimente en 5 V la partie numérique du circuit ainsi que la circuiterie d'entrée du CA/N du capteur de température. L1 et C22 sont utilisés en renfort pour protéger ce CA/N du bruit.

Vous pouvez aussi alimenter le circuit par le port USB pour le modifier ou l'étudier. Lorsque V_{IN} n'est pas appliquée, D3 connecte V_{USB} au rail de 5 V, alimentant de fait le circuit. La pointe n'est dans ce cas pas chauffée, mais vous pouvez encore mesurer sa température, autrement dit elle peut aussi servir de thermomètre !

Logiciel

Le programme coordonne l'ensemble et intervient à plusieurs stades :

- production de la MLI
- mesure de la température
- mesure du courant de chauffage
- mesure de la tension de chauffage
- communication USB
- interaction avec l'utilisateur.

Le code déclenche une chauffe très rapide de la pointe dès la mise sous tension. De fait il ne faut que quelques secondes pour que la température atteigne sa valeur de consigne et que la régulation MLI prenne aussitôt le relais. Notez que cette consigne est stockée dans l'EEPROM de l'ATmega, vous n'avez pas à l'entrer à chaque session.

La température de la pointe est mesurée périodiquement. Le signal MLI est d'abord arrêté (mis au niveau Bas) puis, après un bref instant (qui permet au changement de niveau d'être effectif), le signal sur A1 est échantillonné. La température actuelle est calculée et comparée à la consigne. La valeur du rapport cyclique est ajustée en conséquence si l'écart est trop grand (d'environ 5 °C). La mesure du courant de chauffe se fait quant à elle lorsque le signal MLI est actif (au niveau Haut), sinon l'intensité mesurée serait toujours nulle.

L'interface utilisateur comprend le codeur rotatif et l'afficheur. Le codeur utilise des interruptions produites par changement d'état de broche, l'afficheur la biblio-

thèque graphique d'Adafruit.

La consigne, les températures mesurées ainsi que l'état de la station sont transmis par le port série USB.

Programmation du micrologiciel

Le code de l'ATmega [2] est un croquis Arduino et nécessite donc l'EDI Arduino pour être compilé. Comme la carte du poste est compatible avec les cartes Micro (et Leonardo), c'est ce modèle qu'il faut sélectionner depuis le menu *Gestionnaire de cartes* de l'EDI.

K4 sert à programmer l'ATmega, ce que vous pouvez aussi faire par USB depuis l'EDI Arduino une fois le chargeur de démarrage Micro ou Leonardo gravé.

Assemblage

L'e-choppe vous propose un kit partiel, mais vous pouvez aussi réunir vous-même les composants, vous procurer le circuit imprimé nu auprès d'Elektor et prendre un après-midi pour souder les CMS.

Aidez-vous des **figures 4 et 5** si vous réalisez le projet. La **figure 6** montre un composant particulier : la résistance ampèremétrique de $0,02\ \Omega$.

Le « module » est au final relativement petit et tiendra assurément dans toutes sortes de boîtiers, laissant même, histoire de lui donner un peu de poids, de la place à un bloc secteur de 12 V / 2 A ou à un transformateur doté d'un redresseur, d'un stabilisateur, etc. Notre prototype de la station de soudage a été installé dans un boîtier Teko de type 102 (110×70×46 mm).

Le bon poids sera celui qui empêche le poste de suivre le moindre de vos mouvements comme le ferait une savonnette en laisse. Vous pourriez donc opter pour un boîtier métallique massif et, si cela ne suffit pas à le stabiliser, y ajouter l'alimentation – un transformateur massif. Si vous choisissez de construire vous-même une alimentation séparée de 12 V et 2 A, prenez toutes les précautions d'usage en matière de sécurité électrique. Pour réduire au minimum les risques, nous vous conseillons d'ailleurs de plutôt récupérer une alimentation scellée de 12 V CC d'imprimante ou d'éclairage.

Il vous faudra aussi dénicher un câble :

- souple
- qui résiste aux écoulements et à la chaleur ;
- prévu pour le courant nominal ;

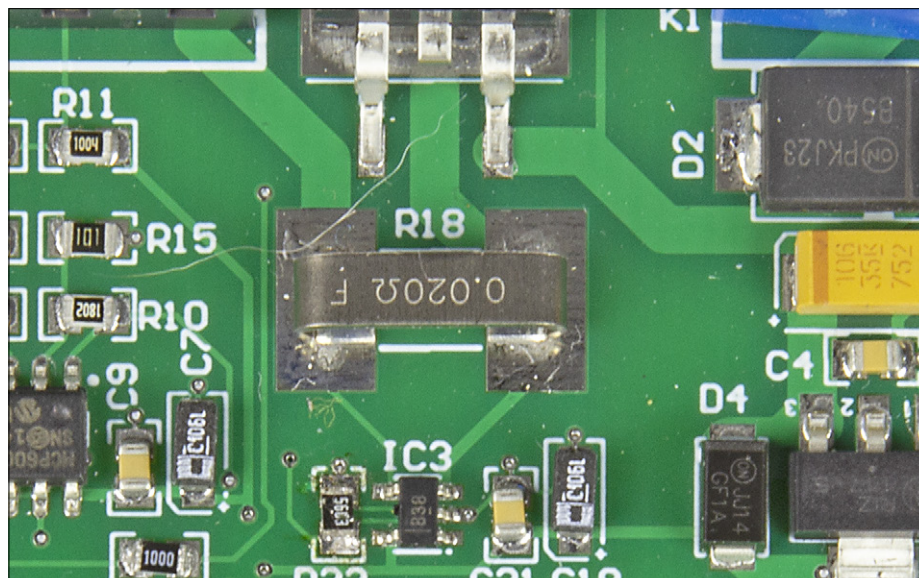


Figure 6. Vue rapprochée du prototype portant la résistance ampèremétrique de $0,02\ \Omega$ et 2 W.

- qui puisse être soudé aux prises jack.

Nous avons trouvé des câbles USB tout à fait appropriés - il suffit de couper les connecteurs et d'utiliser deux fils pour la mise à la terre et l'alimentation.

Pour le boîtier, nous vous suggérons d'y monter une prise reliée au module. Vous connecterez alors la tête de soudage au câble, et le câble à la prise.

En d'autres mots, construisez votre station de soudage comme le sont celles du commerce : si elles se ressemblent toutes, c'est pour de bonnes raisons.

Utilisation

Le mode d'emploi tient sur une ligne : mettez le poste sous tension et réglez la température de consigne quelque part entre $50\ ^\circ\text{C}$ et $450\ ^\circ\text{C}$ en tournant le codeur rotatif. Rien de plus. L'écran indique la valeur de consigne ainsi que la température de la pointe.

La station se met en veille si le codeur n'a pas été tourné après environ quinze minutes. Appuyez sur le bouton du codeur pour la réveiller. Si vous pressez le codeur pendant dix secondes, la station se mettra en veille. ◀

(180348-03 | 150500

version française : Hervé Moreau)



@ **WWW.ELEKTOR.FR**

→ Poste de soudage régulé à construire soi-même, circuit imprimé nu :
www.elektor.fr/diy-solder-station-bare-pcb

→ Poste de soudage régulé à construire soi-même, kit de composants :
www.elektor.fr/diy-solder-station-kit-of-parts

→ Poste de soudage régulé à construire soi-même, microcontrôleur programmé :
www.elektor.fr/diy-solder-station-controller

Liens

[1] Article initial (poste de soudage de CMS avec Platino) :
www.elektormagazine.fr/140107

[2] Code de cet article : www.elektormagazine.fr/180348-03

electronica ils inventent

Clemens Valens (labo d'Elektor)

La seconde édition du concours **electronica Fast Forward** – le tremplin des jeunes pousses organisé par Elektor en collaboration avec le salon **electronica** de Munich – restera dans les mémoires : tous les projets présentés portaient en effet la signature de professionnels aux compétences techniques remarquables.

Trente-deux candidats provenant de pas moins de dix pays étaient en lice, chacun disposant de cinq minutes pour présenter son projet à un jury composé d'experts des sociétés Arrow, Avnet-Silica, EBV Elektronik et, bien sûr, Elektor.

Après trois journées d'auditions, six finalistes furent retenus : Teiimo, Querom, V-Juice, ValCUN, Volabo, et Wizama. Le lendemain, le jury attribua la qualification « wildcard » à Wisebatt, portant ainsi à sept le nombre des finalistes. Tous furent entendus une nouvelle fois, et c'est finalement la start-up Teiimo – spécialisée dans l'électronique vestimentaire et les textiles intelligents – qui fut désignée vainqueur à l'unanimité.

Grâce au parrainage des sociétés Arrow, Avnet-Silica & EBV Elektronik (catégorie *Diamond*) et Trinamic (catégorie *Bronze*), des candidats venus de toute l'Europe, mais aussi d'Australie et d'Amérique, ont pu offrir au jury et au public un aperçu intéressant, parfois même saisissant, de ce que pourrait nous réserver l'avenir. Que de variétés dans les projets : imprimante 3D métal, console pour jeux de société, extensomètre à couche mince,



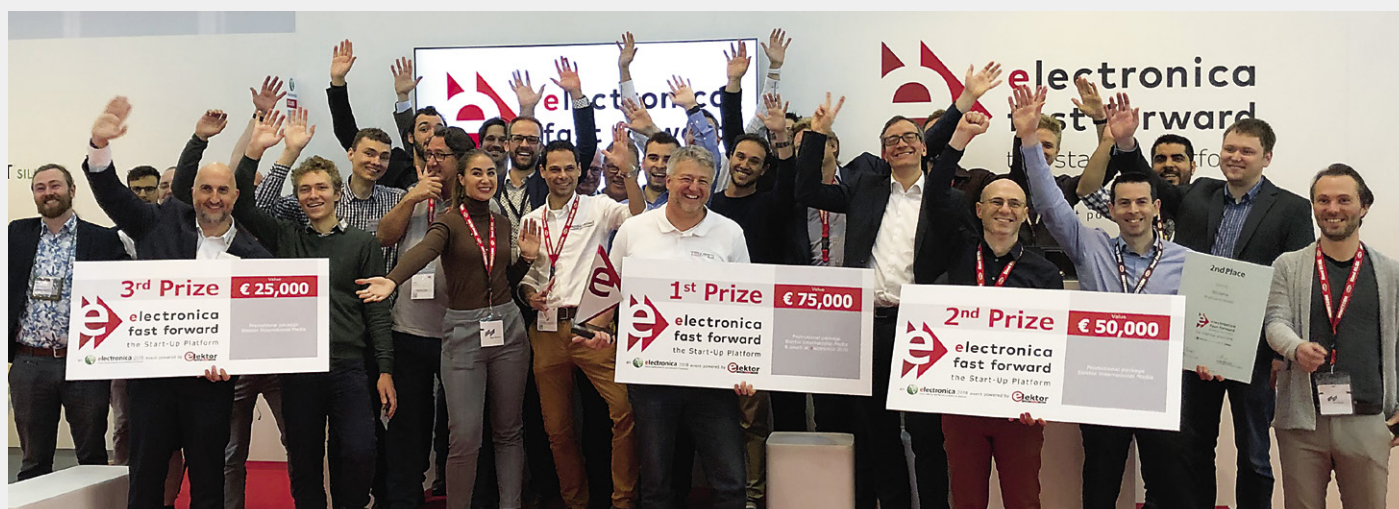
Markus Strecker, de la société **Teiimo**, reçoit le 1^{er} prix des mains de Martin Lechner, le directeur du salon **electronica**. Teiimo a été récompensée pour son système textile de mesure et de transmission de paramètres biologiques vitaux.



Le 2^e prix est revenu à la société **Wizama** pour son plateau de jeu à écran tactile et objets connectés. À la croisée du jeu de société et du jeu vidéo, la console de Wizama offre une nouvelle façon de jouer en famille ou entre amis.

Fast-Forward 2018

aussi notre futur



Le podium

1. 75 000 € de budget marketing et un stand sur le salon electronica 2020 : **Teiimo (Allemagne)** – électronique vestimentaire et textiles intelligents.
2. 50 000 € de budget marketing : **Wizama (France)** – console interactive de jeux de société.
3. 25 000 € de budget marketing : **Querom Elektronik (Allemagne)** – modules convertisseurs CC/CC haute tension.

Mention Honorable (santé les gars !)

750 € : Max & Michael Buschmann –
Cocktail Joe, préparateur automatique de cocktails.

services de prototypage rapide en ligne, systèmes de réalité virtuelle... Mais aussi quel plaisir et quel honneur pour nous de rencontrer des gens si inspirés, et de découvrir leurs façons d'aborder la création de produits électroniques. Ferez-vous partie de l'aventure dans deux ans ?

Article commandité

(180702-04 - version française : Hervé Moreau)

Lien

- [1] Page du concours **electronica Fast Forward 2018** :
www.elektormagazine.fr/labs/contest/e-ffwd-contest



Le 3^e prix a été attribué à **Querom Elektronik** pour ses solutions d'électronique de puissance sur mesure et ses convertisseurs CC/CC haute tension qui offrent des puissances de sortie de 1 à 10 kW, des tensions d'entrée de 260 à 900 VCC, et des tensions de sortie de 12 à 48 VCC.



Les concepteurs de **Cocktail Joe** devant leur préparateur automatisé. En attendant plus de monde autour !



projet 2.0

corrections, mises à jour et courrier des lecteurs

radar Doppler expérimental

Elektor 07-08/2018, p. 48 (160385)

Le circuit imprimé actuellement en vente chez Elektor en est à la version v1.2 ou v1.3 (identiques du point de vue électrique). Par rapport à celui de l'article (v1.1), ces versions présentent les modifications suivantes :

La résistance R45 (47 Ω) est nouvelle et ne figure donc pas dans la liste de composants de l'article. Elle contribue à réduire les parasites de commutation d'IC11. La résistance R44 de la première version (v1.1) est devenue R42 dans v1.2/v1.3, alors que R44 s'appelle R42 dans v1.2/v1.3 (une malheureuse permutation de références). La résistance R44 n'est utilisée que pour les tests et ne doit pas être installée. La résistance R42 ne doit être installée que si l'on utilise IC11 à la place d'IC4 (voir ci-dessous).

Pour être complet :

Tous les composants passifs sont au format 0805, sauf indication contraire. L'inductance L1 est du type Bourns SRR4828A-150M ou équivalent. La référence complète d'IC4 est ADP2370ACPZ-3.3-R7, abrégée en ADP2370-3.3 dans l'article. L'ADP2370 est la version réglable, qui ne convient pas ici.

Régulateur de tension 3,3 V :

Comme mentionné dans l'article, il existe deux variantes du régulateur 3,3 V. Il ne faut pas les installer toutes les deux !

Variante 1 (difficile à souder, moins de composants)	Variante 2 (plus facile à souder, plus de composants)
IC4, R29	C19, C20, C21, D2, D3, IC11, R42, R43, R45
Ne pas installer les composants mentionnés pour la variante 2.	Ne pas installer les composants mentionnés pour la variante 1.



référence de 10 MHz

Elektor 05-06/2018, p. 32 (160594)

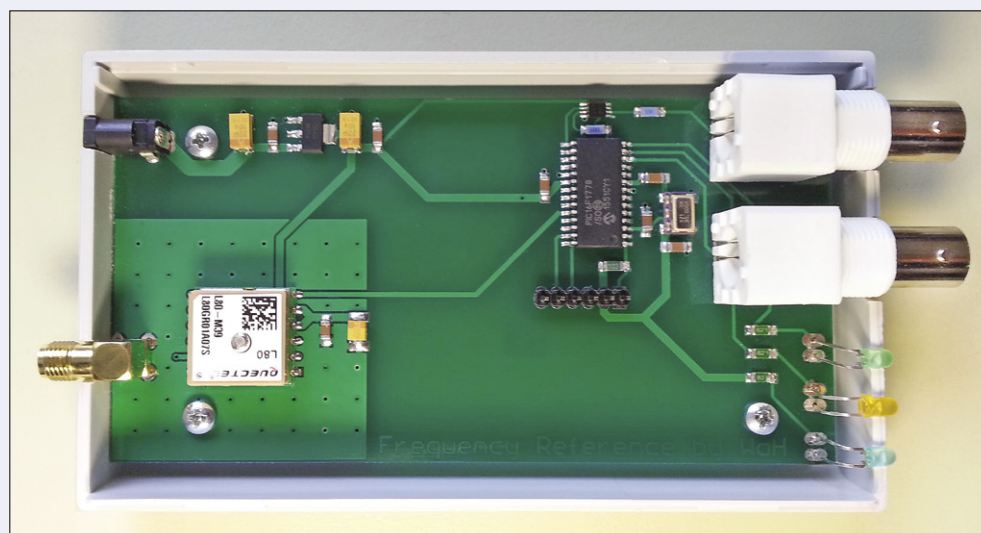
Votre article sur une référence de fréquence stabilisée par GPS répond exactement à mon besoin de stabilisation d'un LNB (Low Noise Block, convertisseur de fréquence de réception) à 10 GHz. Malheureusement, ce n'est pas 10 MHz qu'il me faut pour le LNB, mais 24 MHz. D'où ma question : est-il possible de remplacer le quartz de 20 MHz par un VXCO à 48 MHz, en faisant compter le compteur non pas jusqu'à 160 000 000, mais jusqu'à 384 000 000 avant d'arriver à l'égalité ? Cette fréquence d'entrée pose-t-elle un problème au PIC16F ? Est-ce que le compteur peut traiter cette valeur plus élevée ? Que faudrait-il modifier dans le micrologiciel ?

Karl-Gerhard Ruthemann

Réponse de l'auteur

À ma connaissance, il n'existe pas de PIC qui supportent une fréquence d'horloge externe supérieure à 20 MHz. Dans votre cas, on pourrait diviser la fréquence du VCTCXO par quatre et faire fonctionner le processeur avec une fréquence d'horloge de 12 MHz. On devrait alors modifier le compteur interne en conséquence et insérer un tampon entre le VCTCXO et la sortie. Cela ne devrait pas être un très gros problème.

Willem den Hollander





tubes de qualité spéciale (SQ)

Elektor 09-10/2018, p. 104 (180299-07)

Le lecteur de la première heure que je suis est toujours ravi par les articles de ce genre. Je viens de remettre la main sur un tube que j'avais acheté comme pièce de rechange (photo ci-contre) au début des années 70. S'il pouvait servir à compléter votre collection, je vous l'enverrais volontiers.

Dieter Becker

Le tube PL802 E de la photo est un modèle de remplacement à semi-conducteurs (« E » pour *Ersatz*) du tube à vide PL802. Je suppose qu'il contient un transistor à haute tension. Il n'a donc rien à voir avec les tubes de qualité spéciale présentés dans l'article. Mais je l'ajouterais volontiers à ma collection de tubes rares et d'appareils anciens, que je montre à l'occasion aux visiteurs du labo d'Elektor à Aix-la-Chapelle (Allemagne).

Jan Buiting



tubes de qualité spéciale (SQ)

Elektor 09-10/2018, p. 104 (180299-07)

J'ai une grosse collection de tubes dont je me sers pour construire de nombreux amplificateurs à tubes, surtout en OTL (*Output Transformerless Technology*, technologie sans transformateur de sortie). Je possède aussi beaucoup de tubes d'ordinateur provenant de systèmes informatiques de NCR (Computronic) des années 50, parmi lesquels des 5751 Command de RCA, des 5963 Sylvania Gold, et aussi des E81CC de Siemens (ainsi que des E83CC et des E88CC).

Merci à l'équipe de Rétronique pour ses articles toujours très intéressants.

Werner Frick

Merci pour votre contribution enthousiaste ! Cet article m'a valu beaucoup de réactions des lecteurs d'Elektor. J'avais prévu d'ajouter un paragraphe sur les procédures de test développées par Valvo Allemagne pour les tubes SQ, mais cela aurait exagérément allongé l'article. Je suis également collectionneur d'amplificateurs OTL (même si je me limite aux modèles à impédances de 800 Ω et 1200 Ω). Je pense que le principe de l'OTL est très sous-estimé dans le monde des amplificateurs à tubes. Dans un prochain numéro d'Elektor, je reviendrai peut-être avec un article sur l'amplificateur Philips 1200- Ω -Circlotron.

Jan Buiting



carte ESP32 Pico-Discovery-Board

Elektor 09-10/2018, p. 32 (180341-04)

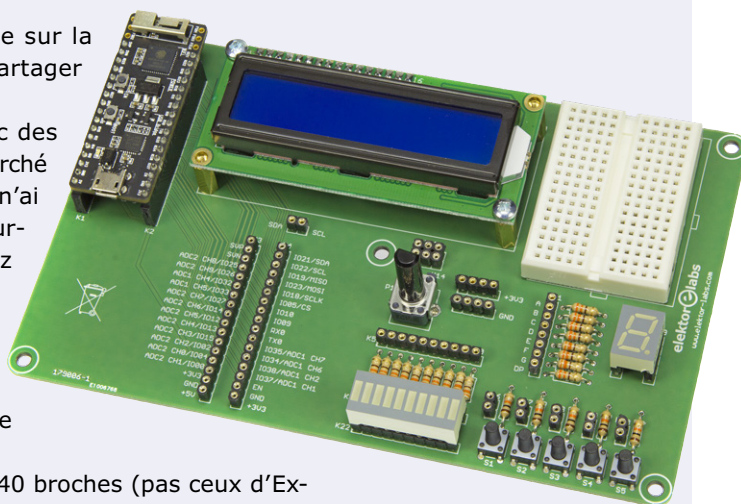
Je viens juste de lire votre passionnant article sur la carte ESP32 Pico-Discovery-Board. Je souhaite partager mon expérience avec vous.

Beaucoup de connexions avec la carte sont réalisées avec des embases femelles. Ces embases ne sont pas vraiment bon marché chez les revendeurs de composants électroniques. Alors je n'ai pas pu résister à la tentation de m'en procurer chez un fournisseur d'AliExpress, dix pièces pour le prix d'une seule chez le fournisseur local, et franco de port.

Ces trucs ont exactement le même aspect, mais quand j'utilise le fil de 0,6 mm habituel, je peux l'enfoncer une ou deux fois et ensuite il ressort tout seul, les contacts à ressort sont fatigués. Alors, j'ai mis ces embases au rebut, avant que je perde complètement mon calme.

J'ai aussi acquis une bonne expérience avec les supports à 40 broches (pas ceux d'Extrême-Orient). On peut les couper facilement avec une lame de couteau chauffée.

Ruedi Heimlicher



feu de phare 2.0

à signature lumineuse réglable

Friedrich Lischeck (Allemagne)

Dans le numéro de novembre 2010, Léo Szumylowycz présentait un circuit qui imite la lumière du faisceau rotatif d'un phare [1]. Le circuit n'utilisait que des composants analogiques, comme le légendaire temporisateur 555 et l'ampli-op LM358, et il fonctionnait à merveille. Seul inconvénient : 29 composants... Avec un microcontrôleur ATtiny45, il n'en faut plus que huit, alors que le nombre de paramètres réglables augmente !

Lorsque vous regardez un phare, vous constatez que l'intensité lumineuse augmente lentement quand le faisceau tourne vers vous, puis il y a un éclat soudain quand la lampe est juste en face de vous, et enfin la luminosité diminue à nouveau. Les périodes d'obscurité et de répétition du signal lumineux constituent, avec la couleur du signal, la signature lumineuse qui

permet de distinguer un phare de manière univoque. On peut facilement simuler ce comportement avec un signal triangulaire auquel on superpose de brèves impulsions ; nous l'avons fait avec un microcontrôleur programmé en BASCOM-AVR® (*Windows BASIC compiler for the AVR family*).

Le circuit

Le schéma repris en **figure 1** illustre bien la simplicité du circuit, comme c'est souvent le cas lorsqu'on utilise un microcontrôleur : il suffit d'ajouter deux résistances, trois potentiomètres, un condensateur et une LED. Le montage est donc très compact et le logiciel pallie le « manque » de composants. La résistance R1 veille à un démarrage correct, R2 limite l'intensité du courant de la LED et le condensateur C1 filtre les éventuels parasites de l'alimentation.

Les potentiomètres connectés aux entrées ADC1, ADC2 et ADC3 de l'ATtiny permettent les réglages suivants :

- P1 : le phare virtuel peut émettre un feu scintillant (signaux brefs et rapprochés), ou un feu à éclat. Dans ce cas, l'intensité du faisceau augmente lentement jusqu'à un éclat, puis diminue lentement, comme avec le montage décrit en 2010. Le potentiomètre P1 détermine le rapport entre la durée d'augmentation/diminution du faisceau et l'éclat, donc la durée de l'éclat. À une extrémité du curseur, il n'y a plus d'éclat, seulement une augmentation du

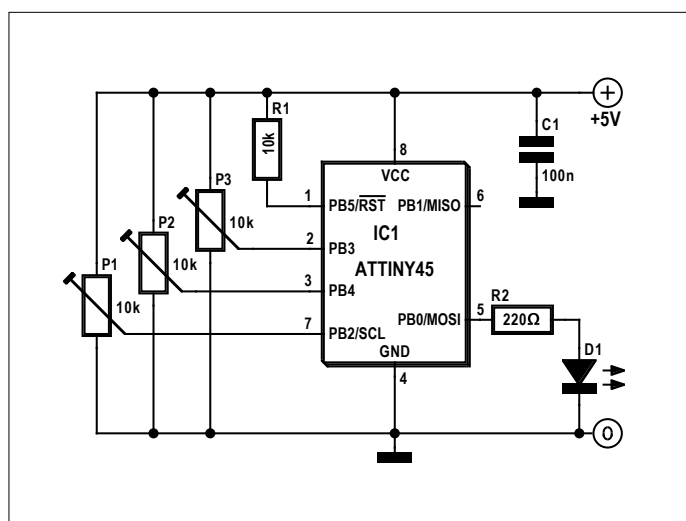


Figure 1. Un circuit des plus simples : seulement huit composants.

faisceau suivie d'une diminution. À l'autre extrémité du curseur, le feu est scintillant. La durée totale du cycle est toujours d'une seconde.

- P2 : la durée des périodes d'illumination et d'occultation (par ex. 1 s allumé et 1 s éteint, puis 1 s allumé et 10 s éteint) correspond à la durée de la signature lumineuse du phare (dans ce cas 13 s). Le potentiomètre P2 permet de régler cette durée, ce qui n'était pas le cas avec le circuit de 2010.
- P3 : il est possible de régler un autre paramètre de la signature lumineuse, à savoir le nombre d'éclats ; c'est une nouveauté de ce montage. Si la durée totale fixée avec P2 n'est pas suffisante pour le nombre d'éclats choisi, elle est automatiquement allongée à la durée des éclats plus 2 s.

Ces divers paramètres permettent déjà de simuler nombre de phares réels.

Le logiciel

L'essentiel du travail est bien entendu effectué par un programme écrit en BASIC et dont le code source est disponible sur la page du projet [2]. Nous détaillons ci-après les aspects les plus intéressants de ce programme.

Pour cette version 2.0 du feu de phare, les temporisateurs *Timer0* et *Timer1* de l'ATtiny45 sont nécessaires. Le premier gère la modulation de largeur d'impulsion (MLI) qui commande la luminosité de la LED ; le second est le signal d'horloge qui commande les périodes d'allumage de la LED. Il y a une astuce : BASCOM ne permet pas la commande directe du temporisateur *Timer1*, il faut donc écrire directement dans le registre *TCCR1* du microcontrôleur.

```
Preload1 = 99 'ceci correspond à 0,01 s
TCCR1 = &H0A 'équivalent à Config Timer1=TIMER,
    Prescale = 512
on Timer1 Zeitbasis1
enable Timer1
enable Interrupts
```

Notez qu'une division préalable par 512 est possible ici, ce qui n'est pas le cas avec des microcontrôleurs plus récents d'Atmel, par ex. l'ATMega8. *Zeitbasis1* (base de temps) est une routine temporelle gérée par interruption. *Timer0* est utilisé pour la MLI et est adressé normalement par BASCOM.

```
Config Timer0 = Pwm , Compare_A_Pwm = Clear_Up ,
    Compare_B_Pwm = Clear_Up , Prescale = 1
Enable Timer0
```

Zeitbasis1 (voir **Listage 1**) constitue le noyau du programme. Cette routine, appelée par interruption, utilise les variables *LEDAn* (LED allumée), *Blitzstart* (début de l'éclat) et *Blitzende* (fin de l'éclat), dont les valeurs sont fixées dans le programme principal. La routine utilise également le champ *Form(N)* du programme principal, qui retourne la valeur de la variable *Zehntel* (dixième) à ce programme.

La routine est appelée tous les centièmes de seconde et comprend deux parties. Dans la première partie, les variables *Hundertstel*, *N* et *Zehntel* sont incrémentées ; *Hundertstel* contient les centièmes de seconde, *Zehntel* les dixièmes. *N*

Listage 1

```
Zeitbasis1:
Timer1 = Preload1
    incr Hundertstel
    incr N
    if Hundertstel>9 then
        Hundertstel=0
        Incr Zehntel
    end if

    if N >100 then N=1
    if LEDAn=1 then
        Ausgabe=Form(n)
        if n>= Blitzstart and n<= Blitzende then
            Ausgabe=255
        OCR0A= Ausgabe 'Nieuwe helderheid instellen
    else
        OCR0A=0
    end if
Return
```

contient aussi les centièmes de seconde, mais sa valeur peut varier de 1 à 100, alors que celle de *Hundertstel* ne va que de 1 à 9 et ne sert qu'à incrémenter les dixièmes de seconde de *Zehntel*.

La luminosité de la LED est déterminée dans la deuxième partie de la routine. Tous les centièmes de seconde (variable *N*), une nouvelle valeur de la luminosité est réglée si la LED est allumée à ce moment (drapeau levé dans *LEDAn* du programme principal). Les valeurs de la luminosité sont entreposées dans *Form(N)* ; les 100 valeurs de *N* couvrent une durée de 1 s, soit la durée d'illumination de la LED. La valeur de *Form(N)* est transférée à la variable *Ausgabe* (sortie).

Ceci ne permet que l'augmentation ou la diminution progressive de la luminosité. Celle-ci peut être modifiée à l'aide des variables *Blitzstart* et *Blitzende*, dont la valeur varie de 1 à 100 chaque centième de seconde lorsque la LED est allumée. Si *N* est supérieur ou égal à *Blitzstart* et inférieur ou égal à *Blitzende*, la luminosité de la LED sera maximale (valeur 255). Cette valeur remplace la valeur actuelle de la variable *Ausgabe*. Avec *OCR0A= Ausgabe*, le rapport cyclique de la MLI est modifié pour commander la LED.

Comme déjà mentionné, cela ne se produit que lorsque le drapeau *LEDAn* est levé. Lorsque ce drapeau est abaissé (LED éteinte), rien de tout cela ne se produit et la valeur de la modulation de largeur d'impulsion est mise à zéro avec *OCR0A=0*.

Programme principal

Le programme principal tourne dans une boucle sans fin (*DO-LOOP*) et comprend deux groupes fonctionnels. Dans un premier temps, les positions des potentiomètres P1 à P3 sont mesurées et des valeurs normalisées attribuées en conséquence aux trois variables *Blitzstart/Blitzende*, signature lumineuse et durée d'identification. Ceci uniquement si la valeur de *LEDAn* est nulle.

Listage 2

```

If Zehntel <> Lzehntel Then
  Incr X
  If X = Wiederkehr Then
    X = 1
    disable Interrupts
    N = 0
    enable Interrupts
  end if
  if X < Kennungsdauer then
    If X = Naechste Then
      Naechste = Naechste + 10
      if LEDAn = 1 then
        LEDAn = 0
      else
        LEDAn = 1
      end If
    end If
  end if
  If X = Kennungsdauer then
    LEDAn= 0
    Naechste=1
  end if
  Lzehntel = Zehntel
End If

```

La deuxième partie (voir **Listage 2**) concerne le basculement de **LEDAn** suivant le tempo souhaité. Chaque exécution de la boucle ne donne pas lieu à une action, mais il faut attendre tous les dixièmes de seconde. Ceci est réglé par les deux variables

Zehntel et **LZehntel**, cette dernière indiquant le dernier dixième nécessitant une action.

L'état de **LEDAn** (1 ou 0) est déterminé toutes les secondes à l'aide des variables **Naechste** (suivant) et **X**. La valeur de **X** est incrémentée d'un tous les dixièmes de seconde, et remise à 1 lors du calcul de la valeur de la signature lumineuse (**Wiederkehr**). La variable **N** (routine **Zeitbasis1**) est alors synchronisée et mise à 0. Afin que cela ne se passe pas lors de la mise en œuvre de la routine d'interruption, les interruptions sont d'abord désactivées puis réactivées par la suite.

Lorsque la valeur de **X** atteint celle de **Naechste**, c'est-à-dire lorsqu'une seconde s'est écoulée, **LEDAn** change d'état. La valeur de **Naechste** est aussi incrémentée de 10, ce qui revient à déclencher l'action suivante une seconde plus tard. Ceci se passe tant que la valeur de **X** est inférieure à celle de la durée d'identification (**Kennungsdauer**) ; lorsque celle-ci est dépassée, **LEDAn** est mise à 0 et **Naechste** à 1, de telle manière que **LEDAn** ne soit modifiée qu'au début d'un nouveau cycle.

Le code suivant est utilisé tant que le drapeau est levé :

```

for I=1 to 10: Form(I)=1:Next I
for I=11 to 20: Form(I)=2:Next I
for I=21 to 30: Form(I)=3:Next I
for I=31 to 40: Form(I)=4:Next I
for I=41 to 50: Form(I)=5:Next I

```

```

For I = 51 To 100
  J = 101 - I
  Form(i) = Form(j)
Next I

```

Les valeurs des éléments 1 à 50 sont initialisées par pas de dix. On retrouve les mêmes valeurs pour les éléments 51 à 100,

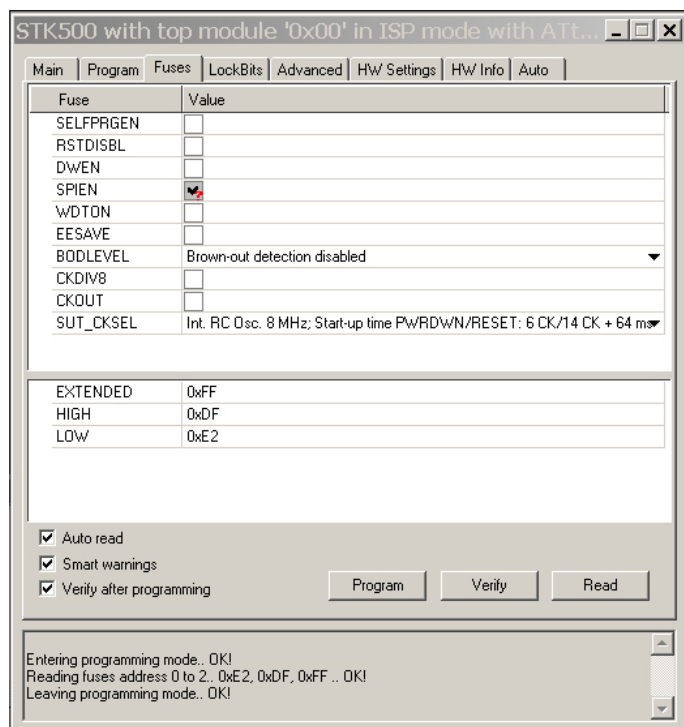


Figure 2. Réglage des fusibles de l'ATtiny45 tels que dans l'AVR-Studio 4.18.

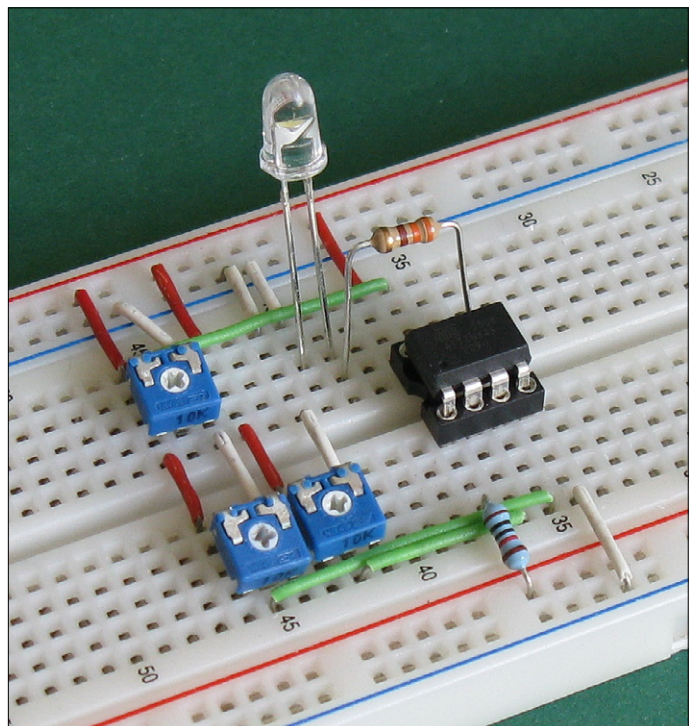


Figure 3. Une platine d'expérimentation suffit pour ce circuit.

mais en sens inverse. La variable `Form(I)` évolue donc comme un signal triangulaire, avec une valeur maximale de 5. Les fusibles de l'ATtiny45 doivent avoir les valeurs suivantes pour que le programme fonctionne correctement :

EXTENDED: 0xFF

HIGH : 0xDF

LOW : 0xE2

Voir aussi la **figure 2**.

Montage

Le circuit peut être monté sur une plaque d'essai ou une platine d'expérimentation (**fig. 3**). Nous vous conseillons d'utiliser un support pour l'ATtiny. Il existe même des supports avec condensateur intégré, comme c'est le cas à la figure 3 ; le condensateur C1 n'est pas nécessaire dans ce cas.

Réglages

Il faut tout d'abord mettre les trois potentiomètres à leur valeur minimale. On règle ensuite la durée des éclats avec P1. Le nombre d'éclats par cycle est quant à lui fixé avec P3. La durée du cycle de la signature lumineuse est encore minimale à ce

moment, et le programme la prolonge automatiquement de deux secondes de plus que ne l'exigent les éclats et les interruptions qui les accompagnent. Lorsque le nombre d'éclats et leur durée sont fixés, on peut régler la durée totale de la signature lumineuse avec le potentiomètre P2. Le feu de phare est désormais prêt à signaler une zone dangereuse ou un port de votre maquette. ◀

(180334-04 – version française : Jean-Louis Mehren)



@ WWW.ELEKTOR.FR

→ Livre « 50 mini projets instructifs à microcontrôleur ATtiny », ISBN 978-2-86661-204-7 :

www.elektor.fr/50-mini-projets-instructifs-a-microcontroleur

→ Ebook en anglais « Microcontroller Basics »,

ISBN 978-0-905705-67-5 :

www.elektor.fr/microcontroller-basics-ebook-en

→ Livre en anglais « Make: AVR Programming »,

ISBN 978-1-449355-78-4

www.elektor.fr/make-avr-programming

Liens

[1] « Feu de phare », Elektor 11/2010 : www.elektormagazine.fr/100202

[2] Page de l'article : www.elektormagazine.fr/180334-04

Publicité

Salon de l'innovation en électronique, mesure, vision et optique

L'INNOVATION INDUSTRIELLE S'INVITE À L'OUEST

**3-4
AVRIL
2019**

**CITÉ DES CONGRÈS
NANTES**

Made by
**GL
events**

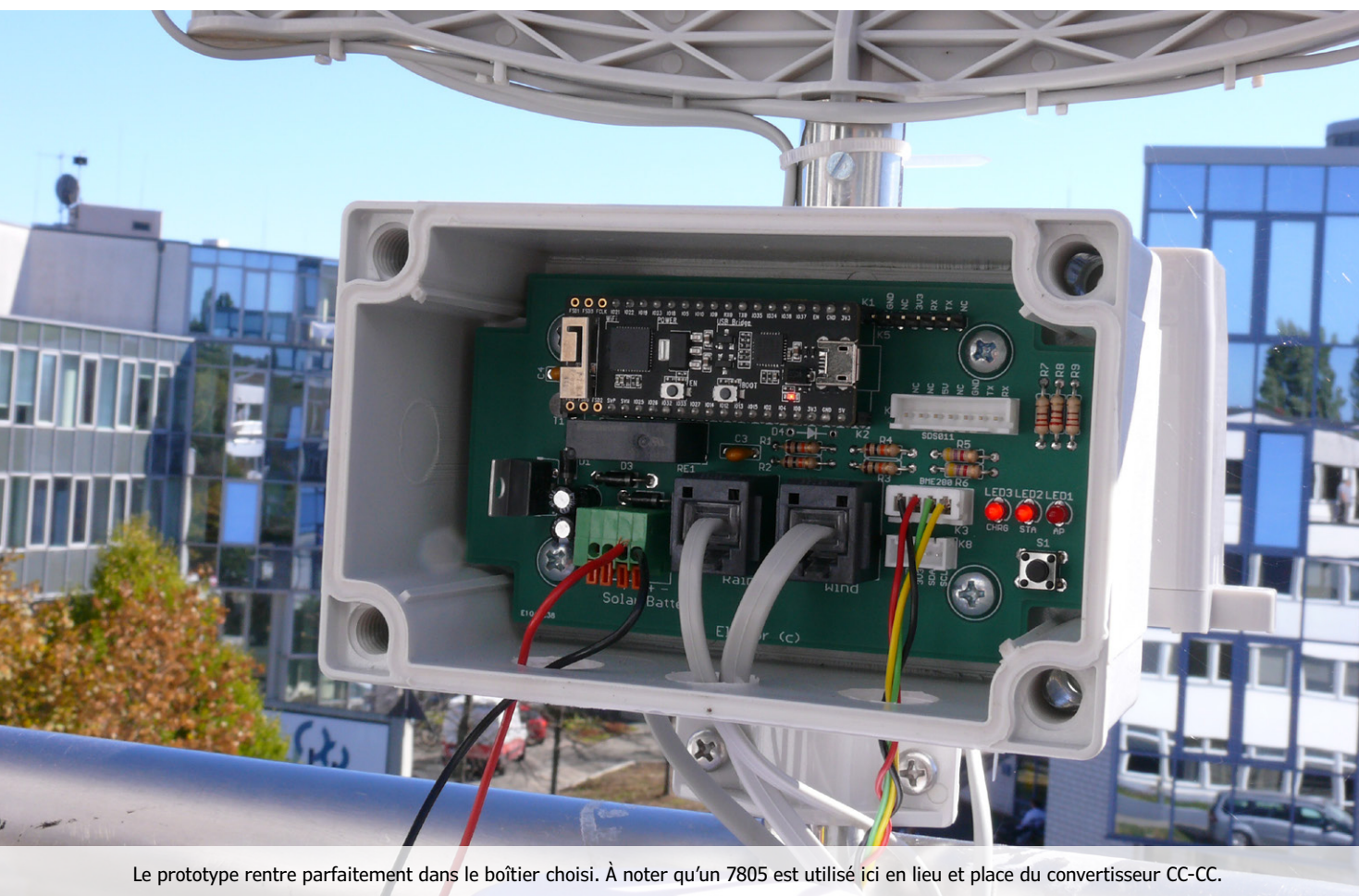
Votre badge gratuit sur www.enova-event.com

station météo à ESP32

lecture des données des capteurs à distance (Wi-Fi)

Roy Aarts (labo d'Elektor) & Thijs Beckers (rédaction d'Elektor)

La carte ESP32 est un outil polyvalent et bon marché, bien adapté à divers projets de domotique. Nous avons choisi de construire une station météo autour de l'ESP32, ainsi tous les résultats des mesures sont disponibles en ligne.



Le prototype rentre parfaitement dans le boîtier choisi. À noter qu'un 7805 est utilisé ici en lieu et place du convertisseur CC-CC.

Caractéristiques

- Mesure de la température et du taux d'humidité de l'air, de la direction et de la vitesse du vent, de la pression atmosphérique et des précipitations
- Détecteur de particules fines en option (Nova Fitness SDS011)
- Connecteurs additionnels pour des modules Grove par ex.
- Compatible *Thingspeak* et *senseBox*
- Configuration via la page web interne de l'ESP32
- Alimentation par panneaux solaires, accu de 12 V, ou adaptateur secteur de 8 à 28 V_{cc}

La station météo à ESP32 mesure les principales caractéristiques de l'air ambiant : température et humidité de l'air, vitesse et direction du vent, précipitations et pression atmosphérique. Il est également possible de mesurer la concentration en particules fines avec un capteur spécialisé, le SDS011 de Nova Fitness [1] dont la résolution est de 0,3 µg/m³. La température, le taux

d'humidité et la pression atmosphérique sont mesurés à l'aide du capteur BME280 de Bosch [2]. Pour la vitesse et la direction du vent, ainsi que les précipitations, on utilise un kit spécial, disponible dans l'e-choppe d'Elektor.

Les résultats des mesures sont envoyés par l'ESP32 vers *Thingspeak* [3] ou *senseBox* [4]. *Thingspeak*, de Mathworks, est une base de données en ligne qui permet le stockage de données et leur traitement ultérieur – par ex. la création de graphiques. Les données peuvent aussi être traitées avec *Matlab*.

SenseBox est une sorte de carte de prévisions météo à code source ouvert. Les utilisateurs qui disposent d'un kit senseBox ou d'un appareil compatible peuvent charger les résultats de leurs mesures ; elles seront ensuite retranscrites sur une carte du monde, visible par tout un chacun.

Le matériel

La station météo est bâtie autour de l'ESP32-PICO-KIT V4 ; son schéma est repris en **figure 1**. Afin de faciliter la mise en boîtier étanche (un modèle

PC 100/60 HT de Fibox) et le raccordement des capteurs, nous avons dessiné un circuit imprimé. L'anémomètre-girouette et le capteur de précipitations y sont reliés via des connecteurs RJ45, les BME280 et SDS011 via des connecteurs JST XH. Le BME280 utilise le bus I²C, le SDS011 une liaison série asynchrone (UART). Nous avons prévu deux connecteurs supplémentaires sur le circuit imprimé pour d'autres capteurs ou périphériques ; le premier permet le raccordement de modules *Grove I²C* de *Seed Studio* [5-6], le second (FTDI) permet une liaison série (UART).

Le circuit est alimenté par un accu au plomb de 12 V, et nous avons prévu un petit extra pour sa recharge par des panneaux solaires. Lorsque l'accu est presque vide, l'ESP32 enclenche un relais, et le déclenche lorsque la recharge est terminée. La tension de l'accu est mesurée via le diviseur de tension R11/R12 (470 kΩ et 100 kΩ) ; le démarrage et l'arrêt de la charge peuvent être réglés par logiciel. La LED3 indique la charge de l'accu. Le relais est commandé par le MOSFET T1, dont le courant de grille est

INFOS SUR LE PROJET

capteurs

mesure

débutant

connaisseur

expert

env. 4 h

Outils de soudage standard

env. 90 €

limité par R10 et qui est protégé par D3 des surtensions produites par la bobine du relais. Le courant des LED est lui aussi limité par des résistances série.

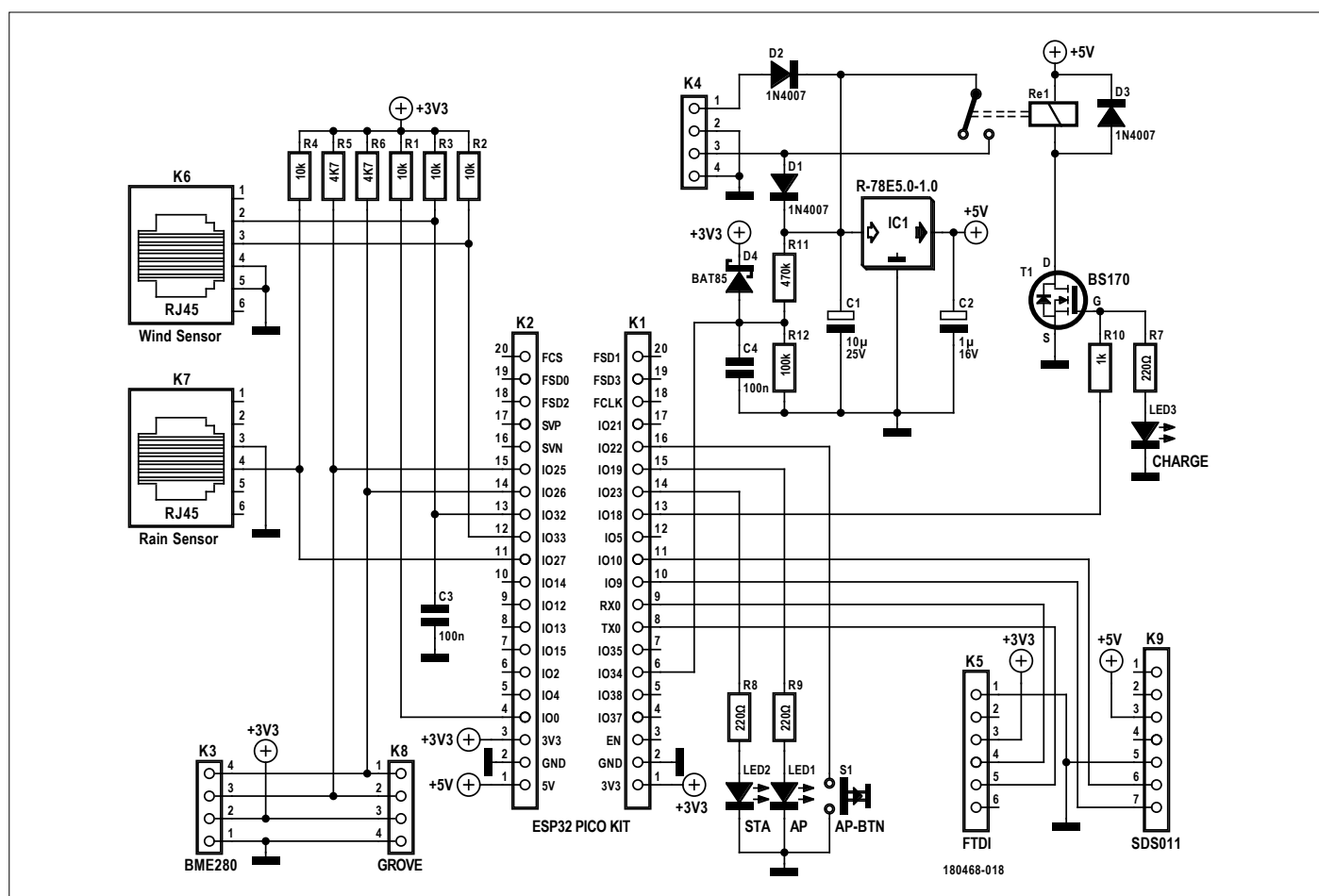
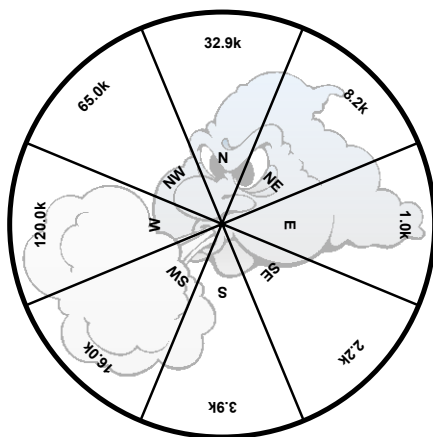


Figure 1. Le schéma montre bien que l'ESP32 est le cœur du système.



Mesures des capteurs disponibles dans le monde entier

Un convertisseur CC-CC à découpage – dont le rendement est beaucoup plus élevé que celui d'un simple 7805 et donc mieux adapté à une alimentation par bat-

Figure 2. La résistance interne de la girouette dépend de la direction du vent.

terie – délivre une tension stable de 5 V. Celle-ci est à son tour convertie en 3,3 V par le régulateur de la carte ESP32-PICO, un 1117, pour l'alimentation du circuit et des capteurs.

La diode D4 protège l'ESP32 d'une éventuelle surtension sur la ligne d'alimentation de l'accu (via le diviseur R11/R12), et le condensateur C4 filtre les parasites pour que la lecture de la tension de l'accu soit relativement stable.

Les résistances R1-R6 sont des résistances de rappel nécessaires au bon fonctionnement des circuits raccordés.

Capteurs de précipitations et de vent

Pour la mesure des précipitations, on utilise un petit bac qui se remplit d'eau de pluie. Celui-ci bascule lorsqu'il est plein et court-circuite brièvement les contacts d'un circuit magnétique, ce qui envoie un signal vers l'ESP32. La fiche de caractéristiques indique un basculement pour 0,33 mm de précipitations.

Une girouette, dans laquelle une résistance change de valeur en fonction de sa position, permet la mesure de la direction (approximative) du vent. La **figure 2** montre la relation entre valeur de la résistance et la direction. Cette résistance forme avec R2 un diviseur de tension, dont la valeur est mesurée par l'ESP32. L'anémomètre est un modèle à coupelles. Un circuit magnétique délivre une impulsion toutes les deux révolutions. La différence de temps entre deux impulsions, multipliée par deux, nous donne la période du signal ; en l'inversant, on obtient la fréquence en Hz. En multipliant cette fréquence par 0,66 – facteur de multiplication donné par la fiche de caractéristiques – on obtient la vitesse du vent en m/s.

La conversion en km/h est simple, on multiplie le résultat en m/s par 3,6. Une rotation par seconde de l'anémomètre correspond donc à une vitesse du vent de 2,4 km/h.

Chargement du logiciel

Pour charger le logiciel sur l'ESP32, il faut tout d'abord installer le noyau Arduino pour cette carte [7]. Il faut ensuite installer l'outil SPIFFS pour pouvoir télécharger la page web vers le système de



LISTE DES COMPOSANTS

Résistances

R1-R4 = 10 kΩ
R5, R6 = 4,7 kΩ
R7-R9 = 220 Ω
R10 = 1 kΩ
R11 = 470 kΩ
R12 = 100 kΩ

Condensateurs

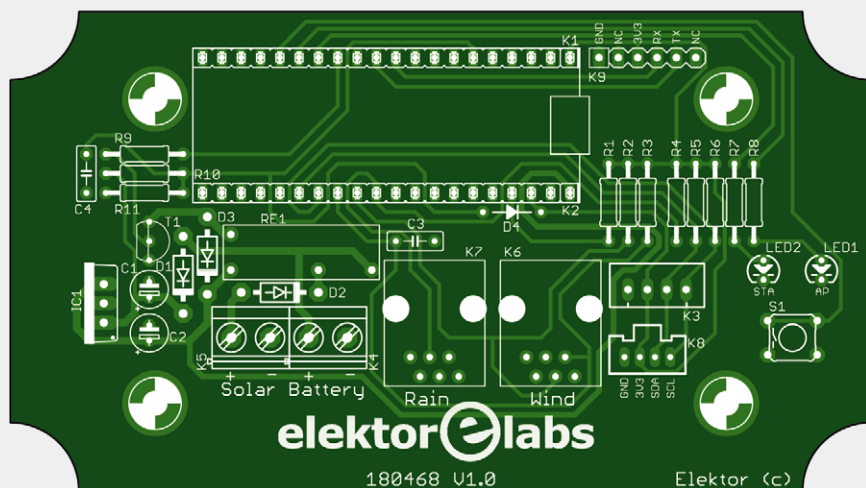
C1 = 10 μF/50 V
C2 = 1 μF/10 V
C3, C4 = 100 nF

Semi-conducteurs

D1-D3 = 1N4007
D4 = BAT85
IC1 = convertisseur CC-CC 5 W
(Recom Power, R-78E5.0-1.0)
LED1, LED2 = LED jaune, 3 mm
LED3 = LED rouge, 3 mm
T1 = BS170

Divers

K1, K2 = barrette femelle droite sécable, à 17 contacts, au pas de 2,54 mm
K3 = connecteur mâle, à 4 contacts, au pas de 2,54 mm
K4 = bornier à 4 contacts, au pas de 2,54 mm (Würth 691403900004B)
K5 = barrette mâle droite sécable, à 6 contacts, au pas de 2,54 mm
K6, K7 = connecteur femelle RJ45 droit pour circuit imprimé
K8 = connecteur Grove, vertical, TWIG-4P-2.0-2.0
K9 = connecteur mâle, à 7 contacts, au pas de 2,54 mm, JST-XH-07-PIN-VER
RE1 = relais ALDP105
S1 = bouton-poussoir, 6×6 mm
Carte de liaison (Breakout Board) BME280, version I²C
Anémomètre-girouette et capteur de précipitations
Boîtier Fibox, réf. PC 100/60 HT
Circuit imprimé, réf. 180468-1, www.elektor.fr
En option : détecteur de particules fines Nova Fitness SDS011



fichiers de l'ESP32 [8]. Si ces étapes se sont déroulées correctement, on peut alors sélectionner l'ESP32-PICO-KIT dans la liste des cartes disponibles et charger la page web et le croquis.

Le logiciel

Comme dit ci-devant, la station est gérée par l'ESP32, et la configuration se fait à l'aide d'une page web entreposée dans la mémoire de la carte. Au démarrage l'ESP32 essaie de se connecter au réseau ; en cas d'échec il lance le serveur qui contient la page web. Le serveur peut aussi être lancé manuellement en appuyant – suffisamment longtemps pour être certain que le circuit soit alimenté – sur le bouton-poussoir qui se trouve sur le circuit imprimé ; on peut également utiliser le bouton EN de la carte ESP32. Si l'utilisateur n'a aucune action dans la page de configuration pendant 10 min, l'ESP32 redémarre et essaie à nouveau de se connecter au réseau. Pour éviter cela, vous pouvez remplacer le bouton-poussoir par un interrupteur ou un pont pour maintenir l'état fermé.

La page de configuration permet de consulter les résultats des mesures, ainsi que la valeur mesurée de la tension d'alimentation. On peut aussi modifier les paramètres de connexion (choix du nom du réseau (SSID) et mot de passe), et ceux de téléchargement (clés de l'interface de programmation (API) pour Thingspeak et senseBox, ainsi que la fréquence de téléchargement).

Pour accéder à la page de configuration, il faut bien entendu être connecté à l'ESP32. La page est à l'adresse IP 192.168.4.1. Si vous utilisez un téléphone sous Android, il faudra supprimer temporairement les données mobiles, car le téléphone bascule automatiquement sur ces données s'il ne détecte pas une connexion internet valable par Wi-Fi, ce qui supprime l'accès à la page de configuration.

La page web est un fichier html qui comprend trois éléments principaux : `<style>` qui définit l'aspect de la page, `<body>` pour sa structure et son contenu, et `<script>` pour les fonctions à exécuter. Javascript doit être activé dans le navigateur pour un fonctionnement correct de la page.

Les communications entre l'ESP32 et la page web se font via une demande « asynchronous XMLHttpRequest ». Lors du chargement de la page une demande http est introduite et envoyée vers l'ESP32 ; celui-ci répond à la demande avec la configuration ou les valeurs



Thingspeak

Thingspeak, de Mathworks, est une base de données en ligne gratuite qui permet la sauvegarde de données. Pour chaque ensemble de données, il faut créer un nouveau canal et des champs (fig. 3). On reçoit ensuite une clé pour l'interface de programmation (API, fig. 4), qui doit être introduite sur la page de configuration de la station météo.

New Channel

Name:

Description:

Field 1: ☒

Field 2: ☒

Field 3: ☒

Field 4: ☒

Field 5: ☒

Field 6: ☒

Field 7: ☒

Field 8: ☒

Metadata:

Tags:

(Tags are comma separated)

Figure 3. Sur Thingspeak, on définit un nouveau canal pour l'application et on indique les données dans les champs utilisés.

Thingspeak

API key:

Wind speed:

Wind direction:

Rain amount:

Temperature:

Humidity:

Airpressure:

PM2.5:

PM10:

Figure 4. On reçoit ensuite une clé à utiliser dans le logiciel pour l'interface de programmation (API).

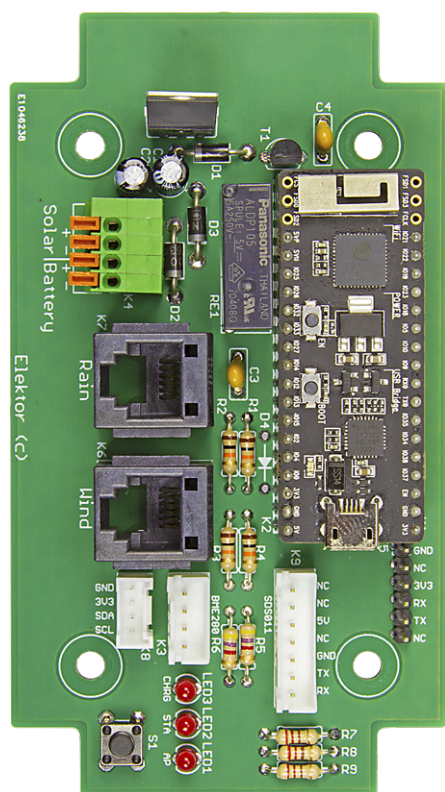
demandées. Si on appuie par ex. sur le bouton `submit` de la page web, une demande est envoyée avec les valeurs que l'on y a introduites.

Passons à la pratique

Il ne vous reste plus qu'à installer votre station météo et à raccorder les capteurs au circuit. Bien entendu, il est important que l'ESP32 ait accès à votre réseau, sinon vous ne pourrez pas charger les données des capteurs. Pour l'installation de ceux-ci, tenez compte de leurs caractéristiques et utilisez surtout votre bon sens : par ex. ne montez pas le capteur de précipitations contre un mur... ❏

(180468-04 -

version française : Jean-Louis Mehren)



SenseBox

SenseBox est un kit qui permet de construire sa propre station équipée d'une batterie de capteurs, dont les résultats de mesure sont envoyés vers *openSenseMap.org*. Là les données de toutes les « boîtes de capteurs » ou d'appareils compatibles peuvent être consultées. Pour enregistrer sa station météo, il suffit d'ouvrir un compte, puis d'indiquer le type de senseBox utilisé ou de configurer un autre type de station compatible avec les paramètres mesurés, les unités de mesure et le type de capteurs (**fig. 5**). C'est cette procédure qu'il faut suivre pour la station météo à base d'ESP32. Après enregistrement de la station, on reçoit un identifiant pour la station et pour chaque capteur ; après saisie de ces identifiants dans la page de configuration de la station météo, les résultats de mesure pourront être téléchargés.

☑ SenseBox	
Station ID	abcdefg
Wind speed ID	abcdefg
Wind direction ID	abcdefg
Rain ID	abcdefg
Temperature ID	abcdefg
Humidity ID	abcdefg
Pressure ID	abcdefg
PM2.5 ID	abcdefg
PM10 ID	abcdefg

Figure 5. Sur *openSenseMap.org*, on crée les capteurs de la station météo à ESP32.



@ **WWW.ELEKTOR.FR**

→ Circuit imprimé nu, réf. 180468-1 :

www.elektor.fr/esp32-ws-pcb

→ Kit avec composants, réf. 180468-71 :

www.elektor.fr/esp32-ws-kit

→ Kit de station météo, réf. 18696 :

www.elektor.fr/ws02

Autres produits :

→ Carte de développement ESP32-PICO-KIT V4, réf. 18423 :

www.elektor.fr/esp32-pico-kit-v4

→ Module BME280 (version I²C, 160109-91), réf. 17954 :

www.elektor.fr/bme280-mouser-intel-i2c-version-160109-91

Liens

- [1] Capteur de concentration en particules fines, SDS011 de Nova Fitness : www.inovafitness.com/en/a/chanpinzhongxin/95.html
- [2] Capteur de taux d'humidité et de pression atmosphérique, BME280 de Bosch : www.elektor.com/bme280-mouser-intel-i2c-version-160109-91
- [3] Plateforme gratuite de l'Internet des Objets : <https://thingspeak.com>
- [4] Site de partage de mesures : <https://sensebox.de/en>
- [5] Produits Seeedstudio chez Elektor : www.elektor.fr/seeedstudio
- [6] Modules Grove de Seeedstudio : www.seeedstudio.com/s/grove.html
- [7] Noyau ESP32 pour EDI Arduino : <https://github.com/espressif/arduino-esp32>
- [8] Système de fichiers pour l'ESP32 : <https://github.com/me-no-dev/arduino-esp32fs-plugin>

obsolète ou « attribué » ?

Jan Visser

Mercredi après-midi, aux environs de 15 h : encore quelques composants à commander pour divers projets en cours, mais tout d'abord vérifier le stock de CMS du labo. Il ne manque heureusement que des composants passifs standard : des résistances en boîtier 0805 de 1, 10 et 100 k Ω plus quelques autres valeurs, et des condensateurs de 1, 10 et 100 nF, eux aussi en boîtier 0805. J'ai quelques contrôleurs et autres composants exotiques sur ma liste de composants, et je sens qu'ils seront difficiles à trouver ; on va toujours commander les composants standard, ce sera toujours ça de pris. Deux heures et quelques tasses de café plus tard : pas de souci pour les contrôleurs et les composants exotiques, ils sont même disponibles en plusieurs versions et en grande quantité chez plusieurs fournisseurs ! Quant aux composants passifs, c'est une autre paire de manches...

Les délais de livraison des résistances et condensateurs standard en boîtier CMS 0805 sont soudain de plusieurs semaines, voire de plusieurs mois, ou alors on vous dit « réapprovisionnement en cours », sans autre précision ! Lorsque vous avez passé commande, certains fournisseurs vous envoient aimablement un message avec la date d'envoi estimée ; j'ai ainsi reçu un courriel pour des condensateurs de 1 μ F, supposés être bientôt disponibles : date de livraison estimée fin 2019. On croit rêver ! J'ai contacté divers fournisseurs et des connaissances chez les fabricants ; les constructeurs de téléphones mobiles, télévisions et autres appareils électroniques achètent apparemment en masse les stocks de composants standard, pour répondre à une demande en hausse.

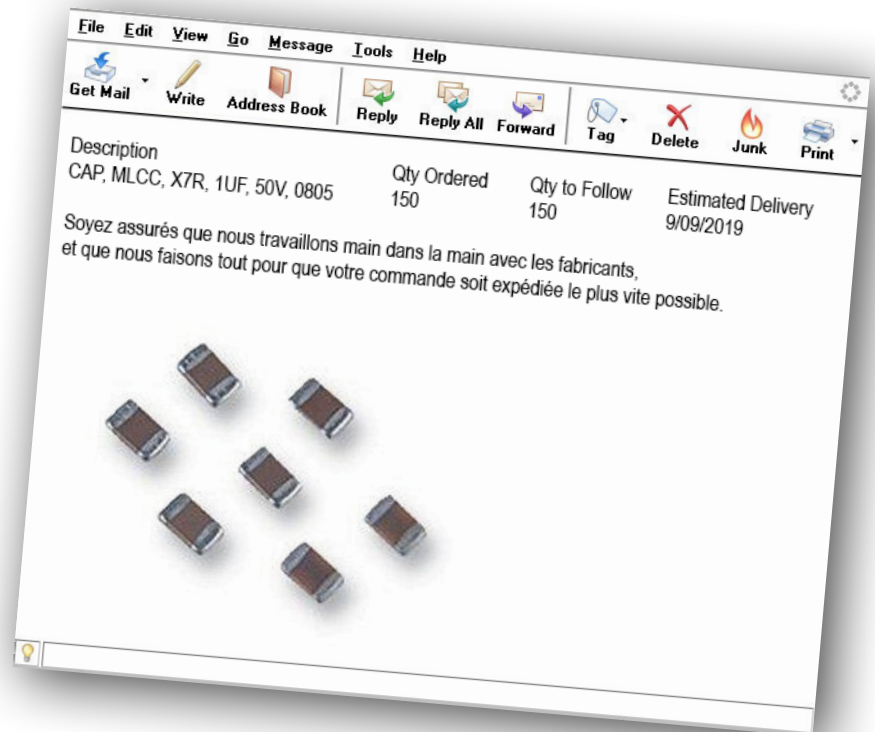
Et il n'y a pas qu'eux : il y a aussi des négociants et d'autres intermédiaires qui achètent les stocks qu'ils peuvent encore trouver et en marchandent la revente comme s'il s'agissait d'actions cotées en bourse. Vous direz que c'est la conséquence normale de la croissance économique. En tout cas, ces négociants qui achètent des stocks de composants

puis les laissent dormir dans un entrepôt jusqu'à ce que les prix aient doublé ou quadruplé, ils n'ont rien à voir avec le monde de l'électronique. Ce ne sont que des requins attirés par l'appât du gain, mais les dommages sont bien réels : des composants tout à fait ordinaires semblent obsolètes (on a l'impression qu'ils ne sont plus fabriqués). En réalité il n'en est rien ; les stocks existent bel et bien, mais ne sont pas disponibles, comme s'ils avaient été « attribués » à quelqu'un. Pour le commun des amateurs d'électronique, les petits labos ou les *markerspaces*, c'est un fléau ! Pas question de négocier l'achat de dizaines de milliers de composants ou de payer une fortune pour quelques résistances... Je travaille chez Elektor depuis pas mal de temps, et je me suis souvenu d'un article de janvier 2006 où on prévoyait déjà un problème de disponibilité. On conseillait de récupérer les circuits intégrés, les contrôleurs et autres composants exotiques sur les appareils usagés pour pouvoir les réutiliser. On y signalait cependant que ça ne valait pas la peine de s'occuper des composants bon marché comme les résistances et condensateurs,

car ils seraient toujours disponibles sur le marché...

Certains ne sont pas affectés par la situation : dans certaines grandes villes d'Inde ou de Chine, des rues entières sont désormais consacrées à la récupération. De nombreuses personnes y œuvrent toute la journée à dessouder des CMS – y compris des résistances et condensateurs ordinaires – qui seront ensuite revendus chez le détaillant du coin comme « remis à neuf ». Qui sait, on verra peut-être le même phénomène se développer à Paris, Londres ou Munich ? Ne vous débarrassez plus de vos appareils sans avoir récupéré les composants qui peuvent l'être, et avant d'en commander des neufs, demandez-vous s'ils ne traînent pas par hasard sur un ancien circuit que vous n'utilisez plus. Nous avons ainsi bouclé la boucle, avec la même question : obsolète ou attribué ? Avec les salutations amicales du labo d'Elektor, Jan Visser

(180308-04 – version française : Jean-Louis Mehren)





vol tous azimuts

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

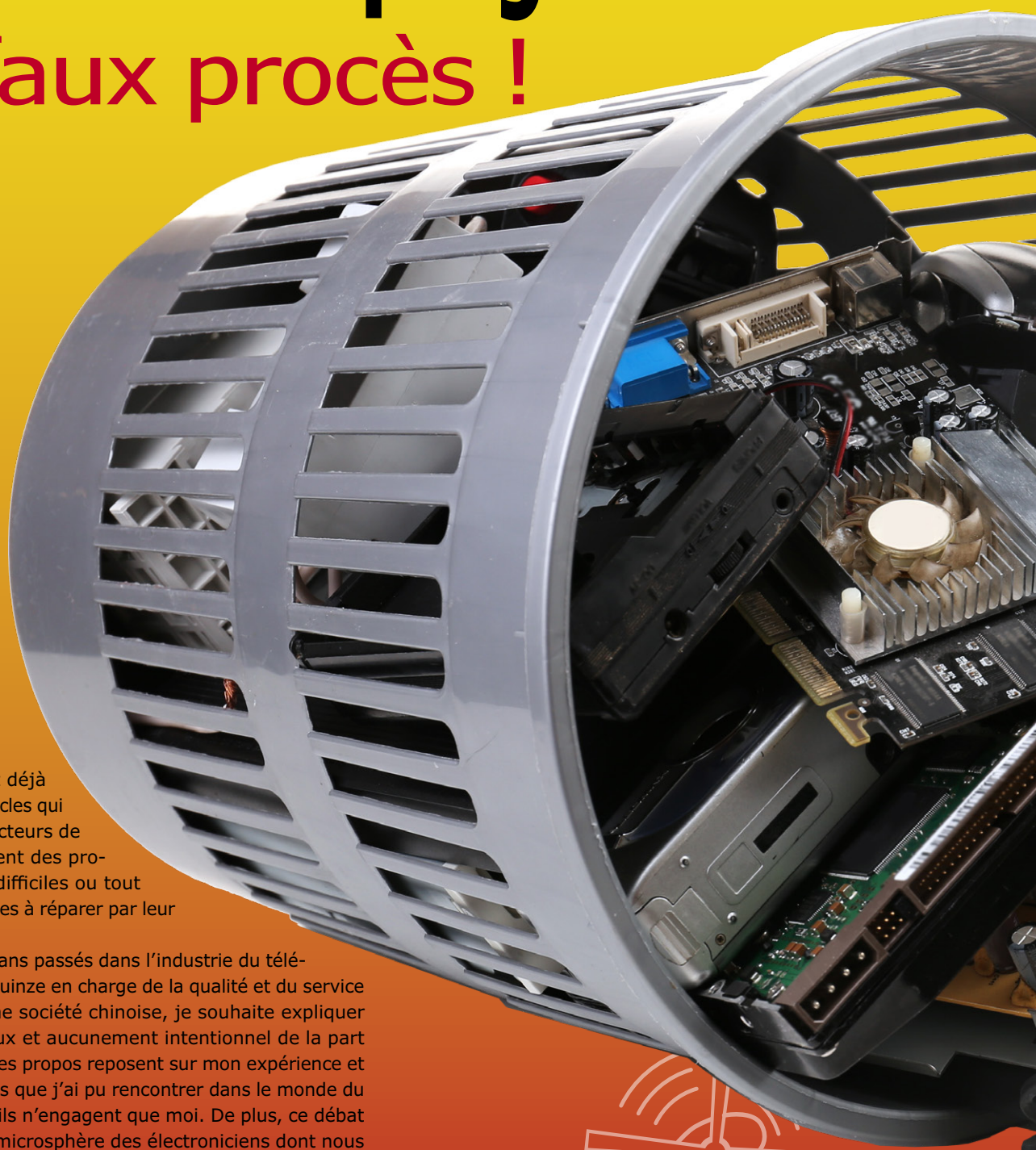
Obsolescence programmée ? Un faux procès !

Par notre envoyé spécial
Laurent Labbe

Attirer le chaland

Vous avez sûrement déjà lu un ou plusieurs articles qui accusent les constructeurs de fabriquer délibérément des produits électroniques difficiles ou tout bonnement impossibles à réparer par leur utilisateur final.

Après plus de trente ans passés dans l'industrie du téléphone mobile, dont quinze en charge de la qualité et du service après-vente dans une société chinoise, je souhaite expliquer pourquoi cela est faux et aucunement intentionnel de la part des constructeurs. Mes propos reposent sur mon expérience et celles des concurrents que j'ai pu rencontrer dans le monde du téléphone mobile — ils n'engagent que moi. De plus, ce débat n'a lieu que dans la microsphère des électroniciens dont nous





(vous) sommes puisque nous lisons cette revue.

Lors de la conception d'un téléphone mobile (ou smartphone), une équipe de projet est responsable de la conception, de la production et du service après-vente. Elle est soumise à des contraintes imposées par la réussite de l'entrée sur le marché. Pour qu'un produit soit un succès, il y a deux facteurs primordiaux :

son aspect et son prix. Il y a dix ans, tout le monde se satisfait d'un téléphone de 20 mm d'épaisseur au prix de 200 €. Aujourd'hui, l'acheteur veut un appareil deux fois moins épais et deux fois moins cher. Dans ce monde du mobile, il y a une vraie dictature du prix et de l'apparence.

L'équipe de projet a donc pour objectif de créer un produit toujours plus sexy, plus fin, sans vis apparentes, sans couvercle pour la batterie (trop épais), et surtout le moins cher possible.

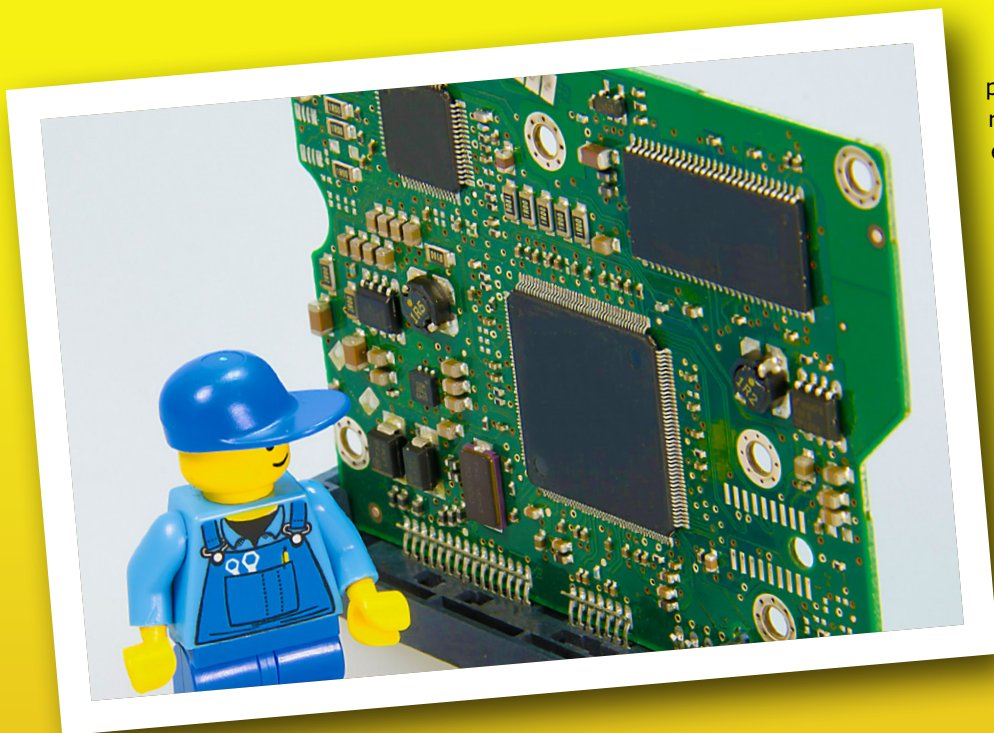
À côté de cela,

deux membres de l'équipe vont tout faire pour que le produit soit réparable : le responsable de la production et celui du service après-vente. L'usine va devoir fabriquer des dizaines de milliers d'appareils par jour et aussi réparer les défauts de production. Habituellement, sur une chaîne de production, environ 90% des produits seront bons du premier coup, mais 10% devront être réparés sur des lignes spécialisées. Enfin, le service après-vente devra gérer entre 5 et 10% des appareils vendus. Ce poste budgétaire peut « exploser » si le produit est irréparable.

L'impact du design est énorme sur la réparabilité. En effet, comment réparer un écran tactile sur un élégant appareil mince ? L'écran tactile est fixé avec de la colle ou un adhésif à double face. Il faut une table chauffante par exemple pour le décoller. Un téléphone avec une face arrière en verre est très joli, mais irréparable par un utilisateur lambda, car le verre est également collé. La même pièce en ABS sera simplement emboîtée et donc facilement remplaçable, mais le téléphone sera moins attrayant, plus épais et le client ne l'achètera pas.

Jouer la sécurité

Il y a un autre facteur à prendre en considération : il faut que les pièces « dangereuses » du téléphone (chargeur, batterie) ne soient pas faciles à démonter par un utilisateur ordinaire. Vous seriez étonné du nombre de personnes qui se croient capables de tout réparer (bien plus que ceux qui le sont réellement). Or, dans certains pays comme les États-Unis, si un utilisateur peut effectuer une opération simple avec un banal outil (comme un tournevis) et qu'il y a un incident, c'est le constructeur qui est responsable. Prenons le cas d'un téléphone avec une batterie LiPo protégée par un blindage et un couvercle que l'on peut retirer. Si le blindage est fixé avec de simples vis apparentes et que l'utilisateur peut les enlever (en dépit



peu plus cher pour avoir un appareil « plus réparable », car de toute façon son produit est sous garantie.

En résumé

Cette idée que des appareils sont délibérément conçus pour tomber en panne juste après l'expiration de leur garantie est un vieux serpent de mer qu'on lit régulièrement. Tous les constructeurs ont un plan de test type pour chaque nouveau produit. C'est un plan issu d'années d'expérience de manière à avoir un appareil qui tombera en panne le moins souvent possible, tout en restant à un prix attractif (on y revient). Pour cela, le fabricant crée des profils d'utilisateurs quotidiens, et pour son test, multiple les valeurs de référence par 1 000 (1 000 jours, soit presque 3 ans). S'il estime qu'un utilisateur

des messages d'interdiction sur le blindage), puis tenter de retirer la batterie avec ce tournevis, la batterie risque d'être percée et de prendre feu. Le responsable sera le constructeur au vu de la loi et non le @#\$\$ d'utilisateur. Des faits similaires se sont déjà produits aux États-Unis. Cela explique aussi pourquoi le boîtier de la plupart des blocs d'alimentation sur secteur est soudé par ultrasons, et non pas vissé...

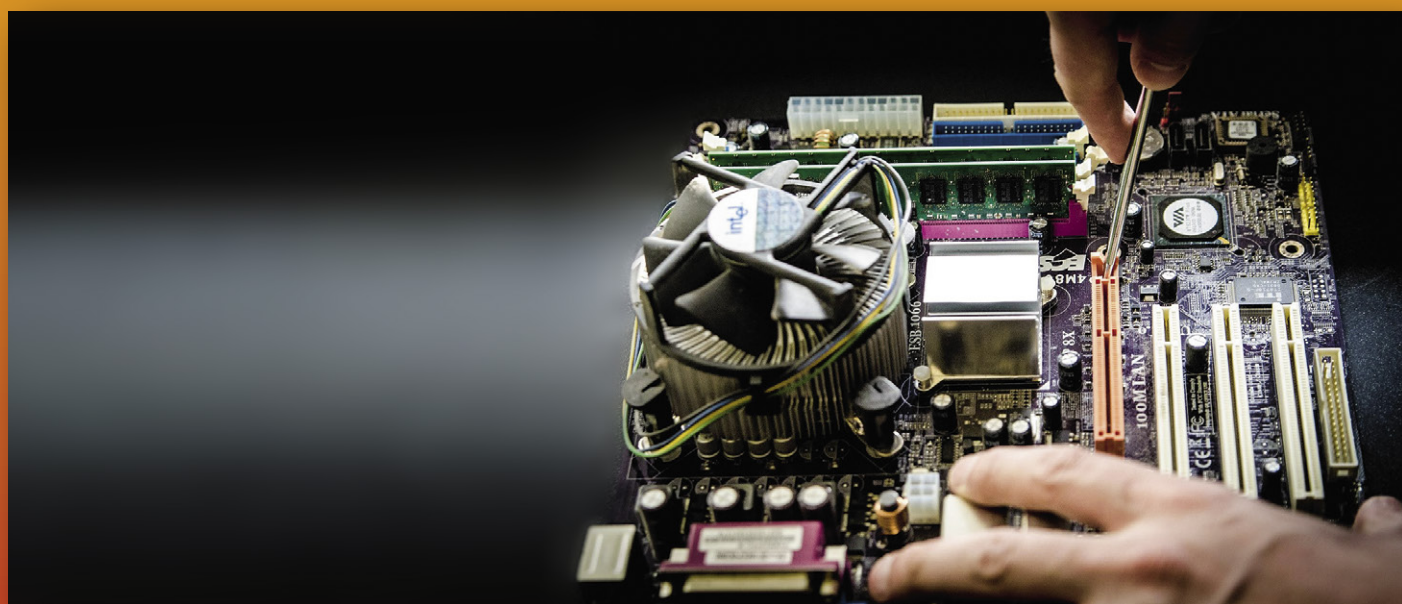
Le prix de revient et par conséquent le prix de vente (que l'acheteur souhaite toujours plus bas) ont un impact direct sur la conception de l'appareil. Par exemple, sur un produit bas de gamme, cela coûte moins cher de souder la nappe d'un afficheur plutôt que de mettre un connecteur, ou encore de souder les fils d'un écouteur au lieu de mettre des contacts à ressort. Cette guerre du « toujours moins cher » a pour conséquence directe que les appareils sont plus difficiles à démonter, donc à réparer. Et aucun utilisateur final n'acceptera de payer un

laisse tomber son téléphone d'une hauteur de

10 cm au maximum seize fois par jour, l'appareil devra donc supporter 16 000 mini-chutes. Et il fait cela pour tous les cas possibles imaginables. Au bout du compte, chaque nouvel appareil doit passer tous ces tests avant sa mise sur le marché. Bien sûr, on pourrait utiliser 3 650 jours (10 ans) comme base de calcul. Dans ce cas, les matériaux et les composants devraient être plus résistants, donc plus chers, d'où un prix de vente plus élevé que le client ne serait pas prêt à payer.

Conclusion

Il est certain que la difficulté à réparer certains appareils est une conséquence directe de l'appétit des consommateurs pour des produits au design liché et que ce n'est aucunement une volonté délibérée des constructeurs. Je suis depuis trente ans dans cette industrie et j'ai participé au développement de centaines de téléphones, jamais il n'a été question de fabriquer un appareil irréparable pour que le client doive en acheter un autre plus vite.



Des pièces détachées coûteuses

Oui, elles sont souvent chères et la réparation d'un produit après la période de garantie est onéreuse. Pourquoi ? Pour cela, il faut regarder la loi, les contraintes industrielles et le marché global. La loi impose la réparation dans le cadre de la garantie pendant une période d'un ou deux ans (suivant les pays). Elle oblige aussi à fournir des pièces de rechange pendant trois ans après la dernière vente. Le constructeur doit donc entretenir un stock de pièces pour le service après-vente souvent pendant la production, lequel sera donc rapidement épuisé. Il faut savoir que la plupart des pièces de cette industrie ont un cycle de vie très court essentiellement à cause de l'évolution de la technologie, il est souvent impossible d'en recommander après coup. Si le constructeur ne peut pas réparer un produit

qui est sous garantie, il doit l'échanger contre un produit neuf, c'est une perte sèche. Il a donc tout intérêt à garder ses pièces détachées pour toutes les réparations sous garantie, les réparations hors garanties étant forcément moins prioritaires et donc chères. Enfin, des pièces de rechange trop bon marché engendrent un marché gris et des trafics sur les pièces entre pays et continents.



Outils de labo indispensables

Voilà deux câbles USB ultra pratique. D'abord il y a le câble USB avec interrupteur intégré. Plus besoin de tirer sur le câble pour éteindre ou redémarrer vos cartes ou machins, il suffit d'appuyer sur le bouton. Fini l'usure du connecteur USB et du câble due aux manipulations trop fréquentes. Ensuite il y a le câble USB à embout extra long (8 mm). Ce câble est indispensable dans les situations où il n'est pas possible de placer le connecteur micro-USB assez près de la paroi du boîtier ou quand l'ouverture n'est pas assez large pour la fiche.

www.elektor.fr/usb-a-to-micro-usb-b-cable-with-switch

www.elektor.fr/8mm-micro-usb-connector-white-2m



180567-01

Vous voulez participer ? Envoyez vos commentaires, suggestions, trucs et astuces à labs@elektor.com



trucs et astuces

par les lecteurs pour les lecteurs

Encore une solution astucieuse qui facilite la vie des électroniciens.



adaptateur pour tester des CMS

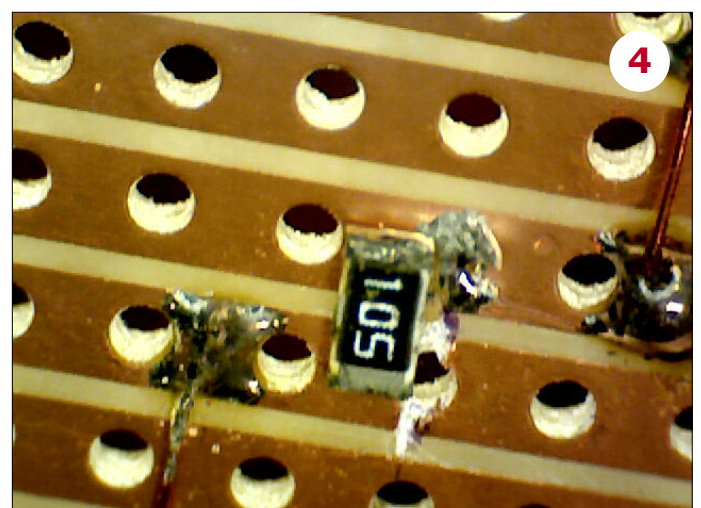
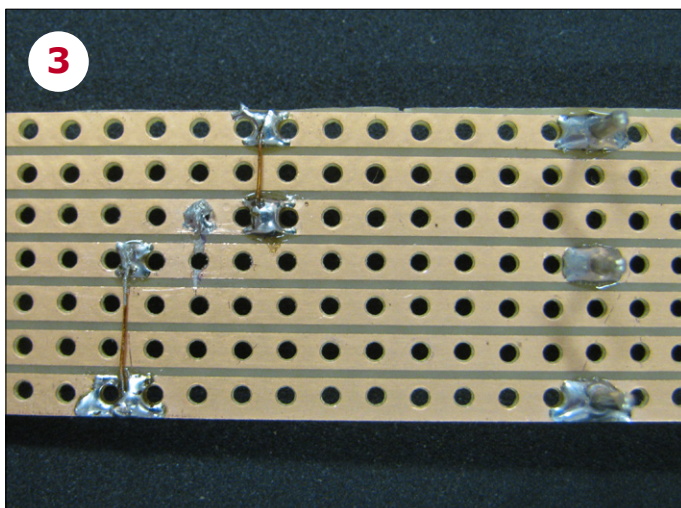
Karl-Ludwig Butte (Allemagne)

Avez-vous déjà essayé de mesurer une résistance CMS ou un transistor CMS à l'ohmmètre ? Vous avez sans doute vécu la même mésaventure que moi : la résistance a sauté comme une puce de la paille pour disparaître définitivement.



Prenons une épingle à cheveux (**fig. 1**), un morceau de carte à trous, trois broches à souder et un peu de fil de cuivre émaillé et, en quelques opérations, nous en ferons un adaptateur pour test de CMS petit mais très utile.

Pour commencer, glissez l'épingle à cheveux par-dessus la carte à trous dans le sens des pistes conductrices. Au premier trou à l'extrémité de l'épingle à cheveux, coupez la piste conductrice au moyen d'un tournevis d'horloger ou équivalent (**fig. 2**). Ensuite soudez deux ponts de fil de cuivre émaillé ainsi que les trois broches comme indiqué sur la **figure 3**. Remplissez le trou au-dessus de la coupure de la piste avec un peu de soudure. Les **figures 4 et 5** montrent comment positionner ultérieurement une résistance et un transistor. Mais pour l'instant, il faut encore monter l'épingle à cheveux. Glissez-la sur la carte à trous (**fig. 6**) et veillez à ce que son extrémité appuie



Vous avez trouvé une solution astucieuse pour un cas tordu ? Vous utilisez un composant ou un outil d'une manière inhabituelle ? Vous avez une idée pour résoudre un problème d'une manière plus simple ou meilleure que précédemment ? Écrivez-nous. Pour chaque astuce que nous publions, il y a 40 € à gagner.

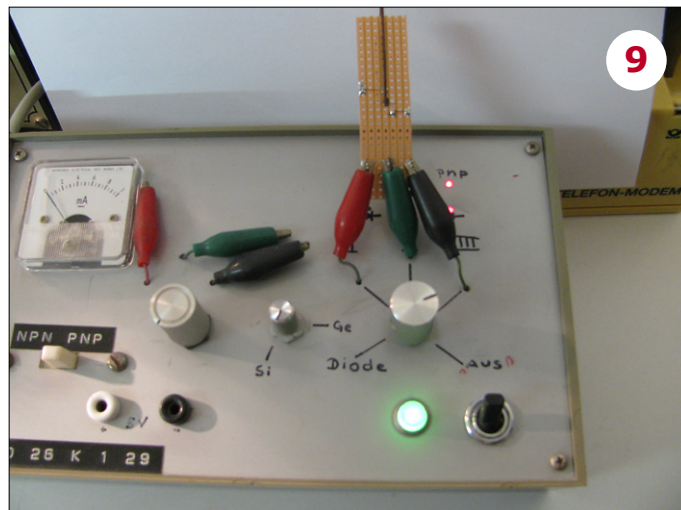
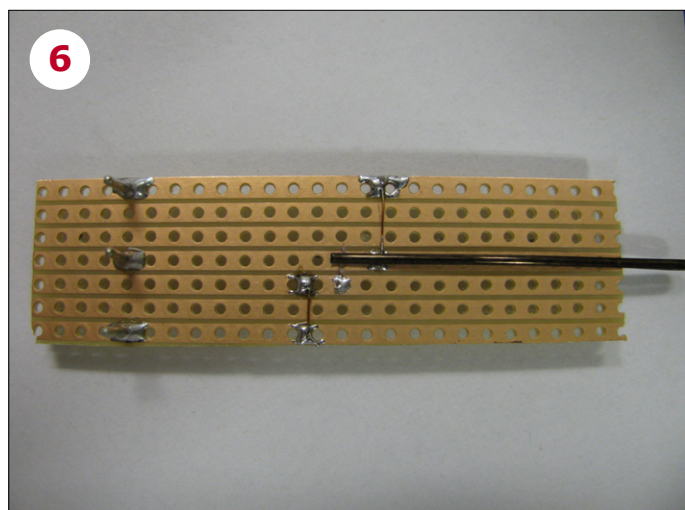
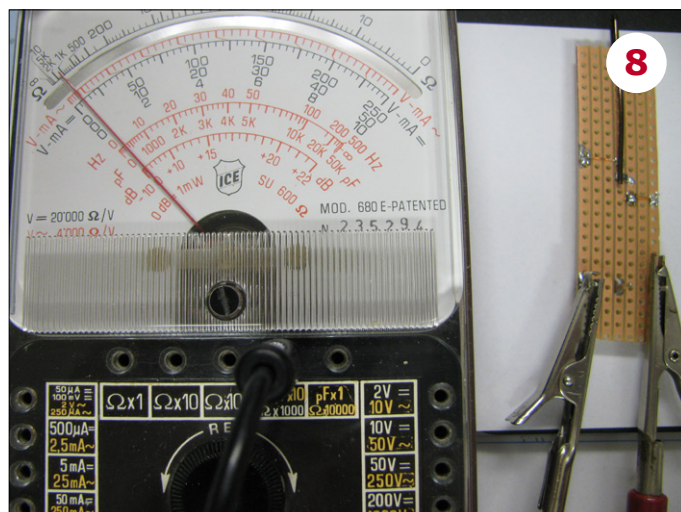
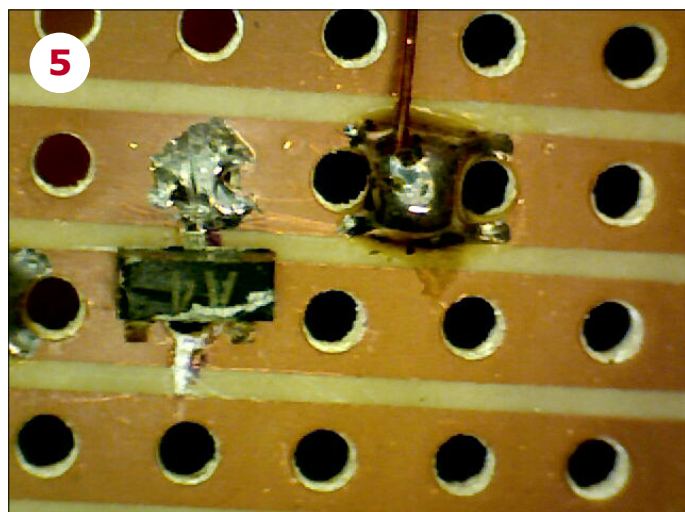
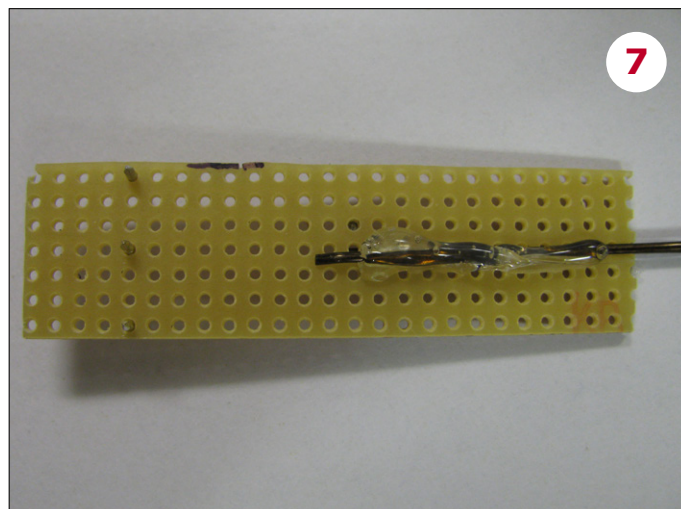


sur le trou avec la coupure de la piste. Retournez la carte et collez l'épingle à cheveux au pistolet à colle (**fig. 7**). À cause des contraintes mécaniques, l'épingle a tendance à s'écarter de la carte, il faut donc la maintenir au contact au moyen d'une pince adéquate jusqu'à ce que la colle durcisse.

On peut alors passer aux essais. Placez une résistance CMS sous l'épingle à cheveux et connectez un ohmmètre aux broches à souder correspondantes (**fig. 8**). Sur l'échelle supérieure à gauche, on lit une valeur de 1 k, ce qui, pour la gamme de mesure choisie, correspond à 1 M Ω . L'inscription sur la résistance doit donc être lue « 105 » et non, comme on pourrait le supposer, « 501 », ce qui signifierait 500 Ω . Le fait de pouvoir maintenant le vérifier est donc très utile.

Il reste le test d'un transistor. Glissez un transistor CMS sous l'épingle à cheveux (**fig. 5**) et connectez un testeur de transistors (**fig. 9**). L'appareil indique que a) le transistor fonctionne correctement, b) qu'il s'agit d'un type PNP, et c) que sa base est connectée à la pince crocodile noire.

(170569-04 – version française : Helmut Müller)





shield anti-ronflement

pour des nuits enfin reposantes

Clemens Valens (labo d'Elektor)

Beaucoup de gens ronflent et beaucoup n'y font pas attention ou semblent ne pas remarquer le bruit qu'ils font. Malheureusement, une autre bonne partie de l'humanité ne peut pas dormir à cause des ronflements de leur compagnon de lit et/ou colocataire. Ce projet est pour eux.

Stress dû aux ronflements

Si vous essayez de vous endormir dans la même pièce qu'un ronfleur, vous vous focalisez sur ses ronflements et leur anticipation, vous finissez stressé et toujours réveillé. Donner des coups de pied, des gifles, ou bien pousser ou piquer le ronfleur ne sont d'aucune aide et généralement augmentent le niveau de stress, rendant encore plus difficile l'endormissement.

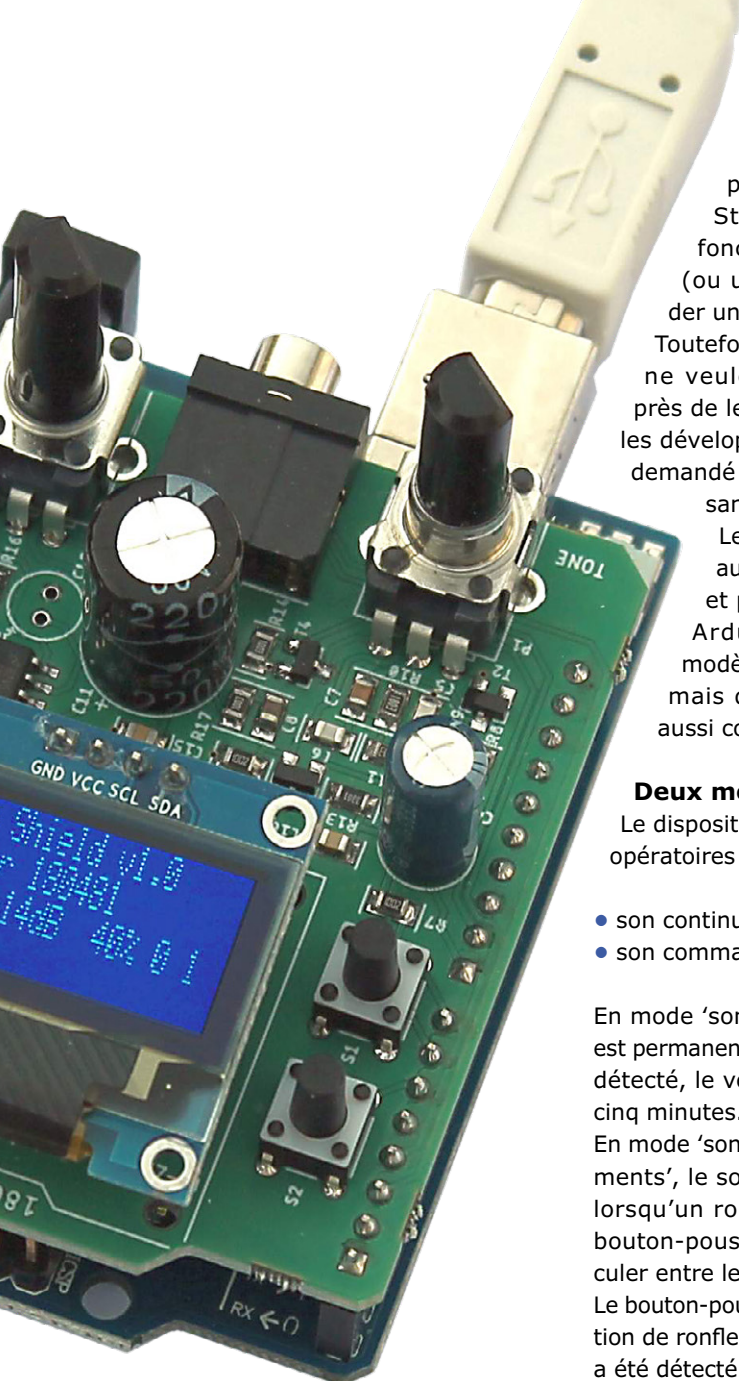
Découvrez le shield anti-ronflement

Le montage présenté ici produit un son agréable sur lequel le « non-ronfleur » va se concentrer pour, espérons-le, se détendre. Le dispositif « écoute » aussi les ronflements. Lorsqu'il en détecte, il augmente le volume du son relaxant pour couvrir les ronflements. Après cinq minutes – à priori suffisantes pour

que le ronfleur s'arrête ou pour que le non-ronfleur s'endorme – le niveau sonore décroît, ce qui termine un cycle et la détection de ronflement reprend.

Un son qui relaxe

La plupart des gens considèrent le bruit de la mer, un calme ressac ou le vent au travers des arbres comme des sons qui relaxent. Le *shield* anti-ronflement



Le *shield* anti-ronflement transforme n'importe quelle chambre à coucher en un labo du sommeil de haute technologie (très différent du labo d'Elektor).

(fig. 1) produit donc un bruit blanc (avec une coloration réglable) qui tend à se rapprocher de ces sons apaisants. Ceux qui préfèrent d'autres sons, comme de l'eau qui coule ou de la musique douce peuvent connecter un module « lecteur de MP3 ».

Issu d'une appli Android

Le *shield* anti-ronflement est issu de l'appli Android « SleepSation » (sic : non, il

n'y a pas de coquille ; disponible sur le Google Play Store). Comme cette appli fonctionne sur un ordiphone (ou une tablette), il faut garder un téléphone près de son lit. Toutefois beaucoup de dormeurs ne veulent pas d'un téléphone près de leur oreiller, c'est pourquoi les développeurs de l'appli nous ont demandé de concevoir un dispositif sans ordiphone.

Le *shield* anti-ronflement est au format des *shields* Arduino et prend place sur une carte Arduino, de préférence un modèle Uno (pour rester petit), mais d'autres types devraient aussi convenir.

Deux modes

Le dispositif a deux modes opératoires :

- son continu ;
- son commandé par les ronflements.

En mode 'son continu', le son relaxant est permanent. Lorsqu'un ronflement est détecté, le volume augmente pendant cinq minutes.

En mode 'son commandé par les ronflements', le son relaxant n'est actif que lorsqu'un ronflement est détecté. Le bouton-poussoir SW2 permet de basculer entre les deux modes.

Le bouton-poussoir SW1 annule la détection de ronflement lorsqu'un ronflement a été détecté et le volume revient à son niveau par défaut (muet ou bas, selon le mode opératoire). Le presser au repos forcera une détection de ronflement, ce qui provoquera l'augmentation du volume au niveau de couverture des ronflements.

Détection du ronflement

Le *shield* anti-ronflement surveille en permanence le niveau sonore ambiant. Dans des conditions normales de sommeil, ce niveau doit être bas. Le ronflement augmente le niveau sonore moyen, mais au même titre que d'autres sources de bruit dans la chambre (changements de position) ou à l'extérieur (circulation automobile). Détecter le ronflement est donc un peu plus compliqué que la simple comparaison du niveau sonore actuel avec une moyenne glissante.

Le *shield* anti-ronflement a besoin de trois ronflements consécutifs, où un ronflement est défini comme un bruit fort

INFOS SUR LE PROJET

santé bien-être

Arduino

débutant

connaissseur

expert

env. 2h

outils de soudage CMS, EDI Arduino

25 €

et court (jusqu'à environ trois secondes) suivi d'une période de silence de cinq à dix secondes. Bien sûr, il peut y avoir de fausses détections et des sons de caractéristiques similaires peuvent aussi déclencher le dispositif, comme trois toussotements lents consécutifs.

Détails sur

le fonctionnement interne

La figure 2 donne le schéma du *shield* anti-ronflement. La détection de ronflement démarre avec un microphone, MIC1, connecté à un amplificateur avec une bande passante de 100 Hz à 3 kHz.

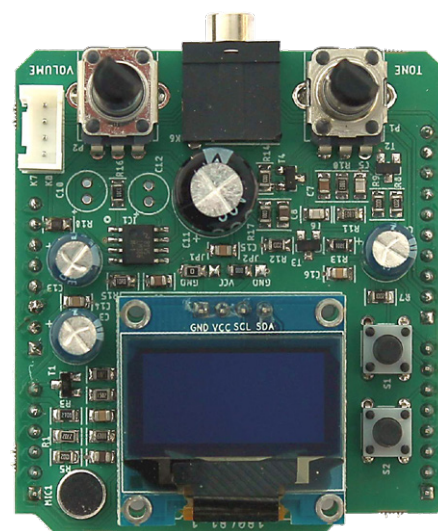


Figure 1. Le *shield* anti-ronflement est une carte d'extension compatible avec l'Arduino Uno.

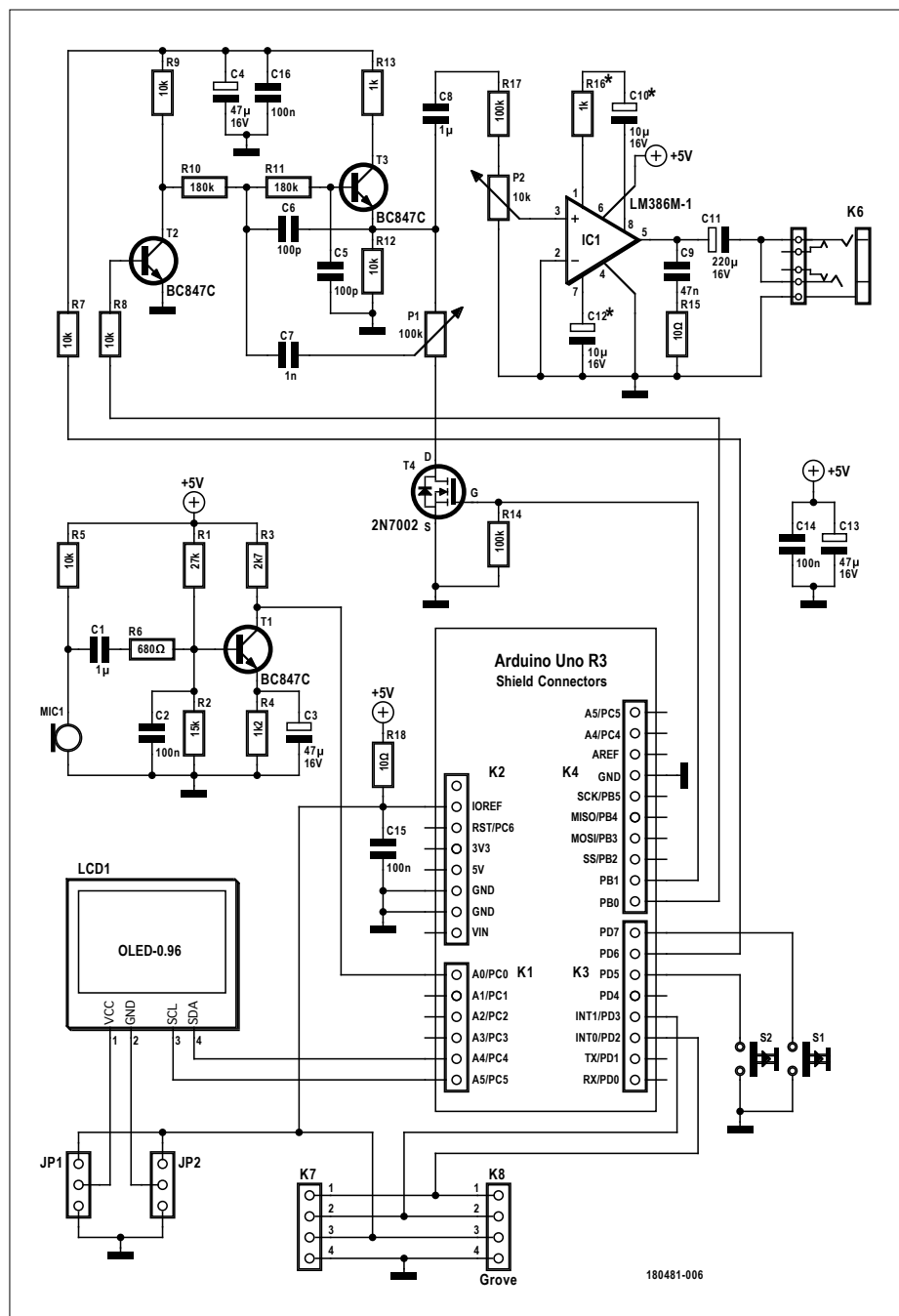


Figure 2. Le schéma du *shield* anti-ronflement est constitué principalement de l'amplificateur de microphone T1, du processeur de bruit commandé en tension T2-T3 et de l'amplificateur de puissance IC1.

La sortie de l'amplificateur est connectée à l'entrée analogique A0 de l'ATmega328 de l'Arduino qui prend en charge la numérisation et le traitement numérique. À l'intérieur de l'unité centrale (MCU), le signal audio est échantillonné à une cadence de 7 kHz. Cette fréquence est supérieure à deux fois la bande passante en entrée, toutefois la pente de l'amplificateur n'est pas très raide, ce qui rend assez difficile l'analyse précise de la fréquence du signal audio. Ce n'est pas vraiment un problème ici, car nous ne

nous intéressons qu'au volume sonore. Pour couvrir les bruits de ronflement, le système produit un bruit blanc. Le logiciel s'en charge avec un registre à décalage à contre-réaction linéaire (LFSR), qui repose sur un registre à décalage de 32 bits et un taux d'échantillonnage de 20 kHz. Le bruit numérique est émis sur la broche D8 de l'Arduino. Le transistor T2 fonctionne en amplificateur commandé en tension pour permettre la modulation d'amplitude du signal de bruit. La tension de commande est pro-

Caractéristiques

- Deux modes de détection des ronflements
- Microphone à électret
- Sortie pour écouteurs
- Écran graphique OLED I²C
- Deux boutons-poussoirs
- Potentiomètre de volume
- Commande de la couleur du bruit par l'utilisateur
- Compatible avec le module « lecteur de MP3 » Grove
- Alimenté par adaptateur de 9 à 12 VDC ou batterie externe USB de 5 V

duite par un signal MLI après filtrage passe-bas, disponible sur la broche D6 de l'Arduino. Le circuit autour du transistor T3 est un filtre pour ajuster la « couleur » du bruit. Pour rester simple, la contre-réaction du filtre est commandée par un classique potentiomètre. Le logiciel peut activer ou désactiver le transistor T4 pour sélectionner deux modes de contre-réaction et donc deux types de bruit. La sortie de l'amplificateur de bruit est connectée à un banal amplificateur de puissance basé sur un LM386 avec commande manuelle du volume. Cet amplificateur peut attaquer des écouteurs ou de petits haut-parleurs.

Lecteur MP3 pour sons personnalisés

Le module « lecteur de MP3 » utilisé est un module Grove de Seeedstudio (**fig. 3**) connecté sur K7. Pour le piloter, un port série est émulé par logiciel sur les broches D2 et D3 de l'Arduino. Si vous vous sentez capable de modifier le code, vous pouvez utiliser d'autres types de modules « lecteur de MP3 » avec des principes de commande similaires. Notez que le connecteur

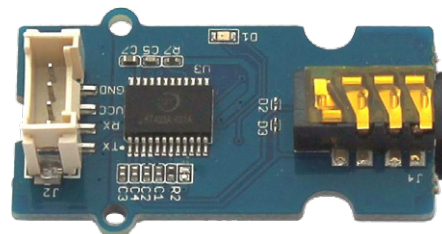


Figure 3. Il est possible de connecter un module « lecteur de MP3 » pour améliorer la qualité du son et faire jouer des sons personnalisés au *shield* anti-ronflement.

Choisissez un haut-parleur de bonne qualité

Il est important que le son apaisant relaxe, donc assurez-vous de connecter un système audio approprié. Un haut-parleur minable produira un son de piètre qualité (plus agaçant que soporifique). Veuillez lire avec attention l'encadré sur l'alimentation. Des commandes de volume et de couleur du bruit sont disponibles pour régler le son.



Figure 4. Pour l'instant, l'écran OLED du *shield* anti-ronflement ne donne que des informations pour le débogage (sans intérêt quand on tente de dormir). Ici, la valeur 298 sur la 4^e ligne est le nombre de secondes à attendre avant le retour au mode continu (le mode anti-ronflement a été déclenché manuellement).

Grove K8 est d'un type particulier avec un pas de 2 mm. Le connecteur K7 offre ici une alternative au pas de 2,54 mm. Malheureusement, le port série logiciel a un impact négatif sur le signal de bruit numérique, car il masque les interruptions pendant les émissions, ce qui introduit des parasites audibles dans le signal de bruit lui aussi commandé par interruption. Si vous n'utilisez pas le module MP3, vous pouvez placer en commentaire (ajouter //) une certaine ligne du programme pour supprimer ces parasites. Un commentaire dans le code indique la ligne concernée.

Écran

L'écran OLED est d'un modèle graphique répandu, de 2,4 cm et 64 × 128 pixels. Ces écrans existent en plusieurs versions,

avec six ou quatre broches, et différents brochages : nous avons choisi le type à quatre broches. Ce type se décline en deux versions : broche 1 = GND, broche 2 = VCC ou l'inverse. Choisissez l'une ou l'autre configuration avec les cavaliers à souder JP1 et JP2.

Un écran c'est pratique pour interagir avec l'utilisateur, mais pour l'instant ce n'est pas très utile. On l'a ajouté parce que l'appli SoundSation a beaucoup d'options de temporisation et d'enregistrement pas (encore ?) implémentées sur le *shield* anti-ronflement.

Sur la première ligne (**fig. 4**), on voit la désignation du dispositif en anglais (*Snore Shield*) et la version du micrologiciel. La deuxième ligne contient le numéro de projet Elektor.

La troisième ligne montre des statistiques sur le son (seuil de détection et niveau sonore ambiant en dB) et le volume (en pourcentage), puis le mode (« 0 » pour le son continu ou « 1 » pour le son com-

Trouvez Elektor en kiosque

avec votre code postal...

Où trouver Elektor près de chez vous ?



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

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

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

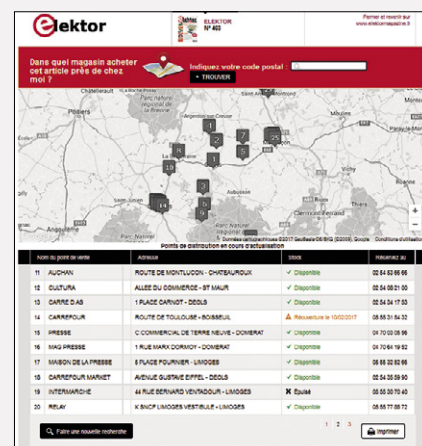
Vous verrez apparaître une carte avec des repères chiffrés qui renvoient chacun à un des points de vente énumérés. Vous connaîtrez aussitôt l'état du stock chez ces différents revendeurs : magazine disponible ou épuisé.

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

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

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

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



mandé par les ronflements) et enfin le type de bruit « 0 » ou « 1 ».

Pour finir, la quatrième ligne montre l'état de la détection de ronflement avec une série de nombres croissants et décroissants. Lorsque le premier nombre arrive à trois, le système entre en mode 'son de couverture du ronflement'. La détection de ronflement s'arrête et ces nombres se figent pendant cinq minutes. Le niveau du volume sur la ligne trois va augmenter doucement vers un pourcentage plus élevé.

Logiciel

Le logiciel nécessaire pour le fonctionnement du *shield* anti-ronflement est à télécharger en [1]. Nous vous invitons, et même encourageons, à l'étudier. Pouvez-vous améliorer l'algorithme de détection des ronflements, ou bien voulez-vous personnaliser les volumes sonores ? Le croquis Arduino complété par des bibliothèques tient dans un seul fichier, et peut être programmé tel quel dans la carte sans outillage spécial. Vous n'avez besoin que d'un EDI Arduino récent (nous avons utilisé la version 1.8.0).

N'hésitez pas à partager vos expériences avec le monde des électroniciens en les publiant sur [1]. ◀

(180481-04

version française : Denis Lafourcade)

Un mot sur les alimentations

Une application Arduino autonome typique serait alimentée par un chargeur de téléphone. Malheureusement, ces chargeurs sont très bruyants électriquement, ce qui nuit à la qualité du bruit relaxant produit par le *shield* anti-ronflement. C'est pourquoi nous vous recommandons d'alimenter le dispositif avec un bon vieux adaptateur mural de 9 VDC (12 VDC maxi) ; une batterie USB externe fera aussi bien l'affaire. En conclusion : n'utilisez pas un chargeur de téléphone ni un port USB de PC.



LISTE DES COMPOSANTS

Résistances

Par défaut : CMS 0805
 JP1, JP2 = 0 Ω
 R15, R18 = 10 Ω
 R6 = 680 Ω
 R13, R16 = 1 k Ω
 R4 = 1,2 k Ω
 R3 = 2,7 k Ω
 R5, R7, R8, R9, R12 = 10 k Ω
 R2 = 15 k Ω
 R1 = 27 k Ω
 R14, R17 = 100 k Ω
 R10, R11 = 180 k Ω
 P2 = 10 k Ω , potentiomètre, vertical
 P1 = 100 k Ω , potentiomètre, vertical

Condensateurs

Par défaut : CMS 0805
 C5, C6 = 100 pF
 C7 = 1 nF
 C9 = 47 nF
 C2, C14, C15, C16 = 100 nF

C1, C8 = 1 μ F

C10, C12 = 10 μ F 16 V, au pas de 2 mm

C3, C4, C13 = 47 μ F 16 V, au pas de 2,5 mm

C11 = 220 μ F 16 V, au pas de 5 mm

Semi-conducteurs

T4 = 2N7002

T1, T2, T3 = BC847C

IC1 = LM386M-1/NOPB

Divers

MIC1 = microphone à électret

K6 = connecteur jack, 3,5 mm

S1, S2 = bouton-poussoir, 6x6 mm

K1 = barrette mâle à 6 picots,

au pas de 2,54 mm

K2, K3 = barrette mâle à 8 picots,

au pas de 2,54 mm

K4 = barrette mâle à 10 picots,

au pas de 2,54 mm

K5 = embase femelle à 6 pôles,

au pas de 2,54 mm

K7 = embase femelle à 4 pôles,

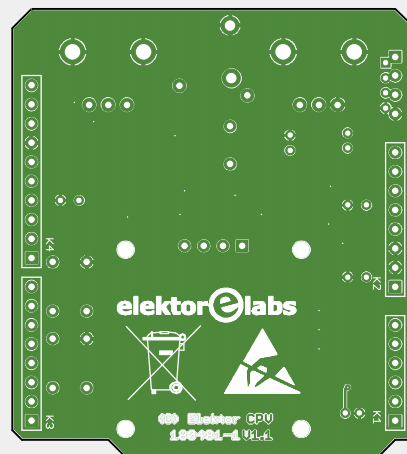
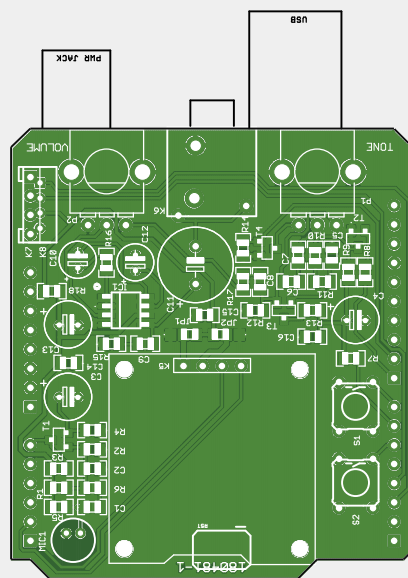
au pas de 2,54 mm

K8 = embase femelle à 4 pôles,

au pas de 2 mm

MOD1 = écran OLED, 2,4 cm, commandé par I²C

Circuit imprimé, réf. 180481-1, www.elektor.fr



@ WWW.ELEKTOR.FR

→ Circuit imprimé nu, réf. 180481-1 :
www.elektor.fr/snore-shield-180481-1

→ Lecteur audio compact programmable Grove MP3 v2, réf. 18668 :
www.elektor.fr/grove-mp3-v2

→ Connecteur universel Grove à 4 broches (au pas de 2 mm), réf. 18671 :
www.elektor.fr/10-x-4-pin-2-mm-pitch

Lien

[1] Page du projet au labo d'Elektor : www.elektormagazine.fr/labs/snore-shield

speed-o-mètre pour coureurs

avec totalisateur et enregistreur

Bera Somnath (Inde)

Pour garder une trace du chemin parcouru et calculer ma vitesse moyenne pendant mon jogging, j'ai associé un Arduino Nano à un récepteur GPS simple et un écran OLED. Avec sa structure modulaire, le circuit tient bien en main ; avec un effort de plus, je pourrais même en faire un porte-clés.



De bon matin, je pratique souvent le jogging et j'ai essayé tout un temps de mesurer avec une certaine précision ma vitesse. Au début, j'ai tenté de le faire au moyen d'un capteur Doppler à haute fréquence, mais sans réussir à mesu-

rer ma propre vitesse. Tout ce que j'ai pu en obtenir, c'est la vitesse relative au moment de longer un objet volumineux. Alors, j'ai renouvelé l'expérience avec un capteur de pression atmosphérique, de température et d'humidité, un

BME280. Je supposais qu'un changement de vitesse influencerait la pression de l'air dans le sens horizontal et qu'un capteur monté perpendiculairement pourrait la mesurer. Eh bien non ! De toute façon, la différence de pression était bien trop petite pour que le BME280 puisse la mesurer.

Finalement, j'ai pris un récepteur GPS [1] couplé à un Arduino Uno, avec lequel j'ai obtenu une précision de vitesse suffisante. Avantage supplémentaire, le GPS donne en plus de la vitesse, l'heure exacte et le chemin parcouru au cours de la randonnée. Ces informations me permettent de déterminer le nombre total des kilomètres couverts et la vitesse moyenne de la course. Un bel écran OLED complète le dispositif.

Philosophie de la mesure - V1

Un brin de réflexion : vous partez courir puis vous faites demi-tour pour rentrer à la maison. Pour un GPS, la distance parcourue, fonction de la longitude et de la latitude du chemin parcouru, sera nulle à l'arrivée, pourtant, vous l'aurez deux fois dans les jambes. Ma solution : j'ai ajouté un bouton-poussoir ('P', S3) sur la ligne numérique 6 du microcontrôleur (cf. **fig. 1**). Vous devez appuyer dessus au moment de partir et chaque fois que le trajet change de direction, donc s'il

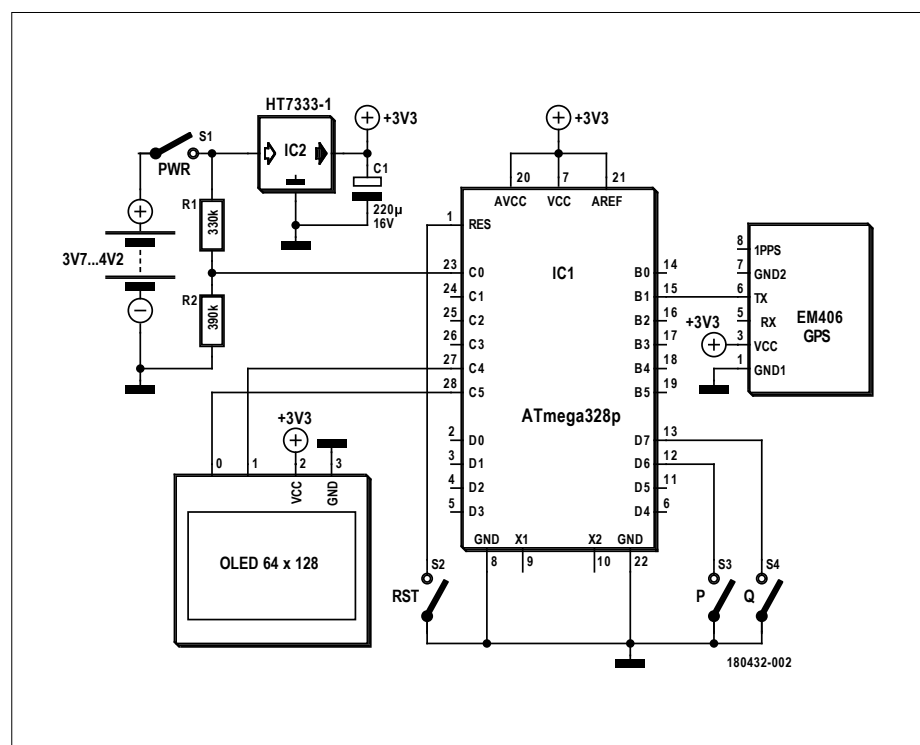


Figure 1. Grâce au recours à des modules, on ne trouve plus que sept composants sur le schéma, tous interconnectés par l'ATmega328P.

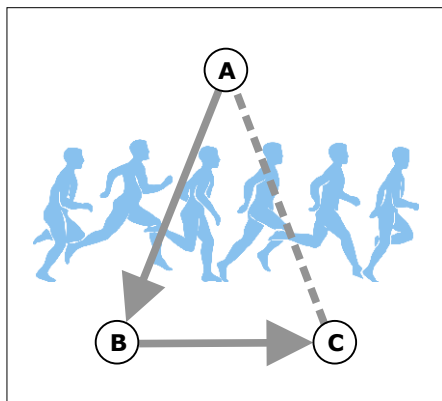


Figure 2. Un joggeur qui court de A vers B puis C et revient en A n'a finalement pas changé de position géographique, mais il a quand même parcouru une certaine distance.

n'est plus en ligne droite.

Comme sur la **figure 2**, disons que vous partez du point A, vers B, puis vers C, pour enfin revenir au point A. En physique, on dit que le travail est nul, et pourtant ! Pour mesurer le déplacement total, commencez par pousser sur le bouton P au point A pour fixer le début du chemin. Poussez-le encore au point B, puis au point C et enfin à l'arrivée au point A. Le trajet complet est donc $AB + BC + CA$; ces segments ainsi que leur durée totale (en secondes) seront enregistrés dans l'EEPROM de l'ATmega328, qui dispose pour cela de 1 024 octets.

Affichage

Quand les boutons-poussoirs P et Q sont au repos, comme à la **figure 3**, l'écran indique, du haut à gauche vers le bas à droite, la latitude, la longitude, le nombre de satellites que le module GPS reçoit, l'heure courante et la vitesse instantanée en km/h. Quand on pousse sur 'P' pour la première fois, il affiche la latitude, la longitude, le nombre de satellites en vue, l'heure courante, la vitesse actuelle et « I/L:AB/AB 00 00 » avec le **premier AB** pour le nombre de mètres du segment actuellement parcouru, le **second AB** pour le nombre total de mètres de trajet couvert. Le **premier 00** correspond au nombre de segments du trajet ; le **second 00** au nombre de secondes écoulées depuis le départ.

Si vous poussez sur 'P' une deuxième fois sont affichés la latitude, la longitude, le nombre de satellites en vue, l'heure actuelle et la vitesse instantanée et par ex. « I/L:CA/(AB+BC+CA) 03 1950 »,

Et pourquoi pas un ordiphone ?

Ma fille m'a demandé pourquoi je n'avais pas utilisé pour ce projet un ordiphone – une demande légitime aux yeux de la nouvelle génération. C'est que j'ai bien des raisons de m'en passer.

- Je n'aime pas emporter mon téléphone quand je fais du jogging. Il me distrairait.
- Sur le thème du jogging, on trouve plus de 30 applis entre 7 et 181 Mo. Les arbres y cachent la forêt.
- Dans certaines de ces applis, vous devez introduire des données personnelles : longueur du pas, masse corporelle, taille, etc. Chaque jour, je cours de 6 à 10 km/h et je me relaxe après. Je n'ai que faire de toutes ces autres possibilités.
- Presque toutes ces applis nécessitent une liaison internet active.
- J'ai là un petit appareil qui tient aisément en main. On peut même en faire un porte-clefs en remplaçant le Nano par un simple ATmega328 en CMS.
- Un ordiphone coûte au moins 150 €, mon circuit est bien moins cher !

où CA et $(AB+BC+CA)$ sont exprimés en mètres, 03 est le nombre de segments et 1950 le nombre total de secondes écoulées depuis le début de la mesure. Avec ces informations, nous pouvons par ex. lire sur la **figure 4** que 12 m ont été couverts dans le segment en cours, que 25 m ont été parcourus sur cinq segments différents et que l'enregistrement a démarré 62 s plus tôt.

Si vous poussez sur le bouton 'Q' (S4), vous verrez (**fig. 5**) la durée totale en secondes depuis le début de la mesure, la distance parcourue en mètres, l'heure actuelle et la vitesse moyenne sur tout le

parcours. Vous pouvez l'actionner autant de fois que vous voulez pour afficher ces données, elles restent en mémoire même après avoir coupé le courant avec S1. Cependant, dès qu'on pousse sur le bouton 'P', les données sont actualisées.

Adaptations - V2

Certains de mes amis qui utilisent aussi le montage pendant leur entraînement quotidien m'ont demandé des modifications, que j'ai bien volontiers réalisées pour eux. C'est cette nouvelle version qui est représentée sur le schéma de la figure 1. Lorsque l'écran par défaut

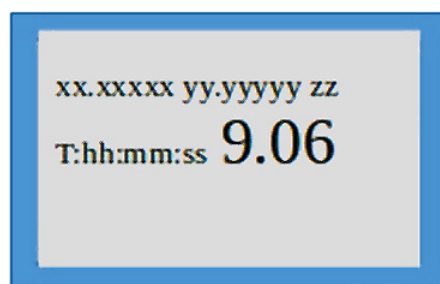


Figure 3. Écran par défaut.

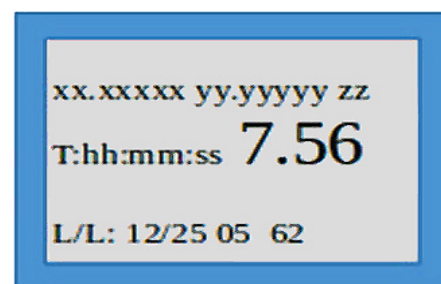


Figure 4. Avec 'P' actionné.

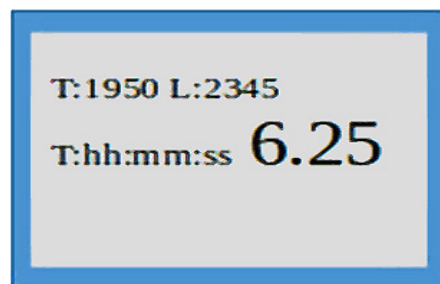


Figure 5. Sous l'action de 'Q'.

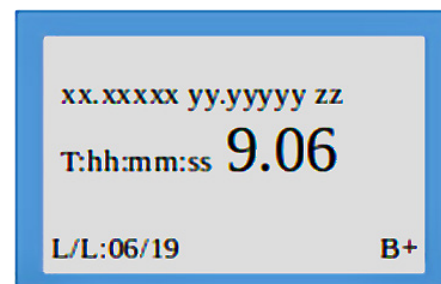


Figure 6. Avec 'P' actionné dans la version modifiée.

est affiché (fig. 4), une pression sur 'P' provoque une remise à zéro du compteur, qui redémarre dès qu'il y a mouvement. J'ai aussi ajouté sur la dernière ligne de l'écran, côté droit, une indication simple de l'état de l'accumulateur. Tant que la tension de l'élément LiPo est supérieure à 3,3 V, on y voit B+, comme à la **figure 6** ; en dessous de cette valeur, c'est B- qui s'affiche, signe qu'il est temps de recharger.

Pour surveiller cette tension, j'ai ajouté le diviseur de tension R1/R2. J'ai aussi interposé le stabilisateur de tension HT7333-1 qui trouve bien sa place ici en raison de sa faible chute de tension et de sa consommation minime au repos.

Construction et essais

L'assemblage est une des parties les plus simples du projet. On part d'un petit morceau de circuit imprimé à double face pour y installer l'Arduino, l'écran OLED et le récepteur GPS en face avant. C'est là aussi que doit se trouver l'antenne du module GPS pour qu'elle ne soit pas cachée par la carte et puisse capter à coup sûr les signaux des satellites GPS. On peut fixer l'accu LiPo à l'arrière du circuit imprimé. La **figure 7** montre mon circuit assemblé. N'oubliez pas de programmer l'Arduino avec le logiciel disponible sur [2].

J'éprouve toujours beaucoup de plaisir à tester mes propres réalisations. Il faut commencer par appliquer une tension correcte au circuit, avec un accu LiPo ou Li-ion, ensuite fermer S1 et S2 (avec une tension plus élevée, comme 9 ou 12 V, S2 peut sans doute rester ouvert). On emporte alors l'appareil pour son jogging, éventuellement en groupe pour épater la galerie avec son nouveau gadget ! Dès que le récepteur

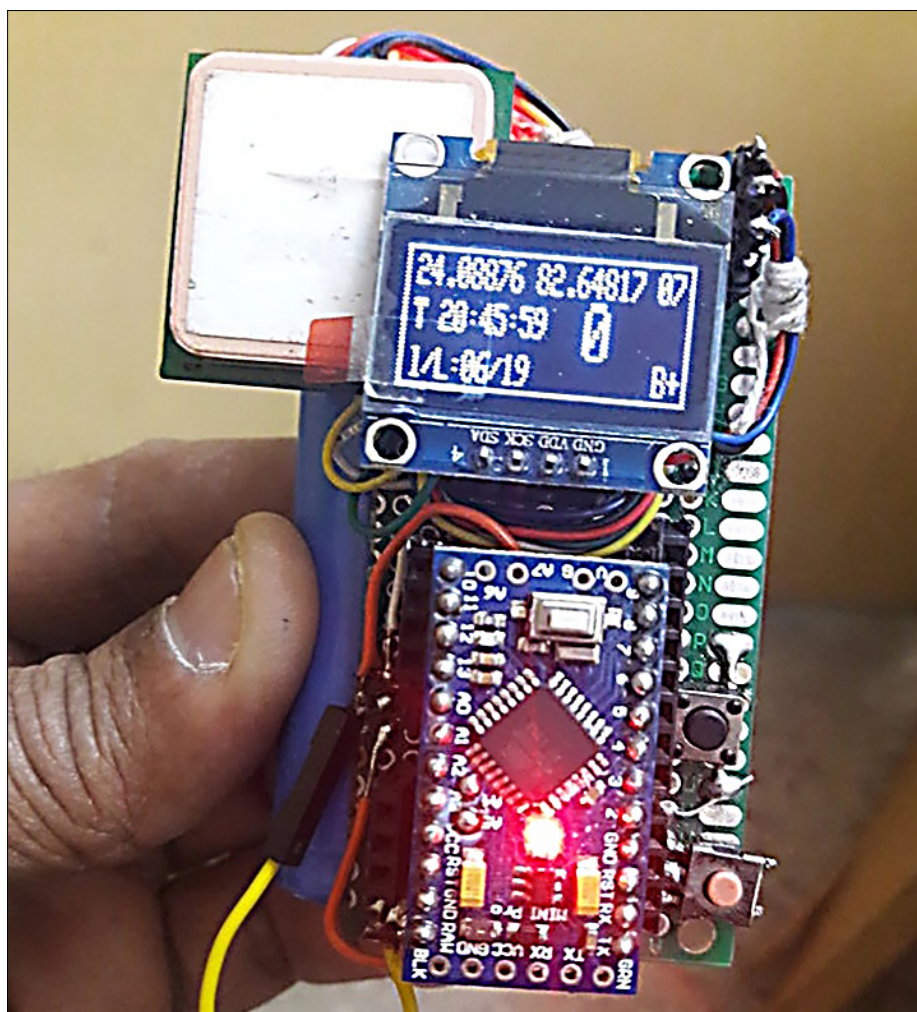


Figure 7. On peut aisément construire l'appareil sur un morceau de circuit imprimé, il est assez petit pour tenir dans la main.

GPS est en contact avec assez de satellites, la LED PPS (*pulse per signal*) se met à clignoter et l'écran OLED montre les informations vues ci-dessus. ◀

(180432-04

version française : Robert Grignard)

En mémoire de



L'auteur dédie cet article à feu sa mère, Sefali Bera, dont la disparition l'a profondément touché ainsi que la communauté où elle a vécu, à Howrah, en Inde. Elle était une grande source d'inspiration pour ses articles.



@ WWW.ELEKTOR.FR

→ Livre « Maîtrisez les microcontrôleurs à l'aide d'Arduino (3^e édition) » :
www.elektor.fr/maîtrisez-les-microcontrôleurs-a-l-aide-d-arduino-3e-edition

→ Module GPS :
www.elektor.fr/gy-neo6mv2

→ Arduino Nano, réf. 17002 :
www.elektor.fr/arduino-nano-3

Liens

[1] Module GPS : www.elektor.fr/gy-neo6mv2

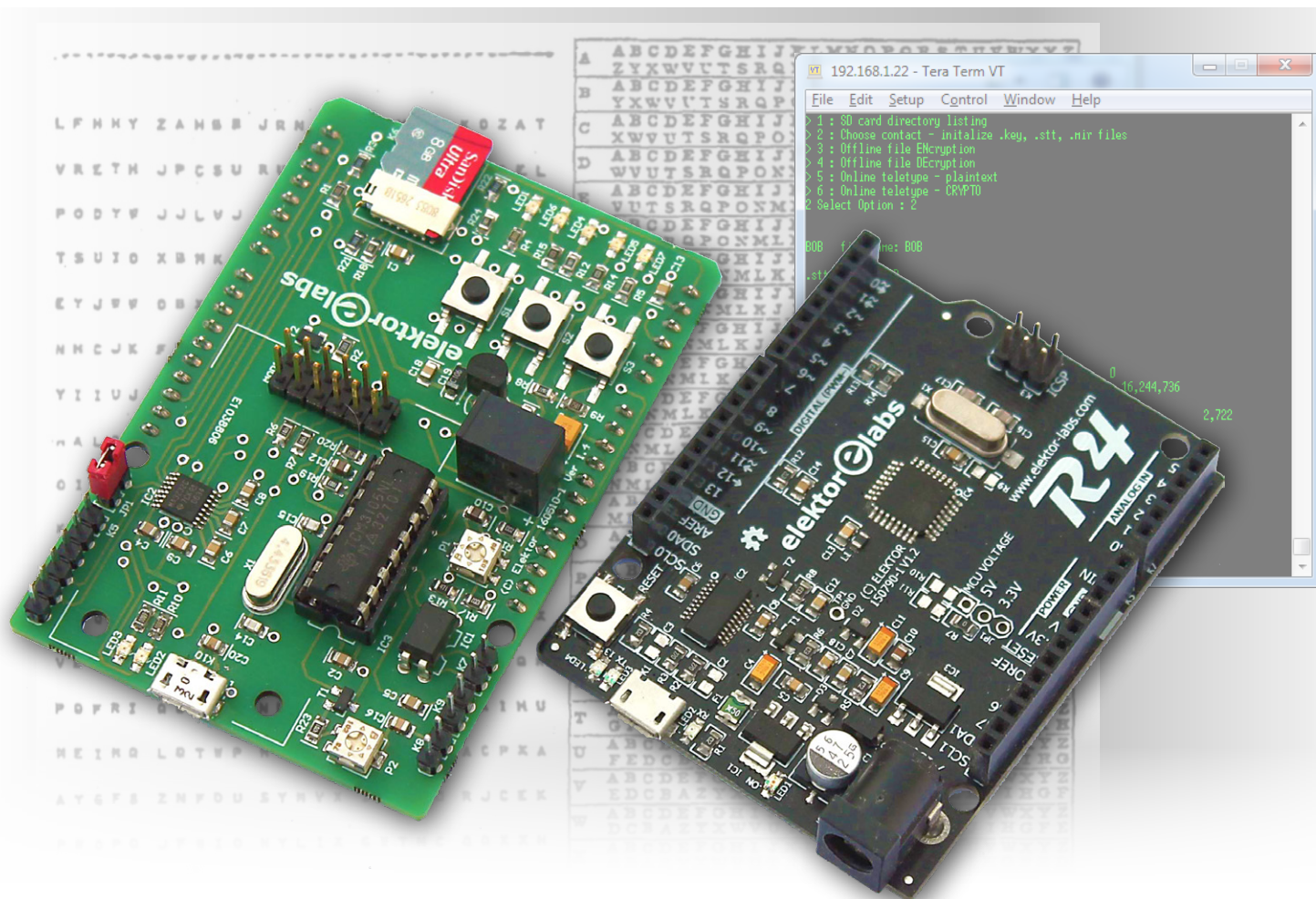
[2] Page de l'article : www.elektormagazine.fr/180432-04

shield cryptographie à masque jetable

misez sur la transparence, gagnez en sécurité

Luka Matic

Il existe toute une panoplie de matériels et logiciels de cryptographie, alors pourquoi vouloir construire son propre système ? La raison est simple : vous ne devriez accorder aucune confiance à ces dispositifs ou programmes, car dans 99 % des cas leur mécanisme interne est complètement opaque !



Inventée par les Allemands au début du XX^e siècle, la machine Enigma (**fig. 1**, et [1] pour une présentation en vidéo) comprenait pour l'essentiel une pile, un clavier, des lampes et un jeu de rotors. Son mécanisme électromécanique était relativement simple au vu de sa fonction. Le nombre total de combinaisons

possibles qu'offraient ses rotors (sa « force cryptographique ») était connu et facile à calculer. Veiller sur son bon fonctionnement était également aisé, le problème des bogues ne se posant pas puisqu'elle opérait sans logiciel. Enigma pouvait résister à toutes les méthodes de cryptanalyse connues à l'époque et ne

disposait d'aucun matériel de communication : l'entrée (le clavier) et la sortie (les lampes) étaient entièrement sous le contrôle d'un opérateur, un autre s'occupant des transmissions radio. De plus :

- il était impossible d'y introduire un cheval de Troie et d'exfiltrer du texte brut vers une adresse IP.

- Elle était immunisée contre les débordements de tampon, et aucun enregistreur de frappe (matériel ou logiciel) ne pouvait intercepter les touches utilisées.
- Elle opérait à des fréquences très basses, donc n'émettait quasiment pas de rayonnement résiduel TEM-PEST [2], en particulier lorsqu'elle était utilisée dans une cage de Faraday presque parfaite, un U-Boot par exemple.

De nos jours, la plupart des étapes de chiffrement sont effectuées par des logiciels exécutés sur un PC généraliste, certains appareils assez coûteux comme les cryptophones se chargeant du reste. Même le meilleur algorithme de chiffrement s'avèrera inutile s'il est exécuté sur un dispositif non sécurisé, un PC p. ex. La plupart des matériels à usage cryptographique se présentent sous la forme d'une boîte noire, et sont donc difficiles à analyser par un profane. Même s'ils sont dépourvus de protection matérielle ou logicielle, leur complexité empêche encore bien souvent leur étude. Vu sous cet angle, vous comprendrez aisément que l'antique Enigma reste de loin ce qui peut se faire de mieux en matière de sécurité ! Ce que je veux dire, c'est qu'un système est en fait plus sûr s'il est facile à analyser. Paranoïa de ma part ? Lisez [3], notamment le chapitre traitant des arbres d'attaque.

Masque jetable

Selon Wikibooks [4], le seul algorithme de chiffrement potentiellement indéchiffrable est celui du masque jetable (*one-time pad*, ou OTP en anglais). Pourquoi « potentiellement » ? Parce seule une application scrupuleuse de certaines règles le rend inviolable.

Chaque message doit ainsi être chiffré avec une clé aléatoire unique (**fig. 2**), cette clé devant :

- être produite par un procédé non déterministe et non reproductible – c'est-à-dire non algorithmique ;
- n'être utilisée qu'une seule fois ;
- ne pas tomber aux mains de l'ennemi.

Le processus de chiffrement est simple : une clé appropriée est générée, puis un XOR est appliqué à cette clé et au message en clair. Quant au déchiffrement, il est tout aussi simple puisqu'il consiste à

combiner le message chiffré et la même clé avec un XOR. Le message chiffré a donc la même longueur que le message en clair. La clé étant aléatoire et tous les messages en clair possibles de même longueur étant également probables, il est dès lors impossible de déterminer le message d'origine à partir de sa seule version chiffrée. Le masque jetable est le seul système cryptographique à fournir une telle preuve.

Un système pratique

La méthode du masque jetable repose sur un algorithme facile à programmer, exécutable par un microcontrôleur d'architecture connue (Harvard). Le code source de l'algorithme étant public, n'importe qui peut l'analyser et le déboguer en détail pour savoir ce que fait le μC . L'algorithme n'utilise que deux tampons RAM de données qu'il faut protéger de tout débordement.

INFOS SUR LE PROJET	
	cryptographie GNRA
	masque jetable
	communication sécurisée
	débutant
	→ connaissseur
	expert
	env. 2 h
	soudage de CMS, programmeur AVR, Atmel Studio
	env. 25 €



Figure 1. Vue partielle de la machine Enigma, le cryptographe électromécanique à rotors.

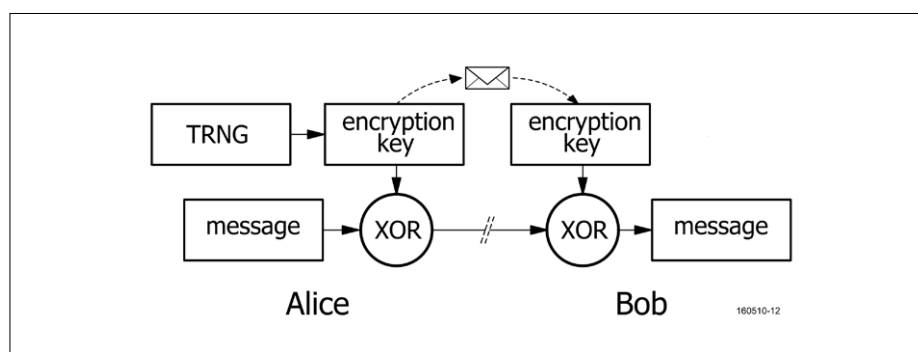


Figure 2. Synoptique du chiffrement par masque jetable. Ici, Alice envoie un message à Bob. (Nota : TRNG = générateur de nombres réellement aléatoires)

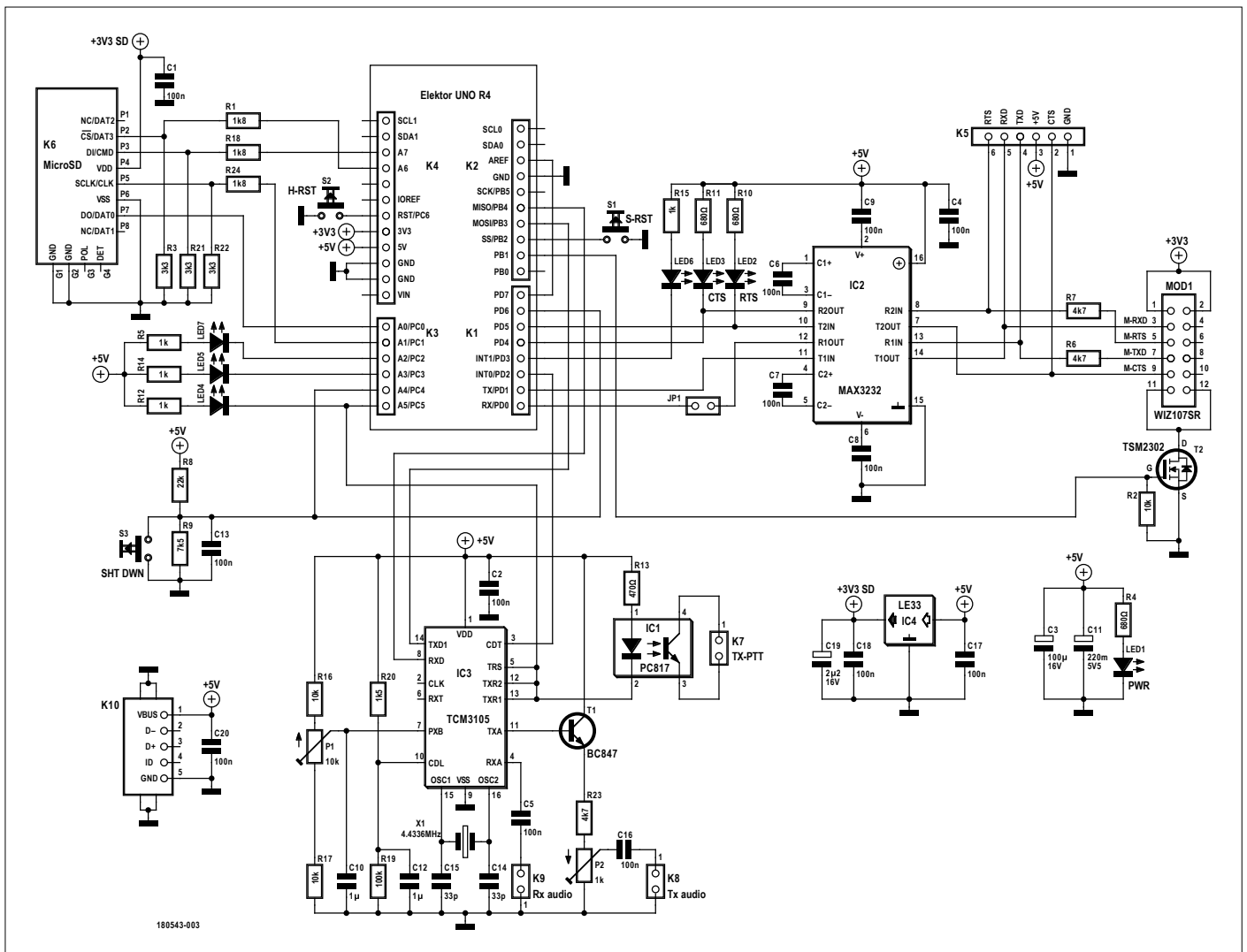


Figure 3. Le schéma du *shield* OTP Crypto peut être découpé en trois parties : les ports RS-232 et Ethernet entourant K5, IC2 et MOD1 ; l'interface audio et son modem à MDF (IC3) ; enfin le logement de la carte SD pour le stockage des données de chiffrement et des messages.

La création de clés de chiffrement fort nécessite un générateur de nombres réellement aléatoires (GNRA, cf. [5]), lui aussi à code et matériel ouverts afin qu'il soit possible d'en analyser chaque facette. Les clés sont stockées sur une carte SD, permettant ainsi des gigaoctets de séquences aléatoires (fini les épais cahiers d'autrefois !).

Le système nécessite un terminal doté d'un port série ou Ethernet pour communiquer avec le μ C, et d'un lecteur de carte SD pour lire/écrire les fichiers. Il s'agira p. ex. d'un ordinateur portable bon marché de 7 pouces, privé de son module Wi-Fi et de sa caméra, et *jamais* connecté à l'internet. Le chiffrement est effectué par le μ C, qui – comme Enigma – est physiquement séparé des autres supports de traitement informatique et de transmission. Cette conception rend le système difficile à attaquer avec un

virus ou un cheval de Troie. Son rayonnement TEMPEST est cependant plus fort que celui d'Enigma.

Construit sans matériel spécial et avec des composants répandus, notre système est facile à assembler et peut être analysé en détail.

Deux modes de fonctionnement

Notre cryptographe peut opérer selon deux modes : en ligne (*Online*) ou hors ligne (*Offline*). En mode *Online*, les messages chiffrés sont envoyés et reçus par modem. En mode *Offline*, les messages et les fichiers sont codés et stockés sur un support tel qu'une carte SD, puis transmis au destinataire par courriel ou voie postale.

Matériel

Le système comprend une carte compatible Arduino surmontée d'un *shield*

baptisé OTP Crypto (**fig. 3**). Nous avons opté pour la carte Elektor Uno R4 en raison de son ATmega328PB, de ses deux ports SPI et de ses deux UART.

Les composants principaux du *shield* OTP Crypto sont :

1. Un logement pour la carte SD stockant les fichiers de clé (modes hors et en ligne) et ceux de chiffrement/déchiffrement (mode hors ligne).
2. Un port RS-232 (MAX3232) pour connecter le *shield* au terminal. Les données série peuvent être envoyées via le connecteur RS-232 (K5) ou par Ethernet au moyen d'un adaptateur Ethernet-RS-232 WIZ107SR (si p. ex. le terminal n'a pas de port série, ce qui sera le cas d'un portable bon marché). R6 et R7 peuvent être utilisées pour déconnecter le module Ethernet si ce dernier est soudé sur le *shield*.
3. Un modem à MDF (TCM3105) pour

l'émission-réception des textes chiffrés via un canal audio analogique (mode *Online*).

Coupure progressive

Pour éviter que la même séquence de chiffrement ne soit utilisée deux fois, il importe de conserver la position actuelle du pointeur de clé ainsi que les valeurs d'autres variables. Cette sauvegarde doit même se faire au plus vite sur la carte SD, au cas où il y aurait une panne d'alimentation. Si cela arrive, le supercondensateur C11 alimente le circuit durant quelques secondes, rendant ainsi la coupure progressive. Toute chute de tension sur PD6 est détectée par le comparateur de tension interne du μ C, ce qui permet d'implanter la coupure progressive dans une routine d'interruption. On peut aussi la déclencher manuellement via S3.

Le transistor T2 déconnecte WIZ107SR (si présent) en cas de panne d'alimentation, car ce module absorbe plus de 200 mA, valeur qui entraînerait une baisse de tension trop grande en raison de la résistance interne de C11, et empêcherait de fait toute écriture sur la carte SD.

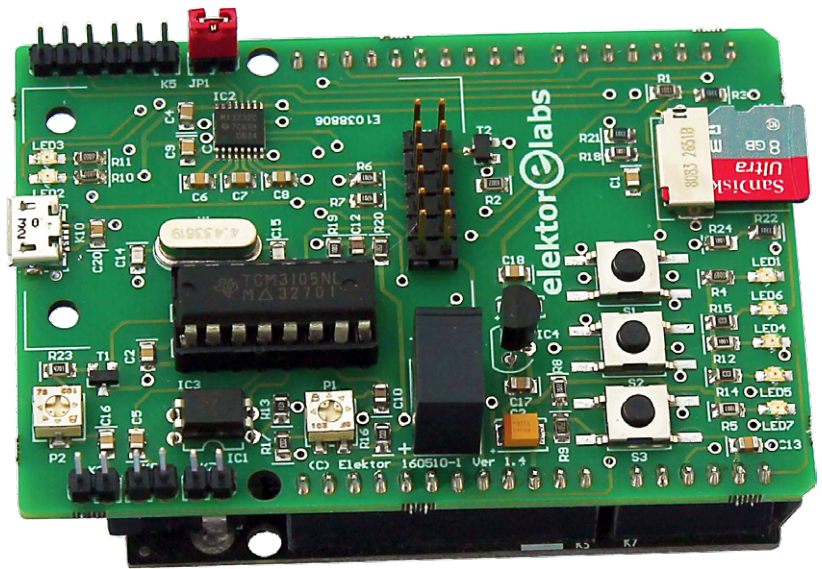
S2 est un bouton de remise à zéro matérielle, à utiliser avec précaution pour éviter toute perte de données critique (p. ex. la valeur du pointeur de clé). S1 déclenche une remise à zéro logicielle, sans perte de données, à utiliser en cas de débordement de l'UART en mode *Online*, ou pour stopper une recherche de pointeur de clé prenant trop de temps.

Les subtilités de l'alimentation

La borne K10 sert à relier une alimentation de 5 V au circuit. Vous pouvez utiliser le connecteur micro-USB de la carte Uno R4 pour flasher le micrologiciel avec le chargeur de démarrage (dans ce cas, JP1 doit être ouvert), mais pas pour alimenter le *shield* durant une utilisation normale, car cela pourrait affecter l'initialisation du μ C. Pour parer à cet aléa, c'est le *shield* qui alimente la carte Uno R4.

Pourquoi tant de LED ?

Le *shield* est doté de nombreuses LED témoin. LED1 indique la mise en service, LED2 et LED3 une activité sur les signaux RTS et CTS de l'UART0 du μ C. LED4 indique une transmission en cours (venant d'UART1) sur le port audio du modem. LED5 signale une réception de données, LED7 une erreur de réception.



LED6 indique le traitement d'une demande de remise à zéro matérielle (via S1), ou clignote lorsque le μ C tente de déchiffrer un message (ou un fichier en mode *Offline*) après perte de la position du pointeur de clé. Si une baisse de tension est détectée en PD6 ou si S3 a été pressé, LED5 et LED6 clignotent tour à tour pour indiquer que le fichier d'état STT (voir ci-dessous) a bien été mis à jour ; elles clignotent simultanément s'il ne l'a pas été (son contenu n'a pas changé ou la tentative d'écriture a échoué).

Canal audio

C'est IC3, un modem à FSK (modulation par déplacement de fréquence, MDF), qui fournit un canal audio pour l'émission-réception des messages chiffrés au moyen, p. ex., d'un téléphone ou d'une radio mobile privée (RMP). IC3 opère à 1200 bauds avec des fréquences de modulation de 1300 Hz et 2100 Hz. Selon la règle de Carson, la largeur de bande du signal sera comprise entre 700 Hz et 2700 Hz, un intervalle satisfaisant même pour un canal audio de faible qualité. P1 fixe le seuil de distinction entre les deux fréquences (0 ou 1). Pour le régler, envoyez un message depuis un autre *shield*, en mode *Online plaintext* (option 5, cf. ci-après) et avec un signal faible. P2 est en position médiane pour satisfaire la connexion au microphone ou à l'entrée de ligne de la plupart des appareils. Réglez-le sur une valeur basse lorsque vous ajustez P1 sur l'autre *shield*. Le photocoupleur IC1 sert à « presser » le bouton d'émission/réception si un appareil unidirectionnel tel qu'une radio VHF/UHF PMR (*private mobile radiocom-*

munications) est utilisé pour émettre/recevoir le signal audio MDF codé.

En mode *Online*, vous pouvez aussi utiliser n'importe quel canal analogique (connexion filaire, téléphone analogique) ou numérique avec compression sans perte (p. ex. G.711, loi A ou loi μ). Cela inclut tout ordiphone relié à l'internet exécutant un logiciel VoIP à codec sans perte. Connectez-le au *shield* via son port audio analogique EAR-MIC. N'utilisez pas de codecs sans perte comme GSM ou ceux utilisés par Skype ou Viber. Ils sont parfaits pour la voix humaine, mais pas pour les signaux d'un modem à MDF car leur hauteur varie bien plus rapidement que la voix humaine.

Comment ça marche ?

Supposons que deux personnes veuillent communiquer à l'aide de *shields* OTP Crypto. Toutes deux doivent partager le même fichier de clé (KEY, créé par le générateur de nombres réellement aléatoires). Nous l'avons dit, une même sous-séquence d'octets aléatoires tirée de ce fichier ne doit jamais servir deux fois. Autrement dit chaque fichier doit comporter un pointeur, et les deux pointeurs doivent être synchronisés. Ils doivent aussi être restaurés en cas de perte de synchronisation. Chaque partie doit donc garder trace des deux pointeurs pour que la même sous-séquence ne serve pas deux fois (ce qui compromettrait aussitôt la sécurité du système). C'est la raison pour laquelle, outre le fichier KEY, chaque partie doit également posséder un fichier d'état (STT) contenant son pointeur de clé (et d'autres données de contrôle) ainsi qu'un fichier dit miroir. Ce fichier MIR, envoyé à ou reçu

de l'autre partie, contient la version chiffrée du fichier STT, et sert à authentifier et vérifier l'état de l'autre partie si celle-ci est en mode *Offline*.

Exemple

Illustrons ce mécanisme en faisant appel aux traditionnels couples Alice et Bob

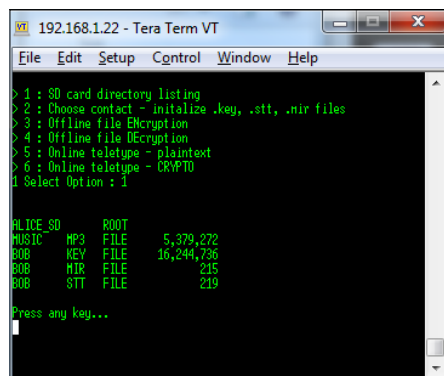


Figure 4. Sélectionnez l'option 1 pour lister le contenu du répertoire *root*.

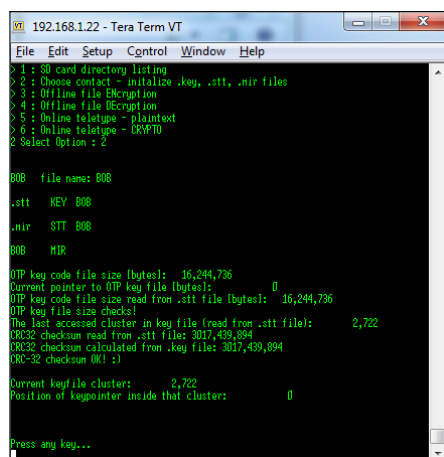


Figure 5. Entrez '2' pour saisir le nom d'un contact.

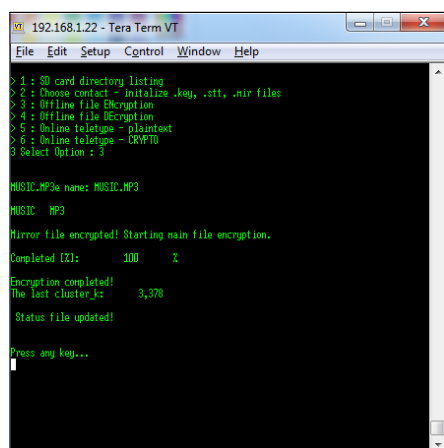


Figure 6. Chiffrement du fichier terminé.

(les « bons »), et Eve et Mallory (les « méchants »). Alice et Bob veulent échanger des informations cruciales. Eve est passive : elle écoute les messages et cherche à en casser le code ; Mallory est actif : il tente de brouiller la communication ou de modifier le contenu des messages ou des fichiers.

La carte SD d'Alice contient les fichiers BOB.KEY, BOB.STT et BOB.MIR, celle de Bob contient ALICE.KEY, ALICE.STT et ALICE.MIR. Les deux fichiers KEY sont identiques (créés par le GNRA) et contiennent entre 16 Mo et 4 Go de données aléatoires. Les fichiers d'état STT sont peu volumineux (pas plus de 512 octets) et contiennent du texte normal, lisible avec un éditeur de texte :

ALICE.STT :

```
POINTER :0000000000 [bytes]
TOTAL SIZE :0016244736 [bytes]
CLUSTER :0000007287 [clusters]
CRC32 :2989374417
```

Alice lit Lewis Carroll

Autres informations vitales...

BOB.STT :

```
POINTER :0000000000 [bytes]
TOTAL SIZE :0016244736 [bytes]
CLUSTER :0000007287 [clusters]
CRC32 :2989374417
```

Bob porte un slip en éponge

Autres informations vitales...

POINTER et TOTAL SIZE indiquent à quel moment un nouveau fichier de clé est nécessaire, la valeur de POINTER progressant lentement jusqu'à la valeur de TOTAL SIZE. CLUSTER indique le numéro du bloc (*cluster*) sur la carte SD vers lequel pointe actuellement POINTER dans le fichier de clé. CRC32 est une somme de contrôle calculée à partir des seize premiers octets du CLUSTER actuel et de quatre octets de la valeur de POINTER (entier de 32 bits stocké dans la SRAM du µC au format *little-endian*). Toute modification de ces valeurs peut ainsi être détectée. Si la valeur de POINTER est modifiée manuellement, le µC recherchera le CLUSTER correct sur la carte, calculera le CRC32, puis mettra à jour le fichier STT. Notez que les valeurs CLUSTER des deux cartes SD n'ont pas à être identiques pour des valeurs de POINTER identiques, puisque les fichiers seront

probablement placés dans des blocs différents de cartes SD différentes (et ils peuvent aussi être fragmentés).

Il importe que le fichier STT commence par la chaîne POINTER suivie d'un espace de fin (soit huit lettres). Le déchiffrement du fichier MIR codé sera considéré comme réussi (en validant le pointeur de clé durant l'opération) si cette chaîne de huit lettres apparaît lors d'une tentative de déchiffrement des huit premiers caractères du fichier MIR. Les quatre valeurs du fichier STT sont des nombres à dix chiffres commençant immédiatement après un deux-points – c'est ainsi que le µC les lit.

Mode Offline

En mode Offline, Alice suivra la procédure suivante pour envoyer un fichier chiffré à Bob :

1. Elle insère sa carte SD dans son ordinateur et y stocke le fichier à chiffrer, par exemple un fichier mp3 nommé MUSIC.MP3.
2. Sur sa carte SD, Alice copie son fichier BOB.STT dans le fichier BOB.MIR (dont l'ancien contenu peut être effacé, il ne sert plus). BOB.MIR contient maintenant du texte brut identique à celui de BOB.STT.
3. Elle insère la carte SD dans le *shield* et le met sous tension.
4. Elle affiche le contenu du répertoire *root* (option 1, **fig. 4**).
5. Elle choisit le ou les destinataires du fichier codé. Il suffit de saisir le ou leurs noms, les extensions seront automatiquement ajoutées aux fichiers (NAME.KEY, NAME.STT, NAME.MIR, **fig. 5**).
6. Elle choisit l'option 3 – *Offline file Encryption* – et saisit le nom de fichier complet, extension comprise. Le µC chiffre d'abord le fichier miroir BOB.MIR, puis le fichier principal MUSIC.MP3. Une fois le chiffrement terminé, le µC met à jour l'état dans le fichier BOB.STT (**fig. 6**).
7. Alice retire sa carte SD du *shield* et l'insère dans son ordinateur. Le fichier MUSIC.MP3 ne peut plus être lu car il est codé. L'ouverture de BOB.MIR, lui aussi codé, montrerait un contenu semblable à celui de la **figure 7**. BOB.STT reste en clair, mais son contenu a changé (**fig. 8**) : la valeur de POINTER a progressé, de même que celle de CLUSTER, et donc celle de CRC32 a elle aussi changé. Le mp3 faisait environ 5 Mo, ce que reflète la valeur de POINTER.

8. Usant d'un moyen de communication non sécurisé de type ordinateur ou ordiphone (la voie postale est aussi une option), Alice envoie un courriel contenant en pièces jointes les fichiers chiffrés MUSIC.MP3 et BOB.MIR. Alice aurait pu les renommer, tout comme Bob après leur réception. Renommer ces deux fichiers accroît la sécurité.

Réception hors ligne

La carte SD de Bob contient les fichiers ALICE.KEY et ALICE.STT (**fig. 9**). Le POINTER de Bob est encore à 0, et atteindra 0005380096 à la fin du chiffrement. Lorsque Bob reçoit les fichiers d'Alice, il suit la procédure de décodage suivante :

1. Bob copie sur sa carte SD les deux fichiers chiffrés reçus d'Alice – MUSIC.MP3 et ALICE.MIR.
2. Bob insère sa carte SD dans son *shield*. La **figure 10** en montre le répertoire *root*.
3. Après sélection de l'option 2 (choix des fichiers avec le nom du contact) et une étape d'initialisation, le système est prêt pour le déchiffrement (option 4, **fig. 11**). Le *shield* tente de déchiffrer ALICE.MIR avec la valeur du pointeur de clé lue dans ALICE.STT de Bob. Le fichier de clé est parcouru jusqu'à ce que la valeur initiale correcte du pointeur de clé soit trouvée. Là encore le déchiffrement est considéré comme réussi si les huit premiers caractères du fichier MIR sont POINTER (avec l'espace de fin). Le fichier MIR est alors affiché, et Bob peut vérifier plusieurs points :

- La position enregistrée du pointeur de clé d'Alice (0000000000, avant qu'elle ait lancé le chiffrement) doit être égale à la valeur du pointeur de clé de Bob avant qu'il ait lancé le déchiffrement. Si la valeur du pointeur d'Alice est inférieure à celle de Bob (quoi qu'il en soit le μC cherchera et trouvera la position correcte), alors il se peut que la sécurité soit compromise, car Alice a réutilisé une séquence de clé. Une partie de ses précédents messages pourrait donc être décodée par Eve.
- La présence de charabia après déchiffrement du fichier MIR pourrait indiquer une corruption du message par Mallory, p. ex. par modification du pointeur de clé dans le fichier MIR. Une parade à ce type d'attaque est de glisser dans le

fichier MIR, avant son chiffrement, quelques courts messages de vérification entre les quatre valeurs à dix chiffres (sans modifier leur formatage, et sans utiliser le deux-points pour les messages additionnels). Cela change la position des caractères dans le fichier MIR, ren-

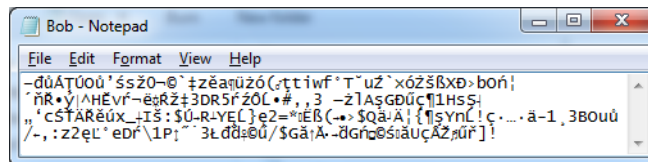


Figure 7. Le fichier BOB.MIR chiffré.

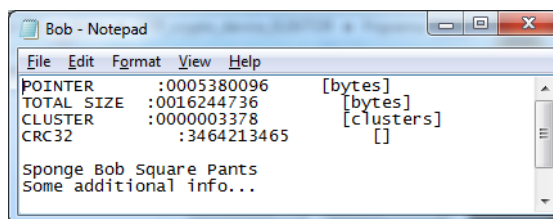


Figure 8. Le fichier d'état BOB.STT reste toujours en clair.

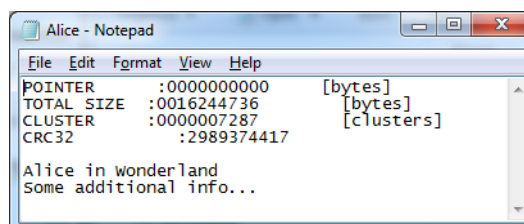


Figure 9. Contenu du fichier ALICE.STT sur la carte de Bob.

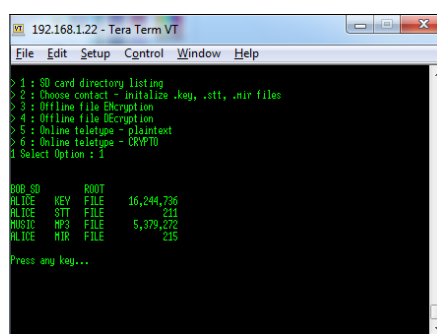


Figure 10. Répertoire *root* de la carte SD de Bob.

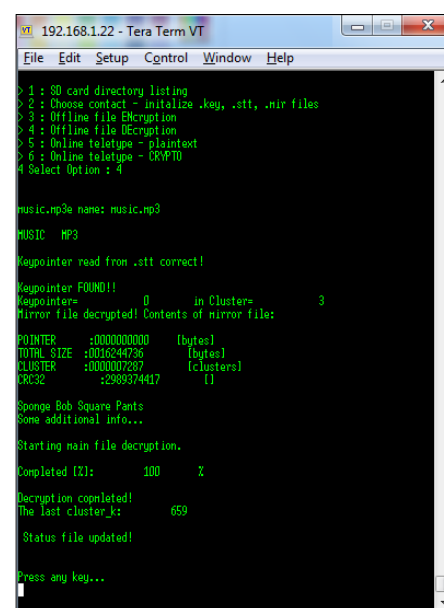


Figure 11. Déchiffrement du fichier terminé. Le fichier doit être lu pour être certain que tout s'est bien déroulé.

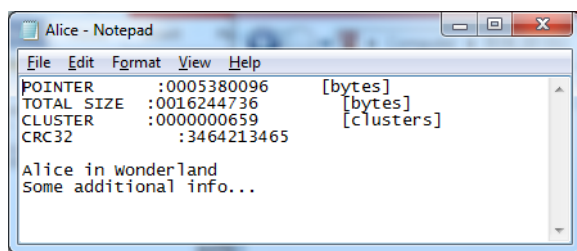


Figure 12. Le fichier d'état sur la carte SD de Bob doit montrer une progression du pointeur de clé.

dant ainsi très probable que Mallory modifie le mauvais octet.

- Bob peut vérifier lui-même la cohérence de toutes les valeurs du fichier MIR déchiffré afin de détecter une tentative d'altération de la part de Mallory.

4. Après déchiffrement du fichier mp3, Bob peut retirer la carte SD du *shield* et l'insérer dans son ordinateur. Il peut écouter le mp3, mais aussi vérifier le contenu de son fichier d'état ALICE.STT (fig. 12). Les deux poin-

teurs de clé sont synchronisés et le système est prêt à chiffrer la prochaine communication.

Mode Online

En mode *Online*, le *shield* vérifie en permanence la valeur du pointeur de clé et en signale toute décroissance, synonyme de réutilisation de la séquence de chiffrement.

Après entrée du contact (comme en mode *Offline*) et initialisation du mode, les cryptographes des deux parties trans-

mettent la chaîne Rd (*Ready*). Celui qui la reçoit est prêt à envoyer le premier message.

Chaque message envoyé par Alice contient :

- le pointeur de clé (4 octets) ;
- le message (charge utile, texte brut, N octets) ;
- une somme de contrôle CRC-16 (2 octets) calculée avec (« sur ») le pointeur et la charge utile.

Cette séquence de 4+N+2 octets est chiffrée par masque jetable, puis il lui est ajouté une seconde somme de contrôle CRC-16 (2 octets) calculée avec la séquence chiffrée.

Réception en ligne

Le message reçu par Bob est déchiffré comme suit :

La somme de contrôle du message codé est calculée pour détecter toute erreur de transmission éventuelle. Si les deux sommes diffèrent, la chaîne Er (*Error*) est envoyée à Alice pour demander une



LISTE DES COMPOSANTS

Résistances

Défaut : 0805, 125 mW, 5 %

- R1, R18, R24 = 1,8 kΩ
- R2, R16, R17 = 10 kΩ
- R3, R21, R22 = 3,3 kΩ
- R4, R10, R11 = 680 Ω
- R5, R12, R14, R15 = 1 kΩ
- R6, R7, R23 = 4,7 kΩ
- R8 = 22 kΩ
- R9 = 7,5 kΩ, 1 %
- R13 = 470 Ω
- R19 = 100 kΩ
- R20 = 1,5 kΩ
- P1 = ajustable 10 kΩ
- P2 = ajustable 1 kΩ

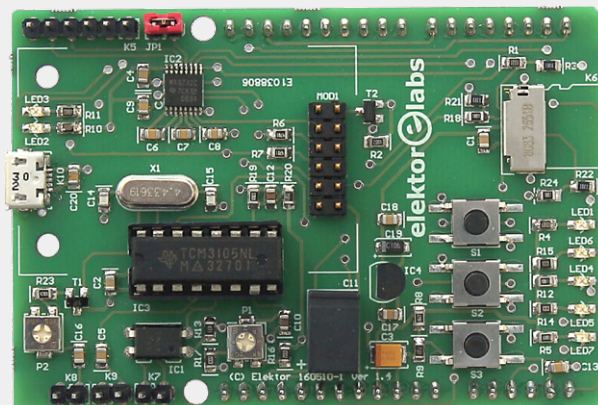
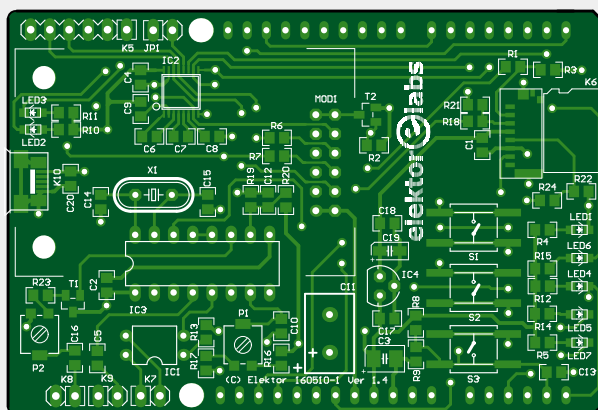
Condensateurs

Défaut : CMS 0805

- C1, C2, C4, C5, C6, C7, C8, C9, C13, C16, C17, C18, C20 = 100 nF
- C3 = 100 µF, 10 V, boîtier B
- C10, C12 = 1 µF
- C14, C15 = 33 pF
- C11 = 0,22 F, 5,5 V, supercondensateur, au pas de 5 mm
- C19 = 2,2 µF, 16 V, boîtier A

Semi-conducteurs

- IC1 = HCPL-817-000E
- IC2 = MAX3232CPWR
- IC3 = TCM3105NL



IC4 = LE33CZ-TR

T1 = BC847C

T2 = TSM2302CX RFG

LED1, LED2, LED3 = verte, 0805

LED4, LED5, LED6 = jaune, 0805

LED7 = rouge, 0805

Divers

- K1 = barrette mâle, 8 contacts, au pas de 2,54 mm
- K2 = barrette mâle, 10 contacts, au pas de 2,54 mm
- K3, K5 = barrette mâle, 6 contacts, au pas de 2,54 mm
- K4 = barrette mâle, 12 contacts, au pas de 2,54 mm
- K6 = connecteur pour carte micro-SD, Molex série 47309
- K7, K8, K9, JP1 = barrette mâle, 2 contacts, au pas de 2,54 mm
- K10 = embase femelle micro-USB type B
- S1, S2, S3 = interrupteur tactile, 6×6, 2 mm, CMS
- X1 = quartz à 4,433619 MHz
- MOD1 = barrette à 12 broches (2×6), au pas de 2,54 mm
- Support DIP, 16 contacts
- Circuit imprimé, réf 160510-1 dans l'e-choppe
- Module série-Ethernet WIZ107SR
- Carte Elektor Uno R4

retransmission. La chaîne Ok est envoyée si la somme est correcte. La chaîne To (*Time-out*) signale un dépassement de délai lorsque la confirmation par l'autre partie prend trop de temps. Mallory pourrait facilement avoir modifié certains octets du message, et manipulé les deux derniers octets afin que les CRC-16 correspondent. S'il tente de brouiller le canal de communication, une erreur de somme de contrôle surviendra.

Le *shield* de Bob entame ensuite le déchiffrement du message en cherchant le pointeur de clé initial correct dans le bloc actuel de la carte SD, ainsi que dans les suivants et les précédents. Le déchiffrement est considéré comme réussi si le CRC-16 décodé est correct, et si le pointeur de clé décodé est identique à celui du *shield* de Bob. La chaîne Dc (*Decryption*) est alors envoyée à Alice.

Mallory serait bien en peine de contourner cette protection. Toute altération du message engendrerait un message impossible à déchiffrer (après vérification des sommes de contrôle des messages codés), ce qui avertirait Alice et Bob de sa présence et de son activité. Mallory pourrait renvoyer un message précédemment enregistré. Il serait déchiffré correctement, mais la valeur du pointeur de clé serait trop basse et le *shield* d'Alice ou de Bob le signalerait.

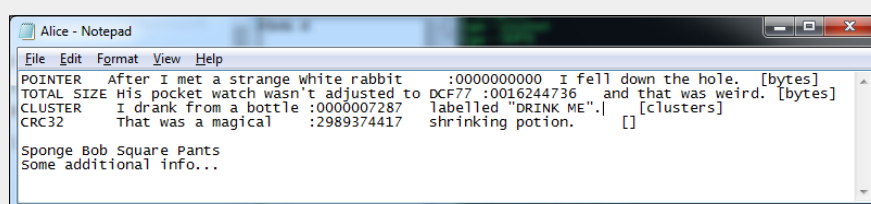
Conclusion

À moins d'une compréhension profonde de ses méthodes, la cryptographie reste l'un des rares domaines où l'on ne peut se fier à des dispositifs étiquetés « prêt à l'emploi » ou « à sécurité intégrée ». La sécurité des données est un processus complexe, pas un produit. En soi, un cryptographe restera donc toujours inachevé.

Celui présenté ici a été conçu pour contrer les nombreuses attaques susceptibles d'être lancées contre un PC généraliste, le dispositif de chiffrement

Comment (encore) renforcer la sécurité

- Défragmenter le fichier KEY accélérera la recherche du pointeur lorsque l'algorithme aura à revenir à un bloc de données antérieur.
- Formatez votre carte SD en FAT32 avec une taille de bloc de 8 à 16 Ko.
- Démarrez votre communication en ligne par un court message. La synchronisation des pointeurs en sera d'autant plus rapide, car la recherche d'un pointeur de clé correct prend plus de temps avec les messages longs.
- En cas de désynchronisation, « jetez » le tampon en utilisant une séquence incohérente (vide de sens) dont la longueur dépendra de votre degré de suspicion. Pour casser le code, Eve a en effet besoin de deux messages cohérents en clair, cryptés avec la même séquence de clé.
- Si votre pointeur de clé a une valeur significativement inférieure à l'autre, augmentez-la manuellement dans le fichier STT. Mieux vaut perdre un peu de l'aléatoire d'une séquence que d'en utiliser une partie deux fois.
- Avant de lancer le chiffrement en ligne, cachez quelques messages de vérification (sans ':', cf. texte) dans votre fichier MIR. En voici un exemple :



- Le chargeur de démarrage de la carte Elektor Uno R4 se montrera utile pour modifier ou déboguer le système, mais il doit être désactivé avec les valeurs de fusibles correctes et retiré de la mémoire flash. La version finale du micrologiciel sans *bootloader* doit être écrite via le port ISP. Un chargeur malencontreusement activé au démarrage peut écraser des pages de la mémoire flash, et mettre sérieusement en péril la stabilité et la sécurité du système.

le plus courant de nos jours. Il a été pensé pour être plus sûr qu'un PC ou un cryptophone de type boîte noire, et je crois qu'il l'est. Mais peut-être ai-je oublié certaines choses ? Postez vos suggestions sur la page du projet [6]. Une fonction de hachage non linéaire serait

sans doute plus efficace que CRC, et je prévois également une protection contre les attaques TEMPEST, mais c'est là un problème complexe qui devra être traité séparément. ◀

(180543-04 - version française : Hervé Moreau)



@ WWW.ELEKTOR.FR

→ Circuit imprimé nu du *shield* OTP Crypto, réf. 160510-1

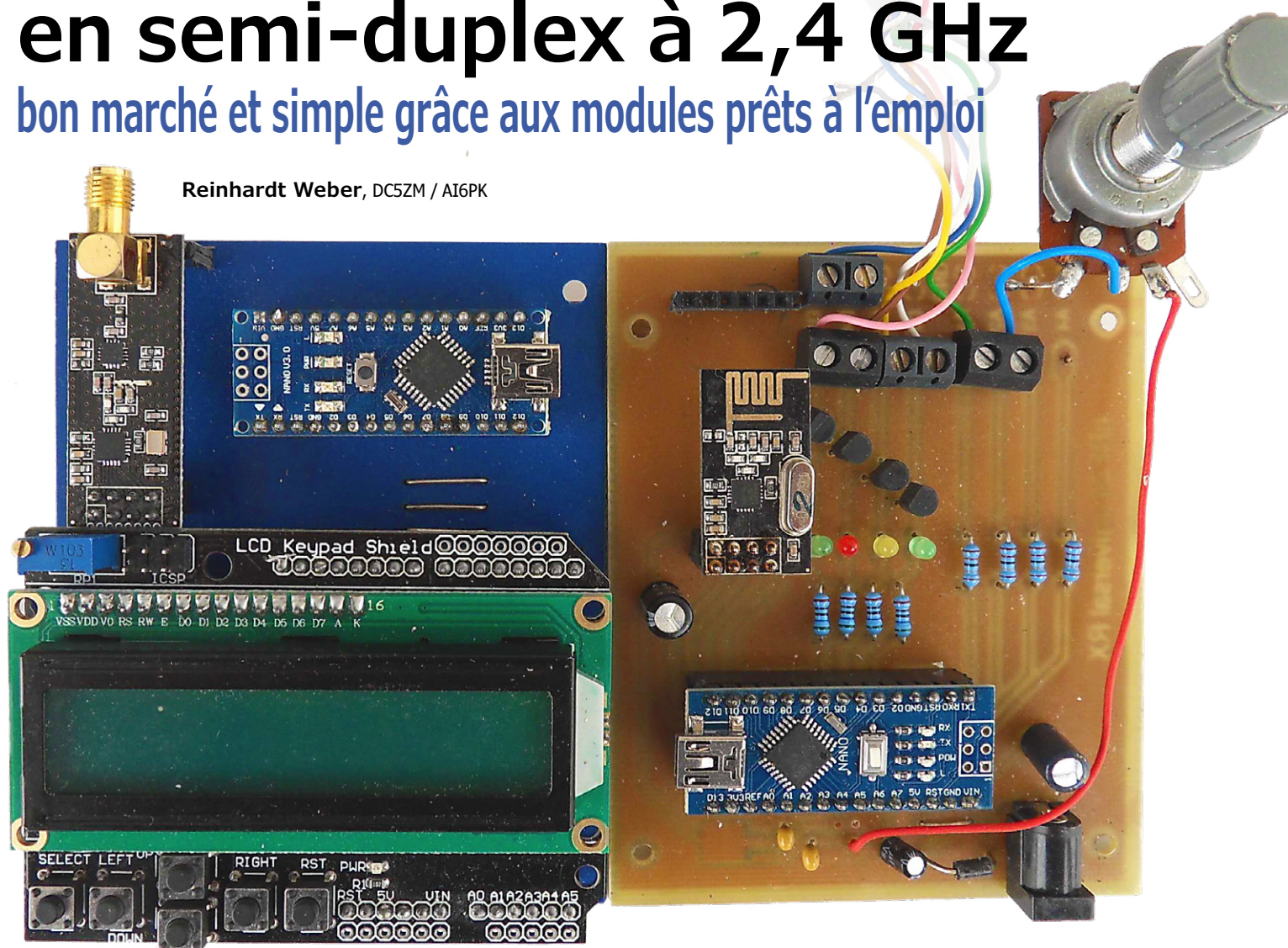
www.elektor.fr/crypto-shield-160510-1

Liens et littérature

- [1] Enigma et ses descendants sur Elektor TV : www.elektormagazine.com/news/elektor-tv-historic-cryptography
- [2] Wim van Eck, 'Electromagnetic Radiation from Video Display Units: An Eavesdropping Risk?', 1985 : <https://cryptome.org/emr.pdf>
- [3] Bruce Schneier, 'Secrets & Lies' : Digital Security in a Networked World', Wiley, 2004 :
- [4] Wikibooks sur masque jetable (OTP) : https://en.wikibooks.org/wiki/Cryptography/One_time_pads
- [5] « générateur de nombres réellement aléatoires », Luka Matić, Elektor mars/avril 2017 : www.elektormagazine.fr/150116
- [6] *Shield* OTP Crypto dans le labo d'Elektor : www.elektormagazine.fr/labs/one-time-pad-otp-crypto-system
- [7] Page de l'article : www.elektormagazine.fr/180543-04

télécommande/télémesure en semi-duplex à 2,4 GHz

bon marché et simple grâce aux modules prêts à l'emploi



Beaucoup d'appareils peuvent être télécommandés via un câble (multiconducteur). Le câblage est souvent complexe et le résultat peu esthétique. L'auteur s'est retrouvé avec deux câbles de commande à six broches à remplacer. Ils reliaient sa station de radioamateur (*shack*) à un rotor d'antenne installé sur la charpente. C'était une motivation suffisante pour concevoir un montage d'usage universel offrant quatre voies hertziennes analogiques et numériques avec une communication en semi-duplex.

Le projet rassemble deux unités électroniques, un émetteur près du *shack* et un récepteur près du dispositif de commande du rotor. Ils communiquent par radio. Comme les deux unités ne peuvent émettre et recevoir simultanément, mais chacune leur tour, nous avons affaire à une liaison en semi-duplex, de sorte

qu'il serait plus juste de parler respectivement d'unité frontale (pour envoyer les ordres) et d'unité distante (la commande du rotor) au lieu d'émetteur et de récepteur. Comme l'ensemble échange des mesures de tension numérisées, il s'agit non seulement de télécommande mais aussi de télémesure. La **figure 1**

donne le diagramme de principe du système de télécommande.

Unités frontale et distante

Des deux côtés, le matériel fait principalement appel à des modules tout faits. Tous deux utilisent un Arduino Nano comme contrôleur. Aux deux extrémités,

la liaison radio est confiée à un module d'émission/réception à 2,4 GHz nRF24. L'unité frontale et l'unité distante ne se distinguent que par leur interface utilisateur : l'unité distante (**fig. 2**) dispose d'entrées analogiques et de sorties numériques *bufferisées* ainsi que de quatre LED (pour la phase de test) ; l'unité frontale (**fig. 3**) est équipée de touches et d'un afficheur LCD. Il s'agit là aussi d'un module tout fait, à savoir la carte d'extension *LCD Keypad Shield*.

Arduino Nano

Il n'y a pas grand-chose à dire sur l'Arduino Nano [1][2]. Largement compatible avec le célèbre Arduino Uno, le standard d'entrée de gamme, le Nano est juste plus petit et encore moins cher. Les deux unités sont alimentées par des blocs secteur. L'Arduino Nano accepte toutefois une tension non régulée de 6 à 20 V, mais il chauffera fortement si cette tension est dans le haut de la fourchette, car il ne contient qu'un simple régulateur linéaire. Si la tension d'entrée est de 7 à 9 V, c'est acceptable. Si la sortie du bloc secteur est supérieure, mieux vaut dissiper une partie de l'énergie excédentaire en intercalant une résistance de 47 Ω dans la ligne (ainsi qu'une diode de protection contre l'inversion de polarité).

Module radio nRF24

Ce module radio fonctionne en émission/réception sur 2,4 GHz. Il est utilisé dans bien des applications, dont de nombreux projets Elektor. Son prix modique n'est pas son seul avantage : l'EDI Arduino comprend la bibliothèque nécessaire pour l'exploiter. Le module fonctionne avec une très large autonomie, dès lors que la bibliothèque est importée dans le programme, nous pouvons oublier la gestion du trafic radio. Le bus SPI de l'Arduino Nano est utilisé pour piloter le module nRF24. 128 canaux sont disponibles, bien sûr les deux modules utilisés doivent être réglés sur le même canal, c.-à-d. la même fréquence.

Il existe un très grand nombre de versions de la carte d'évaluation de la puce nRF24L01 de Nordic : avec antenne imprimée ou externe (connecteur SMA), consommation normale ou ultra basse, avec ou sans régulateur 3,3 V embarqué, avec étage final de puissance et préampli d'entrée LNA et tout un tas de configurations de raccordement. La version à utiliser dépend des données spatiales de l'application. Pour transmettre à la

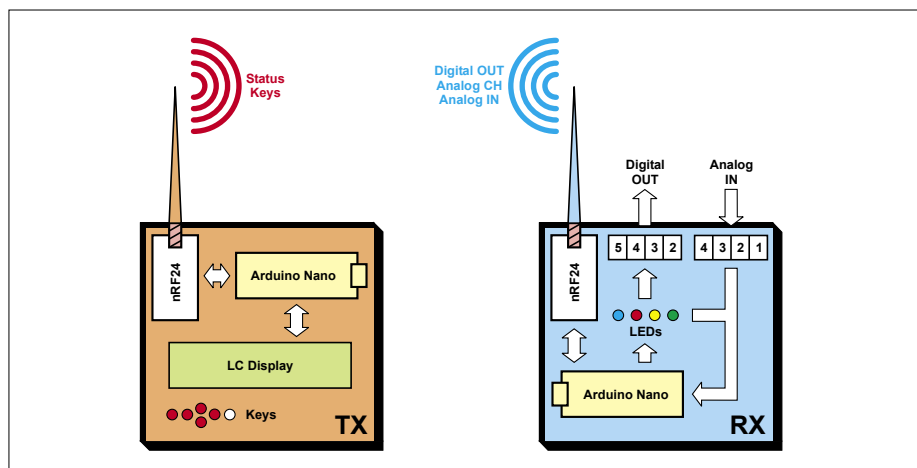


Figure 1. Schéma de principe du système de télécommande/télémètre avec son unité frontale et son unité distante.

charpente à travers un simple plafond de bois, le module à antenne imprimée [3] est plus que suffisant. Si plusieurs étages séparent les deux unités, il faudra se tourner vers une version [4] dotée d'une puissance d'émission et d'une sensibilité maximales. La portée (1000 m, selon

le fabricant) ne peut être atteinte que dans des conditions optimales en extérieur sans obstacle. À tous égards, on peut s'attendre en réalité à une portée de l'ordre de celle obtenue avec un routeur Wi-Fi. D'ailleurs, Nordic ne préconise plus le nRF24L01 pour les nou-

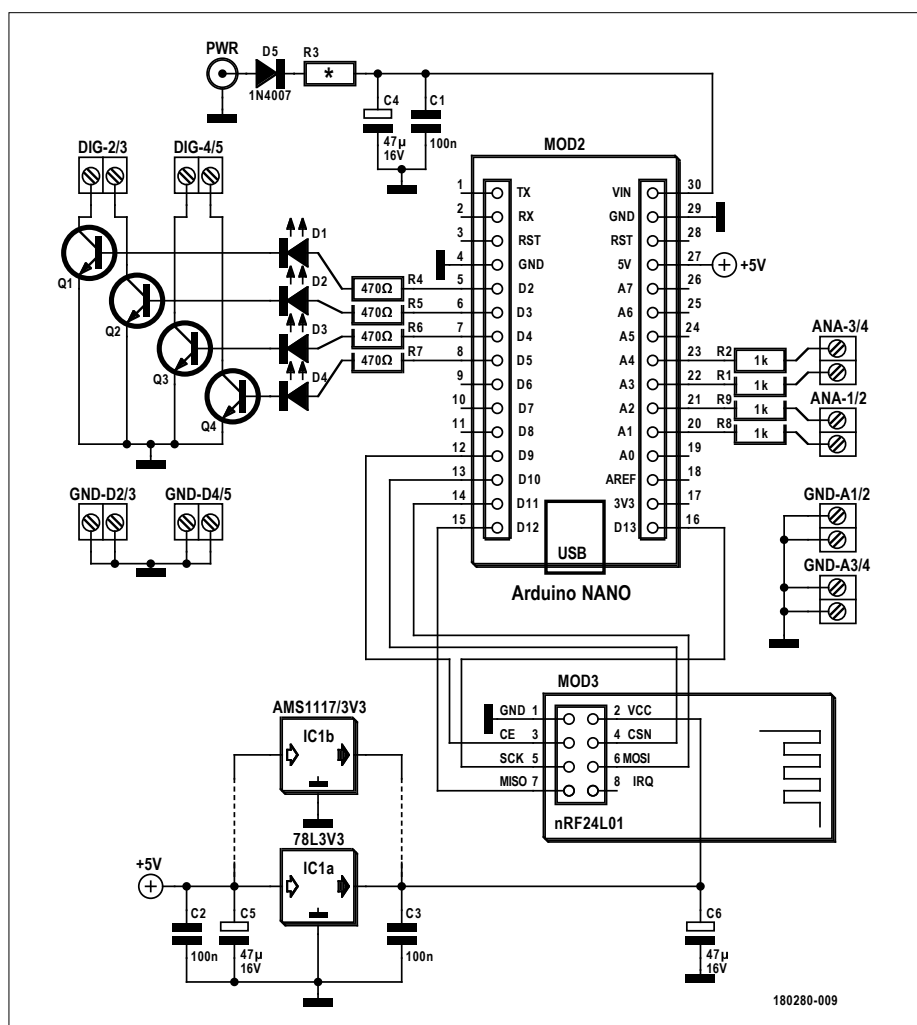


Figure 2. Schéma de l'unité distante avec Arduino et module radio.

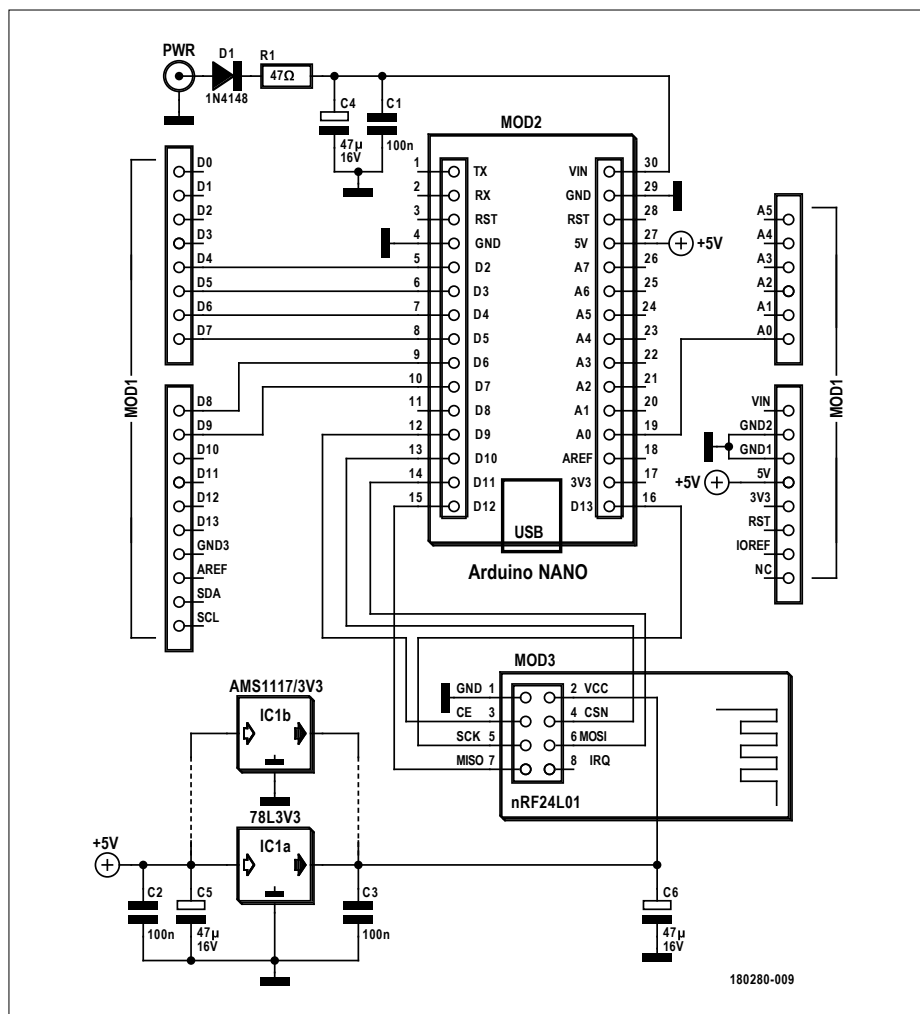


Figure 3. L'unité frontale comprend un Arduino avec carte d'extension « écran & LCD » avec boutons et un module radio.

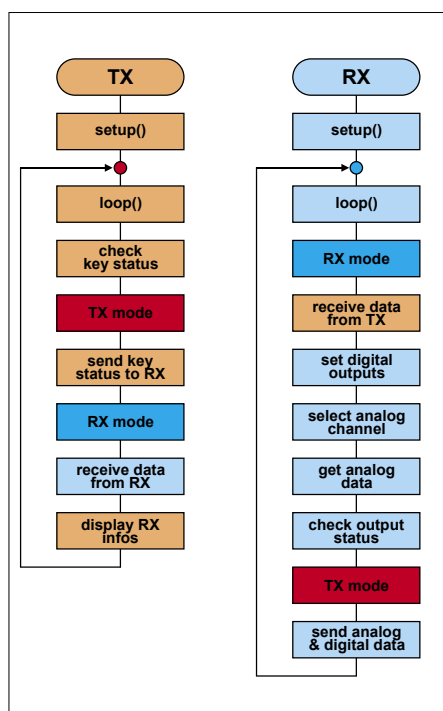


Figure 4. Diagramme des protocoles de communication des deux unités.

veaux projets, mais son successeur, le nRF24L01+, plus puissant. Les stocks d'« anciennes » cartes se trouvant dans le commerce devraient cependant durer encore quelques années.

Le module émetteur/récepteur ne travaille pas en 5 V comme l'Arduino, mais en 3,3 V. En principe, il serait possible de connecter le module émetteur directement sur la sortie régulée 3,3 V de l'Arduino Nano qui peut fournir jusqu'à 50 mA. Cela serait largement suffisant pour le module radio fonctionnant avec l'antenne imprimée (11 à 12 mA), mais le module avec amplificateur nécessite lui 115 mA en émission. C'est pourquoi le module radio est ici alimenté à l'aide d'un régulateur de tension supplémentaire connecté sur la broche 5 V de l'Arduino. À cet égard, il existe d'ailleurs même un module spécialisé, mais nous utiliserons un régulateur de tension tout à fait classique (CMS ou TO92) avec les condensateurs de découplage habituels.

Shield LCD & clavier

La carte d'extension « 1602 LCD Keypad » pour Arduino proposée par ex. par la firme DFRobot [5] de Shanghai s'adapte mécaniquement à l'Arduino Uno, mais aussi sans problème au Nano. L'afficheur de 2×16 caractères n'est pas commandé en série par SPI ou I2C, mais par le bus parallèle. Les broches DB4 à DB7 de l'afficheur sont reliées aux broches D4 à D7 de la barrette d'E/S, et les deux broches de commande *Reset* à D8 et *Enable* à D9. L'afficheur est équipé d'un potentiomètre miniature de réglage du contraste.

Le câblage des cinq boutons-poussoirs SELECT, LEFT, UP, DOWN et RIGHT est inhabituel et surprenant : ils ne sont pas reliés à des entrées tout ou rien, mais comme on peut le voir à gauche du **listage 1**, ils sont câblés ensemble sur l'entrée analogique A0. Ils sont intégrés dans un diviseur résistif de sorte que la tension analogique présente sur l'entrée A0 du contrôleur doit être évaluée pour déterminer quel bouton est actionné. Cela signifie aussi qu'il ne faut enfoncer qu'un bouton à la fois. La carte d'extension possède aussi un autre bouton (RESET) relié à la broche RST de l'Arduino, mais elle n'est pas utilisée dans cette application. Thomas Clausen s'est penché en détail sur cette carte, voir [6].

Protocole de communication en semi-duplex

Le diagramme (**fig. 4**) décrit le déroulement d'une communication entre l'unité frontale (UF) et l'unité distante (UD). En principe, l'UF se contente de détecter la pression d'un bouton et transmet cette information à l'UD qui se charge de la traiter. Les boutons UP, DOWN, LEFT et RIGHT amènent chacun une des sorties binaires à l'état bas ; le bouton SELECT permet de choisir parmi quatre entrées analogiques pour lire l'une de quatre valeurs de mesure possibles.

Après l'initialisation, la boucle principale du croquis de l'UF (**TX.ino**) évalue l'état des boutons : la tension analogique présente sur l'entrée A0 est mesurée, ce qui détermine lequel des cinq boutons a été actionné. Il faut savoir que différentes variantes de cette carte existent dans le commerce et qu'il peut y avoir des différences de valeur des résistances du diviseur de tension. Il est possible qu'il faille changer les seuils du **listage 1** pour que la détection des boutons se fasse correctement. Finalement, le croquis active le

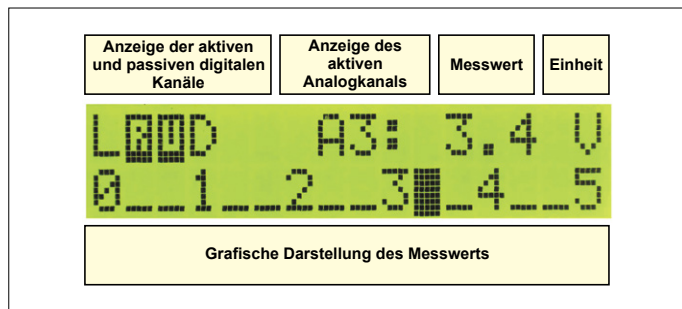


Figure 5. Vue de l'afficheur LCD pour la commande d'un rotor.

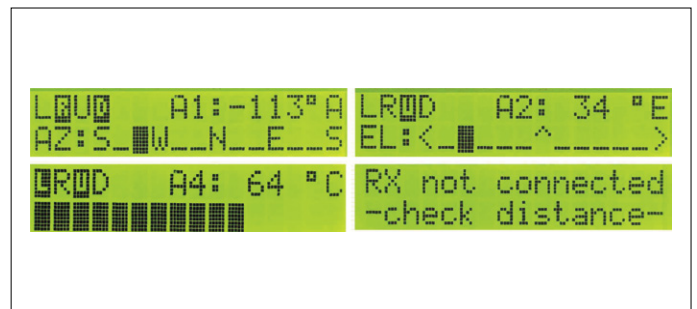


Figure 6. Exemples d'affichage sur le LCD pour différentes applications.

mode émission et l'information représentant le bouton actionné est envoyée à l'UD, puis le mode réception est de nouveau activé.

Après le démarrage du croquis `RX.ino`, l'UD qui se trouve en mode réception interprète l'information reçue en activant soit l'une des quatre sorties tout ou rien, soit l'une des quatre entrées analogiques. Enfin, le croquis lit l'état des quatre sorties tout ou rien, l'entrée analogique activée et la tension présente sur celle-ci. La tension lue est transformée en valeur numérique sur dix bits. L'UD bascule en mode émission et renvoie toutes ces données (sous forme codée) en une salve à destination de l'UF. Après réception, l'UF décode les données et envoie les informations reçues à l'afficheur LCD (**fig. 5**). Lors de cette communication bidirectionnelle, la puce d'émission/réception nRF24L01 vérifie en permanence que la transmission est correcte. En cas de défaillance, un message signalant celle-ci apparaît sur l'écran.

Variantes du logiciel

Les deux programmes téléchargeables sur la page du projet [7] contiennent des exemples pour diverses applications. Initialement, les boutons UP, DOWN, LEFT et RIGHT servaient à déplacer le rotor de l'antenne en mettant les sorties adéquates à l'état bas. Sur le listage de l'UF, nous trouvons par ex. trois fonctions de commutation différentes : LEFT comme poussoir, RIGHT comme commutateur ainsi qu'UP et DOWN comme boutons radio (interrupteurs à action mutuelle opposée). Le bouton SELECT permet de sélectionner tour à tour l'une des quatre entrées analogiques. L'écran LCD affiche aussi bien le numéro de l'entrée analogique que la valeur mesurée. Dans le croquis associé à l'UF, on peut par ex. voir à quoi l'écran peut ressembler lorsqu'il affiche l'azimut ou l'élévation (il n'est pas nécessaire d'utiliser plus de voies analogiques sur un rotor d'antenne). Le croquis montre également comment réaliser l'affichage succinct pour un volt-

mètre ou un thermomètre (**fig. 6**). Pour afficher des caractères spéciaux, l'auteur a utilisé un *générateur de caractères* [8] bien pratique.

Les deux croquis commentés sont vraiment simples et proposent beaucoup de possibilités d'adaptation à vos besoins personnels. Ne tentez pas de modifier les instructions et fonctions concernant la commande du module nRF24, sauf si vous avez l'habitude de ce genre de module radio.

Réalisation

Les schémas électriques et les circuits imprimés ont été réalisés à l'aide du programme EAGLE et sont disponibles sur la page du projet [6]. Il est assez facile de réaliser soi-même les circuits imprimés étant donné la largeur relativement grande des différentes pistes conductrices. Après le câblage des différents circuits, les deux croquis de commande ont été chargés à l'aide de l'EDI Arduino. Les croquis utilisent les bibliothèques `LiquidCrystal.h`, `SPI.h`, `nRF24L01.h` et `RF24.h`. L'une ou l'autre des

Listage 1. Évaluation de l'état des boutons.

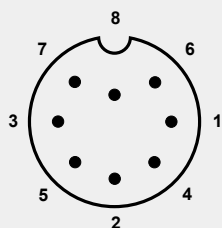
```

void get_key_pushed()
{
    message_to_RX = analogRead(A0)/10; // values from 0 to 102
    delay(100); // key debounce

    if (message_to_RX<5){while(analogRead(A0)<5); message_to_RX = 2;} // Right
    if (message_to_RX> 5 && message_to_RX<20 ) message_to_RX = 3; // Up
    if (message_to_RX>20 && message_to_RX<35 ) message_to_RX = 4; // Down
    if (message_to_RX>35 && message_to_RX<55 ) message_to_RX = 1; // Left
    if (message_to_RX>55 && message_to_RX<85 ) message_to_RX = 5; // Select
}

```


Commande du rotor



- Broche 6 : délivre une tension de 2 à 4,5 V_{cc} selon l'azimut (0° à 450°)
- Broche 1 : délivre une tension de 2 à 4,5 V_{cc} selon l'élévation (0° à 180°)
- Broche 4 : rotation à gauche (actif au niveau bas)
- Broche 2 : rotation à droite (actif au niveau bas)
- Broche 5 : rotation vers le bas (actif au niveau bas)
- Broche 3 : rotation vers le haut (actif au niveau bas)
- Broche 7 : nc
- Broche 8 : masse commune

L'appareil de commande du rotor YEASU G-5600 est équipé de quatre touches de commande : UP et DOWN pour l'élévation et LEFT et RIGHT pour l'azimut. Quand on actionne les touches, elles mettent la ligne de commande à la masse. Ces lignes sont câblées en parallèle à partir des touches et arrivent sur une fiche DIN à 8 broches à l'arrière du boîtier de commande du rotor. En outre, ce boîtier indique le réglage d'azimut et d'élévation sur deux galvanomètres de grande taille. Les tensions analogiques des deux galvanomètres sont retournées par le rotor lui-même. Elles sont situées entre 0 et 4,5 V et correspondent respective-

ment à une élévation de 0° à 180° et un azimut de 0° à 450°. Les deux tensions analogiques sont également disponibles sur la prise DIN placée à l'arrière de l'appareil de sorte qu'une télécommande complète de celui-ci est possible. Rien ne s'oppose donc à ce que l'on dissimule ce boîtier de commande dans les combles afin de n'utiliser qu'un très court câble de liaison au rotor. Le boîtier de commande est piloté par une unité distante qui communique avec une petite unité frontale uniquement par liaison radio ISM.

bibliothèques devrait certainement déjà être installée dans votre EDI Arduino. Si ce n'est pas le cas, vous pouvez vous les procurer par les liens [9][10][11][12] et les installer ensuite dans l'EDI avec *Croquis* → *Charger bibliothèque*.

Comme ce projet repose essentiellement sur des modules tout faits, les circuits imprimés des deux boîtiers peuvent être réalisés en simple face en format Europe, moyennant quelques ponts. Les cinq modules sont disponibles dans les boutiques en ligne pour une somme totale modique de moins de 30 €. Attention aux différentes variantes. En général

les revendeurs chinois ne sont pas très loquaces quant aux caractéristiques des modules qu'ils proposent, de sorte qu'il est parfois compliqué de savoir exactement ce que l'on achète. C'est pour cela qu'il est préférable de choisir un fournisseur mettant à disposition les schémas, la mise en place des composants et des exemples de programmes. ◀

(180280-03 – version française : Yves Georges)

Ce projet a également été publié dans le journal CQ DL d'un club allemand de radioamateurs, le Deutscher Amateur-Radio-Club e.V. (numéro d'avril 2018).

**@ WWW.ELEKTOR.FR**

→ **Module radio**
www.elektor.fr/nrf24l01-150499-91

→ **Arduino Nano**
www.elektor.fr/arduino-nano-3

→ **JOY-iT Nano**
www.elektor.fr/joy-it-nano-v3

Liens

- [1] Arduino Nano : www.elektor.fr/arduino-nano-3
- [2] JOY-iT Nano : www.elektor.fr/joy-it-nano-v3
- [3] Module radio nRF24L01 : www.elektor.fr/nrf24l01-2-4-ghz-wireless-module-8-pin-150499-91
- [4] Module radio nRF2401 avec amplificateur et SMA : www.elec Freaks.com/wiki/index.php?title=2.4G_Wireless_nRF24L01p_with_PA_and_LNA
- [5] Cartes d'extension « 1602 LCD Keypad » : www.dfrobot.com/product-51.html
- [6] Détails de la carte d'extension « LCD Keypad » : www.thomasclausen.net/en/walking-through-the-1602-lcd-keypad-shield-for-arduino/
- [7] Page de l'article : www.elektormagazine.fr/180280-03
- [8] Générateur de caractères pour le LCD : <https://omerk.github.io/lcdchangen/>
- [9] Bibliothèque pour le LCD : <https://playground.arduino.cc/Main/LiquidCrystal>
- [10] Bibliothèque SPI : <https://github.com/PaulStoffregen/SPI>
- [11] Bibliothèque RF24 (1) : www.arduino-libraries.info/libraries/rf24
- [12] Bibliothèque RF24(2) : <https://github.com/maniacbug/RF24>



productronica
fast forward

powered by Elektor

the startup platform

LANCEZ VOTRE STARTUP

SUR
PRODUCTRONICA 2019

○ Participez à l'édition 2019!
12 au 15 novembre 2019 à Munich

informations détaillées :
www.elektormagazine.fr/p-ffwd

Productronica Fast Forward is brought to you by



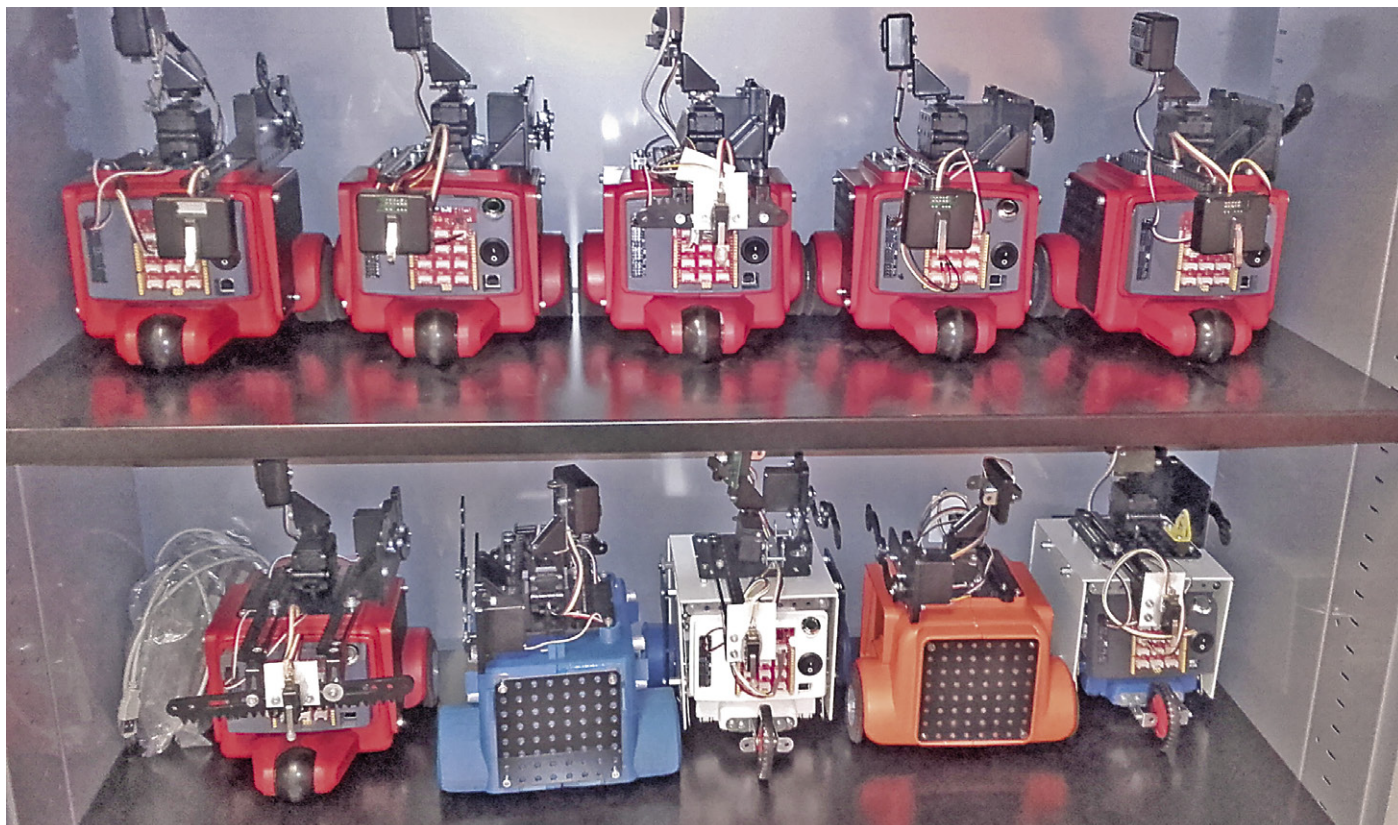
productronica



carte PobDuino

Flowcode et Arduino alliés contre l'obsolescence

Jean-Noël Lefebvre/ooutsidebox (Lyon), Stéphane Huet (Givors)



Vous recherchez un robot programmable et modulable pour travailler avec vos élèves ? Construisez un ou plusieurs robots avec la carte PobDuino. Grâce à elle, vos robots seront pétris de qualités : ils seront programmables en Flowcode (la licence de base suffit) et pourront utiliser des cartes compatibles Arduino pour le branchement de capteurs et d'actionneurs.

Genèse

Stéphane Huet est enseignant en sciences de l'ingénieur au lycée Louis Aragon de Givors (France). Il utilise depuis quelques années le robot *Pob* de la société Pob Technology pour initier ses élèves à la robotique. Le lycée a d'ailleurs participé activement à la conception de la partie mécanique de ce robot à vocation éducative. Le lycée dispose donc d'un parc d'une dizaine de robots, comme beaucoup d'autres écoles en France. Les atouts majeurs de

cette plateforme robotique sont :

- robustesse et modularité mécanique
- nombreux accessoires (servomoteurs et capteurs)
- possibilité de programmer en langage C, et surtout avec le langage graphique ORBEE (ordinogrammes), spécialement développé par Pob Technology

Malheureusement Pob Technology a fermé boutique et *Pob* est maintenant

orphelin. Stéphane a cherché à donner une seconde vie à ses *Pob*. Son cahier des charges imposait entre autres :

- conserver les éléments mécatroniques (mécanique, accus, moteurs, capteurs et actionneurs) ;
- permettre la programmation avec des ordinogrammes, comme avec Orbee ;
- ouvrir la plateforme à l'écosystème Arduino, ce qui permet de bénéfi-

cier de l'offre pléthorique de modules matériels et bibliothèques à code source ouvert.

Commande bicéphale

Pour le logiciel, Scratch est écarté pour deux raisons. Premièrement Scratch ne repose pas sur des ordigrammes, mais plutôt des graphes de type « structogrammes », ce qui ne correspond pas à l'objectif pédagogique. Deuxièmement Scratch ne fonctionne pas de manière autonome sur une carte Arduino, il faut une liaison sans fil avec un ordinateur où Scratch est exécuté (ou alors utiliser Scratch sur une carte Raspberry Pi embarquée).

Après quelques recherches, Jean-Noël propose Flowcode, un candidat sérieux que nombre de lycées utilisent. Toutefois il faut résoudre quelques problèmes :

- L'éditeur de Flowcode parle de compatibilité avec Arduino, mais il ne s'agit que de la faculté à prendre en compte le matériel de certaines cartes Arduino. Par contre, il n'y a pas d'intégration logicielle : il n'est pas possible d'introduire directement les bibliothèques de l'EDI Arduino :- (
- Une fois que vous avez acquis la licence Flowcode de base pour Arduino/AVR, les possibilités restent très limitées. En effet, pour piloter les périphériques (I2C, UART, servos, etc.), il faut soit prendre des licences complémentaires pour accéder aux bibliothèques Flowcode, soit implémenter soi-même ces

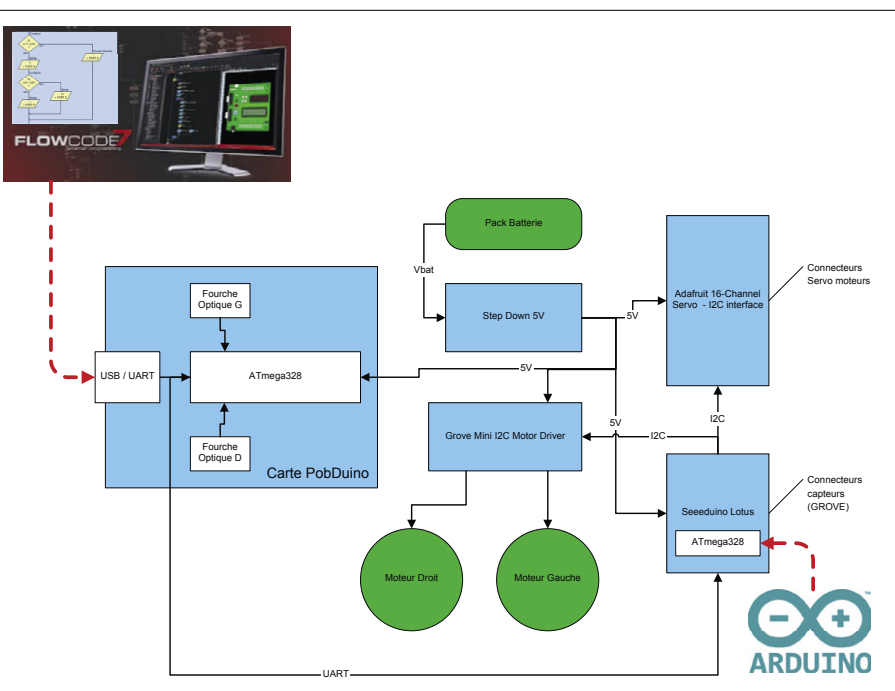


Figure 1. Différents blocs fonctionnels au cœur du système bicéphale PobDuino.

fonctions en langage C puis les intégrer à Flowcode :- (

Pour contourner ces obstacles, l'architecture PobDuino repose sur deux microcontrôleurs ATmega328. Le premier reçoit le code produit avec Flowcode, qui ne fait appel à aucune bibliothèque de périphérique particulière. Toutefois les fonctions de communication UART de base sont implémentées. Le second µC est logé est sur une carte compatible Arduino qui peut exploiter toutes

les bibliothèques et tous les périphériques matériels de l'écosystème Arduino. Une liaison entre les UART des deux µC (simple protocole d'échange de services) permet d'utiliser presque sans limites des ressources Arduino depuis Flowcode. Le synoptique de la **figure 1** donne une vue d'ensemble des différents blocs fonctionnels au cœur du système PobDuino.

Matériel

La seule carte à fabriquer est celle baptisée « PobDuino ». Elle contient :

Principales commandes déjà disponibles

DIGITALREAD, Channel(4-8) → 0/1

Lecture d'une entrée numérique sur le µC de la carte Lotus.

DIGITALWRITE, Channel(8-15), Value(0-100)

Écriture sur une sortie PWM de la carte des servos Adafruit.

ANALOGREAD, Channel(0-3) → 0-100

Lecture d'une entrée analogique du µC de la carte Lotus. Dans ses travaux pratiques, Stéphane utilise cette entrée pour lire les capteurs de suivi de ligne.

DISTREAD, Channel(0-3) → D(cm)

Lecture (en cm) d'un capteur de distance Sharp GP2Y0A21, connecté à une entrée analogique du µC de la carte Lotus.

USREAD, Channel(4-7) → D(cm)

Lecture (en cm) d'un capteur de distance à ultrasons de type Grove (réf. 101020010), connecté à une broche numérique du µC de la carte Lotus.

SERVO, Channel(0-7), Degres(0-180), Speed(0-31/32)

Commande angulaire d'un servo moteur branché sur une sortie de la carte Adafruit.

MOTOR, Side(L/R), Sens(F/R), Speed(0-100)

Commande d'un moteur branché sur la carte de pilotage des moteurs Grove I2C.

MOVE, Sens(F/R), Dist(mm 0-30000), Speed(0-100)

Commande du déplacement en ligne droite du robot (distance mesurée par les roues codeuses / fourches optiques).

ROTATE, Sens(L/R), Angle(1-360)

Commande du pivotement du robot (angle mesuré par les roues codeuses / fourches optiques).

STOP

Arrêt de toute commande de moteur en cours.

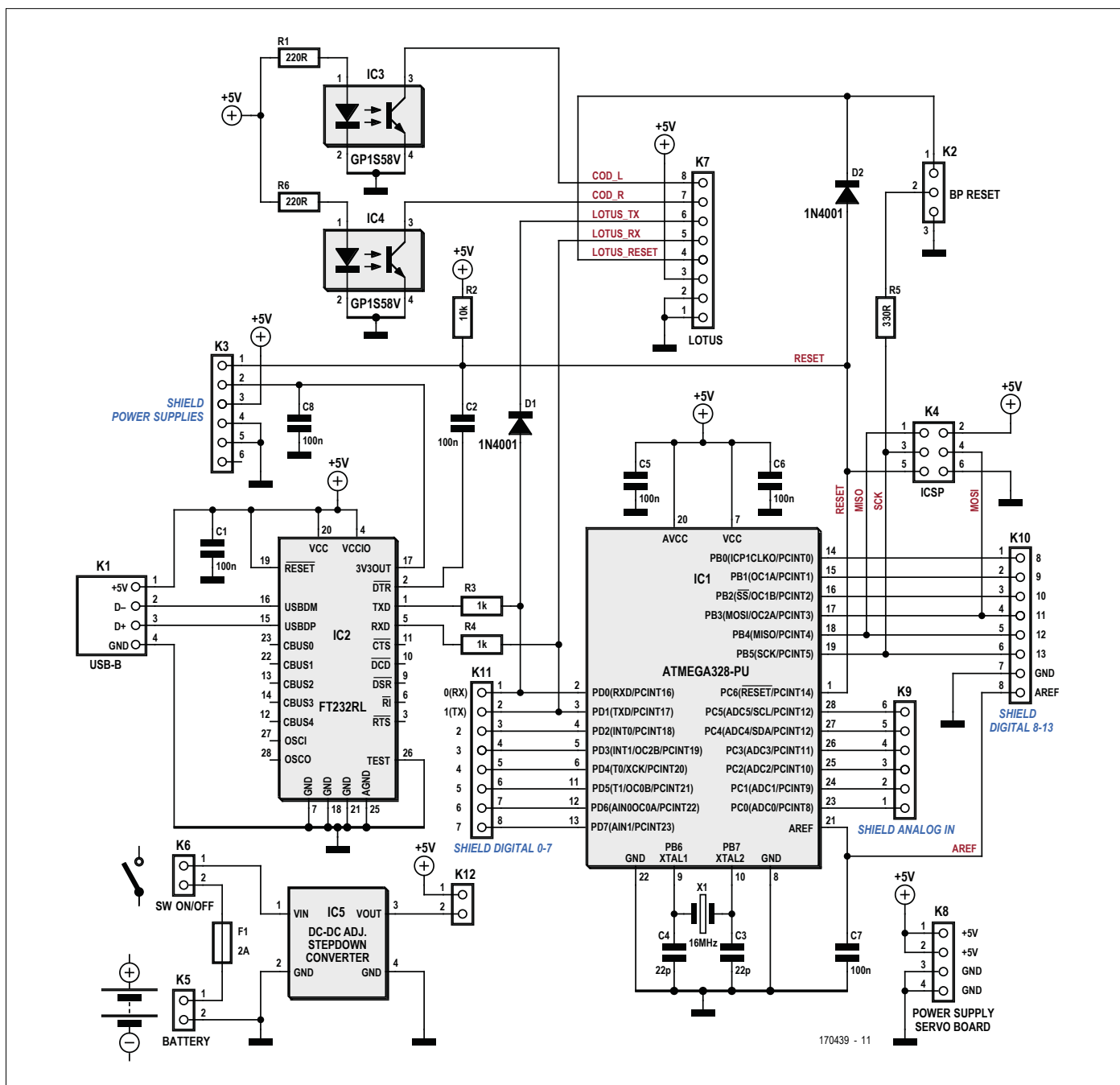


Figure 2. Schéma de la carte PobDuino.

- µC ATmega328
- fourches optiques (pour les roues codeuses des moteurs)
- interface FTDI pour la programmation par USB

Cette carte interprète le code Flowcode, mais elle envoie aussi des commandes à différentes parties du robot. Son schéma (**fig. 2**) est directement inspiré des cartes Arduino Uno (ancienne version). Au cœur de cette carte se trouve le fameux microcontrôleur ATmega328P, accompagné de son quartz à 16 MHz (X1) et des conden-

sateurs de découplage d'alimentation C5, C6 et C7.

La programmation de type ICSP est possible via le connecteur K4, par ex. pour flasher le chargeur d'amorce Arduino. On trouve aussi les incontournables connecteurs au format « shield » : K3, K9, K10 et K11. Pour ce qui est de la communication UART par USB, la mission est confiée à un classique circuit spécialisé FT232RL, c'est le seul composant monté en surface un peu délicat à souder.

Le reste des composants ne correspond pas à un Arduino classique, mais à des fonctions spécifiques au PobDuino : IC5

est un module convertisseur DC/DC qui fournit une tension de 5 V à partir du pack de batteries. Quant à IC3 et IC4, ce sont les fourches optiques qui permettent le comptage des tours de roues du robot.

Les autres cartes du robot sont disponibles dans le commerce.

La carte Seeeduino Lotus comporte le second µC ATmega328P ainsi que douze connecteurs Grove ; on peut donc la relier directement aux capteurs de la série Grove [2]. Une carte d'asservissement Adafruit [3] pilote les servos. Un pilote de moteurs Grove I2C [4] com-

L'assemblage ainsi que la réunion de toutes les cartes sont décrits par le menu et avec de nombreuses photos dans le fichier « Assemblage du PobDuino » à télécharger dans la page de l'article [5]. La **figure 3** donne une vue d'ensemble des liaisons entre les différentes cartes.

Les deux μC ATmega328P dialoguent entre eux via la liaison série UART. Le μC de la carte PbdUino envoie des commandes pour animer le robot, exécutées par le μC de la carte Lotus (un acquittement est envoyé après chaque commande exécutée).

Sur la carte Lotus, il faut téléverser le croquis Arduino `POBDUINO_B.ino` qui fait appel à quelques bibliothèques. La carte est alors prête à exécuter les commandes envoyées soit depuis le moniteur série de l'EDI Arduino, soit par le µC de la carte PopDuino avec Flowcode.

série (RàZ automatique). Le texte « Hello

Jean-Noël Lefebvre, au sein de sa société OOTSIDEBOX, réalise des travaux à façon : maquettes interactives, preuves de concept, prototypes, petits effets spéciaux. Il a également une activité de conseil et développement en électronique (matériel et logiciel). Enfin, il anime des ateliers d'initiation à l'électronique et en particulier à Arduino. Voir www.ootsidebox.com

www.linkedin.com/in/stéphane-huet-ab552a61/

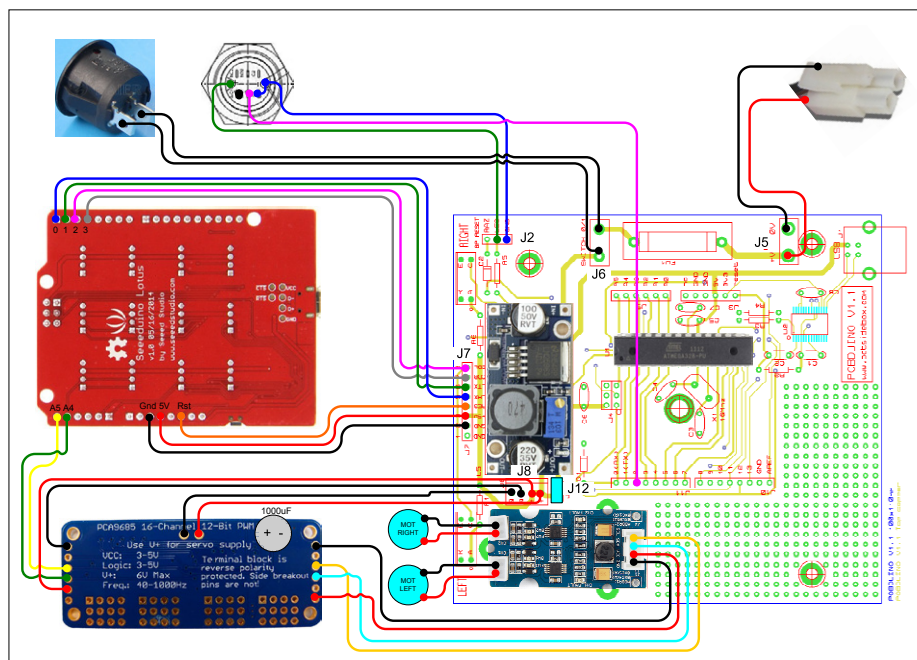


Figure 3. Liaisons entre les différentes cartes.

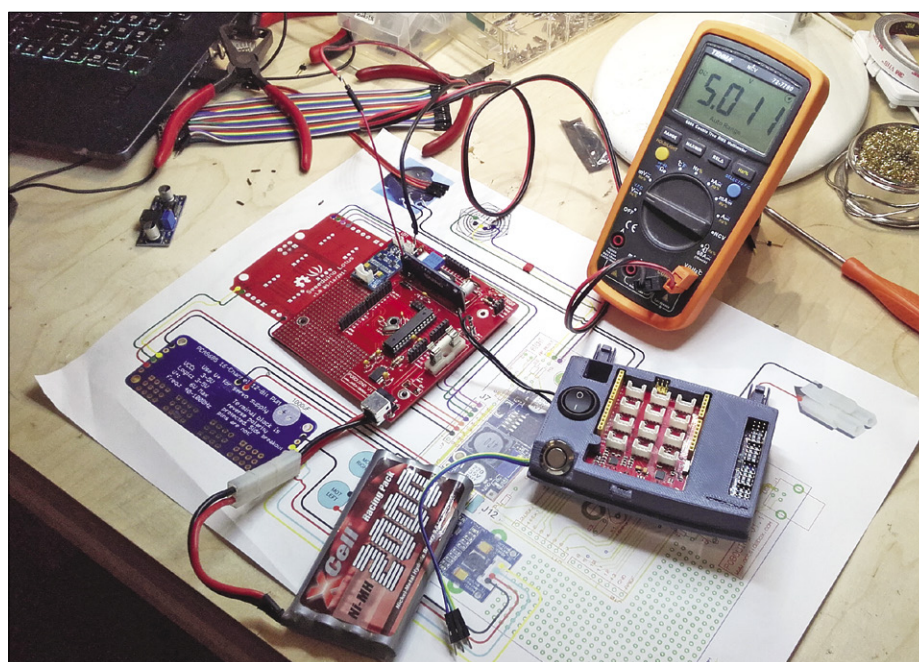


Figure 4. Réglage du convertisseur de tension.

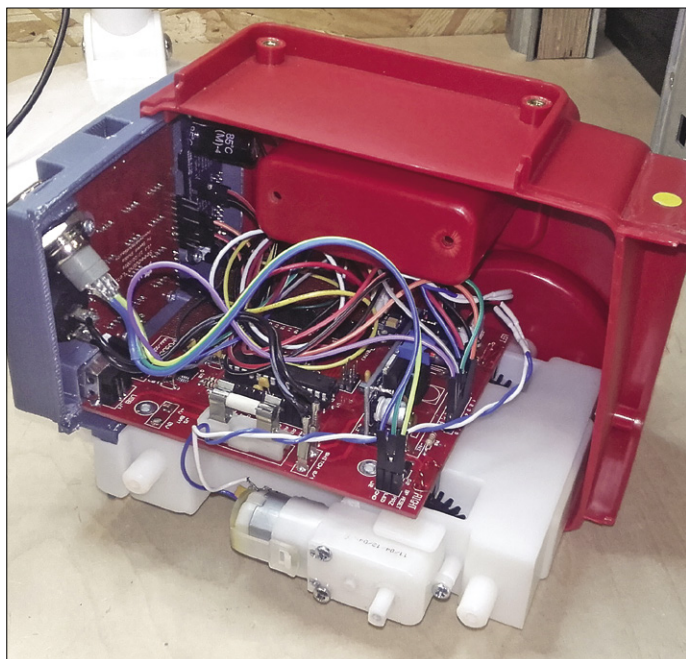


Figure 5. Pobduino avec ses entrailles à l'air.

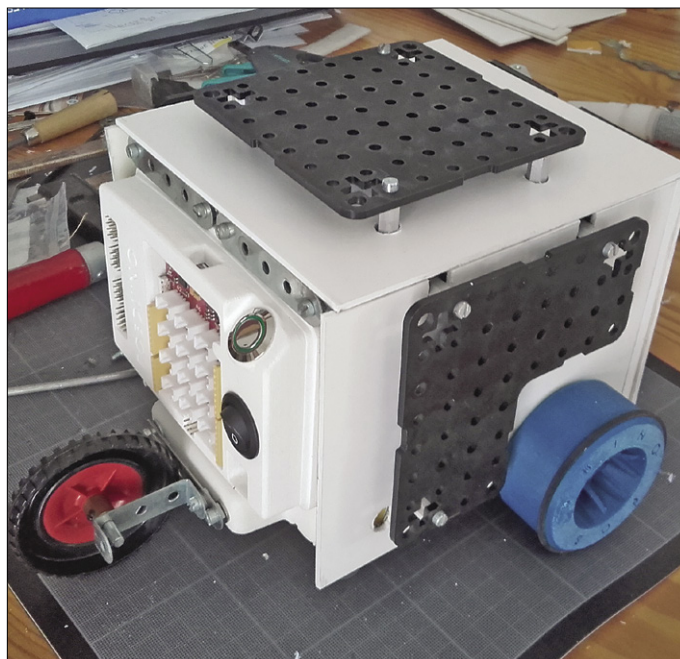


Figure 6. Carcasse de Pobduino bricolée avec des pièces de Meccano et des plaques en plastique blanches.

World » devrait ensuite s'afficher sur le terminal. Le fichier « Programmation de Pobduino dans Flowcode.docx » à télécharger en [5] guide les premiers pas dans Flowcode et détaille les différentes commandes disponibles.

Si ces commandes ne répondent pas à vos besoins, pour en ajouter de nouvelles, il « suffit » de les coder dans le croquis `POBDUINO_B.ino` et de créer les macros flowcode correspondantes. Cela ouvre les portes à l'utilisation de toute une gamme de capteurs et actionneurs pour Arduino, entre autres Grove de Seeedstudio : lecture de MP3, pilotage de LED NeoPixels, lecture de puces RFID, capteurs de lumière....

Conclusion

Grâce à la carte Pobduino à code source ouvert, les clients de Pob Technology

peuvent ressusciter leurs robots Pob. Si vous n'avez pas de Pob, ce n'est pas un souci : Stéphane fournit le fichier d'impression en 3D pour remplacer la face arrière du robot Pob original. Vous pourrez bricoler le reste de la carcasse avec ce que vous aurez sous la main (**fig. 6**). Il est également possible d'utiliser d'autres plateformes mécaniques existantes (Pololu par exemple). Enfin pour les enseignants, le dossier « Initiation programmation » (cf. [5]) contient

sept programmes d'initiation à la programmation conçus pour des élèves de classe de seconde en sciences de l'ingénieur, Stéphane les met ainsi à la disposition de ses collègues. ◀

(170439-01)

Stéphane remercie le lycée Louis Aragon pour le financement de ce projet, ainsi que la plateforme technologique IDPRO (en particulier Marc Benbahi) pour la mise à disposition d'imprimantes 3D [7].



@ WWW.ELEKTOR.FR

→ Kit de démarrage Arduino, compatible Grove, réf. 17724
www.elektor.fr/grove-starter-kit-for-arduino

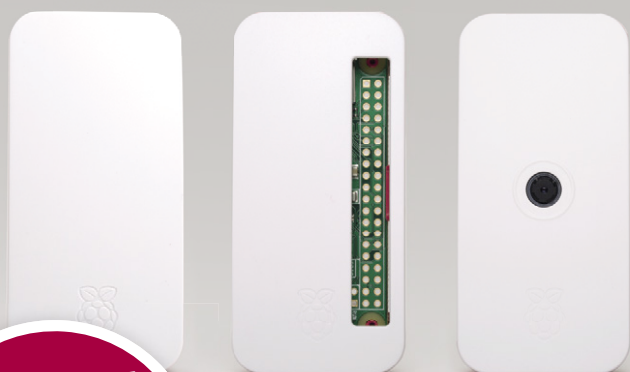
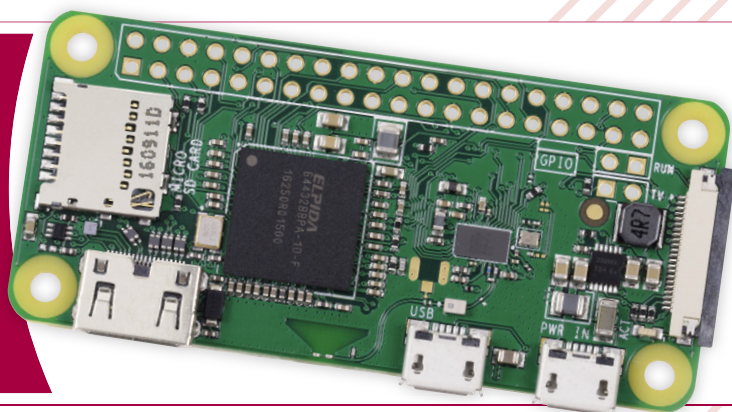
Mini-lidar ToF, compatible Grove, réf. 18436
www.elektor.fr/seeedstudio-grove-tf-mini-lidar-laser-ranging

Liens

- [1] Découvrir Pobduino : hackaday.com/2017/07/27/pobduino-makes-the-most-of-grove et hackaday.io/project/26009-robot-pobduino
- [2] Système Grove : wiki.seeedstudio.com/Grove_System/
- [3] Carte pour les servos « Adafruit 16-Channel 12-bit PWM/Servo Driver - I2C interface - PCA9685 » : www.adafruit.com/product/815 & learn.adafruit.com/16-channel-pwm-servo-driver/downloads
- [4] Carte pour les moteurs « Grove - I2C Mini Motor Driver » : wiki.seeedstudio.com/Grove-Mini_I2C_Motor_Driver_v1.0/
- [5] Page de l'article : www.elektormagazine.fr/170439-01
- [6] Installation du chargeur d'armoire Arduino : www.arduino.cc/en/Tutorial/ArduinoISP
- [7] IDPRO : www.plateforme-idpro.fr

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

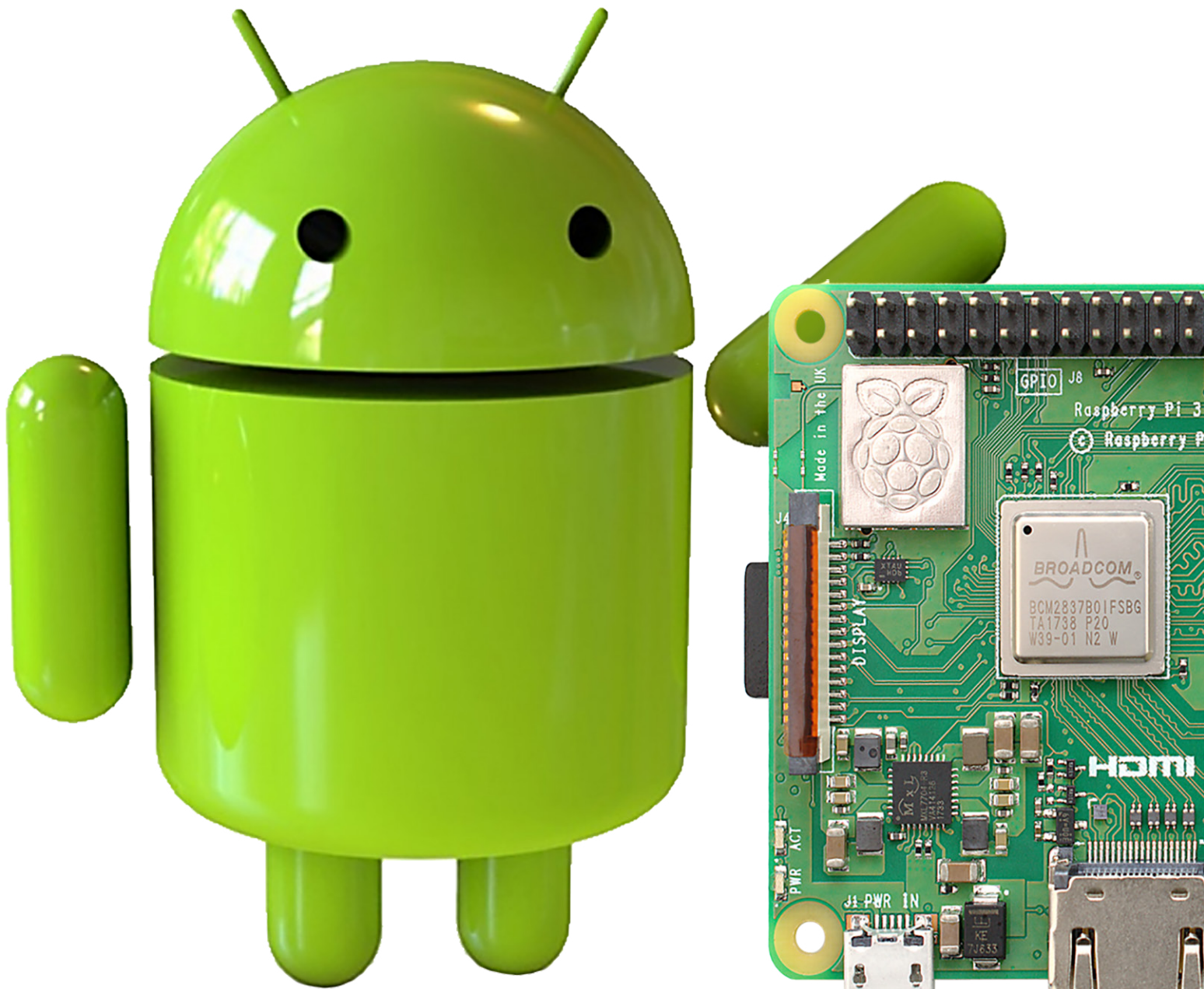
Android Things prend son envol grâce au langage C++

Tam Hanna

La dernière fois qu'Elektor a parlé d'Android Things [1][2], le système d'exploitation était en cours de développement. Entretemps, Google a achevé son travail et propose une variante de type API stable. Cet article décrit brièvement les changements pertinents. Comme toujours, c'est un Raspberry Pi 3 de la première génération qui nous servira d'hôte. Nous aurons également besoin d'un compte Google.

Android Things 1.0 a subi de nombreux changements : en plus d'un nouveau système de permissions et de modifications dans l'API GPIO, Google en a profité pour lancer une interface de

programmation en C++ et améliorer le système de mise à jour. Les étapes qui suivent requièrent un Raspberry Pi 3, une souris, un clavier, un écran et un accès à l'internet. Allons tout



d'abord dans la console de gestion Android Things disponible en [3]. C'est elle qui nous permettra de configurer des périphériques et de compiler des images [4]. Pour les expériences qui suivent, il convient de « créer un produit » dans la console Android Things.

Passons maintenant au modèle créé automatiquement par Google pour composer une configuration de type *build*. La « gestion de modèles » est une concession de Google pour les développeurs qui souhaitent gérer plusieurs révisions matérielles ou logicielles dans une même catégorie de projet.

Comme indiqué dans les derniers articles, la composition de la configuration prend du temps. Une fois que le travail est terminé, téléchargez l'image de développement et installez-la comme à l'accoutumée à l'aide de la carte SD [1].

Nouvelle API GPIO

Ouvrez Android Studio pour créer un nouveau projet. À partir de V3.2, l'EDI offre une option pour Android Things dans le générateur de projet pour créer un projet approprié. Sélectionnez *Android Oreo* comme système cible et créez une activité du type *Android Things Empty Activity*.

Android Things implique l'octroi de permissions lorsque le code à exécuter recourt au matériel. Pour nos expériences avec les broches GPIO, il suffit de saisir la permission suivante dans *AndroidManifest.xml* :

```
<xml version="1.0" encoding="utf-8"?>
<manifest . . .>
    <uses-permission android:name="com.google.android.
        things.permission.USE_PERIPHERAL_IO" />
```

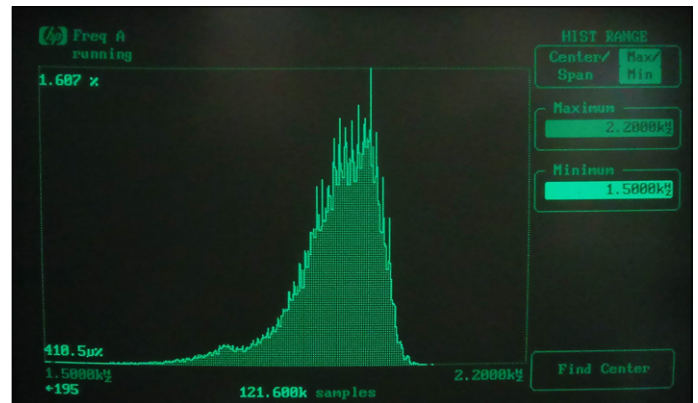
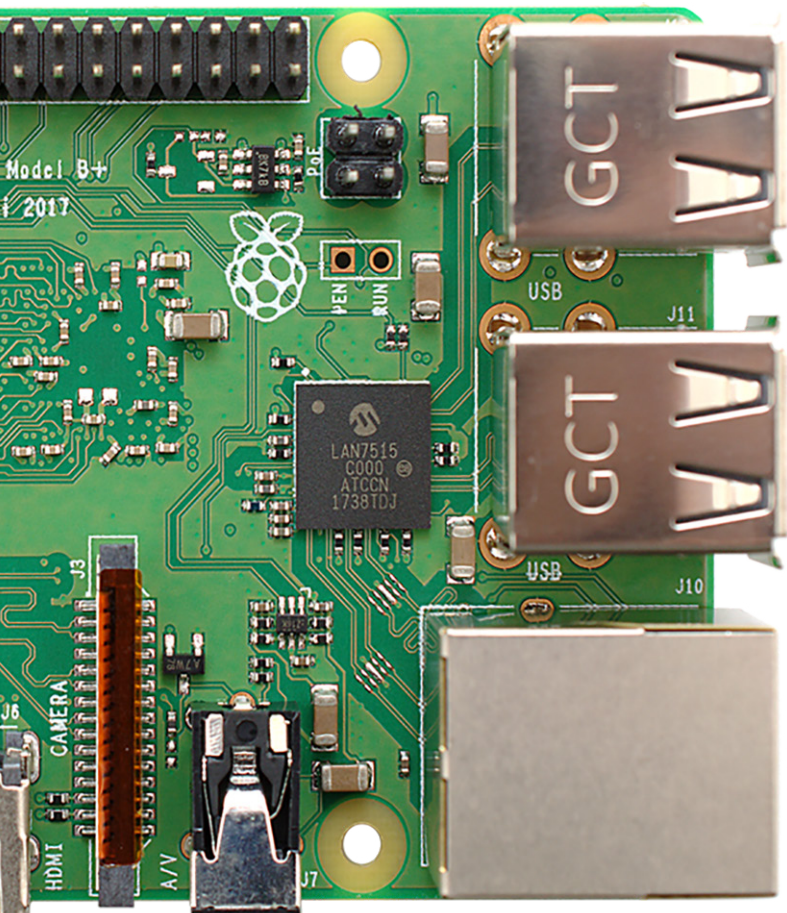


Figure 1. Ce n'est pas la plus stable des formes d'onde.

À l'instar des systèmes d'exploitation, Android Things surveille le *thread* de l'interface utilisateur graphique (GUI). Créez une classe dérivée de *thread* qui sera lancée dans la méthode *onCreate* de *MainActivity*.

Comme dans les articles précédents, nous voulons produire un signal rectangulaire pour avoir une idée de la vitesse d'exécution et de la stabilité temporelle. Le code suivant est nécessaire pour obtenir la forme d'onde caractéristique :

```
public void run() {
    Gpio myGPIO0;
    try {
        PackageManager manager =
            PackageManager.getInstance();
        myGPIO0 = manager.openGpio("BCM2");
```

Android Things 1.0 se distingue de son prédécesseur par le fait que l'initialisation des éléments matériels se déroule différemment. Au lieu du *PeripheralManagerService*, il existe un service de plateforme à part entière qui fonctionne comme n'importe quel autre service système Android. Il s'ensuit que *getInstance()* crée l'instance à utiliser.

Le reste de la fonction se présente comme à l'accoutumée :

```
        myGPIO0.setDirection(Gpio.
            DIRECTION_OUT_INITIALLY_LOW);
        while(1==1){
            myGPIO0.setValue(true);
            myGPIO0.setValue(false);
        }
    }
    catch (Exception e) { }
```

En connectant un analyseur de domaine de modulation à un Raspberry Pi 3 cadencé à 1,2 GHz, on obtient le résultat de la **figure 1**.

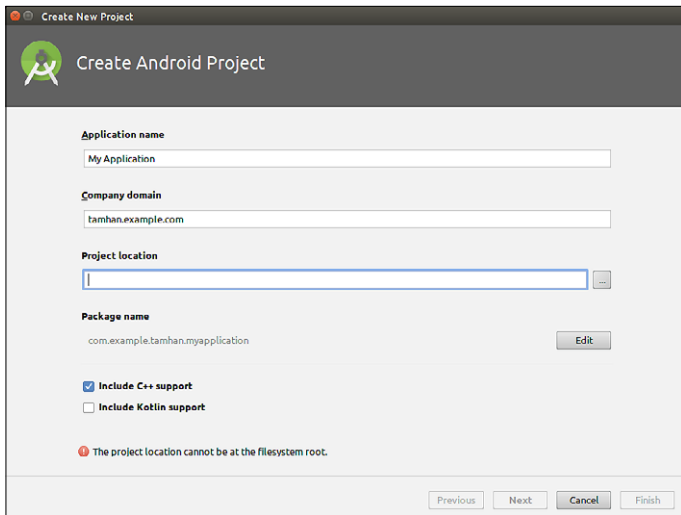


Figure 2. Case à cocher pour activer le support de C++.

Passons maintenant au C++

Si vous avez grandi avec des systèmes embarqués Java comme MicroEJ, votre instinct vous dira que c'est l'implémentation JNI qui posera problème. JNI signifie *Java Native Interface*. Ce système permet aux programmes fonctionnant dans la machine virtuelle Java d'appeler des méthodes natives. Il est indiscutable que l'échange de paramètres entre le code Java et le code natif ne brille pas par sa rapidité. Google propose donc également une API C++ pour remédier, en théorie, au problème. Voyons si nous pouvons obtenir un signal rectangulaire digne de ce nom. Le développement en C++ ne fait pas partie de la distribution standard. Dans Android Studio, ouvrez l'assistant d'installation disponible sous *Tools SDK Manager* et allez dans la rubrique *SDK Tools*. Sélectionnez ensuite l'entrée *NDK* (kit de développement natif) pour télécharger les composants supplémentaires (env. 1 Go).

Certes, Android Studio permet d'inclure le support C++ dans les projets existants, mais pour les solutions contenant peu de code, l'effort n'en vaut pas la peine. Il est préférable de démarrer avec un nouveau modèle de projet. Assurez-vous que la case C++ est cochée. Android Studio utilisera alors uniquement un modèle de projet adapté aux compilateurs C++ (**fig. 2**). Par rapport aux autres projets, un projet C++ se distingue par le fait que l'assistant affiche une étape supplémentaire. Dans le champ standard C++, vous devez choisir *Toolchain Default* afin de sélectionner le compilateur existant pour éviter des problèmes supplémentaires durant l'installation. Un clic sur *Finish* et Android Studio crée, comme à l'accoutumée, un nouveau projet. Celui-ci contiendra une partie native en plus

Firefox reste persona non grata

Si vous travaillez avec la console Android Things et que vous rencontrez des bizarreries « inexplicables », vérifiez le navigateur que vous utilisez. Google n'a toujours pas résolu ses conflits avec Firefox, Edge et Safari - l'expérience montre que les choses se passent mieux avec Google Chrome.

des éléments Java connus.

Les installations NDK récemment téléchargées sont souvent incomplètes. Dans un premier temps, vous devrez effectuer une resynchronisation ou une compilation de l'application. Si *Gradle* détecte l'absence de composantes, un message d'erreur apparaît vous invitant à les télécharger. Dans la plupart des cas, il suffit de cliquer sur le lien indiqué dans le message d'erreur. Android Studio se charge du reste à condition d'avoir une connexion à l'internet.

Comme l'API C d'Android Things n'est que provisoire au moment où nous publions ces lignes, cette interface ne fait pas partie du NDK. Ouvrez plutôt l'URL [5] dans un navigateur pour télécharger la toute dernière version. Au moment de publier cet article, le fichier porte le nom `native-libandroidthings-1.0.0.zip`. Le fichier compressé contient un dossier appelé `native-libandroidthings-1.0` qui renferme l'API Android Things compilée.

Il convient de copier ce dossier dans celui qui contient le fichier `CMakeLists.txt`. Il s'agit d'un fichier de configuration qui pilote la compilation des parties natives du projet. Sur son poste de travail, l'auteur a nommé ce projet *ElektorCGPIO*. Le chemin pertinent s'appelle donc `ElektorCGPIO/app`.

À ce stade, il est recommandé de redémarrer Android Studio afin de mettre à jour l'image du projet. En effet, l'EDI prend son temps pour détecter les changements dans le système de fichiers.

À l'étape suivante, cliquez sur `External Build Files CMakeLists.txt` pour charger le fichier *makefile* de la partie native. Explication : passé une certaine taille, les projets C et C++ ne sont plus compilés manuellement, mais à l'aide d'un fichier *makefile*. C'est une sorte de fichier de contrôle qui indique les étapes nécessaires à l'outil de compilation qui s'appelle lui aussi *make*.

Il convient de modifier la structure du projet pour intégrer notre bibliothèque. On ajoute tout d'abord le passage spécifié par Google au-dessous de la ligne `cmake_minimum_required(VERSION 3.4.1)` - n'oubliez pas d'adapter le chemin à vos besoins :

```
set(CMAKE_MODULE_PATH $ /home/tamhan/Desktop/
stuff/2018September/ElektorATINews/ElektorCGPIO/app/
native-libandroidthings-1.0)
find_package(AndroidThings REQUIRED)
include_directories($)
```

Ce bout de code (*snippet*) à première vue déroutant a pour objet d'ajouter au chemin de chargement du module de l'outil le dossier contenant les composantes précédemment téléchargées et décompressées. Lors de la compilation d'un programme, *CMake* cherche les bibliothèques natives dans divers répertoires afin de les inclure, si besoin est.

Vous y trouverez un autre fichier `include` qui s'occupe d'intégrer la bibliothèque dans le reste du processus de compilation. Plus bas, vous trouverez une structure qui énumère les bibliothèques à charger. Elle aussi a besoin d'être adaptée pour préparer Android Things :

```
target_link_libraries(
    native-lib
    $
    $)
```

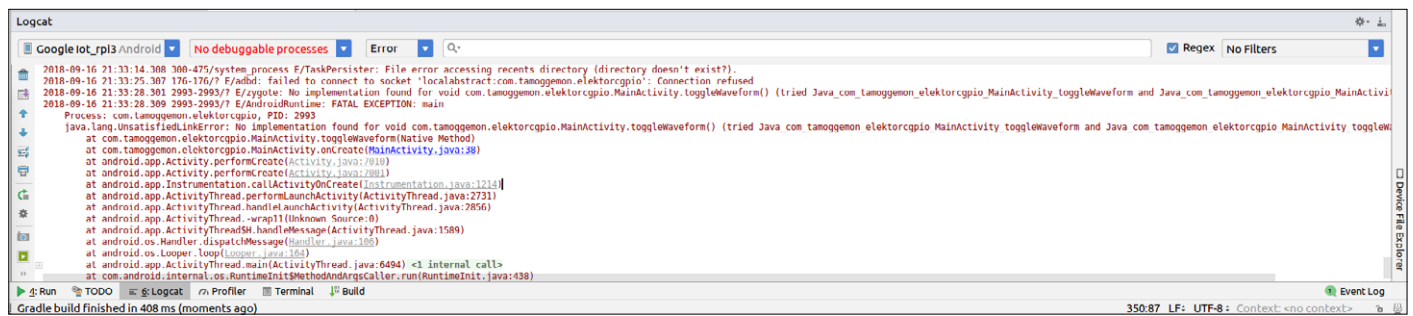


Figure 3. Android Studio affiche le nom de la fonction manquante.

La compilation de la partie native d'un projet Android Studio n'est pas seulement configurée dans *CMakeLists*. Passez maintenant au fichier *build.gradle* du module *App* et entrez un bloc *NDK* dans la partie Android :

```
android {
    compileSdkVersion 27
    defaultConfig {
        ndk {
            abiFilters 'armeabi-v7a', 'x86'
        }
    }
}
```

Étant donné qu'Android Studio et Android Things reposent sur Java, ils sont indépendants de l'architecture CPU au niveau applicatif. Dans le cas de notre API, il s'agit d'un module pré-compilé disponible sous forme de code en assembleur. Nous devons limiter les architectures de processeurs cibles à ARM et x86 – Google ne voit pas l'intérêt de continuer à supporter d'autres processeurs comme MIPS.

Pour pouvoir travailler avec des bibliothèques natives, des modifications sont également nécessaires dans *MainActivity*. Tout d'abord, il y a maintenant un constructeur statique :

```
public class MainActivity extends Activity {
    static {
        System.loadLibrary("native-lib");
    }
}
```

Les constructeurs statiques sont une version spéciale du constructeur standard, appelés lorsque le programme est lancé. Les constructeurs statiques utilisent la fonction *LoadLibrary* pour créer un lien entre le temps d'exécution et la bibliothèque créée par le C.

La référence au temps d'exécution est importante, car lors de la compilation d'un projet combiné Java & C, le compilateur ne vérifie pas si les méthodes saisies dans la page Java sont implémentées dans la page C++. Pour cette raison, il est de la plus haute importance de tester le code basé sur JNI du mieux possible pour parer aux fâcheuses erreurs d'exécution. L'appel à proprement parler de la méthode native s'effectue comme suit :

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

```
...
TextView tv = (TextView) findViewById(R.
id.sample_text);
tv.setText(stringFromJNI());
}
public native String stringFromJNI();
```

N'oubliez surtout pas qu'en Java une méthode native doit être déclarée en se servant du mot clé *native*. Cette ligne signale au compilateur que le développeur certifie avoir créé la méthode. Si le bloc est absent, des erreurs se produiront durant la compilation.

Une question de nom !

Java et C sont des langages de programmation qui ne sauraient être plus différents. Alors que Java fonctionne avec la « mémoire gérée » et fournit des classes d'abstraction complexes notamment pour les chaînes, le développeur C quant à lui programme en fonction du matériel.

Comme une introduction complète à JNI prendrait trop de place, nous préférons vous renvoyer au tutoriel disponible en [6]. Il n'en reste pas moins que, dans la pratique l'« adaptation » des types de données n'est pas la principale préoccupation du développeur. Trouver la bonne syntaxe s'avère plus exaspérant, comme en témoigne le nom très compliqué de la méthode *stringFromJNI* utilisée ci-dessus dans le code C :

```
extern "C" JNIEXPORT jstring JNICALL
Java_com_tamoggemon_elektorcgpio_MainActivity_
stringFromJNI(
    JNIEnv *env,
    jobject /* this */) {
    std::string hello = "Hello from C++";
    return env->NewStringUTF(hello.c_str());
}
```

En plus des différents indicateurs qui signalent que la méthode doit être exportée, vous trouverez également la chaîne *Java_com_tamoggemon_elektorcgpio_MainActivity_stringFromJNI*. Ce nom créé par le compilateur Java permet de détecter la méthode.

Android Studio offre une solution de contournement pour identifier les noms de méthode corrects. Ouvrez *MainActivity.java* et procédez à l'adaptation selon le schéma suivant :

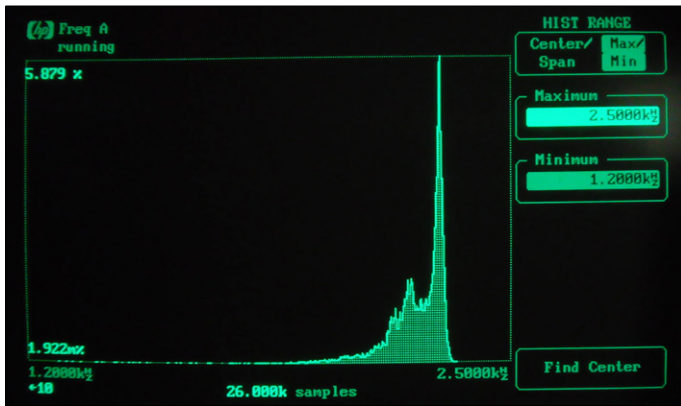


Figure 4. Même en C++, le programme n'est pas beaucoup plus rapide.

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

public native void toggleWaveform();
```

Le code qui vient d'être créé appelle la méthode `toggleWaveform`. Comme cette dernière n'a pas encore été créée, il est clair que l'exécution du code sur le Raspberry Pi 3 conduit à une erreur. Cliquez quand même sur *Play* pour lancer le programme via le pont de débogage Android (ADB).

Le programme clignote un court instant à l'écran, avant de s'afficher à nouveau dans le lanceur Android Things. Dans Android Studio, passez à la section *LogCat* pour régler le filtre sur *Error* dans la liste déroulante afin de contraindre Android Studio à masquer tous les messages qui ne sont pas des erreurs critiques. La fonction manquante apparaît alors à l'écran comme illustré à la **figure 3**.

Les informations que nous venons de recueillir nous permettent de mettre en place une structure méthodologique de base :

```
#include <jni.h>
#include <string>
#include <pio/gpio.h>
#include <pio/peripheral_manager_client.h>
extern "C" JNIEXPORT void JNICALL
Java_com_tamoggemon_elektorcgpio_MainActivity_
toggleWaveform(
    JNIEnv *env,
    jobject /* this */) {
```

LogCat

Un appareil Android en service émet en permanence des messages collectés par le système d'exploitation. Dans Android Studio, mais aussi dans divers autres outils de développement, il existe une fenêtre appelée *LogCat*, que vous pouvez utiliser pour évaluer le contenu du journal.

Les paramètres `JNIEnv` et `jobject` sont toujours présents dans les méthodes JNI, vous pouvez les ignorer. Pour les tâches plus complexes, autorisez l'accès à l'objet d'aide JNI qui vous permet d'accéder aux services d'exécution. Par exemple, JNI retourne des objets qui assurent l'interaction avec des objets ou des variables Java.

L'API Android Things est intégrée grâce aux deux `include` qui commencent par `pio`.

Dans l'étape suivante, nous créons la méthode d'édition réelle. Votre code se présente comme suit :

```
extern "C" JNIEXPORT void JNICALL
Java_com_tamoggemon_elektorcgpio_MainActivity_
toggleWaveform(
    JNIEnv *env,
    jobject /* this */) {
    APeripheralManagerClient* client =
    APeripheralManagerClient_new();
    AGpio* gpio;
    APeripheralManagerClient_openGpio(client, "BCM2",
    &gpio);
    AGpio_setDirection(gpio,
    AGPIO_DIRECTION_OUT_INITIALLY_LOW);
    while(1==1)
    {
        AGpio_setValue(gpio, true);
        AGpio_setValue(gpio, false);
    }
}
```

Ici, rien de spectaculaire, nous utilisons l'API C++ décrite en détail en [7] pour construire un objet « broche GPIO » et obtenir une forme d'onde caractéristique.

Théoriquement, notre programme est prêt à être lancé. Mais voilà que Raspberry Pi se rebiffe, car un de nos autres programmes accapare déjà la broche de port BCM2, ce qui bloque l'accès à notre nouveau programme.

La façon la plus simple de résoudre le problème est de connecter une souris à l'ordinateur de traitement et de procéder à une réinitialisation dans la section *System*. L'autre solution consiste à se mettre sur ADB et supprimer le programme du Raspberry Pi d'une manière plus « douce ».

Nous pouvons maintenant relancer notre programme modèle pour étudier de plus près le comportement en fréquence.

Le diagramme fourni par l'analyseur de domaine de modulation de la **figure 4** montre que l'amélioration est très marginale.

En regardant de plus près l'interface de programmation d'applications Android Things (ou en étudiant les doléances des développeurs), on constate que même en C++ l'accès au matériel est effectué au travers du système de fichiers SYSFS de Linux. L'interaction avec le matériel est ainsi assurée, mais elle est loin d'être performante. Les API comme WiringPi et Cie écrivent directement dans les registres du processeur. Or la procédure est impossible sous Android Things parce que nos programmes ne peuvent pas être dotés des droits de super-utilisateur.

Pour « augmenter les performances », on pourrait travailler directement avec le système de fichiers SYSFS. Ceci permettrait d'augmenter légèrement les performances, mais dans la pratique, l'effort n'en vaut pas la peine.

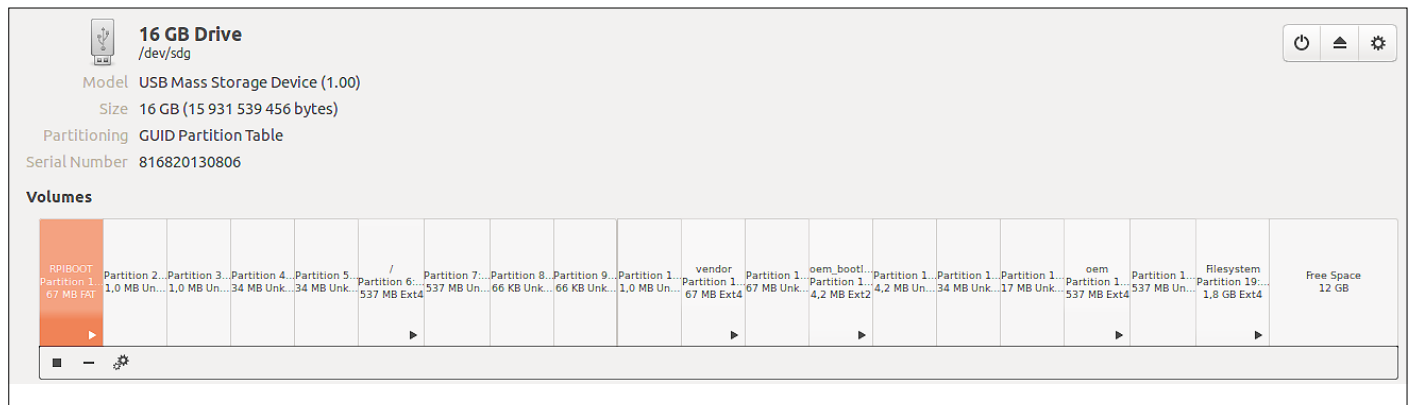


Figure 5. Presque toutes les partitions sont dupliquées, ce qui rend les mises à jour plus sûres.

Mise à jour automatique ?

Si vous êtes abonné aux alertes de sécurité informatique, vous n'échapperez pas à celles qui signalent le piratage des dispositifs de l'IdO. Ce fléau est aggravé par le fait que les utilisateurs négligent de mettre à jour leurs appareils.

Pour parer à cela, Google veut se servir de la console Android Things : les mises à jour régulières ont pour objet de faciliter la tâche des développeurs. Comme cette fonction peut s'avérer très utile dans le cadre d'une exploitation commerciale, nous aimerions vous présenter ici le nouveau concept de mise à jour. Le principe des mises à jour A/B du système (*A/B system updates*) reste le même - la **figure 5** montre qu'il existe un double de toutes les partitions de la carte SD. Les mises à jour entrantes sont dirigées vers l'une des partitions, tandis que l'autre garantit que l'ordinateur continue de fonctionner. Si la mise à jour échoue, Android Things passe tout simplement à la dernière version fonctionnelle pour éviter la panne totale.

Les appareils Android Things se connectent ensuite à un *channel*. Par défaut, Google déclare qu'ils sont au nombre de quatre à démarrer leur vie dans le « canal stable ». Pour être modifié, ce canal requiert une remise à zéro qui efface alors toutes les données de l'utilisateur.

Les nouvelles fonctions d'Android Things 1.0 incluent une API à part entière qui permet au développeur, entre autres, de spécifier quand et comment les mises à jour seront effectuées. Cette façon de procéder *a priori* tatillonne est néanmoins importante puisqu'elle permet d'éviter qu'une mise à jour n'efface les mesures que vous avez accumulées.

Par manque de place, nous préférons vous envoyer vers une implémentation pratique de cette interface de mise à jour, voir en [8].

Conclusion

Les petites modifications apportées à l'API GPIO dans Android Things 1.0 ne seront pas forcément bien accueillies par les développeurs : il aurait été préférable que le code existant puisse être réutilisé plus facilement.

Avec cette nouvelle version, Google renforce la tendance déjà observée dans les versions précédentes : maximisation de la sécurité du système plutôt que maximisation des performances, surtout lorsqu'il y a accès direct à du matériel.

Le système d'exploitation est un environnement confortable et stable pour tous ceux qui font leurs premiers pas dans la création d'interfaces utilisateur sans effort. Si vous recherchez un système d'exploitation en temps réel, Android Things, même dans sa version finale, ne sera pas votre premier choix. ◀

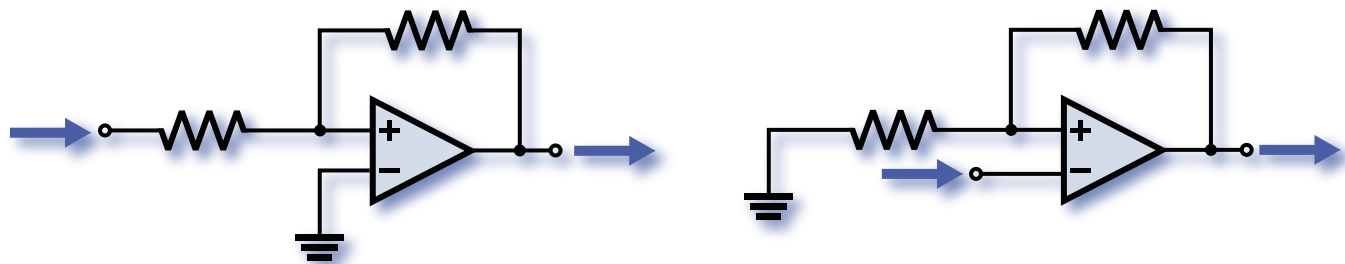
(180305-04 – version française : Pascal Duchesnes)

Liens

- [1] Android sur Raspberry Pi (1), Elektor 3-4/2017 : www.elektormagazine.fr/160361
- [2] Android sur Raspberry Pi (2), Elektor 5-6/2017 : www.elektormagazine.fr/160369
- [3] Console de gestion Android Things : <https://partner.android.com/things/console>
- [4] Documentation relative à la console Android-Things : <https://developer.android.com/things/console>
- [5] API C : <https://github.com/androidthings/native-libandroidthings/releases>
- [6] Tutoriel JNI : <https://developer.android.com/ndk/guides/concepts>
- [7] Description de l'API C++ : <https://developer.android.com/things/sdk/pio/native>
- [8] Mise à jour API : <https://developer.android.com/things/sdk/apis/update>

trigger de Schmitt à amplis op

calculs et circuits



Un trigger (ou bascule) de Schmitt est un circuit classique en électronique. Rien de bien passionnant pourrait-on penser. Un ampli op, quelques résistances, et le tour est joué ? C'est vrai si l'on se contente d'une estimation grossière de l'hystérésis et des seuils de commutation. En revanche si l'on souhaite définir ces paramètres avec précision et par exemple les optimiser par rapport à la série E usuelle, alors calculs et circuits adéquats sont requis. C'est précisément l'objet de cet article.

Volker Schmidt (Allemagne)

Si l'on n'y touche pas souvent, le calcul de triggers de Schmitt à base d'amplis op n'est pas si simple. Une approche des calculs de ces circuits consiste à définir la tension de référence à l'aide de la résistance de rétroaction. Souvent la tension de référence ne se situe pas au centre de la fenêtre de l'hystérésis, en outre les calculs donnent souvent des valeurs tordues pour lesquelles il n'existe pas de diodes Zener entre autres. Pour produire la tension de référence, on se rabat alors sur un diviseur de tension composé de valeurs de résistances des séries E ou une résistance ajustable. Il serait plus pratique de créer la tension de référence à l'aide de diodes Zener, de sources de tension de référence ou de simples diviseurs de tension avec des résistances de la série E12, et de calculer le

circuit en fonction de la tension de référence. Avec des circuits classiques, cela n'est pas possible aisément, mais avec une modification mineure, on peut dans un premier temps définir tension de référence et seuils de commutation, puis dans un deuxième temps calculer les autres composants. Le coût en reste très limité, car seuls trois composants supplémentaires sont requis. Cette approche est possible avec les triggers de Schmitt inverseurs et non-inverseurs.

Trigger de Schmitt non-inverseur

Le comportement du trigger de Schmitt non-inverseur de la **figure 1** est déterminé par la tension aux bornes de R2 et sa plage de tension de sortie. On obtient les seuils de commutation du trigger de Schmitt en ajoutant ou retranchant l'hystérésis ΔU_H à U_{R2} . La tension de référence (**fig. 2**) est idéalement cen-

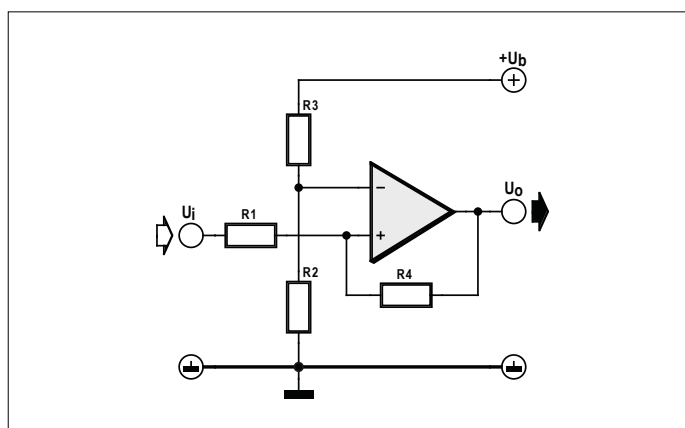


Fig. 1. Le trigger de Schmitt classique non-inverseur à ampli op.

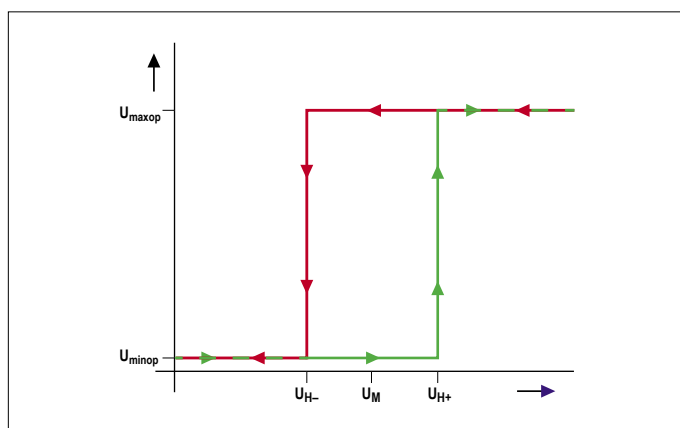


Fig. 2. Diagramme d'hystérésis d'un trigger de Schmitt idéal.

trée entre le seuil de commutation supérieur (U_{H+}) et le seuil de commutation inférieur (U_{H-}), mais ce n'est pas impératif. On doit en outre veiller à ne pas définir une tension de référence trop proche des seuils de commutation. Si on calcule la tension de référence U_{R2} du circuit classique de la figure 1, on s'aperçoit qu'il n'est pas possible de choisir librement un seuil de commutation. Les valeurs U_{H+} et U_{H-} sont liées à la référence non symétrique. Ces valeurs sont théoriquement égales et symétriques seulement si en cas d'alimentation symétrique, l'entrée inverseuse est reliée à la masse, ou en cas d'alimentation non symétrique, elle est reliée à une tension égale à exactement la moitié de la tension de sortie de l'ampli op.

L'origine de ce comportement est le courant qui traverse R4 et R1. Celui-ci dépend de la différence de tension entre la tension de sortie et la tension sur l'entrée inverseuse. Les équations suivantes sont valables seulement si l'intensité du courant reste la même dans les deux états de commutation :

$$U_{H+} = U_{R2} + \Delta U_{H+}$$

$$U_{H-} = U_{R2} - \Delta U_{H-}$$

$$|\Delta U_{H+}| = |\Delta U_{H-}|$$

$$I_{R4} = \frac{U_{maxop} - U_{R2}}{R4} = -\frac{U_{minop} - U_{R2}}{R4}$$

$$I_{R1} = -I_{R4}$$

$$\Delta U_{H-} = -I_{R4} \times R1$$

Sinon, on peut spécifier la valeur de U_{H+} par exemple, mais alors on ne peut plus agir sur la valeur de U_{H-} . La valeur de I_{R4} ne sera alors pas la même pour les deux états de sortie. C'est facile à voir en calculant les paramètres du circuit de la figure 1 pour un ampli op de type CA3140.

Les paramètres initiaux sont :

$U_m = U_{R2} = 1,7 \text{ V}$	$U_B = 9 \text{ V}$	$\Delta U_{H+} = 0,2 \text{ V}$
$U_{H+} = 1,9 \text{ V}$	$U_{maxop} = 6,8 \text{ V}$ (mesuré)	
$U_{H-} = 1,5 \text{ V}$ (souhaité)	$U_{minop} = 0,01 \text{ V}$ (mesuré)	

Calcul du **diviseur de tension** R2/R3 :

$$\frac{R2}{R3} = \frac{U_{R2}}{U_{R3}}$$

$$R2 = R3 \times \frac{U_{R2}}{U_{R3}}$$

Si l'on choisit 10 kΩ pour R3, on obtient :

$$R2 = 10 \text{ k}\Omega \times \frac{1,7 \text{ V}}{7,3 \text{ V}} = 2,33 \text{ k}\Omega$$

Cette valeur sera obtenue avec 2,2 kΩ + 100 Ω. Comme les tensions d'entrée de l'ampli op restent inchangées à la commutation, on a aussi :

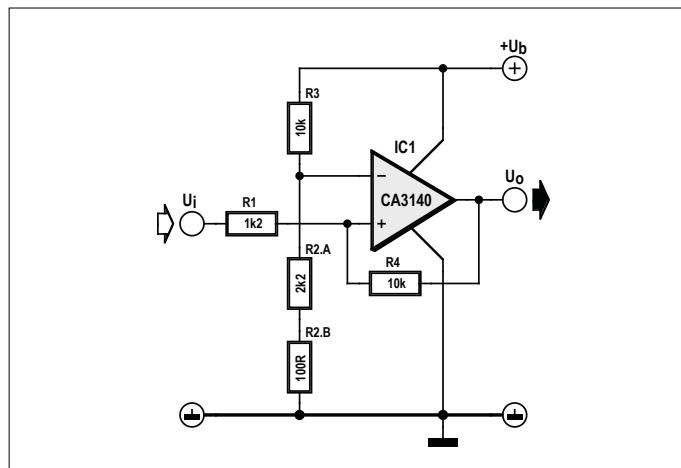


Fig. 3. Le circuit classique et dimensionné d'un trigger de Schmitt non-inverseur.

$$U_p = U_n$$

$$U_{R4} = U_{minop} - U_{R2}$$

$$U_{R4} = 0,01 \text{ V} - 1,7 \text{ V} = -1,69 \text{ V}$$

Si la résistance R4 a une valeur de 10 kΩ, on a :

$$I_{R4} = \frac{U_{R4}}{R4} = \frac{-1,69 \text{ V}}{10 \text{ k}\Omega} = -169 \mu\text{A}$$

und:

$$I_{R1} = -I_{R4}$$

On en déduit la valeur de R1 avec la formule :

$$R1 = \frac{\Delta U_{H+}}{I_{R1}} = \frac{0,2 \text{ V}}{169 \mu\text{A}} = 1,183 \text{ k}\Omega$$

Une valeur de 1,2 kΩ est donc une bonne approximation pour R1. Déterminons à présent U_{H-} , le **seuil de commutation inférieur** :

$$U_{R4} = U_{maxop} - U_{R2}$$

$$U_{R4} = 6,8 \text{ V} - 1,7 \text{ V} = 5,1 \text{ V}$$

$$I_{R4} = \frac{U_{R4}}{R4} = \frac{5,1 \text{ V}}{10 \text{ k}\Omega} = 510 \mu\text{A}$$

$$R1 = \frac{\Delta U_{H-}}{I_{R1}}$$

$$\Delta U_{H-} = R1 \times I_{R1}$$

$$I_{R1} = -I_{R4}$$

$$\Delta U_{H-} = 1,2 \text{ k}\Omega \times -510 \mu\text{A} = -0,612 \text{ V}$$

$$U_{H-} = U_{R2} - \Delta U_{H-} = 1,088 \text{ V}$$

Le seuil de commutation inférieur réel diffère considérablement de la valeur souhaitée. La **figure 3** montre le circuit classique

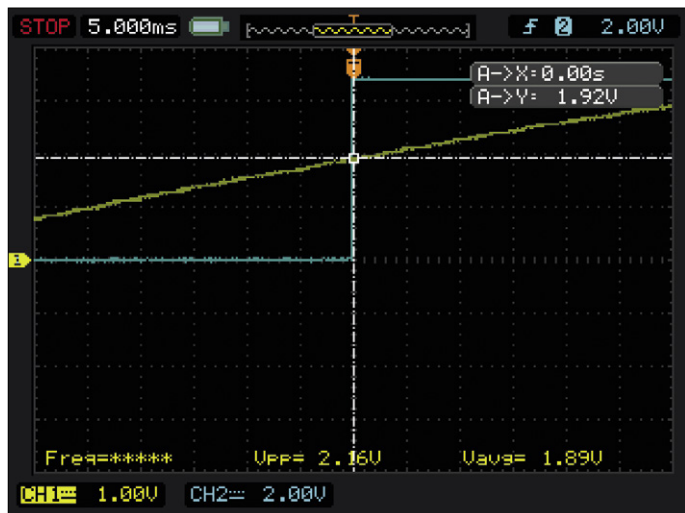


Fig. 4. Oscillogramme du seuil de commutation supérieur du circuit classique non-inverseur.

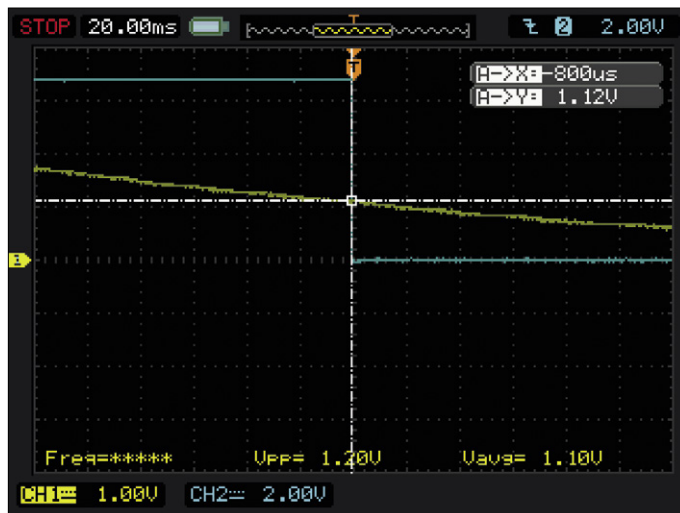


Fig. 5. Oscillogramme du seuil de commutation inférieur du circuit classique non-inverseur.

avec les valeurs des composants. Le calcul et les deux oscillogrammes (**fig. 4** et **fig. 5**) montrent que, en théorie comme en pratique, les seuils de commutation ne sont pas symétriques par rapport à la référence.



Commuter avec précision

Circuit modifié

Le remède serait un couplage rétroactif distinct pour les seuils de commutation inférieur et supérieur. On peut alors choisir librement les deux seuils par un choix approprié des résistances. À cet effet, comme le montre la **figure 6**, il faudra ajouter deux diodes Schottky en plus des deux résistances de couplage rétroactif. Les diodes Schottky, par exemple de type 1N5817, sont utilisées en raison de leur faible chute de tension. La résistance R4 définit le seuil de commutation supérieur, R5

le seuil inférieur. Il est ainsi facile d'adapter l'hystérésis aux exigences. Les paramètres suivants sont utilisés pour le calcul du circuit (là aussi avec un CA3140) :

$U_m = U_{R2} = 1,7 \text{ V}$	$U_B = 9 \text{ V}$	$U_{FD} = 0,21 \text{ V}$ (mesuré)
$U_{H+} = 1,9 \text{ V}$	$U_{\max op} = 6,8 \text{ V}$ (mesuré)	$R_3 = 10 \text{ k}\Omega$
$U_{H-} = 1,5 \text{ V}$	$U_{\min op} = 0,01 \text{ V}$ (mesuré)	$\Delta U_H = 0,2 \text{ V}$

Calcul du **diviseur de tension** R2/R3 :

$$R2 = R3 \times \frac{U_{R2}}{U_{R3}} = 10 \text{ k}\Omega \times \frac{1,7 \text{ V}}{7,3 \text{ V}} = 2,33 \text{ k}\Omega$$

La résistance R2 pourra là aussi être réalisée avec des résistances de 2,2 kΩ et 100 Ω.

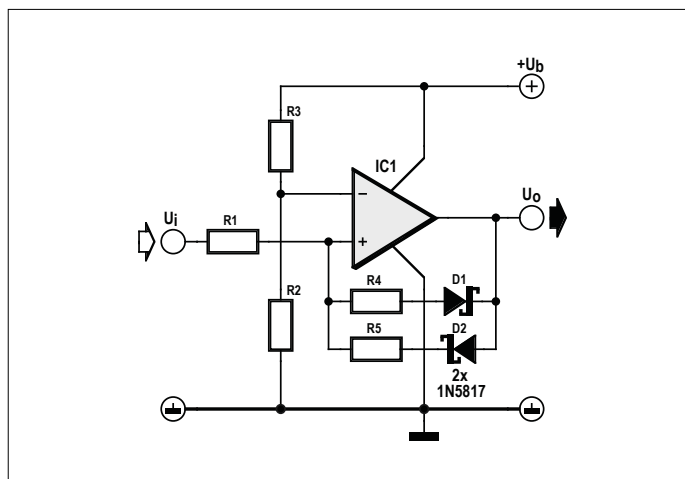


Fig. 6. Principe de modification d'un trigger de Schmitt non-inverseur.

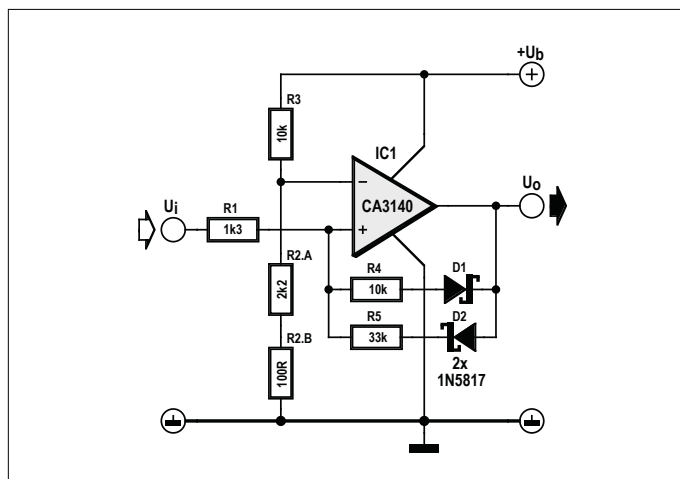


Fig. 7. La modification dimensionnée du trigger de Schmitt non-inverseur.

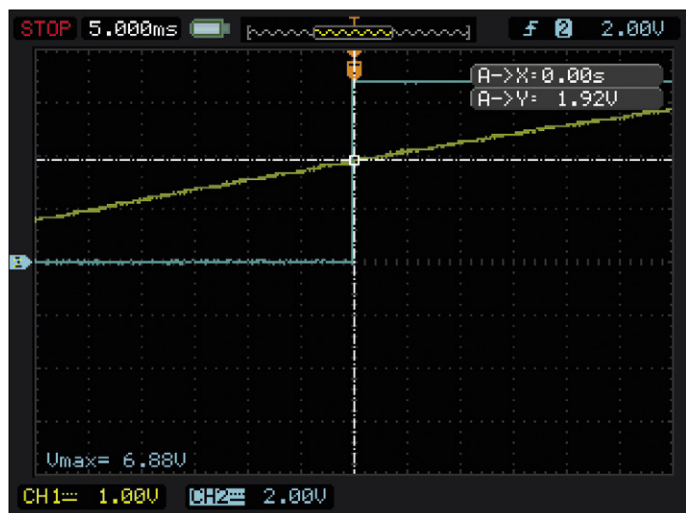


Fig. 8. Oscillogramme du seuil de commutation supérieur de la modification non-inverseuse.

Calcul du **seuil de commutation supérieur** :

$$U_{R4} = U_{minop} + U_{FD} - U_{R2} = 0,01V + 0,21V - 1,7V = -1,48V$$

On obtient un courant I_{R4} de $-148 \mu A$ avec $R4 = 10 k\Omega$. Comme les tensions sur les entrées de l'ampli op sont égales, on a :

$$U_p = U_n$$

et aussi :

$$I_{R1} = -I_{R4}$$

La tension sur l'entrée non-inverseuse doit être supérieure d'une valeur de :

$$\Delta U_{H+} = U_{H+} - U_{R2}$$

afin que le trigger de Schmitt bascule et que sa sortie passe à U_{maxop} . On obtient la valeur de $R1$ par l'équation :

$$R1 = \frac{\Delta U_{H+}}{I_{R1}} = \frac{0,2V}{148 \mu A} = 1,351 k\Omega$$

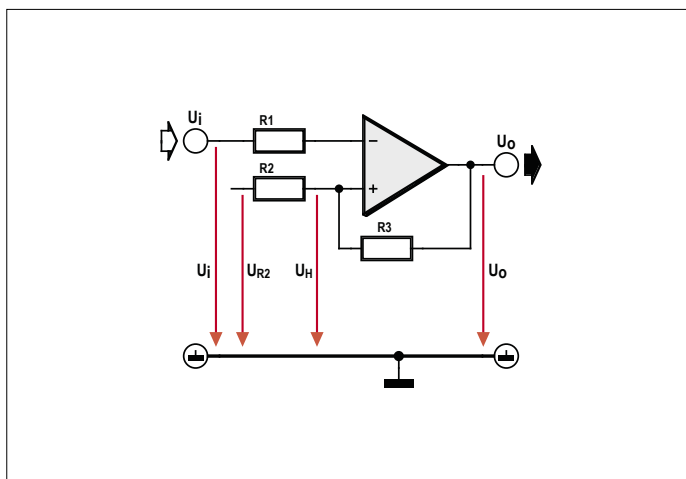


Fig. 10. Trigger de Schmitt inverseur classique.



Fig. 9. Oscillogramme du seuil de commutation inférieur de la modification non-inverseuse.

La résistance $R1$ peut ainsi être choisie avec une erreur minime dans la série E24 : $1,3 k\Omega$.

On obtient pour le **seuil inférieur de commutation** :

$$I_{R5} = \frac{U_{maxop} - U_{FD} - U_{R2}}{R5} = 148 \mu A \rightarrow \text{seuil de commutation supérieur !}$$

$$R5 = \frac{U_{maxop} - U_{FD} - U_{R2}}{I_{R5}} = \frac{6,8V - 0,21V - 1,7V}{148 \mu A} = 33,04 k\Omega$$

Pour $R5$, la valeur $33 k\Omega$ de la série E12 convient parfaitement. La **figure 7** décrit le circuit avec les valeurs des composants. En observant les deux oscillogrammes (**fig. 8** et **fig. 9**), on note une très bonne correspondance avec les valeurs théoriques.

Trigger de Schmitt inverseur

Il est également possible avec le trigger de Schmitt inverseur de calculer les seuils de commutation à partir de la tension de référence. Tout comme pour la version non-inverseuse, une petite modification du circuit classique de la **figure 10** est possible, ce qui aboutit à la version de la **figure 11**. Ici aussi, les deux seuils sont découplés à l'aide de deux résistances et de

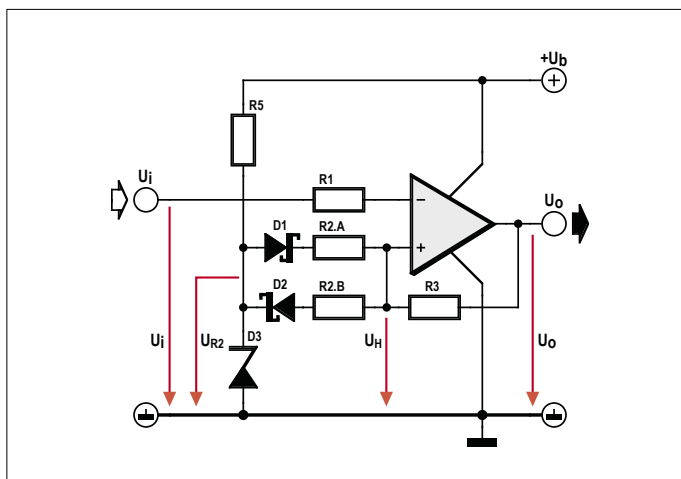


Fig. 11. Principe de la modification d'un trigger de Schmitt inverseur.

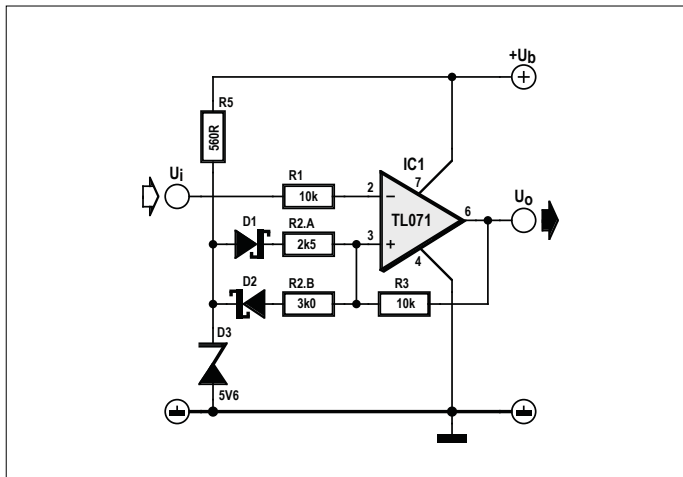


Fig. 12. La modification dimensionnée du trigger de Schmitt inverseur.

deux diodes Schottky. Cette fois ces composants ne se trouvent plus dans la boucle de rétroaction, mais dans la liaison vers la tension de référence. Le fractionnement de R2 en R2.A et R2.B assure que, selon l'état de commutation, la tension de comparaison adéquate est appliquée sur l'entrée non-inverseuse. On choisit d'abord R3, puis on en déduit R2.A et R2.B.

Pour le **seuil de commutation inférieur**, les égalités $U_a = U_{\minop}$ et $U_H = U_{H-}$ s'appliquent, de même que les équations suivantes :

$$\frac{U_{H-} - U_{\minop}}{R3} = \frac{U_{R2} - U_{FD} - U_{H-}}{R2.A}$$

$$R2.A = R3 \times \frac{U_{R2} - U_{FD} - U_{H-}}{U_{H-} - U_{\minop}}$$

On peut utiliser les égalités $U_a = U_{\maxop}$ et $U_H = U_{H+}$ pour le **seuil de commutation supérieur** :

$$\frac{U_{\maxop} - U_{H+}}{R3} = \frac{U_{H+} - U_{R2} - U_{FD}}{R2.B}$$

$$R2.B = R3 \times \frac{U_{H+} - U_{R2} - U_{FD}}{U_{\maxop} - U_{H+}}$$

Le circuit modifié est à présent dimensionné pour un cas concret, et à cette occasion nous vérifions que le fonctionnement est correct. Les paramètres suivants s'appliquent en cas d'utilisation d'un TL071 :

$U_m = U_{R2} = 5,6 \text{ V}$	$U_B = 9 \text{ V}$	$R_3 = 10 \text{ k}\Omega$
$U_{H+} = 6,6 \text{ V}$	$U_{\maxop} = 8,2 \text{ V}$ (mesuré)	$U_{FD} = 0,21 \text{ V}$ (mesuré)
$U_{H-} = 4,6 \text{ V}$	$U_{\minop} = 1,48 \text{ V}$ (mesuré)	

L'équation suivante s'applique au **seuil de commutation inférieur** :

$$R2.A = R3 \times \frac{U_{R2} - U_{FD} - U_{H-}}{U_{H-} - U_{\minop}} =$$

$$\frac{5,6 \text{ V} - 0,21 \text{ V} - 4,6 \text{ V}}{4,6 \text{ V} - 1,48 \text{ V}} \times 10 \text{ k}\Omega = 2,532 \text{ k}\Omega$$

La résistance R2.A pourra facilement se composer de 1,5 kΩ et 1 kΩ.

On a pour le **seuil de commutation supérieur** :

$$R2.B = R3 \times \frac{U_{H+} - U_{R2} - U_{FD}}{U_{\maxop} - U_{H+}} =$$

$$\frac{6,6 \text{ V} - 5,6 \text{ V} - 0,21 \text{ V}}{8,2 \text{ V} - 5,6 \text{ V}} \times 10 \text{ k}\Omega = 3,038 \text{ k}\Omega$$

En utilisant pour R2.B les valeurs 1,8 kΩ et 1,2 kΩ, on obtient une erreur d'environ 1,3%. La **figure 12** montre le circuit avec les valeurs des composants. Les oscillogrammes (**fig. 13** et **fig. 14**) montrent des seuils réels très proches des valeurs calculées.

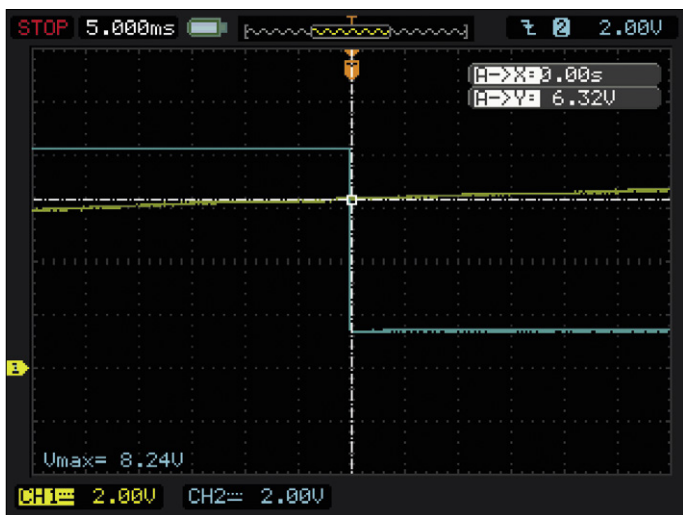


Fig. 13. Oscillogramme du seuil de commutation supérieur de la modification inverseuse. L'évolution de la tension de sortie (ligne bleu-vert) est également inversée.

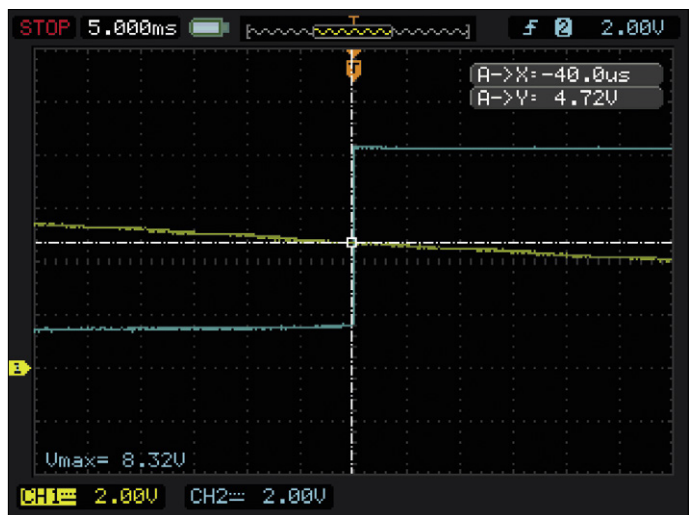


Fig. 14. Oscillogramme du seuil de commutation inférieur de la modification inverseuse.

Parameter-Legende

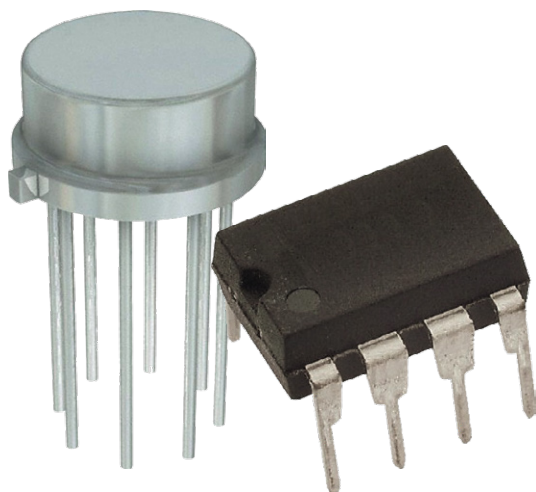
Paramètre	Signification
U_b	Tension d'alimentation
$U_{\max op}$	Tension max. de sortie de l'ampli op
$U_{\min op}$	Tension min. de sortie de l'ampli op
U_H	Seuil de commutation
U_{H+}	Seuil de commutation supérieur
U_{H-}	Seuil de commutation inférieur
U_M	Tension moyenne
ΔU_{H+}	Hystérésis positive
ΔU_{H-}	Hystérésis négative
U_{R2}	Tension de référence
U_{FD}	Chute de tension de la diode Schottky

Les valeurs de $U_{\min op}$, $U_{\max op}$ et U_{FD} sont à déterminer par la mesure, elles jouent un rôle central pour la précision. Les feuilles de spécifications des amplis op et des diodes donnent des valeurs avec des plages de tolérance souvent trop grandes.

Conclusion

Même si le sujet semble un peu aride, vous remarquerez lors de votre prochain calcul d'un trigger de Schmitt que les deux circuits des figures 7 et 12 sont faciles à dimensionner avec les équations détaillées ici. En revanche, l'effort requis ne se justifie qu'en cas de besoin de seuils très précis. Si ce n'est pas le cas, le circuit « normal » de la figure 3 suffira amplement. Vous disposez également des équations. Bon amusement !!

(160340 – version française : Xavier Pfaff)



À propos de l'auteur

Après de nombreuses années en tant qu'ingénieur de projet pour des systèmes télécommandés et des automates programmables, Volker Schmidt travaille en tant qu'indépendant dans la branche informatique. L'électronique est son dada depuis l'âge de 12 ans. Il se consacre aujourd'hui aux microcontrôleurs AVR, mais aussi aux circuits analogiques.



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.

janvier 2019

- ◇ **Cartexpo** (salon INTERNATIONAL de la CARTE POSTALE DE COLLECTION)
11 au 12/01 – Paris
www.cartexpo.fr
- ◇ **Salon de la plongée sous-marine**
11 au 14/01 – Paris
www.salon-de-la-plongee.com/fr/accueil.html
- ◇ **Convention annuelle FFDomotique 2019**
16 au 17/01 – Saint-Ouen
www.ffdomotique.org
- ◇ **Salon du travail et de la mobilité professionnelle**
17 au 18/01 – Paris
www.salondutravail.fr
- ◇ **Cidisc** (convention internationale du disque de collection)
19 au 20/01 – Paris
www.jukeboxmag.com
- ◇ **Semaine du son**
21/01 au 03/02 – Paris et partout en France
www.lasemaineduson.org
- ◇ **Forum international de la cybersécurité**
22 au 23/01 – Lille
www.forum-fic.com
- ◇ **46^e congrès de la mobilité intelligente**
23 au 24/01 – Paris
www.congres-atecitsfrance.fr
- ◇ **Salon de la Radio & Audio Digital**
24 au 26/01 – Paris
www.salondelaradio.com
- ◇ **46^e festival international de la bande dessinée**
24 au 27/01 – Angoulême
www.bdangouleme.com
- ◇ **SEPEM Industries Nord**
29 au 31/01 – Douai
douai.sepem-industries.com
- ◇ **Biogaz Europe**
30 au 31/01 – Nantes
www.biogaz-europe.com



(presque) tout ce que vous avez toujours voulu savoir sur... les boîtiers



photo : Bopla Gehäuse Systeme GmbH

Thijs Beckers

Vous concevez un circuit, vous dessinez un circuit imprimé, et puis vous essayez de trouver un boîtier. C'est bien entendu la marche... à ne pas suivre ! Il est préférable de commencer par choisir le boîtier en fonction du montage, et ensuite de dessiner le circuit imprimé en fonction du boîtier. Ainsi les deux s'emboîteront parfaitement. Nous vous donnons ci-après quelques conseils pour le choix de ce boîtier.

Lorsqu'on manipule du matériel électronique, on a toujours en tête – ou du moins on devrait – que l'électricité et les composants doivent être traités avec un minimum de respect. Les composants sont parfois très sensibles à l'électricité statique, et le courant peut, dans le pire des cas, s'avérer mortel. Un environnement (trop) humide n'est en général pas conseillé, nous le savons tous, et une température trop élevée peut être tout aussi dommageable, surtout si l'électronique est dans un boîtier fermé. Ceci nous donne déjà une petite idée de ce que l'on attend du boîtier.

Q Dans quel environnement le boîtier sera-t-il utilisé ?

R La fonction principale d'un boîtier est de protéger ce qu'il contient, à l'endroit où il est utilisé, contre tout ce qui pourrait entraîner un mauvais fonctionnement de son contenu. Pour les protections contre les solides et l'eau, il y a des normes internationales ; celles-ci sont regroupées sous l'« Indice de Protection » (IP), repris dans les tableaux ci-après. De l'autre côté de l'Atlantique, on préfère parfois les normes de l'association des fabricants d'équipements électriques (*National Electrical Manufacturers Association*, NEMA [1]).

Q De quel matériau le boîtier sera-t-il constitué ?

R En gros, vous avez le choix entre du plastique et du métal. Pour de l'électronique standard, on préfère sou-

vent le plastique qui offre le plus d'avantages, notamment en matière d'isolation électrique et de prix. Pour des applications industrielles, la rigidité et la solidité du métal seront des critères déterminants. Il faut aussi considérer la conduction thermique et la résistance au feu, ainsi que la protection contre les acides et les bases, la corrosion, les rayures ou les variations de température.

Il faut aussi tenir compte de la biodégradabilité des plastiques. L'organisme de normalisation ASTM International (*American Society for Testing and Materials* [2]) établit des recommandations techniques pour ce type de matériaux. La notion de biodégradabilité est cependant assez floue : après 1 000 ans, n'importe quel plastique est dégradé...

Q Quelle sera la forme du boîtier ?

R Les goûts et les couleurs, ça ne se discute pas. Toutefois il ne faut pas négliger l'aspect d'un boîtier. D'un point de vue commercial, ce facteur est même parfois plus important que l'ergonomie. Pour nos projets, nous nous contenterons sans doute d'un boîtier « standard » plutôt que d'une fabrication sur mesure et hors de prix...

Si vous souhaitez un rendu plus professionnel, vous pouvez aussi faire réaliser une face avant, les offres abordables ne manquent pas. Et si vous tenez à tout faire vous-même, vous pouvez toujours vous rendre dans un *fab lab* : atelier de fabrication où vous aurez accès à de l'outillage professionnel ou

industriel et où vous pourrez donner libre cours à votre imagination et à votre sens artistique.

Avant tout adapté à l'application

Prenons quelques exemples pratiques de montages de notre magazine pour illustrer nos propos. Le circuit imprimé du capteur du détecteur de rayonnement amélioré décrit en novembre 2011 [3] devait être dans un boîtier étanche à la lumière et autres rayonnements, mais qui devait pouvoir être ouvert pour y placer un échantillon. Il fallait donc une cage de Faraday, soit un boîtier en métal, qui protège aussi de la lumière. Les tensions et courants en jeu étaient faibles, il n'y avait pas de critère spécifique pour la sécurité. Choix final pour le prototype : une boîte à biscuits !

Dans le cas de la nouvelle horloge Nixie pilotée par GPS, dont le montage a été publié en mai 2016 [4], les tubes devaient



être bien visibles et le récepteur GPS ne pouvait pas être dans un boîtier métallique. Un boîtier en acrylique transparent s'imposait donc.

Pour terminer, prenons un montage plus récent : le thermomètre à bargraphe Nixie [5]. Le look de l'appareil était bien entendu très important, et nous avons opté ici aussi pour un boîtier en acrylique, tout en profitant des propriétés de conduction lumineuse intrinsèques du matériau.

Ces exemples montrent bien que le choix d'un boîtier pour chaque montage repose sur des critères divers et variés. C'est pourquoi nous l'affirmons haut et fort : réfléchissez d'abord au boîtier, puis arrangez-vous pour que votre électronique s'en accommode !

(180460-04 – version française : Jean-Louis Mehren)

Liens

- [1] NEMA : www.nema.org
- [2] ASTM International : www.astm.org
- [3] « compteur alpha/bêta/gamma - détecteur de rayonnement amélioré », Elektor, 11/2011 : www.elektormagazine.fr/110538
- [4] « nouvelle horloge Nixie pilotée par GPS », Elektor, 05/2016 : www.elektormagazine.fr/150189
- [5] « thermomètre à bargraphe Nixie », Elektor, 07-08/2018 : www.elektormagazine.fr/160705

Normes IP pour les boîtiers

Premier chiffre

IP	Protection contre	Signification
0x	Pas de protection	
1x	Gros objets (> 50 mm)	Protection contre les contacts accidentels et superficiels avec la main. Protection contre les corps solides de taille > 50 mm
2x	Objets de taille moyenne (> 12,5 mm)	Protection contre un contact avec un doigt. Contact sûr uniquement pour les appareils de mesure. Protection contre les corps solides de taille > 12,5 mm
3x	Petits objets (> 2,5 mm)	Protection contre un contact avec un outil. Contact sûr uniquement pour les appareils de mesure. Protection contre les corps solides de taille > 2,5 mm
4x	Très petits objets (> 1 mm)	Protection contre un contact avec un outil ou un fil. Protection contre les corps solides de taille > 1 mm
5x	Poussière	Protection totale contre les contacts, le boîtier étant complètement fermé. Pas de protection absolue contre la poussière, mais suffisante pour éviter les défauts de fonctionnement.
6x	Étanche	Protection totale contre les contacts et la poussière, le boîtier étant étanche.

Deuxième chiffre

IP	Classe	Protection
x0	Pas de protection	–
x1	Gouttes, Type I	Gouttes verticales
x2	Gouttes, Type II	Gouttes à 15° de la verticale
x3	Éclaboussures	Pluie (10 l/min) jusqu'à 60° de la verticale
x4	Projections	Projections (10 l/min) sous n'importe quel angle
x5	Jets d'eau	Jets d'eau (12,5 l/min) sous n'importe quel angle
x6	Forts jets d'eau	Forts jets d'eau (100 l/min) sous n'importe quel angle
x7	Immersion	Immersion temporaire (30 min à 1 m)
x8	Étanche	Submersible dans des conditions spécifiées
x9	Étanche	Taux d'humidité supérieur à 90% ou nettoyage à haute pression



embrouillamini d'ampli

ça marche, mais pas comme on le pense !

Thomas Scherer (Allemagne)

Un circuit publié dans un projet de lecteur est passé au travers du contrôle de qualité des équipes d'Elektor. C'est évidemment regrettable, mais ce circuit est quand même intéressant : il fonctionne, mais pas du tout comme l'avait imaginé son auteur. Cette histoire est donc passionnante et fort instructive.

Voici ce dont il s'agit : dans le numéro de septembre/octobre 2018 est paru un « amplificateur BF pour sinus modulé en fréquence » [1] avec le sous-titre « un plaidoyer pour l'analogique ». Voilà un titre susceptible d'avoir un effet hypnotique sur tout électronicien passionné par l'audio ou la HF. Cela donne envie de dévorer l'article pour découvrir comment ça marche.

Historique

Bien sûr, pour nombre de lecteurs du magazine Elektor de l'ère post-analogique, tout ce qui ne parle pas de bits et d'octets semble tiré d'un grimoire ésotérique. Mais Elektor est aussi lu par de véritables experts de l'analogique et il ne fallut pas longtemps après la publication pour que nous recevions des courriels et des commentaires truffés de critiques. Et hélas, il faut bien le reconnaître, ces critiques étaient justifiées. Que s'est-il passé ? Notre lecteur

Hans-Norbert Gerbig nous a envoyé pour publication un circuit de son cru. Par un malheureux concours de circonstances, l'analyse de ce montage a été confiée, à titre exceptionnel, à un de nos collaborateurs moyennement pointus en électronique analogique et technique HF. Habituellement cela ne pose pas de problème, car si quelque chose paraît étrange, l'examineur n'hésite pas à consulter ses collègues plus expérimentés. Nous pouvons alors faire part de nos doutes à l'auteur et les lever. Malheureusement ce ne fut pas le cas ici.

Notre collaborateur a effectivement noté des choses bizarres, mais des problèmes de délais l'ont empêché de partager ses doutes. La loi de Murphy est implacable : tout ce qui peut aller de travers ira de travers, et il n'y a pas eu de seconde vérification de l'article. Le circuit et sa description ont donc été publiés avec leurs erreurs. Cela ne devrait pas arriver – et pourtant cela arrive de temps

à autre. Alors, il ne nous reste plus qu'à vous présenter nos excuses : nous sommes sincèrement désolés !

Mais à quelque chose malheur est bon : les considérations sur le circuit, les principes sous-jacents et les suppositions fausses sont particulièrement instructifs. C'est pourquoi nous allons en discuter tous les aspects significatifs et donner une fin heureuse à cette histoire !

Hypothèses de base

L'amplificateur « BF pour sinus » devait être la réponse analogique aux amplificateurs audio à technique numérique et leurs signaux à modulation de largeur d'impulsion (MLI). Dans son principe, il est constitué d'un générateur sinusoïdal HF, d'un circuit intégré amplificateur analogique du commerce et d'un filtre passe-bas pour la démodulation. Dans l'article [1], on trouve le schéma de principe de la **figure 1**. Et quelqu'un avec de bonnes connaissances en audio et en hautes fréquences va d'ores et déjà avoir des doutes : ça ne peut pas fonctionner ! Et pourquoi ? Il n'y a pas qu'une seule réponse à cette question.

1. Un filtre passe-bas ne peut pas démoduler un signal HF présentant une forme de courbe symétrique. Si la fréquence du signal se situe largement au-delà de la fréquence de coupure du filtre, les variations rapides du signal sont purement et simplement éliminées. Le filtre remplit son office : il élimine la HF et laisse passer la BF. C'est indépendant de la forme de la courbe. Peu importe qu'il s'agisse de signaux sinusoïdaux, triangulaires, carrés ou de forme quelconque. Le type de la modulation n'a (presque) rien à voir.

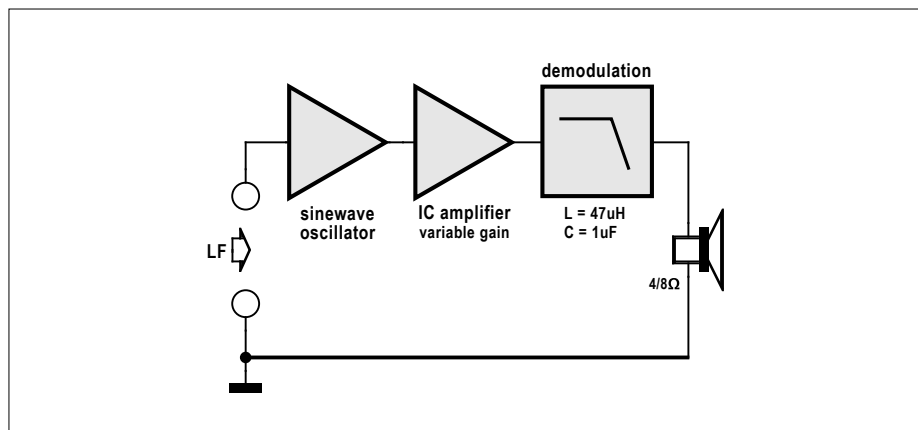


Figure 1. Le schéma fonctionnel précise le principe qui ne fonctionne malheureusement pas.

aucun effet, peu importe que ce soit une modulation de fréquence ou d'amplitude. Le signal modulé sur la HF n'apparaît en sortie d'un filtre passe-bas que s'il modifie la largeur d'impulsion du signal HF, c'est-à-dire si la modulation a un effet sur l'intégrale de la courbe de l'onde. C'est précisément la fonction du filtre passe-bas à la sortie d'un amplificateur numérique MLI : il intègre.

2. La bande passante d'un circuit intégré amplificateur est très limitée. Par exemple le LM386 : selon sa feuille de caractéristiques, ce composant n'amplifie que jusqu'à 300 kHz. À 1 MHz, son gain n'est plus que de 10. Un amplificateur audio est donc déjà lui-même un filtre passe-bas. Au-delà de 1 MHz, le gain diminue de 30 dB/décade. À 10 MHz, il vaut donc 0,3 environ, et on n'a plus affaire à un amplificateur, mais à un atténuateur. CQFD : un amplificateur audio ne peut pas amplifier de la HF. C'est également valable pour un TBA820 et d'autres puces audio. C'est pourquoi ce circuit ne peut pas fonctionner, même si un filtre passe-bas était capable de démoduler un signal HF modulé en fréquence. Ce dont il est incapable, comme montré en 1.

Il s'ensuit que l'hypothèse de base du circuit est doublement fautive. Un amplificateur « BF pour sinus » ne peut donc pas fonctionner, peu importe comment on tourne ou considère la chose.

Et pourtant, le circuit amplifie !

Si l'on considère le circuit complet de la **figure 2**, qu'on le réalise et que l'on connecte sur son entrée une source audio et à sa sortie un haut-parleur et qu'on l'alimente sous 5 V environ, on est surpris d'entendre des sons. Alors, ça marche quand même ! ?

En effet, l'amplificateur amplifie. Un bon technicien des circuits le constate assez vite. Les idées sont une chose, mais seules les mesures sont vraiment fiables, alors j'ai réalisé le circuit et « prouvé » ce qui suit par la mesure. J'y reviendrai plus tard. À gauche sur le schéma, on a un oscillateur à quartz construit autour du transistor. Un signal BF sur son collecteur est censé moduler le signal qu'il produit. La soi-disant HF modulée est alors prélevée sur la résistance d'émetteur et appliquée au travers d'un potentiomètre de réglage de 10 kΩ sur l'entrée de l'amplificateur BF, dont le gain en continu est réglable par un autre potentiomètre de 10 kΩ (sur la broche 8) dans la plage 20 à 200. On a ensuite, à droite,

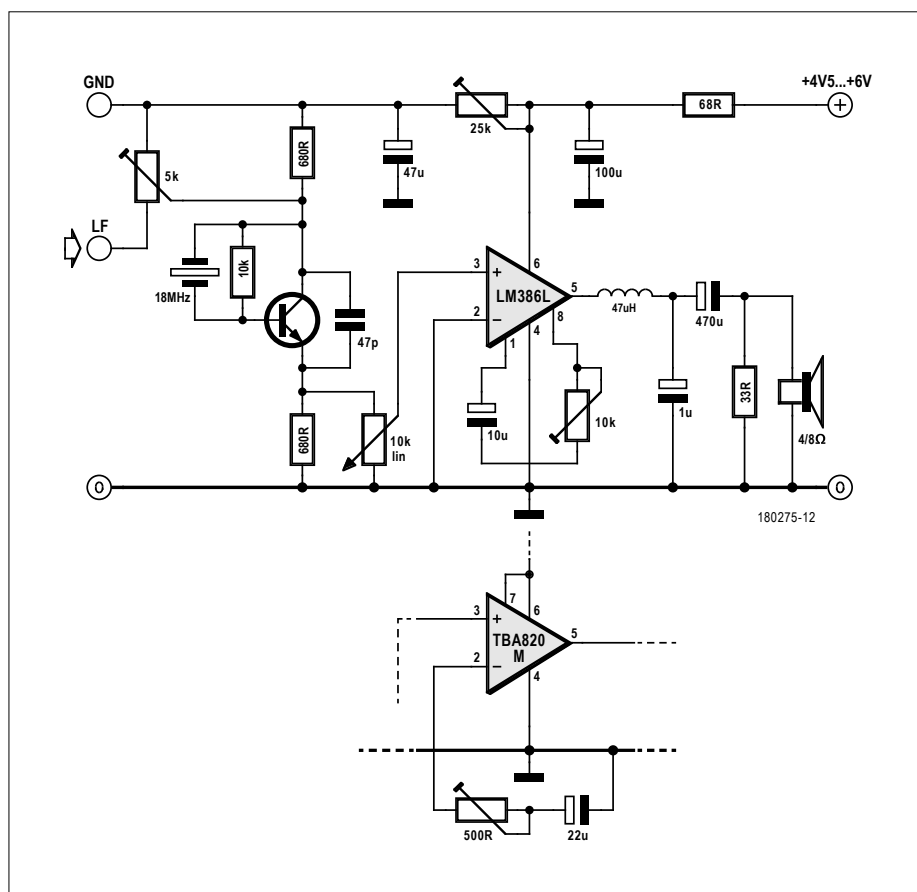


Figure 2. Le circuit concret de l'amplificateur « BF pour sinus ».

un filtre passe-bas constitué d'une inductance de 47 μH et d'un condensateur de 1 μF. Le condensateur de 470 μF est le condensateur de sortie normal, qui sert à bloquer la composante continue du signal de sortie. Avec le potentiomètre de 25 kΩ en haut, on règle le point de fonctionnement du générateur sinusoïdal. Voilà la description du circuit tel qu'il a été imaginé.

Alors, pourquoi ce circuit amplifie-t-il quand même ? En supposant que l'oscillateur fonctionne et produit effectivement de la modulation en fréquence, il y a quand même une composante BF non négligeable qui parvient à l'entrée de l'amplificateur. Pour la BF, la résistance active composée du transistor et de sa résistance de base est égale au quotient de la valeur de la résistance (10 kΩ) par le gain en courant du transistor (typiquement 100 pour un BF494), soit 100 Ω environ. Avec les valeurs indiquées, le signal appliqué au collecteur apparaît donc sur le potentiomètre de 10 kΩ « amplifié » d'un facteur 0,8. Du côté chaud on a donc au mieux un mélange additif de HF et de BF, qui arrive tel quel

à l'entrée de l'amplificateur audio. Il n'y a pas de filtre passe-haut qui éliminerait la basse fréquence. Et cet amplificateur est, comme déjà dit, un filtre passe-bas qui n'amplifie que les signaux BF. Le filtre passe-bas, en sortie, n'a donc pratiquement plus rien à faire. On pourrait aussi bien le supprimer.

Un long discours pour en dire peu : dans ce circuit, l'amplificateur amplifie parce qu'il est un ampli audio quasi normal entouré d'un peu de décoration sous la forme de composants inutiles. On pourrait supprimer tout ce qui se trouve à gauche du potentiomètre de 10 kΩ (sur la broche 3), ainsi que le potentiomètre de 25 kΩ, la résistance de 68 Ω et le filtre passe-bas et l'on obtiendrait un amplificateur qui n'en marcherait que mieux. Le générateur sinusoïdal ne sert strictement à rien.

Mais comme l'amplificateur amplifie malgré les composants HF, l'auteur a cru que son idée était correcte.

Et en réalité ?

Pour que personne ne puisse dire : « oui, mais tout ça n'est que de la théorie », j'ai

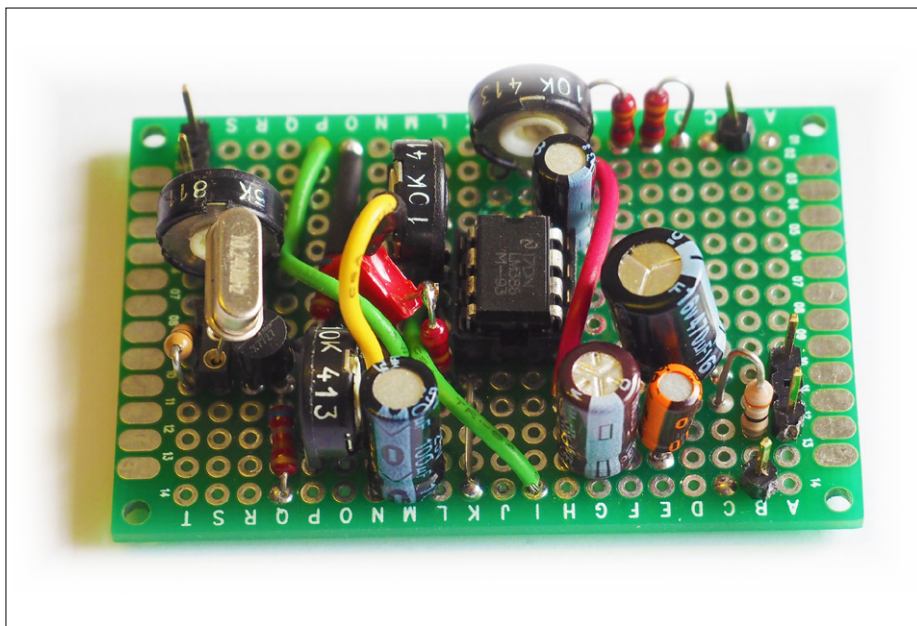


Figure 3. Mon prototype « vite fait » du circuit de la figure 2.

rendu l'évidence encore plus évidente en réalisant le circuit de la figure 2 sur une carte à trous. Pas très joli comme on peut le voir sur la **figure 3**, mais opérationnel. J'ai remplacé le quartz de 18 MHz par un autre de 10,24 MHz tiré d'un tiroir de mon bazar. J'avais même un BF494. Il n'y a que l'ampli que j'ai dû commander. Une première mesure sur le collecteur du

transistor montra la HF annoncée avec un niveau de 55 mV efficaces et la fréquence attendue. L'oscillateur fonctionne donc en effet. L'amplitude HF est réglable dans certaines limites avec le potentiomètre de 25 k Ω .

La **figure 4** montre la HF après connexion d'une source audio de 600 Hz et d'amplitude appropriée. Comme affirmé par

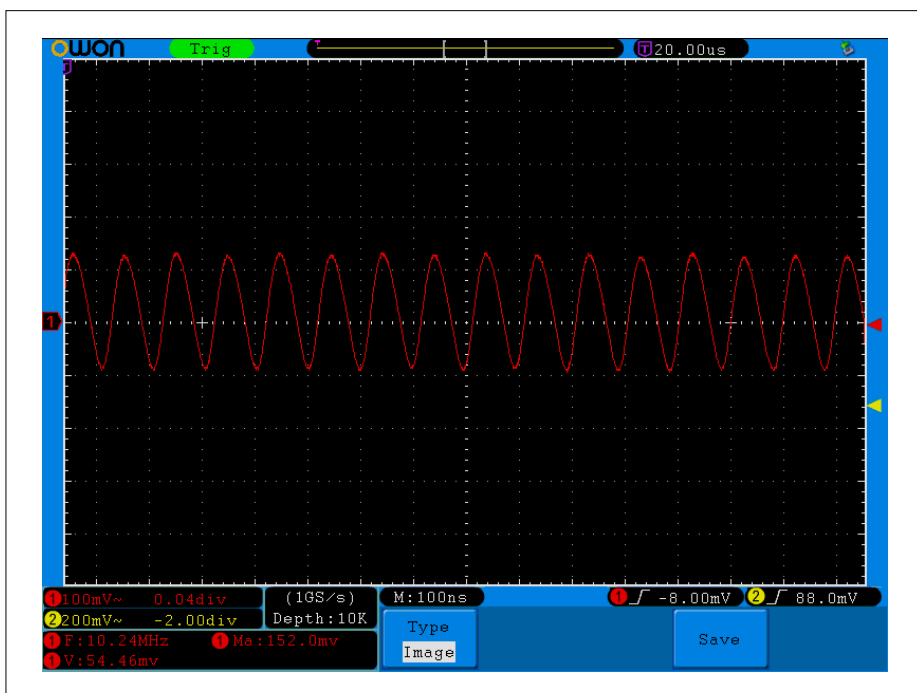


Figure 4. Le signal de 10,24 MHz « modulé » en 600 Hz. Aucune trace d'une quelconque modulation.

l'auteur, on n'y voit aucune modulation d'amplitude, mais pas non plus de modulation de fréquence, du moins à l'œil nu. Et de même avec une résolution horizontale et donc temporelle différente.

Si l'on fait varier la base de temps dans le sens de la BF, un signal apparaît (**fig. 5**) avec la composante de 10,24 MHz superposée à la sinusoïde de 600 Hz appliquée à l'entrée. Il ne pouvait guère en être autrement.

À la sortie, le signal attendu de 600 Hz « amplifié » arrive sur le haut-parleur. Les parasites visibles sur la **figure 6** sont dus à mon montage volant et peu soigneux et n'ont rien à voir avec le principe du circuit.

Comme on peut le constater, dans ce cas, la pratique rejoint la théorie.

Suite

Et ce n'est pas tout. Le circuit de la figure 2 contient encore quelques détails qui montrent ce qu'il vaut mieux ne pas faire, car on n'a certainement pas là un exemple de meilleure pratique de conception de circuit. Ça commence avec un détail qui a une grosse influence : le filtre passe-bas constitué de la résistance de 68 Ω et du condensateur de 100 μ F, en haut à droite, est bien joli, mais faire passer aussi l'alimentation de l'ampli par la résistance n'a pas de sens. Au contraire, l'ampli se met à écrêter dès que la puissance du signal est relativement faible. Si l'on veut en tirer plus que quelques dizaines de milliwatts, il vaut mieux connecter la broche 6 directement à l'alimentation.

Le fait que la masse de la source du signal ne soit pas connectée à la masse de l'ampli rend le circuit pratiquement inutilisable. Pour cette raison, mon montage a commencé par refuser de fonctionner, car la masse de mon générateur de fonctions n'était pas libre de potentiel, pas plus que la sortie de mon alimentation de laboratoire connectée au circuit. Il s'en est suivi que mon générateur a mis le côté supérieur de la résistance de collecteur à la masse. En conséquence, il n'y avait ni son, ni HF mesurable sur la résistance d'émetteur. Deux condensateurs de 1 μ F en série avec les bornes du générateur de fonctions ont remédié au problème, au prix toutefois des parasites visibles sur la figure 6, induits par la boucle de masse ainsi créée. *Nota bene* : il ne faut jamais modifier la masse de référence des signaux sans nécessité. Mais toutes les bonnes choses vont par

trois : il n'y avait toujours pas de son, même avec les deux condensateurs. En manipulant le potentiomètre de 25 k Ω en haut de la figure 2, j'obtenais parfois quelque chose comme un faible son. Et pourquoi ? Pour osciller et propager le signal HF, le transistor a besoin d'un courant minimal, lequel crée une chute de tension dans la résistance d'émetteur. Quelques centaines de millivolts et, en fonction de la position du potentiomètre de 10 k Ω , la sortie de l'amplificateur rejoint le positif de la tension d'alimentation et il n'y a plus d'amplification. Le problème provient du manque de découplage entre cette tension et la broche 3. Il aurait fallu mettre un condensateur entre l'émetteur du transistor et l'extrémité chaude du potentiomètre de 10 k Ω qui va à la broche 3. Après avoir ajouté ce condensateur, j'ai pu, au moyen du potentiomètre de 25 k Ω , optimiser le courant de fonctionnement du transistor et obtenir une amplitude HF assez grande, sans bloquer l'ampli par une polarisation intempestive.

Pour finir, au cours d'un agréable entretien téléphonique avec Monsieur Gerbig, j'ai pu établir qu'il ne s'agissait pas d'un « oscillateur piloté » [2]. Même si c'était le cas, on ne pourrait pas utiliser un signal BF – un mélange spectral de fréquences – pour piloter la fréquence d'oscillation afin d'obtenir une modulation. On peut toutefois capter l'oscillateur en FM ou en ondes courtes, il y a donc une certaine modulation en fréquence et même en amplitude, mais cela reste très faible et est sans doute un artefact dû à la non-linéarité du transistor.

Conclusion

En électronique analogique, il y a plein d'occasions de se tromper. L'expérience ainsi que l'existence de règles empiriques maintes fois vérifiées par des collègues sont très utiles pour ne pas se fourvoyer dans des situations de non-fonctionnement. Ce qui est très facile. L'amplificateur « BF pour sinus » n'en est pas un exemple exceptionnel. Sincèrement : qui ne s'est jamais planté au cours de la conception d'un circuit ? Moi-même, je peux en chanter non seulement des chansons, mais aussi des arias entiers. Mais on apprend de ses erreurs. Et nous ne répéterons pas celles décrites ici. ◀

(180589-04 – version française : Helmut Müller)

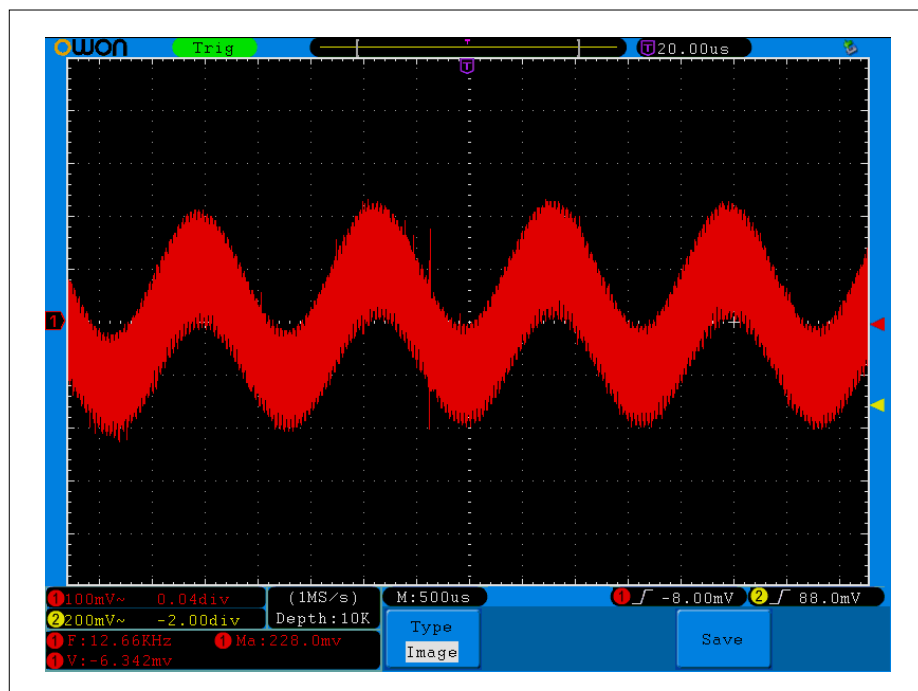


Figure 5. L'amplificateur audio reçoit le mélange d'un signal de 600 Hz superposé à une composante à 10,24 MHz, qu'il élimine.

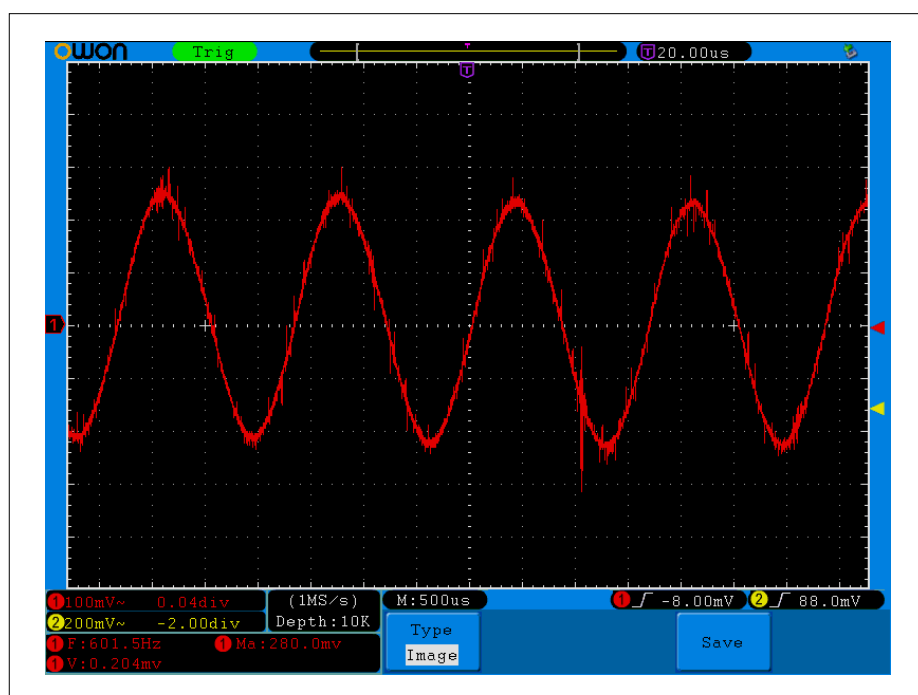


Figure 6. Le signal de 600 Hz légèrement amplifié à la sortie du circuit. Il n'y a déjà plus de HF à l'entrée du filtre passe-bas.

Liens

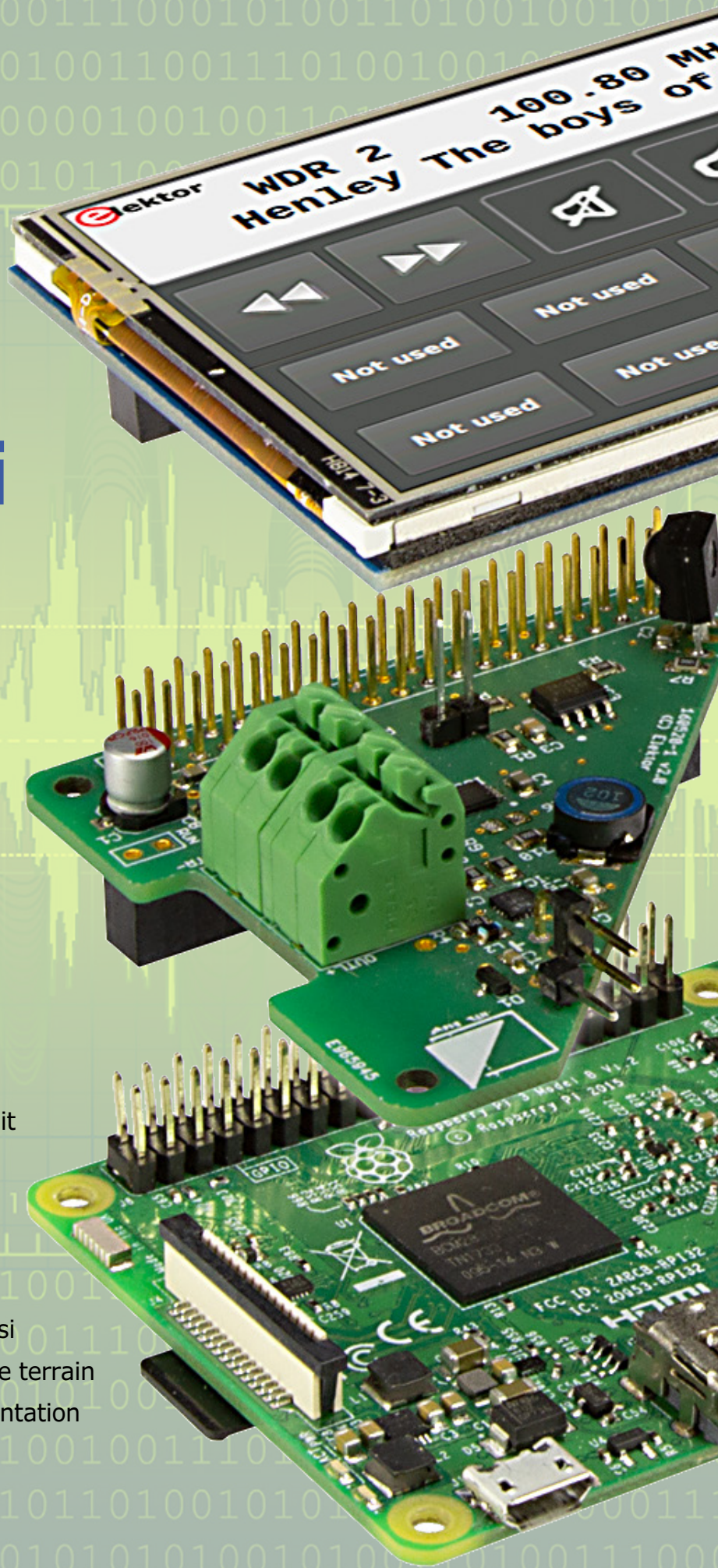
- [1] « amplificateur BF pour sinus modulé en fréquence », Elektor 09-10/2018 : www.elektormagazine.fr/180275-04
- [2] 'Funktechnik ohne Ballast', Otto Limann, Franzis-Verlag, 12^e édition, pp. 148/149

radio FM avec RDS

HAT pour Raspberry Pi

Franz Parzer, Fabian Bugelmüller,
Chritoph Fornezz (HTL Steyr, Autriche)
avec Ton Giesberts et Mathias Claußen
du labo d'Elektor

Construire une radio FM avec un nano-ordinateur Raspberry Pi et un tricorne fait maison monté en impériale (HAT), voilà un montage intéressant. Toutefois ce projet offre bien plus. Le logiciel (pilotes et interface graphique écrits spécialement pour cette application) ainsi que le matériel constituent un formidable terrain d'expérimentation assorti d'une documentation fouillée.





INFOS SUR LE PROJET



Raspberry Pi 2 ou 3 (B+)

HAT récepteur FM

écran tactile



débutant

→ connaisseur

expert



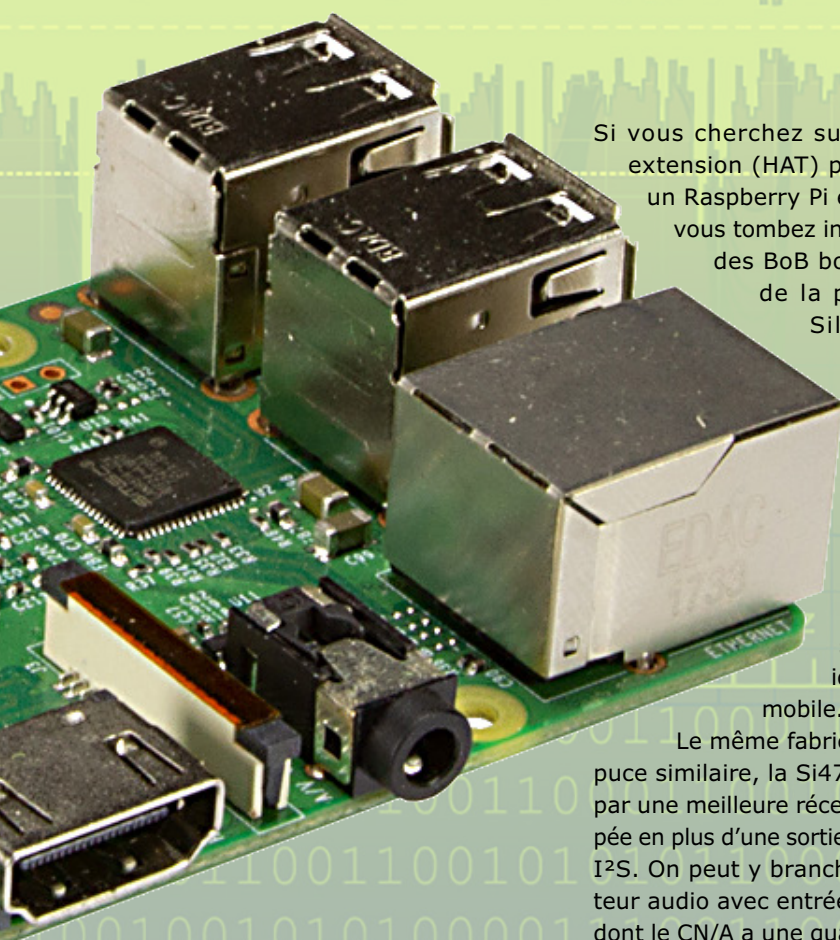
env. 4 h



poste à souder pour CMS,
four à refusion conseillé



env. 65 €



Si vous cherchez sur l'internet une extension (HAT) pour transformer un Raspberry Pi en récepteur FM, vous tombez inévitablement sur des BoB bon marché dotés de la puce Si4703 de Silicon Laboratories. Ce circuit intégré constitue une radio VHF complète avec RDS (depuis l'entrée d'antenne jusqu'aux sorties analogiques stéréo), il est idéal pour l'usage mobile.

Le même fabricant propose une puce similaire, la Si4731, caractérisée par une meilleure réception, mais équipée en plus d'une sortie numérique audio I²S. On peut y brancher un amplificateur audio avec entrée numérique I²S, dont le CN/A a une qualité supérieure à celle du Si4703. En prime, notre radio

disposera d'un capteur IR qui permet de la télécommander avec une zap-pette universelle. Comme interface graphique, nous utilisons un écran tactile LCD de 3,5 pouces.

Ce matériel est épaulé par du logiciel sous Linux, qui, outre le pilote pour la puce utilisée, contient une appli radio spéciale pour faciliter la commande du récepteur.

Notre chapeau radio pour RPi fonctionne avec toutes les versions « normales » du RPi à partir du type 2 ; oubliez le Raspberry Pi 1. Nous avons aussi testé l'extension sur le modèle Zéro, elle fonctionne, mais sans garantie...

Récepteur étendu

Le schéma de la **figure 1** montre les quelques composants nécessaires à la radio FM.

Si4731

La puce du récepteur radio est le *Broadcast AM/FM Radio Receiver Si4731-D60* [1] logé dans un minuscule boîtier QFN à 20 broches de Silicon Laboratories avec

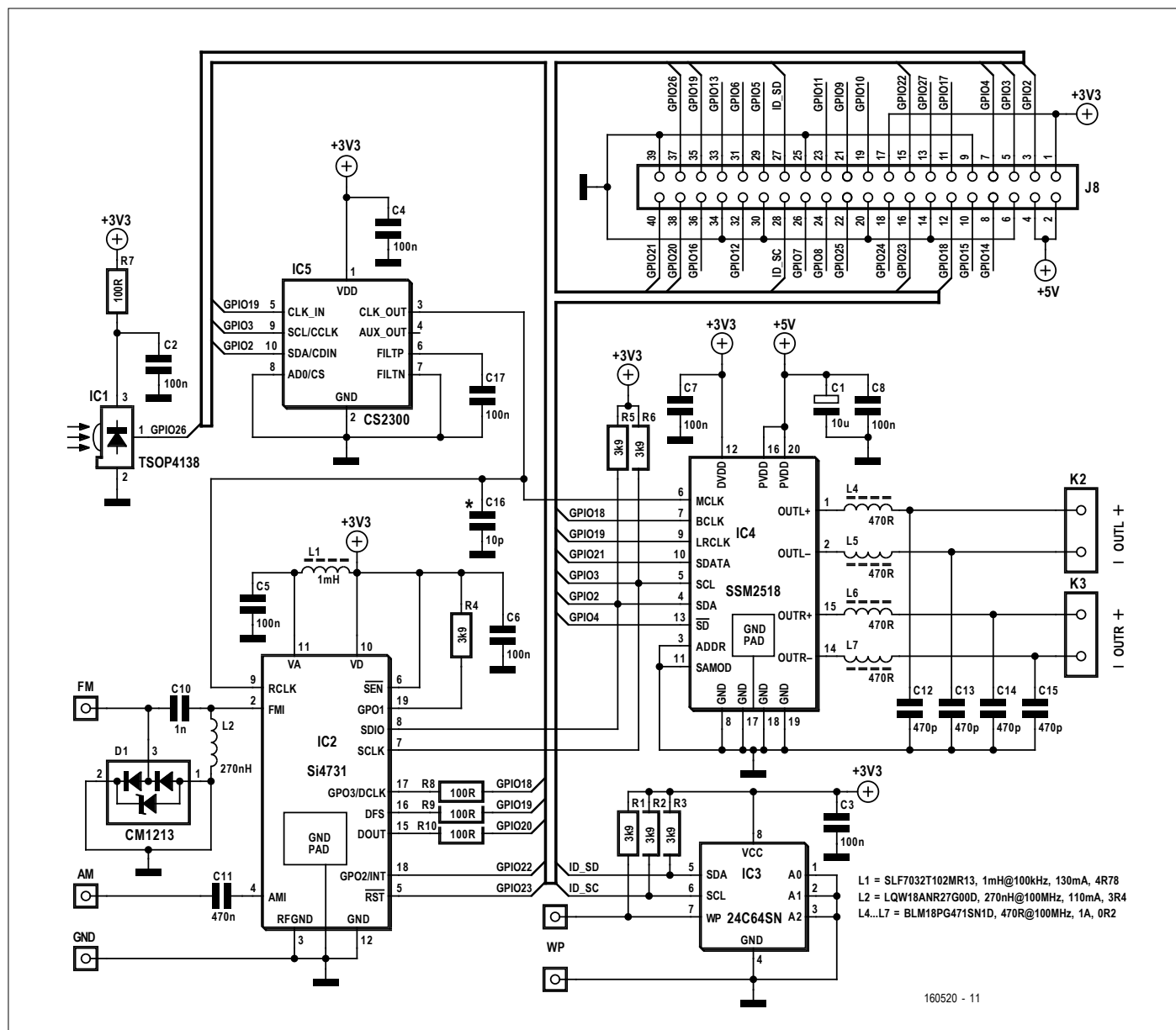


Figure 1. Schéma de la radio FM.

la version D60 du micrologiciel. En plus de la gamme FM (64 à 108 MHz), elle reçoit la gamme PO de 520 à 1710 kHz en AM (**fig. 2**). L'interface numérique audio du Si4731 travaille en mode esclave et accepte de nombreux formats audio (MSB en tête), y compris les modes I²S justifiés à gauche. La puce exploite le RDS et assume le format I²S échantillonné à 32, 44,1 et 48 kHz. C'est le RPi qui configure le Si4731 par l'interface I²C. Pour la réception de l'AM et de la FM, la puce dispose de deux entrées distinctes. En FM, L2 simule la résonance. La diode de suppression SPD1 s'occupe de la protection contre les DES tandis que C10 filtre la composante continue du signal. On capte généralement mieux la FM avec un fil de 75 cm qu'avec une antenne

télescopique, mais en AM, nous n'avons pas encore essayé ; ce qui convient le mieux, c'est l'antenne ferrite ou un cadre de 180 à 450 μ H avec en parallèle un condensateur de 470 nF (C11) sur l'entrée AM du Si4731. Le nombre de spires dépend forcément des caractéristiques du bâton de ferrite. Il suffit de regarder sur l'internet pour trouver comment fabriquer une antenne convenable. Les condensateurs C5 et C6 servent de découplage, L1 stabilise la tension d'alimentation, R8 à R10 sont des résistances de protection, comme l'indique la fiche technique.

SSM2518

Pour la reproduction, nous avons choisi le SSM2518 [2] d'Analog Devices, un

amplificateur en classe D avec interface audio série numérique, piloté par bus I²C et qui accepte les données au format I²S. Chaque sortie stéréo fournit 2 W sur 4 Ω . Le câblage du circuit s'accorde aux spécifications de la fiche technique. L'alimentation de 3,3 V, DVDD pour le numérique, est découplée par C7. Sur le 5 V de PVDD, on trouve C1 et C8 sur l'alimentation de puissance de la partie analogique.

Le condensateur C1 de découplage du 5 V est au polymère, il a une RSE extrêmement basse de 30 m Ω . Selon la fiche technique, on ne doit insérer des filtres aux sorties vers les haut-parleurs que si les lignes font plus de 20 cm. En pratique, nous avons constaté que même en respectant cette condition, les conduc-

teurs rayonnent malgré tout tellement de hautes fréquences qu'il est impossible de recevoir le RDS. La solution a consisté à filtrer les sorties avec des perles de ferrite (L4 à L7) et condensateurs (C12 à C15). La polarisation haute du bus I²C réclame R5 et R6.

Récepteur IR

Le récepteur IR (IC1, un TSOP4138) capte les signaux à 38 kHz d'une télécommande de type RC5 (mais aussi les signaux à 36 kHz d'une télécommande de téléviseur ou de chaîne hi-fi). Il les amplifie, les filtre et les démodule, puis envoie les informations sur la ligne GPIO26 du Raspberry Pi, qui les interprète et transmet les ordres pour commander la radio.

EEPROM

Pour que le matériel réponde aux exigences (cf. ci-dessous) du couvre-chef du RPi, il faut une EEPROM qui contient les informations nécessaires. C'est IC3, un banal circuit 24C64 qui communique lui aussi par bus I²C.

CS2300

Lors d'un essai préliminaire, nous avons utilisé un oscillateur comme générateur d'horloge. Nous avons remarqué qu'il ne fournissait pas un signal d'horloge identique à celui du bus I²S, ce qui faisait perdre des bits. Comme l'horloge du I²S est exactement 1/256^e de celle du système, nous avons recruté le multiplicateur de fréquence CS2300 pour multiplier par 256 le rythme sur le bus I²S. Les horloges système et I²S sont ainsi synchrones et sans gigue.

Le CS2300-CP-CZZZ de Cirrus Logic est configuré et commandé par I²C. Le circuit reproduit le schéma de la fiche technique. Le condensateur C16 de 10 pF sur CLK_{Out} pour affaiblir les harmoniques n'est pas indispensable, il n'est pas installé.

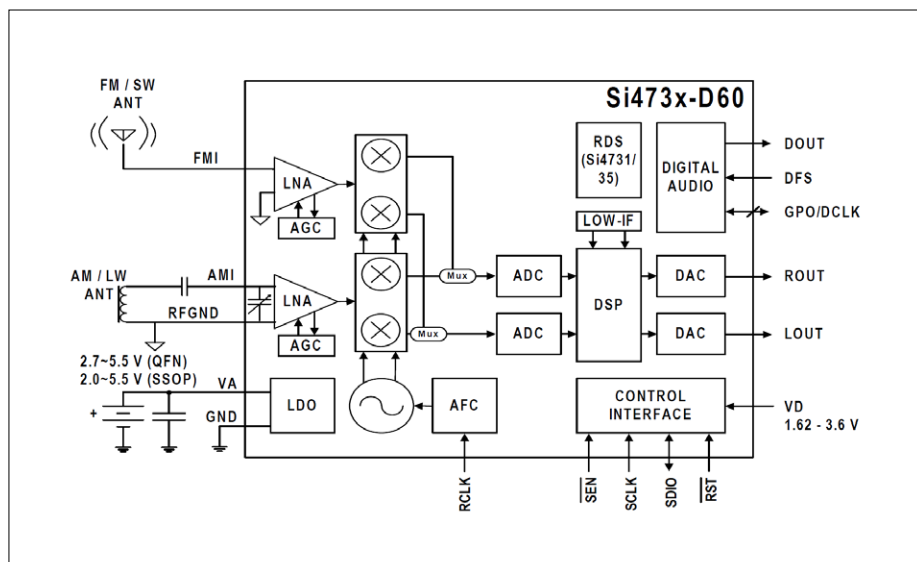


Figure 2. Circuit interne du Si4731 (Silicon Laboratories).

Alimentation

Peu de choses à dire à ce sujet : utilisez pour alimenter la radio un bloc secteur de qualité capable de fournir 3 A (pas seulement sur le papier !), c'est qu'il n'est pas seulement en charge de la radio et du RPi, de l'écran et du récepteur, mais aussi de l'ampli audio de 2x2 W. Et que dire de la qualité d'un bloc secteur qui ne coûte que 3 € ?

Du logiciel pour le matériel

Pour pouvoir utiliser le matériel sous Linux, il faut que des pilotes particuliers soient disponibles pour chaque composant du matériel dans le noyau (*kernel*) de Linux et qu'ils soient montés. Il y a déjà dans le noyau un pilote pour l'ampli en classe D, mais pour le syntoniseur de la radio, il faut modifier celui qui est présent pour l'adapter à la famille Si476x. La plupart des modifications ont dû être réalisées du fait qu'il s'agit de deux familles assez différentes. Pour le

multiplicateur de fréquence, on a aussi dû reprendre le pilote d'un autre composant et le modifier.

Le développement d'un pilote est pénible et demande de l'expérience en programmation, mais il est aussi fastidieux d'incorporer le pilote dans le noyau et de le compiler sans erreur. Pourtant, pas de panique ! Pour le matériel de la radio FM, nous avons préparé une image terminée et fin prête, il n'y a plus qu'à la copier sur la carte SD du Raspberry Pi. Cela fonctionne aussi avec toute distribution telle que Raspbian ou Kodi. L'application *piRadio* réalisée sur mesure est aussi disponible dans l'image.

Mais si vous voulez vous pencher sur le code du logiciel ou créer votre propre distribution (éventuellement avec une autre interface utilisateur), vous trouverez dans les nombreuses documentations réunies sur la page Elektor du projet [4] des conseils, des pilotes, des explications et encore bien plus.

Liens

- [1] Spécifications des HAT : <https://github.com/raspberrypi/hats>
- [2] Fiche technique du Si4731 : www.silabs.com/documents/public/data-sheets/Si4730-31-34-35-D60.pdf
- [3] Fiche technique du SSM2518 : www.analog.com/media/en/technical-documentation/data-sheets/ssm2518.pdf
- [4] Page du projet : www.elektormagazine.fr/180119-04
- [5] Radio FM au labo d'Elektor : www.elektormagazine.fr/labs/fm-radio-receiver-with-rds-for-raspberry-pi
- [6] PiRadio au labo d'Elektor : www.elektormagazine.fr/labs/piradio-for-fm-radio-receiver-with-rds-for-raspberry-pi-160520-1
- [7] Radio FM sous Volumio : <https://github.com/rpi-Receiver/>
- [8] Radio FM sur Github : <https://github.com/ElektorLabs/160520-FM-Radio-Receiver-with-RDS-for-Raspberry-Pi>

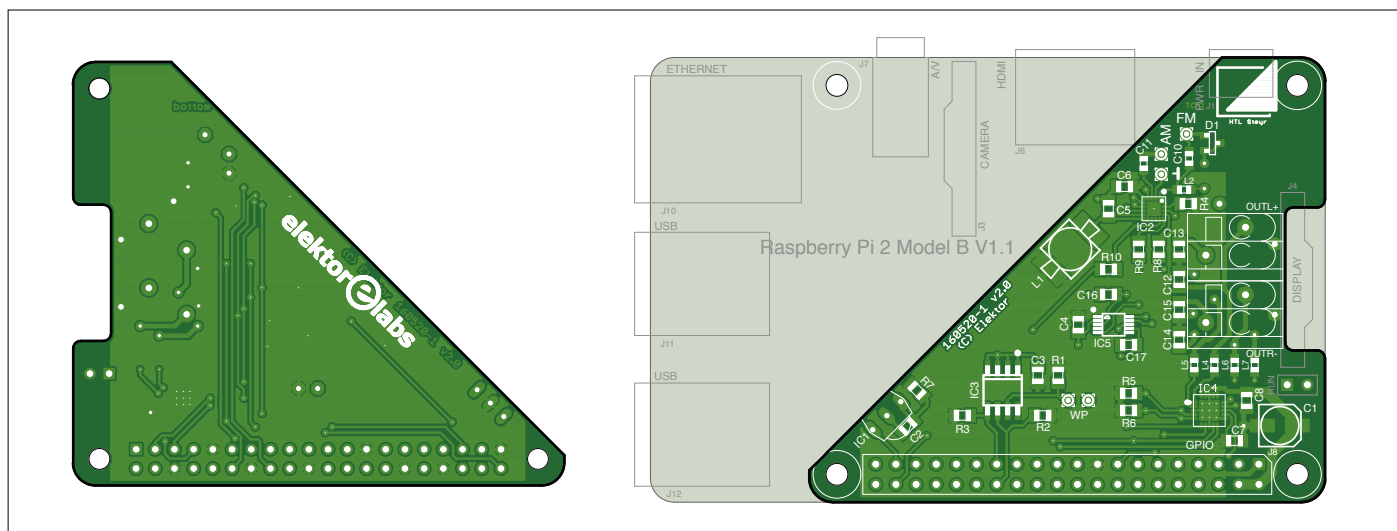


Figure 3. Tracé des pistes du chapeau de RPi pour la radio FM.

Mon chapeau pour RPi, c'est un tricorn

Un chapeau (HAT) est une carte d'extension pour Raspberry Pi qui correspond aux spécifications du HAT [1]. Il se branche sur les 40 broches de GPIO du RPi. Deux de ces broches sont réservées exclusivement à la connexion d'une « EEPROM ID » dans laquelle sont enregistrées les informations sur la carte, les fonctions dont elle dispose, quelles GPIO doivent être configurées et comment, et si le RPi et le HAT sont alimentés par le port micro-USB du RPi ou bien par le HAT. En substance, un HAT doit satisfaire les exigences suivantes :

- conditions de base imposées aux cartes d'extension
- données valables dans l'EEPROM (informations sur le fabricant,

carte des GPIO et arborescence de l'appareil)

- barrette à 40 contacts
- avec une alimentation par le HAT (*backpowering*), il doit pouvoir délivrer au moins 1,3 A au RPi.

Il n'est écrit nulle part que le circuit imprimé du chapeau (**fig. 3**) doit être rectangulaire ! Notre matériel est installé sur une carte triangulaire, avec une découpe pour la connexion de l'écran tactile. Non seulement c'est original, mais cela permet d'économiser sur les coûts de production. Ce qui compte, c'est que les groupes de composants puissent rester séparés les uns des autres et soient disposés dans un ordre logique. Comme le récepteur IR et l'EEPROM n'ont rien à voir avec la fonction principale, ils sont nettement séparés des autres groupes.

Le récepteur radio se trouve en haut de la carte, le multiplicateur de fréquence est au milieu et l'ampli en classe D est en bas, à droite. Près du récepteur, on trouve l'alimentation, la FM, l'AM et la périphérie des sorties numériques, disposées séparément (autant que possible sur une carte si petite). Pour l'ampli, on place le filtre de manière à ce que les perles de ferrite soient tout près de la puce et les condensateurs, à proximité immédiate des bornes de raccordement des haut-parleurs. À côté de l'EEPROM, on trouve des broches pour le cavalier de la protection en écriture. Il faut l'insérer pour pouvoir écrire dedans.

On fixe au RPi la carte du récepteur au moyen d'entretoises M2,5 (mâles/femelles) de 17 mm de long (**fig. 4**). L'écran prend place encore au-dessus avec quatre entretoises de 17 mm (également M2,5 mâles/femelles) à côté de la nécessaire prise Ethernet et un écrou M2,5 vient compenser le peu d'épaisseur de la carte. Fixez le chapeau radio sur le RPi et vissez quatre entretoises M2,5 de 14 mm de long (aussi mâles/femelles) dans les entretoises inférieures. Les quatre entretoises sur lesquelles repose l'écran tactile se trouvent ainsi à la même hauteur ; le filetage intérieur est dirigé vers le haut. Ceci réduit la contrainte sur le connecteur à 26 contacts de l'afficheur. Si vous n'installez pas l'écran LCD et que vous le remplacez par un écran (tactile) externe, inutile de monter des entretoises de 14 mm et l'écrou, fixez le chapeau du RPi avec six vis M2,5.

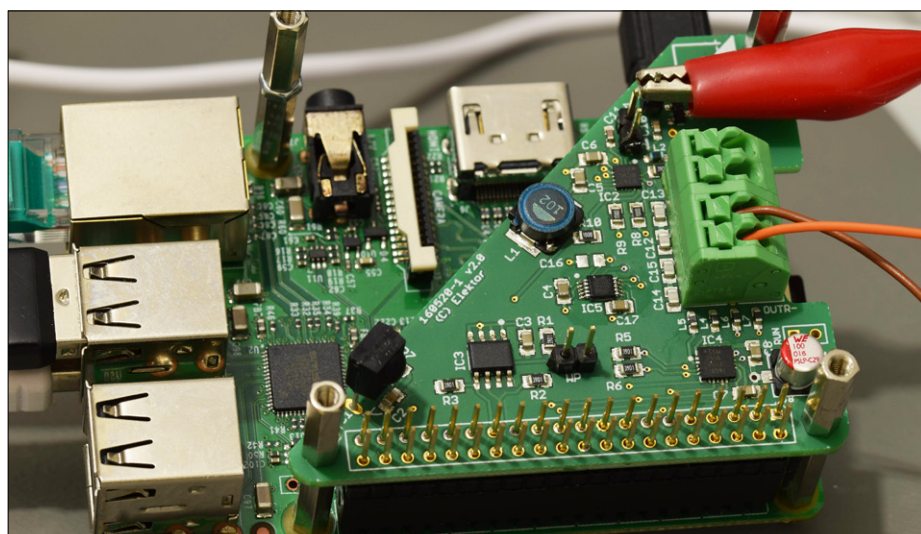


Figure 4. Montage du chapeau sur le Raspberry Pi.

Du logiciel pour la radio

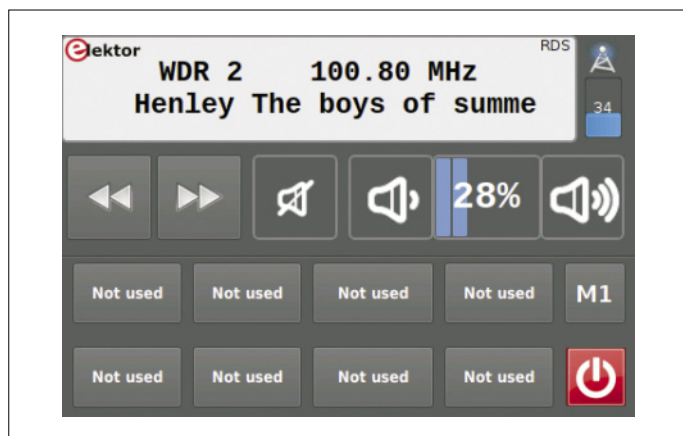


Figure 5. L'afficheur principal de la piRadio.



Figure 6. L'émetteur est en mémoire.

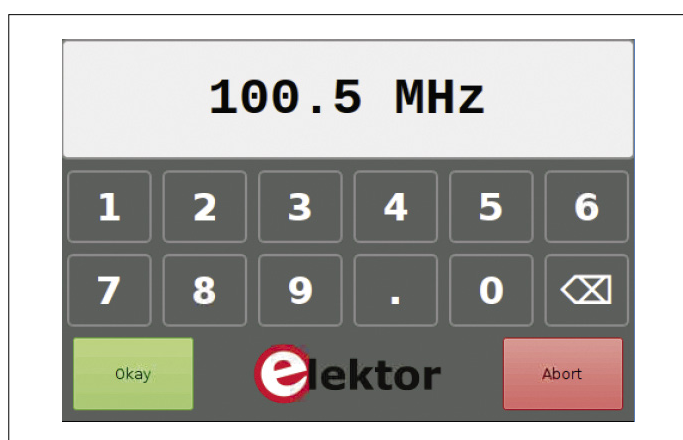


Figure 7. Introduction directe de la fréquence.

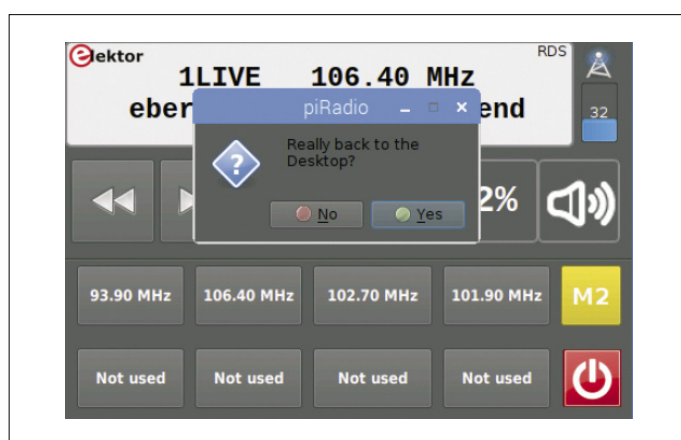


Figure 8. Quitter la radio pour retourner vers Raspbian ?

Si vous cherchez des applis radio, vous rencontrerez vite KRadio, Gnomeradio ou RDS-Surveyor. Il existe également des applis TV qui permettent aussi d'écouter la radio. Chaque appli a ses avantages et ses inconvénients, mais pour profiter de la télécommande IR et de l'écran tactile, il n'y en a pas de bonne, elles sont conçues pour un ordinateur de bureau avec souris et clavier. Un petit écran de 3,5 pouces à faible résolution est d'une utilisation malaisée.

C'est pourquoi Mathias Claußen du labo d'Elektor a développé un programme baptisé piRadio pour piloter le récepteur FM. Le logiciel a été spécialement conçu pour ce matériel-ci avec un écran tactile de 3,5 pouces et 480×320 pixels, comme le modèle populaire Waveshare35a. La manipulation en est totalement intuitive. La **figure 5** montre l'écran de la piRadio. Quand en haut, tout à droite, l'abréviation RDS est affichée, à sa gauche, on peut lire le nom de l'émetteur et sa fréquence. Sur la deuxième ligne apparaît

le texte actuellement transmis par RDS. Si le récepteur ne peut décoder que le nom de la station, l'écran montre une ligne avec une police plus claire pour une meilleure lisibilité. Et s'il n'y a aucune réception d'information RDS, seule la fréquence actuelle est affichée. La force du champ reçu est inscrite dans un petit bargraphe, tout à droite sous le symbole d'antenne. La réception est d'autant meilleure que la valeur est élevée. Dans la zone du bas, il y a huit boutons d'abord étiquetés « Not used ». Ils correspondront à des émetteurs mémorisés. Il y en a quatre pages que l'on feuillette avec le bouton *M[x]*, ce qui donne en tout la possibilité de choix entre 32 stations. Pour mémoriser une fréquence, on va d'abord sur la page à utiliser, on choisit alors la fréquence à l'aide des boutons avec flèches, puis on pousse pendant 5 s sur le bouton sélectionné de l'écran tactile. Sur l'écran, une étoile repère apparaît dans le coin supérieur droit sous le RDS (**fig. 6**) quand l'émetteur est enregistré.

Comme il est assez ennuyeux de changer la fréquence avec les flèches, nous avons implémenté une interface pour une introduction directe. Appuyer sur l'écran renvoie à un masque pour entrer la fréquence (sur la télécommande, utiliser le bouton Menu ; sur le clavier, la touche *[E]*). La fréquence actuelle s'affiche et vous pouvez facilement introduire une nouvelle fréquence, l'ancienne s'efface dès la première entrée. Pour 98,55 MHz, saisissez simplement *[9][8][.][5][5]*. En cas de faute de frappe, poussez juste sur le bouton de retour en arrière. Si la fréquence entrée est dans une plage valide, le bouton *Okay* apparaît. Si vous n'êtes pas satisfait de votre choix, poussez sur *Abort* et retournez à l'interface utilisateur principale. Le réglage à l'aide d'un clavier suit le même principe. Sur une télécommande, *a priori* le bouton à gauche du '0' sur le pavé numérique correspondra au retour en arrière, le bouton à droite du '0' au point décimal et bien sûr OK vaut *Okay*.

Il faudra peut-être tâtonner pour trouver les bons boutons. Après la sauvegarde, vous pouvez sélectionner les stations mémorisées avec la télécommande ou le clavier, en saisissant [xy], x étant le numéro de page et y celui de la station. Par exemple, pour choisir la deuxième station de la première page, tapez 12 et pour la quatrième station de la deuxième page, tapez 24. Pour arriver sur une station de la première page, le choix est direct, on peut ne donner que le numéro de l'émetteur et attendre une seconde. Et pour un autre émetteur sur la même page, vous pouvez saisir [0y]. Si vous choi-

sissez un numéro de station qui n'est pas disponible ou s'il n'y en a aucun d'enregistré, la radio ne change pas de fréquence. Entre la fenêtre de texte et les touches de sélection de station, on trouve à gauche les flèches pour changer de fréquence. Il faut rester dessus pendant trois secondes avant que la recherche vers le haut ou le bas commence. Sur une télécommande, on se sert des boutons d'avance rapide et de rembobinage ; sur un clavier, on a les flèches vers la gauche et vers la droite. Pour augmenter la vitesse de recherche ou la diminuer, on pousse sur les boutons à flèche vers le haut ou vers le bas. Pour changer le volume d'écoute, on se sert des touches correspondantes ou de celle de silencieux pour un tout ou rien. Pareil pour une télécommande, mais sur un clavier, on utilise les touches *plus*, *moins* et *M*.

Reste le bouton d'allu-

mage et d'extinction. Une brève action dessus arrête la radio, mais une pression prolongée à 10 s fait apparaître un petit symbole de bureau dans le coin supérieur droit de l'écran. Quand vous lâchez la touche, une fenêtre de message surgit pour demander si vous voulez vraiment retourner vers le bureau. Cliquez sur *Yes* pour fermer l'application et aller au bureau (**fig. 8**). Sinon, le message s'efface et l'application se poursuit. Le **tableau 1** rassemble les commandes et saisies que l'on peut effectuer sur l'écran tactile, ainsi que sur un clavier ou une télécommande.

Informations complémentaires

Nous mettons donc à votre disposition une image, basée sur Raspbian, dans laquelle sont déjà intégrés le pilote du matériel radio et de l'écran ainsi que le paquet LIRC pour la télécommande par RC5. L'API Qt sur laquelle repose l'interface utilisateur est installée et les données de démarrage automatique adaptées. L'image est prête à l'emploi, il suffit de la copier sur la carte SD du Raspberry Pi pour pouvoir utiliser directement toutes les fonctions de la radio FM. Cependant, lors de l'élaboration du projet, de nombreux documents dérivés, des textes et des codes de programme ont également été créés, nous les mettons aussi à disposition sur la page du projet [4] et les deux pages associées au labo d'Elektor [5] [6].

- À l'origine, cette radio FM constituait

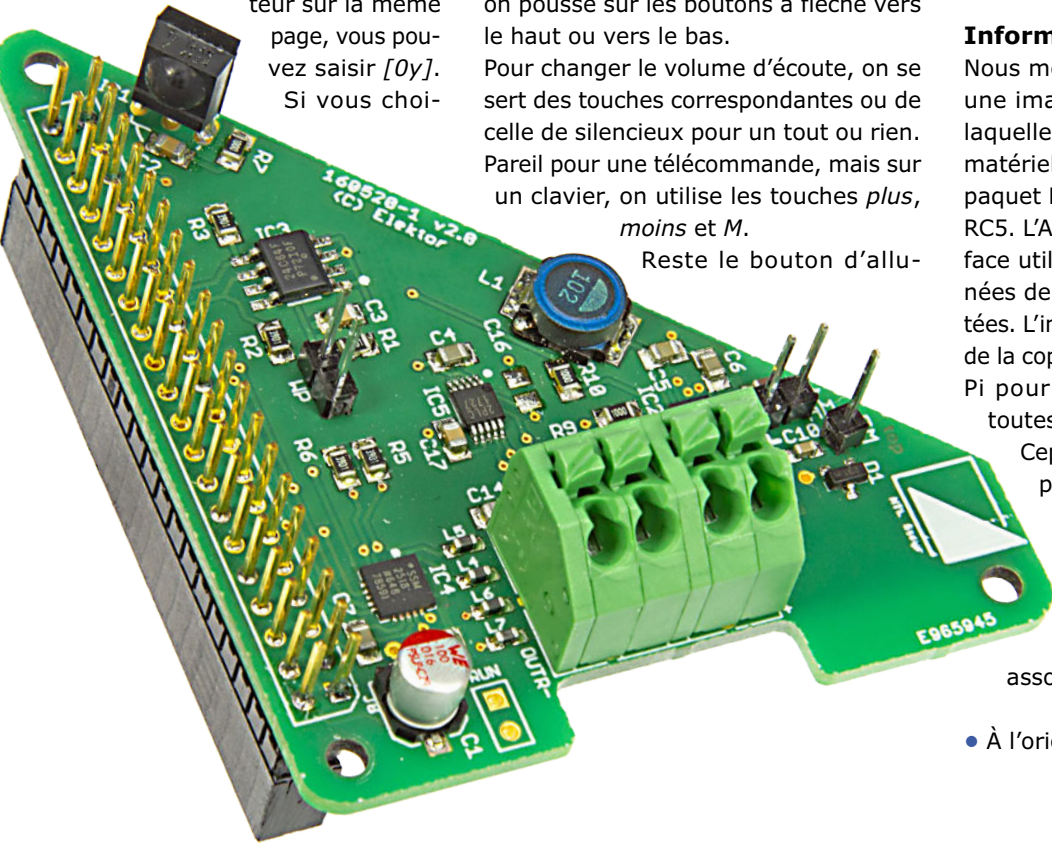


Tableau 1. Opérations sur l'appli piRadio par écran tactile, clavier et télécommande.		
Fonction	clavier	télécommande
Monter le son	+	Vol+
Baisser le son	-	Vol-
Commuter le silencieux	M	MUTE
Éteindre	Q	POWER
Monter en fréquence	flèche vers le haut	CH+
Descendre en fréquence	flèche vers le bas	CH-
Accélérer la recherche en fréquence	flèche à droite	Vidéotexte rouge
Ralentir la recherche en fréquence	flèche à gauche	Vidéotexte bleu
Afficher/masquer le réglage de fréquence	E	MENU
Effacer une fréquence	effacer (BACKSPACE)	bouton à gauche du '0' sur le pavé numérique ou ENTRÉE
Point décimal	Point ou virgule	bouton à droite du '0' sur le pavé numérique ou A-B
Régler la fréquence donnée	ENTER	OK
Annuler et sortir	ESCAPE ou E	MENU



LISTE DES COMPOSANTS

Résistances

R1 à R6 = 3,9 k Ω , 0,125 W, 1 %, CMS 0805
R7 à R10 = 100 Ω , 0,125 W, 1 %, CMS 0805

Condensateurs

C1 = 10 μ F, 16 V, 20 %, 5 mm radial CMS, 0,03 Ω /2,2 A
C2 à C8, C17 = 100 nF, 50 V, 10 %, X7R, CMS 0805
C10 = 1 nF, 50 V, 10 %, X7R, CMS 0603
C11 = 470 nF, 16 V, 10 %, X7R, CMS 0603
C12 à C15 = 470 pF, 50 V, 5 %, C0G/NP0, CMS 0805
C16 = 10 pF, 50 V, $\pm$ 0,5 pF, C0G/NP0, CMS 0805, non installé

Inductances

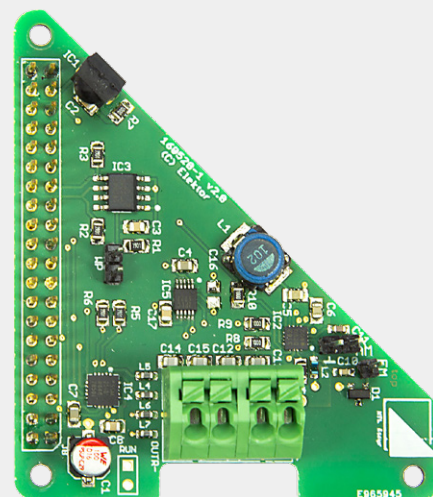
L1 = SLF7032T102MR13-2PF, 1 mH@100 kHz, 130 mA, 4,78 Ω (TDK)
L2 = LQW18ANR27G00D, 270 nH@100 MHz, 110 mA, 3,4 Ω (Murata)
L4 à L7 = BLM18PG471SN1D, 470 Ω @100 MHz, 1 A, 0,2 Ω (Murata)

Semi-conducteurs

D1 = CM1213A-01SO, CMS SOT-23 (ON Semiconductor)
IC1 = TSOP4138 à 3 fils (Vishay)
IC2 = SI4731-D60-GM, CMS à 20 broches, QFN (Silicon Labs)
IC3 = CAT24C64WI-GT3, SOIC-8 (ON Semiconductor)
IC4 = SSM2518CPZ, LFCSP à 20 broches WQ (Analog Devices)
IC5 = CS2300-CP-CZZ, MSOP-10 (Cirrus Logic)

Divers

K2, K3 = bornier encartable à 2 vis, au pas de 3,5 mm
J8 = embase à 2x20 picots GPIO pour RPi (extra long)
4x entretoise de 17 mm M2,5 (mâle-femelle), p.ex. Ettinger 05.12.173
4x entretoise de 14 mm M2,5 (mâle-femelle), p.ex. Ettinger 05.12.143
5x écrou M2,5
embase à 1x5 picots, verticale,



au pas de 2,54 mm, répartie sur les pastilles AM/FM/PW

J8 = écran tactile LCD de 3,5», p.ex. 3.5inch RPi LCD (A) de Waveshare Raspberry Pi à partir du modèle 2
Circuit imprimé, réf. 160520-1

Ne pas installer C9, L3, K1

le projet de fin d'études des auteurs, Fabian Bugelmüller et Christoph Fornezzi, à la Grande école technique fédérale de Steyr (Autriche). Leur document (*RpiReceiver.pdf*) décrit toute la conception du matériel et des pilotes, le contenu de l'EEPROM, l'intégration sous Raspbian (et bien d'autres) dans tous les détails.

- Il y a dans le téléchargement du projet un habillage (*overlay*) pour Raspbian (*rpi-receiver-linux-rpi-4.9.y*), constitué par le micrologiciel spécial nécessaire à la radio FM et qui doit s'intégrer au noyau de Raspbian. La manière de s'y prendre est expliquée en détail dans les instructions d'installation.
- Pour faire fonctionner convenablement la radio, il faut modifier Raspbian en installant le paquet LIRC pour la télécommande IR et en redirigeant l'interface utilisateur HDMI vers l'écran tactile connecté aux GPIO. Tout cela est aussi expliqué dans les instructions d'installation.
- Le labo d'Elektor est à l'origine du logiciel de l'application piRadio. Le code source est disponible pour permettre d'effectuer aisément des modifications : organisation de l'écran, ajout de fonctions, nombre

de stations mémorisées, utilisation d'une télécommande RC5.

- L'appli a été développée avec le Toolkit Qt libre en C++. Vous trouverez aussi sur la page internet du labo d'Elektor [6] d'autres développements logiciels, y compris une API WebSocket.
- Si vous ne souhaitez apporter aucune modification à la radio FM avec piRadio et, si vous voulez la mettre en marche directement, vous trouverez une image prête à l'emploi pour la carte SD du Raspberry Pi.
- On peut aussi utiliser la radio avec une autre appli comme RDS-Surveyor. Ce programme en Java se trouve également dans le téléchargement avec les instructions d'installation. RDS-Surveyor révèle vraiment **tous** les détails des émissions

du *Radio Data System*.

- Une vidéo montre le maniement de la radio FM et de l'interface.
- Il y a un tutoriel sur la manière d'intégrer le matériel sous Volumio [7].
- Il y a aussi un tutoriel (en anglais) pour savoir comment intégrer Qt à Raspbian [8].

Que vous vouliez seulement construire une radio FM, ou bien modifier légèrement le matériel et le logiciel, ou encore appliquer à vos montages les connaissances apportées par ce projet, vous trouverez sûrement votre bonheur dans les pages internet liées à ce montage ! ◀

(180119-04

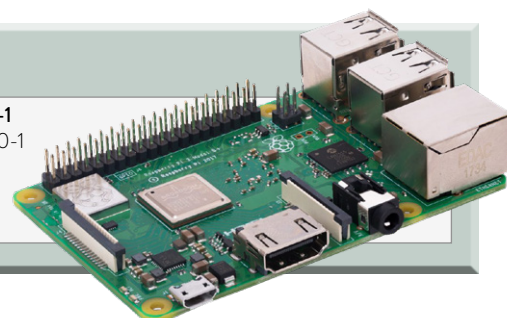
version française : Robert Grignard)



@ WWW.ELEKTOR.FR

→ Circuit imprimé nu, réf. 160520-1
www.elektor.fr/rpi-receiver-160520-1

→ Raspberry Pi 3 B+
www.elektor.fr/rpi3b+



se lancer dans Verilog

les fondamentaux du langage de description de matériel

Jörg Zollmann (Allemagne)

À côté de VHDL, Verilog est le langage de description de matériel (*Hardware Description Language*, HDL) pour CPLD et FPGA le plus connu et le plus répandu. Voici une vue d'ensemble des éléments les plus importants de ce langage.

Pour mémoire : un langage de description de matériel sert à spécifier un circuit par une description textuelle ; on peut alors vérifier son fonctionnement au moyen d'une simulation. Le langage comprend également la transposition, dite *synthèse*, de cette description en un circuit réel au moyen d'outils logiciels.

Verilog et VHDL disposent tous deux d'une grosse communauté d'utilisateurs, tant dans l'industrie que chez les amateurs. Pour quelqu'un travaillant sur des projets CPLD ou FPGA, il peut donc être intéressant de maîtriser les fondamentaux des deux langages. Les rudiments de Verilog présentés dans cet article devraient suffire à résoudre de nombreux problèmes et à comprendre des projets plus complexes comme le « Do it Yourself Processor » abordé dans un prochain numéro d'Elektor.

Conventions

La syntaxe de Verilog ressemble beaucoup à celle du langage C. La courbe d'apprentissage n'est habituellement pas aussi pentue que pour le VHDL ; on obtient plus rapidement ses premiers succès, avec sensiblement moins de lignes de code. Contrairement à VHDL, Verilog est sensible à la casse. Les noms des fichiers sources de Verilog se terminent habituellement en *.v (*.vh, *.sv, *.svh). Comme en C, les commentaires de fin de ligne commencent par une double barre // ou sont délimités par /* et */ pour les commentaires multilignes. Les espaces sont ignorés. Les identificateurs (noms de variables) comprennent les caractères alphanumériques, le trait de soulignement et le dollar (a-z A-Z 0-9 _ \$). Ce dernier ne doit toutefois pas apparaître en tête du nom. Comme équivalent des accolades, Verilog utilise les mots clés `begin` et `end`. Comme pour le développement de logiciel, des styles de codage différents sont apparus en Verilog, où les indentations et le nombre de déclarations `begin/end` diffèrent. Il n'y a pas que pour les débutants qu'il peut s'avérer utile d'encadrer chaque bloc séquentiel et chaque branchement conditionnel d'un `begin/end`.

Représentation des nombres et types de données

Dans Verilog, la représentation des nombres suit la règle `<taille>'<signe><base de numération>nombre`. `<taille>` donne la taille du champ en nombre de bits et non en nombre de chiffres. Les valeurs entre crochets sont facultatives ; par défaut, les nombres sont interprétés comme décimaux, non

Tableau 1. La logique à quatre valeurs de Verilog.

Valeur logique	Interprétation
0	0 logique ou faux (<i>false</i>)
1	1 logique ou vrai (<i>true</i>)
x	inconnu (indifférent)
z	haute impédance

signés. Par exemple, le nombre 42 peut être écrit des manières suivantes :

```
6'b101010 // binaire (bit de poids fort en premier)
6'o52      // octal
6'h2a      // hexadécimal
6'd42      // décimal
6'b10_1010 // le trait de soulignement '_'
            // est utilisé
            // pour une meilleure lisibilité
'd42       // taille 32 bits par défaut
```

Verilog ne connaît que deux classes de types de données. L'une est la classe *net*, dont le membre le plus important est le fil (*wire*). Comme son nom l'indique, il est utilisé pour la modélisation des connexions. L'autre est la classe 'registre' (*register*), dont le représentant le plus important est `reg`. L'entier (*integer*), le réel (*real*), le temps (*time*) et le temps réel (*realtime*) sont aussi de type 'registre', mais pour l'instant d'importance moindre pour la synthèse. Pour la classe 'registre', il s'agit de mise en mémoire, c'est-à-dire que si une valeur doit être sauvegardée provisoirement, ce sera obligatoirement dans un `reg`. Mais cela ne signifie pas que l'usage d'un `reg` entraîne nécessairement la synthèse d'une cellule de mémoire (bascule) comme on le verra plus tard. Les types de données de Verilog suivent une logique à quatre états (**tableau 1**). Pour les deux classes de types de données, il existe des opérations arithmétiques et logiques. Le **tableau 2** donne les opérateurs les plus importants, leurs fonctions et un court exemple.

En l'absence de déclaration de type, Verilog utilise le type *wire* (1 bit). Lorsqu'on oublie de déclarer le type d'un signal, l'outil de synthèse ou de simulation ne déclare pas une erreur, mais

un avertissement qu'une variable inconnue est implicitement considérée de type `wire`.

Verilog ne nécessite pas d'inclusion de paquets ou d'appel de fonctions de conversion de type, ce qui rend le code plus simple et plus compact, mais cela présente un inconvénient : le compilateur Verilog tolère quelquefois des choses impossibles sous VHDL. Par exemple, l'affectation de vecteurs de largeurs différentes ne donne lieu le plus souvent qu'à un avertissement (troncature). Cela a pour conséquence que le circuit peut n'être pas synthétisable ou réalise une fonction différente de celle prévue. Par exemple, déclarons un vecteur de type `reg[7:0]`, soit un registre d'un octet et affectons-lui différentes valeurs :

```
reg [7:0] bus1 = 8'b1011; /* bus1 = 00001011
    les bits de poids fort sont mis à 0 */
reg [7:0] bus2 = 3'b1101; /* bus2 = 00000101
    le 4e bit est tronqué */
```

Lors de l'interprétation de la valeur, l'indication de taille est prioritaire. Quand la valeur est affectée au `reg[7:0]`, les bits de poids fort sont mis à 0.

Module

En Verilog, l'unité de base du code est le module. Un module Verilog commence par le mot-clé `module` et se termine par

`endmodule`. Le mot-clé `module` est suivi du nom du module, puis viennent une liste optionnelle de paramètres puis une liste de ports. Cette liste comporte les noms de toutes les entrées-sorties. Dans les plus vieilles versions de Verilog n'apparaissent dans cette liste que les noms des signaux ; depuis 2001, il est possible d'y ajouter les déclarations de ports, donc le sens des signaux (entrée/sortie) et les déclarations des types des signaux. La liste des ports est suivie de la description fonctionnelle proprement dite du module comprenant les initialisations et les instructions.

```
module a (input wire a);
// module vide
endmodule
```

Verilog encourage l'organisation hiérarchique des circuits. Un module peut instancier d'autres sous-modules. Si ceux-ci doivent avoir des comportements différents, on peut l'indiquer par un paramètre (*parameter*), passé au module dans la liste des paramètres, qui se distingue de la liste des ports par le caractère `'#'`.

Le module suivant a un paramètre `N` avec la valeur par défaut 1. Lors de l'instanciation du module `m`, cette valeur peut être changée. Le module `m` se compose d'un sous-module `a_1`, qui est une instanciation du module présenté ci-dessus. Lors de l'instanciation de modules, les signaux de l'instance supérieure

Tableau 2. Opérateurs dans Verilog.

Opérateurs logiques	Description	Exemple avec : <code>reg [1:0] A = 2'b01; reg [1:0] B = 2'b10; reg [1:0] C = 2'b00;</code>
!	NOT	<code>! A;</code> // 0 logique
&&	AND	<code>A&&B;</code> // '1 && 1' = 1A&&C; // '1 && 0' = 0 logique
	OR	<code>A C;</code> // '1 0' = 1 logique
Opérateurs bit à bit		
~	Not	<code>C&(~A);</code>
&	And	<code>A&B;</code>
	Or	<code>A C;</code>
^	Xor	<code>A^B;</code>
Opérateurs arithmétiques		
*	multiplication	<code>C = A*B</code> // = 2'b10
/	division	<code>C = A/B</code> // = 2'b00
+	addition	<code>C = A+B</code> // = 2'b11
-	soustraction	<code>C = B-A</code> // = 2'b01
**	exposant	
%	modulo	
Opérateurs de comparaison		
>; <;	plus grand ; plus petit	<code>if (A>B) (A>C)</code>
>=; <=;	plus grand ou égal ; plus petit ou égal	<code>if (A>=B) (A>=C)</code>
==	Égalité	<code>if (A == B)</code>
Autres opérateurs		
>>	décalage à droite	<code>A<<1</code> // A=2'b10
<<	décalage à gauche	<code>B>>1</code> // B=2'b01
{}	concaténation	<code>{A,B}</code> // 4'b0110
{ {} }	réplication	<code>{2{A}}</code> // 4'b0101

sont passés au niveau inférieur soit explicitement par le nom du signal, soit implicitement par la position dans la liste des ports (voir **listage 1**).

```
module m #(parameter N = 1) ( input wire a);
// sous-module a: instance a_1
a a_1(.a(a));
// le module a crée l'instance a_1
// Si le module instancié possédait
// aussi un paramètre N, cette ligne
// s'écrirait: a #(N(10))a_1(.a(a));
endmodule
```

Encore et toujours

Dans Verilog, la directive `always` (toujours) est un mot-clé important. Elle décrit un bloc procédural et est, comme son nom

l'indique, appelée en permanence. À la synthèse, elle sert à la description des logiques combinatoire et séquentielle. On ne peut affecter un signal à un signal de type `reg` qu'à l'intérieur d'un bloc `always`. Un bloc `always` peut avoir une liste de sensibilité qui assure que les instructions définies dans le bloc ne seront exploitées que si, depuis le dernier point de simulation dans le temps, une valeur a été affectée à l'un des signaux de la liste. Un bloc `always` est de ce fait comparable à un processus VHDL. Dans Verilog, une liste de sensibilité commence avec le caractère '@'. Ce caractère signifie que tous les signaux de la liste entre parenthèses qui suit seront surveillés. S'il ne faut surveiller que le front du signal, cela est précisé par l'un des mots-clés `posedge` ou `negedge`.

Pour décrire un circuit séquentiel synchrone, on utilise toujours `always @ (posedge clk)`. Tous les blocs d'un ensemble, même de niveaux hiérarchiques différents, qui dépendent d'un même signal d'horloge (`clk`) sont exécutés simultanément.

Listage 1. Nuage logique.

```
module logic_cloud_verilog
(
    input wire a,
    input wire b,
    input wire c,
    output reg d,
    output reg e
);

wire [3:0] abc;

// Affectation permanente
// L'opérande de gauche d'une affectation
// permanente doit être du type wire
assign abc = {a,b,c};
// concaténation de a,b c et d avec {}
// Le * dans la liste de sensibilité est un
// caractère générique
// qui signifie que tout signal se trouvant dans
// le bloc
// déclenche le bloc
always @ (*) begin
    d = 1'b0; // Cette affectation par défaut
// empêche la production de bascules implicites
    if (abc == 3'b001) begin // begin..end optionnel
        d = 1'b1;
    end else if (abc == 3'b101) begin
        d = 1'b1;
    end
end

always @ (*) begin // begin..end optionnel
    case (abc)
        3'b000 : e = 1'b1;
        3'b011 : e = 1'b1;
        default : e = 1'b0;
    endcase
end

endmodule
```

Listage 2. Bascule D avec réarmement asynchrone et validation des impulsions d'horloge.

```
module d_ff_verilog
(
    input wire clock,
    input wire reset_n,
    input wire ena,
    input wire d,
    output reg r_q,
    output reg r_qn,
    output wire w_qn,
    output wire w_notOut
);

assign w_qn = ~r_q; // Affectation permanente
// -> ceci
// sera un tampon 'not'
not not_1(w_notOut,r_q); // Utilisation d'une
// primitive Verilog

// Verilog-2001 : permet la séparation
// par virgule dans une liste de sensibilité
// always @ (posedge clock, negedge reset_n)
always @ (posedge clock or negedge reset_n) begin
    if (reset_n == 1'b0) begin
        r_q <= 1'b0; // Valeur de
        // réarmement de la sortie
        r_qn <= 1'b1;
    end else begin
        if (ena == 1'b1) begin // clock enable
            r_q <= d; // Affectation permanente
            // -> ceci sera un tampon 'not'
            r_qn <= ~r_q; // Ceci est traduit en
            // une seconde
            // bascule avec une sortie inversée
        end
    end
end

endmodule
```

Pour les débutants, c'est toujours la plus grosse difficulté conceptuelle. À l'intérieur du bloc `always`, il y a des affectations dites procédurales qui sont exécutées en séquence. Il y a deux affectations de ce genre, les affectations bloquantes ($A=B;$) et les affectations non bloquantes ($A<=B;$). Une affectation bloquante ($=$) est intégralement exécutée avant de passer à l'instruction suivante. Pour une affectation non bloquante ($<=$), on enchaîne sur la suite avant que les résultats précédents aient été entièrement traités. Le choix de l'une ou de l'autre n'est pas toujours évident et ne peut être décidé sans une connaissance approfondie de la structure interne du simulateur Verilog, qui fait partie de la spécification du langage. Les détracteurs de Verilog considèrent souvent ce fait comme la plus grande faiblesse de ce langage. En pratique, cela ne joue qu'un rôle secondaire, car on applique une règle générale : en logique combinatoire, on utilise une affectation bloquante ($=$), en logique séquentielle, une affectation non bloquante ($<=$). Si vous vous en tenez à cette règle, et sauf pour quelques exceptions exotiques, vous devriez échapper aux maux de tête. Pour en savoir davantage sur ces différentes affectations, on peut consulter [1].

Exemple

Le **listage 1** montre un module décrivant exclusivement de la logique combinatoire. Selon la règle générale ci-dessus, dans ce module, toutes les affectations doivent être bloquantes ($=$). Ce circuit n'a pas de fonction utile, mais utilise toute une série de constructions du langage qu'il est intéressant d'examiner de plus près. Pour commencer, on déclare un vecteur de 3 bits de type `wire`. Ensuite, on donne à ce vecteur les valeurs des entrées `a,b,c` par une affectation permanente. Comme `a,b,c` sont des `wire` larges de 1 bit, on les rassemble avant l'affectation avec l'opérateur de concaténation `{}`. **Important** : l'opérande à gauche d'une affectation permanente doit être de type `wire`. Le premier bloc `always` qui suit illustre l'utilisation de `if/else`. Une règle empirique stipule que pour éviter la création d'une bascule, toute variable du bloc qui reçoit une affectation pour une condition doit en recevoir une pour toutes les conditions. Une manière simple de respecter cette règle est d'exécuter une affectation inconditionnelle par défaut au début du bloc, comme ici l'instruction `d = 1'b0;`. Le deuxième bloc `always` montre un exemple d'une instruction `case` qui fonctionne comme en C. Le cas `default` quoique optionnel devrait toujours être présent.

Le **listage 2** représente une description d'une bascule avec un réarmement asynchrone et une validation des impulsions d'horloge. Selon la règle, toutes les instructions décrivant une logique séquentielle doivent être non bloquantes ($<=$). Dans cet exemple, c'est le cas de toutes les affectations à l'intérieur du bloc `always`, qui décrit une bascule inverseuse et une non inverseuse. La sortie non inverseuse est de plus inversée au début du module au moyen d'une primitive. Dans Verilog, les primitives sont des modules prédéfinis (par ex. NOT, AND, OR, XOR), qui peuvent être instanciés comme tout module normal. Ici, on instancie le module `not_1` et on affecte à la sortie le `wire w_notOut` et à l'entrée le `reg r_q`. L'affectation `assign w_qn = ~r_q;`, une ligne plus haut, a exactement le même effet que l'inversion avec la primitive. Le **figure 1** montre le résultat de la traduction



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.

février 2019

◇ Salon de l'analyse industrielle

05 au 06/02 – Paris

www.analyse-industrielle.fr

◇ C!PRINT

05 au 07/02 – Lyon

salon-cprint.com

◇ 20^e Colloque Annuel du Syndicat des Énergies Renouvelables

06 au 07/02 – Paris

ser-evenements.com

◇ DeciElec Embedded Systems

(Systèmes connectés et embarqués | Électronique de puissance | Management des systèmes énergétiques)

06 au 07/02 – Toulouse-Labège

www.decielec.com/fr/

◇ MEA2019 (More Electric Aircraft)

06 au 07/02 – Toulouse

www.see.asso.fr/site-manifestation/22807_mea2019

◇ Salon des entrepreneurs

06 au 07/02 – Paris

www.salondesentrepreneurs.com

◇ Mondial des métiers

07 au 10/02 – Lyon

www.mondial-metiers.com

◇ Aero'Nov Connection (industries aéronautique et spatiale)

13 au 14/02 – Massy

www.aeronov-connection.com

◇ BE POSITIVE (transition énergétique des bâtiments et territoires)

13 au 15/02 – Lyon

www.bepositive-events.com

◇ Lighting Days 2019

13 au 15/02 – Lyon

lighting-days.com/fr

◇ 5^e colloque national photovoltaïque

14/02 – Lyon

ser-evenements.com

◇ Techinnov

14/02 – Paris

techinnov.events

◇ Paris Manga & Sci-fi Show

16 au 17/02 – Paris

www.parismanga.fr

◇ Japan Expo Sud

22 au 24/02 – Marseille

www.japan-expo-sud.com/fr/

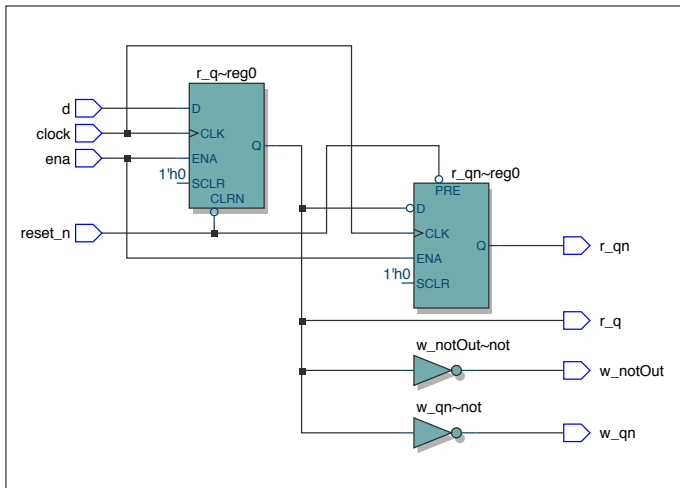


Figure 1. Bascule D dans la visionneuse RTL.

par l'outil de synthèse Quartus de la description du listage 2. Comme prévu, les deux sorties `w_notOut` et `w_qn` passent par un tampon inverseur, alors que les sorties `reg` sont directement connectées aux sorties des bascules. On remarque aussi que les deux signaux de la liste de sensibilité sont connectés aux deux entrées de la bascule sensibles au front du signal.

Simulation

Pour la simulation dans Verilog, on utilise un module, nommé *testbench* (banc de test), qui instancie le schéma à vérifier sous la forme d'un circuit à tester (*Device Under Test*, DUT) auquel il fournit les signaux appropriés. Le **listage 3** montre un banc de test utilisable pour la bascule du listage 2. Dans le banc de test, qui n'est généralement pas synthétisable, est produit un signal d'horloge à l'intérieur d'un bloc `always`. Pour introduire un décalage temporel des signaux, Verilog utilise le caractère '#'. Le bloc initial utilisé dans le banc de test possède les mêmes caractéristiques qu'un bloc `always`, sauf qu'il n'est exécuté qu'une seule fois au début de la simulation. Pour afficher un texte sur la sortie du simulateur, on utilise la fonction système `$display` (les fonctions système de Verilog commencent par un '\$' et ne sont pas synthétisables).

Conclusion

Bien entendu, Verilog possède bien d'autres fonctions qui ne peuvent pas être toutes mentionnées dans l'espace limité de cet article. Les exemples présentés ici fournissent toutefois une bonne approche de Verilog. Pour en savoir plus sur Verilog, on peut analyser une série d'exemples proposés par Edaplayground [2] ou se plonger dans l'un des nombreux tutoriels disponibles sur le web [3]. ◀

(180562-04 – version française : Helmut Müller)

Listage 3. Banc d'essai de la bascule D.

```
'timescale 1ns/1ps
// timescale est une directive du compilateur où
// 1 incrément de temps = 1 ns et
// résolution = 1 ps
module tb_dff;

    reg clk;
    reg rst_n;
    reg r_d;
    wire w_q;
    wire w_qn;
    wire w_qn_1;
    wire w_qn_2;

    // Créer une horloge à 50 MHz
    always begin
        #10; // Attendre un délai de 10 ns
        // avant d'inverser clk_tb
        clk = ~clk;
    end

    d_ff_verilog DUT(
        .clock      (clk),
        .reset_n    (rst_n),
        .ena        (1'b1),
        .d          (r_d),
        .r_q        (w_q),
        .r_qn       (w_qn),
        .w_qn       (w_qn_1),
        .w_notOut   (w_qn_2)
    );

    initial begin
        $display("Hello World ");
        clk = 0;
        rst_n = 0;
        r_d = 1'b0;
        #100;
        r_d = 1;
        rst_n = 1;
        #200;
        $stop();
    end

endmodule
```

Liens

- [1] 'Nonblocking Assignments in Verilog Synthesis, Coding Styles That Kill!', Clifford E. Cummings : www.sunburst-design.com/papers/CummingsSNUG2000SJ_NBA.pdf
- [2] EDA playground : www.edaplayground.com
- [3] Tutoriels et exemples Verilog : www.nandland.com/verilog/tutorials/index.html

recherche une/un **ARCHITECTE OPTOELECTRONIQUE****MISSIONS**

- Conception et fabrication de capteurs de mesure basés sur des modèles analogiques et numériques
- Réalisation de systèmes optoélectroniques avec intelligence embarquée
- Missions collaboratives :
 - Production de systèmes de mesure (parties optique et électronique)
 - Élaboration de projets intégrant optoélectronique, mécanique, outils de traitement du signal et de post traitement des données de mesure
- Réalisation de procédures de qualification des produits, mise en œuvre des tests, synthèse des résultats
- Veille : état de l'art des technologies des composants optoélectroniques et méthodes de conception de capteurs optoélectroniques
- Analyse du marché : recherche de produits innovants, brevets, concurrence

PROFIL

- Formation Bac+2 à Bac+5 en optoélectronique et traitement du signal, électronique embarquée type FPGA ou autre
- Expérience avérée en conception de circuits optoélectroniques et de capteurs optiques
- Expérience avérée en traitement analogique et/ou numérique du signal (Labview, VHDL, Verilog...)
- Connaissance de la programmation logicielle (par ex. : langage C)
- Anglais technique : niveau intermédiaire – Allemand : notions
- Valeur ajoutée :
 - connaissances mécaniques (phénomènes statiques et dynamiques) et optiques (diffusion, diffraction de la lumière)
 - connaissances dans la création de luminaires
- Être autonome et dynamique, à l'aise dans une petite entreprise, y faire preuve de réactivité et d'un bon esprit d'équipe.
- Être curieux et aimer créer, savoir détourner des composants pour leur donner une utilité nouvelle, préférer réparer et améliorer, penser qu'art et technique peuvent se combiner.

Optel Thevon : TPE spécialisée dans le développement, la fabrication et la commercialisation de capteurs optoélectroniques innovants depuis 1971. Clientèle exclusivement composée de grands comptes. Environnement et motivation type start-up.

Contact : job@optel-thevon.fr ou **01 43 58 75 26**

Poste en CDI, basé à Paris (métro Gambetta), temps partiel possible, déplacement peu fréquent.



Pour développer leur présence dans les pays francophones, les magazines **Elektor** et **MagPi** souhaitent renforcer en France leur **équipe de marketing et de forces de vente**, et recherchent activement :
leur cheville ouvrière commerciale dynamique et ambitieuse

MISSIONS

- Définition, élaboration et mise en pratique d'une stratégie commerciale adaptée au marché francophone
- Vente d'espace publicitaire (papier & en ligne) pour les magazines **Elektor** et **MagPi**
- Prospection et fidélisation de clients (notamment sur les salons professionnels)
- Recherche avec les annonceurs et leurs agences de solutions inédites pour répondre à l'évolution rapide des marchés
- Recherche de collaborateurs pour démultiplier votre activité
- Veille concurrentielle et recherche de modalités novatrices en collaboration avec la rédaction et les équipes internationales

PROFIL

- Solide formation de niveau supérieur
- Connaissance des médias et des réseaux de communication de l'électronique et de la micro-informatique francophones ; savoir identifier leurs besoins
- Expérience dans la vente de services et le développement commercial
- Goût et sens du commerce, à l'aise avec les chiffres
- Parler français et anglais, et peut-être l'allemand et/ou le néerlandais
- Et surtout :
 - Ne pas craindre de sortir des sentiers battus pour imaginer et proposer des solutions inédites
 - Aimer les défis et faire preuve d'imagination et pugnacité

Si vous pensez que vous pourriez être ce maillon essentiel entre nos magazines Elektor et Magpi et le marché francophone, veuillez nous contacter sans tarder : redaction@elektor.fr



HORS CIRCUITS

B.A.-BA des mélangeurs de fréquences radio

tout ce qu'il faut savoir pour bien les choisir

Robert Lacoste (Chaville)

Oscillateur local, fréquence intermédiaire, passif ou actif, simple ou double, gamme de fréquences... Vous êtes perdu ? Robert Lacoste démystifie les mélangeurs de fréquences radio et détaille les paramètres essentiels pour ne pas se tromper.

Quand j'ai commencé à jouer avec un fer à souder, à l'âge de 11 ou 12 ans (c'est-à-dire il y a plus de 40 ans), mon père m'a montré son stock de vieux magazines d'électronique. Un vrai trésor ! De nombreux articles présentaient des schémas de récepteurs radio, avec des termes plus qu'étranges comme « mélangeur », « fréquence intermédiaire » ou « superhétérodyne ». J'ai essayé de comprendre, mais en vain. Comment une simple diode pouvait-elle transformer une fréquence en une autre ? Par facilité, je me suis réfugié pendant nombre d'années dans le numérique et ses '0' et '1' plus faciles à maîtriser. Je n'ai recroisé le monde de la radio que bien plus tard, au hasard de mon parcours professionnel. Comprendre enfin comment fonctionne un mélangeur de fréquences a été une vraie révélation ! Et vous, savez-vous comment ça marche ? Non ! Cet article est donc pour vous.

C'est quoi un mélangeur ?

Commençons par un exemple. Imaginez que vous souhaitiez réaliser un récepteur pour recevoir un signal émis à la fréquence de 340 MHz (ce n'est bien sûr qu'un exemple). Vous vous doutez qu'il est plus facile de manipuler un signal électrique à basse fréquence plutôt qu'à 340 MHz,

non ? Il est donc utile de « décaler » ce signal vers une fréquence inférieure avant de le traiter. Supposons que vous souhaitez le transposer à la fréquence de 40 MHz, où l'on trouve des filtres sélectifs bon marché. Comment diable passer de 340 MHz à 40 MHz sans perdre la modulation du signal qui nous intéresse ? Avec un mélangeur pardi ! Il s'agit d'un circuit électronique qui permet de transposer une fréquence vers une autre, plus basse (pour un récepteur) ou plus élevée (pour un émetteur).

Commençons par considérer ce mélangeur comme une boîte noire. Il possède un port d'entrée, un port baptisé OL (pour « Oscillateur Local ») et un port de sortie (**fig. 1**). Pour un récepteur, l'entrée est traditionnellement appelée « RF » et la sortie « FI » (Fréquence Intermédiaire) ; c'est l'inverse pour un émetteur. Comment se comporte un mélangeur parfait dans le cas d'un récepteur ? Si on applique un signal de fréquence F_{RF} sur son entrée RF et un signal de fréquence F_{OL} sur son entrée OL, il délivre sur sa sortie FI la somme de deux signaux, de fréquences respectives $F_{RF} - F_{OL}$ et $F_{RF} + F_{OL}$. Un exemple pour que cela soit plus clair ? Pour notre récepteur FM, on pourrait utiliser $F_{OL}=300$ MHz. Le signal de sortie contiendra des signaux de fréquences $340-300=40$ MHz et $340+300=640$ MHz, et un filtre passe-bas isolera facilement le premier.

Voici quelques autres caractéristiques avant de voir comment ça marche. Un mélangeur est un dispositif absolument linéaire, tant qu'il est utilisé correctement : le signal en sortie aura une amplitude double si l'amplitude du signal d'entrée double, la modulation d'amplitude est donc conservée. De même la modulation de fréquence ou de phase d'un signal appliqué en entrée se retrouve inchangée sur la sortie. Enfin, et c'est important, de nombreux mélangeurs sont des dispositifs passifs et réversibles : avec le même mélangeur, vous pouvez injecter un signal sur son port FI et utiliser le port RF comme sortie, ou l'inverse.

Une pincée de trigonométrie...

Dans cette rubrique, je fais tout mon possible pour éviter les maths, mais le fonctionnement des mélangeurs repose sur une petite formule trigonométrique. Respirez normalement et répétez après moi :

$$\cos(a).\cos(b) = \frac{1}{2} [\cos(a+b) + \cos(a-b)]$$

Vous avez sûrement déjà vu cette formule. Regardez-la bien, elle montre que la multiplication de deux sinus donne un signal qui est la somme de deux sinus, respectivement de fréquences 'somme' et 'différence'. Si vous préférez une version graphique de cette formule, regardez la **figure 2**. Dans cet exemple, j'ai multiplié un signal sinusoïdal de fréquence 300 MHz par un autre de fréquence 340 MHz. Le résultat est bien une sinusoïde lente de 40 MHz ($340-300$), additionnée à une sinusoïde à haute fréquence de 640 MHz ($340+300$).

Vous avez tout compris ou presque. On peut donc construire un mélangeur de fréquences avec n'importe quel circuit qui « multiplie » deux tensions. En fait, cela revient à dire que ça marchera peu ou prou tant qu'on utilise un circuit électro-

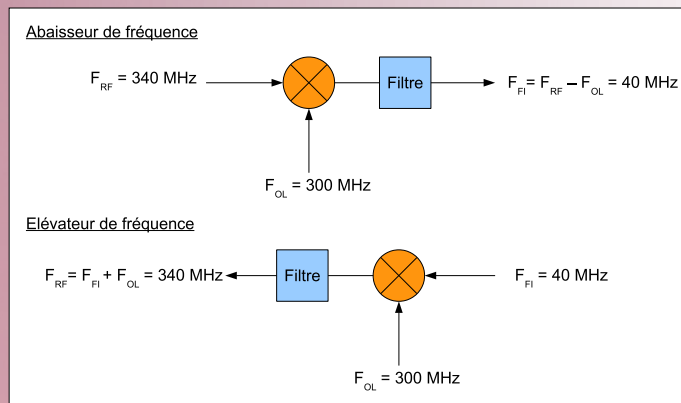


Figure 1. Un mélangeur à trois ports : une entrée pour le signal, une entrée pour l'oscillateur local (OL) et une sortie. Il peut être utilisé pour élever ou abaisser la fréquence d'un signal RF.

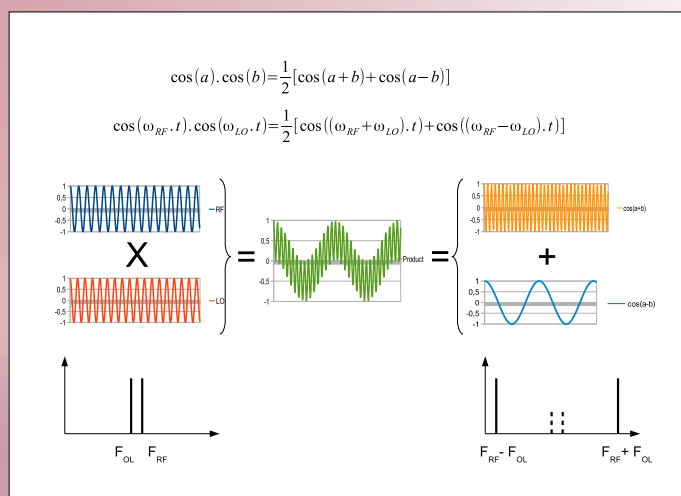


Figure 2. Lorsque deux signaux sinusoïdaux sont multipliés, le résultat est la somme de deux sinus, de fréquences 'somme' et 'différence'. C'est le mécanisme de base utilisé par un mélangeur.

nique dit « non linéaire ». Pourquoi ? Imaginez que vous avez deux tensions v_1 et v_2 . Ajoutez-les, et envoyez-les à travers un circuit non linéaire. Sa sortie sera une fonction non linéaire de $x=v_1+v_2$, appelons-la $f(x)$. Si elle était linéaire, on aurait $f(x)=ax+b = a(v_1+v_2)+b$, et il ne se passerait rien de palpitant. Mais si la fonction est non linéaire, elle pourra toujours être approximée par un polynôme, disons $f(x)=a+bx+cx^2+$ etc. Regardons juste le terme cx^2 :

$$cx^2 = c(v_1+v_2)^2 = c \times v_1^2 + c \times v_2^2 + 2c(v_1 \times v_2)$$

C'est gagné ! Toute fonction non linéaire fait apparaître le produit des deux tensions d'entrée. Ainsi, n'importe quel dispositif non linéaire pourrait être utilisé comme mélangeur, mais bien sûr, certains sont plus efficaces ou plus « propres » que d'autres !

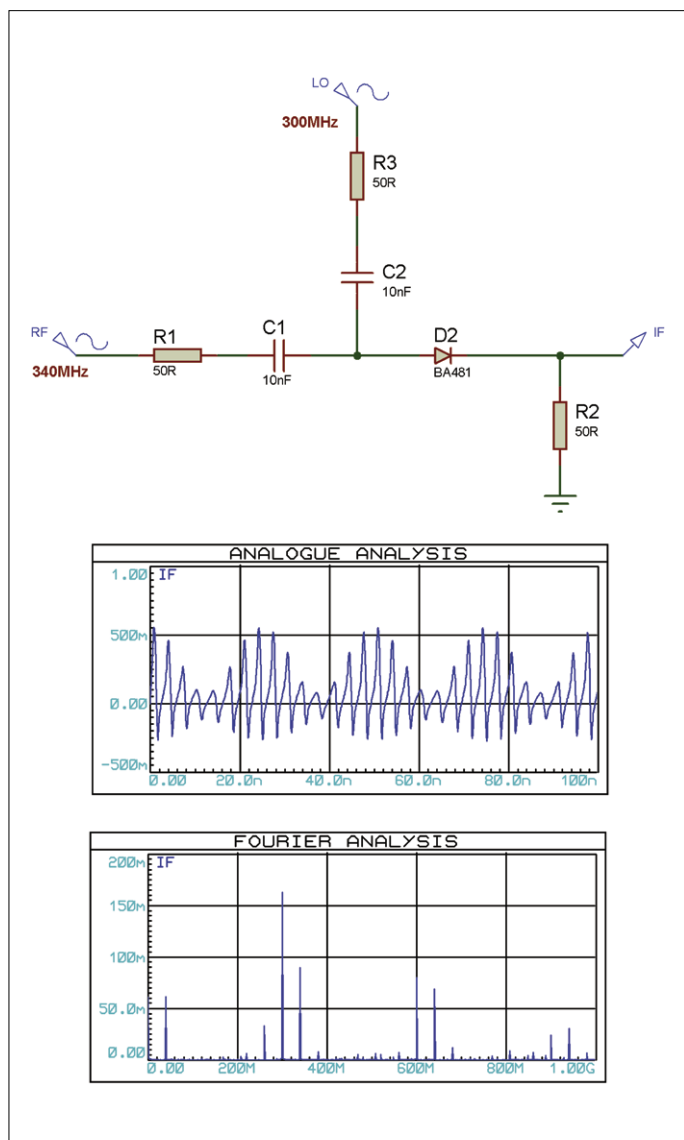


Figure 3. Simulation d'un mélangeur réalisé avec une simple diode. Les fréquences RF et OL sont respectivement 340 et 300 MHz. Sur le spectre du signal de sortie, ces deux fréquences sont encore visibles, mais aussi les fréquences 'différence' (40 MHz) et 'somme' (640 MHz), entre autres.



Figure 4. Un bel exemple de mélangeur à diode, extrait si je me souviens bien d'un analyseur de spectre HP8551 de 1965. La diode est la petite cartouche, encapsulée à l'intérieur du connecteur BNC visible à l'arrière.

Soulevons le capot

Quel est le dispositif électronique non linéaire le plus simple ? C'est la diode, car le courant qui la traverse n'est pas proportionnel à la tension appliquée à ses bornes. Cela nous amène au bon vieux schéma du **mélangeur à diode simple**, dit **asymétrique** (**fig. 3**). Sur cette figure, j'ai également représenté le résultat de la simulation du circuit, réalisée sous Proteus (Labcenter), vous pouvez le reproduire facilement avec tout simulateur Spice.

Pour mieux comprendre ce schéma, sachez qu'en général le signal de l'oscillateur local (OL) a une puissance bien supérieure à celle du signal d'entrée. Ce signal de l'OL pilote donc le comportement de la diode : lorsque le signal de l'OL est suffisamment positif, la diode est conductrice, alors que celle-ci est bloquée si le signal de l'OL est dans son alternance négative. De là, le petit signal appliqué sur le port RF se retrouve sur la sortie, mais haché par les alternances du signal de l'OL. On multiplie donc en fait le signal RF par un signal carré de même fréquence que l'OL. Cela donne un mélangeur, mais plus complexe que l'exemple élémentaire présenté plus haut. En effet un signal carré comprend la fréquence fondamentale, mais aussi tous les harmoniques impairs du signal de l'OL. De là un tel mélangeur à base de diodes crée les fréquences 'somme' et 'différence', mais aussi de nombreuses autres fréquences. En fait, il produit la somme de toutes les fréquences de la forme $(n \times F_{OL} + m \times F_{RF})$, où n et m sont des entiers quelconques, positifs ou négatifs. On aura donc en sortie, comme prévu, les fréquences $F_{RF} + F_{OL}$ et $F_{RF} - F_{OL}$, mais aussi des fréquences plus éloignées comme $2 \times F_{RF} - F_{OL}$ ou $7 \times F_{RF} - 5 \times F_{OL}$! Comme nous le verrons plus loin, cela rend la vie plus drôle...

Même s'ils sont utilisés de temps en temps comme l'illustre la **figure 4**, ces mélangeurs à simple diode ont des inconvénients : en particulier, les signaux d'entrée (RF et OL) se retrouvent sur la sortie avec un fort niveau. On dit que leur « isolation » est faible. Des topologies plus complexes ont donc été inventées, comme le **mélangeur équilibré simple** (qui utilise deux diodes et atténue fortement le signal soit LO soit RF, mais pas les deux), ou le très populaire **mélangeur équilibré double**. Celui-ci est construit avec quatre diodes et deux transformateurs (**fig. 5**). Il atténue fortement les signaux RF et LO. Regardez la figure 5 qui montre la simulation de son fonctionnement en utilisant les mêmes signaux RF et OL que pour la figure 3. Vous verrez que le spectre de sortie est beaucoup plus propre : les fréquences 'différence' et 'somme' sont bien visibles, même si d'autres signaux sont présents, mais à des niveaux inférieurs.

Comment fonctionne un tel mélangeur doublement équilibré ? Comme pour le mélangeur simple, le signal de l'oscillateur local, qui a une forte puissance, rend successivement chaque paire de diodes passante. La polarité de la connexion entre les transformateurs d'entrée et de sortie est ainsi inversée à chaque alternance du signal de l'OL, ce qui permet d'obtenir le même type de découpage qu'avec la configuration à diode unique, mais avec une bien meilleure isolation et une réduction des signaux parasites grâce à l'architecture équilibrée.

Et dans la vraie vie ?

Ces simulations sont très sympathiques, mais voyons à quoi ressemble un mélangeur en vrai. Pour cela, rien ne vaut l'ex-

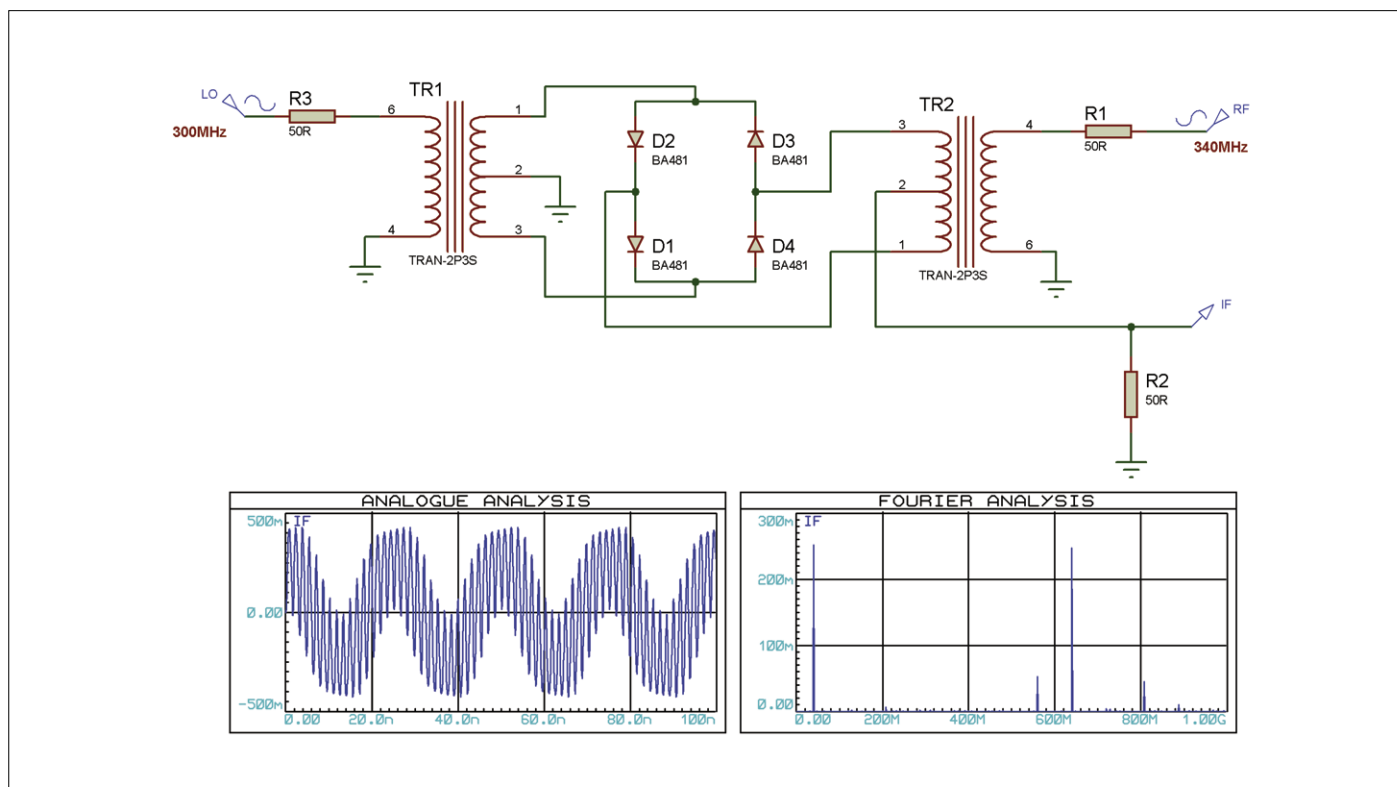


Figure 5. Un mélangeur équilibré est un peu plus complexe, mais le spectre produit est beaucoup plus propre. Comparez avec la figure 3 pour vous en convaincre !

périence. J'ai retrouvé dans un tiroir un mélangeur à large bande doublement équilibré de la société Macom (**fig. 6**). Après avoir branché son port OL à un générateur RF de labo, réglé sur 300 MHz et une puissance de +10 dBm (soit $1 \text{ mW} \times 10^{10}/10 = 10 \text{ mW}$), j'ai connecté son port RF à un second générateur réglé sur 340 MHz et une puissance variable. Enfin, j'ai connecté un analyseur de spectre au port FI et j'ai observé. Vous pouvez voir le résultat de l'expérience sur les figures 7 à 9, respectivement une puissance de signal RF de -20 dBm (10 μW), -10 dBm (100 μW) et 0 dBm (1 mW).

Commençons par la **figure 7** où la puissance RF injectée est faible (-20 dBm, soit 10 μW). Vous retrouvez le signal qu'on attendait à 40 MHz ? L'échelle est de 100 MHz par division horizontale, c'est donc la raie la plus à gauche, à 0,4 division du bord de l'écran. La fréquence de 640 MHz est aussi visible (6,4 divisions), mais il y a aussi plein d'autres fréquences parasites : 0 Hz, 260 MHz, 300 MHz, 340 MHz, 560 MHz, 600 MHz, 860 MHz, 900 MHz, 940 MHz, etc. Ce sont les autres termes non linéaires de l'équation, tous de la forme $(n \times F_{OL} + m \times F_{RF})$ comme attendu.



Figure 6. Pour mes expériences, j'ai utilisé ce mélangeur MACOM. Il est spécifié jusqu'à 4 GHz.

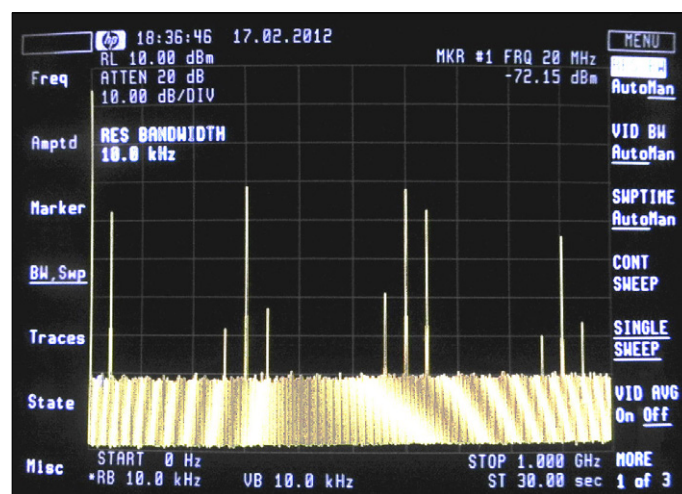


Figure 7. Le spectre de sortie du mélangeur MACOM, avec RF=340 MHz à une puissance de -20 dBm.

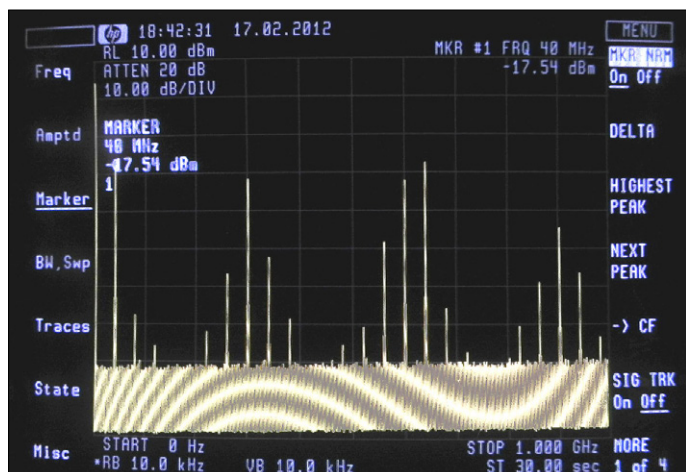


Figure 8. Même expérience avec une puissance d'entrée de -10 dBm, les fréquences parasites sont plus fortes et plus nombreuses.

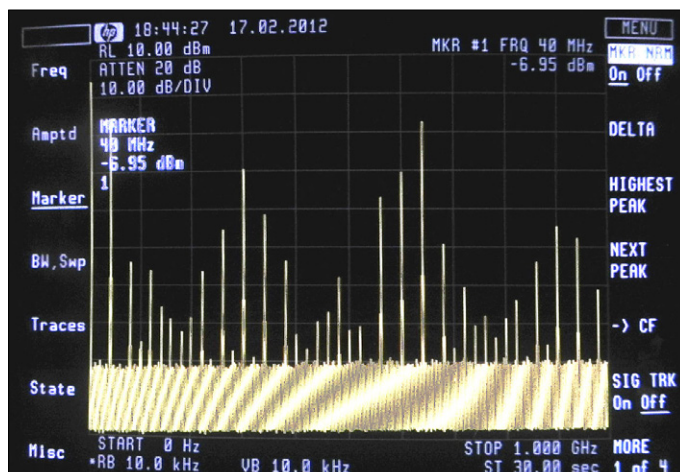


Figure 9. Enfin avec une puissance d'entrée de 0 dBm. Beaucoup d'harmoniques et de parasites sont visibles.

Regardez maintenant les **figures 8 et 9**, où la puissance RF injectée est plus élevée (respectivement -10 dBm et 0 dBm). Vous voyez comme les signaux parasites sont de plus en plus nombreux et de plus forte amplitude lorsque l'amplitude du signal d'entrée augmente ? C'est tout simplement parce que les harmoniques produits par un dispositif non linéaire comme

une diode sont... non linéaires. Leur amplitude augmente beaucoup plus vite que l'amplitude du signal d'entrée... Un très bon exercice est d'essayer de déterminer d'où provient chaque raie du signal de sortie. Je l'ai fait pour vous sur l'exemple du signal le plus fort (0 dBm), regardez le résultat sur la **figure 10** : rien n'est dû au hasard, tout s'explique !

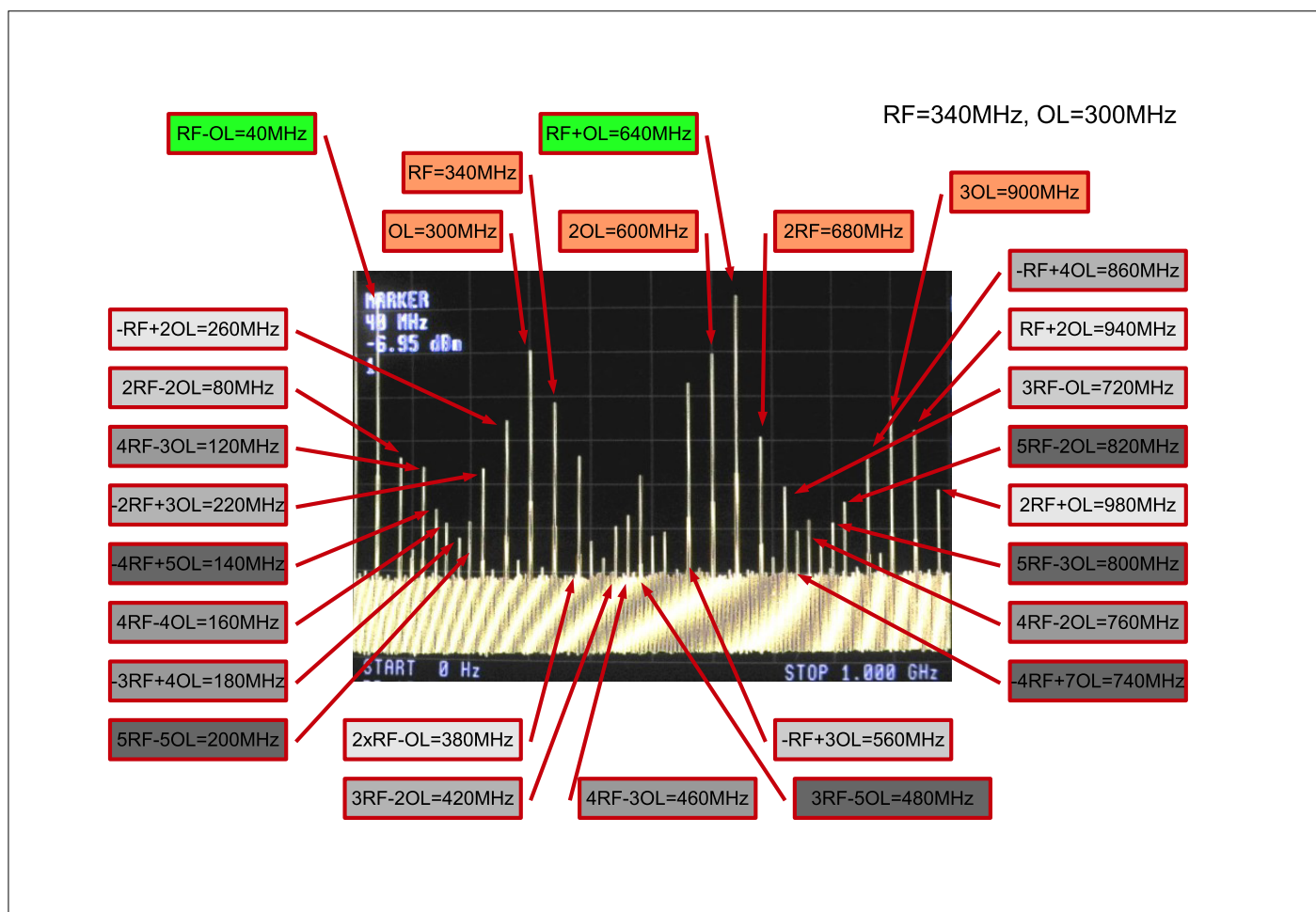


Figure 10. À partir du spectre de la figure 9, il est intéressant de rechercher l'origine de chaque raie parasite. Je l'ai fait pour vous...

Critères de choix

Comment choisir un mélangeur pour votre prochain projet RF ? Vous pouvez fouiller dans les catalogues des fournisseurs spécialisés comme Mini-Circuits, Synergy Microwave, Analog Devices, Maxim et les autres, mais à quoi faut-il faire attention ? Le premier paramètre est bien sûr la [gamme de fréquences](#), plus exactement les gammes de fréquences puisque chaque mélangeur est spécifié pour une gamme donnée sur chacun des ports : RF, OL et FI. N'oubliez pas que les ports RF et FI peuvent être permutés au moins pour les mélangeurs passifs (certains mélangeurs sont « actifs » en ce sens qu'ils incluent un amplificateur sur le chemin RF et/ou OL). Vous devrez également vérifier l'impédance caractéristique des ports du mélangeur : la plupart sont conçus pour une impédance standard de 50 Ω , mais ce n'est pas toujours le cas, en particulier pour les circuits intégrés. Dans ce cas, vous devrez concevoir un circuit d'adaptation d'impédance pour obtenir les meilleures performances.

L'autre paramètre important est le [niveau du signal de l'OL](#). Chaque mélangeur est spécifié pour une puissance donnée qui doit être appliquée sur ce port. La règle de base ? Pour qu'un mélangeur fonctionne correctement, la puissance de l'OL doit être supérieure de plusieurs ordres de grandeur à la puissance du signal d'entrée. Une différence d'au moins 10 dB est un strict minimum, mais plus elle est grande, plus le spectre de sortie sera propre. Vous découvrirez donc que les meilleurs mélangeurs nécessitent des puissances de l'OL très élevées, ce qui n'est pas toujours facile à fournir.

Vérifiez ensuite la [perte de conversion](#) du mélangeur. Il s'agit de l'efficacité de la conversion de fréquence de RF à FI ou vice-versa. Pour un mélangeur passif, elle est théoriquement d'au moins 3 dB, car la puissance est répartie entre les fréquences 'somme' et 'différence', mais en pratique elle est de l'ordre de 6 à 10 dB. Un dernier mot sur les pertes de conversion : vous vous souvenez quand j'ai parlé de facteur de bruit il y a quelques mois (cf. [1]) ? Le [facteur de bruit](#) d'un mélangeur passif est égal à sa perte de conversion, donc plus cette perte est faible, mieux c'est pour un récepteur...

Quoi d'autre ? Vous pourrez également vérifier les isolations du mélangeur, c'est-à-dire l'[amplitude des fuites entre les ports](#) : de OL vers RF et de OL vers FI, mais aussi de RF vers FI. Ces fuites sont liées à la topologie du mélangeur, à la qualité

et à l'appairage des composants internes. Plus l'équilibre du mélangeur est bon, plus les fuites seront faibles. Des valeurs entre 20 et 30 dB, c'est déjà très bien. Enfin, le paramètre appelé [P1dB](#) est le point de compression à 1 dB du mélangeur. C'est la puissance d'entrée pour laquelle la perte de conversion du mélangeur est supérieure de 1 dB à sa valeur à bas niveau. Ce P1dB correspond donc à une puissance où le mélangeur « compresse » le signal d'entrée en raison de sa propre saturation. En général on reste assez loin de cette puissance limite, car les signaux parasites croissent vite dans ces eaux-là... Pour aller plus loin, je vous recommande une très bonne note d'application publiée par Mini-Circuits sur les paramètres d'un mélangeur (cf. [2]).

Conclusion

Comme d'habitude, j'ai l'impression d'avoir couvert seulement 10% du sujet, et je n'ai probablement pas été assez précis pour ces 10%-là. Quoi qu'il en soit, j'espère que vous avez apprécié cette rapide plongée dans le monde des mélangeurs. Bien qu'ils puissent sembler plus exotiques qu'une porte logique ou un microcontrôleur, les mélangeurs sont amusants et pas si complexes une fois les bases comprises. Je vous encourage donc chaudement à expérimenter. Même si vous n'avez pas de générateurs RF et d'analyseur de spectre sous la main, vous pouvez bricoler des expériences intéressantes avec des bouts de ficelle : un mélangeur fonctionne en effet aussi en basse fréquence ! Reproduisez donc les schémas proposés pour des fréquences audio, en utilisant par exemple la carte son de votre PC comme analyseur de spectre.

Dans le prochain article, je continuerai dans le même esprit en vous présentant une autre famille de mélangeurs : les modèles I/Q. ◀

(180661-01)

Cet article a été publié dans la revue Circuit Cellar (n°263, juin 2012).

Liens et références

- [1] « bruit et sensibilité des récepteurs - une introduction », Elektor 12/2016 : www.elektormagazine.fr/160307
- [2] 'Understanding Mixers – Terms Defined, and Measuring Performance', Mini-circuits : www.minicircuits.com/app/AN00-009.pdf
- [3] Mélangeur de fréquence s(en anglais) : http://en.wikipedia.org/wiki/Frequency_mixer
- [4] Tutoriel en anglais sur les mélangeurs HF et le mélange HF : www.radio-electronics.com/info/rf-technology-design/mixers/rf-mixers-mixing-basics-tutorial.php
- [5] Tutoriel en anglais sur les mélangeurs HF et microondes, Marki Microwave : www.markimicrowave.com/menus/appnotes/mixer_basics_primer.pdf
- [6] 'RF Mixers', Iulian Rosu, YO3DAC / VA3IUL : www.qsl.net/va3iul/RF%20Mixers/RF_Mixers.pdf
- [7] Suite logicielle de CAO électronique & simulateur Isis Proteus : www.labcenter.co.uk
- [8] MD-525-4, 'double balanced mixer' de MACOM Technology Solutions Inc. : www.macomtech.com



mise en conformité et régulation de la tension du secteur

plus question de péter les plombs !

Jan Buiting (rédacteur de la rubrique Rétronique)

Comme la tension du secteur est maintenant stable et fiable, on pourrait croire qu'elle est bien conforme dans nos ateliers d'électronique. Il faut avoir subi un choc, constaté le claquage d'un fusible ou les bizarreries d'un appareil pour envisager que la distribution du courant à 230 V~ fait des siennes. Il en va tout autrement avec les appareils de collection, surtout quand vous les allumez pour la première fois : le réseau est alors un souci capital, voire mortel. Voyons comment éviter de ruiner nos chers appareils au moment de les tirer d'un long sommeil. Et aussi comment nous protéger.

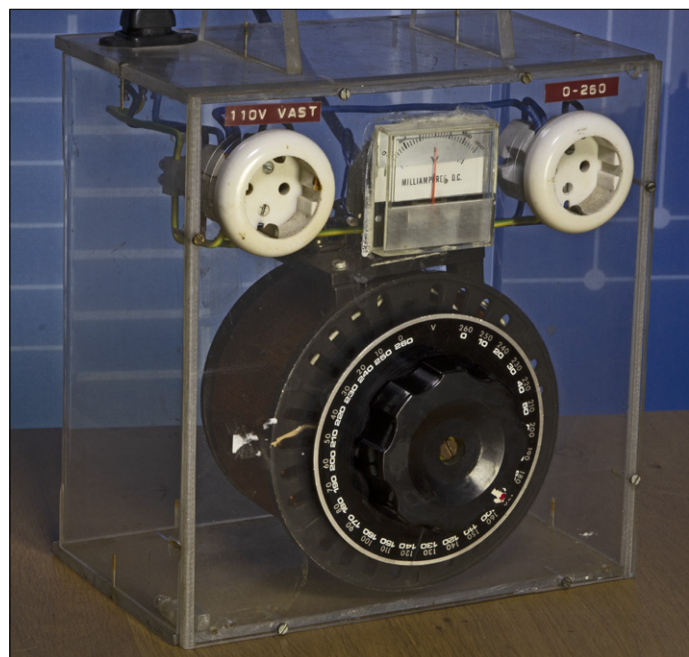


Figure 1. Le variac de 850 VA du labo d'Elektor est maintenant confié aux doigts habiles des rétroniciens.

Voici un nouvel épisode de rétronique, l'aînée des chroniques régulières du magazine Elektor. Cette fois, il s'agit de l'alimentation secteur CA et de la manière sûre d'y brancher des appareils anciens ou de la distribuer aux lieux de travail et labos, même pour de l'attirail moderne. Quelles sont les options ?

Variac

Variac était (de 1934 à 2002) une marque de fabrique de General Radio aux États-Unis pour désigner un autotransformateur variable. C'est un moyen sûr de faire monter la tension de ligne de 10 V~ environ jusqu'à la tension nominale et même un peu plus, soit de 10 à 250 V~. La version de base est un autotransformateur à large bouton de réglage manuel et avec un cadran gradué pour indiquer la tension de sortie approximative. À l'intérieur, le bouton entraîne, le long d'une glissière, un curseur mobile en carbone qui peut faire contact avec n'importe quelle position du bobinage en fil de cuivre de l'autotransformateur. La position relative du curseur détermine la tension de sortie, comme le frotteur d'un potentiomètre. Le gros variac réglé à distance a une commande par moteur, avec le grand avantage de vous dispenser de la proximité d'une chose encombrante, extrêmement lourde et qui ronronne sans cesse sur ou sous votre bureau.

Un variac consiste en une inductance de dizaines de henrys qui peut produire un son étrange, comme un petit coup de sifflet, au moment de l'allumage, même sans charge. Ou bien faire sauter les plombs.

Comme tout autotransformateur, le variac n'a qu'un seul bobinage qui, bien que doté d'une dérivation, n'isole pas la charge de la tension du secteur. Certains débutants pourraient s'imaginer que, puisque la sortie ne porte que quelques dizaines de volts, c'est une basse tension, donc inoffensive. Basse, d'accord, mais la pleine tension du secteur (L) peut se trouver sur n'importe quelle extrémité si la prise de courant n'est pas « polarisée », pour repérer le neutre, ce qui n'est que rarement le cas en France comme aux Pays-Bas.

Le variac de la **figure 1** a été longtemps un fidèle compagnon au labo d'Elektor, mais il a été mis au placard, vu qu'on ne s'en servait plus. Il donne 4 A, donc un bon 850 VA. Nous l'utilisons pour les projets d'amplificateur audio de grande puissance pour atteindre graduellement en sécurité la tension d'alimentation voulue sans disjoncter et jeter tout le labo dans la débâcle. Il a repris du service maintenant pour certains appareils qui, dans le cadre de la rétronique, ont besoin d'attention lors du redémarrage, surtout pour éviter le danger inhérent d'explosion des condensateurs électrolytiques HT.

À l'origine, le variac est un modèle pour le 220 V des années 1970 avec une gamme légèrement différente de la tension nominale actuelle de 230 V~. Son voltmètre a pâle figure, il n'est fiable que jusqu'à 150 V environ, on lui adjoint donc un multimètre numérique Fluke. À part cela, je considère ce variac comme un instrument précieux dont je ne pourrais pas me passer pour la remise en service d'un nouveau revenant des temps anciens. Par sécurité électrique, mais aussi pour impressionner tous ceux qui ne travaillent que sur de modestes tensions, le variac rétronique d'Elektor est logé (**figures 2 et 3**) dans un boîtier en Plexiglas™ muni d'une poignée de transport faite de gros câble électrique gainé de PVC. Avec ses 12 kg, les plus jeunes le trouvent bien lourd. Le curseur fait des étincelles quand il est chargé à 800 W.

Avec des alimentations en CMS et certaines à ajustage automatique pour 110 à 240 V~, le variac n'est pas recommandé. Il m'est arrivé de griller l'alim du type inverseur/oscillateur d'un oscilloscope en utilisant le variac pour faire monter lentement la tension alternative. Trop lentement, parce que l'alim veut sa tension nominale très vite. Lors du parcours en dessous de ce niveau (comme 150 à 220 V~), l'oscillation n'arrive pas à démarrer et un courant continu excessif et prolongé circule dans certains enroulements du transformateur avec des conséquences fâcheuses. Le variac ne convient pas non plus pour des charges trop inductives ou capacitives qui peuvent entrer en résonance ou faire sauter le fusible domestique.

On trouve parfois un variac à prix ridiculement bas de nos jours, il prenait la poussière ou bloquait une porte. Je vous conseille d'en prendre un quand vous le pouvez.

Stabilisateur de tension pour quelqu'un ?

Dans le temps, les immeubles qui abritaient de grosses machines électriques et de l'électronique de bureau, allègrement branchées sur le seul et même câble d'alimentation de puissance, subissaient des pics de tension, des baisses, des pannes et des interférences, sources constantes d'ennuis. Certains fournisseurs de télécoms ont même refusé d'installer des centraux téléphoniques et d'autres dispositifs sensibles si

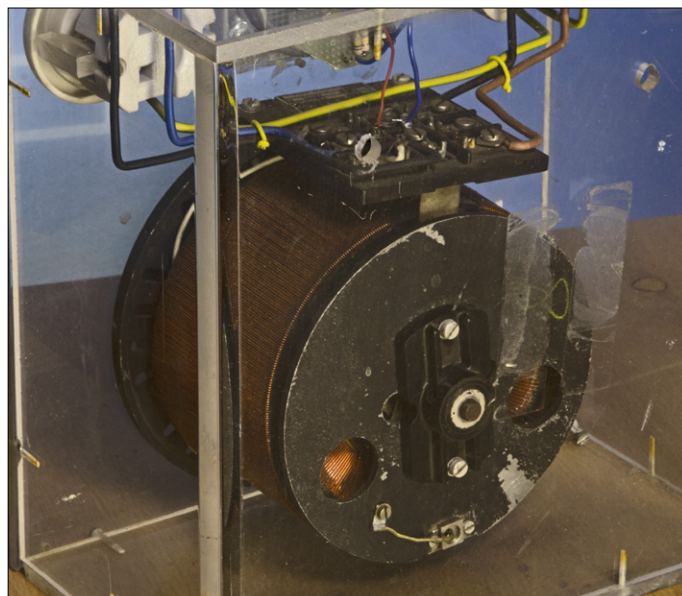


Figure 2. L'arrière du variac « portable » dans sa loge transparente en Plexiglas. Le contact mobile, le balai, n'est pas visible, il est entre la bobine et le couvercle rond.

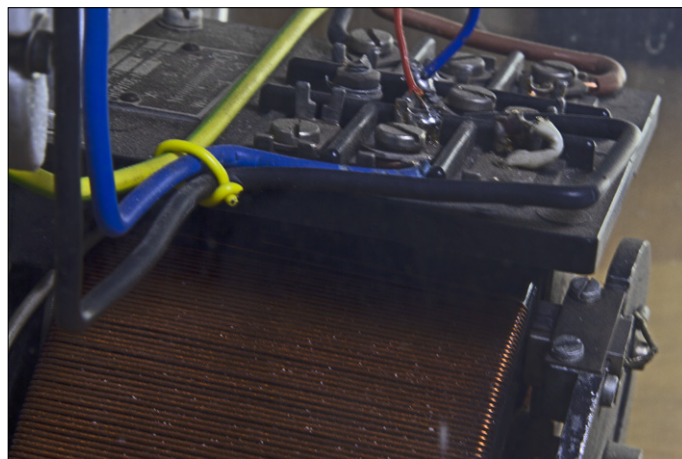


Figure 3. Il y a une belle quantité de précieux fil de cuivre caché dans le variac. Voyez ici le bornier à vis et les spires de l'autotransformateur avec un peu de poussière sur le fil de cuivre.

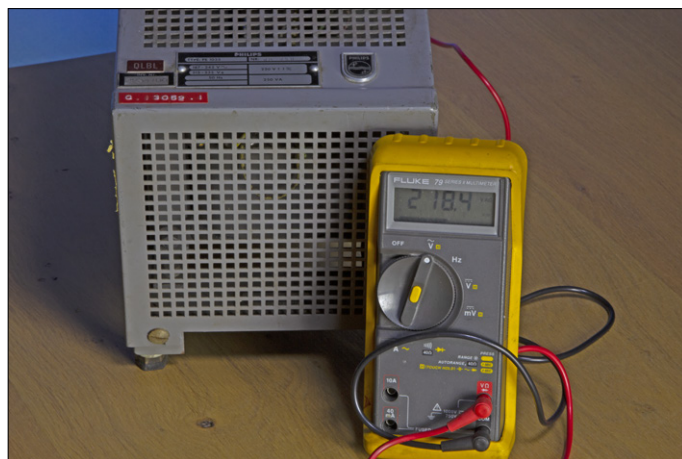


Figure 4. Délaisse pendant plus de 30 ans, le PE1033 n'a rien perdu de sa stabilité : la tension de sortie reste toujours dans la fourchette spécifiée de 1 % autour de 220 V~. Tension d'entrée : 232 V~.

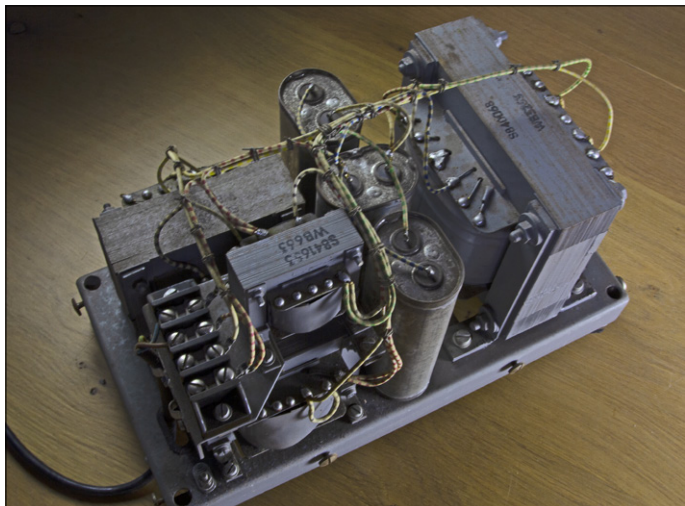


Figure 5. Le stabilisateur du secteur PE1033 ne contient que d'énormes transformateurs et des condensateurs.

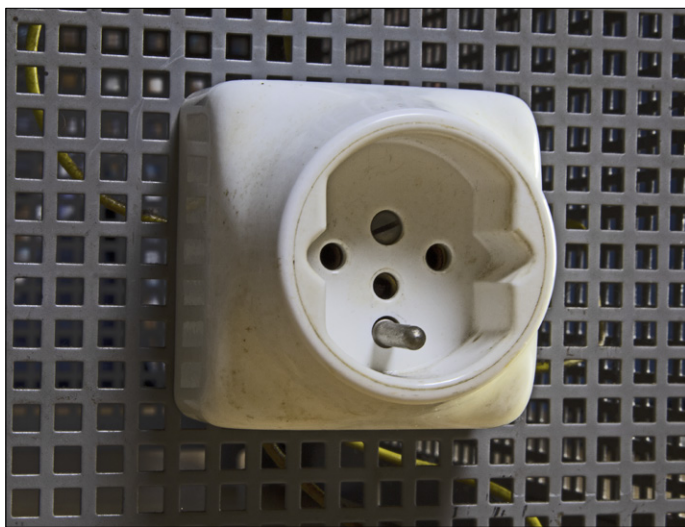


Figure 6. Un quidam a fixé une prise secteur en porcelaine sur le boîtier du PE1033. Elle donne à penser que l'instrument est portable, ce qui est faux, il est fait pour travailler à demeure.

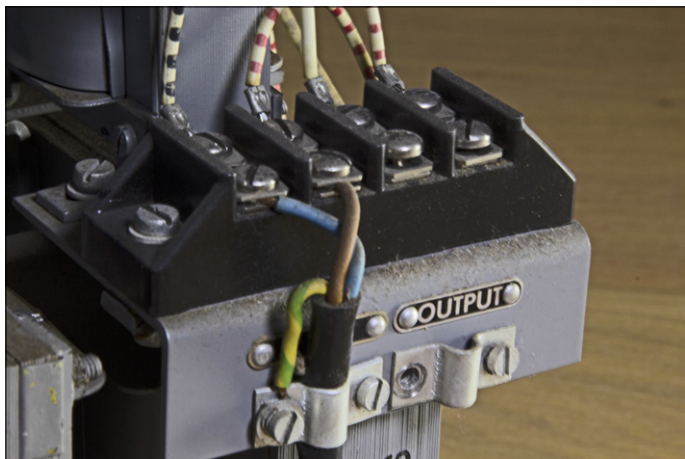


Figure 7. Le stabilisateur secteur PE1033 de Philips des années 1960 a été conçu pour un câblage permanent sur le réseau électrique 220 V~ dans les labos, les usines et les bureaux, dans un endroit à l'écart.

l'alimentation électrique se révélait déficiente. Idem pour l'éclairage des bureaux, les calculateurs et autres ordinateurs et serveurs. Et imaginez les pertes de tension supplémentaires aux États-Unis où la tension de service était à peine de 110 à 117 V~ comparée aux 220 V~ à travers l'Europe.

Solution dans les labos et les usines, on a installé des stabilisateurs du secteur dans des endroits soigneusement choisis, tant du point de vue électrique qu'acoustique ! On voit un tel appareil à la **figure 4**, en service adéquat certifié par la lecture du multimètre Fluke à 218.4 V~ ! Cette chose grise est un PE1033 Philips des années 1960, avec les caractéristiques suivantes :

- tension de sortie : 220 V~ ± 1 %
- puissance de sortie : 250 VA
- tension d'entrée : 187 à 242 V~
- puissance d'entrée : 315 à 335 VA
- fréquence : 50 Hz

Mis à part ce qu'il y a sur sa plaque signalétique, je ne sais rien de plus de cet appareil. Il y a dedans quatre transformateurs, trois gros condensateurs (5 μ F/500 V) et le câblage (**fig. 5**). Le PE1033 pèse 25 kg, il est assez compact (32×18×17 cm), donc malcommode à déplacer. Il repose sur quatre pieds en caoutchouc et ce n'est pas que pour la sécurité électrique, on s'en rend compte quand on l'allume. Pas de bruit sourd, mais un bourdonnement constant insupportable. J'ai lu sur internet que ce bruit provenait des transformateurs travaillant avec de fortes pertes dans le fer pour maintenir la résonance à 50 Hz avec les condensateurs. Une stabilisation à 1 % et la suppression de tout ce qui transite par les fils du secteur, sauf le 50 Hz : comment est-ce possible ? Chez moi (souvent 230 à 235 V), le PE1033 présentait une forme d'onde qui, comparée à celle d'entrée, était aussi propre qu'un sifflement. Lors de la mesure, la tension sur ma paillasse était lourdement infestée de salves et de surtensions par mon installation photovoltaïque de 7 kWc qui sortait à peine 4,8 kWc par un après-midi nuageux.

Impensable de nos jours, la perte de puissance subie avec ce type de stabilisateur : avec une charge de 105 W (un oscilloscope Tektronix 310), j'ai mesuré 280 W de puissance consommée. La stabilité et la propreté de l'alimentation sont à ce prix, sans parler du bruit et de la chaleur incessants !

Sur mon PE1033, à l'arrière du boîtier, on a négligemment monté une prise de style belge ou français (de l'époque) (**fig. 6**), connectée par vis aux bornes OUTPUT (L et N) et au châssis (GND). Les paires de bornes INPUT et OUTPUT sont à l'intérieur du boîtier, prévues pour une connexion permanente (**fig. 7**).

Le PE1033 peut exceller en stabilisation, filtrage et ondulation résiduelle, il ne dispose pas de l'isolation électrique complète du secteur tant convoitée. De plus, il n'a pas de plage d'ajustage. Pour cela et d'autres revendications, il faudra s'adresser ailleurs.

Transfo ajustable qui dispense du choc électrique avec le châssis TV

Sur notre table de travail, nous voulons au moins une sortie alimentée par une source alternative qui soit ajustable sur une large plage (comme les 10 à 260 V~ du variac) et une double isolation du secteur, donc avec un vrai transformateur dont primaire et secondaire sont assurés d'une bonne isolation l'un de l'autre (disons 2 kV). Avec fusible et calme.

Un appareil qui satisfait ces deux exigences et plus existe. Un ancien réparateur radio et TV m'en a donné un. Clairement, il tenait à sa santé et, dans les années 1970, il a convaincu son

patron de lui installer un Grundig RT 5A (**fig. 8**). C'était alors le nec plus ultra de la sécurité électrique et de la souplesse dans un seul instrument. Mais de nouveau, c'est lourd : 18 kg. L'Allemand Grundig était un grand constructeur de radios et de télévisions et c'est sans surprise que l'on voyait couramment les séries RT (de *Regel-Trenn*, réglage et séparation) de transformateurs chez les réparateurs radio/TV. Le mien est un modèle de 3,5 A (800 VA) avec lectures séparées sur voltmètre et ampèremètre à bobine mobile.

Les symboles sur la face avant du RT 5A (**fig. 9**) sont importants, ils indiquent le traditionnel sérieux allemand : la classe à double isolation avec fusible, bobinages totalement séparés, interrupteur. Sur internet, le reproche principal fait sur le RT 5A est sa tendance systématique à déclencher le disjoncteur à l'allumage. Les protections des installations actuelles en 230 V~ sont semi-automatiques rapides ou moyennes, elles répondent bien plus vite que les fusibles lents de 16 A en porcelaine des années 70. Mon RT 5A n'y fait pas exception. Même s'il y a un fusible lent de 10 A, j'envisage d'ajouter un frein au démarrage avec une grosse résistance d'arrêt (comme 10 Ω , 25 W) en série avec le primaire du transfo. Elle limitera le courant d'appel durant 500 ms avant d'être court-circuitée par un relais. Pas question de laisser sous tension le RT 5A en permanence en raison du bruit et de l'énorme dépense d'énergie.

Si vous ouvrez le RT 5A, l'électronique est assemblée impeccablement, mais la construction mécanique est trop fragile pour supporter un transfo de plus de 15 kg. Afin de limiter le poids de l'instrument, les concepteurs de Grundig ont lésiné sur la tôle d'acier utilisée, elle est beaucoup trop mince pour l'usage prévu.

Conclusion

Chacun de ces trois instrument a ses avantages et ses inconvénients, tout dépend de leur usage. Le variac est bon pour un usage à l'extérieur, vu qu'il est encore portable. Je trouve sa puissance disponible de 800 W convenable pour la plupart des appareils inconnus que je veux allumer en sécurité pendant environ six heures. Sa tension de sortie fixe à 115 V est pratique pour le matériel américain. Dommage qu'il n'ait pas d'ampèremètre pour indiquer un défaut avant la première fumée. Le PE1033 est un dinosaure, champion uniquement pour le filtrage de la tension du secteur. Sa prédestination réside dans la stabilisation de la tension à 1 % près, elle était primordiale à son époque, mais son utilité est aujourd'hui limitée.

Le RT 5A est l'instrument à privilégier, il offre une isolation électrique totale, on peut l'ajuster de 0 à 250 V~. En raison de la faiblesse de son boîtier, il convient moins en usage nomade, à l'inverse du variac. Pour s'en servir maintenant, il lui faut un démarrage progressif !

Par hasard, il se trouve que je dispose de ces trois appareils dans la collection de matériel rétronique, je les ai obtenus au prix de la ferraille et je peux les employer au gré de la nécessité. Aucun fusible grillé ! ◀

(180574-04 – version française : Robert Grignard)

EST^D 2004

www.elektor.tv



Rétronique est une rubrique mensuelle sur les pages glorieuses et jaunies de l'électronique, avec occasionnellement des montages de légende décrits dans Elektor.

Si vous avez des suggestions de sujets à traiter, merci de les télégraphier à redaction@elektor.fr



Figure 8. L'instrument RT 5A de Grundig Electronic des années 1970 combine un variac à un transformateur réglable, sortie isolée du potentiel du secteur. Important pour des dépannages sur les vieilles TV et les radios dont le châssis peut subir la tension de la phase (L). Remarquez que la prise de courant en face avant n'a pas de terre, ce qui est à la fois sage et souhaitable.



Figure 9. Qui aurait besoin d'un manuel d'utilisation ou d'un schéma si tout ce qu'il faut savoir est imprimé symboliquement sur la face avant du transformateur variable isolé RT 5A de Grundig ?

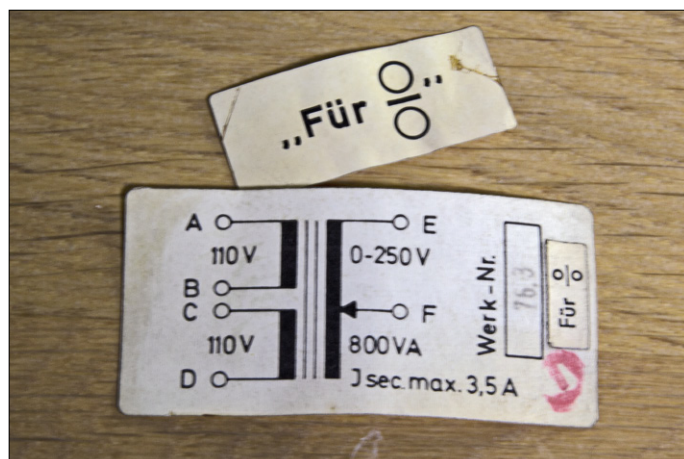


Figure 10. Deux petits autocollants tombés du RT 5A quand j'en ai ouvert le boîtier résumant avec pertinence la composition électrique de l'instrument.



système de fichiers interplanétaire

une nouvelle Toile mondiale à l'horizon...

Un robot astromobile de l'Agence spatiale américaine (NASA), Curiosity, sillonne actuellement le sol martien. Il fait partie de l'avant-garde de reconnaissance de la planète rouge, et on espère que ces éclaireurs seront rejoints par des humains dans un futur proche. Lorsque ces pionniers commenceront à s'installer, ils voudront eux aussi surfer sur la Toile et pouvoir consulter Wikipédia.

Source: NASA

Tessel Renzenbrink (Pays-Bas)

Cependant il y a un hic avec la structure du web actuel (*World Wide Web*, WWW), c'est la latence des communications entre Mars et la Terre. Le navigateur d'un ordinateur martien a besoin de 4 à 24 min pour établir la communication avec le serveur terrien de Wikipédia, temps qui dépend de la position des deux planètes sur leur orbite.

Le système de fichiers interplanétaire (*Interplanetary File System*, IPFS), un nouveau protocole de distribution de contenu, pourrait être la solution, et pas seulement pour une future situation hypothétique. Le problème de la latence se pose déjà aujourd'hui pour ceux qui sont loin des centres de données ou qui ont une liaison lente, comme c'est souvent le cas dans les zones rurales ou les pays en voie de développement.

L'IPFS n'est pas seulement plus rapide que le web, mais il permet aussi de résoudre d'autres problèmes comme la censure, les atteintes à la vie privée, et les attaques par déni de service (*Distributed Denial of Service*, DDoS). Cela semble trop beau pour être vrai, et il faut tout d'abord regarder de plus près les défauts du système actuel.

Les défaillances du World Wide Web

Le système actuel est très centralisé (**fig. 1a**). Dans un tel réseau, il y a un seul nœud central, et toutes les com-

munications entre ordinateurs doivent passer par ce nœud. On voit tout de suite où le bât blesse : si on supprime le nœud central, le réseau ne peut plus fonctionner !

Atteintes à la vie privée

En plus de cette vulnérabilité, un système centralisé conduit à un déséquilibre du rapport de force. Le nœud central règne en maître absolu sur les échanges entre membres : il peut refuser de transférer des données, de les lire, ou exiger un droit de passage. Ce pouvoir peut mener à des abus en matière de respect de la vie privée. Prenez par exemple Facebook ou Google, dont nous utilisons massivement les serveurs, ou YouTube, alimenté en vidéos par des millions d'internautes. Les géants de l'information se sont positionnés entre le contenu de la toile et les personnes qui veulent consulter ce contenu. L'idée derrière le WWW était bien de permettre l'échange de données et d'informations entre les gens, mais il faut désormais passer par des intermédiaires comme Google ou Facebook. Le prix à payer : la mise à disposition de nos données personnelles...

Censure

La centralisation du web résulte du modèle client-serveur, sur lequel il est basé (**fig. 2**). Si on veut par ex. visiter Wikipédia, il faut d'abord se connecter à un serveur sur lequel le site est hébergé. Si l'accès à ce serveur est bloqué, les informations ne sont plus accessibles aux

clients. C'est ce qui s'est passé en 2017 en Turquie, lorsque les autorités ont interdit l'accès à Wikipédia.

Attaques par déni de service

C'est aussi le modèle client-serveur qui permet les attaques par déni de service, où un serveur est inondé d'un très grand nombre de demandes d'information. Le serveur est alors saturé, ce qui empêche les utilisateurs avec des requêtes légitimes d'y avoir accès. La vulnérabilité vient à nouveau du fait que l'information n'est accessible que via un seul nœud central.

Les promesses de l'IPFS

Le système de fichiers interplanétaire semble très prometteur : il est rapide et résistant à la censure, et ce sont les utilisateurs qui contrôlent leurs données. Ceci vient du fait qu'il s'agit d'un réseau distribué (**fig. 1c**) et pas centralisé ; cette configuration est de loin la plus résiliente. Chaque ordinateur du réseau fonctionne comme un nœud par lequel l'information peut transiter, et si quelques-uns de ces ordinateurs sont supprimés, le réseau lui continue à fonctionner. Il n'y a pas de hiérarchie dans un réseau distribué, tous les nœuds sont égaux.

L'IPFS n'est pas non plus basé sur un modèle client-serveur pour l'échange de données ; les fichiers sont partagés en pair à pair (*peer-to-peer*). Chaque fichier peut donc être hébergé par n'importe quel ordinateur du réseau ; celui qui sou-

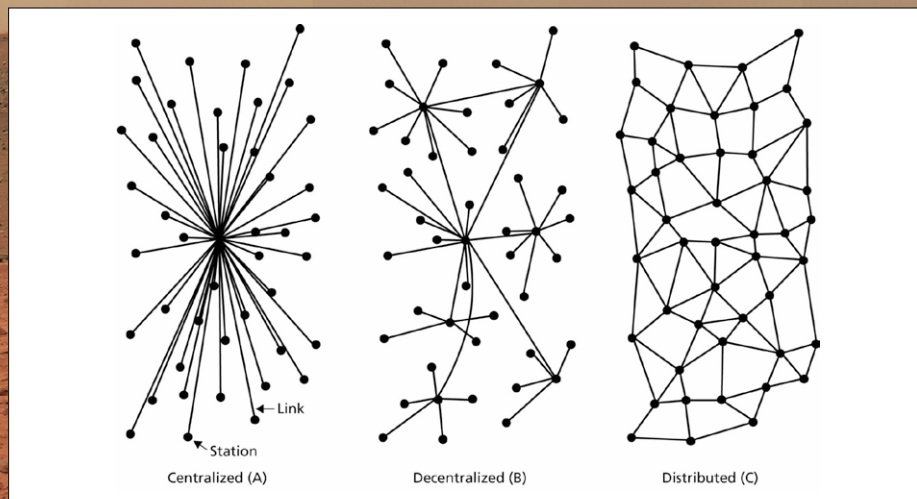


Figure 1. Ce schéma a été établi par l'électrotechnicien Paul Baran pour une étude sur les réseaux de communication résilients. Cette étude a été publiée en 1962 par la Rand Corporation sous le titre *On Distributed Communications Networks* (<https://bit.ly/2AFAqdp>).

haite télécharger un fichier demande une copie à l'ordinateur hôte le plus proche. L'ordinateur ayant téléchargé le fichier devient à son tour un hôte et peut aider à la distribution ultérieure du fichier (fig. 3).

La transformation invisible de la Toile

Ces propriétés et d'autres caractéristiques techniques de l'IPFS font qu'il ne souffre pas des vulnérabilités du WWW. Une attaque par déni de service ne peut pas fonctionner, parce qu'il n'y a pas un serveur central à corrompre, et c'est aussi ce qui fait sa force contre la censure. Lors de l'interdiction de Wikipédia en Turquie, une copie a circulé sur un réseau IPFS. Cette copie était hébergée par plusieurs ordinateurs, ce qui rendait beaucoup plus difficile d'en empêcher la diffusion.

Les utilisateurs contrôlent aussi leurs données. Si vous voulez partager une vidéo, vous la mettez à disposition sur le réseau, en l'hébergeant par ex. sur votre propre ordinateur qui la diffusera à la demande. L'information est donc disponible directement, et pas confinée sur des plates-formes comme YouTube ou Facebook.

Enfin, l'IPFS a la capacité de fonctionner hors ligne (*offline*) et localement, et ne doit donc pas être relié en permanence au réseau internet lui-même. Les futurs Martiens auraient avantage à utiliser l'IPFS. Le premier à vouloir consulter Wikipédia devra prendre son

mal en patience pour que la copie soit téléchargée d'un serveur terrien, mais il n'y aura plus de temps d'attente – ou presque – pour les suivants : Wikipédia sera disponible sur Mars.

L'aspect sans nul doute le plus impressionnant est que les concepteurs du système veulent le mettre en place sans que personne ne s'en aperçoive. Le protocole actuel HTTP est remplacé par l'IPFS de manière invisible pour les utilisateurs. Juan Benet, qui est à l'origine d'IPFS, en a parlé lors d'un séminaire à Stanford [1] : « Si personne ne s'est aperçu de la modification, alors nous avons gagné. L'interface n'a pas changé, car si vous essayez d'obliger les gens à changer leurs habitudes, ils n'utiliseront pas un nouveau système. C'est comme ça qu'il faut procéder pour améliorer l'internet : on n'y voit que du feu. *Cela ne peut qu'aller mieux !* ». ❏

(180572-04

version française : Jean-Louis Mehren)

IPFS est un projet à code source ouvert ; tout un chacun peut l'utiliser ou y contribuer. Pour de plus amples informations : <https://ipfs.io>.

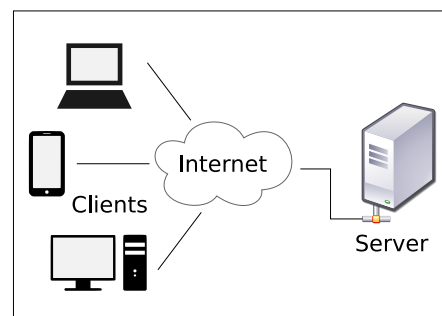


Figure 2. Le modèle client-serveur (David Vignoni : <https://bit.ly/2t7aPmM>).

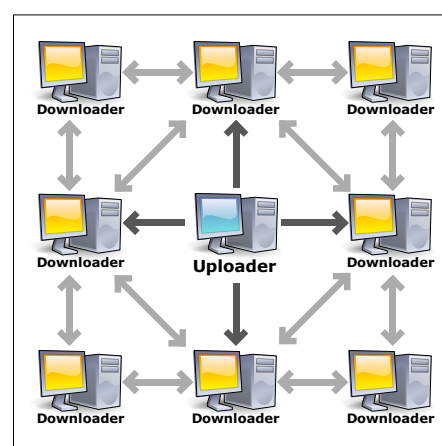


Figure 3. Réseau BitTorrent (Scott Martin : <https://bit.ly/2PCnuNu>).

Lien :

[1] Séminaire à Stanford, 'IPFS and the Permanent Web', Juan Benet : <https://bit.ly/2qncrtx>



bienvenue dans votre e-choppe

la rédaction recommande



Walabot Creator, un capteur 3D hors du commun

Le Walabot est un capteur 3D qui utilise l'émission/réception d'ondes radio hautes fréquences pour créer une image de l'environnement en face de lui. Les murs ou autres obstacles ne l'arrêtent pas, et ce n'est pas tout ! Ses concepteurs l'ont aussi doté d'un moniteur de respiration et d'une fonction de poursuite.

Le Walabot Creator a été examiné sous toutes les coutures. Lisez le banc d'essai qui lui a été consacré :

www.elektormagazine.fr/news/banc-d-essai-walabot-creator-un-capteur-3d-hors-du-commun

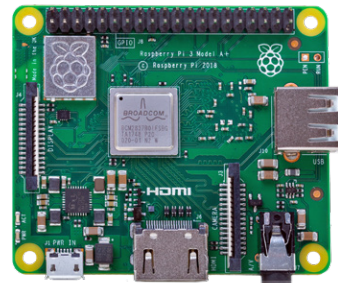
Thijs Beckers (rédacteur chez Elektor)

www.elektor.fr/walabot-creator-pack



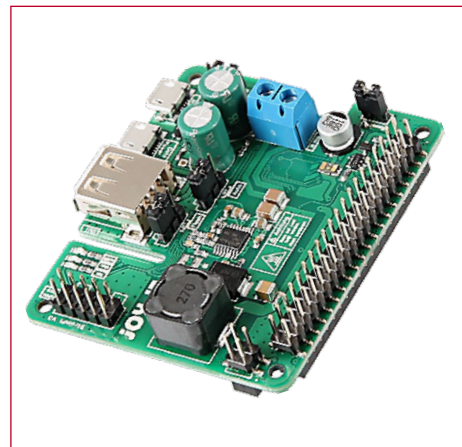
Vos favoris :

1. Raspberry Pi 3 (modèle A+) avec Compilation 2 RPI GRATUITE
www.elektor.fr/rpi-3a-plus



2. Walabot Creator Pack – capteur 3D programmable
www.elektor.fr/walabot
3. Testeur intelligent de CMS MS8911 (LCR-mètre) de Mastech
www.elektor.fr/ms8911
4. RasPad – tablette Raspberry Pi
www.elektor.fr/raspad#
5. Raspberry Pi 3 (modèle B+)
www.elektor.fr/raspberry-pi-3-model-b-plus
6. Kit de l'horloge à tubes VFD avec ESP32
www.elektor.fr/vfd-horloge

StromPi 3 – carte d'alimentation pour Raspberry Pi



Il vous envoie un courriel lorsque sa batterie de secours est vide ! Avec le StromPi 3, oubliez les pannes de courant inattendues ! Cette carte d'alimentation accepte des tensions d'entrée comprises entre 6 et 61 V et peut débiter jusqu'à 3 A (elle est donc assez costaud pour les projets gourmands en énergie). Vous pouvez également y brancher une batterie rechargeable qui servira de source d'alimentation de secours pour votre Raspberry Pi.



Prix (membres) : 38,66 €

www.elektor.fr/strompi-3

Poste de soudage Weller WE 1010 à commande numérique



avec kit d'apprentissage

Puissant poste de soudage à un canal de 70 W, avec fer à souder WEP 70 et support de sécurité PH 70. Également inclus : pince coupante diagonale de 170 mm ; fil d'étain WSW SCN M1 (100 g) ; pointes de fer à souder ETA 1,6 mm, ETB 2,4 mm



Prix (membres) : 188,10 €

www.elektor.fr/weller-we1010-edu

Kit PORSCHE Carrera Race Engine



Le moteur de course Carrera 547 a été le premier moteur Porsche spécialement conçu pour les courses. Avec les 300 pièces du kit Carrera Race Engine, vous pouvez assembler (sans collage) ce moteur à l'échelle 1:3. Ce moteur à quatre cylindres à plat totalement fonctionnel, de 1953, aura fière allure sur votre bureau.

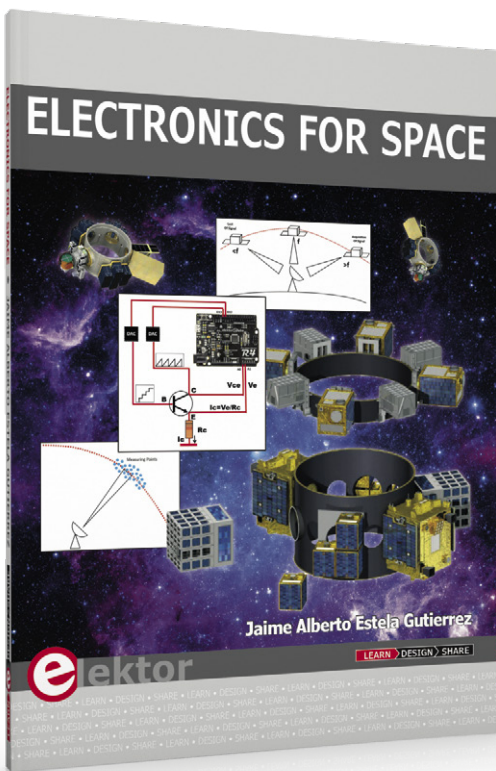


Prix (membres) : 206,10 €

www.elektor.fr/carrera-engine



nouveau



Electronics for Space

(livre en anglais)

Aujourd'hui, les voyages aux confins de l'espace ne sont plus un mythe, mais bien une réalité. L'industrie spatiale en croissance constante doit donc proposer de nouveaux produits et services. Le marché des petits satellites en particulier est en pleine expansion. Bien entendu l'innovation est au cœur de toute cette industrie.

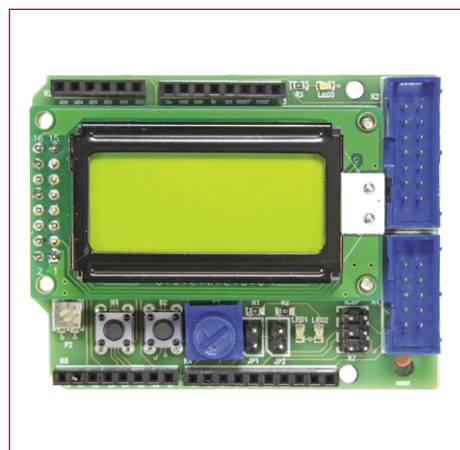
L'une des principales difficultés rencontrées par les bureaux d'études est de trouver des professionnels avec des connaissances théoriques et pratiques en conception électronique spatiale. Le but de ce livre est d'encourager les ingénieurs en électronique à entrer dans le domaine fascinant de l'électronique spatiale.



Prix (membres) : 22,46 €

www.elektor.fr/electronics-for-space

Shield d'expérimentation Arduino version 2.0



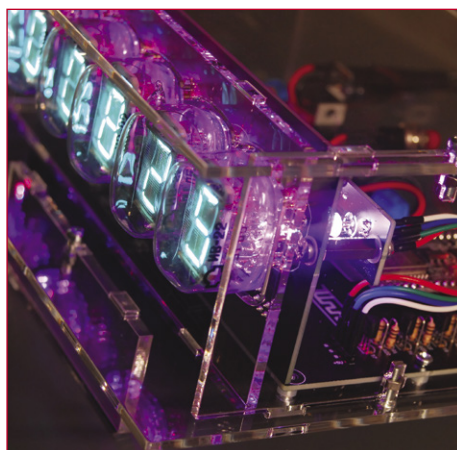
Le shield proposé en 2014 améliorait la carte Arduino avec des LED, des boutons-poussoirs, un écran LCD et un accès à l'interface UART et aux broches du microcontrôleur. Ce shield a suscité – et suscite encore – beaucoup d'intérêt ; une révision s'imposait donc ! La version 2.0 de ce shield permet maintenant de connecter n'importe quel écran LCD standard avec connexion à 2x8 trous.



Prix (membres) : 29,25 €

www.elektor.fr/160593-91

Horloge à tubes VFD et liaison Wi-Fi par ESP32



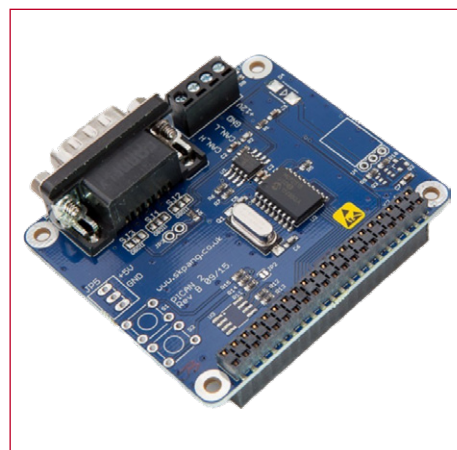
Cette horloge à l'allure vintage affiche l'heure et de la date sur six tubes russes, des VFD type IV-22 et deux indicateurs VFD DM160 (traits séparateurs entre heures, minutes et secondes). Pour sa mise à l'heure, elle se connecte par Wi-Fi à un serveur NTP de l'internet. La structure de l'horloge est 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

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 **21 janvier 2019** à l'adresse **hexadoku@elektor.fr**

Les gagnants

La solution de la grille du numéro de novembre/décembre 2018 est **386BF**.

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

Frederik **Andriof** (Allemagne), Henk **Brand** (Pays-Bas), Francis **Biette** (France), Freddy **Ghys** (Belgique) et Alexandr **Papazyan** (Russie).

Bravo à tous les participants et félicitations aux gagnants !

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

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

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.

Contrôlez absolument tout, depuis n'importe où, en gardant l'esprit tranquille

Créez des systèmes intelligents, connectés et sécurisés



www.microchip.com/SmartConnectedSecure

microchip
DIRECT
www.microchipdirect.com

Le nom et le logo de Microchip ainsi que les logos de Microchip sont des marques déposées de Microchip Technology Incorporated aux États-Unis et dans d'autres pays. Toutes les autres marques commerciales ci-dessus mentionnées sont la propriété exclusive de leurs propriétaires respectifs.
© 2018 Microchip Technology Inc. Tous droits réservés. DS00002758A. MEC2220Fre08/18


MICROCHIP

LES OUTILS INDISPENSABLES AUX ÉLECTRONICIENS

La WE1010 Education Bundle de Weller

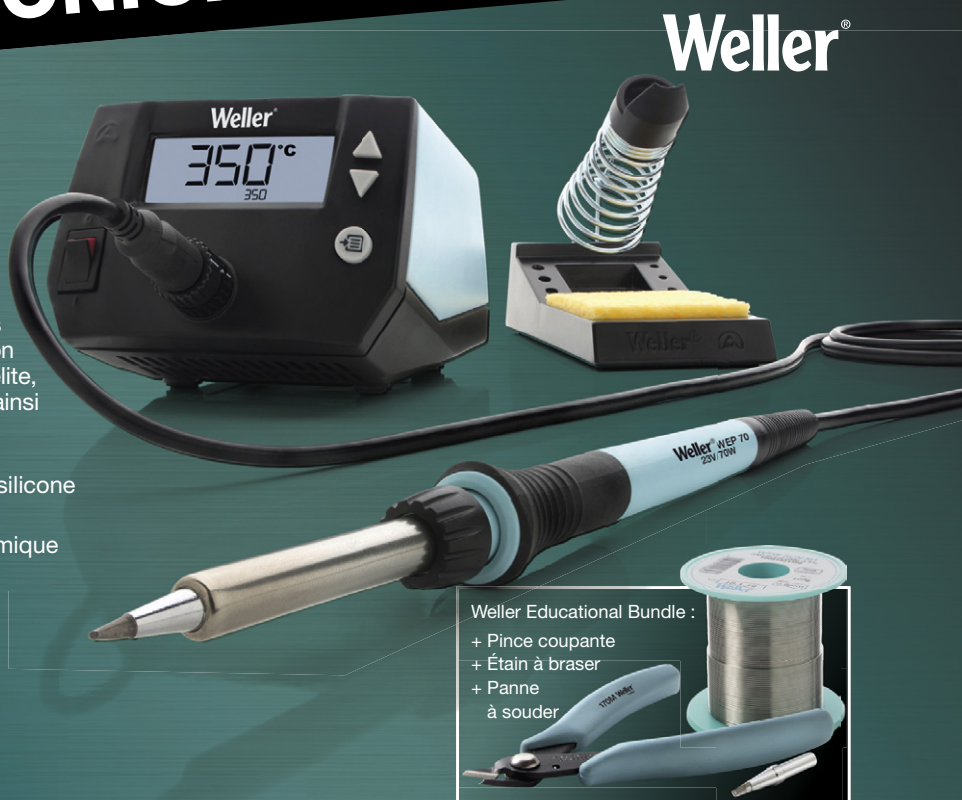
70 watts pour des exigences quasi professionnelles !

La station de soudage la plus puissante de sa catégorie ! Avec ce kit de formation, vous êtes parfaitement armé pour tous les exercices. En plus de la station de soudage WE1010, ce kit de formation comprend une pince coupante 170 mm Weller Xcelite, une bobine d'étain à braser Weller WSW SCN N1 ainsi qu'une panne à souder ETB 2,4 mm.

- Station de soudage ESD, fer à souder et câble silicone résistant à la chaleur
- Les températures stables et le verrouillage thermique protègent la panne à souder et le composant
- Une protection par mot de passe protège tous les réglages

Référence : WELLER WE1010EB

NOUVEAU **201,18** (€ 167,65)



Weller Educational Bundle :

- + Pince coupante
- + Étain à braser
- + Panne à souder

Multimètre numérique TRMS

XLABS

Avec éclairage LED intégré.

- Afficheur éclairé avec 6 000 points
- Mesure de la tension jusqu'à 1 000 V c.c./750 V c.a.
- Mesure de l'intensité jusqu'à 10 A c.a./c.c.
- Mesure de la fréquence, de la capacité et de la résistance
- Mesure du facteur de marche : 5-95 %



MULTIMÈTRE UNIVERSEL POUR TOUTE SORTE DE MESURES

EN 61010-1
CAT III 1000 V
EN 61010-1
CAT IV 600 V
EN 61010-1

Référence : VAL MS0205

MEILLEURE VENTE **40,31** (€ 33,59)

Jeu de tournevis 6 pièces

RND lab

Manche ergonomique pour une prise confortable remarquable.

- Lames en acier au chrome-molybdène vanadium
- Empreintes : 2 cruciformes : PH0/PH00 4 plates : 1,6/2,0/2,4/3,0 mm



Référence : RND 550-00197

NOUVEAU **15,12** (€ 12,60)

Lampe loupe à LED avec lentille en verre véritable

Eclairage sans ombre et agrandissement des détails.

- Lentille : Ø 100 mm, 3 dioptries
- Diffuseur annulaire
- Lumière : blanc froid (6 400 K)



A+
(A++ - E)

Référence : LL ZD-2012B-2H au lieu de 25,16

ECONOMISEZ 20 % **20,12** (€ 16,77)