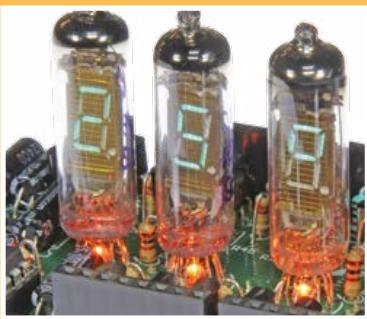




elektor

DÉCOUVRIR • CRÉER • PARTAGER



shield à VFD
pour Arduino Uno



**Wattmètre
CA/CC**



**USB déguisé en
pile AAA**

carte d'E/S pour Android

**L'électronique embarquée : pilotez-la
depuis votre téléphone portable**



microcontrôleurs ARM pour néophytes

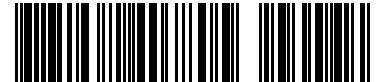
– 7^e partie : fonctions pour les experts

• **briques et briquettes TinkerForge** •

cours intensif d'assembleur – 2^e partie : mini-carte de développement et dé électronique • **mise à disposition des fichiers**

Gerber du labo • **e-BoB BL600** – 6^e partie : *Low Energy* – événements : *TIMER*, échange de données, connexion • **codes CMS** • **hors-circuits avec R. Lacoste** : comprendre l'amplificateur à transistor • **connecteurs coaxiaux**

Dans cette édition :
5 projets du labo
2 cours de programmation
4 nouveaux modules & PCB
... de quoi bien reprendre
après la pause estivale





THINK PROTOTYPE INNOVATE *

DESIGNSPARK

**Replacez l'innovation au centre
de votre processus de conception**

Notre suite d'outils pour le prototypage rapide a été conçue pour vous aider à passer du concept au prototype en un temps record.

Grâce à ces outils adaptés aux besoins des concepteurs de produits et ingénieurs en électronique, retrouvez le temps de vous consacrer à votre passion pour l'innovation et concevez ainsi les produits qui changeront le monde de demain.

Tout cela est rendu possible grâce à



Téléchargez gratuitement nos outils de prototypage rapide sur designspark.com



#RapidPrototyping

ISSN 0181-7450

Dépôt légal : août 2015

CPPAP 1113 U 83713

Directeur de la publication : Donatus Akkermans



Elektor est édité par :

PUBLITRONIC SARL

c/o Regus Roissy CDG

1, rue de la Haye

BP 12910

FR - 95731 Roissy CDG Cedex

@ : service@elektor.fr

Tél. : (+33) 01.49.19.26.19

du lundi au vendredi de 10h à 13h

Fax : (+33) 01.49.19.22.37

www.elektor.fr | www.elektormagazine.fr

Banque ABN AMRO : Paris

IBAN : FR76 1873 9000 0100 2007 9702 603

BIC : ABNAFRPP

Publicité :

Fabio Romagnoli +32 485 65 40 90

fabio.romagnoli@eimworld.com

DROITS D'AUTEUR :

© 2015 Elektor International Media B.V.

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

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

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

Héraldique japonaise

Cet été, avec mon neveu Jules (8 ans), nous visitons l'exposition « Samouraïs et chevaliers » au château de Malbrouck, près de Luxembourg. Les sociétés médiévales européennes et japonaises étaient différentes, mais présentent bien des similitudes : chevalier et samouraï respectent tous deux un code d'honneur (code de la chevalerie ou *bushido*, voie du guerrier), ils sont à cheval, le chevalier combat à la lance ou l'épée, le samouraï manie l'arc et le sabre...

Jules est fasciné par les couleurs et les dessins des blasons sur l'écu (bouclier) des chevaliers. Le guide explique que les blasons permettaient aux chevaliers de se reconnaître lors des batailles et des tournois. Dans les salles sur les samouraïs, nous découvrons leur *mon* ou *kamon*, dessins d'animaux ou de végétaux stylisés, utilisés comme une espèce de badge d'identification. Seuls les guerriers ou les courtisans peuvent porter des *mon*. Au milieu du XIX^e siècle, les samouraïs perdent ce privilège, l'empereur autorise l'utilisation du *mon* pour tous ; seul le chrysanthème impérial reste réservé à l'empereur. De nombreuses familles adoptent alors un *mon*, transmis ensuite de génération en génération, comme les armoiries en Europe.

Les règles de l'héraldique japonaise sont moins contraignantes que celles des blasons occidentaux, mais Jules a observé que souvent le *mon* s'inscrit dans un cercle. Bien vu, puisque aujourd'hui encore, les graphistes nippons, pour créer les logos de sociétés contemporaines, s'inspirent volontiers des *mon*. En 2011, Japan Airlines réintroduisait son logo circulaire, avec une grue japonaise aux ailes déployées de couleur rouge, ouvertement inspiré du blason d'une famille de samouraïs. Le fondateur de Mitsubishi a créé son logo en combinant les trois feuilles de chêne du clan Tosa (son premier employeur) et les trois diamants superposés de son *mon* familial : Mitsubishi signifie littéralement trois (*mitsu*) diamants (*ishi*).

Depuis notre visite de cette expo instructive*, je ne regarde plus les logos circulaires de la même manière. Ni celui des marques japonaises comme Toyota, Subaru, Nissan, Yamaha... à quelles familles de samouraïs ont-elles emprunté leur *mon* ?, ni celui d'Elektor sous la bannière duquel j'ai l'honneur et le plaisir de servir.

Mariline Thiebaut-Brodier

* L'exposition *Samouraïs et chevaliers* fermera ses portes le 1^{er} novembre 2015.

Notre équipe

Rédactrice en chef :

Mariline Thiebaut-Brodier (redaction@elektor.fr)

Rédaction internationale :

Harry Baggen, Jan Buiting, Jaime González-Arintero,
Jens Nickel

Laboratoire :

Thijs Beckers, Ton Giesberts, Luc Lemmens,
Clemens Valens (responsable), Jan Visser

Coordination :

Hedwig Hennekens

Ont coopéré à ce numéro :

Robert Grignard, Hervé Moreau, Kévin Petit,
Guy Raedersdorf, Thierry Destinobles,
Jean-Louis Mehren, Denis Meyer

Service de la clientèle :

Cindy Tijssen

Graphistes :

Giel Dols, Mart Schroijen, Patrick Wielders

Elektor en ligne :

Daniëlle Mertens

- 8 Elektor : guide de connexion(s)
- 74 des nouvelles du monde d'Elektor
- 76 l'e-choppe d'Elektor
- 82 hexadoku

DÉCOUVRIR | **CRÉER** | **PARTAGER**

- 6 bienvenue dans la section DÉCOUVRIR
- 7 connecteurs coaxiaux drôle de composant n°18
- 10 µC ARM pour neophytes pour passer de 8 bits à 32 bits - 7^e partie : fonctions pour les experts
- 16 cours intensif d'assembleur (2) mini-carte de développement et délectronique
- 24 hors-circuits avec R. Lacoste : comprendre l'amplificateur à transistor sexagénaire – vaillant & toujours indispensable

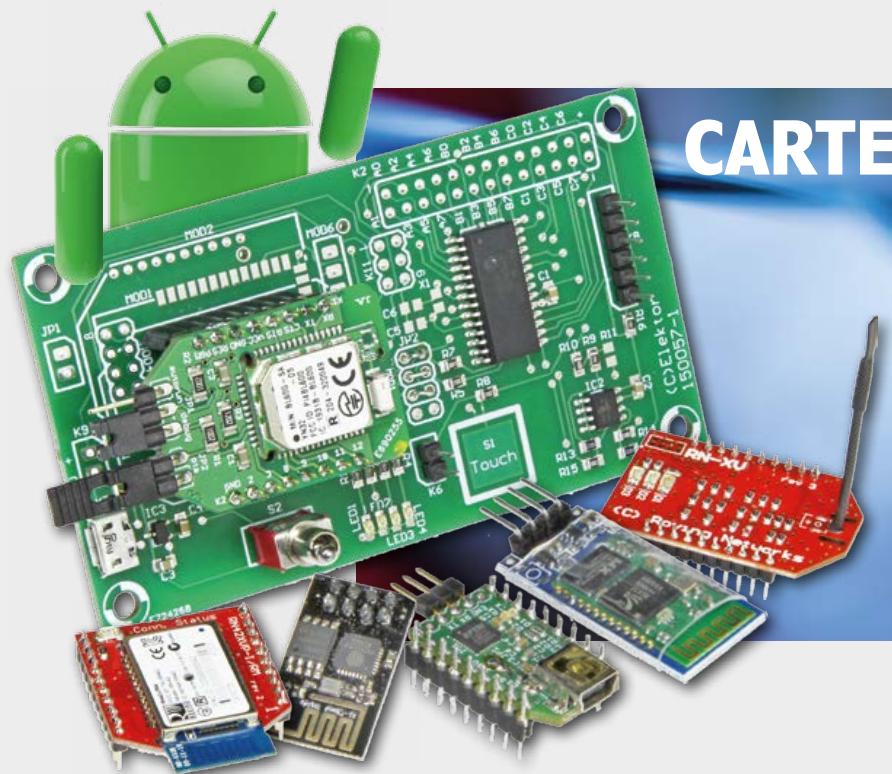
DÉCOUVRIR | **CRÉER** | **PARTAGER**

- 29 bienvenue dans la section CRÉER
- 32 carte d'E/S pour Android l'électronique embarquée, pilotez-la depuis votre téléphone portable
- 40 USB déguisé en pile pour remplacer une pile AAA
- 42 wattmètre CA/CC haute précision, gammes étendues
- 50 e-Bob BL600 Module Bluetooth Low Energy 6^e partie – Low Energy – événements : TIMER, échange de données, connexion
- 56 shield à VFD pour Arduino Uno créez une horloge, un voltmètre, un thermomètre

SHIELD À VFD POUR ARDUINO UNO

CRÉEZ UNE HORLOGE, UN VOLTMÈTRE, UN THERMOMÈTRE...

Oui, vous connaissez le VFD ! *Vacuum Fluorescent Display*, c'est l'afficheur fluorescent de votre four à micro-ondes, par exemple. Ce n'est pas un Nixie, mais ce type de tube lui ressemble et celui décrit ici est russe. En fait, il y en a quatre sur un *shield* pour carte Arduino Uno, ce qui permet de construire différents appareils. Les logiciels pour horloge, voltmètre et thermomètre sont déjà rédigés et il y a, en prime, un programme de démonstration.



42

WATTMÈTRE CA/CC HAUTE PRÉCISION, GAMMES ÉTENDUES

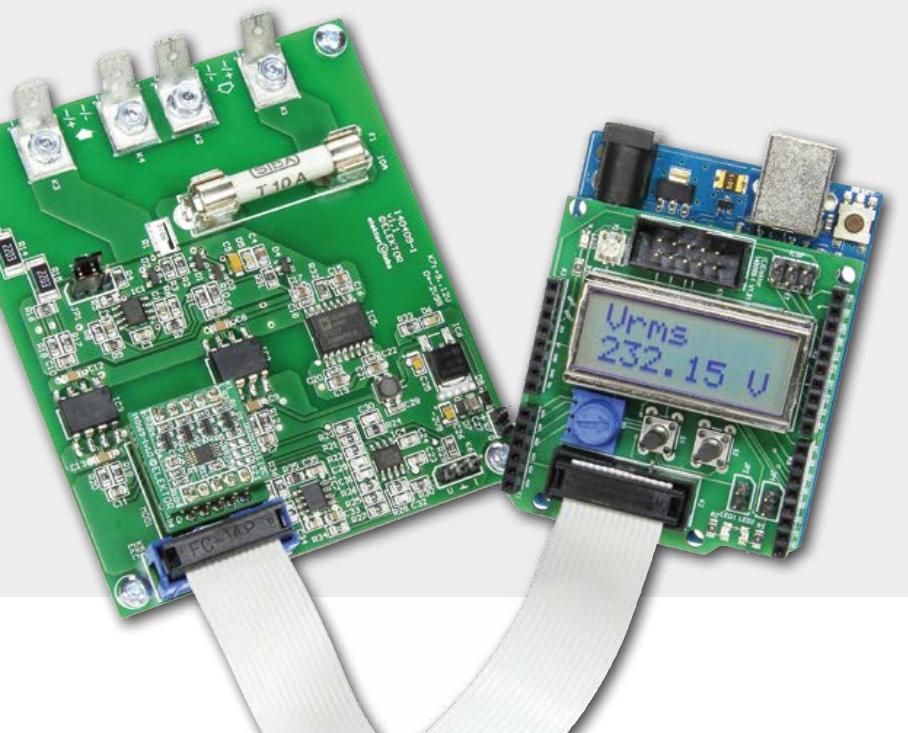
Entreprendre des mesures sur le secteur suppose une sécurité absolue. Tension, courant ou puissance, aucune différence à cet égard, exigez toujours une séparation galvanique entre le circuit sous moyenne tension et l'appareillage à manipuler. Ce projet assure une isolation de classe II pour mettre à l'abri votre Arduino Uno, son *shield* d'expérimentation Elektor et... votre personne. Dans la foulée, vous pourrez sans souci brancher sur les sorties appropriées multimètre et oscilloscope.

56



D'E/S POUR ANDROID L'ÉLECTRONIQUE EMBARQUÉE, PILOTEZ-LA DE VOTRE TÉLÉPHONE PORTABLE

32



DÉCOUVRIR CRÉER PARTAGER

- 61 bienvenue dans la section PARTAGER
- 62 briques et briquettes TinkerForge jettent un pavé dans la marre du prototypage rapide
- 66 électronique corrections, mises à jour et compléments d'articles
- 68 .LABorama véritablement (f)estival
- 70 sélection de fichiers Gerber du labo fichiers des circuits imprimés Elektor mis à disposition des lecteurs
- 72 codes CMS à quel composant avons-nous affaire ?

► bientôt sur ces pages

Système de mesure audio MEASSY

Cet appareil compact réunit dans le même boîtier un excellent préamplificateur de microphone, avec alimentation fantôme, et un petit ampli de puissance audio, le tout pour effectuer des mesures sur des ensembles de haut-parleurs.

Amplificateur audio compact de 60 W

Cet amplificateur de puissance est l'œuvre de composants exclusivement discrets et aisément disponibles. Le circuit est d'une construction aisée, n'occupe que peu de place sur le circuit imprimé, ce qui ne l'empêche pas d'offrir une excellente qualité sonore.

Red Pitaya se décline en radio FM stéréo

Nouvelle démonstration de l'universalité du système de développement Red Pitaya. Ajoutez-lui un petit préamplificateur et un traitement numérique efficace, vous en ferez un décodeur des émissions FM stéréo, RDS compris.

Sous réserve de modification.

Le numéro d'octobre paraîtra le 23 septembre.

bienvenue dans la section DÉCOUVRIR



Jens Nickel

De retour de la Maker Faire de Hanovre

Lorsque vous lirez ces lignes, la *Maker Faire* de Hanovre sera déjà terminée. Je profite tout de même de l'occasion pour remercier tous les collègues de *Make!*. Ce fut vraiment un événement exceptionnel, vous aviez même prévu le soleil ! L'engouement des jeunes électroniciens faisait particulièrement plaisir à voir.

Les huit stations de soudage de notre atelier (photo) n'ont pu refroidir qu'après que le dernier « dé électronique » ait été monté. Nous avons également pu découvrir de nouvelles cartes mère et de nouveaux projets. Les modules matériels de *Brick 'R' Knowledge* (www.brick-rknowledge.de) m'ont particulièrement plu, car ils se prêtent à la formation (mais pas uniquement). La « SenseBox » est aussi un projet passionnant. Ce kit de capteurs (IoT) comprend une plate-forme Internet pour l'échange de données

collectées comme la pression atmosphérique, la température, l'humidité de l'air et le rayonnement lumineux. Il y a actuellement deux versions du kit, une pour la formation et une pour les amateurs.



Apprendre à programmer (2)

J'ai reçu de nombreux courriels après l'édition « Apprendre à programmer », merci beaucoup ! Un lecteur a par exemple attiré mon attention sur le *Project Oberon*. Je n'avais encore jamais entendu parler d'une telle idée : du matériel basé sur des FPGA (voir www.xilinx.com/publications/archives/xcell/Xcell91.pdf, p. 30). On a également eu la bonne idée de me recommander d'utiliser un

launchpad (littéralement une rampe de lancement) de TI. Un autre lecteur recherche (comme moi) une carte programmable en C# afin de n'avoir recours qu'à un seul et même langage de programmation pour le PC et les systèmes embarqués. Peut-être que nous aurons prochainement une Raspberry Pi 2 qui tourne sous Windows. Je m'en réjouis d'avance.

Il y avait quand même quelque chose

Voilà un nouvel épisode de la série « *mon étourderie* a encore frappé, le matériel ne fonctionne pas comme je le voudrais ». J'ai enfin essayé de porter l'analyseur MIDI de notre dernière édition de la carte Arduino Uno sur notre carte à serveur web Xmega. J'ai adapté mon configurateur EFL (*Embedded Firmware Library*) pour qu'il crée d'un simple clic un joli projet Xmega avec tous les fichiers nécessaires. Le code source indépendant de la plate-forme de l'analyseur MIDI était déjà intégré. Il me fallait encore modifier dans le code le numéro de l'interface UART à laquelle

est connecté le module MIDI. La compilation n'a pas produit d'erreur, mais qu'est-ce que c'est que ça ? La LED d'alimentation sur le module MIDI est restée éteinte quand j'ai connecté la carte Xmega. J'ai testé avec une deuxième carte, ça a fonctionné sans problème. En regardant le schéma, j'ai vite compris mon erreur : il manquait un cavalier sur la première carte, ce qui m'avait échappé. La leçon de l'histoire est qu'il faut toujours consulter la documentation avant d'utiliser le matériel, même si l'on pense le connaître sur le bout des doigts. ►

(150319 – version française : Thierry Destinobles)

connecteurs coaxiaux

drôle de composant n°18

Neil Gruending (Canada)

Si les connecteurs coaxiaux sont idéaux pour les radiofréquences, ce n'est pas par hasard : la broche qui transporte le signal est entourée par la masse, cela permet de contrôler l'impédance du connecteur, cruciale en RF. On trouve beaucoup de modèles ; voyons les plus courants et leurs utilisations.

Contre toute attente, les connecteurs les plus familiers ont longtemps été employés dans les applications A/V. Les prises RCA (**fig. 1**) ont une bande passante d'environ 10 MHz, plus que suffisante pour transmettre les signaux audio et vidéo des *home cinémas*.

Les connecteurs filetés F (**fig. 2**) utilisés pour la télévision et l'internet par le câble possèdent une bande passante bien plus grande : jusqu'à 1 GHz dans certains cas, parfait pour les signaux à haute fréquence tels que ceux en sortie d'une tête de réception TV par satellite. Les connecteurs coaxiaux ne sont toutefois pas cantonnés à votre salon.

L'un des premiers connecteurs coaxiaux a été conçu dans les années 30 : le modèle UHF ou SO-239 (Amphenol). Assez robuste, il est capable de supporter de fortes puissances, c'est le connecteur idéal pour les antennes de cibistes et radioamateurs. Sa limite haute en fréquence se situe à env. 200 MHz, à cause de son impédance non-linéaire au-delà de ce seuil. Avec le recul, le U dans sa désignation devrait plutôt signifier « usurpé » : on considère que la bande UHF commence à 300 MHz.

Pour les fréquences plus élevées, il y a le connecteur BNC (Bayonet Neill-Concelman) (**fig. 3**) doté de deux baïonnettes pour un verrouillage en un quart de tour.

Conçu à l'origine pour l'armée, il est aujourd'hui largement utilisé dans beaucoup d'applications avec une bande passante jusqu'à 2 GHz. Les électroniciens le rencontrent sur les appareils de mesure tels que les générateurs de signaux et oscilloscopes.

Il existe également une version filetée des connecteurs BNC : TNC (Threaded Neill-Concelman). Le filetage permet d'éviter les encoches de verrouillage qui restreignent la gamme de fréquences des BNC ; la limite supérieure pour les TNC est d'environ 11 GHz. Ils sont généralement utilisés pour les antennes. BNC et TNC sont disponibles en deux impédances : 50 et 75 Ω. Même si leurs dimensions diffèrent, attention à ne pas les confondre. Les modèles 75 Ω ont un diélectrique plus fin que les 50 Ω, mais ce n'est pas garanti !

Une variante plus courante du BNC à filetage est le connecteur N (**fig. 4**) inventé en 1940, par Paul Neill des Laboratoires Bell. Beaucoup plus gros qu'un BNC, il est capable de transporter beaucoup plus de puissance. Il en existe également deux versions, 50 et 75 Ω, mais il sera impossible de les confondre : un connecteur 50 Ω endommagerait un socle 75 Ω.

Un autre modèle courant est le connecteur SMA (**fig. 5**) inventé dans les années 60. Il est bien plus maigre qu'un BNC, cela lui permet de supporter des fréquences jusqu'à 18 GHz. Il n'est en aucun cas conçu pour des puissances élevées ou un grand nombre de connexions/déconnexions, mais ce n'est nullement un problème pour les installations fixes. Nous n'avons passé en revue qu'un petit échantillon des nombreux types de connecteurs coaxiaux disponibles. Les modèles classiques présentés ici sont toujours largement utilisés ; une preuve de leur bonne conception et de leur durabilité. ↗

(150341 – version française : Kévin Petit)



1



2



3



4



5



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

Elektor : votre guide

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

57

pays

246817

membres actifs

10

experts &



elektor.post

L'hebdo d'Elektor

Les signaux du changement fusent autour de nous. Avec son rythme hebdomadaire, la lettre électronique elektor.post permet de les suivre à une allure soutenue. Une semaine sur deux, un montage inédit !

www.elektor.com/newsletter



communauté Elektor

Devenez membre, Green ou Gold

Pro ou débutant, rejoignez la communauté. C'est le plus sûr moyen de ne rien rater, ni en électronique classique, ni en techniques embarquées modernes. Vos atouts : accès direct à elektor.labs, forums, lettres d'information hebdomadaires, projets inédits bimensuels, offres exceptionnelles, archives, moteurs de recherche. Les formules Green et Gold donnent droit à de nombreux avantages : **GREEN**, c'est le magazine sous forme numérique, sans papier. **GOLD**, c'est la formule complète avec la version imprimée.

www.elektor.com/memberships



elektor.TV

Pour y voir plus clair

L'image vidéo filmée sans façon est devenue un rival stimulant pour le texte typographié et mis en page ! Les anecdotes visuelles ne manquent pas dans la vie d'un labo d'électronique, surtout quand ça commence à fumer. Et souvent trois plans filmés remplacent efficacement de longs discours. Regardez elektor.tv !

www.youtube.com/user/ElektorIM



Elektor PCB Service

Des cartes à la carte

Adieu perchlorure de fer, bienvenue aux magnifiques circuits imprimés du service de gravure à la demande d'Elektor. Très professionnel et très rapide, très avantageux et tellement plus commode !

www.elektorpbservice.com



elektor.labs

découvrir, créer & partager

Au cœur de la matrice, elektor.labs (le labo d'Elektor) est ouvert et accessible à tous, c'est l'incubateur où éclosent les circuits. Petits et grands, analogiques ou numériques, d'avant-garde ou nostalgiques, ils y sont tous transformés en matière première raffinée, prête à l'emploi, testée et documentée pour vos propres créations.

www.elektor-labs.com



elektor.academy

À cheval sur la courbe d'apprentissage

Webinaires, séminaires, cours, présentations, ateliers, lectures, formation en entreprise sont quelques-unes des méthodes pédagogiques utilisées par Elektor pour diffuser la connaissance de l'électronique à tous les niveaux aussi bien parmi les professionnels passionnés que pour les amateurs motivés.

www.elektor-academy.com

de connexion(s)

29

477

233521

10:33

AUGUST 04 2015

auteurs

publications

visiteurs (mois)

date de référence



elektor.magazine

Plus de 500 pages d'électronique inédite chaque année

Le magazine est le vaisseau amiral, affrété tous les mois par la rédaction internationale d'Elektor pour vous embarquer vers des contrées électroniques nouvelles. Chaque édition, sur papier ou en format numérique, ne se contente pas de rester à la hauteur des précédentes, mais cherche à les surpasser.

www.elektormagazine.fr



e-choppe Elektor en ligne

Votre panier d'achats pour l'électronique

Le magazine et le labo d'Elektor proposent, en coopération avec des partenaires choisis, des produits et des services de haut niveau. Notre e-choppe, véritable caverne d'Ali Baba, est ouverte toute l'année sans interruption pour les électroniciens du monde entier.

www.elektor.fr

3 formules pour rester connecté avec Elektor !



livres et DVD Elektor

La puissance de l'information

Elektor aborde tous les domaines de l'électronique : de la programmation des 8 bits aux ARM, des antennes aux diodes zener, des µC aux tubes... Nos ouvrages font référence et autorité dans le monde entier, aussi bien pour les techniques classiques que pour les innovations les plus récentes. Lire pour (meilleur) comprendre.

www.elektor.fr

Formule GREEN

92,50 € par an

- ✗ 10 x magazine imprimé
- ✓ 10 x magazine numérique
- ✓ l'accès à l'archive d'Elektor
- ✓ 10 % de remise dans l'e-choppe
- ✓ l'accès à Elektor.labs
- ✓ un DVD annuel
- ✓ des offres exclusives
- ✓ 26 nouveaux projets inédits

www.elektor.fr/formule-greencard

Formule GOLD

127,50 € par an

- ✓ 10 x magazine imprimé
- ✓ 10 x magazine numérique
- ✓ l'accès à l'archive d'Elektor
- ✓ 10 % de remise dans l'e-choppe
- ✓ l'accès à Elektor.labs
- ✓ un DVD annuel
- ✓ des offres exclusives
- ✓ 26 nouveaux projets inédits

www.elektor.fr/formule-goldcard

Formule gratuite !

- ✗ 10 x magazine imprimé
- ✗ 10 x magazine numérique
- ✗ l'accès à l'archive d'Elektor
- ✗ 10 % de remise dans l'e-choppe
- ✗ l'accès à Elektor.labs
- ✗ un DVD annuel
- ✓ des offres exclusives
- ✓ 26 nouveaux projets inédits

www.elektor.fr/inscription

fonctions pour les experts

μC ARM pour néophytes pour passer de 8 bits à 32 bits - 7^e partie

Activation du détecteur de baisse de tension

À l'image des microcontrôleurs (μC) 8 bits, les μC d'ARM ont eux aussi un BOD (**Brown Out Detector** = détecteur de baisse de tension) ; il déclenche une interruption ou réinitialise la MCU lorsque la tension tombe en deçà d'une valeur prédéfinie. Comme ce seuil de tension se situe en général en dessous de la tension d'alimentation recommandée du μC, le BOD remplit une fonction de protection ; en cas de chute de la tension d'alimentation par exemple, il arrête le μC bien avant tout désastre tel qu'une perte de données.

Dans le cas du SAM D20, digne champion du Cortex-M0+ d'ARM, c'est aussi le cas. Il en a même deux, vu que le μC utilise plusieurs tensions d'alimentation en interne. BOD12 est configuré en usine. MCU en fonctionnement, il est toujours fonctionnel et surveille la tension d'alimentation de 1,2 V de la CPU fournie par le régulateur de tension interne. Plus intéressant, le BOD33 ; il suit la tension d'alimentation de 3,3 V sur VDDANA. Ce BOD peut soit, en *Continuous Mode*, surveiller la tension en continu, soit encore, en *Sample Mode*, effectuer une mesure unique sur tout flanc d'horloge spécifié pour

se rendormir ensuite. En interne, un prédiviseur (*prescaler*) peut diviser l'horloge de 1 kHz de source interne par une valeur au maximum de 65536, en fonction du paramétrage. Il existe en outre une hystérésis réglable, comprise entre 35 mV et 170 mV en général.

Passons au programme correspondant « *Test of the BOD-module* » qui configure correctement le BOD33 et le paramètre. Dans le **listage 1**, il y a d'abord création d'une structure de configuration (*config_bod*) classique, préparamétrée ensuite. Certains paramètres sont modifiés dans les variables de structure, la structure transmise au BOD33 et enfin le détecteur de baisse de tension activé. Voici la définition de la fonction : mode continu, la réaction lorsque la tension de seuil est atteinte est une réinitialisation (*Reset*), la tension de seuil est 0x27 et l'hystérésis avec les tensions de seuil V_{BOD} et V_{BOD+} est activée (**tableau 1**). Si la fonction d'hystérésis n'est pas utilisée, les valeurs V_{BOD} de la 2^e partie du tableau (**fig. 1**) s'appliquent.

On trouve dans la colonne *niveau BOD33* les valeurs de configuration possibles pour les seuils de tension. Ici,

Listage 1. La fonction élémentaire de configuration du BOD33

```
void configure_bod(void)
{
    struct bod_config config_bod;
    bod_get_config_defaults(&config_bod);

    config_bod.mode          = BOD_MODE_CONTINUOUS;
    //config_bod.mode          = BOD_ACTION_NONE;
    config_bod.action        = BOD_ACTION_RESET;
    config_bod.level         = 0x27;
    config_bod.hysteresis   = true;

    bod_set_config(BOD_BOD33, &config_bod);
    bod_enable(BOD_BOD33);
}
```

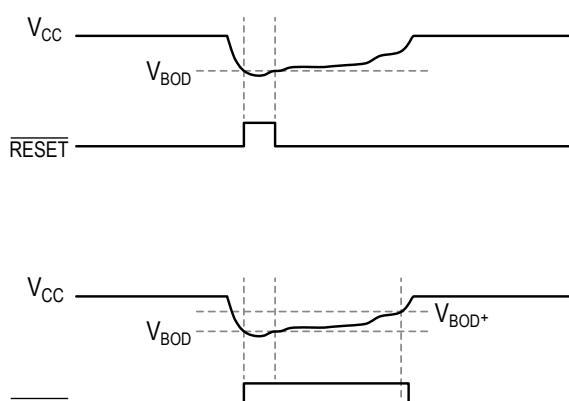


Figure 1. C'est ainsi que travaille le BOD – en haut, sans hystérésis ; en bas, avec (diagrammes et copies d'écran : Atmel).

Viacheslav Gromov (Allemagne)

les précédents, nous avons parlé de tous les éléments périphériques et des bibliothèques ASF correspondantes. Nous abordons ici des fonctions Les projets proposés sont relativement indépendants, vous pouvez que vous voulez, ou non, lire et essayer. Cette avant-dernière partie sur quelques trucs et astuces. Nous vous offrons en ligne un projet ire sur l'Event-System (voir www.elektormagazine.fr/1503).

nous utilisons 0x27 (39 en décimal), qui correspond à une V_{BOD+} typique de 2,84 V et une V_{BOD-} typique de 2,77 V. La modification de cette valeur dans le programme permet donc de jouer sur les tensions de seuil.

Dans la fonction *Main*, tout ce que l'on fait, après la fonction *system_init*, c'est appeler la fonction de configuration du BOD dont nous venons de parler. La boucle sans fin est vide. Vous pouvez insérer ici votre code ; le BOD33 devenu actif procède à une réinitialisation lorsque la tension d'alimentation tombe en deçà du seuil paramétré.

Vous pouvez modifier les configurations du BOD en fonction de votre projet. Si dans la fonction de configuration, vous remplacez l'instruction

```
config_bod.action      = BOD_ACTION_RESET;
```

par l'instruction (commentée dans le code)

```
config_bod.action      = BOD_ACTION_NONE;
```

alors l'instruction `bod_is_detected(BOD_BOD33)`, qui délivre la valeur true si le BOD33 a détecté jusqu'à cet instant une chute en dessous du seuil de tension, vous permet d'interroger le BOD « en manuel » et d'exécuter les actions voulues en cas de tension insuffisante. Il vous faudra alors, par `bod_clear_detected(BOD_BOD33)`, réinitialiser cet indicateur de détection avant le prochain appel de l'instruction.

Selected Modules

- ▷  Generic board support (driver)
 - ▷  BOD - Brown Out Detector (driver)
 - ▷  PORT - GPIO Pin Control (driver)
 - ▷  SYSTEM - Core System Driver (driver)

Figure 2. Seule la bibliothèque BOD a été incluse dans l'Assistant ASE

Vous pouvez maintenant transférer le projet vers la carte. Vous n'allez rien voir cependant ; en effet pour voir le BOD en action, il faut que la tension soit inférieure au seuil défini et que vous ayez mis du code dans la boucle sans fin. Pour ce programme, seule la bibliothèque BOD a été incluse dans l'Assistant ASF (fig 2).

Pour plus d'informations sur le BOD33, cf. feuille de caractéristiques du SAM D20, p. 143 sqq. et sur <http://goo.gl/yVtRvK>.

Tableau 1. Les seuils de tension du BOD33 avec et sans hystérésis (source : Atmel).

Tableau 2: Les seuils de tension du BOD33 avec une hystérésis (source : STMicroelectronics).					
symbole	niveau BOD33	condition	min.	typ.	max.
V_{BOD+}	6	hystérésis ON	--	1,715 V	1,745 V
	7		--	1,75 V	1,779 V
	39		--	2,84 V	2,92 V
	48		--	3,2 V	3,3 V
V_{BOD-} OU V_{BOD}	6	hystérésis ON ou hystérésis OFF	1,62 V	1,64 V	1,67 V
	7		1,64 V	1,675 V	1,71 V
	39		2,72 V	2,77 V	2,81 V
	48		3,0 V	3,07 V	3,2 V

Fonctions de sommeil et de réinitialisation du SAM D20

Dans la bibliothèque de pilotes du système de tout nouveau projet, on trouve aussi des instructions pour les fonctions de sommeil et de réinitialisation du SAM D20. Dans cette bibliothèque, vous connaissez déjà l'instruction `system_init()` qui initialise l'ensemble de la MCU par exemple. D'autres instructions importantes concernent principalement les fonctions de sommeil et de réinitialisation du SAM D20 :

- Lors de son appel, la fonction `system_reset()` réinitialise toute la MCU. Seuls les éléments périphériques ne pouvant/ne devant pas être réinitialisés (comme la RTC) sont épargnés.
- En cas d'appel, la fonction `system_get_reset_cause()` fournit une constante symbolique (valeur) identifiant la source de la dernière réinitialisation (par le logiciel ou par le BOD, par exemple).
- `System_set_sleepmode()` définit le mode de sommeil pour le prochain appel de l'instruction `system_sleep()` ce qui permet, selon le mode, de réduire un peu la consommation. Le seul paramètre de cette instruction est la constante symbolique pour le mode de sommeil voulu

(`SYSTEM_SLEEP_MODE_IDLE0` par ex.). Le tableau 1 récapitule les éléments périphériques fonctionnant encore/ne fonctionnant plus selon les modes de sommeil.

- La fonction `system_voltage_reference_enable()` ne requiert, comme paramètres, que la constante symbolique pour la source de tension de référence (`SYSTEM_VOLTAGE_REFERENCE_TEMPSENSE` par ex. pour la référence de tension de la sonde de température interne), qu'active cette instruction. On a, pour désactiver une référence de tension, l'instruction `system_voltage_reference_disable()`, avec le même paramètre.
- L'instruction `system_get_device_id()` renvoie un entier à 32 bits. Cette valeur représente l'ID unique de la MCU, qui différencie des MCU de même type. La plupart des MCU de la famille Cortex-M0+ d'ARM en ont un.

Pour plus d'informations, voir le document www.atmel.com/Images/Atmel-42120-SAM-D20-System-Driver-SYSTEM_Application-Note_AT03260.pdf.

Tableau 1. Aperçu des éléments périphériques et système (dés)activés dans les différents modes de sommeil
(source : Atmel).

mode sommeil	horloge CPU	horloge AHB	horloge APB	sources horloge	horloge système	32 kHz	mode régulateur	mode RAM
IDLE 0	à l'arrêt	activé	activé	activé	activé	activé	normal	normal
IDLE 1	à l'arrêt	à l'arrêt	activé	activé	activé	activé	normal	normal
IDLE 2	à l'arrêt	à l'arrêt	à l'arrêt	activé	activé	activé	normal	normal
STANDBY	à l'arrêt	à l'arrêt	à l'arrêt	à l'arrêt	à l'arrêt	à l'arrêt	basse conso.	polaris. source/ drain

Trucs & astuces pour le SAM D20

Consommation de courant des périphériques

S'il y a longtemps que vous n'avez pas lu la feuille de caractéristiques du SAM D20, il faudrait le faire maintenant ! En janvier 2015, sa « grande » fiche technique a été mise à jour ; certaines informations y sont clarifiées, de nouvelles ajoutées (sous forme de tableaux par ex.). D'où une nouvelle pagination. En outre, ça vaut la peine (vu que vous connaissez tous les éléments périphériques maintenant) de lire le chapitre 32, page 571 (p. 570 auparavant) qui décrit les caractéristiques électriques de la MCU et des différents éléments. Vous y lirez, entre autres, les tensions et courants d'alimentation admissibles. Il faudra jeter un coup d'œil au tableau 32-8, qui donne la consommation de courant de la MCU dans les différents modes de sommeil, et le tableau 32-10 celle des différents périphériques. Pour les projets basse consommation (Low Power), ces informations sont capitales ; on pourra voir exactement quels sont les périphériques nécessaires et savoir quand mettre le µC en sommeil.

FreeRTOS

Même sur le « petit » µC Cortex-M0+ SAM D20, on peut faire tourner un RTOS (système d'exploitation en temps réel) mais sans la moindre interface utilisateur graphique. FreeR-

TOS, également inclus dans l'ASF, est d'utilisation fréquente pour la famille M. Il est accompagné de quatre programmes d'exemple, à voir le cas échéant. FreeRTOS est un système indépendant et multi-plateforme que le fabricant peut adapter au µC et à ses périphériques. Les avantages de ces SE sont leurs structure et niveau d'abstraction élevé. Ils vous permettent de mettre en œuvre de grands projets, rapidement, sans penser aux bits et octets des registres. Mais la prise en main prend du temps en raison de la structure (des tâches). Pour plus d'infos au sujet de FreeRTOS et un guide de démarrage rapide, cf. www.freertos.org.

Autres programmes intéressants

Intentionnellement, nous n'avons pas couvert dans ce cours certaines bibliothèques, car elles reposent sur des cartes d'extension. Il est néanmoins intéressant d'en voir la structure. Jetez donc un coup d'œil aux programmes d'exemple pour cartes SD avec différents systèmes de fichiers, pour les périphériques Flash d'Atmel et pour l'étalement de la source d'horloge 8 MHz. Pas besoin de tout comprendre, saisir les fonctions de base et les bibliothèques incluses est déjà un grand pas. Penchez-vous aussi sur la déclaration API correspondante.

Aide et petits trucs pour la programmation

Il a été signalé dans le forum diverses erreurs de communication entre la carte SAM D20 et Atmel Studio. Si vous aussi avez des problèmes de reconnaissance de la carte ou de mise à jour du micrologiciel, il vaut la peine de jeter un œil à la page de support d'Atmel (**fig. 1**) à l'adresse www.atmel.com/design-support. Vous pouvez y faire des recherches au sujet de votre problème et de sa solution ; en cas d'échec, vous pouvez faire appel aux experts d'Atmel.

Puisque nous en sommes à parler d'Atmel Studio 6 : le compilateur pour la famille SAM que contient cette suite est très intelligent et pardonne bien des choses. Si vous lisez, par exemple, dans un manuel d'application que le paramètre d'un type d'instruction doit prendre la forme d'un pointeur vers la structure et que vous n'entrez que la structure elle-même en tant que paramètre, Studio reconnaît la saisie et l'accepte, mais ne manque pas, après la compilation, de vous en informer dans les « *Warnings* ». C'est pourquoi il arrive parfois, même dans les programmes d'exemples d'Atmel, qu'il y ait des structures au lieu de pointeurs vers ces dernières. Il en va de même, par exemple, pour des prototypes de fonction, qui ne sont d'ailleurs que rarement nécessaires. Ce « bon caractère » du compilateur de Studio facilite énormément

la vie du concepteur.

Un dernier petit truc : à la compilation, si une erreur de *Make File* est affichée de manière soudaine et inattendue, recompilez le programme - bien souvent cela fonctionne.

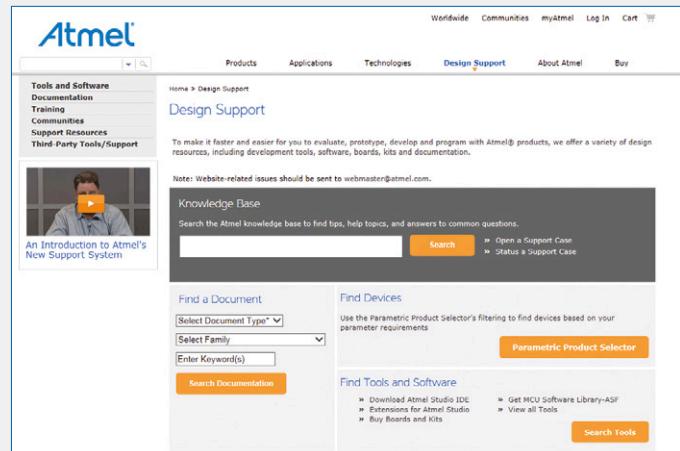


Figure 1. Voici à quoi ressemble la page Design Support sur le site Atmel.

Contrôleur d'accès aux périphériques (PAC)

Ce contrôleur typique de Cortex-M peut bloquer l'accès en écriture de certains registres de périphériques sélectionnés. Cela varie d'un élément périphérique à un autre, cependant il s'agit souvent d'un registre de commande. À cause de ce blocage par le périphérique PAC, la CPU ne peut pas modifier de valeurs dans ces registres. On évite ainsi des accès intempestifs et donc des erreurs. La bibliothèque ASF requise est très simple. Le projet « *PAC Test1* » illustre le processus typique de blocage et son inhibition. Comme le montre le **listage 1**, le programme se résume à la fonction *Main*. Dans celle-ci, on a, après les instructions d'initialisation, blocage par le PAC des registres du port par l'instruction

```
system_peripheral_lock(SYSTEM_PERIPHERAL_ID(PORT),
~SYSTEM_PERIPHERAL_ID(PORT)); .
```

Celle-ci ne contient que l'ID de l'élément périphérique concerné et l'ID inversé par ~. Cet ID inversé constitue simplement une fonction de protection, pour éviter qu'il soit trop facile de bloquer quelque chose par erreur avec cette instruction.

La boucle sans fin fait clignoter la LED0 de la carte. Avant de modifier les bits dans le registre de port (par le biais de l'instruction de commutation *toggle* que nous connaissons déjà), le blocage est levé, puis réactivé. S'il ne se produit pas, pendant ce temps, d'interruption agissant dans certaines conditions sur le port, toutes les interruptions sont, avant inhibition du blocage, désactivées avec `system_interrupt_enter_critical_section();`

et après le blocage, réactivées avec

```
system_interrupt_leave_critical_section();
```

Listage 1. Extrait du fichier *Main* comportant le code de commutation de la LED.

```
system_peripheral_lock(SYSTEM_PERIPHERAL_ID(PORT), ~SYSTEM_PERIPHERAL_ID(PORT));
while(true){
    system_interrupt_enter_critical_section();
    system_peripheral_unlock(SYSTEM_PERIPHERAL_ID(PORT), ~SYSTEM_PERIPHERAL_ID(PORT));
    port_pin_toggle_output_level(LED_0_PIN);
    system_peripheral_lock(SYSTEM_PERIPHERAL_ID(PORT), ~SYSTEM_PERIPHERAL_ID(PORT));
    system_interrupt_leave_critical_section();

    delay_ms(500);
}
```

L'instruction d'inhibition du blocage `system_peripheral_unlock()` requiert exactement les mêmes paramètres que l'instruction de blocage.

`Delay_ms(500)` paramètre la période du clignotement à 1 s. Pour tester ce programme, il vous suffit de le transférer vers la carte. La **figure 1** montre toutes les bibliothèques incluses dans l'Assistant ASF.

Pour plus d'infos sur le PAC, cf. feuille de caractéristiques, p. 34 sqq. et sur <http://goo.gl/kyuzWc>.

Selected Modules

- ▷ Generic board support (driver)
- ▷ Delay routines (service) **systick**
- ▷ PAC - Peripheral Access Controller (driver)
- ▷ PORT - GPIO Pin Control (driver)
- ▷ SYSTEM - Core System Driver (driver)

Figure 1. Ces bibliothèques ont été incluses dans notre projet. Les bibliothèques Delay et PAC se sont ajoutées aux bibliothèques standard.

Mémoire non volatile (NVM)

NVM est l'acronyme de Mémoire Non Volatile. Dans cette zone de mémoire structurée comme une mémoire EEPROM, la MCU peut stocker des données à conserver après une coupure de la tension. Comme le montre la **figure 1**, une partie de la NVM est réservée à l'amorce (*bootloader*) et aux registres internes (pour l'étalonnage des oscillateurs par ex.). On peut utiliser la zone centrale dans son programme. Pour plus d'infos sur la structure et le calcul des adresses, cf. www.atmel.com/Images/Atmel-42114-SAM-D20-Non-Volatile-Memory-Driver-NVM_Application-Note_AT03247.pdf.

Examinons ici le projet « *First test with NVM* », qui traite les trois instructions les plus importantes : effacer, écrire et lire. Après effacement de la mémoire NV, on stocke une chaîne qui, par action sur SW0, est transmise aussi souvent que l'on veut à l'interface UART. Dans ce code, on découvre, après inclusion d'`asf.h`, les prototypes de fonction et les déclarations de tableaux, la fonction de configuration pour l'U(S)ART. On a ensuite une petite fonction de configuration pour la NVM : la structure de configuration créée est d'abord remplie avec le préparamétrage puis la valeur `false` est affectée à la variable de structure

Listage 1. Voici à quoi ressemble la fonction compacte de configuration de la NVM.

```
void configure_nvm(void)
{
    struct nvm_config config_nvm;
    nvm_get_config_defaults(&config_nvm);
    config_nvm.manual_page_write = false;
    nvm_set_config(&config_nvm);
}
```

`manual_page_write` pour faire travailler la mémoire en mode *écriture automatique*. Le **listage 1** montre le transfert à la NVM du paramétrage, le **listage 2** donne le noyau du programme. Ici, une boucle *For* permet de remplir le tableau `page_buffer` avec 64 valeurs (la taille de mémoire d'une page), 0 à 63. On commence, par précaution, avec l'instruction

```
while(nvm_erase_row(100 * NVMCTRL_ROW_PAGES *
    NVMCTRL_PAGE_SIZE) != STATUS_OK){},
```

par effacer cette page. La boucle *While* attend la fin de l'exécution complète de l'instruction. Il nous faut, comme paramètre, l'adresse de page calculée (cf. note d'application). Il est possible dès lors d'écrire le tableau `page_buffer` avec la fonction

```
while(nvm_write_buffer(100 * NVMCTRL_ROW_PAGES *
    NVMCTRL_PAGE_SIZE, page_buffer, NVMCTRL_PAGE_SIZE) != STATUS_OK){}
```

dans la NVM qui se trouve aussi dans une boucle *While* et requiert, comme paramètres, l'adresse calculée, le tableau et la taille de page.

Dans la boucle sans fin, l'instruction de lecture permet de transférer le contenu de la NVM dans le tableau `page_buffer_receive`. Cette instruction requiert les mêmes paramètres que l'instruction d'écriture et se trouve elle aussi au cœur d'une boucle *While*. Les instructions qui suivent transfèrent le tableau à l'interface U(S)ART :

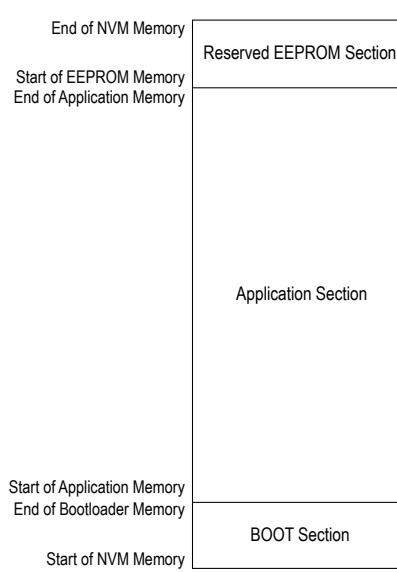


Figure 1. Vue de la cartographie de la mémoire NVM. En ce qui nous concerne, seule la section centrale est intéressante.

Listage 2. L'extrait le plus important de la fonction Main.

```

for (uint32_t i = 0; i < NVMCTRL_PAGE_SIZE; i++) {
    page_buffer[i] = i;
}

while(nvm_erase_row(100 * NVMCTRL_ROW_PAGES * NVMCTRL_PAGE_SIZE) != STATUS_OK){}
while(nvm_write_buffer(100 * NVMCTRL_ROW_PAGES * NVMCTRL_PAGE_SIZE, page_buffer, NVMCTRL_PAGE_SIZE) != STATUS_OK){}
while (true) {
    while(port_pin_get_input_level(BUTTON_0_PIN)){}
    while(nvm_read_buffer(100 * NVMCTRL_ROW_PAGES * NVMCTRL_PAGE_SIZE, page_buffer_receive, NVMCTRL_PAGE_SIZE) != STATUS_OK){}
    while(usart_write_buffer_wait(&usart_instance, &page_buffer_receive, 64) != STATUS_OK){}
    while(!port_pin_get_input_level(BUTTON_0_PIN)){}
}

while(nvm_read_buffer(100 * NVMCTRL_ROW_PAGES * NVMCTRL_PAGE_SIZE, page_buffer_receive, NVMCTRL_PAGE_SIZE) != STATUS_OK){}
while(usart_write_buffer_wait(&usart_instance, &page_buffer_receive, 64) != STATUS_OK){}

```

Les deux instructions sont encadrées par deux boucles *While* qui consultent le niveau de SW0. Ainsi les deux instructions ne sont exécutées qu'en cas d'action sur SW0. Transférez le programme vers la carte et ouvrez ensuite le programme de terminal avec les paramètres de la **figure 2**. Appuyez sur SW0 pour lire les valeurs stockées dans la NVM.
Dans ce projet, on a inclus les bibliothèques NVM et U(S)ART dans la version avec interrogation.

Pour plus d'infos, cf. feuille de caractéristiques, p. 25 sqq. et sur <http://goo.gl/afjKJj>.

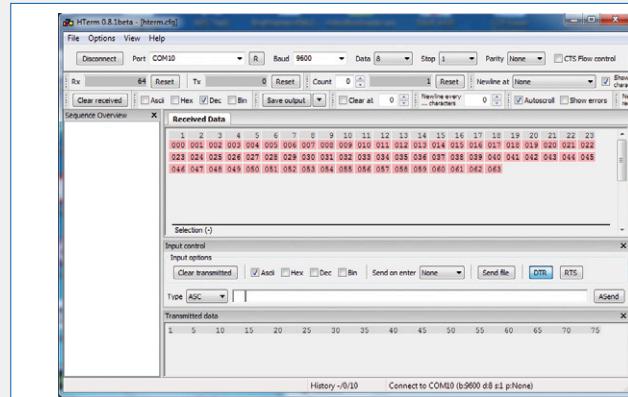


Figure 2. On retrouve ici les paramètres du terminal et les valeurs de sortie suite à une action sur SW0.

Instructions pour les interruptions et l'horloge

Les notes d'application www.atmel.com/Images/Atmel-42122-SAM-D20-System-Interrupt-Driver_Application-Note_AT03261.pdf et www.atmel.com/Images/Atmel-42119-SAM-D20-Clock-Management-Driver-CLOCK_Application-Note_AT03259.pdf mentionnent nombre d'instructions relatives aux interruptions, horloge et autres systèmes ; à consulter !

En voici une sélection :

- `System_interrupt_set_priority()` n'a besoin, comme paramètre, que du vecteur d'interruption de l'interruption requise (sous forme de constante symbolique) et du niveau de priorité de 0 à 3 (une constante symbolique aussi). Cette instruction vous permet de définir ou modifier des priorités d'interruptions. `System_interrupt_get_priority()` ne requiert que le vecteur d'interruption et fournit alors le niveau de priorité de l'interruption concernée.
- `System_interrupt_get_active()` se passe de paramètre et renvoie le vecteur d'interruption de l'interruption en cours. Cette instruction peut être appelée dans une ISR par exemple.

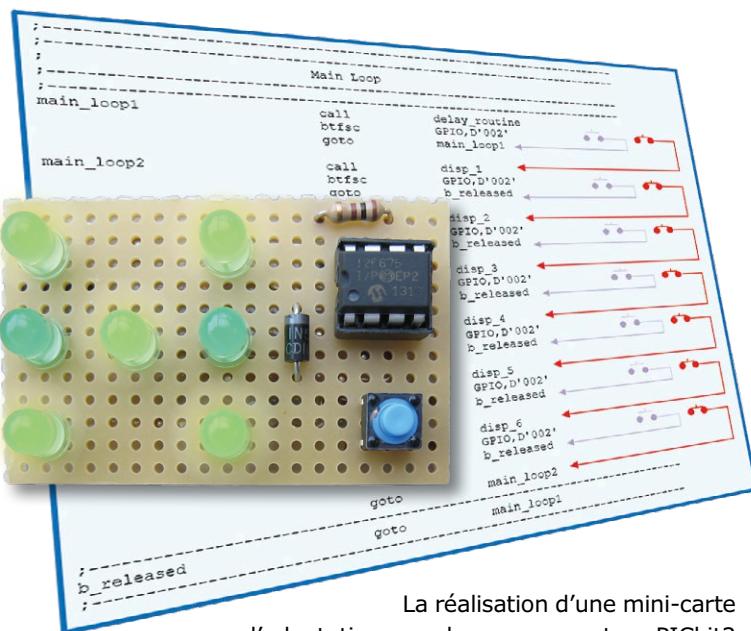
- Avec des instructions telles que `system_clock_source_osc32k_set_config()`, qui requièrent une structure de configuration en tant que paramètre, on peut, en cours d'exécution du programme, activer, désactiver et configurer les sources d'horloge en respectant le modèle ASF typique, avec paramétrage des structures de configuration.
- `system_clock_source_get_hz()` permet de voir quelle source d'horloge produit alors quelle fréquence. Seul paramètre requis, la source l'horloge (sa constante symbolique). `System_apb_clock_source_get_hz()` est similaire et fournit la fréquence actuelle du signal d'horloge du bus périphérique (APB).

Des instructions telles que `system_cpu_clock_set_divider()` permettent de définir les diviseurs de fréquence d'horloge pour les sources d'horloge correspondantes. Le diviseur est fourni à l'instruction comme paramètre sous forme d'une constante symbolique. ▶

(150324 – version française : Guy Raedersdorf)

cours intensif d'assembleur

(2) mini-carte de développement et dé électronique



La réalisation d'une mini-carte

d'adaptation pour le programmeur PICkit2

prend quelques minutes. Ce type de carte facilite énormément la programmation du microcontrôleur (μ C) correspondant : enficher le μ C, connecter la carte au programmeur, lancer le logiciel de programmation, transférer par son biais le micrologiciel (*firmware*) assemblé et l'affaire est réglée. Sans gros efforts, il est possible de fabriquer des cartes d'adaptation pour des μ C avec différents nombres de broches ; chaque μ C de Microchip comporte les broches de programmation requises : V_{pp}, ICSPDAT et ICSPCLK. Il faut donc, avec ces cartes d'adaptation, relier lesdites broches (sans oublier V_{cc} et GND) aux broches de même nom du programmeur PICkit.

Pour notre μ C PIC12F675 à huit broches, on réalisera une variante à huit broches comme celle de la **figure 1**. Convaincu ? Simplissime. Un mini-morceau de platine à trous, un support pour circuit intégré à huit broches et une embase à six broches, c'est tout. On pourra, le cas échéant, supprimer la LED et sa résistance-série, mais le témoin d'alimentation du μ C

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

Dans la 1^{ère} partie, nous avons abordé les bases de la programmation en assembleur et les aspects matériels du contrôleur PIC utilisé. Nous continuons avec une mini-carte de développement faite maison et de nouvelles instructions d'assembleur. Notre application de « démonstration » prendra la forme d'un dé (électronique).

reste utile. La **figure 2** montre comment raccorder la platine d'adaptation terminée au programmeur.

Zoom sur les registres

Avant de découvrir d'autres instructions d'assembleur utilisées dans notre application de « démo », jetons un rapide coup d'œil aux registres les plus importants.

Registre de configuration (Config)

Si la plupart des registres du μ C se trouvent en SRAM, le registre de configuration fait exception. Le registre possède 14 bits, mais, pour l'instant, seuls quatre de ces bits sont pertinents. Les trois LSB (**Least Significant Bit** = bit de poids faible) FOSC2:FOSC0 servent à la configuration de l'oscillateur. Dans le cours, nous utilisons les deux paramétrages suivants de ces trois bits :

010 : oscillateur HF piloté par quartz. Dans ce mode, on utilise l'oscillateur externe. La fréquence est définie par un quartz externe entre les broches 2 (GP4) et 3 (GP5). On peut ainsi pousser la fréquence d'horloge jusqu'à 20 MHz maximum. Le quartz occupe deux broches, seules quatre des six E/S potentielles restent disponibles.

100 : oscillateur interne. Par ce mode, on active l'oscillateur

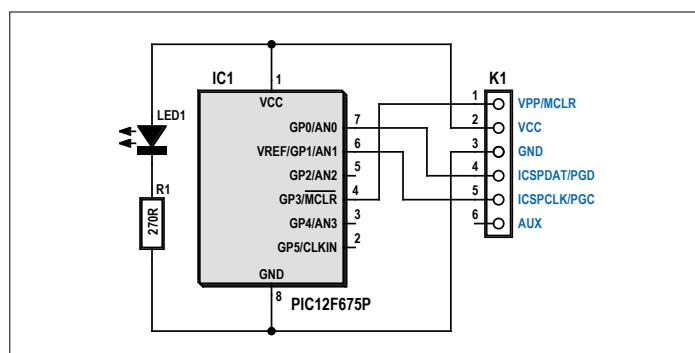


Figure 1. Schéma de la carte d'adaptation, une carte de développement en version minimale.

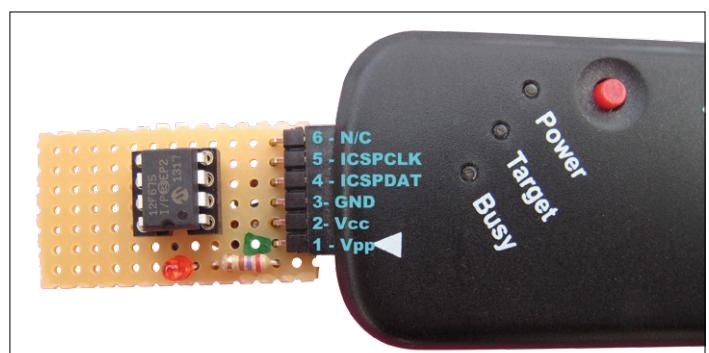


Figure 2. Carte d'adaptation terminée, connectée au programmeur PICkit2.

IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C
bit 7				bit 0			

Figure 3. Signification des bits du registre d'état (*Status Register*).

interne de 4 MHz. La fréquence d'horloge est moins précise qu'avec un pilotage par quartz, mais, avec sa tolérance de $\pm 1\%$, elle suffit à de nombreuses applications. Son gros avantage : permettre de se passer de composants additionnels, GP4 et GP5 restent disponibles comme E/S d'usage général.

Le bit 5 (MCLRE) du registre de configuration commande la fonction de la broche 4. Si MCLRE = 1, la broche 4 sert de *Master Clear* – c.-à-d. de broche de réinitialisation (*reset*). Si MCLRE = 0, la broche 4 peut être utilisée comme entrée, avec en corolaire la perte de la fonction de réinitialisation.

Registre d'état (*Status*)

Le registre d'état contient des informations relatives à des opérations arithmétiques terminées. La **figure 3** donne la signification des différents bits du registre.

Les deux MSB (**Most Significant Bit** = bit de poids fort) IRP et RP1 sont réservés et doivent toujours être à 0 en écriture. Ils ne sont pas utilisés avec le PIC12F675.

Le bit 5, RP0, est le *Register Bank Select Bit*. Il sert (cf. 1^{ère} partie du cours) à la sélection de la banque de mémoire. S'il est à 0, toutes les instructions qui suivent accèdent aux emplacements de mémoire 00h à 7Fh – sinon, s'il est à 1, aux emplacements 80h à FFh.

Les deux bits TO et PD révèlent l'origine de la dernière réinitialisation (cf. fiche de caractéristiques [1]).

Le bit 2 Z (= Zero) est activé si le résultat de la dernière opération est égal à zéro – il n'est pas impératif qu'il ait s'agit d'une opération arithmétique.

Les deux LSB indiquent les débordements suite à une addition ou une soustraction (états *Carry*).

Registre des options (*Option*)

Le registre des options permet d'activer d'autres paramètres. Notre application de « démo » (cf. plus loin) donne un exemple : l'effacement du bit 7 active la fonction de polarisation haute (*pull-up*) des entrées.

Cartographie de la mémoire

Pour programmer en assembleur, il faut connaître la cartographie de la mémoire du microcontrôleur utilisé. Généralement (et donc ici), la mémoire est divisée en trois zones :

- La mémoire de programme (Flash)
- La mémoire de données (SRAM)
- L'EEPROM

La mémoire de programme est utilisée, en règle générale – et même exclusivement avec le PIC12F675 – pour y stocker le logiciel. La largeur d'un mot y est de 14 bits. Si cette information n'a pas grande importance en programmation, elle nous apprend que la mémoire de programme d'un PIC12F1675 ne comporte pas 1024 **octets** (de 8 bits chacun), mais 1024 **mots** (de 14 bits chacun).

La mémoire de données regroupe tous les registres spécifiques au processeur et à l'application. Hormis les registres du processeur préparamétrés, le contenu de la mémoire de données est indéfini à la mise sous alimentation. Elle perd son contenu à la disparition de l'alimentation – il s'agit de SRAM volatile. Contrairement à la mémoire de programme, la mémoire de données est, classiquement, organisée en octets (8 bits chacun). La fiche de caractéristiques [1] (p. 9 sqq) donne une description détaillée de la cartographie de la mémoire. À retenir : les 64 octets de mémoire de l'application se trouvent dans la plage d'adresses entre 20h et 5Fh.

Par logiciel, on pourra, stocker dans l'EEPROM, mémoire non volatile organisée en octets elle aussi, des valeurs qui « survivent » à une réinitialisation ou à un redémarrage. Cependant, il faut éviter les écritures trop fréquentes.

Autres instructions

Passons aux instructions dont, outre celles déjà évoquées dans la 1^{ère} partie de cours, nous aurons besoin pour notre projet de « démo ». À noter que la fiche de caractéristiques de tout µC de Microchip donne une liste exhaustive de toutes les instructions [1].

Notre description commence par un complément d'informations sur une instruction déjà utilisée dans la première partie.

DECFSZ

Dans la 1^{ère} partie, nous avons parlé de la fonction de l'instruction DECFSZ (= **D**E**C**rement **F** and **S**kip if **Z**ero) pour les boucles. En dehors des boucles, cette instruction permet aussi de procéder à des branchements généraux lors de l'exécution du programme (une boucle n'est qu'un type de branchement spécifique). La **figure 4** montre l'ordinogramme.

Supposons qu'il y a, à l'emplacement de mémoire 20h, une variable X. Un branchement complet requiert trois instructions (cf. **fig. 5**). En utilisant l'argument « 0 » avec DECFSZ, le résultat de la soustraction (valeur dans « X » moins 1) est écrit dans le registre W. Si le résultat est 00h, on saute l'instruction suivante et la troisième instruction goto branche_B est exécutée.

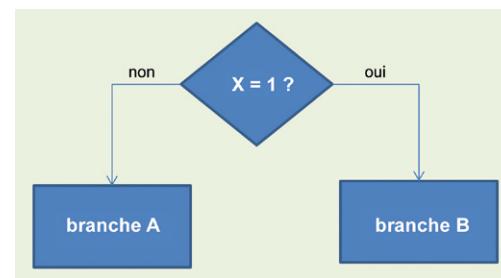


Figure 4. Ordinogramme d'un branchement conditionnel.

exemple de branchement DECFSZ :

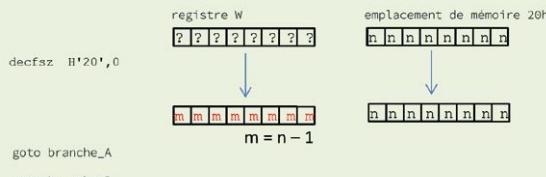


Figure 5. Exemple de code d'un branchement conditionnel.

Cela n'a lieu que si la valeur de X est, avant décrémentation, de 01h. Dans tous les autres cas, on se retrouve à branche_A.

INCFSZ

Vous savez tout de l'instruction DECFSZ. INCFSZ est identique à ceci près qu'il faut lire incrémentation au lieu de décrémentation.

BTFSS et BTFSC

Pour les deux instructions BTFSS (= **B**it **T**est **F** and **S**kip if **S**et) et BTFSC (= **B**it **T**est **F** and **S**kip if **C**lear), il s'agit aussi de branchements. Elles permettent de vérifier l'état des différents bits d'un emplacement de mémoire et d'effectuer le branchement correspondant.

La syntaxe est :

btfss f,b

ou

btfsc f,b

L'argument *f* a une valeur comprise entre 00h et 7Fh et représente l'adresse de l'emplacement de mémoire. L'argument *b*

exemple BTFSS / BTFSC :



Figure 6. Exemple de boucle avec les instructions BTFSS / BTFSC.

est un nombre entre 00h et 07h ; il définit le bit à tester. Ces instructions ne modifient ni le contenu du registre W, ni celui de l'emplacement de mémoire.

Dans l'exemple de la **figure 6**, l'emplacement de mémoire 20h a la valeur 98h (= 10011000b). À gauche l'instruction BTFSS vérifie si la valeur du bit 00h est de 1. Si oui, on saute l'instruction suivante goto branche_A. Cependant, comme la valeur du bit 0 est 0, ce saut n'a pas lieu et l'instruction goto branche_A est alors exécutée.

À droite on voit l'effet de l'instruction BTFSC. Ici on effectue un saut si le bit 00h a la valeur 0 – ce qui est le cas pour la valeur de l'emplacement de mémoire 20h. On saute donc l'instruction goto branche_A et le programme poursuit avec l'instruction goto branche_B.

ADDWF

Cette instruction ajoute le contenu du registre W à celui d'un emplacement de mémoire.

La syntaxe est :

addwf f,d

L'argument *f* désigne l'adresse-cible et *d* une valeur comprise

exemple ADDWF - résultat écrit dans le registre W :

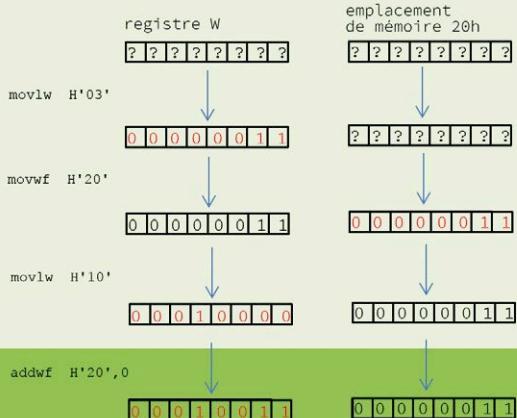


Figure 7. Exemple de code ADDWF : résultat dans le registre W.

exemple ADDWF - résultat écrit dans l'emplacement de mémoire :

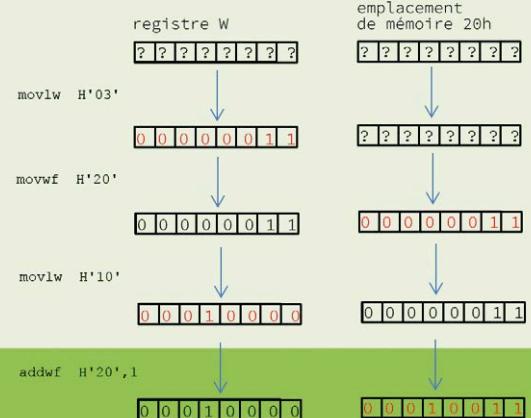


Figure 8. Exemple de code ADDWF : résultat en SRAM.

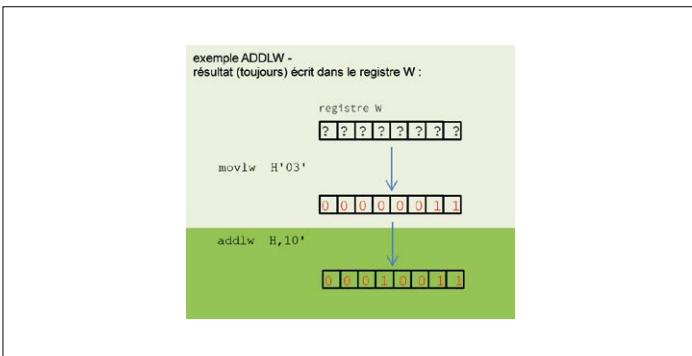


Figure 9. Addition d'une constante au (contenu du) registre W.

entre 00h et 7Fh. L'argument *d* peut être à 0 ou 1 ; il définit l'endroit de stockage du résultat. Si *d* = 0, le résultat est stocké dans le registre W et l'emplacement de mémoire reste inchangé. Si *d* = 1, le contenu de l'emplacement est remplacé par le résultat.

Dans l'exemple ci-après, nous allons calculer la somme de 3 + 16 (en décimal). On stocke tout d'abord 3 dans l'emplacement de mémoire 20h ; on écrit ensuite 16 (10h) dans le registre W et on effectue ensuite l'addition. Le résultat est écrit soit dans le registre W (**fig. 7**), soit en 20h (**fig. 8**).

ANDWF

Pour ce qui est de ses syntaxe et utilisation, cette instruction est identique à ADDWF. C'est la fonction qui diffère : ANDWF effectue l'opération logique ET (AND), bit par bit.

ADDLW

À l'image de l'instruction ADDWF, ADDLW effectue elle aussi une addition. ADDWF ajoute quant à elle une valeur de mémoire au registre W tandis qu'ADDLW ajoute une constante au registre W. La syntaxe est :

addlw k

L'argument *k* a une valeur comprise entre 00h et FFh ; il s'agit de la constante additionnée (au contenu) du registre W. Le résultat est enregistré dans le registre W.

Dans l'exemple de la **figure 9**, on commence par charger la valeur 03h dans le registre W, on additionne ensuite la constante 10h.

ANDLW

Cette instruction est identique à ADDLW à ceci près que l'opération ET (AND) remplace l'addition.

CLRF et CLRW

Nous avons déjà rencontré l'instruction CLRF dans la partie 1 (un emplacement mémoire donné est mis à 00h). De façon similaire, l'instruction CLRW met le registre W à 00h.

La syntaxe est :

clr f

ou

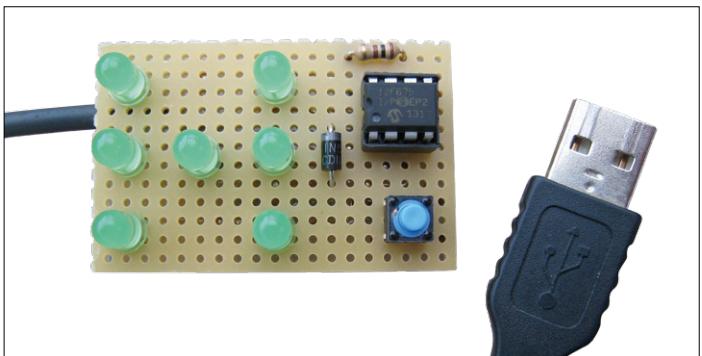


Figure 10. Prototype du dé électronique.

clrw

INCF et DECF

Ces instructions nous permettent d'incrémenter (INCF ; +1) ou de décrémenter (DECF ; -1). La syntaxe est :

incf f,d

ou

decf f,d

Les arguments *f* et *d* ont la même signification que dans le cas de l'instruction ADDWF. En fonction de *d*, le résultat est écrit dans le registre W ou à l'emplacement de mémoire d'origine.

Dé électronique

La réalisation d'un dé électronique n'est pas bien plus compliquée que l'incontournable clignotement de LED. Comme le circuit est, pour le simplifier au maximum, alimenté par un port USB d'un PC, la **figure 10** comporte un port USB. Sinon, le dé n'a rien à faire avec l'interface USB, on pourrait tout autant l'alimenter avec un adaptateur secteur, ou trois piles (rechargeables) en série.

Si vous examinez un vrai dé, vous constaterez que les points se trouvent à sept positions différentes. La **figure 11** illustre la représentation de chacun des chiffres. Il nous faut, pour traduire ces points par des LED, exactement quatre sorties d'un µC. Si cela vous étonne, en voici le pourquoi : les paires de LED 6 et 7, 2 et 3 ainsi que 4 et 5 ne sont toujours allumées ou éteintes qu'en ensemble. Chaque paire est donc pilotée par sa propre broche de port GPIO. La LED1 a sa propre ligne de commande. Pour éviter que LED1 ne soit plus claire que les autres, elle est dotée d'une résistance-série, R1. Bien que la connexion directe de LED aux broches GPIO (cf. le schéma de la **figure 12**) ne soit pas vraiment conforme aux règles, les

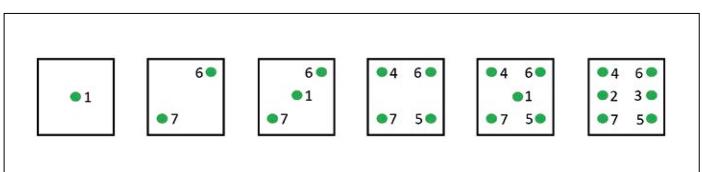


Figure 11. Représentation des chiffres sur un dé.

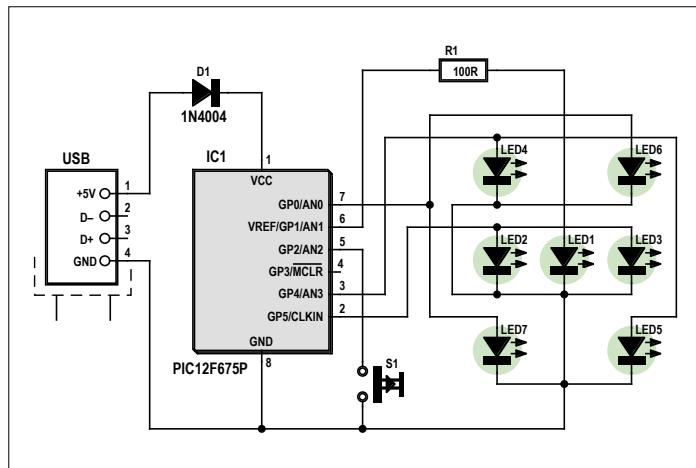


Figure 12. Schéma du dé électronique.

faibles courants mis en jeu ne posent pas de problème réel. Aux fins d'économiser de l'énergie, le 5 V du port USB peut être diminué de 0,7 V environ par D1 et ramené à 4,3 V. En cas d'alimentation par trois piles NiMH, soit $3 \times 1,2$ V, on pourra supprimer la diode. Pour lancer le dé, on appuie brièvement sur le bouton S1.

Petite note : en raison de leur tension directe plus élevée, on ne pourra pas utiliser de LED bleues ou blanches dans ce circuit.

Micrologiciel du dé

Avant de se lancer dans la programmation, il est bon de bien spécifier les fonctions du logiciel de commande. Voici le cahier des charges du dé :

- Après la mise en fonction (*reset*) mettre le dé en démo. Celle-ci consiste à faire défiler tous les chiffres (1 à 6) jusqu'à une action sur le bouton S1.
- L'implémentation de la logique du dé prend la forme d'une boucle sans fin :

```

;-----[LED Maestro]-----*
;-----v 1.05 - 14.03.2015-----*
;-----Hardware: PIC12F675 used-----*
;-----Osc.: Internal osc. used-----*
;-----POWER.....: 5V USB-----*
;-----[Pins connections]-----*
;-----GP2 -- button-----*
;-----GP3 -- N/C-----*
;-----GP4-----GP0-----*
;-----GP5-----GP1-----GP0-----*
;-----GP0-----GP4-----*
;-----[INCLUDE "P12F675.INC"]-----*
_U002      _U002      EQU      B'00000110000100'      ;MCLR = input ①
;-----[Variable Definition]-----*
;-----TIMER1      EQU      H'20'      ;Used in delay routine
;-----TIMER2      EQU      H'21'      ;"
;-----[-----]
;-----bsf      STATUS,RP0      ;Switch to register bank 1
;-----movwf     D'11001100'      ;:GP0, 1, 4, 5 - out; GP2, 3 - in
;-----TRISIO
;-----clrf      ANSEL      ;GPIO are digital I/O's
;-----bcf      OPTION_REG,D'007'      ;enable pull-ups
;-----movlw     B'00000100'
;-----movwf     WPU      ;enable pull-up for GP2
;-----btfss    GPIO,D'002'      ;Switch Back to reg. Bank 0
;-----return
;-----[-----]

```

Figure 13. Code du dé électronique : initialisations.

- En cas d'action sur S1, les valeurs 1 à 6 s'affichent successivement très rapidement. En raison de la persistance rétinienne, on voit le chiffre 7 (toutes les LED allumées).
- Le comptage rapide se termine au lâcher de S1. L'état instantané du compteur s'affiche définitivement.

Allons-y.

Le micrologiciel du dé comporte deux parties : lancement du dé et affichage. Il faut commencer par définir certains paramètres après réinitialisation ou redémarrage.

Dans le code de la **figure 13**, le repère 1 identifie cette configuration. On commence par la sélection de l'oscillateur interne et le paramétrage de GP3 en entrée – sans MCLR ensuite. Au repère 2, on attribue les emplacements de mémoire 20h et 21h aux variables TIMER1 et TIMER2.

Aux repères 3 et 4, on travaille avec la banque de mémoire 1 puisque le bit RP0 du registre d'état est à un.

Au repère 3, les deux premières instructions (`movlw B'11001100'` et `movwf TRISIO`) font que GP0, GP1 GP4 et GP5, se trouvent configurées en sorties par mise à zéro des bits correspondants du registre TRISIO. Les LED sont connectées à ces quatre broches. Comme elle sert à la lecture de l'état du bouton, GP2 est configurée en entrée. Le bit 2 du registre TRISIO est donc mis à un. L'instruction `clrf ANSEL` désactive la fonction analogique. Au repère 4, on contourne l'implantation d'une résistance de polarisation haute externe pour le bouton. Le µC PIC12F675 comporte en effet sur toutes les broches GPIO configurées en entrée (hormis GP3), des résistances de polarisation haute dites « molles » (*weak pull-up*). Pour cela, le bit 7 du registre des options doit être mis à zéro. Dans le programme, c'est l'instruction `bcl OPTION_REG,D'007'` qui le fait. Ensuite, avec les bits du registre WPU, on définit pour lesquelles des broches les résistances de polarisation internes doivent être activées. Si le bit x du registre WPU est mis à un, la résistance de polarisation haute de la broche GPx correspondante est active. En principe, les résistances de polarisation haute internes ne sont pas actives car elles peuvent augmenter la consommation de courant jusqu'à 400 µA. Si c'est vraiment important, on plantera une résistance de polarisation haute externe de forte valeur ou si nécessaire une résistance de polarisation basse externe.

Après l'initialisation, le programme entre dans la boucle de « démo » (**fig. 14**). L'implémentation à ce niveau est facile – tout se passe dans le sous-programme `start_effect`, auquel il est tout simplement fait appel ici. Une action sur S1 termine la partie de « démo ».

La routine de « démo »

La routine de « démo » illustrée en **figure 15** est parlante. On y reconnaît six parties de programme similaires, une par chiffre. La représentation des chiffres se fait par un sous-programme : `disp_1` se charge d'allumer la LED du chiffre 1. `disp_2` provoque l'allumage du chiffre 2 et ainsi de suite.

Après l'affichage d'un chiffre donné, le logiciel attend avant le chiffre suivant. C'est la fonction de `delay_routine` – une routine déjà utilisée dans la 1^{ère} partie du cours. L'instruction `btfss GPIO,D'002'` mérite le détour. Elle sert, bouton ouvert, à sauter par-dessus l'instruction `return` et ainsi à produire l'affichage du chiffre suivant. En cas d'action sur le bouton, ce saut n'est plus effectué puisque l'entrée fournit un 0. Le programme

continue depuis ce return. Ceci termine le sous-programme *start_effect* et donc la démo. Dès lors on retrouve l'exécution du programme principal.

Venons-en aux routines d'affichage : le code de cette partie (**fig. 16**) est étonnamment simple. On y retrouve six routines. Dans chacune d'entre elles, la première instruction `movlw` détermine par le biais d'une constante quelles sont les LED allumées ou éteintes. Dans les constantes binaires, un « 0 » correspond aux LED éteintes et un « 1 » aux LED allumées. L'instruction `movwf` GPIO envoie ces informations au port. On attend ensuite un court instant (sous-programme *dr2*) avant de terminer la routine d'affichage. Dans la partie de « démo », cette courte attente n'aurait pas lieu d'être, mais elle ne gêne en rien.

Programme principal

La structure de cette partie est similaire à la routine de « démo ». Dans la **figure 17**, la partie étiquetée *main_loop1* fait en sorte que, bouton relâché, il y ait affichage ininterrompu du chiffre en cours. Comme dans la routine de « démo », on a aussi, dans le programme principal, interrogation de la valeur actuelle de l'entrée GP2 et en cas d'action sur le bouton poursuite du comptage avec affichage de la valeur actuelle. Tout va si vite qu'on a l'impression que le chiffre « 7 » est affiché. Après relâchement du bouton, l'instruction `goto b_released` est exécutée ; le comptage s'arrête – l'état (du compteur) à cet instant reste affiché. Ce chiffre est le résultat du (lancer de) dé. Depuis l'étiquette *b_release*, on ressaute à *main_loop1* qui arrête l'exécution du programme. On aurait aussi pu sauter directement à *main_loop1* ; pour rendre le programme plus vivant, on pourrait ajouter des effets sous l'étiquette *b_release*, par exemple, faire s'éteindre le comptage progressivement ou faire clignoter le résultat et ne passer qu'ensuite à l'affichage statique.

Dans le sous-programme *dr2*, on détermine la vitesse de changement des chiffres. En fait, on pourrait aussi s'en passer. Essayez de ralentir l'exécution du programme au point de voir les chiffres changer.

Comme à l'accoutumée, le logiciel est téléchargeable depuis le site Elektor [4].

Perspectives

Nous voici arrivés à la fin de la 2^e partie de ce cours d'assemblleur. Nous y avons décrit de nouvelles instructions et vu comment réaliser à peu de frais (et d'efforts) un dé électronique fonctionnel. On pourra, selon le même modèle, réaliser des applications sensiblement plus complexes. Vous avez pu constater ainsi que la programmation en assemblleur est plus facile qu'on ne le pense au premier abord.

J'ose espérer que cela vous a plu ! Dans le prochain épisode, nous verrons comment utiliser un µC simple tel que le PIC12F675 pour remplacer le temporisateur NE555. ↗

(150274 – version française : Guy Raedersdorf)

Liens

- [1] PIC12F675 : www.microchip.com/wwwproducts/Devices.aspx?product=PIC12F675
- [2] MPLAB IDE : www.microchip.com/pagehandler/en-us/family/mplabx
- [3] Première partie du cours: www.elektormagazine.fr/130483
- [4] Logiciel : www.elektormagazine.fr/150274

```
-----  
;  
-----  
Effect  
-----  
clrff GPIO start_effect
```

Figure 14. Code du dé électronique : saut vers la routine de « démo ».

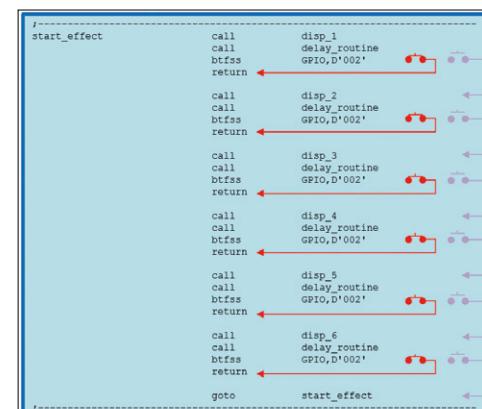


Figure 15. Code du dé électronique : production des chiffres dans la routine de « démo ».

```
-----  
disp_1:  
    movlw B'00000010'  
    movwf GPIO  
    dr2  
    return  
  
-----  
disp_2:  
    movlw B'00000001'  
    movwf GPIO  
    dr2  
    return  
  
...  
-----
```

Figure 16. Code du dé électronique : routine d'affichage.

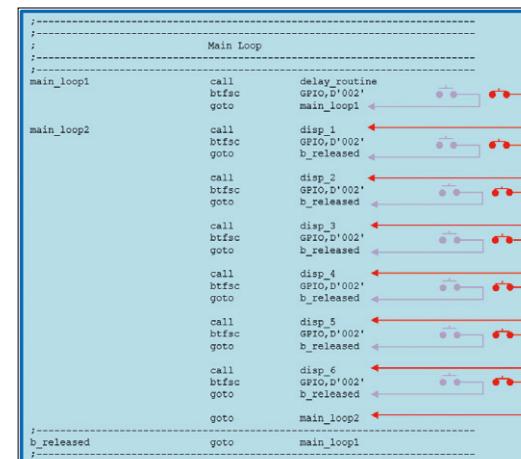


Figure 17. Code du dé électronique : le programme principal.

Technobot 2015

électronicien(ne)s de demain ?

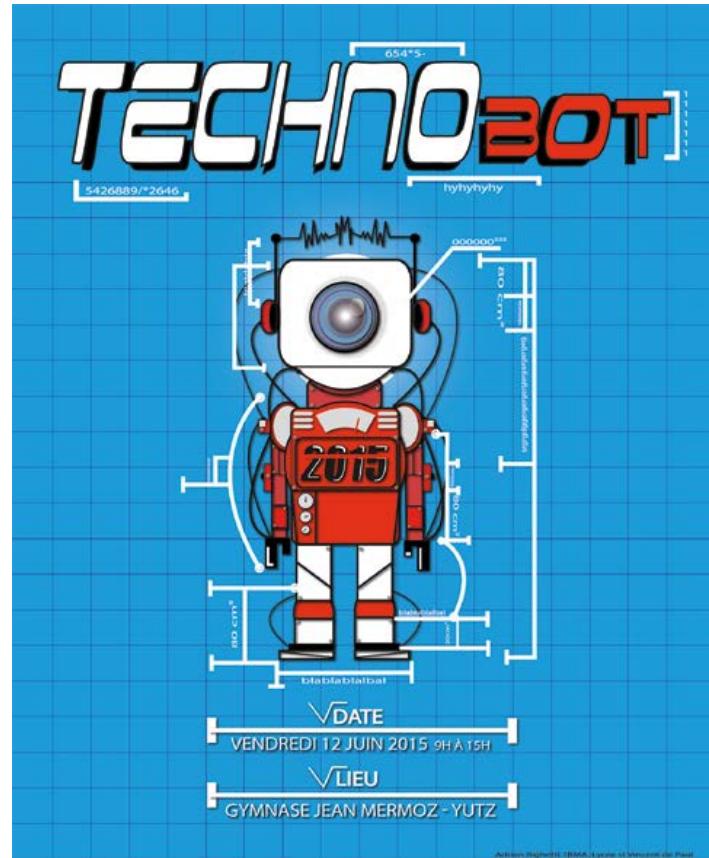
Marilene Thiebaut-Brodier, Elektor

Comment intéresser les collégiens à la technique et les inciter à s'orienter vers les filières technologiques ? Passer de consommateur de nouvelles technologies à acteur, programmeur... Le vendredi 12 juin 2015, dans une ambiance surchauffée (tant en température qu'en émotions) a eu lieu la 4^e édition du Technobot qui a réuni 21 collèges de l'académie de Metz-Nancy autour d'un tournoi de robots. Pour ces 600 élèves de 3^e, c'était le point d'orgue d'une année de travail.

François Heckel (Hettange-Grande) [1] et Olivier La Nève (Yutz) ont organisé un 1^{er} Technobot en 2012 pour que les élèves de leurs collèges respectifs se rencontrent. Depuis, ces deux professeurs de technologie forment leurs homologues pour qu'ils les rejoignent dans l'aventure. Dans le cadre de leurs heures de technologie, les élèves de 3^e construisent un robot pour l'une des trois catégories suivantes :

Design : l'habillage du robot doit être particulièrement soigné sans oublier un certain nombre de critères techniques (ventilation du circuit, accessibilité du compartiment à piles et du connecteur de programmation...) – **figure 1.** Les robots de cette catégorie doivent pouvoir suivre une ligne.

Suivi de ligne : le plateau de jeu est une planche de bois blanche semi-lisse, sur laquelle sont tracées des lignes avec de



l'adhésif noir de 19 mm de large. Chaque robot en compétition doit parcourir plusieurs tracés, de complexité croissante (virage serré, angle droit, croisement, absence de ligne sur quelques centimètres...). Les points sont attribués en fonction de la distance parcourue et du temps de parcours.

Formule robot : il s'agit d'une course de vitesse entre deux robots autonomes sur une piste ovale fermée. Les robots sont



Figure 1 – Participants au concours de design : place à l'imagination !



Figure 2 – Le robot Darwin (utilisé en lycée technique) suit la balle rouge grâce à sa caméra.



Figure 3 - Bras robotisé réalisé avec une imprimante 3D.

placés dans le même couloir, mais à des points opposés. Le gagnant est celui qui rattrape son adversaire en moins de trois tours de piste.

Quelle que soit la catégorie choisie, les élèves doivent construire leur robot pour un prix maximum de 90 €, sans utiliser de kits, et fabriquer eux-mêmes au moins une pièce (roues, châssis...). Des élèves et des professeurs de lycée professionnel leur viennent en aide pour les questions techniques et la mise à disposition de matériel particulier (imprimante 3D, fraiseuse...). Nombre de robots reposent sur des PICAXE parce que leur programmation est graphique.

Robots en surpoids

Le jour J, après des épreuves de qualification au collège, les équipes retenues soumettent leur « œuvre » au contrôle administratif et technique afin d'obtenir une immatriculation. La charte à respecter permet entre autres de vérifier que le robot a bien été réalisé par les élèves et non par leur professeur. En outre la programmation doit être effectuée exclusivement par les élèves.

Pour la catégorie *design*, chaque équipe présente son projet à un jury composé d'un chef des travaux et de professeurs de langue. Les élèves peuvent alors obtenir des points de bonus s'ils glissent quelques phrases dans une ou plusieurs langues étrangères.

La dernière étape avant de rejoindre la piste est la pesée (moins de 1 kg) et le contrôle du gabarit (27x35x34 cm). Certains concurrents ont été amenés par exemple à alléger leur engin de 30 g en y accrochant des ballons gonflés à l'hélium. La dernière étape avant de rejoindre la piste est la pesée (moins de 1 kg) et le contrôle du gabarit (27x35x34 cm). Certains concurrents ont été amenés par exemple à alléger leur engin de 30 g en y accrochant des ballons gonflés à l'hélium.



Figure 4 – Le même jour avait lieu une épreuve de mini-sumo scolaire, réservée aux clubs de robotique : chaque participant cherche son adversaire pour le pousser hors du cercle noir.



Figure 5 – Gymnase en surchauffe au moment des finales.



Figure 6 – Trophées remis en jeu chaque année.

Les finalistes issus des manches du matin se retrouvent l'après-midi après que leur robot ait été déprogrammé. Le stress augmente encore parce qu'il faut recharger le code dans des conditions moins confortables qu'en classe.

Association TechTic&Co | La Culture scientifique et technique pour tous

L'association TechTic&Co [2] collabore activement à l'organisation du Technobot. Cette association créée en 2011 a pour objectif, auprès des jeunes notamment, de donner du sens à la science et aux nouvelles technologies, de les rendre visibles, palpables, de les éléver au rang de culture...

Ses fondateurs sont partis du constat paradoxal que les jeunes plébiscitent les nouvelles technologies, mais vivent les sciences uniquement comme un mode de sélection, et

non plus comme une culture. Leur objectif (ambitieux) est de monter des actions pour créer des passerelles entre les sciences et le grand public, mais aussi entre les sciences et le monde industriel.

Outre le Technobot, l'association est impliquée dans FesThi'Sciences (festival des sciences pour petits et grands), des Cafés des Sciences, des Ateliers des Sciences...

D'une manière générale, lors de la conception, les élèves sous-estiment les problèmes de chauffe ou encore la finesse nécessaire pour les réglages.

Croisements

Les professeurs sont attentifs à ce que chaque élève trouve sa place dans le projet « robot ». Un élève peu à l'aise avec la programmation ou la mécanique pourra apporter sa pierre à l'édifice sur un autre plan (dossier de présentation, habillage du robot...). En outre, comme les professeurs de plusieurs disciplines (langues, arts plastiques...) interviennent pendant la préparation ainsi que la présentation finale, il s'agit là d'un bel exemple d'EPI (enseignement pratique interdisciplinaire) dont on a entendu parler ces derniers mois à propos de la Réforme du collège.

Le partenariat collèges-lycées techniques permet aux collégiens de découvrir l'existence du lycée technique et d'envisager une orientation vers cette voie. En classe de seconde, ils pourront se perfectionner dans les clubs de robotique. Ils seront alors épaulés par des élèves et professeurs d'écoles d'ingénieurs qui eux-mêmes concourent à Technobot² (réservé aux post-BAC). Grâce à ces rencontres, l'enseignement supérieur entre en contact avec le collège, un décloisonnement inhabituel dans l'Éducation Nationale.

Si vous aussi, vous cherchez à développer la fibre technique de vos élèves, avec un peu ou beaucoup d'électronique, venez nous parler de vos aventures. Contactez la rédaction (redaction@elektor.fr). ▶

(150460)

Liens

- [1] <http://labriquerie.net/actualites-lecteur/items/technobot-2015-4eme-edition.html>
- [2] <http://www.techtic-co.eu/>

comprendre l'amplificateur à transistor sexagénaire – vaillant & toujours indispensable

Robert Lacoste (Chaville)

Revenir aux bases, ça peut mener loin. Beaucoup d'électroniciens, et même, j'en ai peur, des électroniciens professionnels, à l'aise avec des microcontrôleurs et autres circuits complexes, peuvent être déroutés par un simple petit transistor. Je vous propose donc quelques rappels sur l'art et la manière d'amplifier un signal avec un simple transistor bipolaire (ou BJT, pour *Bipolar Junction Transistor*). Pas d'inquiétude, la théorie à connaître restera au niveau de la loi d'Ohm !

Le transistor est votre ami

Le transistor BJT est né en décembre 1947. Ses inventeurs, Walter H. Brattain et John Bardeen, travaillaient dans l'équipe de William Shockley aux laboratoires Bell d'où sont issues tant d'autres inventions techniques révolutionnaires. Il existe deux versions de transistors bipolaires, appelés respectivement NPN et PNP. Je ne parlerai dans cet article que du NPN, mais tout ce que j'en dirai s'applique aussi au PNP à condition de permettre la polarité de l'alimentation. Un transistor BJT a trois pattes : collecteur (C), émetteur (E) et base (B). La magie, ou

plus exactement la physique liée à sa structure semi-conductrice interne, implique que les tensions et les courants à ses bornes sont liés. Voyons cela avec le montage dit en émetteur commun (**fig. 1**) : l'émetteur est à la masse. Tout le fonctionnement du transistor tient ici en deux règles :

Règle 1 : Le courant circulant via le collecteur est proportionnel au courant circulant dans sa base. Le ratio entre

ces courants est appelé gain de courant du transistor, noté h_{FE} (de l'anglais h pour *hybrid equivalent circuit* et F pour *forward current amplification* et E pour *current amplification*. Les majuscules indiquent ici qu'il s'agit de courant continu).

On a donc :

$$I_C \approx I_B \times h_{FE}$$

Règle 2 : La tension entre base et émetteur est stable, et reste à peu près égale à 0,6 V (comme pour toute diode bipolaire) :

$$V_{BE} \approx 0,6 \text{ V}$$

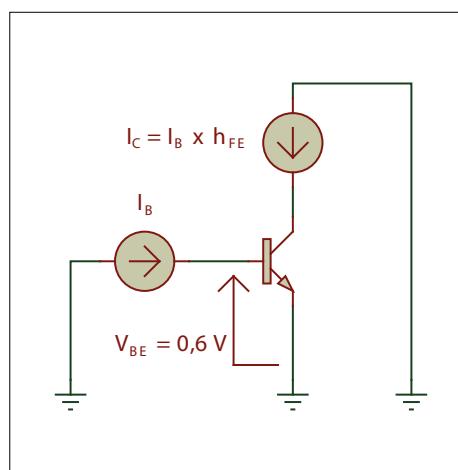


Figure 1 : On dit d'un transistor NPN qu'il est monté en émetteur commun quand son émetteur est connecté à la masse. Deux équations dictent son comportement.

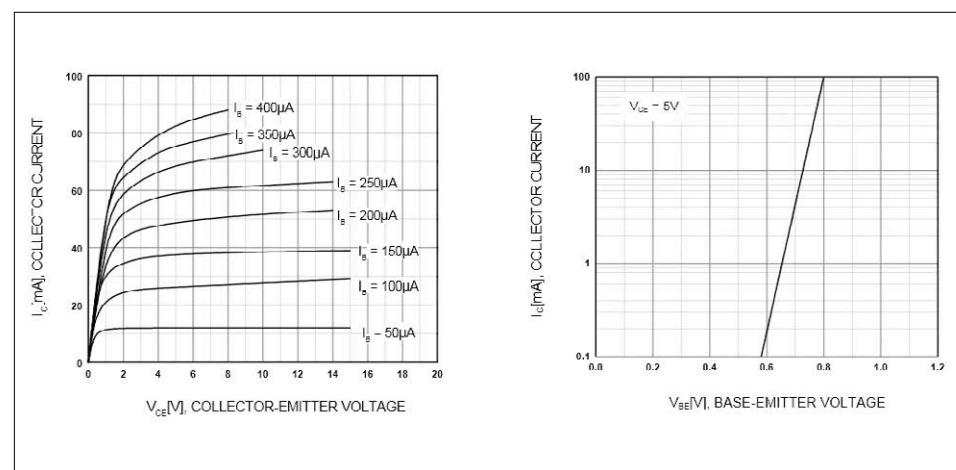


Figure 2 : Caractéristiques clés du BC238B (source : Fairchild SemiConductor)

En pratique si la tension entre base et émetteur est inférieure à ce seuil, le transistor est bloqué : aucun courant ne circule dans le collecteur. Dès que cette tension atteint le seuil de 0,6 V, le transistor devient passant, mais vous ne pourrez pas augmenter cette tension significativement au-dessus de 0,6 V. Le transistor fonctionnera alors en **amplificateur de courant**. Par exemple si vous avez un transistor avec un gain de 100 et injectez un courant de 1 mA dans la base, il circulera un courant de 100 mA par le collecteur. L'intensité totale du courant à travers le collecteur sera la somme des deux courants, soit 101 mA. En réalité, cette explication est approximative, en raison de la complexité physique du transistor. Si vous êtes intéressés par plus de détails, recherchez « modèle d'Ebers-Moll » sur Wikipedia.

Pour rester concrets, prenons l'exemple d'un bon vieux transistor NPN, le fidèle BC238B de *Fairchild Semiconductors*. Tout autre NPN serait utilisable (2N2222, 2N3904 voire BC847 si vous préférez les CMS). Comme ils sont à peu près interchangeables, Elektor les appelle des TUN, pour *transistor universel NPN*. Les principales caractéristiques du BC238 sont reproduites en **figure 2**. Le graphique à gauche illustre la relation entre tension collecteur/émetteur (V_{CE}) et courant de collecteur (I_C), et ce pour différents courants de base (I_B). Regardez par exemple la courbe pour $I_B=200 \mu A$. Dès que la tension V_{CE} dépasse 2 ou 3 V,

Revenir aux bases,
ça peut mener loin.

le courant circulant par le collecteur est presque constant, autour de 50 mA. Cela signifie bien sûr que le gain de ce transistor est de 250 (50 mA divisé par 200 μA). Le graphique de droite illustre la tension de base V_{BE} , qui n'est pas strictement également à 0,6 V, mais en reste proche.

Important : Le gain des transistors est tout sauf précis. En sortie d'usine, la dispersion sur leur gain réel est énorme d'un exemplaire à l'autre. Pire, ce gain change en fonction des conditions de fonctionnement du transistor et en particulier de la température de jonction. Pour le BC238B, il varie de 180 à 460 ! Une réalité que le concepteur devra prendre en compte.

Polarisation ?

Commençons à jouer avec un BC238B. Simuler un transistor est un jeu d'enfant avec l'un des nombreux outils de calcul numérique de type Spice. J'utilise VSM, la version de Spice incluse dans la suite de CAO Proteus (Labcenter). Sur le montage simple d'un BC238B en émetteur commun (**fig. 3**), le collecteur est raccordé à une source d'alimentation +10 V CC via une résistance de 1 k Ω , tandis qu'une résistance de 1 M Ω relie la base à cette même alimentation de +10 V. Comme la tension appliquée à la base est de 10 V, largement supérieure au seuil de 0,6 V,

le transistor est passant et cette tension restera aux environs de 0,6 V : la valeur simulée est d'ailleurs de 0,66 V. Le courant dans la base peut se calculer simplement par la loi d'Ohm : $I = U/R$, soit $(10 V - 0,66 V)/1 M\Omega = 9,34 \mu A$.

Le courant dans le collecteur est obtenu en multipliant cette valeur par le gain du transistor. La simulation donne 2,9 mA, soit un gain de 310 pris en compte par le simulateur. Comment calculer la tension du collecteur ? La chute de tension aux bornes de la résistance du collecteur est aussi donnée par la loi d'Ohm : $U=R \times I = 1 k\Omega \times 2,9 \text{ mA} = 2,9 \text{ V}$. Sous 10 V de tension d'alimentation, la tension entre collecteur et masse doit donc être de $10 - 2,9 = 7,1 \text{ V}$, comme le confirme la simulation.

Imaginons maintenant que vous utilisez ce BC238B pour amplifier un signal alternatif, par exemple pour faire un petit amplificateur audio. C'est simple : partez du schéma de la figure 3, et ajoutez le signal alternatif sur la base du transistor. Ce signal supplémentaire augmentera ou diminuera le courant continu déjà appliqué sur la base par la résistance de 1 M Ω , et ces fluctuations se retrouveront sur la tension de collecteur, amplifiées par le gain du transistor.

Comment concevoir un tel ampli ? Il faut tout d'abord fixer ce qu'on appelle le point de fonctionnement du transistor, c'est-à-dire qu'il faut définir les tension et courant continus lorsqu'aucun signal n'est appliqué à l'entrée. Commencez par fixer la valeur de la résistance de collecteur en fonction de l'impédance de sortie sou-

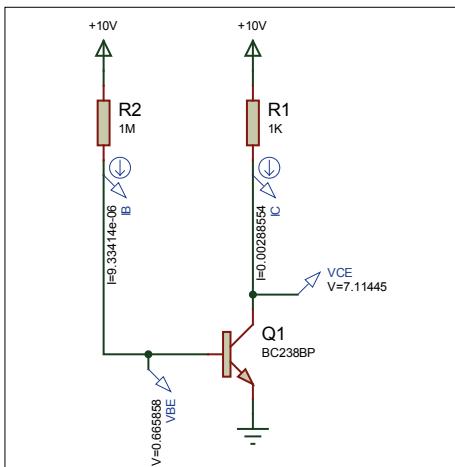


Figure 3 : Sur cette simulation, l'intensité du courant de collecteur est 310 fois plus forte que celle du courant de base. La tension de base reste, comme on s'y attend, de l'ordre de 0,6 V.

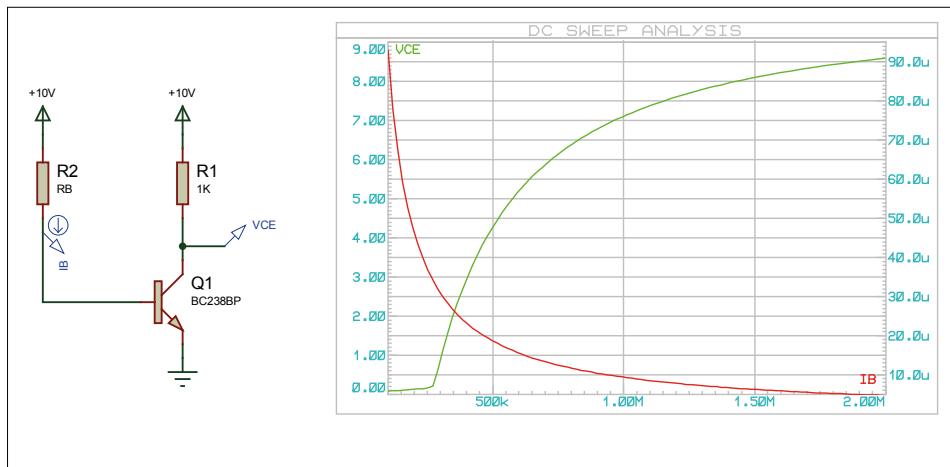


Figure 4 : Cette simulation montre la tension de collecteur (vert) et de base (rouge), en fonction de la valeur de la résistance de base. À une certaine valeur intermédiaire, ici près de 560 k Ω , la tension de collecteur est proche de $V_{cc}/2 = 5 \text{ V}$.

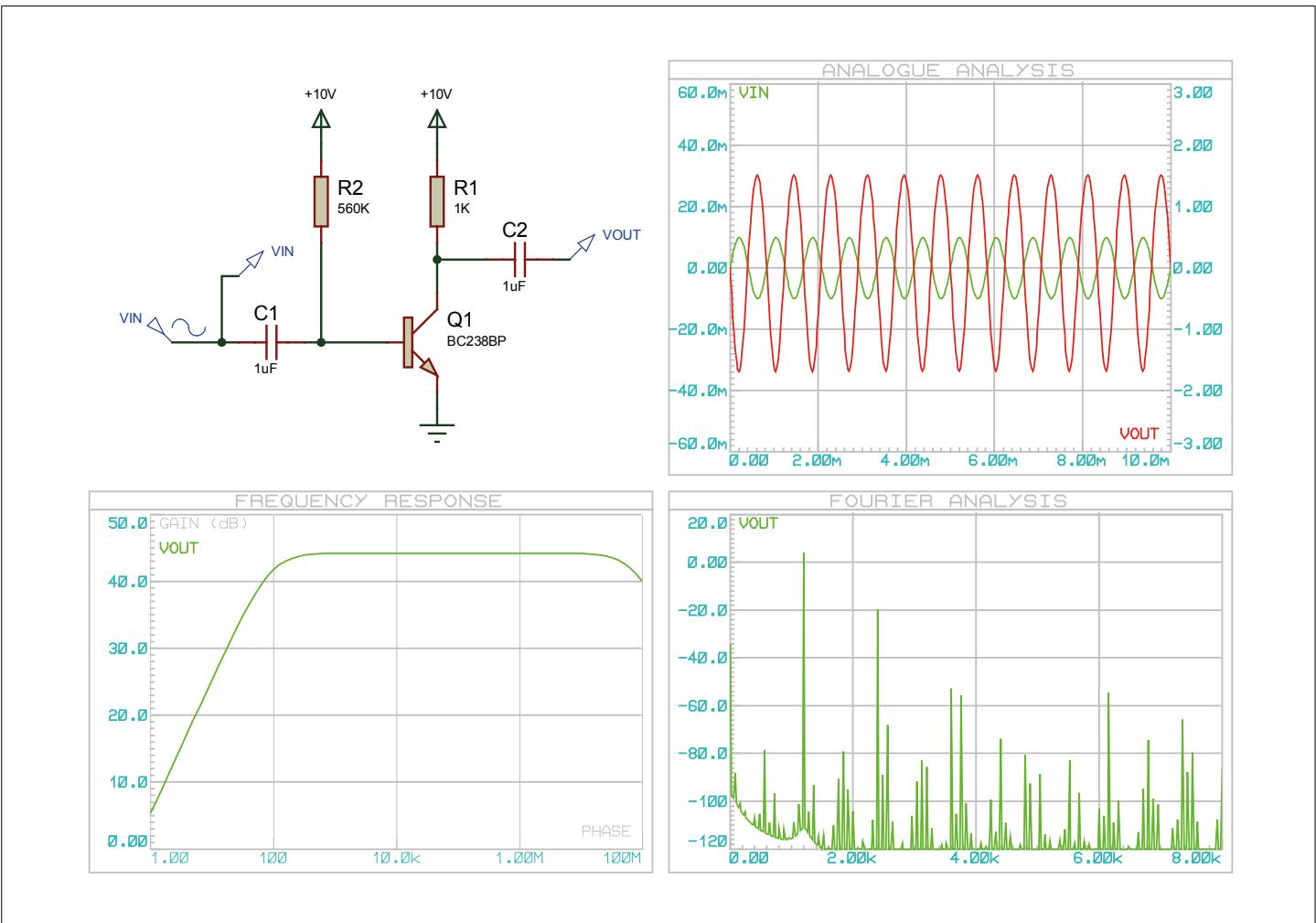


Figure 5 : Un amplificateur à polarisation fixe. Un signal alternatif est injecté sur la base via un condensateur. La simulation temporelle montre que la tension de sortie est de $\pm/1,6\text{ V}_{\text{pp}}$ avec $\pm/10\text{ mV}$ en entrée, avec une bande passante de 100 Hz à 10 MHz et au-delà. La distorsion harmonique reste de l'ordre de 1 % (harmonique 2 à -25 dBc).

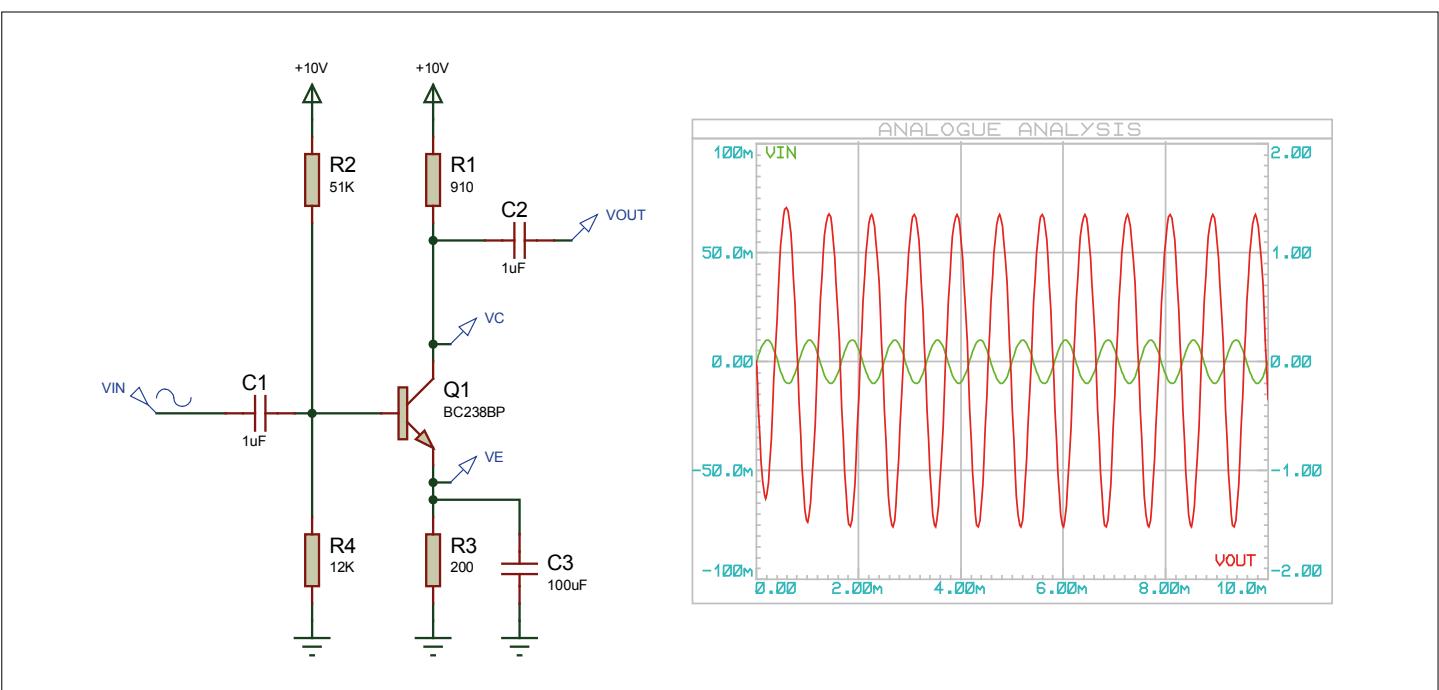


Figure 6 : Une polarisation stabilisée par l'émetteur nécessite deux résistances et un condensateur supplémentaires.

haitée, disons $1\text{ k}\Omega$. Calculez ensuite la résistance de polarisation de la base (c'est-à-dire la résistance entre base et alimentation), de sorte que la tension continue sur le collecteur soit approximativement égale à la moitié de la tension d'alimentation. Cela permet d'avoir la plus grande dynamique possible. J'ai fait une petite simulation montrant la tension continue sur le collecteur en fonction de la valeur de la résistance de base (**fig. 4**) : une résistance de base d'à peu près $560\text{ k}\Omega$ répond bien au besoin et donne une tension continue de sortie de $10/2 = 5\text{ V}$. Avec cette valeur, un courant continu circule en permanence dans le transistor, son intensité est de l'ordre de $I = U/R = 5\text{ V}/1\text{ k}\Omega = 5\text{ mA}$. Un tel amplificateur est dit de classe A.

Nous avons un transistor correctement polarisé en continu sur la base duquel nous pouvons injecter le signal alternatif à travers un condensateur de découplage qui bloque toute composante continue. Le signal de sortie pourra de même être récupéré sur le collecteur via un autre condensateur de découplage (**fig. 5**). La capacité de ces condensateurs est directement liée à la fréquence la plus basse que vous voulez amplifier. Vous pouvez la calculer (sachant que l'impédance d'un condensateur est $Z = \frac{1}{2\pi f C}$), ou la simuler. Ici $1\text{ }\mu\text{F}$ permet d'avoir une fréquence de coupure basse de l'ordre de 100 Hz.

Un simulateur de type Spice permet de tracer la courbe de réponse en fréquence et de faire des graphes temporels comme avec un oscilloscope. Comme illustré en figure 5, avec 20 mV de signal d'entrée, notre petit circuit délivre 3,2 V crête-crête. Il s'agit donc bien d'un amplificateur. Son gain de tension, ici de 160, n'est pas identique au gain en courant du transistor (310). Le gain en tension est en fait toujours inférieur au gain h_{FE} . La relation entre les deux n'est pas directe (cherchez *Hybrid Pi Model* sur l'internet si vous souhaitez plus d'explications).

Stabilisation par l'émetteur

Nous avons un amplificateur avec juste un transistor NPN, deux résistances et deux condensateurs. Est-ce suffisant ? Rarement, à cause de la grande dispersion du gain des transistors. À moins de trier les transistors après en avoir mesuré individuellement le gain et de vous assurer que la température restera stable, vous n'aurez qu'un amplificateur au gain en tension plus qu'incertain. Il pourrait même ne pas marcher du tout. Pourquoi ? Imaginez que vous remplacez le transistor par un autre exemplaire ayant un gain deux fois supérieur. La même résistance de base donnera un courant dans le collecteur d'intensité deux fois plus forte. La chute de tension continue dans la résistance du collecteur sera donc deux fois plus grande, proche de 10 V au lieu

de 5 V, et l'amplificateur n'amplifiera plus, au mieux, que les alternances négatives du signal alternatif d'entrée !

Il faut améliorer ce schéma en stabilisant le gain de l'amplificateur, c'est-à-dire à le rendre moins dépendant du paramètre h_{FE} du transistor. La méthode la plus simple, appelée stabilisation par l'émetteur (fig. 6), nécessite deux résistances et un condensateur supplémentaires. Une résistance est ajoutée entre émetteur et masse, avec en parallèle un condensateur de forte capacité. Le but est de déplacer la tension continue de l'émetteur un peu en dessus 0 V, typiquement vers 1 V. Une autre résistance est alors ajoutée entre la base et 0 V. On verra les calculs dans une minute, mais intuitivement que se passe-t-il si le gain du transistor augmente, quelle qu'en soit la raison ? Le courant circulant par le collecteur (et par l'émetteur) augmentera. Donc la chute de tension causée par la résistance entre émetteur et masse augmentera et la tension de l'émetteur par rapport à la masse montera. Mais, attendez, la tension de polarisation de la base est fixée par deux résistances entre 0 V et 10 V. Donc, si la tension de collecteur augmente, la différence de tension entre base et émetteur diminue. Et ça réduira l'intensité du courant coulant par la base, qui, à son tour, réduit le courant de collecteur, compensant ainsi l'augmentation

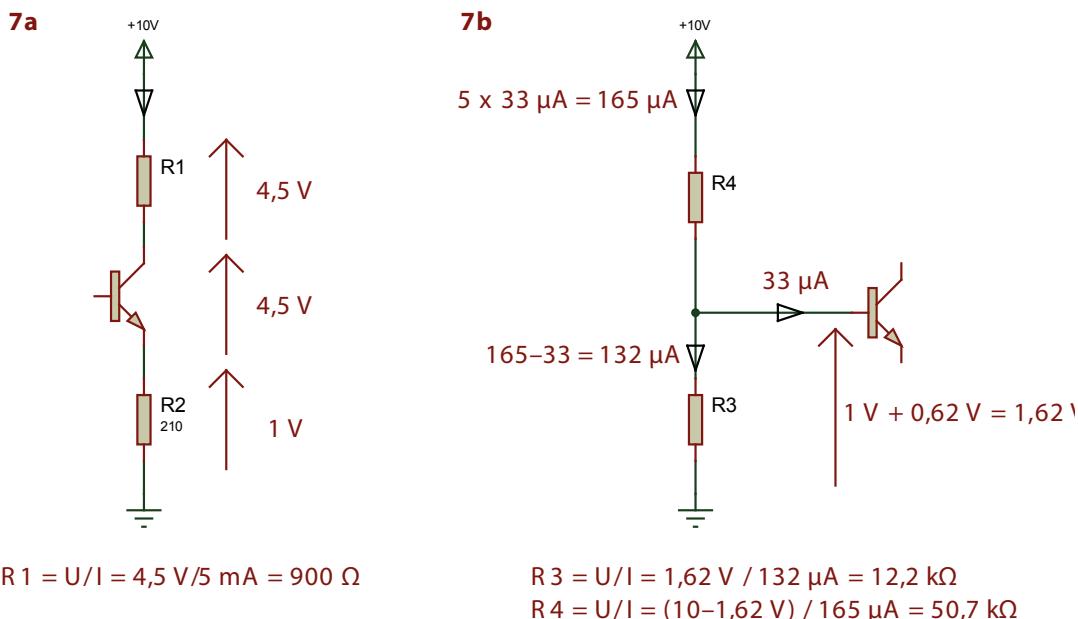


Figure 7 : Voici comment calculer la résistance de collecteur (7a) et les résistances du pont de base (7b) dans le cas d'une stabilisation par l'émetteur.

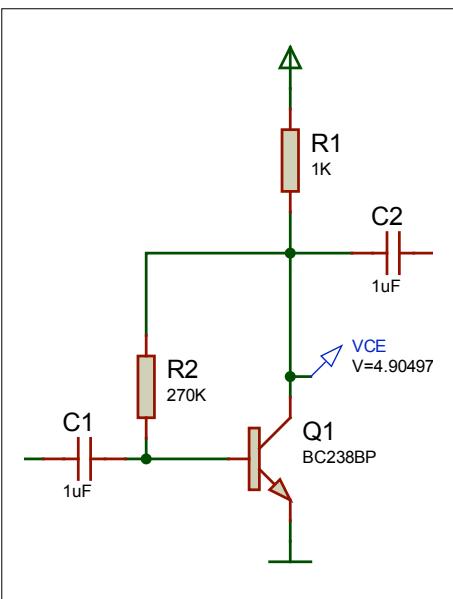


Figure 8 : La variante stabilisée par le collecteur, sans augmentation du nombre de composants.

du gain du transistor. C'est une sorte de stabilisation de gain automatique !

Comment calculer tout ça ? Commencez par déterminer une résistance d'émetteur pour avoir 1 V de chute de tension (**fig. 7a**). Rappelez vous que 1 kΩ sur le collecteur donneait une chute de tension de 5 V, il faut donc sur l'émetteur une valeur de résistance grossièrement cinq fois plus faible. J'ai utilisé la valeur normalisée la plus proche, soit 210 Ω. Il faut ensuite réduire un peu la résistance de collecteur pour compenser ce 1 V perdu et garder un courant moyen de 5 mA, ce qui donne ici 910 Ω.

Le calcul des deux résistances de base est un peu plus complexe. Un point de départ raisonnable est de fixer le courant circulant dans ces deux résistances égal à cinq fois le courant circulant dans la base, soit ici $5 \times 33 = 165 \mu\text{A}$. Une application des lois de Thévenin et d'Ohm permet alors de s'en sortir (**fig. 7b**). Ici j'ai trouvé respectivement 51 kΩ et 12 kΩ. Enfin, la capacité du condensateur entre émetteur et masse doit être assez grande. Vous pouvez le simuler, ou prendre comme

moi un 100 μF.

Quel est le gain en tension de cet amplificateur à gain stabilisé ? La simulation donne 130 (fig. 6), un peu plus faible que les 160 obtenus sans stabilisation, mais c'est le prix à payer pour une marge de tranquillité !

Et par le collecteur ?

La stabilisation par l'émetteur est simple et efficace. Une autre solution présente l'avantage de ne nécessiter aucun composant de plus que la version non stabilisée : c'est la stabilisation par le collecteur (**fig. 8**). Elle consiste à polariser la base avec une résistance connectée non pas à l'alimentation mais au collecteur. Si le gain réel du transistor augmente, le courant de collecteur augmentera et la tension de collecteur diminuera. Comme ici la base est polarisée par cette tension, le courant de base diminuera, stabilisant ainsi le gain de l'amplificateur. Astucieux, non ?

Côté calcul, la démarche est apparentée à celle du circuit non stabilisé, la valeur de la résistance de base peut être calculée par la loi d'Ohm $R = U/I$, avec ici $U = (V_{cc}/2)/I_B$. On trouve 270 kΩ. Par rapport à la stabilisation par l'émetteur, cette solution a le double inconvénient d'un gain en tension inférieur et d'une stabilisation moins bonne, mais dans certains cas, c'est acceptable.

Armé de mon simulateur, j'ai comparé les performances en stabilité des trois solutions, en remplaçant simplement le BC238B par un 2N2222 (dont le gain est significativement plus faible). Avec une polarisation fixe, la tension continue sur le collecteur s'est décalée de 5 V à 6,6 V et le gain en tension a été réduit de 156 à 105 (-32 %). Une stabilisation par l'émetteur apporte comme prévu un énorme gain de stabilité (-5,7 % seulement). Enfin, avec la stabilisation par le collecteur, le gain a chuté de 15 %, performances intermédiaires donc, mais avec trois composants de moins.

Rien de neuf sous le soleil

Cet article pourrait avoir été écrit il y a soixante ans, sauf qu'en 2015, bien des électroniciens semblent avoir oublié qu'un transistor peut avantageusement remplacer un amplificateur opérationnel. L'économie réalisée n'est pas négligeable, mais elle n'est pas le seul avantage. Par exemple, ma société a récemment développé pour l'un de nos clients un système de détection d'alarme où un signal piézo-électrique devait être amplifié avant détection, avec des contraintes de consommation drastiques associées à une durée de vie de plusieurs années sur pile bouton. Devinez quelle a été la meilleure solution : un transistor, grâce à ses courants de polarisation très faibles !

J'espère que cet article, en expliquant de façon limpide le rôle des quelques composants passifs autour du transistor monté en émetteur commun, vous aura permis de mieux comprendre le fonctionnement d'un amplificateur simple à transistor. Pour consolider cet acquis, je vous recommande, comme toujours, d'EX-PE-RI-MEN-TER : téléchargez un simulateur Spice, par exemple l'excellent et gratuit LTSpice de Linear Technology, et reproduisez les exemples de cet article. Modifiez les valeurs et analysez ce qui se passe. Ensuite vous sortirez le fer à souder pour vérifier que vos simulations correspondent à la vraie vie !

Dans mon prochain article et dans le même esprit, je reviendrai sur les différentes classes d'amplificateurs. ▶

Cet article a été publié en anglais dans la revue Circuit Cellar (n° 279).

(150403)

Sources et liens

Proteus, Labcenter, www.labcenter.co.uk

BC238B BJT transistor, Fairchild Semiconductor

http://en.wikipedia.org/wiki/Bipolar_junction_transistor

www-personal.engin.umd.umich.edu/~fmeral/ELECTRONICS%20I/004x%20BJT%20/0041%20BJT%20Bias.pdf

bienvenue dans la section CRÉER

Clemens Valens, Elektor Labs

PARTAGER

CRÉER

DÉCOUVRIR



Des licences tout sauf poétiques

Les licences sont un sujet compliqué. Si, comme la majorité des produits commerciaux, vos projets sont fermés, alors ils sont a priori protégés par la législation internationale en matière de droit d'auteur et de commerce.

Aussi complexe soit-elle, il existe en effet une jurisprudence qui précise ce qui peut être fait et ne pas l'être. Là où les choses se compliquent vraiment, c'est lorsque vous souhaitez mettre sur le marché une création, qu'elle soit matérielle, logicielle ou les deux, sous licence *open source*. Le problème vient de la nature même du mouvement *open source*.

Une licence *open source* permet de modifier l'objet de la licence, ce qui souvent se traduit par l'apparition de nombreuses œuvres « dérivées », toutes légèrement différentes. Cette « dérive » se répercute sur les licences elles-mêmes : il en existe tellement qu'il est pratiquement impossible de trouver celle qui convient parfaitement. Les licences sont des documents légaux qui peuvent servir devant un tribunal, c'est pour cette raison qu'elles sont très difficiles à comprendre.

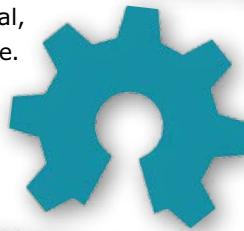
Et avec ça elles ne cessent d'évoluer. Vous souhaitez léguer votre œuvre à la communauté en lui accolant quelque licence *open source* ? Attention au marais de charabia juridique dans lequel vous risquez de vous embourber. Et oubliez tout badin : « C'est gratuit, faites-en ce que vous voulez, je décline toute responsabilité », ce type de formulation serait sans valeur lors d'un procès, toujours possible si un zigoto fait quelque chose d'épouvantable avec votre œuvre.



Open Source
Initiative

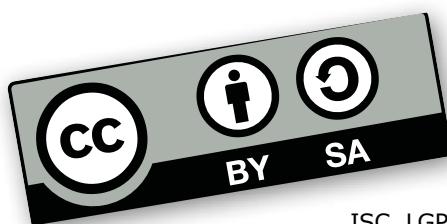


Free as in Freedom



open source
hardware

Une liste inutile et incomplète (mais bien classée) de licences



AAL, AFL-3.0, AGPL-3.0, Apache-2.0, APL-1.0, APSL-2.0, Artistic-2.0, BSD-2-Clause, BSD-3-Clause, BSL-1.0, CATOSL-1.1, CC BY, CC BY-NC, CC BY-NC-ND, CC BY-ND, CC BY-SA, CC-BY-NC-SA, CDDL-1.0, CECILL-2.1, CERN OHL, CNRI-Python, CPAL-1.0, CU-AOPL-1.0, ECL-2.0, EFL-2.0, Entessa, EPL-1.0, EUDataGrid, EUPL-1.1, Fair, Frameworkx-1.0, GPL-2.0, GPL-3.0, HPND, IPA, IPL-1.0, ISC, LGPL-2.1, LGPL-3.0, LPL-1.02, LPPL-1.3c, MirOS, MIT, Motosoto, MPL-2.0, MS-PL, MS-RL, Multics, NASA-1.3, Naumen, NCSA, NGPL, Nokia, POSL-3.0, NTP, OCLC-2.0, OFL-1.1, OGTS, OSL-3.0, PHP-3.0, PostgreSQL, Python-2.0, QPL-1.0, RPL-1.5, RPSL-1.0, RSCPL, SimPL-2.0, Sleepycat, SPL-1.0, TAPR, UPL, VSL-1.0, W3C, Watcom-1.0, WXwindows, Xnet, Zlib, ZPL-2.0. ▶



(150332-I – version française : Hervé Moreau)

Bienvenue dans Elektor Labs

C'est dans Elektor Labs que les projets grands et petits, analogiques et numériques, vieux jeu et *méga cool* prennent forme pêle-mêle pour vous permettre de les réaliser à votre goût.

Notre offre : La célébrité



La plupart des électroniciens créatifs sont modestes. Trop discrets. Ce n'est pas parce qu'elle est griffonnée sur un rond de bière qu'une trouvaille épataante ne mériterait pas l'attention. Elektor Labs vous aide à affiner le fruit de vos études jusqu'à la perfection. Notre rédaction et nos illustrateurs donneront à votre création l'ampleur éditoriale qu'elle mérite, mais c'est toujours votre nom qui figurera en tête de l'article, car c'est *vous* l'auteur du projet. En plus, vous serez rémunéré honorablement, même si le plus grand honneur restera celui d'être publié en plusieurs langues et lu dans le monde entier. Auteurs de livres, de blogs ou de vidéos, vous ne serez pas moins bien traités. Étudiant(e)s et jeunes électronicien(ne)s, une publication dans une revue comme Elektor ne déparera pas votre C.V. !

Notre histoire

Elektor Labs existe depuis les années 1970. En ce temps-là, c'est la même personne qui soudait les circuits et écrivait les articles. Le labo n'a pas seulement vu arriver et passer le transistor, le circuit intégré, le microprocesseur et les composants montés en surface, mais il s'est toujours porté à l'avant-garde des nouvelles vagues technologiques pour diffuser la bonne parole et les rendre accessibles au plus grand nombre.

Notre équipement

Il y a l'eau courante, le gaz, l'électricité et des machines à café. Trois des plus belles pièces du château d'Elektor sont réservées au labo, mais nous avons bien du mal à ne pas envahir le reste.

Nos paillasses débordent de cartes, de composants en vrac, d'accessoires mécaniques, de prototypes et d'un fatras ingérable. L'appareillage lourd est heureusement dans un local séparé.

Nos produits

Nos produits sont dans le magazine et sur nos sites. Nous produisons texte et illustrations utilisés par la rédaction pour assembler les articles, mais surtout les circuits imprimés, certains assemblés prêts à l'emploi, ainsi que le logiciel et les composants programmés, des kits, des modules, des outils, des accessoires, des vidéos et un service d'info technique.

375

propositions

52

études en cours

168

projets achevés

591

projets en tout



Nos principes

Les réalisations et les produits sortis des tuyaux de Labs répondent tous à des exigences sévères. Les produits des articles du magazine estampillés LABS doivent fonctionner avec l'appareillage étalonné disponible au labo. La correspondance entre schéma et liste de composants doit être parfaite. Les kits sont soumis à des tests par échantillonnage périodique. Nous suivons la directive ROHS et les autres normes de sécurité applicables à notre situation. Les erreurs constatées font l'objet d'une publication.

Nos webinaires

Autrefois nos ingénieurs les plus bavards testaient leurs prototypes en discutant. À la vue d'un micro, ils se taisaient. Pour les faire parler sur elektor.tv, il a fallu un talent de reporter. Elektor Labs présente aussi des webinaires où l'on parle bien et beaucoup. Ils sont annoncés dans notre lettre d'information Elektor.POST!

Nos experts et nos concepteurs

L'équipe est formée d'électroniciens expérimentés de tout plumage qui, avec ou sans prestigieux diplômes, cumulent quelque 200 années d'expérience en électronique. En plus, Labs est au cœur d'un réseau d'experts consultés ponctuellement, notamment quand ça coince.

elektor e-labs

Sharing Electronics Projects

Home Proposals In Progress Finished

Search

Log In

Plus on est de fous d'électronique, plus on rit !

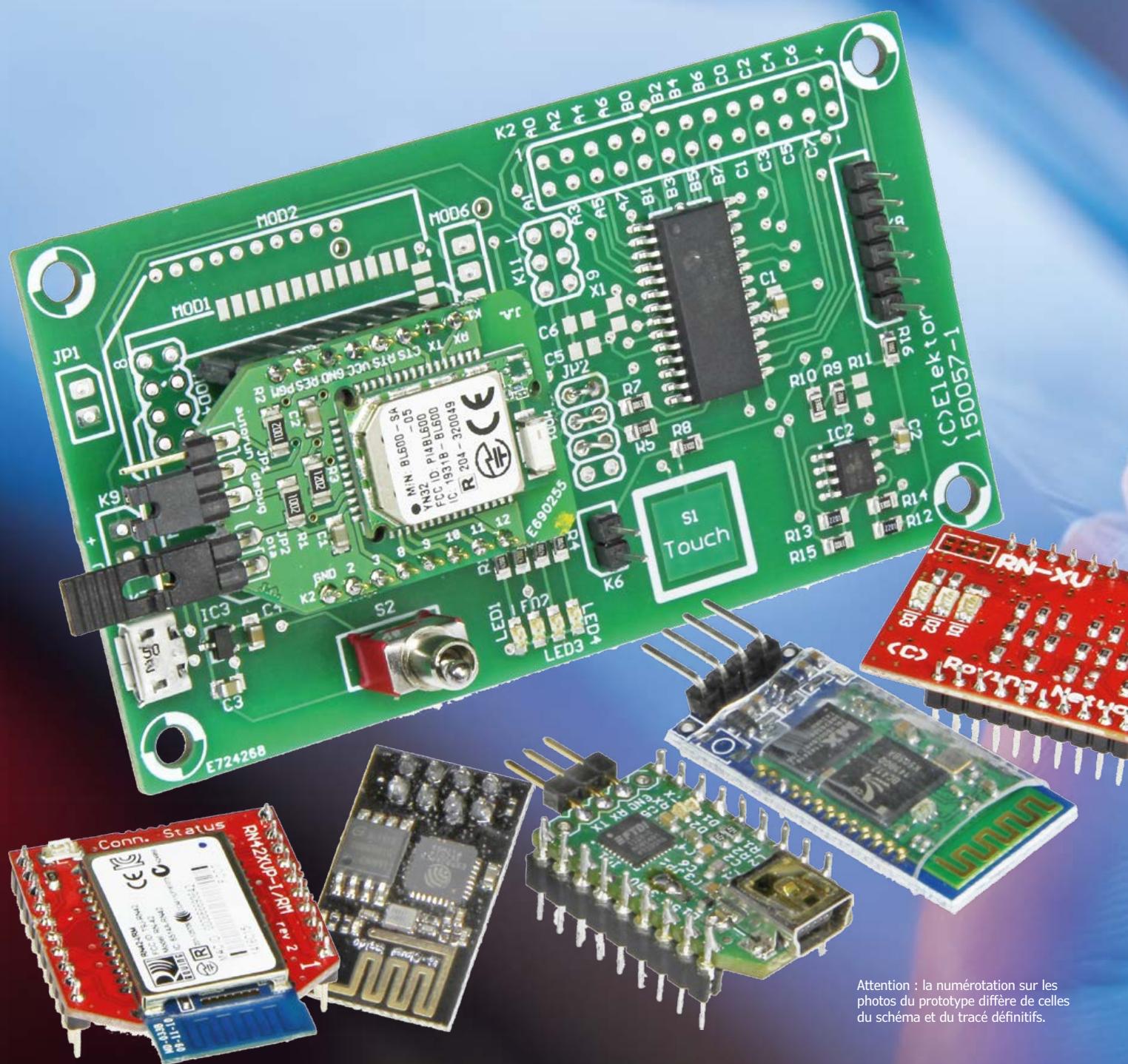
Notre site communautaire elektor-labs.com est le port I/O bidirectionnel idéal pour mettre en vedette **votre projet** et pour suivre ceux des autres. Il est partagé par des milliers d'électroniciens comme vous. Avec eux et comme eux, ouvrez votre atelier personnel sur le monde. En circulant, vos idées s'enrichiront, vos circuits s'amélioreront. Les projets les plus suivis sur www.elektor-labs.com sont mûris par l'équipe d'Elektor-Labs et certains deviennent des articles (rémunérés !) dans le magazine.

Read/Write ?

Si vous souhaitez publier sous votre nom et pour votre bénéfice un projet dans ce magazine, en quatre langues et avec la présentation soignée d'Elektor, pour être lu par des dizaines de milliers d'électroniciens dans le monde, **rejoignez la communauté avec une carte de membre GREEN ou GOLD** (www.elektor.fr/membres). Les membres peuvent publier des projets sur le site elektor-labs.com, les autres se contentent de regarder.

carte d'E/S pour Android

l'électronique embarquée,
pilotez-la depuis votre téléphone portable



Attention : la numérotation sur les photos du prototype diffère de celles du schéma et du tracé définitifs.

Elbert Jan van Veldhuizen (Pays-Bas)

Avec son énorme puissance de calcul à un prix abordable, son écran tactile et ses multiples possibilités de communication sans fil, un appareil Android est tout indiqué pour commander toutes sortes de circuits électroniques. Ajoutez-lui 22 lignes d'E/S et vous pourrez à distance commuter facilement, réaliser des mesures, lancer des moteurs, recevoir des signaux analogiques et en créer, bien d'autres choses encore. Ce premier article expose le matériel nécessaire.



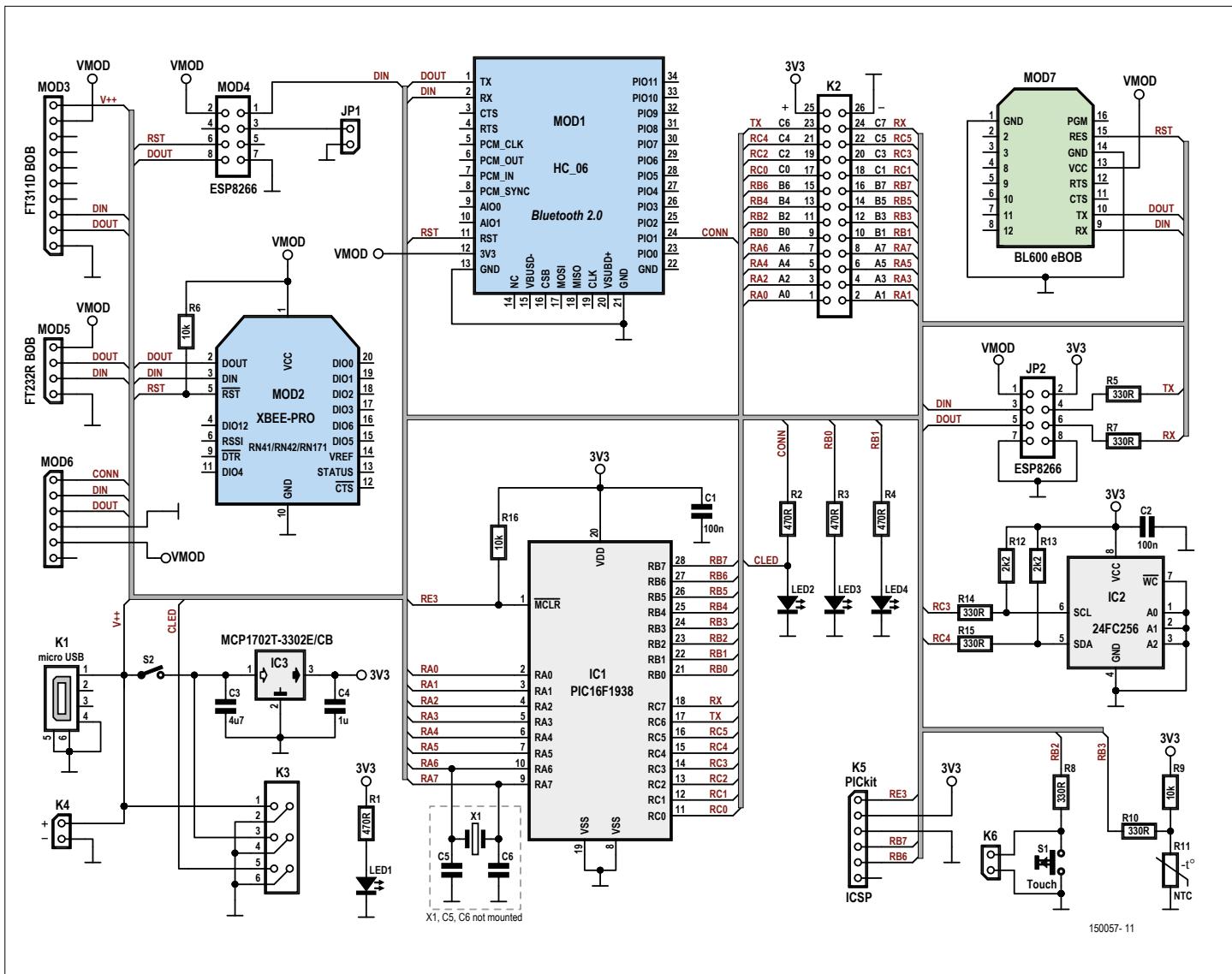


Figure 1. Schéma de la carte d'E/S. En pratique, il y a beaucoup moins sur la carte, puisqu'on n'y installe qu'un seul module de communication.

Tableau 1. Aperçu des caractéristiques de la carte d'E/S pour Android

fonction	nb de broches (max 22) / valeur	remarques
entrées numériques	22	dont 8 broches à résistance de rappel vers le haut « molle » en option
sorties numériques	22	I_{sortie} max 25 mA
C A/N	11	Résolution de 10 bits, env. 20 Ké/s
interrupteurs sensitifs	8	dont 7 broches partagées avec le C A/N
MLI	8	par logiciel, 4 x 2 en mode en pont
compteur	à 16 bits	maximum 8 MHz
débit binaire	max 100 tâches par seconde	liaison à 9600 bauds
mémoire flash de données	32 Ko	sur 25FC256
suppléments sur carte	4 LED, 1 CTN, 1 touche tactile	
tension d'alimentation	4 à 13 V non stabilisés 3,3 V stabilisés	
consommation (sans charge)	max 30 mA (Bluetooth) max 250 mA (Wi-Fi) max 5 mA (USB) plus le courant pour le module USB	
connexion d'alimentation	micro-USB, connecteur encartable	

Nous allons décrire une carte et un exemple d'appli pour échanger des informations sur 22 lignes d'entrée/sortie au moyen de commandes simples. La carte est capable de communiquer avec un téléphone tactile ou une tablette par Bluetooth, Wi-Fi ou USB. Ses ports peuvent traiter des entrées et des sorties numériques, mais aussi servir de C A/N, utiliser une touche capacitive, moduler des impulsions ou les compter. Son micrologiciel est aisément modifiable, on ajoute ainsi des fonctions dans le microcontrôleur pour des applications en temps réel, le contrôleur est pourvu d'un chargeur d'amorçage à cet effet. Pour commander cette carte d'E/S pour Android, j'ai développé des classes Java (des bibliothèques) aux dénominations parlantes. Avec elles, vous n'aurez pas le souci des détails de la programmation des fonctions de l'appli.

Ce premier article dévoile le circuit imprimé et les fonctions du microcontrôleur. Dans les épisodes suivants, nous verrons les possibilités du Bluetooth, du Wi-Fi et du port USB, ainsi qu'une appli pour piloter la carte d'E/S Android. Nous étudierons ensuite comment réaliser un enregistreur de données, adapter le micrologiciel et le transférer avec le chargeur d'amorçage. Avant de fouiller dans le matériel, voyons ce qu'il peut faire en pratique. On sait déjà qu'il met à disposition 22 broches d'E/S. La carte porte aussi un interrupteur à touche capacitive, soudé à demeure, quatre LED et une résistance CTN. C'est en tout cas un circuit imprimé pratique pour acquérir de l'expérience. Le **tableau 1** détaille les caractéristiques de la carte d'E/S Android avec les différentes options d'alimentation.

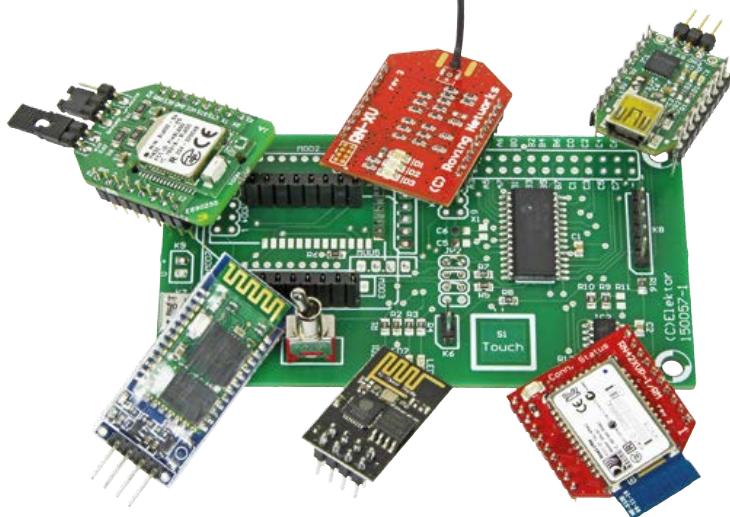
Le circuit

Le schéma de la carte est à la **figure 1**. Le nombre de composants réellement présents sur le circuit imprimé est bien moindre que ce que le schéma présente. Il y a en effet de nombreux modules de communication qui peuvent y prendre place. Parmi eux, la plupart des modules pour le Bluetooth sont constitués de cartes de liaison (*Break out Board*, BoB) : e-BoB BL600 (avec Bluetooth V4, réf. 140270-91, ici Mod7), BoB (Mod1) avec module HC-06 (avec Bluetooth V2), carte à embase avec HC-06 (Mod6), carte avec RN41 au format Zigbee (Mod2), carte avec HC-06 au format Zigbee (Mod2). Comme modules Wi-Fi, on peut y mettre aussi bien un RN171-XV (au format ZigBee) qu'un ESP8266 (à embase). Il y a aussi des connecteurs pour brancher l'e-BoB Android d'Elektor (130516-91) et la passerelle Elektor e-BoB FT232R USB/série (110553-91), tous deux pour une liaison USB avec un PC, une tablette ou un téléphone tactile (par câble USB-OTG).

Tous ces modules fournissent des données sérielles qui transittent par l'appareil Android vers un PC. Elles passent par un microcontrôleur PIC, ici un PIC16F1938 avec pas moins de 24 broches d'E/S, chargé d'un micrologiciel qui interprète les commandes pour configurer ces broches de manière à y envoyer des données numériques ou y recevoir des signaux en analogique ou numérique.

Les lignes d'E/S du microcontrôleur sont toutes accessibles sur K2, une embase à deux rangées de treize picots. Ces généralités exposées, allons voir de près les composants dans le schéma.

Partons du régulateur de tension de 3,3 V, IC3 sur la carte. Pour lui fournir du courant, il y a là deux options : un connecteur K1 micro-USB ou deux pastilles à souder, voire une embase



K4. La puce IC3 est un MCP1702-33, un régulateur à faible déchet qui délivre la tension de 3,3 V au module Bluetooth ou Wi-Fi et au reste du circuit. C'est l'interrupteur S2 qui met sous tension. Si vous n'en voulez pas, remplacez-le par un pont de câblage sur les deux pastilles de gauche.

Le microcontrôleur PIC16F1938 de Microchip (IC1) totalise 28 broches. La liaison entre lui et les modules Bluetooth, Wi-Fi ou USB passe par le bloc de cavaliers JP2 pour le débogage et les résistances de sécurité R5 et R7. Toutes les lignes d'E/S sont disponibles sur le connecteur K2.

Publicité

Schaeffer AG

Designer de faces gratuit

CONCEVEZ – NOUS PRODUISONS

Des plaques de qualité professionnelle

Dès une pièce et pour un prix modéré !
Téléchargez notre Designer de faces avant gratuitement sous www.schaeffer-ag.de, concevez votre plaque puis commandez-la directement.

www.schaeffer-ag.de

Liste des composants

Résistances :

R1, R2, R3, R4 = 470 Ω 1%, CMS 0805
 R5, R7, R8, R10, R14, R15 = 330 Ω 1%, CMS 0805
 R6, R9, R16 = 10 k Ω 1%, CMS 0805
 R11 = CTN 10 k Ω CMS 0805 (p.ex. Vishay NTCS0805E3103JHT)
 R12, R13 = 2,2 k Ω 1%, CMS 0805

Condensateurs :

C1, C2 = 100 nF/25 V, 10%, CMS 0805
 C3 = 4,7 μ F/25 V, 10%, CMS 0805
 C4 = 1 μ F/25 V, 10%, CMS 0805

Semi-conducteurs :

LED1, LED2, LED3, LED4 = LED verte faible courant, CMS 0805
 IC1 = PIC16F1938-I/SO, SOIC28 (programmé : 150057-41*)
 IC2 = 24FC256-I/SN, SOIC8
 IC3 = MCP1702T-3302E/CB, SOT23A

Divers :

K1 = connecteur micro-USB 2.0 type B, CMS
 K2 = embase à 2x13 picots, pas de 2,54 mm
 K3 = embase à 2x3 picots, pas de 2,54 mm
 K4 = embase à 2 picots, pas de 2,54 mm
 K5 = embase à 6 picots, pas de 2,54 mm
 S2 = commutateur inverseur, pas de 2,54 mm
 circuit imprimé : 150057-1*

ou

circuit imprimé avec tous les CMS déjà soudés : 150057-91*

module de communication (n'en installer qu'un seul) :

Mod1 = module Bluetooth HC-06/HC-05
 à souder directement sur la carte

Mod2 = module Bluetooth HC-06/HC-05 au format ZigBee

Mod2 = module Bluetooth RN41XVC

Mod2 = module WLAN RN171 (avec antenne fouet)
 à monter sur deux embases à 10 picots au pas de 2 mm

Mod3 = e-BoB Android (Elektor, 130516-91*)
 à monter sur une embase à 10 picots au pas de 2,5 mm

Mod4 = module Wi-Fi ESP8266 (raccordement sur 2x4 broches)
 à monter sur embase à 2x3 picots au pas de 2,5 mm

Mod5 = e-BoB passerelle FT232R USB/série (Elektor, 110553-91*)
 à monter sur une embase à 3 contacts au pas de 2,5 mm (embase à 3 picots sur e-BoB)

Mod6 = module HC-06 (version à 6 broches)
 à monter sur embase à 6 contacts au pas de 2,5 mm

Mod7 = e-BoB BL600 (Elektor, 140270-91*)
 à monter sur embase à 8 contacts au pas de 2,5 mm

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

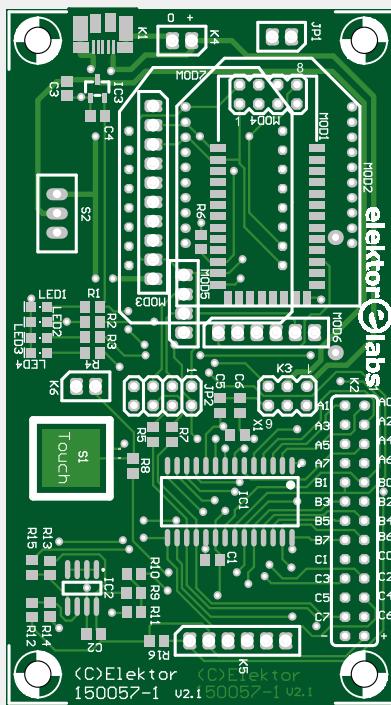


Figure 2. Le circuit imprimé pour la carte d'E/S. Il y a sept options différentes de montage disponibles pour les modules de communication.

Comme mémoire supplémentaire, pour l'enregistrement de données, par exemple, la carte dispose d'une mémoire flash (IC2) série I²C de 32 Ko. Les lignes de bus I²C de cette puce sont protégées par R14 et R15, les résistances de rappel haut sont R12 et R13. On trouve aussi une résistance CTN (R11) pour des mesures de température, elle forme avec R9 un diviseur de tension. Les LED1 à LED4 sont des témoins pour les différents signaux. La première indique la présence de la tension de 3,3 V. LED2 ne travaille que si le module HC-06 est installé. On peut utiliser les LED3 et 4 dans différentes fonctions, par exemple LED3 signale que le chargeur d'amorce est activé. Le capteur capacitif S1 est constitué d'une surface soudée sur le circuit imprimé. Le connecteur K6 est directement en parallèle sur cet interrupteur sensitif, nécessaire à l'activation du chargeur d'amorçage dans le PIC.

Le circuit imprimé est prévu pour recevoir un quartz pour le PIC. Le logiciel du chargeur décrit ici ne s'en occupe pas. Pour le mettre en œuvre, il faut un autre chargeur qui adapte le mot de configuration. Alors, les broches A6 et A7 de K2 ne peuvent plus être utilisées par le micrologiciel.

Un circuit imprimé universel

On pourrait vraiment s'y perdre dans le dessin des pistes de la **figure 2**, avec toutes ces implantations possibles de modules divers, mais à y regarder de plus près, on découvre où se situent précisément les raccordements pour les modules MOD1 à MOD6. Reste à faire le choix de l'unique module désiré qui occupera la place. Exception faite des connecteurs et commutateurs, tout est en CMS, ce qui ne simplifie pas la construction à la main. C'est pourquoi l'e-choppe [1] propose le circuit imprimé 150075-91 sur lequel tous les CMS et les connecteurs sont déjà soudés. Mais comme les différents modules ne seront décrits en détail que le mois prochain, il est préférable de patienter encore un peu.

Il y a sur le circuit imprimé plusieurs connecteurs et embases pour différents usages. Pour la clarté, reportez-vous au **tableau 2** qui en donne un aperçu complet.

En usage normal, JP2 sert au passage des signaux TX et RX entre le PIC et le

Tableau 2. Fonctions des connecteurs et embases

K1	connecteur micro-USB pour alimenter carte et module
K2	embase à 2x13 picots pour 24 lignes d'E/S du PIC, 3,3 V et masse
K3	embase à 2x3 picots pour connecter en chaîne plusieurs cartes d'E/S
K4	connecteur d'alimentation à 2 broches pour carte (4 à 13 V)
K5	connecteur de programmation à 6 broches pour PICkit 2 ou 3
K6	embase à 2 picots pour lancer le chargeur d'amorce par bouton ou par liaison
JP1	embase à 2 picots pour mise à niveau du micrologiciel du module ESP8266
JP2	embase à 2x4 picots pour configurer les lignes de signal et d'alimentation vers les autres modules
MOD1 à MOD7	connecteurs pour raccorder les différents modules Bluetooth / Wi-Fi / USB (cf. liste des composants)

module en place. D'habitude, on relie la broche 3 à la 4 et la broche 5 à la 6, tandis que les broches 7 et 8 (GND) sont déjà interconnectées par une piste, ce qui est pratique quand on veut brancher un module externe sur JP2. Sur la broche 2, on trouve l'alimentation de 3,3 V fournie à la carte par le régulateur de tension, alors que la broche 1 est reliée à la connexion d'alimentation du module. Il y a deux manières d'alimenter le module :

Le module doit être alimenté en 3,3 V. C'est le cas des modules Bluetooth et Wi-Fi. Alors, il faut relier ensemble les broches 1 et 2.

Lors de l'installation d'une passerelle e-BoB FT232R USB/série (110553-91) ou d'un e-BoB Android (130516-91), il faut les alimenter en 5 V par leur connecteur USB et dans cette circonstance, les broches 1 et 2 sur JP2 ne peuvent en aucun cas être reliées entre elles.

Les commandes

Inutile d'attendre que vous ayez une carte d'E/S entre les mains pour vous parler des commandes dont vous aurez à vous servir pour gérer la carte.

Supposons qu'une liaison avec le module (Bluetooth, Wi-Fi ou USB) soit en service. C'est qu'il y a une communication série transparente entre l'appareil Android et le microcontrôleur sur la carte d'E/S pour Android. Là, c'est une appli qui permettra d'envoyer des commandes au microcontrôleur. Le micrologiciel les interprète et agit en conséquence sur les E/S. Ces ports ont été numérotés d'après les dénominations du microcontrôleur : A0 à A7, B0 à B7 et C0 à C7 (cf. aussi **tableau 3**). On dispose de quatre types de commandes :

- **W** pour *Write*, permet d'écrire une valeur donnée sur une broche spécifique. Format : *W broche valeur*. Exemple : *w a3 1* passe la broche d'E/S A3 au niveau haut.
- **R** pour *Read*, permet de lire la valeur sur une broche déterminée. Format : *R broche*. Exemple : *r b4* lit la valeur de la broche B4. Le microcontrôleur répond selon le format : *R broche valeur*, donc dans l'exemple par *R B4 1*.
- **S** pour *Settings*. On détermine ainsi la fonction d'une broche. Format : *S broche fonction*, la fonction est codée par un nombre. C'est 0 pour

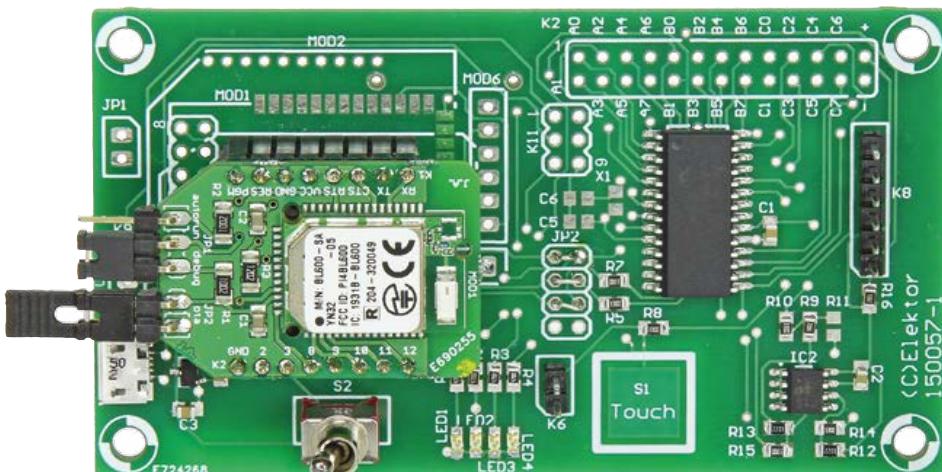


Figure 3. Le circuit imprimé assemblé, l'e-BoB BL600 d'Elektor y est installé.

Tableau 3. Fonctions possibles des lignes d'E/S

ligne	entrée hte impédance (0)	entrée à rappel vers le haut (1)	sortie (2)	C A/N (3)	Cap sense (4)	MLI (5)	spécial (6)
A0	X		X	X			
A1	X		X	X			
A2	X		X	X			
A3	X		X	X			
A4	X		X		X		
A5	X		X	X	X		
A6	X1		X			X	
A7	X1		X			X	pont MLI avec A6
B0	X	X	X	X	X	X	
B1	X	X	X	X	X	X	pont MLI avec B0
B2	X	X	X	X	X		
B3	X	X	X	X	X		
B4	X1	X1	X	X	X		
B5	X1	X1	X	X	X		
B6	X	X	X			X	
B7	X	X	X			X	pont MLI avec B6
C0	X1		X				compteur
C1	X1		X			X	
C2	X1		X			X	pont MLI avec C1
C3	X1						I2C: SCL
C4	X						I2C: SDA
C5	X		X				
C6	TX (vers module Bluetooth/Wi-Fi/USB)						
C7	RX (d'un module Bluetooth/Wi-Fi/USB)						
S							enregistreur
X							horloge
Y							numéro de version/série
Z							réglage global

¹ Possibilité de faire changer la valeur

faire une entrée à haute impédance, 1 crée une entrée avec résistance de rappel vers le haut « molle » (*weak pull-up*). Le 2 détermine une sortie. Un 3 transforme la broche en entrée de C A/N, tandis que le 4 en fait une entrée de capteur capacitif. Avec un 5, on obtient une sortie MLI. Le 6 est réservé à une fonction spéciale. Exemple : *s b3 3* fait de la broche B3 une entrée de C A/N.

En supplément, on peut encore fixer d'autres paramètres. À chaque broche sont attribués trois registres d'état à 8 bits, adressables en faisant suivre le numéro de broche de 1, 2 ou 3. C'est ainsi que dans l'exemple *s b32 8*, le paramètre du deuxième registre d'état est mis à la valeur 8. L'état du registre 0 est d'ailleurs la fonction de la broche précédemment adressée ; donc *s b3 3* est identique à *s b30 3*.

- **G** pour *Get*, permet de lire un réglage. Format : *g broche*. Le microcontrôleur, lui répond sous la forme *G Broche valeur*. Exemple : *g b3* donne *G B30 3*.

Les commandes et numéros de broches ne sont pas sensibles à la casse, mais le microcontrôleur répond toujours en lettres capitales. Lors du débogage, comme le contrôleur répète en écho les entrées, il est pratique d'écrire les commandes en minuscules, de manière à distinguer facilement le texte de l'appareil Android en petites lettres de celui de la carte d'E/S en majuscules.

Un document séparé (150057-W) qui reprend toutes les fonctions est disponible gratuitement sur [1].

Dans la deuxième partie de cette série, nous examinerons les modules Bluetooth, Wi-Fi et USB et nous verrons une appli qui utilise la carte d'E/S pour Android pour l'insolation et la gravure de circuits imprimés.

(150057 – version française : Robert Grignard)

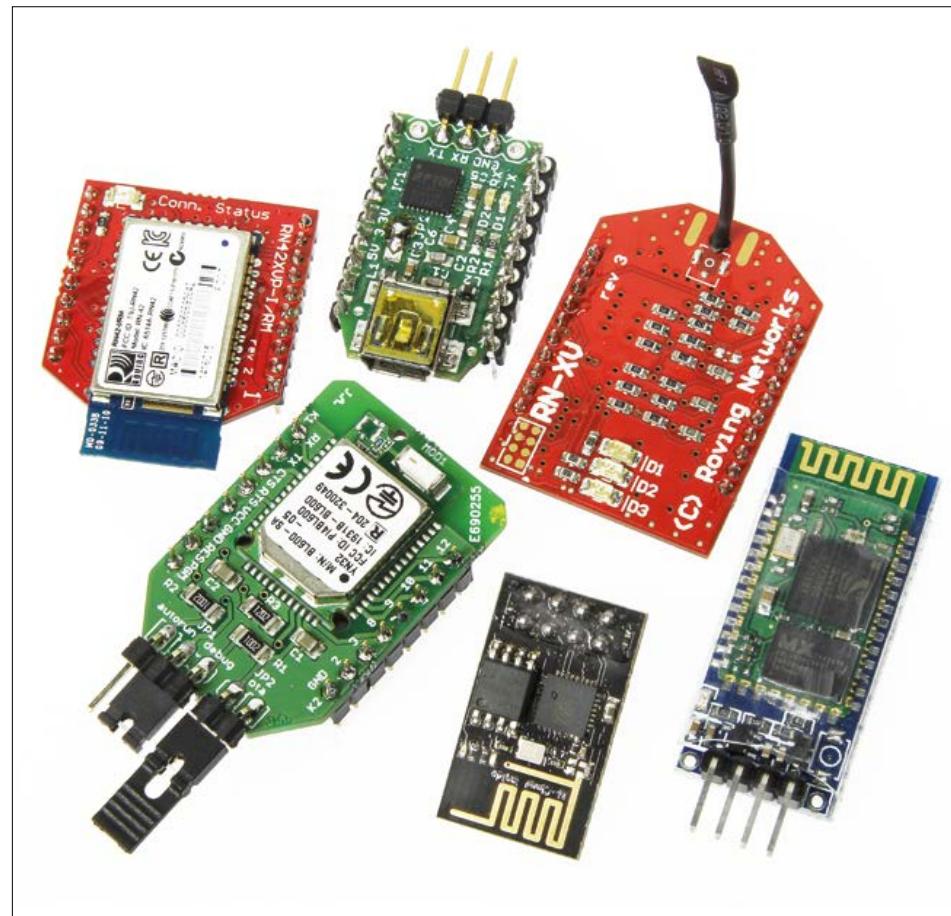


Figure 4. Quelques modules compatibles avec la carte d'E/S pour Arduino.

Liens

[1] www.elektormagazine.fr/150057



Webinaire sur la carte d'E/S pour Android :

le 17 septembre 2015, l'auteur de ce projet présentera un webinaire sur la carte universelle qu'il a développée. Ne le manquez pas ! La participation est gratuite, il suffit de s'inscrire à temps sur

www.elektor.fr/webinar.

Publicité



TOOLS, SOFTWARE & SERVICES FOR YOUR DESIGN PROCESS
From idea to product concept

28-29 October 2015 Kortrijk Xpo

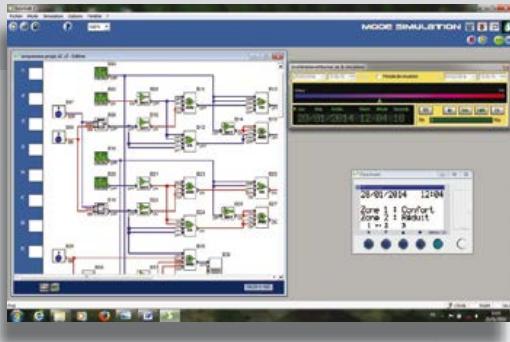
INFO & FREE REGISTRATION www.prototypingxpo.com

Powered by **howest**  Organisation Kortrijk **Xpo**

plus facile et plus intelligente ✓

votre domotique avec Zelio Logic

étudiez et programmez, puis réalisez et mettez en service un automate programmable en Ladder et FBD



Ce livre montre comment exploiter le module programmable compact Zelio Logic de SCHNEIDER, ses 12 entrées/sorties (4 entrées logiques, 4 entrées analogiques ou logiques et 4 sorties à contact).

Le logiciel pour PC Zelio Soft 2 fourni avec ce module est un outil de programmation graphique en LADDER ou FBD.

Le premier projet décrit, une station de pompage domestique, permet au lecteur de s'initier à ces deux langages.

Les électrotechniciens seront à l'aise avec LADDER (langage « à contacts »), les électroniciens préféreront FBD (blocs de fonction comparables à des circuits intégrés).

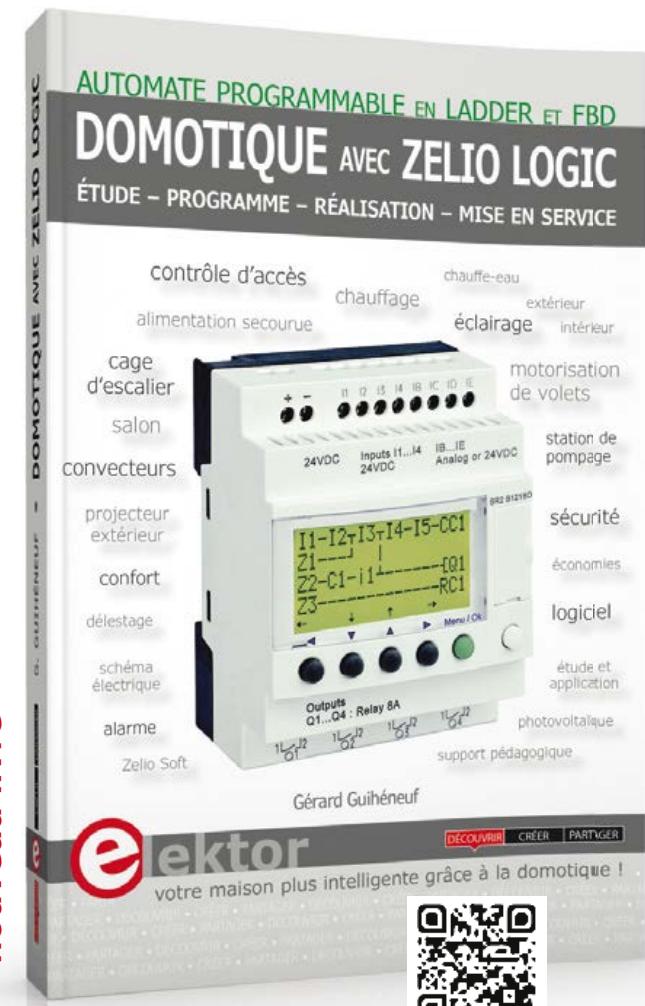
15 projets faciles à reproduire et à adapter à vos besoins :

- station de pompage domestique
- gestion multi-mode de l'éclairage d'une habitation
- gestion centralisée de l'éclairage d'une habitation
- gestionnaire de chauffage pour convecteurs
- centrale de chauffage biénergie à accumulation
- optimisation de la gestion du chauffage et de l'énergie
- alimentation secourue de circuits spécialisés
- centrale d'alarme filaire

Le dernier chapitre intéressera particulièrement les collégiens et leurs professeurs : les 7 projets présentés correspondent au thème « Domotique et confort » du nouveau programme de l'Éducation Nationale et sont complétés par des propositions d'activités pédagogiques :

- accès contrôlé à un local
- éclairage d'un salon, d'un hall d'entrée
- éclairage d'une cage d'escalier d'un immeuble
- motorisation de volets roulants
- fonctionnement d'un projecteur extérieur
- fonctionnement optimisé de convecteurs électriques

Un petit automate programmable est la solution technique idéale, bon marché et facile à mettre en œuvre dans votre logement, pour **automatiser** l'éclairage, **optimiser** la consommation d'électricité, **piloter** le chauffage ou pour vous **protéger** contre les intrusions.



nouveau livre

ISBN 978-2-86661-198-9 | 400 pages - 46,50 €

info et commande : www.elektor.fr/domo-zelio

USB déguisé en pile pour remplacer une pile AAA

Danny Winkler (Allemagne)

Le point commun des piles c'est qu'au bout d'un certain temps elles sont vides. Avec les accus, le problème ne disparaît pas complètement puisqu'il faut régulièrement les recharger. L'alimentation par piles est un inconvénient si un appareil prévu pour





être mobile, donc sans connecteur pour une alimentation externe, est utilisé de manière permanente, toujours au même endroit. Toutefois s'il y a de l'électronique, il y a une solution : une pseudo-pile avec raccordement à un port USB permet de délivrer du courant en continu.

À plusieurs reprises, Danny Winkler a été agacé par son petit lecteur/enregistreur MP3 : à chaque fois qu'il voulait l'utiliser pour enregistrer un morceau de musique intéressant, la pile était vide. Pas de chance ! Il n'y avait pas de prise pour une alimentation externe. En principe, Danny est fidèle aux composants électroniques traversants, mais un jour ou l'autre, il lui

fallait tout de même s'attaquer à ces maudits petits composants CMS. C'est ainsi que l'envie de régler son problème récurrent de baladeur MP3 a été une motivation suffisante pour se lancer dans ce domaine nouveau pour lui. « Parfait », pensa-t-il, « Cela me permettra de faire d'une pierre deux coups ». Il se mit à l'œuvre et conçut un circuit électronique de la taille

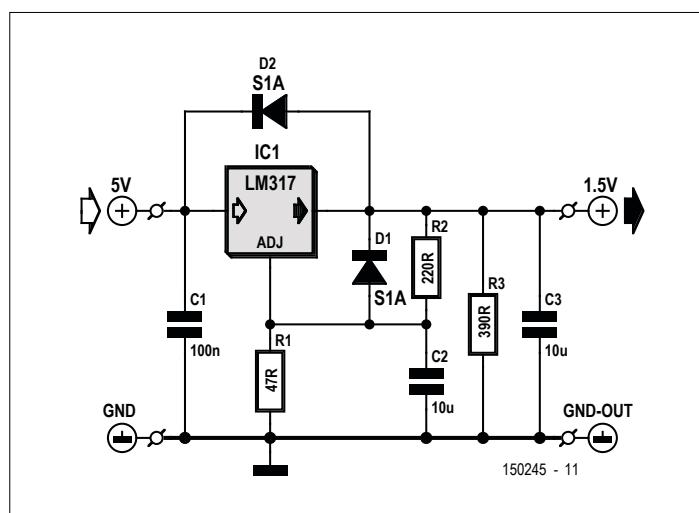


Figure 1. Schéma du circuit qui se substitut à la pile.

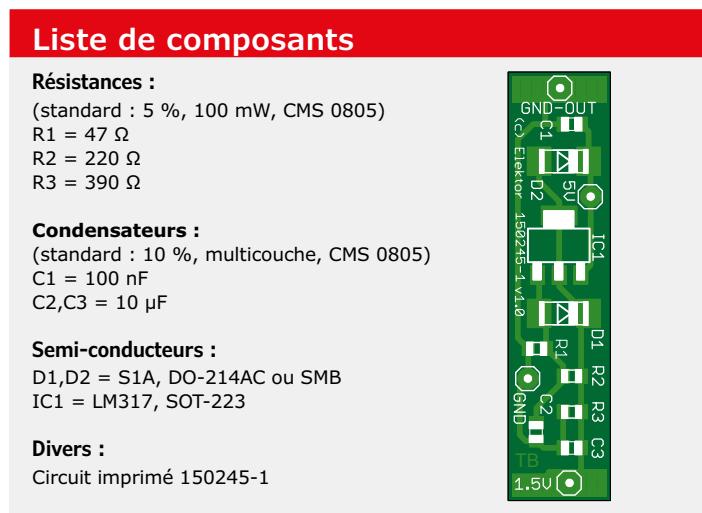


Figure 2. Circuit imprimé coté composants, au format AAA.

d'une pile AAA, qui aurait une alimentation externe. En utilisant des composants CMS, il était certain de réussir à réduire les dimensions de son circuit imprimé, pour que celui-ci remplace la pile dans le logement prévu à cet effet. Il avait raison.

Circuit de substitution de pile AAA

Primo, il faut donner au circuit imprimé la forme exacte d'une pile AAA. Secundo, pour alimenter le circuit, il faut trouver un chargeur USB pour ordiphones *and Co.* à un prix abordable. Une petite alimentation bon marché délivre 5,2 V avec une charge d'environ 1 A. Maintenant, il ne manque plus qu'un régulateur de tension pour transformer les 5,2 V en 1,5 V, tension typique d'une pile. Bingo : le schéma de la **figure 1** s'est pratiquement dessiné tout seul !

Comme Danny a une longue expérience de l'utilisation des régulateurs de tensions variables LM317 « grand format », il veut cette fois utiliser la version miniature en boîtier SOT-223. La tension de sortie est ajustée à 1,5 V par R1 et R2, suivant la formule $U_{\text{sortie}} = 1,25 \text{ V} \times (1 + R1/R2)$. Quelques condensateurs veillent à ce que le fonctionnement soit stable, les deux diodes protègent les bornes du régulateur de tension contre les tensions inappropriées. Et voilà le travail.

Pseudo-pile

Maintenant, il reste encore à concevoir le circuit imprimé de remplacement de la pile AAA. La **figure 2** montre la face des composants. Le dessin du circuit imprimé ainsi que les données pour la CAO sont disponibles ici [2]. Les **figures 3** et **4** montrent le prototype assemblé.

Les composants CMS choisis sont de taille 0805, donc loin de ressembler à un amas de poussières. En outre, il y a de la place sur le circuit imprimé, ce qui permet donc de les souder sans avoir besoin de retenir son souffle pour éviter de les déplacer. Ainsi même les opposants aux CMS ou les fanatiques des composants traversants oseront, comme Danny, se lancer dans un tel projet. Cette réalisation ne nécessite même pas de pâte à souder ou autres ustensiles spécifiques aux composants CMS. Il est tout à fait possible d'utiliser de l'étain traditionnel. Les brucelles sont fortement recommandées. Aux extrémités du circuit, on soude de petites plaques en cuivre fin, qui feront office de bornes comme sur une pile (**figure 5**).

Après le montage, et la vérification des soudures ainsi que de la polarité des diodes, il est opportun de mesurer la tension de sortie, avant de risquer un test avec un appareil portatif (encore) vivant. Comme bloc d'alimentation, si vous vous êtes procuré une de ces alimentations bon marché, vous pouvez tout bonnement couper le connecteur USB devenu inutile et relier directement les deux fils rouge et noir au 5 V et à la masse de notre circuit imprimé. ■

(150245)

Liens

- [1] Feuille de caractéristiques du LM317 : www.ti.com/lit/ds/symlink/lm117.pdf
- [2] www.elektormagazine.fr/150245

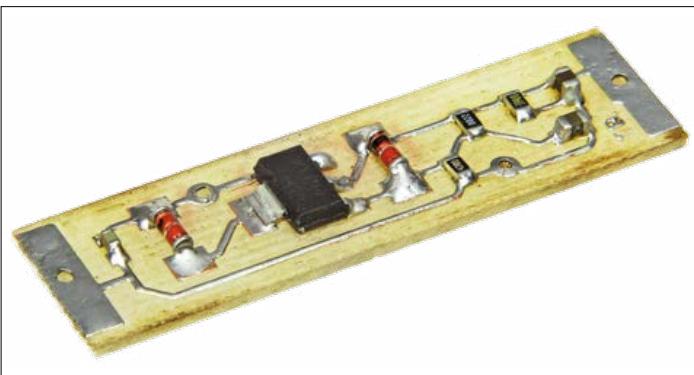


Figure 3. Le prototype terminé.



Figure 4. Prototype avec connecteur USB soudé.

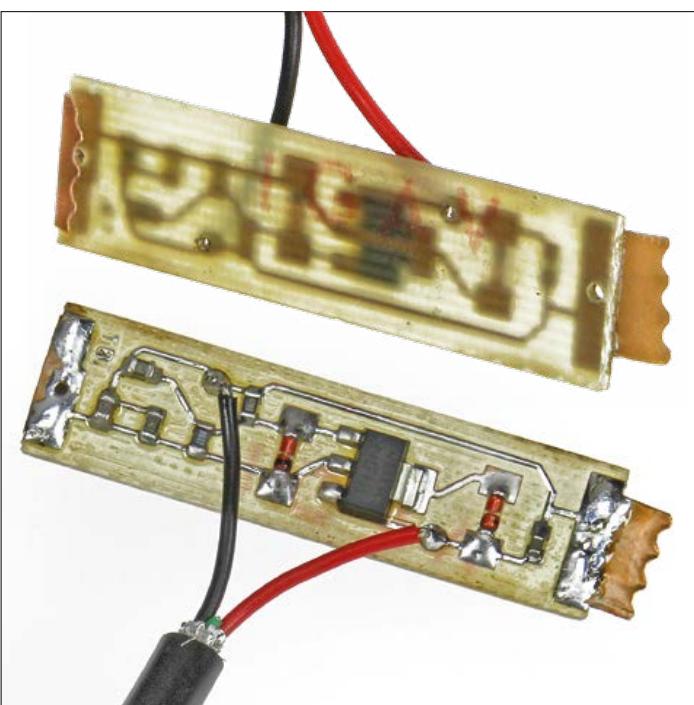
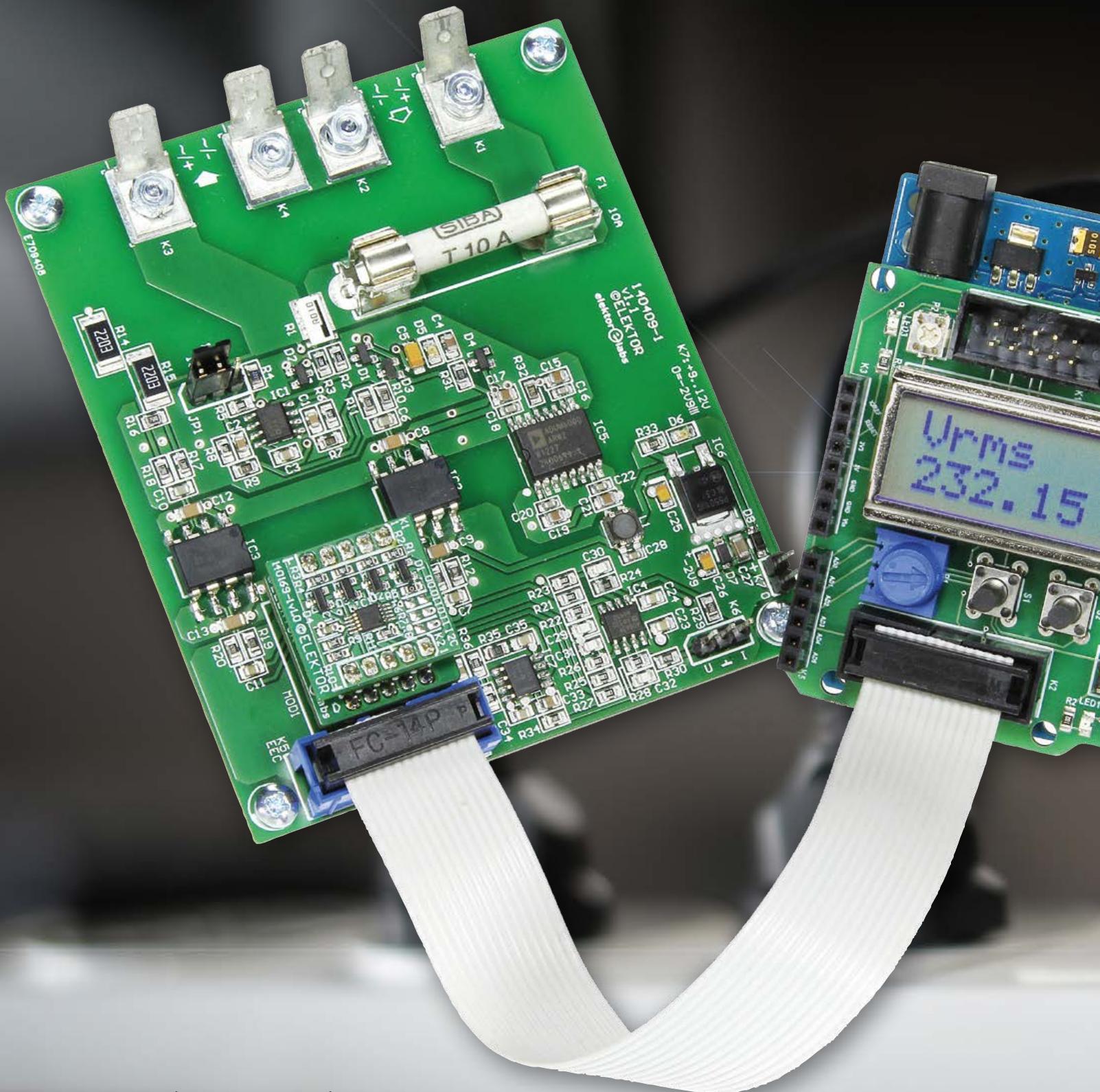


Figure 5. Deux petits morceaux de cuivre soudés aux extrémités du circuit imprimé assurent le contact dans le logement de la pile.



Dans le numéro de janvier & février 2015, nous avons présenté la carte de liaison e-BoB ADS1115 [1] qui offre quatre voies de conversion A/N à 16 bits. La même puce était apparue à l'occasion de la réception d'ELFes [2] et trônait sur le module C A/N [3], enregistreur à 16 bits, tous deux parus en septembre 2014. Mais avec la sortie d'une carte

porteuse de la minuscule puce déjà soudée (notre commode e-BoB), il nous a semblé intéressant de lui accorder d'autres occupations. Elle dispose d'un taux d'échantillonnage limité à 860 Hz, mais elle peut rendre de grands services dans les mesures à très basse fréquence. L'application toute trouvée

concerne les mesures sur la distribution électrique, la consommation en particulier. Des compteurs de puissance, on en trouve à peu de frais dans le commerce, mais ce sont des systèmes intangibles sur lesquels on ne peut rien souder ni programmer ; même pas y brancher un multimètre ou un oscilloscope, un PC, encore moins.

wattmètre CA/CC

haute précision, gammes étendues

matériel : Ton Giesberts (Elektor Labs) logiciel : Luc Lemmens (Elektor Labs)

Entreprendre des mesures sur le secteur suppose une sécurité absolue. Tension, courant ou puissance, aucune différence à cet égard, exigez toujours une séparation galvanique entre le circuit sous moyenne tension et l'appareillage à manipuler. Ce projet assure une isolation de classe II pour mettre à l'abri votre Arduino Uno, son *shield* d'expérimentation Elektor et... votre personne. Dans la foulée, vous pourrez sans souci brancher sur les sorties appropriées multimètre et oscilloscope.

Notre projet se compose d'un circuit imprimé de mesure sur lequel sont montés l'e-BoB ADS1115 et une carte de filtrage. L'électronique sur le circuit imprimé de mesure observe scrupuleusement la séparation entre les conducteurs de puissance, que ce soit en continu (CC) ou en alternatif (CA) d'une part et les circuits de mesure de l'autre. Côté secondaire,

l'ADS1115 numérise les signaux pour les transmettre par l'intermédiaire d'un *shield* d'expérimentation [4] à l'Arduino Uno qui les traite et les affiche. Ces valeurs sont celles de la puissance, de la tension et du courant efficaces, mais aussi la puissance apparente [VA] et le facteur de puissance en alternatif. En outre, l'appareil peut présenter la puissance, la tension et le cou-

rant comme valeurs équivalentes (*TRMS*) à celles obtenues en courant continu.

Séparation galvanique

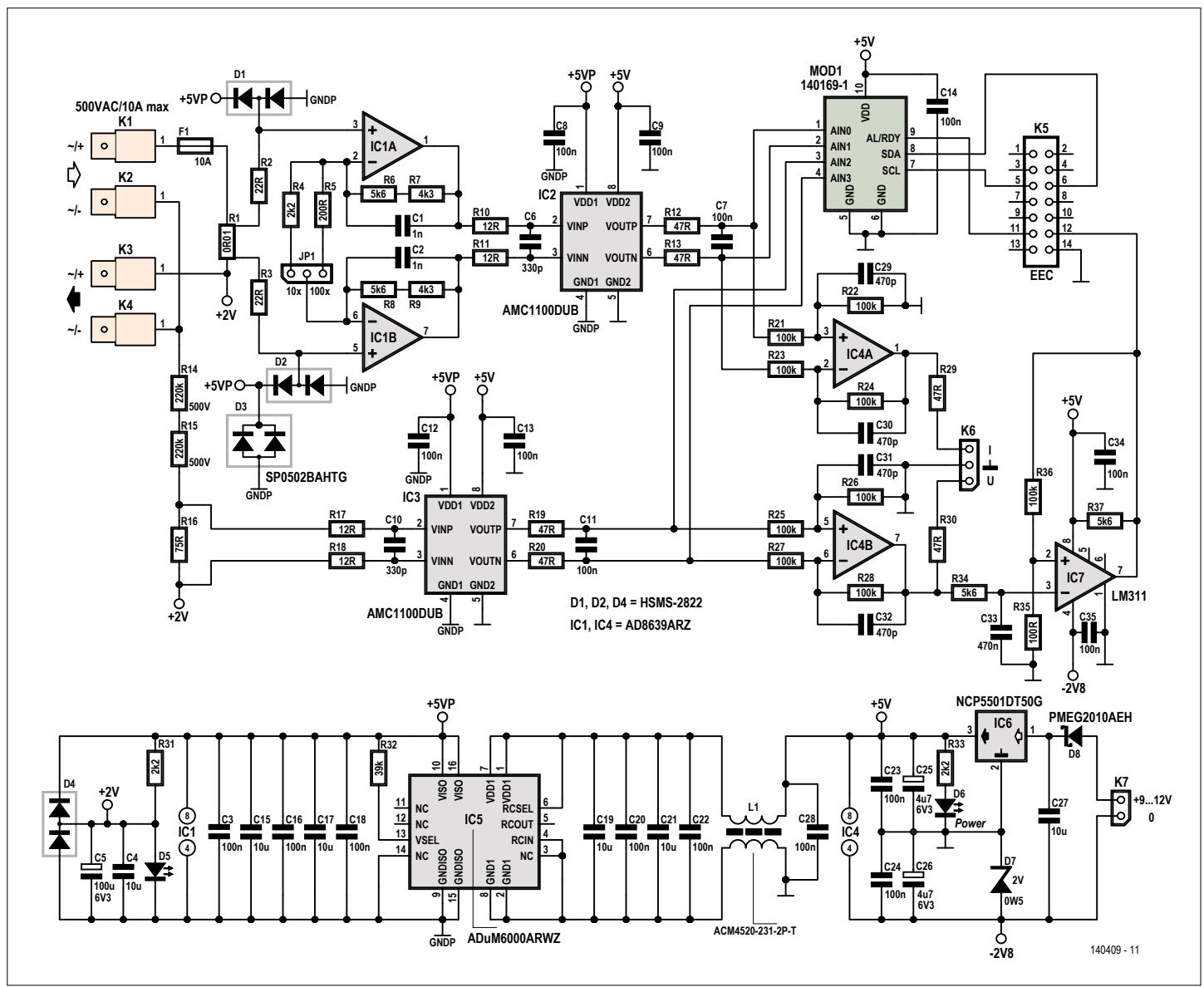
On utilise la plupart du temps un wattmètre pour des mesures sur des appareils alimentés sur le secteur, il est donc de la plus haute importance de respecter un parfait isolement galvanique entre la

Caractéristiques techniques

- Trois gammes de mesure : 10 A, 1,77 A et 0,177 A
- Tension d'entrée max. : 500 V_{CA}/300 V_{CC}
- Puissance minimale mesurable : 0,1 W
- Sorties de mesure isolées pour tension et courant
- Isolement entre côtés primaire et secondaire selon la classe II
- Calcul des valeurs de mesure sur Arduino Uno
- Affichage des mesures sur LCD à deux lignes
- Affichage de tension, courant, puissance efficace, puissance apparente et facteur de puissance en alternatif
- Affichage en continu de la puissance, de la tension et du courant

« amplificateur d'isolement totalement différentiel » affiche une bande passante minimale de 60 kHz et une rigidité diélectrique de 4 250 V_{crête}, soit une tension de service de 1 200 V_{crête}. Ce n'est vrai que si le circuit imprimé est à la hauteur, auquel cas, on obtient une isolation de classe II, ce qui rend le circuit très sûr. La puce est livrée sous boîtier CMS dit Gullwing-8, on dirait une DIP-8 écrasée en lui écartant les pattes ! Le circuit a évidemment besoin de deux alimentations totalement séparées, l'une pour l'entrée, l'autre pour la sortie. Nous aurions pu prendre des modules standard d'alimentation non stabilisée, mais la plupart de ces modules ne sont pas qualifiés pour la classe II et il leur faudrait encore des composants supplémentaires tels que des stabilisateurs de tension séparés. Mieux

partie de mesure et celle du traitement des données et des sorties de mesure. Cette séparation est d'habitude confiée à des photocoupleurs et des amplificateurs opérationnels. Mais tout cela nécessite beaucoup de composants. Une autre solution est d'utiliser des amplificateurs d'isolement spéciaux. Nous en avons trouvé des modèles faits pour cela chez Texas Instruments avec l'AMC1100. Cet



vaut utiliser un circuit intégré conçu spécialement dans ce but. Nous avons choisi un ADuM6000 d'Analog Devices, un convertisseur CC/CC isolé avec sortie stabilisée à 5 ou 3,3 V. La tension d'isolation V_{IORM} s'élève à 864 V_{crête}.

Circuit

La manière la plus simple de mesurer du **courant**, c'est de mesurer la tension aux bornes d'une résistance en série avec la charge. C'est ainsi que presque tout multimètre fonctionne. La **figure 1** couvre l'ensemble du circuit dans lequel, à gauche, on reconnaît les languettes de raccordement K1 à K4 pour les mesures, deux en entrée, deux en sortie. La résistance *shunt* R1 de 0,01 Ω se situe entre K1 et K3 et en série avec elle, il y a un fusible de 10 A.

Ce sont des languettes Faston qui servent de connecteurs d'entrée et de sortie, elles sont boulonnées sur le circuit imprimé. Elles assurent de bonnes liaisons tant mécaniques qu'électriques et acceptent des cosses prévues pour de gros fils. Pour influencer le moins possible la consommation dans le circuit à mesurer, le *shunt* doit avoir la résistance la plus petite possible, mais quand on mesure de faibles courants, la chute de potentiel devient tellement faible qu'elle est difficile à déceler. Une section d'amplification est alors nécessaire, avec un gain réglable. Elle se situe devant l'amplificateur d'isolation et met en jeu le double amplificateur opérationnel IC1. C'est un AD8639 caractérisé par l'annulation automatique du décalage d'entrée (*auto-zero*) et par son excursion totale en sortie (*rail-to-rail*). Le décalage d'entrée typique est de 3 µV, maximum 9 µV, alors que la dérive maximale ne fait que 0,06 µV. Un excellent choix pour cette application, mesures en CC comprises. Le produit du gain par la bande passante (GPB) s'élève à 1,35 MHz, ce qui signifie qu'au facteur d'amplification le plus grand demandé, la largeur de bande est encore de plus de 10 kHz, bien plus que nécessaire.

Nous utilisons IC1 en configuration d'amplificateur d'instrumentation au gain différentiel de $1 + 2 \times (R6 + R7)/RG$, avec $RG = R4$ ou $R5$. Sans cavalier sur JP1, l'amplification, à savoir la tension entre la sortie de IC1A et IC1B divisée par la tension sur R1, est de 1. Le cavalier sur JP1 permet de sélectionner un gain de 10

ou de 100. Ces réglages d'amplification, combinés au gain de IC2 permettent trois gammes de mesure : 17,7 A (1 x, limité par le fusible F1 à 10 A), 1,77 A (10 x) et 0,177 A (100 x) pour des courants de forme sinusoïdale. Les valeurs efficaces (*RMS*) avec des signaux à facteur de crête élevé seront moindres. Selon le niveau de bruit, la puissance minimale que le circuit peut mesurer est inférieure à 0,1 W (la valeur de mesure du convertisseur A/N à -50 dB de la pleine échelle).

Pour maintenir le signal à l'intérieur du domaine de mode commun d'IC1 sur toutes les gammes de mesure, un côté de R1 est relié à une tension de +2 V obtenue à l'aide de la LED D5, le témoin d'alimentation pour le primaire, le côté « chaud ». Elle sert aussi de stabilisateur de tension, à la manière d'une zener.

L'amplificateur d'isolation AMC1100 (IC2) du signal d'entrée ne peut pas accepter de tension différentielle supérieure à 250 mV. Il amplifie huit fois, donc la tension de sortie maximale sera de 2 V. La puce IC2 est dotée d'une entrée différentielle avec un circuit de condensateur commuté. La spécification du bruit de sortie est de 3,1 mV_{eff}, sans plus de détails. Si elle vaut pour toute la largeur de bande, on pourra diminuer le niveau de bruit en rétrécissant la bande passante avant le convertisseur A/N. Nous avons donc ajouté, sur une petite carte séparée, un filtre à quatre sections (cf. encadré Module de filtres) avant le C A/N. Sans lui, le rapport signal/bruit serait de 645. En réduisant la bande passante d'entrée du C A/N, nous pouvons l'améliorer d'un facteur $\sqrt{60}$, soit 5 000. En conséquence, nous pouvons mesurer de plus petits signaux sur le *shunt*.

Pour la mesure de la **tension**, cette fois, on part des bornes K1/K2 et c'est IC3, le deuxième AMC1100 qui opère. Le diviseur de tension R14 à R16 affaiblit la tension d'environ 5 900 fois, si bien qu'il en reste 39 mV sur les entrées de IC3 pour une tension secteur de 230 V. Dans le diviseur, un côté de R16 est relié à la tension de référence de +2 V. Les sorties des deux AMC1100 sont raccordées aux entrées de l'e-BoB ADS1115.

Notre vœu était bien de pouvoir mesurer de hautes tensions en toute sécurité sur un oscilloscope ou un multimètre. Les sorties différentielles des AMC1100 nous compliqueraient la vie, une tension

asymétrique est préférable. Le double amplificateur opérationnel IC4, encore un AD8639, va s'en occuper. Il transforme les signaux symétriques issus des deux amplificateurs d'isolation en une réplique asymétrique de même amplitude. Le décalage de mode commun de 2,55 V (pour $V_{DD} = 5$ V) présent aux sorties des AMC1100 est alors fortement réduit, mais pas tout à fait. Selon la différence entre les composantes continues des sorties de chaque AMC, il se peut qu'il y ait un décalage de quelques millivolts (< 12 mV) aux sorties de IC4.

Le domaine de mode commun en entrée de l'AD8639 va de -0,1 V jusqu'à 3 V sous alimentation asymétrique de 5 V. Il faut que les tensions de sortie des AMC s'inscrivent dans ce domaine. De 2,55 V, les diviseurs de tension R21/R22 et R25/R26 abaissent la tension continue à 1,275 V. D'après nos mesures, le signal différentiel de sortie maximal que les AMC peuvent délivrer s'élève à 2,5 V (1,25 V crête). Donc la tension maximale sur les entrées de IC4 restera inférieure à 1,275 V_{cc} plus la moitié de la tension alternative de pointe, 0,625 V, ce qui fait 1,9 V. Nous sommes bien dans le domaine de mode commun acceptable par l'AD8639.

Le comparateur IC7, un LM311, effectue la détection du passage par zéro de la tension. Il a une sortie à collecteur ouvert. En lui fournissant une alimentation dissymétrique (+5 V et -2,8 V), cette détection s'opère facilement et avec peu de composants. La résistance de collecteur, c'est R37 et le réseau R36/R35 assure une sérieuse réaction pour détecter les passages par zéro de manière fiable. La sortie du comparateur est reliée à la broche 12 de K5. La détection du zéro la plus précise se produit sur les impulsions négatives, au moment où la sortie de IC7 est encore basse et il n'y a pas d'hystérisis. L'entrée non inverseuse du comparateur est alors encore au niveau de la masse.

La transmission des données vers l'Arduino passe par le connecteur EEC K5. Les signaux I²C du C A/N et quelques autres arrivent par K2 sur un *shield* d'expérimentation installé sur une carte Arduino Uno. L'alimentation de 5 V pour le côté secondaire est fournie par le régulateur à faible déchet IC6. C'est suffisant pour la plupart des puces de ce circuit, seule IC4 a besoin d'une tension négative pour déli-

vrer un signal alternatif sans décalage en continu. En outre, nous voulons aussi pouvoir mesurer des tensions continues, donc les sorties de K6 doivent transmettre aussi des tensions négatives. Aussi a-t-on créé une masse virtuelle au moyen d'une diode stabilisatrice (D7) de 2 V dans la liaison de masse du régulateur IC6.

La tension d'alimentation isolée pour la partie entrée des AMC1100 et de l'ampli d'entrée IC1 est fournie par IC5, un

ADuM6000. On règle le stabilisateur intégré à la valeur de 5 V en reliant VSEL à Viso et on lui adjoint la bobine de mode commun L1 pour atténuer d'éventuels parasites. Avec 3 A, elle est surdimensionnée, mais nous l'avons choisie en raison de sa basse résistance série ($2 \times 50 \text{ m}\Omega$).

Mesures avec Arduino

Avec l'ADS1115, nous disposons donc d'un seul C A/N dont la fréquence

d'échantillonnage n'est que de 860 Hz pour numériser quatre entrées. Impossible donc de tout traiter en temps réel au débit maximal. Il faudrait donc alterner les mesures en commutant par le bus I²C, mais cela occasionne trop de retard entre les mesures. Le sujet a été abordé dans *Bruits de labo* du numéro de juillet & août 2015, pp. 126, sous le titre « à la recherche du code ». Nous avons finalement choisi de prélever une salve

Liste des composants - mesure

Résistances :

R1 = 0,01 Ω, 2 W, 1 % (Ohmite FC4L64R010FER)
 R2, R3 = 22 Ω, 0,25 W, 5 %, CMS 0805
 R4, R31, R33 = 2,20 kΩ, 0,125 W, 1 %, CMS 0805
 R5 = 200 Ω, 0,125 W, 1 %, CMS 0805
 R6, R8, R34, R37 = 5,6 kΩ, 0,125 W, 1 %, CMS 0805
 R7, R9 = 4,3 kΩ, 0,125 W, 1 %, CMS 0805
 R10, R11, R17, R18 = 12 Ω, 0,125 W, 1 %, CMS 0805
 R12, R13, R19, R20, R29, R30 = 47 Ω, 0,125 W, 1 %, CMS 0805
 R14, R15 = 220 kΩ, 1,5 W, 1 %, 500 V, CMS 2512
 R16 = 75 Ω, 0,125 W, 1 %, CMS 0805
 R21 à R28, R36 = 100 kΩ, 0,125 W, 1 %, CMS 0805
 R32 = 39 kΩ, 0,125 W, 5 %, CMS 0805
 R35 = 100 Ω, 0,125 W, 1 %, CMS 0805

Condensateurs :

C1, C2 = 1 nF/50 V, 5 %, CMS 0805, COG/NPO
 C3, C7, C11, C14, C23, C24, C34, C35 = 100 nF/50 V, 10 %, CMS 0805, X7R
 C4, C15, C17, C19, C21 = 10 µF/10 V, 10 %, CMS 0805, X7R
 C5 = 100 µF/6,3 V, 20 %, CMS Case A (1206), tantalé
 C6, C10 = 330 pF/50 V, 5 %, CMS 0805, COG/NPO
 C8, C9, C12, C13 = 100 nF/50 V, 10 %, CMS 1206, X7R
 C16, C18, C20, C22, C28 = 100 nF/25 V, 10 %, CMS 0603, X7R
 C25, C26 = 4u7/6,3 V, 10 %, CMS Case R (0805), tantalé
 C27 = 10 µF/25 V, +80/-20 %, CMS 1206, Y5V
 C29 à C32 = 470 pF/50 V, 5 %, CMS 0805, COG/NPO
 C33 = 470 nF/16 V, 10 %, CMS 0805, X7R

Inductance :

L1 = ACM4520-231-2P-T, CMS, bobine de choc mode commun 3 A, 2x50 mΩ, 230 Ω@100 MHz

Semi-conducteurs :

D1, D2, D4 = HSMS-2822-TR1G, CMS SOT-23
 D3 = SP0502BAHTG, CMS SOT-23
 D5, D6 = LED verte, CMS 0805
 D7 = BZT52C2V0-7-F, CMS SOD-123 (diode zener 2 V/0,5 W)
 D8 = PMEG2010AEH, 20 V/1 A, CMS SOD-123F
 IC1, IC4 = AD8639ARZ, CMS SOIC-8
 IC2, IC3 = AMC1100DUB, CMS Gullwing-8 (SOP-8)
 IC5 = ADuM6000ARWZ, CMS RW-16 (SOIC_W-16)
 IC6 = NCP5501DT50G, CMS DPAK3
 IC7 = LM311D, CMS SOIC-8

Divers :

K1 à K4 = languette Faston 6,35 mm avec raccord à vis 3,3 mm
 K5 = embase à 2x7 picots au pas de 2,54 mm
 K6, JP1 = embase à 3 picots au pas de 2,54 mm
 K7 = embase à 2 picots au pas de 2,54 mm
 MOD1 = support SIL à 2x5 contacts au pas de 2,54 mm
 F1 = porte-fusible encartable, 32 A/600 V (6,3 x 32 mm)
 F1 = fusible 10 A (T, retard), 500 VCA/300 VCC, 6,3 x 32 mm
 module 140169-91 (e-BoB ADS1115)*
 circuit imprimé du wattmètre : 140409-1*

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

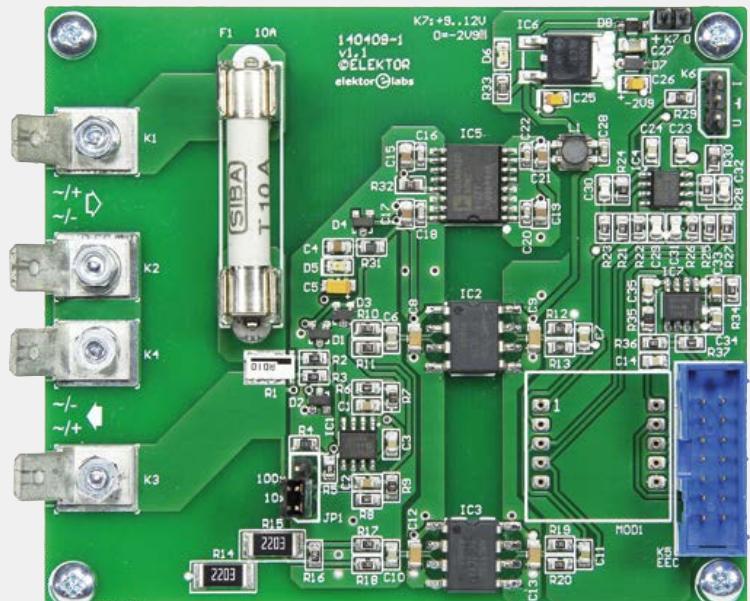
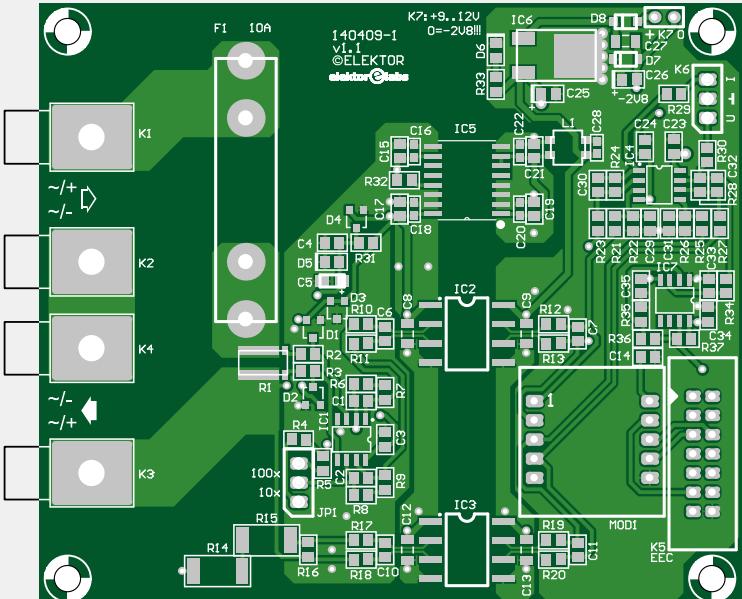


Figure 2. Le tracé des pistes du circuit imprimé de mesure. Au milieu, sous les trois grandes puces, on voit une large bande sans cuivre pour assurer la séparation galvanique entre côtés primaire et secondaire.

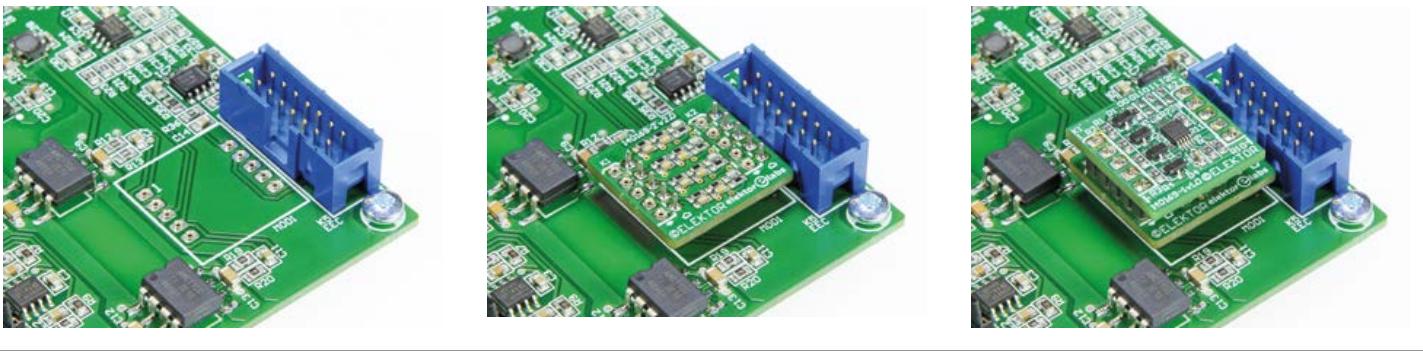


Figure 3. La carte du convertisseur A/N et celle des filtres se superposent, reliées par le connecteur MOD1 au circuit imprimé de mesure.

de 172 échantillons de mesure de tension, suivie d'une autre du même nombre de valeurs du courant. Pour déterminer la relation de phase entre tension et courant, les deux salves démarrent lors d'un passage par zéro de la tension. Raison pour laquelle le wattmètre dispose du comparateur IC7 pour la détection, en liaison avec Arduino par la broche 9 sur le port B1. Au cas où ce signal de synchronisation manquerait, le logiciel passerait automatiquement en mesure en CC, pour revenir en CA dès le retour de ces impulsions.

Le logiciel pour le wattmètre est un *sketch* pour Arduino Uno, disponible gratuitement sur [5], qui mesure et ensuite calcule les grandeurs suivantes (cf. aussi [6]).

Pour l'alternatif :

$$P_{VA} = \frac{\sum_{i=1}^N V[i] \times I[i]}{N}$$

$$V_{RMS} = \sqrt{\left(\sum_{i=1}^N V^2[i] \right) / N}$$

$$I_{RMS} = \sqrt{\left(\sum_{i=1}^N I^2[i] \right) / N}$$

$$P_{RMS} = V_{RMS} \times I_{RMS}$$

$$P - \text{factor} = \frac{P_{RMS}}{P_{VA}}$$

Pour le continu :

$$V = \left(\sum_{i=1}^N V[i] \right) / N$$

$$I = \left(\sum_{i=1}^N I[i] \right) / N$$

$$P = V \times I$$

Sur le LCD, toutes ces valeurs de mesure s'affichent à tour de rôle telles qu'on les voit à la **figure 4**.

Construction

Le circuit imprimé de mesure 140409-1 de la **figure 2** est largement dimensionné. On y voit presque partout des CMS, ce qui peut donner du souci aux soudeurs moins aguerris, mais il n'y a pas de composants très petits. Ce sont les CMS qu'il faut planter en premier lieu, avant les embases, les connecteurs et le porte-fusible. Suivent les languettes à visser solidement par boulon avec rondelle grower. L'e-BoB ADS1115 (140169-1) est disponible complet, prêt à l'emploi dans l'e-choppe. Le circuit imprimé de filtres 140169-2, il faudra le construire vous-même, il porte des CMS sur les deux faces. En face supérieure, il y a deux connecteurs à 5 contacts pour la liaison avec l'e-BoB ADS et deux embases à 5 picots sur la face inférieure pour le branchement sur le circuit imprimé de mesure. Ces deux cartes doivent s'enfoncer l'une sur l'autre, filtres en dessous, avant de brancher le tout sur le circuit imprimé de mesure par le connecteur MOD1. La carte Arduino Uno revêtue du circuit imprimé d'expérimentation est à

relier au circuit imprimé de mesure, via un câble plat à 14 conducteurs, terminé par des connecteurs à sertir. Regardez les photos.

Il est important que le circuit imprimé de mesure soit enfermé dans un boîtier bien isolé pour empêcher tout contact avec la tension du secteur pendant les mesures.

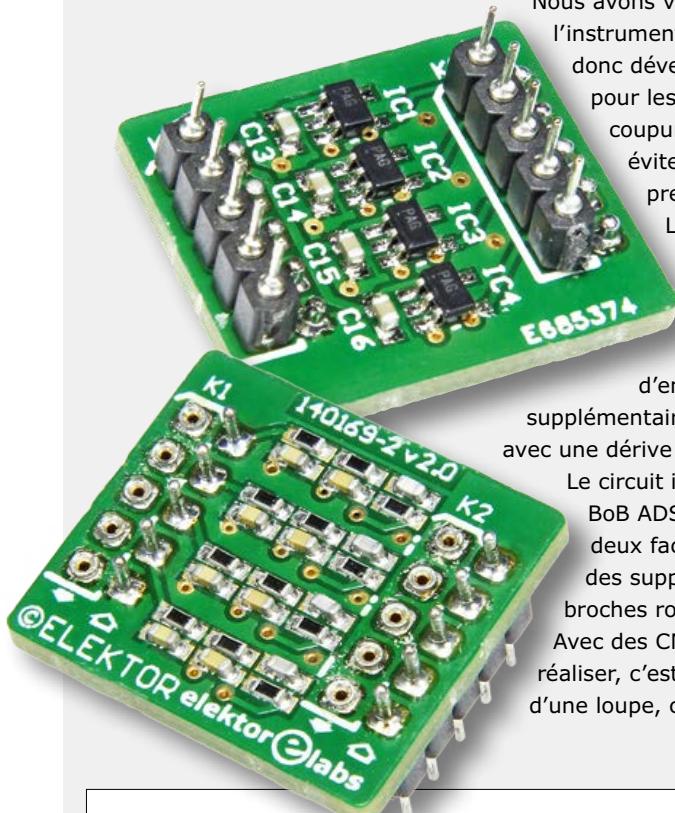
D'un côté du boîtier, on peut laisser dépasser les languettes, mais il est plus sûr de munir ces connexions de douilles pour fiche banane ou de prises de courant, modèle européen éventuellement. De l'autre côté, prévoir une découpe pour passer le câble plat. On peut aussi installer quelques sorties de mesure sur K6. Le câble d'alimentation est à relier à K7. Sur le *shield* d'expérimentation, il faut encore câbler deux autres liaisons pour nantir le connecteur EEC de tous les signaux nécessaires. Pour ce faire, installez au moyen de fil isolé ou de cuivre émaillé les raccords entre K2-11 et K4-1 ainsi qu'entre K2-12 et K4-2.

On alimente le circuit de mesure par un adaptateur secteur qui fournit au moins 200 mA sous 9 à 12 Vcc. L'Arduino est alimenté par son propre adaptateur.

Réglage

Le logiciel permet de régler les décalages d'entrée, l'amplification ou l'atténuation pour toutes les mesures de tension et de courant sur la section d'entrée analogique du circuit imprimé de mesure. Les valeurs obtenues après étalonnage sont conservées dans l'EEPROM de l'Arduino, si bien qu'il ne faut en principe réaliser qu'un seul étalonnage. Lors de l'initialisation, le logiciel vérifie si les adresses de mémoire correspon-

Module de filtres

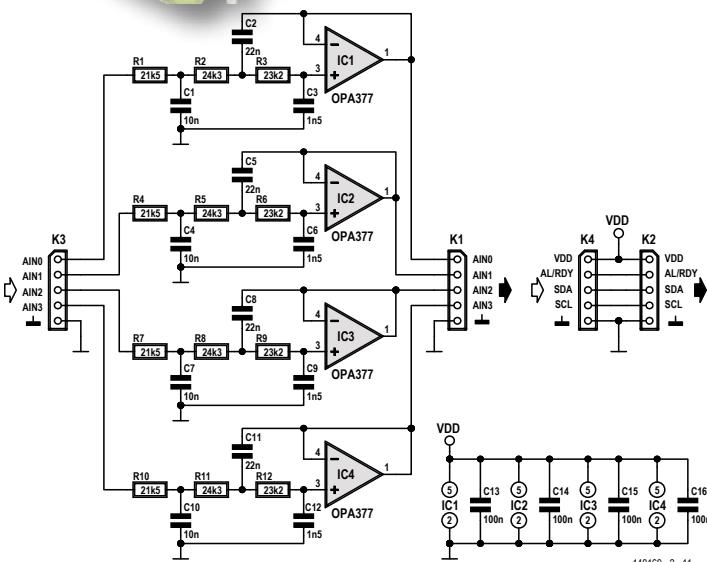
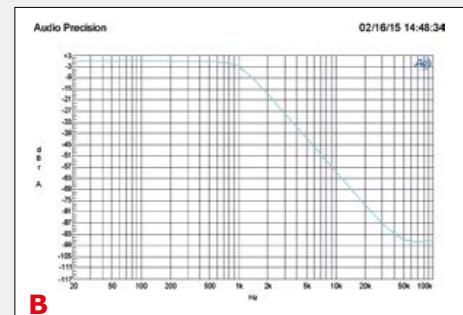
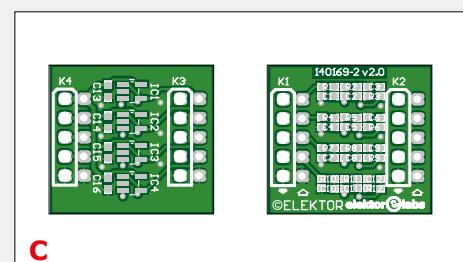


Nous avons vu qu'il était possible d'augmenter considérablement la sensibilité de l'instrument en réduisant la largeur de bande devant l'entrée du C A/N. Nous avons donc développé un module constitué de quatre filtres passe-bas du troisième ordre pour les quatre entrées de l'ADS1115. Ils sont accordés sur une fréquence de coupure de 1 kHz, une valeur qui peut sembler haute, mais il fallait absolument éviter d'influencer la phase des signaux de mesure à 50 ou 60 Hz et de leurs premiers harmoniques.

Le type de filtre choisi (**figure A**) est le Sallen&Key avec une caractéristique Butterworth d'ordre trois (courbe de la **figure B**). Les amplificateurs opérationnels utilisés sont des OPA377, ils offrent l'excursion totale en sortie et de très bonnes caractéristiques en matière de bruit malgré les entrées CMOS. Le courant de polarisation d'entrée typique n'est que de 0,2 pA, ils n'occasionnent donc aucun décalage supplémentaire avec les résistances à film. Le décalage d'entrée n'est que de 0,25 µV avec une dérive typique en température de 0,32 µV/°C.

Le circuit imprimé a été conçu pour s'adapter précisément aux dimensions de l'e-BoB ADS qui le surplombe. Pour y arriver, il a fallu y planter des CMS sur les deux faces (**figure C**). La carte possède deux connecteurs sur la face supérieure, des supports de puce SIL, par exemple ; de l'autre côté, deux embases à fines broches rondes pour rendre le tout enfichable.

Avec des CMS sur les deux faces, cette petite carte n'est certainement pas facile à réaliser, c'est le prix à payer pour la compacité. Travaillez avec soin et servez-vous d'une loupe, c'est bien pratique dans ce cas-ci.

**A****B****C**

Liste des composants - filtres

Résistances :

- R1, R4, R7, R10 = 21,5 kΩ, 1 %, 0,1 W, CMS 0603
- R2, R5, R8, R11 = 24,3 kΩ, 1 %, 0,1 W, CMS 0603
- R3, R6, R9, R12 = 23,2 kΩ, 1 %, 0,1 W, CMS 0603

Condensateurs :

- C1, C4, C7, C10 = 10 nF/50 V, 5 %, C0G/NP0 ou X7R, CMS 0603

C2, C5, C8, C11 = 22 nF/50 V, 5 %, C0G/NP0 ou X7R, CMS 0603

C3, C6, C9, C12 = 1,5 nF/50 V, 5 %, C0G/NP0 ou X7R, CMS 0603

C13, C14, C15, C16 = 100 nF/50 V, 10 %, X7R, CMS 0603

Semi-conducteurs :

IC1, IC2, IC3, IC4 = OPA377AIDBV7

Divers :

K1, K2 = connecteur à 5 contacts

au pas de 2,54 mm

K3, K4 = embase à 5 picots

au pas de 2,54 mm

circuit imprimé du module des filtres :

140169-2*

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

dantes ont déjà été utilisées, donc qu'elles contiennent autre chose que la valeur 255. Sinon, il dirige automatiquement vers la routine d'étalonnage au lieu de commencer les mesures tout de suite. On peut demander un nouvel étalonnage en actionnant S1 après la mise sous tension. En cours d'étalonnage, à chaque étape, l'affichage vous guide en indiquant ce qu'il faut faire et vous invite à pousser sur S1 pour voir le paramètre correspondant et sa valeur actuelle.

On commence par le décalage de la tension de mesure (*OV input*). Avec les entrées K1 et K2 ouvertes, utilisez les boutons-pousoirs S1 et S2 sur le *shield* d'expérimentation pour atteindre le zéro-tage qui s'affiche sur le LCD. En l'absence d'action pendant 5 s, le logiciel passe de lui-même au réglage du facteur d'amplification pour la tension (*input V*). Connectez pour cela entre K1 et K2 une tension continue de valeur connue et relativement haute, d'au moins 50 V, avec un bon multimètre en parallèle ; utilisez S1 et S2 pour que l'affichage sur le LCD corresponde à la lecture sur le multimètre. Encore 5 s d'attente pour le réglage du décalage en courant (*OA input*). Refaites le même zérotage que précédemment.

Finalement, c'est le facteur d'amplification pour la mesure du courant (*input I*) qu'il faut régler. Branchez une source de courant continu sur K1/K2 et une charge en série avec un ampèremètre sur K3/K4. Faites coïncider les deux lectures comme vous l'avez fait pour la tension. ■

(140409 – version française : Robert Grignard)

Liens

- [1] www.elektormagazine.fr/140169
- [2] www.elektormagazine.fr/140035
- [3] www.elektormagazine.fr/130485
- [4] [Shield d'expérimentation : 140009-91](http://www.elektormagazine.fr/140009-91)
- [5] www.elektormagazine.fr/140409
- [6] <http://openenergymonitor.org/emon/buildingblocks/ac-power-arduino-maths>



Figure 4. Les valeurs mesurées s'affichent l'une après l'autre sur le LCD. En CA : tension, courant, puissance efficace, puissance apparente et facteur de puissance ; en CC : tension, courant et puissance.

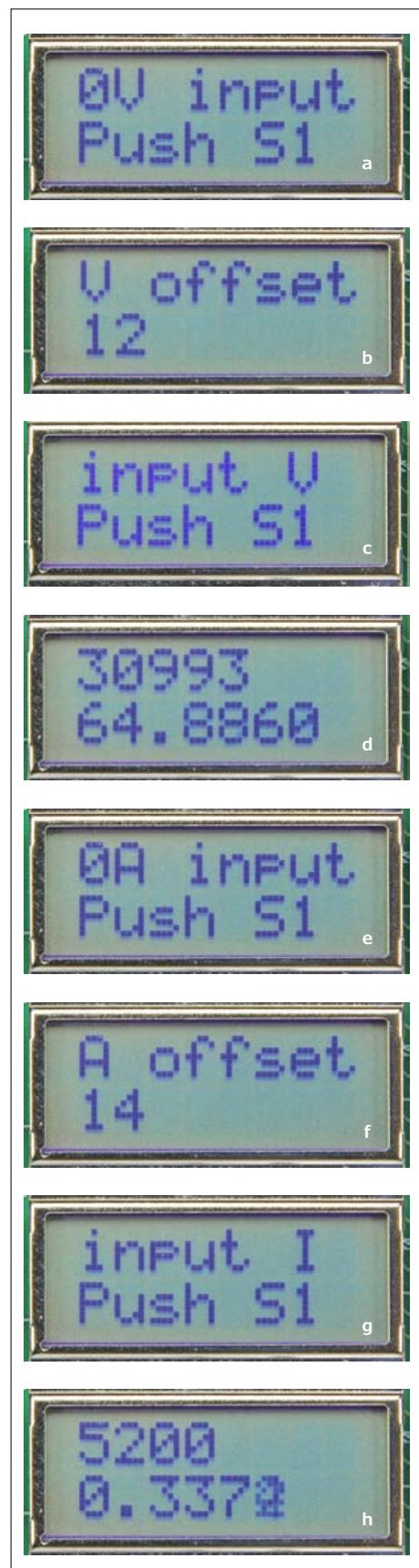


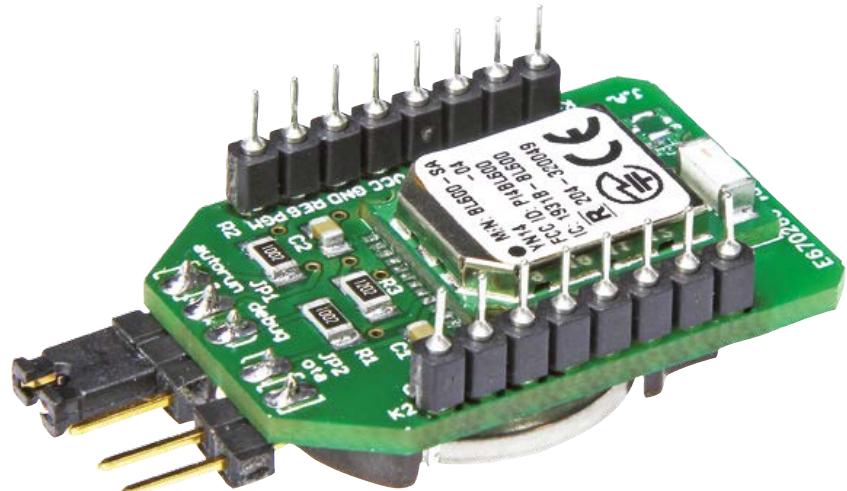
Figure 5. Les affichages lors de l'étalonnage. Se suivent : décalage pour la mesure de tension (a/b), gain pour la mesure de tension (c/d), décalage pour la mesure de courant (e/f) et gain pour la mesure de courant (g/h).

e-BoB BL600

Bluetooth Low Energy

(6^{ème} partie)

- Low Energy, 5 µA
- L'événement TIMER
- Les événements d'échange de données
- L'événement de connexion



Jennifer Aubinais (Paris) elektor@aubinais.net

Dans les précédents articles, nous avons épluché avec vous les bases de la programmation de notre e-BoB BL600 et montré comment le mettre en oeuvre. Le sixième et dernier article de cette série s'intéressera à sa faible consommation (Low Energy) qui est précisément son point fort ! Ensuite vous volerez de vos propres ailes.

Low Energy, 5 µA

En mode de sommeil profond, la consommation n'est que de 0,4 µA, mais, dans ce mode, le réveil ne peut se faire que par une remise à zéro ou par un changement d'état d'une entrée. Nous utiliserons plutôt le mode veille, avec une consommation de 5 µA, et c'est notre programme qui réveillera le module, p. ex. à l'aide des TIMERS.

Pour illustrer ces explications, nous n'utiliserons qu'une seule LED. Si vous connectez d'autres composants à votre BL600, il faudra adapter votre programme afin de garder les 5 µA en mode veille. Pour mettre en évidence la faible consommation du module, nous le ferons fonctionner alternativement en mode normal et en mode veille. Connectez votre ampèremètre selon le schéma : lorsque la LED est éteinte, l'intensité du courant est de 5 µA.

Le code source du programme de test *b1600-6.sb* présenté ci-dessous est disponible sur le site d'Elektor [1]. Si vous voulez y apporter des modifications, reportez-vous aux articles précédents qui donnent toutes les informations à ce sujet.

Les étapes :

- **Arrêt du Bluetooth :** le Bluetooth est arrêté par la fonction *BleAvertStop* et réveillé par la fonction *BleAdvertisStart*. L'alternance sera réalisée avec des TIMERS (paragraphe *TIMER* pour la description du code en **rouge** et **vert**).
- **LED éteinte :** Tout ce qui est connecté au module consomme. Donc, notre LED sera mise à l'état LOW par la

fonction *GPIOWrite*. Nous l'allumerons quand le Bluetooth est démarré et l'éteindrons quand il est arrêté (avec des fonctions dans le *TIMER* -- voir ci-dessous).

- **UART arrêté en mode NON DEBUG :** Le dispositif de communication série du module consomme aussi. Nous l'éteindrons par la fonction *UartClose*, les ports TX et RTS seront mis en sortie avec comme état *High* et *Low* respectivement. Deux modifications s'imposent ici pour éviter une erreur *~~~default~~~* :

- Si nous fermons le port série au lancement du programme, dans le programme *MAIN*, le tampon du port série est encore plein. Pour qu'il se vide, nous attendrons donc 1000 ms (durée arbitraire) avant de le fermer, en utilisant un *TIMER* dans le code en **mauve**.
- Une fois le port série clôturé, le handler *HandlerLoop* transfère les données reçues par le Bluetooth vers le port série et inversement avec la fonction *BleVspUartBridge*. Nous mettrons cette fonction en commentaire et ferons notre propre code en **orange** (cf *Les événements d'échange de données*).

• Mesure

La résistance interne élevée de certains ampèremètres peut gêner la mesure en mode réveillé. Avec une résistance de 100 Ω p. ex. pour un courant de 10 mA (LED allumée), la chute de tension aux bornes de l'ampèremètre sera de 1 V : le module ne fonctionne donc plus... Choisissez le calibre 2000 µA au lieu de 200 µA.

L'événement TIMER

Le TIMER, présenté dans l'article de mai 2015 [4] sur le chevillard, est utilisé pour créer une alternance d'arrêts et de relances du Bluetooth. Pour cela, il faut coder :

- la déclaration de son handle : *OnEvent EVTMR1 Call FuncTimer1*
- la fonction : *Function FuncTimer1()*
- le déclenchement du TIMER : *TIMERSTART(1,4000,0)*

Pour faire apparaître sur un ampèremètre la faible consommation du module, nous activons le Bluetooth (LED allumée) pendant 25 ms (fonction *bleadvertstop*, code en **rouge**), et l'arrêtions (LED éteinte) pendant 4000 ms (fonction *bleadvertstart*, code en **vert**).

n = Bleadvertstart(0,Adrt\$,25,4000,0)

Le TIMER servira encore à retarder la clôture du port UART.

Les événements d'échange de données

Les données reçues par le Bluetooth et sur le port série sont traitées par un handler, comme nous l'avions fait p. ex. pour intercepter les données provenant de votre téléphone pour commander une LED tricolore [4]. Ici, il nous pose un problème, car nous avons fermé le port UART ; c'est pourquoi nous avons supprimé le code *BleVspUartBridge* (mis en commentaire). Pour illustrer cet exemple, le texte reçu en Bluetooth sera renvoyé en Bluetooth, mais avec un décalage d'un caractère (j'y reviens ci-dessous).

L'événement de connexion

Dans les articles précédents, nous avons déjà capté l'événement *EVBLEMSG* pour pouvoir traiter les événements de connexion et déconnexion. Dans le programme du thermomètre dans l'article de janvier/février 2015 [1], lors de la connexion, la première étape est de récupérer la valeur de la CTN, la deuxième consiste à l'envoyer au téléphone tactile et la dernière à déconnecter le module du téléphone. Cette fois nous allumerons une LED sur la sortie 2 du module lorsqu'il est connecté ; elle s'éteindra lors de la déconnexion.

N'oubliez pas l'initialisation du port 2 en sortie dans le programme principal. On va copier le Handler *EVBLEMSG* des messages du Bluetooth de la bibliothèque *cli.manager.slib* et créer notre propre handler en le renommant *MyHandlerBleMsg* (code en **bleu**).

Création de notre programme

Pour clore cette énumération des éléments à recycler pour créer ici notre programme *Low Energy*, il reste à évoquer le programme *upass.vsp.sb* utilisé comme base dans d'autres articles pour les services UART [1][4][6].

Celui-ci sera remanié et servira à modifier les données reçues par notre module. Pour cela, nous le renommons *BL600-6.sb* et changeons deux paramètres dans cette première version :

- **ENABLE_DEBUG_PRINTS** : 0 (mode *debug* inactif)
- **DEVICENAME** : «JA_TEST»
- Vous pouvez tester le fonctionnement de votre programme en le transférant à votre e-Bob BL600. Celui-ci sera lancé automatiquement, sinon saisissez la commande

AT+RUN [3]. Vous pouvez alors tester la communication de votre programme avec le programme *SERIAL* de *TOOL-KIT* sur votre téléphone tactile [2].

Les deux TIMERS :

- Code en **vert** :

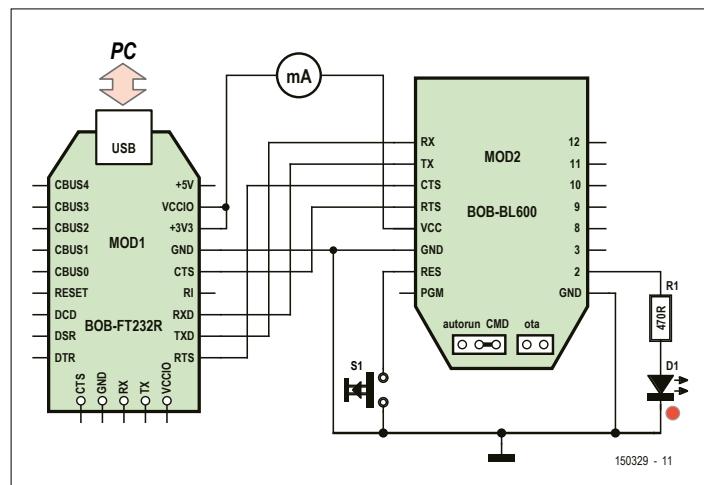


Figure 1. Comme d'habitude, nous utiliserons la passerelle USB-série FT232 d'Elektor [8] pour communiquer avec le BL600.

Liste des composants :

Résistances :

R1 = 470 Ω

Semi-conducteurs :

D1 = LED 3 mm (couleur au choix)

Divers :

S1 = poussoir

MOD1 = e-Bob FT232 assemblé 110553-91 [7]

MOD2 = e-Bob BL600 assemblé 140270-91 [7]

ou circuit imprimé 140270-1 [7]

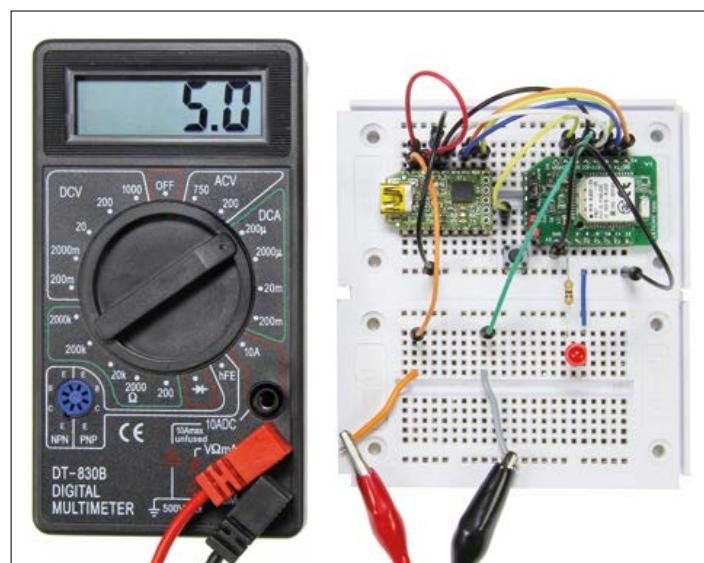


Figure 2. Photo de l'assemblage expérimental sur une plaque d'essais.

- la déclaration de son handle :
OnEvent EVTMR1 Call FuncTimer1
- la fonction : *Function FuncTimer1()*
- allumer la LED : *GpioWrite(2,1)*
- démarrer le Bluetooth : *BleAdvertStart*
- le déclenchement du prochain TIMER : *TIMERSTART(2,25,0)*

● Code en rouge :

- la déclaration de son handle :
OnEvent EVTMR2 Call FuncTimer2
- la fonction : *Function FuncTimer2()*
- éteindre la LED : *GpioWrite(2,0)*
- arrêter le Bluetooth : *BleAdvertStop*
- le déclenchement du prochain TIMER :
TIMERSTART(1,4000,0)

La fonction *BleVspUartBridge* doit être supprimée si nous voulons fermer l'UART et donc consommer moins. Pour attester le fonctionnement du module, nous renvoyons le texte reçu avec un décalage d'un caractère : a devient b, d devient e.... (code en orange).

● Nous avons récupéré le code de la fonction *HandlerLoop* dans la bibliothèque *cli.upass.vsp.splib*. Pour éviter des doublons qui provoqueraient une erreur de compilation, renommons notre fonction *MyHandlerLoop*.

- Les quatre handlers faisant appel à la fonction *HandlerLoop* :

- des données arrivent sur le port UART du module
- des données arrivent sur le port Bluetooth du module
- le tampon du port Bluetooth du module est vide
- le tampon du port UART du module est vide

- La fonction *BleVspUartBridge* crée une boucle entre le téléphone et le port UART. Nous mettons en commentaire cette ligne de code.

- La fonction *BleVSpRead* permet de lire les données reçues par le Bluetooth. Attention, la limite imposée par le fabricant est de 20 caractères.
- La fonction *STRPOS* permet de savoir si le texte reçu se finit par un *carriage return*.
- Pour renvoyer le texte reçu en le décalant d'un caractère, nous utiliserons les deux fonctions *StrGetChr* (conver-

e-BoB BL600

Rappel des caractéristiques du module BL600 :

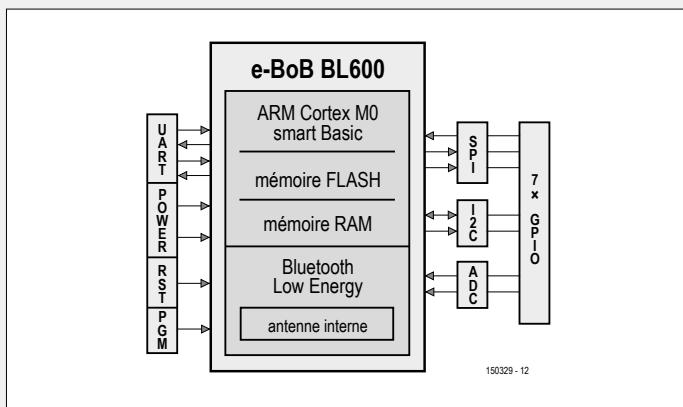
- interfaces UART, I²C, SPI
- 28 entrées/sorties polyvalentes (GPIO)
- 6 entrées analogiques (CAN à 10 bits)
- consommation :
 - 5 µA en veille 0,4 µA en sommeil
 - 10 mA durant l'émission
- programmation aisée en langage smartBASIC
- puce nRF51822 de *Nordic Semiconductor*.
- faible encombrement : 19 x 12,5 x 3 mm
- portée en champ ouvert : jusqu'à 20 m

Bluetooth en BASIC !

Le langage de programmation smartBASIC (orienté événements) du BL6000-SA simplifie l'intégration du Bluetooth dans vos applications en facilitant non seulement la gestion des capteurs connectés directement au module, mais aussi la transmission des valeurs (à nos enfants ☺) mesurées à n'importe quel récepteur Bluetooth v4.0 (téléphone ou tablette tactiles, ordinateur, passerelle...)

Caractéristiques de la carte de liaison e-BoB d'Elektor avec le BL600-SA :

port K1	port série utilisé pour le chargement du programme dans le BL600 broches d'alimentation (3,3 V) ligne de remise à zéro (RESET). broche PGM (éventuelle mise à jour du logiciel du module)
port K2 7 entrées/sorties logiques	2 entrées analogiques à 10 bits (broches 2 et 3) port I ² C (broches 8 et 9) port SPI (broches 10, 11 et 12)
cavalier JP1	cmd : commandes AT autorun
cavalier JP2	OTA (Over The Air)



Le programme et l'application ANDROID

Sur le site du fabricant [7] du BL600, vous trouverez le code source de l'application *Toolkit* pour Android (et aussi iOS) qui comprend les services suivants :

- BPM (pression artérielle)
- HRM (rythme cardiaque)
- Proximity
- HTM (thermomètre médical)
- Serial (UART)
- OTA (Over The Air)
- Batch

Pour prospérer, les réseaux d'objets connectés ont besoin de communication sans fil à faible consommation d'énergie. Cette carte de liaison du module Bluetooth LOW ENERGY est l'accessoire rêvé pour explorer l'IoT.

tir un caractère d'une chaîne à une position voulue en un octet) et *StrSetChr* (changer la valeur d'un caractère par la valeur d'un octet à une position définie).

- Pour finir, le texte sera envoyé en Bluetooth par la fonction *BleVspWrite*.

Le programme n'enverra plus de données sur le port UART, nous pouvons donc clore le port UART, après un délai arbitraire de 1000 ms (code en **mauve**) :

- la déclaration de son handle : *OnEvent EVTMR0 Call FuncClose*
- la fonction : *Function FuncClose()*
- fermeture du port : *UartClose()*

Nous vous avons proposé en téléchargement [6] le code source simplifié à l'extrême d'une application pour téléphone Android utilisant seulement le service UART. Nous avons utilisé *Android Studio*, disponible sous Windows, MAC OS et Linux [7]. Pour rappel, le fabricant a créé une bibliothèque *laird_library_ver.0.18.1.1.jar* afin d'accélérer le développement d'applications Android en Bluetooth normal et *Bluetooth Low Energy*.

Les fonctions décrites dans cette série d'articles :

entrées / sorties :

- paramétrage de la broche du BL600 (*nFunction* = 1 ou 2) :
rc = GPIOSETFUNC(nSigNum, nFunction, nSubFunc)
- lecture du signal 0 ou 1 : *rc = GPIOREAD(nSigNum)*
- écriture (*nnewValue* = 0 ou 1) : *rc = GPIOWRITE(nSigNum, newValue)*

convertisseur analogique numérique :

- paramétrage de la broche du BL600 (*nFonction* = 3) :
rc = GPIOSETFUNC(nSigNum, nFunction, nSubFunc)
- lecture du signal (0 à 1023) : *rc = GPIOREAD(nSigNum)*

port I²C :

- ouverture du port : *rc = I2COpen(nClockHz, nCfgFlags, nHandle)*
- écriture d'une valeur à 8 bits :
rc = I2CWriteReg8(nSlaveAddr, nRegAddr, nRegValue)
- lecture d'une valeur à 8 bits :
rc = I2CReadReg8(nSlaveAddr, nRegAddr, nRegValue)
- fermeture du port : *I2Close(handle)*

- les ports TX et RTS seront mis en sortie avec comme état *High* et *Low* respectivement : *GPIOSetFunc*
- le déclenchement du TIMER dans le programme MAIN : *TIMERSTART(0,1000,0)*

Nous voulons intercepter l'état de connexion pour allumer la LED lors de la connexion ou l'éteindre lors de la déconnexion (en réalité, la LED clignotera du fait des TIMER1 et TIMER2). Le code en **bleu** de la fonction *HandlerBleMsg* vient de la bibliothèque *cli.upass.vsp.splib* :

- Nous avons récupéré le code de la fonction *HandlerBleMsg* dans la bibliothèque *cli.upass.vsp.splib*. Pour éviter des doublons qui provoqueraient une erreur de compilation,

port SPI :

- ouverture du port : *rc = SpiOpen(nMode, nClockHz, nCfgFlags, nHandle)*
- lecture / écriture : *rc = SpiReadWrite(stWrite\$, stRead\$)*
- fermeture du port : *SpiClose(handle)*

Outils pour compiler et transférer

Tous les outils et exemples sont disponibles au téléchargement [7] lorsque vous demanderez l'ouverture d'un compte sur le site du fabricant, spécifiez : « Elektor reader ». Il suffit de télécharger puis de décompresser le document *Firmware Files version 1.5.70.0 - Revision 5* qui contient :

- des exemples de programmes dans le répertoire *smartBASIC_Sample_Apps*
- le programme *UwTerminal.exe* dans le répertoire *smartBASIC_Sample_Apps*
- la bibliothèque dans le répertoire *smartBASIC_Sample_Apps/lib*
- des exemples (*UserManualExampleCode*)
- la configuration spécifique de Notepad++ pour le smartBASIC (*smartBASIC(notepad++.xml)*)

Commandes courantes

- AT I 0 : numéro de la révision du BL600
- AT I 3 : version du logiciel du BL600
- AT+DIR : liste des programmes dans le BL600
- ATZ : remise à zéro du BL600
- AT&F 1 : effacement de la mémoire et redémarrage du BL600
- AT+RUN "xxxx" : exécution du programme xxxx

renommons notre fonction ***MyHandlerBleMsg***.

- Lors de la connexion BLE_EVBLEMSGID_CONNECT :
 - on arrête les TIMERS : *TIMERCANCEL*
 - la LED est allumée : *GpioWrite(2,1)*
 - Lors de la déconnexion BLE_EVBLEMSGID_DISCONNECT :
 - la LED est éteinte: *GpioWrite(2,0)*
 - on relance le premier TIMER : *TIMERSTART(1,10,0)*

Tous les éléments sont à présent réunis pour créer votre programme BL600-6.sb. Après avoir vérifié la faible consommation de notre module, il est souhaitable de modifier la valeur du TIMER référencé 1 en passant de 4000 ms à 300 ms p. ex. une meilleure connexion de votre téléphone tactile : ***TIMERSTART(1,300,0)***.

Le moment est venu de vous lancer dans la programmation de votre BL600, sur votre propre e-BoB, pour votre application ANDROID ; les informations nécessaires se trouvent toutes dans l'article du mois dernier [6].

Conclusion ouverte

Avec ce résumé de ce que nous avons décrit depuis le projet initial du thermomètre Bluetooth, nous fermons cette série d'articles avec une double satisfaction : nos efforts pour faciliter la mise en oeuvre de ce module ont été reconnus et appréciés aussi bien par les lecteurs d'Elektor qui achètent l'e-BoB que par le fabricant du BL600-SA.

D'autres projets basés sur le BL600 attendent dans nos labos respectifs, chez l'auteure et chez Elektor. De plus, le BL600 a un cousin, le BL620, un autre module Bluetooth, mais avec le rôle du *maitre*, qui facilite encore plus la communication. Vous le retrouverez bientôt dans le magazine Elektor... ►

(150329)

```
'//*****  
'// Laird Technologies (c) 2013  
'// Jennifer AUBINAIS (c) 2015  
'// ++++++  
  
'//*****  
'// Definitions  
'//*****  
  
#define AUTO_STARTUP 1  
'//Set this to 0 to disable all debugging messages  
#define ENABLE_DEBUG_PRINTS 0  
#define DEVICENAME "JA_TEST"  
#define DEVICENAME_WRITABLE 1  
  
***** code here *****  
  
'//*****  
'// Library Import  
'//*****  
#include "lib\cli.upass.vsp.splib"  
'//*****  
'// Global Variable Declarations  
'//*****  
DIM text$  
'//*****  
'// Function and Subroutine definitions  
'//*****  
  
'//=====  
'// CLOSE UART  
'//=====  
Function FuncClose()  
    DIM rc  
    UartClose()  
    rc = GPIOSetFunc(21,2,1) '// TX  
    rc = GPIOSetFunc(23,2,0) '// RTS  
    TIMERSTART(1,10,0)  
ENDFUNC 1
```

```

'//=====================================================================
'// Receive data
'//=====================================================================

function MyHandlerLoop()
  //BleVspUartBridge()
  DIM n, rc, tempo$, tx$
  DIM pos, return$
  // Wait return from received data
  tx$ = "0D"
  return$ = StrDehexize$(tx$)
  tempo$ = ""
  n = BleVSpRead(tempo$,20)
  text$ = text$ + tempo$
  pos = STRPOS(text$,return$,0)
  IF ( pos >= 0 ) THEN
    DIM i, x
    pos = pos - 1
    FOR i = 0 TO pos
      x = StrGetChr(text$,i)
      rc = StrSetChr(text$,x+1,i)
    NEXT
    rc = BleVspWrite(text$)
    text$ = ""
  ENDIF
endfunc 1

'//=====================================================================
'// TIMER 1
'//=====================================================================

FUNCTION FuncTimer1()
  dim rc, Adr$
  Adr$ = ""
  // led on
  GpioWrite(2,1)
  rc = bleadvertstart(0,Adr$,25,25,0)
  TIMERSTART(2,25,0)
ENDFUNC 1

```

Liens**[1] Elektor janvier/février 2015 n°439/440**

www.elektormagazine.fr/140190

Thermomètre sans fil Bluetooth Low Energy

Affichage de la température à distance sur votre téléphone tactile

[2] Elektor mars 2015 n°441

www.elektormagazine.fr/140270

e-Bob BL600 : module de communication Bluetooth Low Energy | 1^{ère} partie :

La communication sans-fil sur un plateau

[3] Elektor avril 2015 n°442

www.elektormagazine.fr/150014

e-Bob BL600 | 2^e partie :

Éditer, compiler, transférer un programme avec le module Bluetooth Low Energy

[4] Elektor mai 2015 n°443

www.elektormagazine.fr/150129

e-Bob BL600 | 3^e partie :

Programmer en smartBASIC le module Bluetooth Low Energy

[5] Elektor juin 2015 n°444

www.elektormagazine.fr/150130

e-Bob BL600 : Bluetooth Low Energy | 4^e partie :

Le port I²C et son capteur de température

[6] Elektor juillet/août 2015 n°445/446

www.elektormagazine.fr/150272

e-Bob BL600 : Module Bluetooth Low Energy | 5^e partie :
Port SPI & convertisseur numérique-analogique
Application Android

[7] https://laird-ews-support.desk.com/?b_id=1945**[8] www.elektor.fr/ft232r-usb-serial-bridge-bob-110553-91**

```
'//=====
'// TIMER 2
'//=====
FUNCTION FuncTimer2()
// led off
GpioWrite(2,0)
rc = bleadvertstop()
TIMERSTART(1,4000,0)
ENDFUNC 1
'//=====
'// This handler is called when there is a BLE message
'//=====
function MyHandlerBleMsg(BYVAL nMsgId AS INTEGER,
BYVAL nCtx AS INTEGER) as integer
// Inform libraries
ConnMngrOnBleMsg(nMsgId,nCtx)
AdvMngrOnBleMsg(nMsgId,nCtx)
select nMsgId
case BLE_EVBLEMSGID_CONNECT
    TIMERCANCEL(2)
    TIMERCANCEL(1)
    DbgMsgVal(" --- Connect : ",nCtx)
    ShowConnParms(nCtx)
    // set at High
    GpioWrite(2,1)
case BLE_EVBLEMSGID_DISCONNECT
    DbgMsgVal(" --- Disconnect : ",nCtx)
    // set at Low
    GpioWrite(2,0)
    TIMERSTART(1,10,0)
    ***** code here *****
endselect
endfunc 1
'-----
'// TIME OUT = nothing
'-----
```

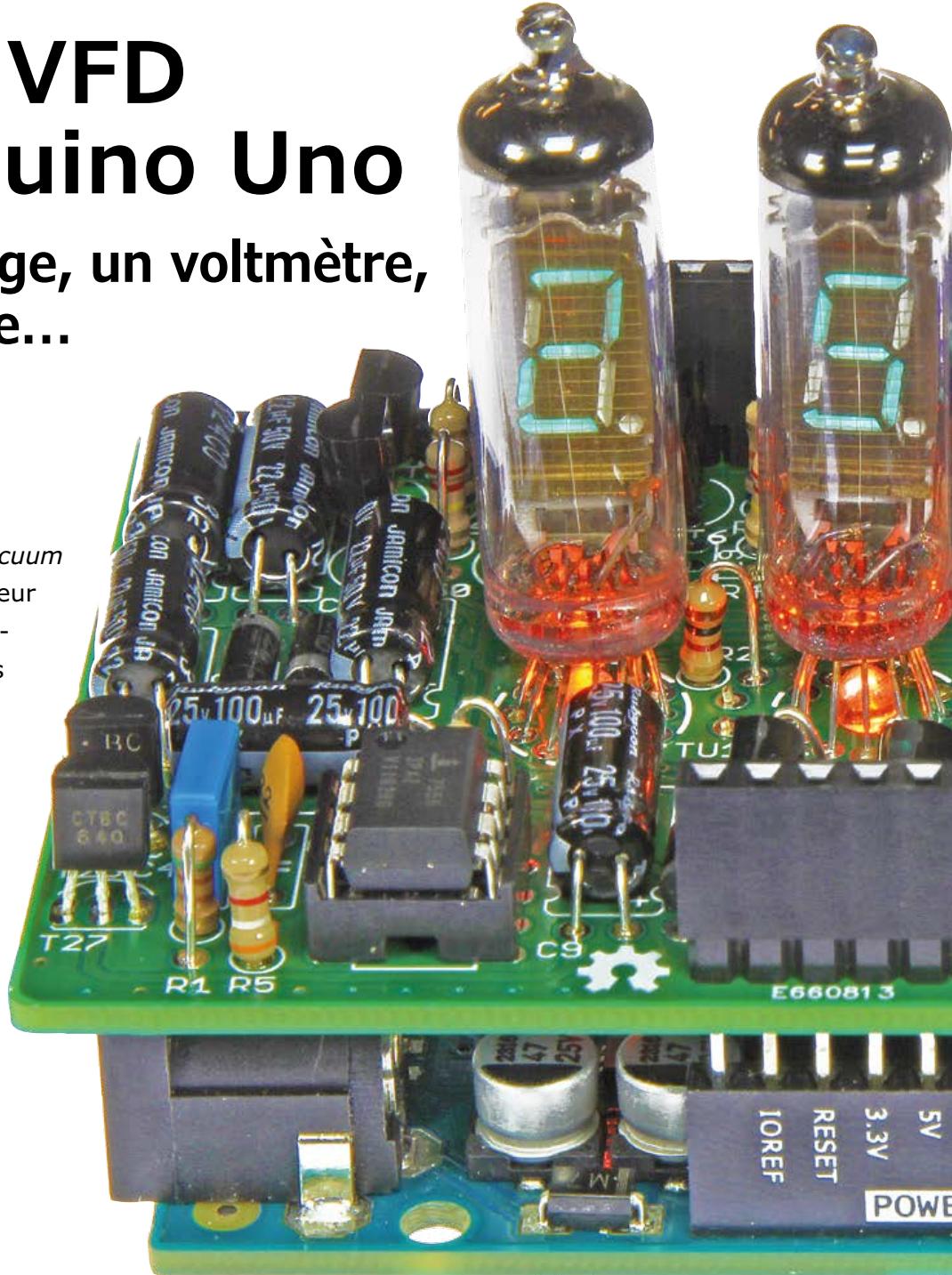
```
Function MyHandlerTimOut() as integer
    'NOTHING
EndFunc 1
'*****
'// Handler definitions
'*****
ONEVENT EVTMR1 CALL FuncTimer1
ONEVENT EVTMR2 CALL FuncTimer2
OnEvent EVTMR0 Call FuncClose
OnEvent EVBLEMSG         call MyHandlerBleMsg //
EVBLEMSG indicate when significant BLE event occurs.
OnEvent EVBLE_ADV_TIMEOUT Call MyHandlerTimOut
//all events have the same handler
OnEvent EVVSPRX          call MyHandlerLoop //
EVVSPRX is thrown when VSP is open and data has arrived
OnEvent EVUARTRX          call MyHandlerLoop //
EVUARTRX = data has arrived at the UART interface
OnEvent EVVSPTXEMPTY      call MyHandlerLoop
OnEvent EVUARTTXEMPTY     call MyHandlerLoop
'*****
'// main
'*****
text$ = ""
// pin 2 output at low
rc = GPIOSetFunc(2,2,0)
rc = bleadvertstop()
IF (ENABLE_DEBUG_PRINTS == 1) THEN
    UartRsp(0)
    TIMERSTART(2,10,0)
ELSE
    PRINT "low Energy"
    TIMERSTART(0,1000,0)
ENDIF
WaitEvent
```

shield à VFD pour Arduino Uno

créez une horloge, un voltmètre, un thermomètre...

Ilse Joostens (Belgique)

Oui, vous connaissez le VFD ! *Vacuum Fluorescent Display*, c'est l'afficheur fluorescent de votre four à micro-ondes, par exemple. Ce n'est pas un Nixie, mais ce type de tube lui ressemble et celui décrit ici est russe. En fait, il y en a quatre sur un *shield* pour carte Arduino Uno, ce qui permet de construire différents appareils. Les logiciels pour horloge, voltmètre et thermomètre sont déjà rédigés et il y a, en prime, un programme de démonstration.



Avec leur air rétro, les tubes Nixie jouissent d'un regain d'intérêt, notamment sur l'internet, mais Elektor qui n'est pas en reste a publié plusieurs projets pour les mettre en valeur. Ce projet n'utilise pas de tubes Nixie à proprement parler, mais des tubes à vide VFD, des afficheurs fluorescents à sept segments. C'est un peu différent et l'engouement qu'ils suscitent va croissant. L'aspect didactique du projet n'a pas été oublié, vous n'y trouverez aucun CMS ni composant pénible à dénicher. D'ailleurs, la plupart des électroniciens enthousiastes

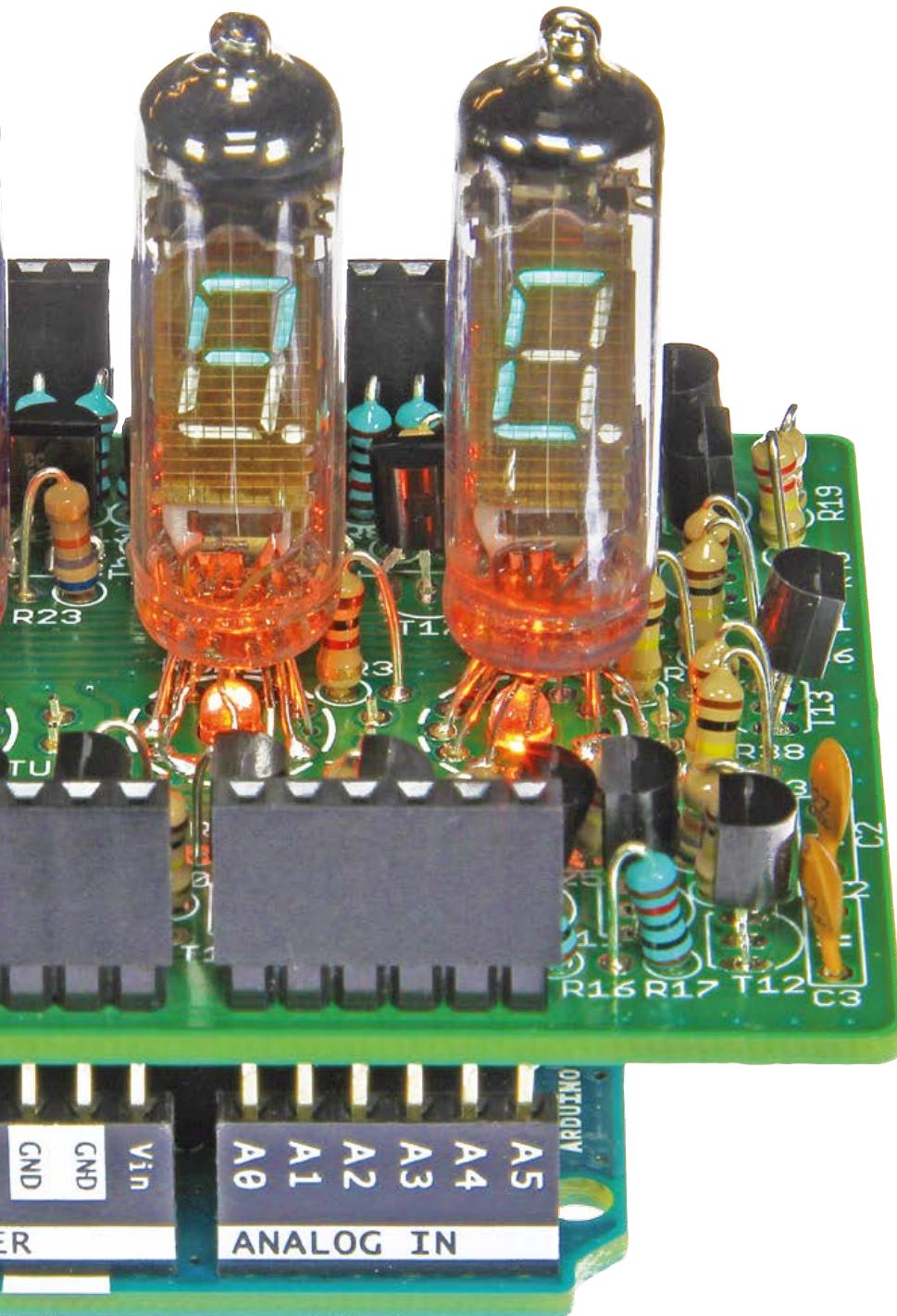
ont certainement déjà chez eux la plupart des composants. En outre, la question de la sécurité ne se pose pas puisque le *shield* ne porte aucune tension dangereuse, tout au plus du 35 V.

Comment fonctionne un tube IV-3 ?

Ce type de tube, de construction plus rudimentaire que les tubes récents - il n'a qu'une grille - remonte à l'époque soviétique. Il est représenté à la **figure 1**, c'est un tube miniature dont le chiffre mesure 8 mm de haut. À l'intérieur de l'ampoule

de verre vide d'air, il y a un filament, une grille, des anodes revêtues de phosphore pour former sept segments et un point décimal, le tout sur un substrat isolant en céramique.

Soumis à un courant électrique, le filament chauffe et émet des électrons. Comme le filament est enduit d'une couche d'oxyde d'un métal alcalin, c'est donc une cathode non isolée, capable de libérer beaucoup d'électrons déjà à basse température ; de ce fait, le filament n'est que faiblement visible et donc ne dérange pas la lecture.



Caractéristiques techniques

- compatible avec les cartes de développement Arduino
- tension d'alimentation de 12 V stabilisés, prélevée sur la carte Arduino
- applications possibles : horloge, thermomètre, voltmètre, compteur, tableau de score...
- exemples de sketch pour Arduino disponibles

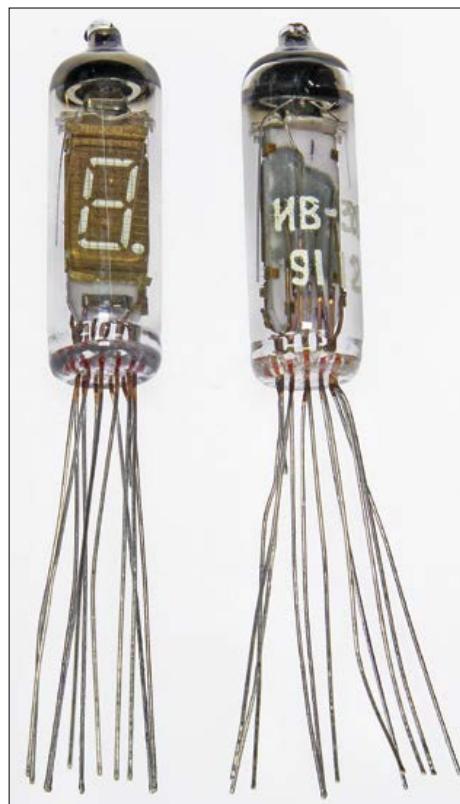


Figure 1. Les faces avant et arrière d'un tube d'affichage fluorescent IV-3.

Dès qu'on applique une tension positive sur une anode du tube, donc sur un segment, le champ électrique créé attire les électrons qui formaient un nuage autour du filament. Ils viennent frapper la couche de luminophore qui, sous l'impact, s'éclaire ici d'une couleur bleu-vert. Le principe est similaire à celui du tube-image, sauf que dans un VFD le phosphore choisi délivre une lumière bien visible avec un faible niveau d'énergie des électrons. Une trentaine de volts suffisent à éclairer les segments, par opposition aux milliers de volts nécessaires dans un tube-image.

Le tube comporte aussi une grille qui, soumise à une légère tension négative par rapport au filament, bloque le flux d'électrons, quelle que soit la tension sur les anodes, comme dans n'importe quelle triode à vide [1]. L'ensemble du tube est alors éteint. La grille d'arrêt permet de multiplexer les tubes VFD comme on le fait avec les afficheurs à LED et c'est exactement ce qu'on fait dans ce projet. Notons aussi la présence dans l'ampoule du VFD d'un fixateur de gaz (*getter*). C'est la couche d'un gris argenté à l'arrière du tube, destinée à capturer par

réaction chimique les molécules gazeuses qui auraient échappé à la pompe à vide ou se libèrent des électrodes pendant le fonctionnement.

Conception du circuit

Dans le circuit de la **figure 2**, on distingue quatre parties : un convertisseur élévateur de tension qui transforme le 12 V d'alimentation en une tension d'environ 32 V, une alimentation pour les filaments, un ensemble de tampons de commande pour les 7 + 1 segments et la grille de chacun des quatre tubes, puis un transis-

tor de commutation et quatre LED pour l'éclairage d'arrière-plan.

Le convertisseur survolteur est en fait un tripleur de tension bâti sur un 7555, un étage tampon et quelques diodes Schottky et condensateurs. En théorie, la tension de sortie devrait s'élever à 36 V sous une alimentation de 12 V, mais chacun sait qu'en pratique, ce sera un peu

moins, en raison aussi de la tension de déchet sur D1 dans la ligne de masse ; on peut compter sur 32 V, assez pour les tubes IV-3 utilisés.

Le 7555, IC1, est monté en oscillateur avec C1 et R5 pour en fixer la fréquence. Un 7555 ne peut pas délivrer beaucoup de courant, aussi y a-t-il derrière lui un étage *push-pull* avec T26 et T27. Il faut à

présent se rendre compte que dans l'état haut, le 7555 ne peut pas fournir une tension suffisante pour pousser T26 à la saturation complète. Raison pour laquelle le 7555 est alimenté à travers R4 sur la tension d'alimentation doublée au nœud D3/D4/C6. En pratique, eu égard à la chute sur R4, sa tension d'alimentation avoisine 14 V, de quoi éléver la base de

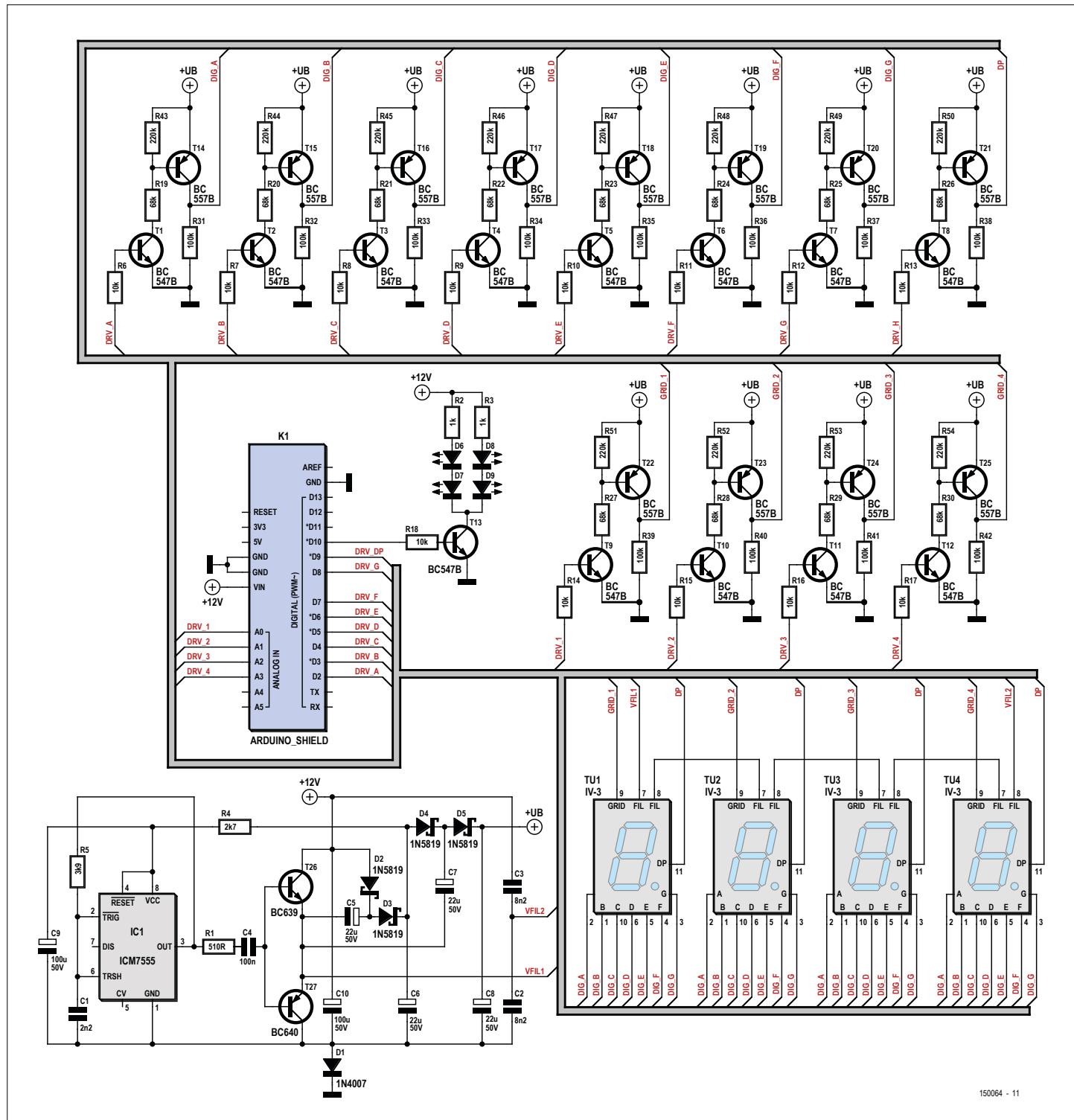


Figure 2. Le circuit de pilotage des tubes VFD se compose d'un tripleur de tension par IC1 et d'une ribambelle d'étages pilotes pour les segments et les grilles (T1 à T12, T14 à T25).



Ne mappelez plus jamais Nixie, je suis VFD !

T26 au-dessus de sa tension de collecteur. Avec pour résultat que la jonction base/collecteur entre en conduction et que la tension de sortie serait limitée par le 7555, s'il n'y avait R1 et surtout C4, lequel restitue une pointe de courant vers la base de T26 au moment où la sortie du 7555 passe du niveau bas au niveau haut. Les condensateurs C5 à C8 du survolteur peuvent ainsi se charger rapidement. Le rendement du circuit en est rehaussé, si bien que les transistors T26 et T27 ne doivent pas être refroidis, même sous forte charge en sortie.

Enfin, la chute de tension d'à peu près 0,7 V sur D1 remonte le niveau par rapport à la masse. Quand la tension de grille d'un tube VFD est ramenée à la masse, sa tension est alors négative par rapport à celle du filament, ce qui éteint à coup sûr l'ensemble du tube.

Impossible dans ces conditions d'alimenter les filaments des tubes en continu, d'autant qu'ils sont tous reliés en série pour des raisons de facilité et d'uniformité de la luminosité. On dispose déjà d'une tension alternative dans le survolteur, il faut l'utiliser judicieusement. D'un côté, on part du nœud commun aux émetteurs de T26 et T27, de l'autre, du nœud entre C2 et C3, qui bloquent la composante continue, mais dont l'impédance permet aussi de conformer l'intensité qui parcourt les filaments. En tenant compte de la fréquence de l'oscillateur déterminée par R5/C1, il s'agit de trouver les valeurs de C2 et C3 appropriées, on devine qu'elles sont assez critiques.

La commande des sept segments et du point décimal utilise chaque fois un circuit identique à ceux de sélection des tubes par la grille. Ils comportent chacun une paire complémentaire de transistors. Avec un niveau haut venant d'Arduino (5 V), le transistor PNP envoie la tension de 32 V sur la grille ou le segment correspondant. Un niveau logique bas entraîne la grille ou le segment à la masse à travers une résistance de 100 kΩ.

Les segments et le point décimal sont reliés en parallèle sur les quatre tubes, de sorte que douze lignes de données suffisent à la commande multiplexée. Ce *shield* n'utilise pas les lignes SDA et SCL,

elles restent donc accessibles pour une autre puce ou une horloge I²C.

Le logiciel

Differentes programmes sont dès à présent disponibles pour la réalisation d'une horloge, d'un voltmètre ou d'un thermomètre, avec aussi un programme de démonstration. Tous sont des sketches Arduino et vous pouvez en télécharger le paquet complet en fichier d'archive sur [2]. Le code source de chacun d'eux est enrichi d'explications détaillées sur le fonctionnement et l'utilisation ; les bibliothèques éventuellement nécessaires y sont jointes.

L'horloge indique la date et l'heure. Comme l'Arduino Uno ne dispose pas d'horloge en temps réel, c'est à vous d'introduire ces données par liaison série. Le plus simple est d'utiliser le moniteur série de l'IDE d'Arduino.

Le voltmètre mesure la tension sur la broche A5 de la carte Uno et en montre la valeur sur les tubes VFD. La plage d'entrée s'étend de 0 à 5 V, représentée par 0.00 à 5.00 V.

Le thermomètre reçoit du capteur par interface à un fil (1-wire) la valeur de la température et l'affiche en degrés Celsius. Le programme est compatible avec les puces DS18B20, DS18S20 et DS1820. Il sonde périodiquement le bus pour savoir s'il est en relation avec une puce compatible, auquel cas il prend en compte la température et montre sa valeur sur les tubes. Ce procédé permet d'échanger les capteurs en cours de fonctionnement. Le programme de démonstration produit une animation sur les tubes et une autre en MLI sur les LED. Il permet aussi de s'assurer que la construction a été réalisée correctement.

Chaque programme comporte une partie de commande des tubes, basée sur une routine de service d'interruption (ISR). On active cycliquement les tubes à tour de rôle. Il n'y a jamais qu'un seul tube allumé à la fois. Voici les étapes à effectuer :

- débrancher aussi vite que possible la grille et les segments du tube actif ;

- attendre 40 µs pour éviter les images fantômes ;
- sélectionner le tube suivant dans la file ;
- adresser les broches 2 à 9 pour adapter les segments ;
- activer la grille du tube.

Si le temps entre la coupure d'une grille et l'activation de la suivante est trop court, il se peut que des segments du premier tube scintillent légèrement sans nécessité. On appelle ce phénomène *ghosting*. Un délai de 40 µs, trouvé expérimentalement, permet de l'éviter. Notez que cette durée est un peu plus longue en réalité, du fait de l'exécution d'autres instructions.

La récurrence du multiplex est de 250 Hz, suffisante pour entretenir la persistance rétinienne. Chaque chiffre est visible 62,5 fois par seconde pendant presque 4 ms.

Les informations à afficher se trouvent dans une table de données structurée à quatre entrées appelée *tube_list*, une pour chaque tube. Chaque entrée contient un drapeau binaire pour l'état du point décimal et un index vers une table qui contient les combinaisons de segments pour former les différents chiffres.

Dans le programme de démonstration, vous trouverez aussi le pilotage des LED en MLI. Chaque cycle est divisé en huit étapes. La fréquence des pas est de 500 Hz, celle de la MLI de 62,5 Hz. À ces fréquences, l'œil ne perçoit pas le clignotement des LED, même avec un rapport cyclique de 1/8.

La routine de service d'interruption (ISR) est associée au temporisateur 1. Le langage de programmation d'Arduino ne prévoit rien pour la gestion des interruptions. Le code source des programmes utilise la bibliothèque Libc d'AVR pour implémenter cette routine en code C. Notez que le langage pour Arduino fait aussi appel à Libc d'AVR.

La construction

Le circuit imprimé de la **figure 3** a été développé pour ce circuit. On n'utilise que des composants traversants, mettant ainsi la construction à la portée de soudeurs débutants.

L'e-choppe d'Elektor vous propose un kit complet avec les quatre tubes. L'assemblage est décrit en long et en large dans une brochure séparée disponible aussi sur [2], mais nous allons ici en parcourir

brièvement les points les plus importants. Tous les composants s'installent sur le côté de la sérigraphie. Commencez par implanter les résistances, diodes, condensateurs et le support pour le circuit intégré. Les résistances se montent toutes verticalement, l'un des fils replié le long du corps. Les condensateurs électrolytiques en revanche sont couchés sur le circuit imprimé, après avoir plié les connexions à angle droit.

Vient ensuite le tour des embases. La bonne idée est de les insérer à leur place et de les connecter provisoirement sur une carte Uno, puis de retourner le sandwich pour souder quelques broches de chaque connecteur. De quoi avoir la certitude que tout sera correctement aligné. On terminera les soudures après avoir retiré la carte Uno.

On s'occupe alors des transistors, puis des quatre LED pour l'éclairage d'arrière-plan qui se glissent dans les trous situés en dessous des tubes. L'endroit où il faut les raccorder n'en est pas bien éloigné, mais il y a lieu de plier les fils à la forme voulue pour qu'ils puissent entrer dans les

trous prévus à cet effet. N'hésitez à faire chaque fois plusieurs essais pour que le tout s'adapte convenablement.

Les tubes VFD s'installent en dernier lieu. Ils sont pourvus de fils de connexion assez longs à enfiler dans le circuit imprimé. Si vous éprouvez des difficultés, coupez les bouts des fils pour donner une forme hélicoïdale à leurs terminaisons : avec des longueurs différentes, l'insertion est plus aisée. Enfoncez-les jusqu'à ce que le bas du tube arrive un rien sous le sommet des embases. Alignez les tubes avec soin et soudez un fil de chacun d'eux. Vérifiez alors l'alignement des quatre tubes et leur verticalité avant de souder les autres connexions, parce qu'après, vous ne pourrez plus rien corriger. Reste à enficher le 7555 comme cerise sur le gâteau.

Passons au logiciel qu'il faut télécharger sur [2] et charger le sketch de démonstration dans l'Arduino via son connecteur USB. On débranche l'Arduino du PC, on colle un morceau de bande isolante sur le connecteur USB pour empêcher qu'il vienne au contact des points de

soudure sur le *shield*. Vous pouvez alors raccorder le *shield* à l'Arduino. Branchez sur l'Arduino un adaptateur secteur qui fournit 12 V stabilisés et un courant d'au moins 300 mA.

Branchez alors l'adaptateur sur le secteur et, si tout est bien monté et soudé, vous verrez les tubes se mettre à compter en boucle sans fin et l'intensité lumineuse des LED varier lentement, preuve que votre *shield* VFD est en ordre de marche. Vous pouvez essayer d'autres exemples de logiciel ou programmer vous-même une application pour le couple Arduino et *shield*. ►

(150064 - version française : Robert Grignard)

Liens

[1] www.youtube.com/watch?v=aiszksJs9C8

[2] www.elektormagazine.fr/150064

Liste des composants

Résistances :

R1 = 510 Ω
 R2, R3 = 1 kΩ
 R4 = 2,7 kΩ
 R5 = 3,9 kΩ
 R6 à R18 = 10 kΩ
 R19 à R30 = 68 kΩ
 R31 à R42 = 100 kΩ
 R43 à R54 = 220 kΩ

Condensateurs :

C1 = 2,2 nF
 C2, C3 = 8,2 nF
 C4 = 100 nF
 C5 à C8 = 22 µF/50 V radial
 C9, C10 = 100 µF/25 V radial

Semi-conducteurs :

D1 = 1N4007
 D2 à D5 = 1N5819 diode Schottky
 D6 à D9 = LED 3 mm, couleur au choix
 T1 à T13 = BC547B
 T14 à T25 = BC557B
 T26 = BC639
 T27 = BC640

Divers :

TU1 à TU4 = tube VFD IV3(a) à 7 segments
 support à 8 contacts pour IC1
 ensemble d'embases à picots compatibles
 Arduino : 1 x 6 picots, 2 x 8 picots,
 1 x 10 picots
 kit complet (avec 4 tubes VFD) : 150064-71*

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

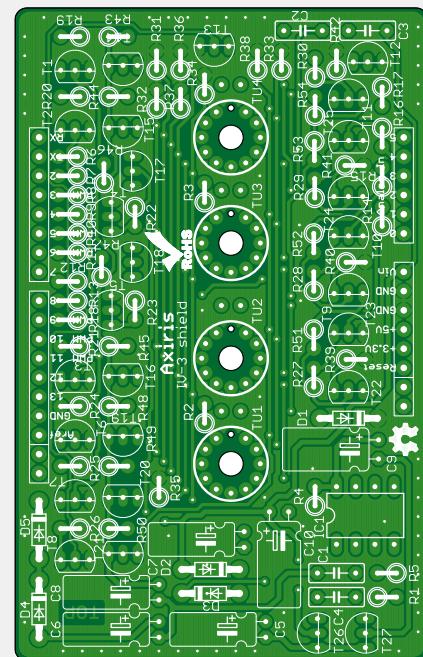
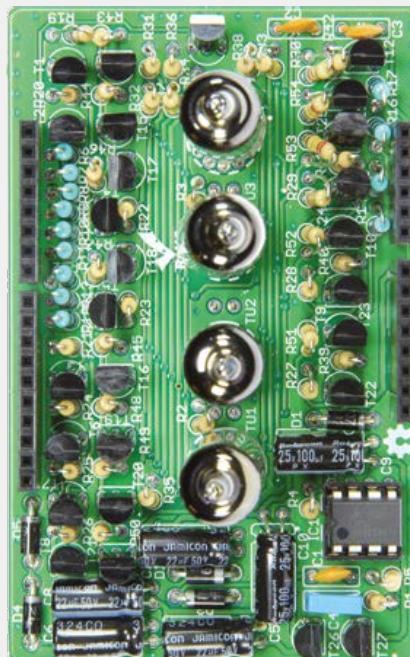


Figure 3. Pas de complication, rien que des composants traversants sur ce circuit imprimé.

bienvenue dans la section **PARTAGER**

PARTAGER

CRÉER

DÉCOUVRIR



Jaime González-Arintero jaime.glez.arintero@eimworld.com

Moi jeu

J'ai récemment regardé un documentaire qui tentait d'expliquer pourquoi nous aimons les jeux. La conclusion était que la vie elle-même est un jeu, mais qu'en l'absence d'un « but du jeu » clairement défini nous aurions besoin d'objectifs pour avancer, objectifs que nous créerions nous-

mêmes si aucun ne nous était donné. Peut-être. Génération oblige, je jouais aux jeux vidéo lorsque j'étais ado. Je les trouvais plus enrichissants que les cours du lycée, guère divertissants à mes yeux. À l'époque les jeux vidéo étaient très chers, et les partager entre amis était considéré comme normal. Pas étonnant alors que cette pratique si répandue ait entraîné l'apparition de protections contre la copie toujours plus complexes. L'accès à l'internet était réservé à quelques privilégiés, mais nous pouvions profiter du réseau de la bibliothèque pour chercher une solution sur un forum ou poser des questions sur un réseau IRC. Une fois rentrés chez nous, seuls face au coffre-fort à ouvrir, nous essayions alors tel ou tel truc hors-ligne. Je ne suis pas très fier de ce passé, mais à l'époque nous n'étions que des gamins. Le piratage était un défi, il fallait être malin et c'était vraiment marrant. Aujourd'hui je paie de bon cœur tout support dont

j'ai besoin, matériel ou numérique.

La plupart des jeux (et des logiciels) pouvaient être piratés en clonant le CD voulu ou en créant un lecteur virtuel. D'autres nécessitaient la réécriture de registres, fichiers et DLL, ou encore le recours à un générateur de clés. Ces méthodes étaient bien connues, et même souvent décrites dans de louches fichiers .zip. Vous suiviez les instructions, et hop, votre jeu était prêt. Parfois tout ne se passait pas aussi facilement. L'installateur pouvait créer une clé d'après l'ID de votre ordi ou à partir d'un numéro de série local et l'incorporer à l'exécutable. Si vous étiez chanceux et si l'application n'était pas chiffrée, vous pouviez alors récupérer la clé avec un éditeur hexadécimal. Dans ce cas, même si vous n'aviez fait que suivre les instructions d'un autre, vous aviez vraiment l'impression d'être le roi des pirates ! Le côté marrant de l'histoire est qu'après avoir « craqué » et installé ces jeux sur ma machine (hors-ligne), je n'y jouais que rarement. Parfois je ne me donnais même pas la peine de les paramétrier, ni d'aller au bout des tutoriels. Jouer était infiniment moins drôle que de craquer le jeu. Le vrai jeu était de faire en sorte que « ça marche ».

Ma vision des choses a commencé à changer à l'université. Les cours avaient du sens et j'avais des « objectifs » à atteindre. Craquer des jeux ne me passionnait plus, pas plus que d'y jouer. Je découvrais que concevoir mon propre système d'aide au stationnement (aussi simple que foireux) ou un gradateur était bien plus amusant que de dégommer des aliens numériques.



Une partie de mon cerveau ne servait pas, Braid l'a activée. Jamais des concepts tels que le temps de propagation n'avaient été si bien représentés.



Si vous ne connaissez pas Portal, essayez-le. Ce jeu merveilleux et hallucinant changera la façon dont vous vous regardez dans un miroir.



Avec Fez, il vous faudra trouver votre chemin et résoudre des puzzles en alternant entre 2D et 3D. Vous pensez l'avoir déjà fait ? Pas de cette façon !

La ludosophie, ça existe ?

Mais là où je voulais vraiment en venir avec ma *ludosophie* de pacotille, c'est qu'on ne fait jamais rien en pure perte. Alors si comme moi vous avez gardé une âme de geek ou de geekette, voici pour vous une courte liste de jeux qui devraient stimuler votre intellect. Vous le voyez, mon désamour pour les jeux vidéo n'aura pas duré. Et, non, ce ne sont pas des jeux comme *The Incredible Machine* ou de simples puzzles qui rappellent la conception de circuits. Et même si vous avez passé une dure journée à router un circuit imprimé, rien ne vous empêche de relever d'autres défis !

Pour voir d'autres copies d'écran, des images plus grandes, des liens actifs, ou pour commenter et partager cet article, rendez-vous sur : [\[http://po.st/playharder\]](http://po.st/playharder). ↗

(150337 – version française : Hervé Moreau)

briques et briquettes

TinkerForge

jettent un pavé dans la marre du prototypage rapide

Clemens Valens (Elektor.Labs)

Pour réaliser le prototype d'un système électronique complexe en peu de temps, il existe des modules spécialement conçus à cet effet. Comme ces modules sont nombreux, il est difficile de dénicher celui qui convient le mieux. Ce sera peut-être le cas des briques et briquettes de TinkerForge.



Le système de test utilisé pour cet article était constitué d'une brique pour moteur à courant continu et de deux briquettes : *Pushbutton* et *Rotary Poti* (potentiomètre).

Le marché du prototypage rapide est en pleine explosion ; voyez le nombre d'outils proposés et ceux en développement. Entre les solutions à bas coût pour amateurs comme l'Arduino et les outils professionnels des grosses multinationales, il y en a pour tous les goûts et tous les budgets. Toutefois ils ont un point commun : rendre le prototypage de systèmes électroniques aussi facile que possible.

L'un des systèmes de prototypage rapide qui a atterri sur mon bureau venait de chez TinkerForge [1], une entreprise allemande naissante qui a bossé dur pour nous faire la totale. Il

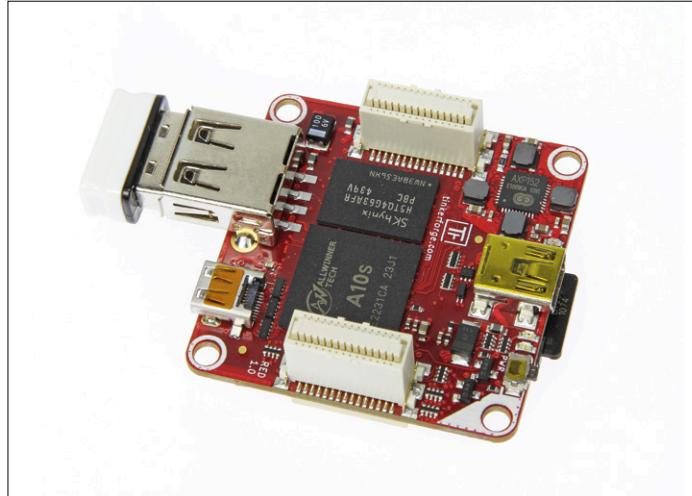
est difficile de m'impressionner, pourtant j'ai apprécié la qualité de construction de leurs modules dès la sortie de la boîte.

Quésaco ?

Les modules TinkerForge sont des briques de base de l'électronique qui peuvent être empilées et reliées pour créer un système complexe. Seuls, les modules ne font rien ; c'est à l'utilisateur d'écrire le logiciel pour faire fonctionner le système. Ce logiciel est une application PC qui peut être écrite dans divers langages (de C/C++ à LabVIEW et Mathematica).

Les modules sont livrés sous forme de briques (*bricks*), briquettes (*bricklets*), extensions et accessoires. Une brique est un module à microcontrôleur doté d'une fonction spéciale (p.ex. pilote de moteur) ; une briquette contient un capteur (p.ex. un potentiomètre) ou un actionneur (p.ex. une LED). Un système comprend typiquement une brique à laquelle sont reliées des briquettes. L'application PC communique avec la brique (envoi de commandes et demande de données) ; la brique communique avec ses éventuelles briquettes.

Important : le PC est indispensable puisque c'est lui qui exécute l'application, pas la brique, à moins que vous n'investisiez dans une brique *RED*, une brique capable de le remplacer. Une brique ne peut gérer qu'un nombre limité de briquettes, qui dépend de la brique, mais empiler des briques permet d'augmenter le nombre de connexions pour briquettes. Les briquettes, en revanche, sont uniquement reliées par câble. C'est logique, puisqu'elles sont habituellement montées quelque part dans le système, près d'un moteur ou d'un panneau de commande, ou autre. L'empilage est aussi utile pour ajouter des fonctions à un système.



La brique RED avec clé Wi-Fi et carte SD.

Les modules d'extension permettent d'interconnecter des empilages et de commander des briques ou des piles à travers un réseau — parfait quand le système est installé dans un endroit isolé ou inaccessible et doit être commandé depuis une salle de contrôle. Enfin, les accessoires sont les connecteurs, boulons et écrous, afficheurs et autres babioles nécessaires pour compléter un système, et disponibles auprès d'autres sources.

Pour qui ?

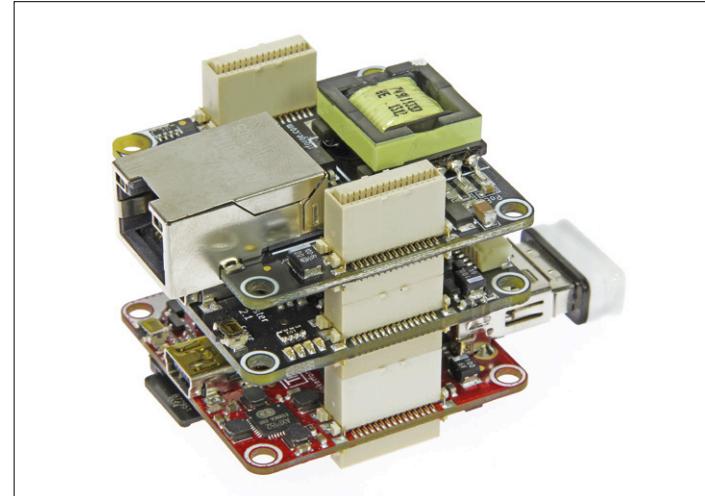
Les modules TinkerForge visent ceux qui souhaitent construire quelque chose qui a besoin d'électronique pour fonctionner, mais qui n'ont pas le temps, ni les compétences ou le désir (toute combinaison est valable) de concevoir leur propre matériel électronique. La tendance est d'appeler les membres de cette espèce des *Makers*, ou *tinkerers* (bidouilleurs), d'où le nom TinkerForge. L'utilisateur n'a pas besoin d'avoir une solide connaissance de l'électronique, mais il devra savoir programmer ; plus il sera à l'aise avec la programmation, mieux ce sera.

Des didacticiels détaillés sont disponibles pour le démarrage, que l'on soit programmeur chevronné ou non. Les applications seront écrites en C/C++, C#, Delphi/Lazarus, Java, JavaScript, LabVIEW, Mathematica, MATLAB/Octave, Perl, PHP, Python, Ruby, Shell ou Visual Basic .NET ; pour les plateformes mobiles, on utilisera C/C++ (IOS), C# (Windows Phone) ou Java (Android). Il y a très peu de produits au monde qui prennent en charge autant de langages de programmation.

Pour quoi faire ? C'est mieux ?

Pour répondre à ces questions, j'ai joué avec le système. J'ai commencé par le didacticiel *First Steps* (premiers pas) du site web, qui utilise la brique *DC*. Il ne s'agit pas d'une alimentation pour une pile, mais d'une brique avec un pilote de moteur à courant continu. Reliez la brique au PC via son port mini-USB ; reliez un moteur c.c. aux connecteurs à ressorts (attention, ils sont raides). Maintenant installez le logiciel :

Brick Daemon (brickd) — un petit programme qui tourne en arrière-plan et s'occupe de la communication entre la brique et l'application de l'utilisateur.



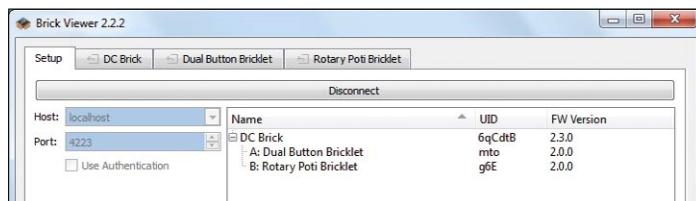
Empilez ! Brique RED au-dessous, brique *Master* au milieu et module d'extension Ethernet au-dessus.

Brick Viewer (brickv) — un outil astucieux qui affiche la configuration du système, quelle briquette est reliée à quelle brique, etc. Il fournit aussi des interfaces pour commander les briques et briquettes.

Brick Daemon démarre automatiquement lorsque vous reliez une brique, tandis que **Brick Viewer** doit être lancé manuellement. Cliquez sur le bouton *Connect* pour établir le contact avec votre système. La brique devrait apparaître dans l'arbre, ainsi qu'un onglet y afférant dans l'interface utilisateur. Le *Viewer* affiche aussi l'UID de la brique, nécessaire pour écrire une application. Cet identifiant unique permet d'adresser un objet dans le système.

En activant l'onglet interface utilisateur de la brique *DC* dans le *Viewer*, j'ai pu commander la vitesse et la direction de mon petit moteur.

La prochaine étape est de vérifier que vous pouvez faire la même chose en écrivant votre propre application. Pour la suite,



Brick Viewer relié à une brique DC, elle-même reliée à deux briquettes. Notez les UID dont vous aurez besoin pour la programmation.

je suppose que vous connaissez les bases du langage de programmation choisi et que les compilateurs/interpréteurs nécessaires sont installés sur votre ordinateur.

Pour commencer installez la couche API de liaison (*binding*) pour votre langage de programmation, voir les instructions détaillées pour votre langage sur le site TF. J'ai choisi Python 2.7, mais Python 3 est aussi pris en charge.

Pour vérifier que tout fonctionne, téléchargez les exemples pour le langage de votre choix. Comme je voulais jouer avec la brique DC et Python, j'ai essayé le script `example_configuration.py`. Vous devez le modifier avec le concours de *Brick Viewer* pour qu'il fonctionne avec votre configuration. Ouvrez le script dans un éditeur et modifiez les paramètres HOST, PORT et UID afin qu'ils correspondent aux informations affichées dans le *Viewer*. HOST et PORT sont probablement corrects, mais pas l'UID (unique à chaque module) ; changez-le et enregistrez le fichier. Lancez le script, le moteur commence à tourner. Appuyez sur <Enter>, le script s'arrête, mais le moteur continue de tourner. Pour le stopper manuellement, entrez les commandes suivantes sur la ligne de commande Python :

```
>>> ipcon = IPConnection()
>>> dc = DC('6qCdtB', ipcon) # Remplacez l'UID par
celui de votre brique
>>> ipcon.connect("localhost", 4223)
>>> dc.set_velocity(0)
```

Les trois premières lignes représentent les étapes nécessaires à l'établissement d'une connexion avec une brique (vous remarquerez la manière dont sont utilisés les paramètres HOST, PORT et UID) ; la dernière ligne stoppe le moteur. N'oubliez pas de fermer la connexion proprement après :

```
>>> ipcon.disconnect()
```

Si vous expérimentez avec la brique DC, n'oubliez pas qu'il vous faut activer le moteur avant de pouvoir le commander (pourquoi ?). Utilisez pour cela la commande `dc.enable()`. Une fois arrivé là, vous êtes prêt à explorer le reste des produits de TinkerForge.

Briquettes

Il existe déjà un nombre considérable de briquettes : E/S analogiques, baromètre, distance, GPS, IO16, joystick... Les briquettes sont reliées aux briques et rien qu'à elles, sauf la brique RED qui n'accepte pas de briquettes. La brique Master accueille jusqu'à quatre briquettes, deux pour la brique DC. Des câbles à dix fils, difficiles à fabriquer soi-même, relient les briques aux briquettes ; il sera plus facile de les acheter, plusieurs longueurs sont disponibles. Mettez votre brique hors

tension avant d'y relier une ou plusieurs briquettes.

J'ai relié la briquette potentiomètre *Rotary Poti* (on voit bien les origines allemandes) et la briquette *Dual Button* à ma brique DC. Après la mise sous tension de DC et la connexion au Viewer, les briquettes sont apparues au-dessous de la brique DC, avec un onglet pour chacune. L'onglet potentiomètre montre un graphique de la position du potentiomètre (en degrés) et l'onglet de la briquette aux deux boutons permet d'activer les LED qu'elle comporte et commander leur fonctionnement ; c'est bien sympa tout ça.

Les exemples fournis facilitent la création d'une application simple qui commande le moteur (potentiomètre pour la vitesse et poussoirs pour marche/arrêt). Les LED des poussoirs servent de témoin visuel. C'est étrange, les poussoirs semblent fonctionner à l'envers ; j'ai peut-être fait une erreur, script d'exemple disponible ici [2].

Extensions

Pour être utile, une pile composée de briques et briquettes est reliée à un PC qui exécute *Brick Daemon* et le programme d'application. Comment commander une pile éloignée de l'ordinateur ? Avec les extensions Tinkerforge. Ces modules permettent à deux piles de communiquer aussi facilement que si le système était sur votre bureau. Il en existe actuellement quatre types : Ethernet, Wi-Fi, RS-485 et Chibi (une pile de communication sans-fil 802.15.4 à code source ouvert qui n'a pas atteint la célébrité espérée par ses développeurs).

Pour utiliser les extensions, il faut la brique Master — la brique DC ne suffit pas. J'ai essayé l'extension Ethernet avec la brique Master ; tout avait l'air de bien se passer, mais je n'ai pas réussi à comprendre comment l'utiliser. L'extension Ethernet fonctionne aussi avec la brique RED. Le site de TF dit que l'on peut utiliser un téléphone ou un autre appareil mobile pour commander la pile sans avoir besoin du *Daemon*, mais il n'y a aucune explication. La documentation manque ici, ainsi que les exemples. J'ai cependant découvert une petite astuce pour déterminer l'adresse IP de l'extension lorsqu'elle opère en mode DHCP, c-à-d lorsqu'elle obtient son adresse IP depuis un serveur sur le réseau. Il suffit de sélectionner *Static IP* pour le paramètre Address et l'adresse courante s'affiche. Notez-la et restaurez la configuration d'origine. N'enregistrez pas la configuration. Comme je n'avais pas d'autre extension à tester, j'en suis resté là.

La brique RED

Il faut un PC pour faire tourner l'application sur les piles de briques. Pour s'affranchir de cette contrainte, TinkerForge a créé la brique *Rapid Embedded Development* ou RED, une brique spéciale capable d'exécuter l'application PC. En fait, il s'agit d'un petit ordinateur Linux (basé sur Debian) capable d'exécuter les applications écrites dans la plupart des langages pris en charge (sauf les langages propriétaires LabVIEW, Mathematica et MATLAB). Le programme peut être écrit, mis au point et testé sur le PC. Une fois prêt, vous le transférez sur la brique RED où il est exécuté sans aucun changement. L'exécution des programmes peut être planifiée (au démarrage, toutes les heures, etc.) et surveillée. Il est même possible d'exécuter plusieurs programmes simultanément.

Si vous avez déjà bricolé un Raspberry Pi ou un autre système Linux embarqué, vous connaissez la complexité et la frustra-

tion qu'occasionne leur configuration. TinkerForge a résolu ce problème pour la brique *RED* en intégrant un panneau de configuration étendu dans le *Brick Viewer*. Grâce à cette interface graphique, il est très facile de configurer une connexion Wi-Fi (il suffit d'ajouter une clé USB Wi-Fi), et de voir ce qui se passe sur la carte. Il y a également une console pour les invétérés de la ligne de commande. L'extinction ou le redémarrage de la carte sont aussi pris en charge ici, mais pas faciles à trouver : il faut utiliser le bouton *System* situé dans le coin supérieur droit de l'onglet de la brique *RED*. Toutes les cartes avec Linux embarqué devraient être accompagnées d'un outil similaire. Une fois la brique *RED* reliée à un réseau, via une clé Wi-Fi ou l'extension Ethernet, et son adresse IP déterminée, vous vous connecterez à l'interface web qui affiche des informations sur les programmes installés. Vous pouvez remplacer cette interface par la vôtre : les applications web écrites en HTML/JavaScript, Python et PHP sont prises en charge.

La brique *RED* possède également une interface HDMI qui, avec un écran convenable, permet d'afficher une interface graphique au démarrage. Notre kit de test était livré avec un écran tactile, mais le capteur tactile ne fonctionnait pas, malgré la LED qui indiquait que tout devrait fonctionner. Peut-être un réglage manquant quelque part ?

Gardez le contact

Au début du travail avec briques et briquettes, il est fortement recommandé de visiter régulièrement le forum de discussion général [3] où sont annoncées les mises à jour logicielles et matérielles. Le forum est très actif. Lorsque j'ai commencé à préparer cet article, j'ai téléchargé *Brick Viewer* 2.2.2, deux mois plus tard la dernière version était 2.2.5 et les micrologiciels des briques avaient été mis à jour plusieurs fois.

Sources ouvertes

Pour ce qui est des briques, briquettes et extensions, les sources sont mises à disposition. Par conséquent, vous avez accès au code source de chaque module ainsi qu'aux schémas.

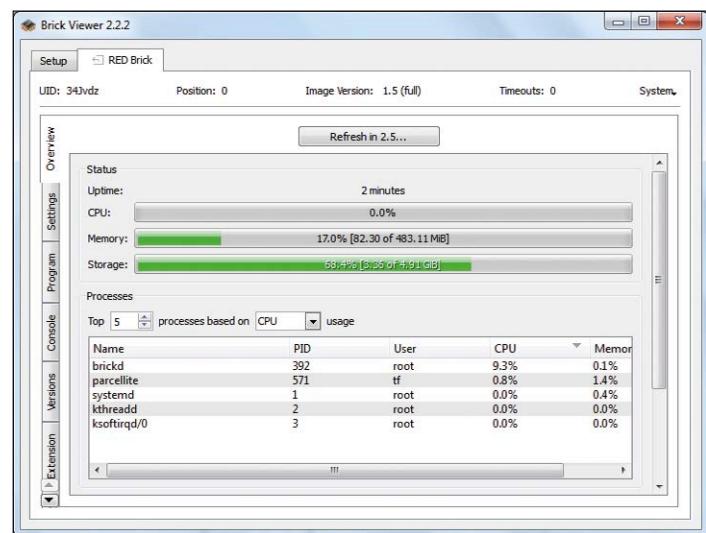
Conclusion

Maintenant que nous avons vu ce qu'il est possible de faire avec les modules TinkerForge nous pouvons répondre à la question en suspens : est-ce mieux ? Cela dépend de votre application. Personnellement, j'ai été impressionné par ce système de prototypage rapide. Il prend en charge suffisamment de langages de programmation pour que quiconque a programmé au moins une fois dans sa vie puisse y trouver chaussure à son pied. Il y a beaucoup de documentation bien écrite et des exemples à profusion. Il y a bien quelques imperfections dans le *Viewer* et, oui, on est un peu seul face aux fonctions avancées, mais on peut toujours communiquer avec d'autres utilisateurs ou l'équipe de TinkerForge sur TinkerUnity.org. Vous avais-je dit que les composants sont de première qualité ? Chapeau bas à TinkerForge ! ►

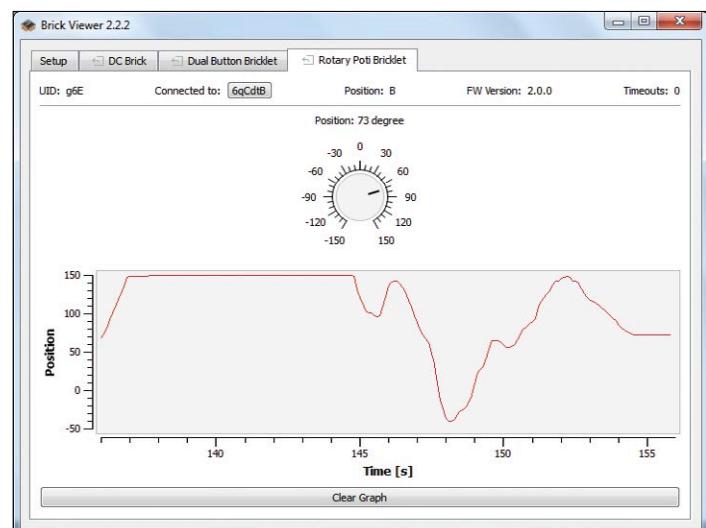
(140543 – version française : Kévin Petit)

Liens

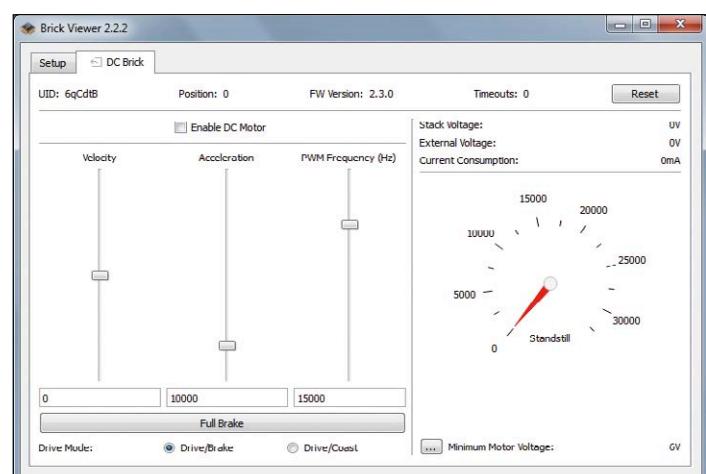
- [1] www.tinkerforge.com
- [2] www.elektormagazine.fr/140543
- [3] www.tinkerunity.org/forum/index.php



L'interface de configuration Linux de la brique *RED* dans le *Viewer*. La liste des programmes actifs inclut le *Brick Daemon* *brickd*.



Vous pouvez voir ici l'état du potentiomètre de la briquette.

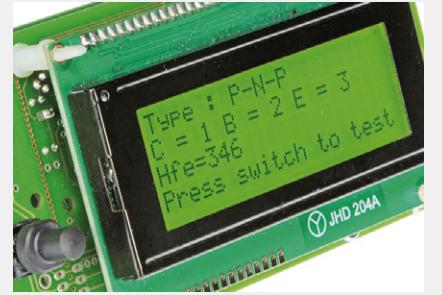


L'interface utilisateur pour la brique *DC* dans le *Viewer* vous permet de commander la vitesse et le sens de rotation du moteur.

Testeur de transistors Platino

Elektor mars 2015, page 62 (130544)

ADDENDA. L'article ne fait qu'effleurer le rôle du transistor T1. Une fois construit le testeur selon le schéma, ce composant ne revêt pas d'importance particulière du point de vue de l'utilisateur. T1 sert à l'auto-calibrage du testeur, sous le contrôle du port PA3. La façon dont il opère est évidente si l'on se réfère au code source en BASCOM AVR. L'article ne fait pas non plus référence à la fréquence du quartz de la carte Platino ; elle doit être de 8 MHz (ce que le code source du projet permet également de vérifier). [www.elektormagazine.com/130544]

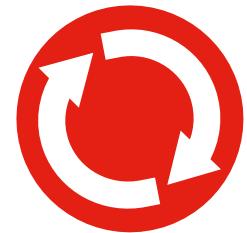


électronique

corrections, mises à jour et compléments d'articles

Compilé par Jaime González-Arintero

Au menu de cette session : ajouts, coquilles, étourderies, et même un fait historique erroné ! Nous nous efforçons bien sûr d'atteindre le « zéro faute », mais la perfection n'étant pas de ce monde... Heureusement, vos yeux attentifs sont là pour repérer cette première petite erreur, ce deuxième petit oubli, cette troisième négligence, ce... hum... bref, merci à tous pour vos contributions !

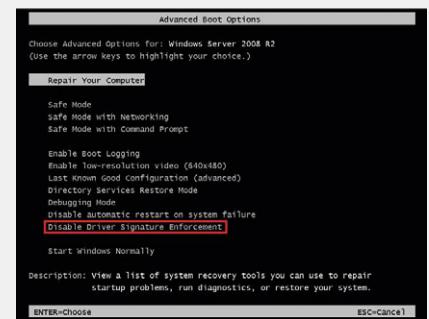


Analyseur UART/RS-232

Elektor avril 2015, page 38 (140126)

MISE À JOUR. Sous Windows 8 et versions ultérieures, il se peut que l'installation du pilote USB pour l'analyseur échoue. Le problème vient du « Contrôle obligatoire des signatures de pilotes » de Windows qui bloque le pilote (il n'est pas signé). La désactivation de ce contrôle obligatoire est malheureusement assez compliquée, mais vous trouverez sur Internet de nombreux tutoriels décrivant la marche à suivre. Téléchargez la rustine dont le lien est donné ci-dessous : avec le Contrôle obligatoire des signatures de pilotes désactivé, le pilote devrait s'installer sans broncher.

[<http://po.st/disableDSE>]



Synthétiseur J²B, plateforme musicale numérique ouverte

Elektor janvier & février 2015, page 88 (140182)

MISE À JOUR. Clemens Valens, chef d'orchestre d'Elektor.Labs, a agrémenté son populaire synthétiseur J²B de trois nouveaux micrologiciels, dont l'un transforme le J²B en boîte à rythmes (rudimentaire). Ces trois programmes sont des « portages » des codes de l'Atmegadrum, de l'Atcyclotron et du Delayertron de Soulsby Synthesizers. Avec l'Atcyclotron original, le synthétiseur J²B regroupe donc désormais quatre instruments uniques en un seul. L'archive contenant les quatre micrologiciels peut être téléchargée depuis la page du projet Elektor.Labs, ou via le lien : [<http://po.st/J2Bv3>]



Convertisseur USB / 4 UART, RS-232/422/485

Elektor novembre 2014, page 36 (140033)

CORRECTION. Sur le schéma, les numéros des broches 2 et 3 du régulateur LM2937 (IC2) ont été inversés. Le brochage correct est : 1 (INPUT), 2 (GND), 3 (OUTPUT). Le schéma corrigé est téléchargeable depuis le lien ci-dessous. Cette erreur n'affecte ni les cartes assemblées ni le reste des fichiers.

[<http://po.st/140033schematic>]



µC ARM pour néophytes, passer de 8 bits à 32 bits (3)

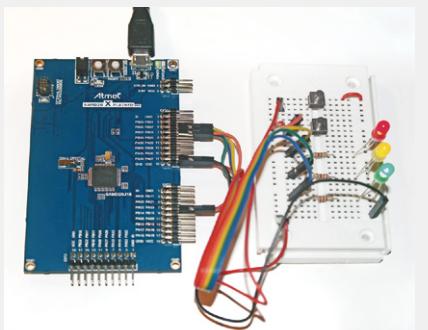
Elektor avril 2015, page 10 (150041)

CORRECTION. Une simple erreur de lettre peut avoir de grands effets ! Sur le listage 1, parmi les fonctions de configuration du WDT et de l'EIC, figurent deux instructions identiques (situées dans la seconde partie du code et séparées par quatre lignes). La lettre désignant le port est incorrecte dans les deux cas : le port doit être « A », et non « B ». Ces deux instructions doivent donc être écrites :

```
config_extint_chan.gpio_pin_mux = MUX_PBxxA_EIC_EXTINTx;
```

Les lettres « x » représentent le numéro de la broche et le numéro du canal. Vous pouvez corriger vous-même les deux lignes, ou télécharger le code corrigé depuis la page web associée à cet article. C'est l'auteur qui a repéré l'erreur, donc merci à Viacheslav Gromov !

[www.elektormagazine.com/150041]



Convertisseur USB-série multiprotocole

Elektor avril 2015, page 70 (130542)

CORRECTION. L'encadré *Qu'est-ce qu'un port série ?* explique que la norme RS-232 a pour origine les économies que l'on cherchait à réaliser en reliant par liaison distante des terminaux à une unité centrale. En réalité la norme RS-232 spécifiait à l'origine une interface entre un équipement terminal de traitement de données (ETTD, p. ex. un serveur) et un équipement terminal de circuit de données (ETCD, c.-à-d. un matériel de transmission de données comme un modem). Cette interface permettait de relier directement par câble croisé deux ETTD.

Merci à **Ernst Stippl**, de Vienne, d'avoir rétabli la vérité historique !



Commuter un port AVR (trucs & astuces entre lecteurs)

Elektor avril 2015, page 9 (150027)

ADDENDA. Cette astuce est relativement ancienne puisque les premiers AVR dont les broches pouvaient être commutées de la sorte datent d'avant 2005. La fiche technique de l'ATTiny13 donne l'information dans *I/O Ports -> Ports as General Digital I/O -> Toggling the Pin* (section parfois aussi intitulée *Reading the Pin*). Les registres des broches font normalement partie des 32 premiers registres d'E/S (adresses inférieures à 0x20) et sont donc encore accessibles bit par bit. L'expression *PINB.5=1* du code est convertie par BASCOM en une commande SBI. Si la mémoire flash de l'AVR est supérieure à 8 Ko et a une table **IVT** (*Interrupt Vector Table*) d'une largeur de deux mots, on peut p. ex. commuter directement des ports au moyen d'une interruption dans la table IVT elle-même. Si la broche est définie comme entrée, la résistance interne de rappel vers le haut peut être activée et désactivée selon le même procédé. Toutefois, pour certaines séries AVR (d'environ 2007 à 2010), ces résistances de *pullup* ne sont pas activées ou désactivées via les registres de direction de données ou les registres de ports, mais par leurs propres registres d'activation.

Une broche de µC peut être indépendante des registres en raison d'une autre fonction (comme un *reset*) mais néanmoins rester accessible (ça marche avec p. ex. l'ATTiny 4/5/9/10), ce qui permet d'effectuer des opérations XOR, p. ex. pour calculer ou vérifier un bit de parité.

Merci à **Stefan Rosenthal** pour cette précision !

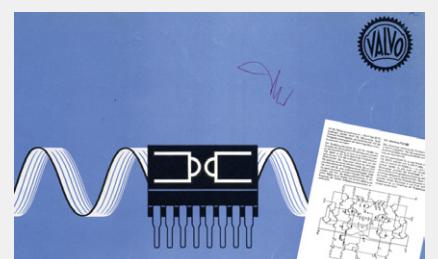


Gyrateur intégré TCA580

Elektor avril 2015, page 23 (150024)

ADDENDA. **Christoph Kessler** nous a fort aimablement envoyé la note d'application allemande et les cinq pages de la fiche technique du gyrateur intégré TCA580. Ces documents datant de 1974, les nostalgiques ne manqueront pas d'y jeter un œil. Les deux fichiers peuvent être téléchargés depuis le lien ci-dessous. Merci encore Christoph !

[<http://po.st/TCA580DE>]



(150288 – version française : Hervé Moreau)

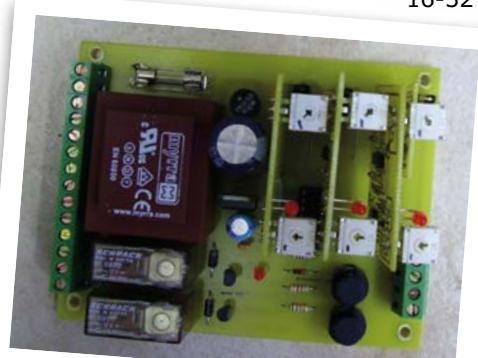
.LABorama

véritablement (f)estival

Si l'électronique est mondiale, les saisons sont locales. Difficile donc de savoir si partout cet été notre globe n'aura été qu'une boule de glace léchée par le soleil, mais pour ce qui est des fers à souder à la fête sur elektor-labs.com, ils auront été brûlants. La preuve avec ce florilège de projets, terminés ou en cours, mais toujours formidables. L'automne sera-t-il encore plus chaud ? Qui sait !



<http://po.st/astroclock>



<http://po.st/levelsensor>

Il est temps d'en savoir un peu plus

Cette horloge astronomique fournit pléthore d'informations basées sur les coordonnées de votre position actuelle et les affiche sur pas moins de quatre afficheurs ! Heure GMT, phases lunaires, date, ville, saison, température, signe zodiacal, humidité, pression... L'horloge modifie même son aspect en fonction du cycle diurne. Futurs maîtres du monde, cette horloge est pour vous !

Et voici... notre nouveau batteur. Il est analogique.

Bien sûr, les logiciels de production musicale sont pratiques, mais rien ne remplacera jamais la sensation qu'on éprouve en ajustant ses propres rythmes en tournant de vrais boutons. Cette boîte à rythmes intègre un séquenceur à 16-32 pas et un contrôle MIDI supplémentaire, mais les douze sons différents sont entièrement analogiques. Nostalgique des années 80 ?



<http://po.st/beatbox>

J'ai foi en mon capteur de niveau

Vous cherchez un capteur de niveau de liquide digne de confiance pour surveiller un puits ou un réservoir ? Ce montage comporte un module émetteur et trois récepteurs, qui reçoivent des alertes émises par trois sondes (haut, bas, et niveau critique/alarme). En outre, ce capteur peut activer une pompe pour éliminer l'eau d'un sous-sol inondé, ou pour automatiser le contrôle du niveau d'un réservoir p. ex. Si vous n'avez besoin que d'une alarme à plusieurs niveaux, la première partie du circuit peut fonctionner seule.

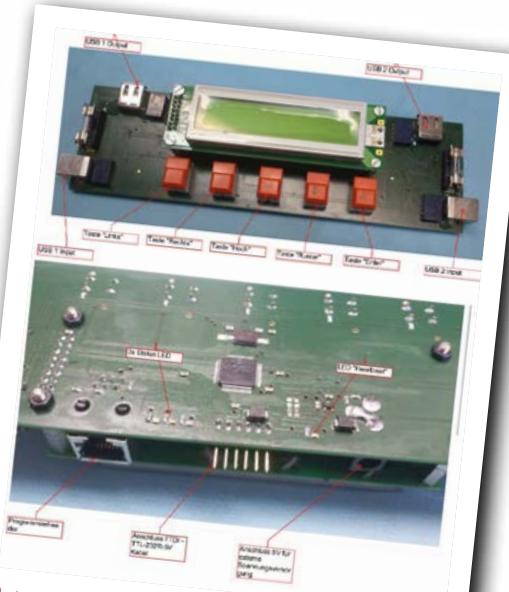




<http://po.st/1qinabox>

USB, ne dépasse pas les bornes !

Cette unité se révèlera bien pratique pour garder sous contrôle des ports USB. Elle peut mesurer et limiter le courant de deux ports en fonction de valeurs que vous pouvez définir, et empêche de façon efficace les surintensités. Les valeurs en temps réel sont affichées sur l'écran, mais sont également disponibles pour analyse via le connecteur pour câble FTDI USB/TTL 5 V.



<http://po.st/USBcontrol>

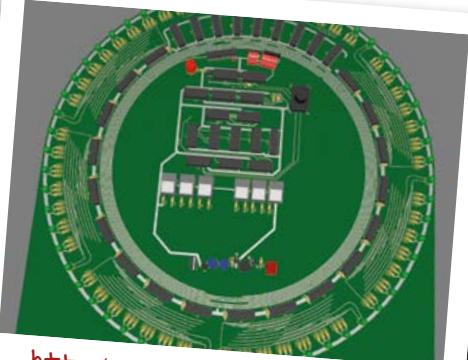
Couteau suisse électronique

Votre projet nécessite le recours à plusieurs fonctions, mais vous ne les voulez pas éparsillées sur plusieurs cartes ? L'avantage avec l'électronique, c'est qu'il est toujours possible de créer son propre couteau suisse ! C'était le cas du projet « 13 in a box », qui devient « 19 in a box » puisqu'entre-temps il a été enrichi de six fonctions, dont la mesure de tension, la commande de servos, le décodage DCF77, la commande de LED NeoPixel, un buzzer... En attendant les prochaines !

8 transistors pour en obtenir 1

Entendez par là : pour obtenir une radio AM. Ce projet amusant, gratifiant et par ailleurs idéal pour se frotter au monde de la RF, montre comment assembler et tester une radio AM superhétéodyne complète. Les quatre premiers transistors composent les étages de réception (y compris l'oscillateur local), les quatre autres forment un

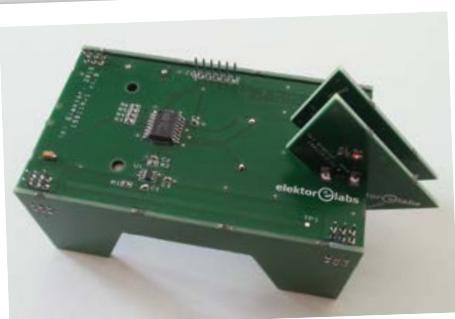
amplificateur simple à trois étages. Le circuit peut être alimenté par quatre piles AA, et donc être incorporé dans un appareil portatif.



<http://po.st/TTLclock>

Horloge à TTL

Ce projet est l'histoire d'une horloge numérique qui voulait devenir analogique, mais ne pouvait fonctionner qu'avec des circuits intégrés TTL. Pas d'aiguilles ici, mais des LED disposées en cercle. Pas de microcontrôleur non plus (trop facile, trop tendance !), mais l'horloge n'en reste pas moins configurable au moyen de commutateurs DIP qui permettent de modifier la forme des « LED-aiguilles » selon un système de codage. Une horloge numérique à l'âme analogique et zéro code, mais 100 % cool !



<http://po.st/hacktor>

Une carte qui a du chien

Hacktor, la plateforme de développement modulaire en forme de chien origamiesque, peut désormais aboyer ! Et si elle peut s'exprimer lorsque son capteur de proximité AtmelQtouch détecte votre présence, c'est grâce à un circuit intégré de lecture audio qui envoie un « wouf-wouf » dans le haut-parleur situé sous sa plate bedaine. Hacktor est programmable via son connecteur SPI et facile à incorporer dans un projet. Et en plus il fait bien sur un bureau, non ? ↗

(150338 – version française : Hervé Moreau)

sélection de fichiers Gerber du labo

Thijs Beckers (Elektor Labs)

Depuis peu, quelques fichiers Gerber pour la fabrication de circuits imprimés sont proposés à la vente sur www.elektor.fr. Que pouvez-vous en faire ?

Un fichier Gerber contient toutes les informations nécessaires à la fabrication d'un circuit imprimé. Il comprend la description des diverses couches du circuit imprimé, telles que les couches conductrices, la sérigraphie des composants, ou le masque pour la pose de vernis, et toutes les autres données requises pour la production du circuit, par exemple pour le perçage. Le format du fichier est celui d'une image vectorisée à deux dimensions. La spécification actuelle est la RS-274X version 2 (Gerber étendu) ; la spécification RS-274-D (Gerber standard) est désormais obsolète.

Le format a été créé par la société Gerber Systems Corp., fondée par Joseph Gerber, d'où le nom. Il fut conçu pour piloter les tables traçantes destinées à l'industrie des circuits imprimés dans les années 60 et 70. Gerber était le leader du marché, le format est donc devenu un standard de fait ; et il l'est resté. Nous avons décidé de mettre à la disposition de nos lecteurs certaines données de conception du labo, dont une partie des fichiers de fabrication de nos circuits imprimés – très appréciés pour leur qualité – via le site www.elektor.fr. Nous espérons ainsi favoriser la reproduction des projets Elektor, et encourager la communauté grandissante des « faiseurs » (*makers*).

Pour les fichiers Gerber proposés à la vente, vous êtes libre de choisir le fabricant de vos circuits imprimés, qui seront tels que ceux produits par le labo d'Elektor. Après tout ce sont les mêmes fichiers que ceux que nous utilisons pour nos prototypes et la production. Vous ne devriez donc avoir aucun mal à obtenir des circuits imprimés de qualité professionnelle chez n'importe quel fabricant de circuits imprimés.

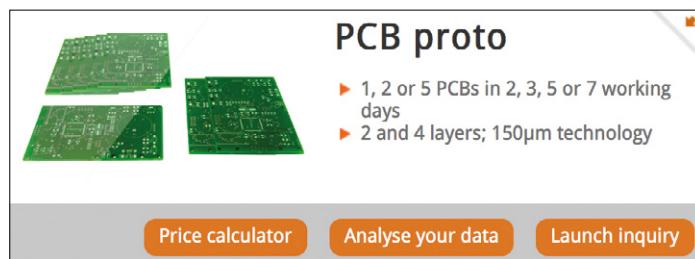
Notez que tous nos circuits ne sont pas mis à disposition sur le site. La sélection est faite par le labo, et il est préférable que vous utilisez le mot-clé « Gerber » pour rechercher sur le site les fichiers disponibles.

Pour la commande, nous ne pouvons pas vous détailler la procédure à suivre chez chacun des fabricants du marché. À titre d'exemple, nous vous expliquons ci-après la marche à suivre pour Eurocircuits, un de nos fournisseurs « testé et approuvé ». La procédure ne devrait pas être trop différente chez les autres fabricants, bien que les services d'Eurocircuits soient très étendus ; toutes leurs options ne seront peut-être pas disponibles chez les autres. ▶

(150342 – version française : Jean-Louis Mehren)



Étape 1. www.eurocircuits.com : « Price Calculator ».



Étape 2. Puis « Analyse your data ». Pour un seul circuit, vous pouvez utiliser « PCB proto » ou « NAKED proto ».



Étape 3. Pour associer vos données personnelles (adresses d'expédition et de facturation par exemple) à vos fichiers Gerber, vous devez vous enregistrer sur le site (ou vous identifier, si c'est déjà fait).

Analyse your data

PCB name *	<input type="text" value="120272-1"/>
Upload data file	<input type="button" value="Choose File"/> <input type="file" value="120272-1 ...s v1.0.zip"/>
Purchase reference *	<input type="text" value="First batch"/>
Article reference	<input type="text" value="Elektor Universal Pream"/>
Project reference *	<input type="text" value="My Board"/>
Delivery term	<input type="button" value="7 Working days"/>
Quantity *	<input type="text" value="2"/>
<input type="button" value="Continue"/>	

* Mandatory field.

Étape 4. Après identification, soumettez votre projet et les données des circuits imprimés, et téléchargez vos fichiers Gerber (compressés dans un fichier zip ou équivalent).

Price calculator Shopping basket Checkout items

Shopping basket – To start the manufacturing process, select an item(s) and "Proceed to checkout" !

Proceed to checkout **Modify** **View details** **Files** **Delete** **History** **Download PDF offer** **Edit administrative details** **Ask a question**

Number Item name Service Status Created d **Search**

Items ready for checkout

	PCB Visualizer®	Offer	Number	Type	Status	Item name	Service	Quantity	Delivery days	Unit price	Net price	Stencil
<input type="checkbox"/>	PCB Visualizer®	B0433027	PCB	Ready to checkout	120272-1	PCB Proto	2	7 Working days	33.68	67.36	-	

You can order all jobs with status 'Ready for Checkout' immediately, even when the PCB Visualizer process is still running or failed.

Items in analysis

No record(s) found.

Visualizer link in [Visualizer] column lets you view PCB layers visually, during processing you will not be able to modify your data files.

Étape 5. Vous cliquez ensuite sur « Continue » pour vérifier vos données. Via « PCB Visualizer », vous aurez un aperçu du produit fini (Eurocircuits offre de nombreuses options de visualisation). C'est très utile pour déceler d'éventuelles erreurs ou difficultés lors du processus de fabrication. Confirmez vos choix (vous pouvez ajouter d'autres circuits à votre commande) et validez la commande (« Proceed to checkout »).

Price calculator Shopping basket **Checkout items**

Checkout items

Back **Submit** Add new address

Delivery address
Elektor International Media - 1, AlleeKasteel Limbricht, Limbricht - 6141 AV, Netherlands

Invoice address
11, Postbus, Susteren - 6114 ZG, Netherlands
Elektor International Media BV

Étape 6. N'oubliez pas de vérifier vos adresses (expédition et facturation) avant de confirmer une dernière fois (« Submit »). Vos circuits imprimés de qualité professionnelle devraient être produits et livrés sous peu !

The screenshot shows the Eurocircuits PCB Visualizer interface. On the left, there are several tabs: PCB Configurator, PCB Checker, Customer data, PCB proto, Stencils, Material, Technology, and PCB definition. The PCB proto tab is active, showing delivery format (Single PCB), PCB quantity (2), delivery term (7 working days), PCB width (90.00 mm), PCB height (90.00 mm), and board name (120272-1). The Board buildup section shows a 2-layer board with top and bottom copper layers. The Summary section provides a quick overview of the order: Service (PCB proto), Delivery term (7 working days), Estimated shipment date (13-07-2015), Quantity (2 PCBs), Board surface / Order surface (0.81 dm² / 1.62 dm²), Prices (Single PCB: € 33.68, Total boards: € 67.36, Express transport: € 0.00, VAT 0.00%, Total gross: € 67.36), and a 'Save changes' button. The Bird's Eye View section shows the physical layout of the PCB with various components and traces. A note says: "Click the 'Launch inquiry' button in case you are having troubles configuring your PCB. Our sales team will review your input and generate an offer." Below this are sections for Alternatives (2 PCBs, 7 working days, Net: € 33.68, Select; 5 PCBs, 7 working days, Net: € 88.81, Select; 10 PCBs, 7 working days, Net: € 124.55, Select) and a Customized matrix.

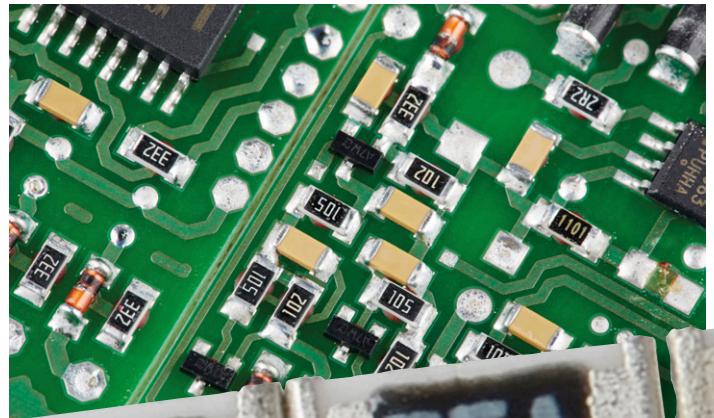
La page de visualisation d'Eurocircuits montre les détails de la production de votre circuit imprimé ; elle permet de détecter et corriger d'éventuelles erreurs.

codes CMS

À quel composant avons-nous affaire ?

Harry Baggen (rédition Pays-Bas)

Les composants sont de plus en plus petits, par conséquent il y a toujours moins de place pour indiquer leur type ou leur valeur. Les fabricants utilisent donc un code pour identifier le composant, code qui n'est pas évident à déchiffrer. Quelques sites sur la toile – décrits ci-après – facilitent heureusement l'identification de ces composants.



Les circuits électroniques actuels n'accueillent pratiquement que des composants montés en surface (CMS, SMD – *Surface Mounted Devices* en anglais), afin de diminuer les coûts de production et la taille des appareils. Les composants traversants standard sont encore disponibles, mais l'offre se réduit de jour en jour. En outre, les fabricants de semi-conducteurs ont toujours à leur catalogue beaucoup de (nouveaux) composants, mais uniquement en boîtiers pour montage en surface. En tant qu'amateurs, nous n'avons donc guère le choix : il faut s'adapter aux CMS.

Dans les années 90, il est apparu que les CMS allaient jouer un rôle de plus en plus prépondérant dans l'industrie, et de nombreux électroniciens furent persuadés que c'en était fini des prototypes et petites séries. Mais tout n'était pas si noir : avec de la patience, quelques essais et de nouveaux outils, il était tout à fait possible de manipuler (au sens propre) des CMS. La taille des composants s'est encore réduite depuis, mais cela n'empêche toujours pas la construction d'un prototype ou la réparation d'un circuit. Il est vrai que beaucoup d'électroniciens ont dû changer de paire de lunettes et acquérir une loupe – voire un microscope – pour travailler avec ces composants de (très) petites dimensions.

La petite taille pose cependant un autre problème : l'impression des caractéristiques sur le boîtier ou le corps du composant. Les fabricants n'ont rien trouvé de mieux que de recourir au marquage d'un code cryptique, donc difficile à déchiffrer. Lorsque vous commandez des composants neufs, les indications sur l'emballage sont claires ; ce n'est pas le cas pour des composants en vrac ou soudés sur un circuit imprimé, et

souvent il n'est pas évident de déterminer leur type ou leur valeur. Nous avons donc recherché quelques sites où ces codes sont expliqués et traduits.

Résistances et condensateurs

Commençons par les résistances : le code utilisé comprend trois ou quatre chiffres, en fonction de la précision de la valeur de la résistance. La ressemblance avec les résistances traditionnelles est là, les couleurs sont remplacées par des chiffres : les deux ou trois premiers indiquent la valeur, le dernier le facteur de multiplication. La plupart des résistances d'une précision de 1% sont cependant marquées suivant le système EIA-96, avec trois caractères : deux chiffres, qui n'indiquent pas directement la valeur, et une lettre qui représente le facteur de multiplication. Sur le site *Your Hobby-Hour* [1], vous trouverez un tableau avec les valeurs correspondant au code EIA-96, ainsi

EIA-96 system									
		Resistor Code C		Code Value		Code Value		Code Value	
1	38C	01	100	17	147	33	215	02	102
		02	102	18	150	34	221	03	105
		03	105	19	154	35	226	04	107
		04	107	20	158	36	232	05	110
		05	110	21	162	37	237	06	113
		06	113	22	165	38	243	07	115
		07	115	23	169	39	249	08	118
		08	118	24	174	40	255	09	121
		09	121	25	178	41	261	10	124
		10	124	26	182	42	267	11	127
		11	127	27	187	43	274	12	130
		12	130	28	191	44	280	13	133
		13	133	29	196	45	287	14	137
		14	137	30	200	46	294	15	140
		15	140	31	205	47	301	16	143
		16	143	32	210	48	309		
		Multiply factor		Resistance = 243·100		Code Multiply			
		Significant values		1 Z 0.001		2 YR 0.001			
		Multiply factor		X 0.1		XS 0.1			
		Resistance = 243·100		A 1		B/H 10			
				D 1000		E 10'000			
				F 100'000					
www.resistorguide.com your guide to the world of resistors									

Comme la place disponible sur le corps des composants se réduit à peau de chagrin, les fabricants n'ont rien trouvé de mieux que de recourir à un marquage cryptique.

qu'un « calculateur » qui traduit automatiquement le code en valeur de résistance. Vous retrouverez aussi tableau et calculateur sur le site resistorguide.com [2], ainsi qu'une vidéo qui explique les points importants du code.

La situation est un peu plus complexe pour les condensateurs. Pour les gros, comme les chimiques, il y a suffisamment de place pour indiquer la valeur et la tension en clair sur le composant. Pour les plus petits, on emploie quelquefois le même code que pour les résistances [3], ou bien il n'y a aucun marquage (c'est aussi le cas pour les petites inductances) ! Il faut donc dessouder et mesurer la valeur avec un capacimètre ; quant à la tension de service, elle demeure inconnue. La couleur du boîtier correspond parfois à une plage pour la valeur du condensateur, c'est suffisant dans certains cas pour éviter la mesure [4].



Semi-conducteurs

Pour les petits CMS à deux ou trois broches, voire plus (diodes, transistors et petits circuits intégrés tels que ampli op ou régulateur de tension), la détermination du type exact est un peu plus compliquée. On retrouve sur la plupart des boîtiers deux ou trois caractères, pour le type du composant, et souvent quelques caractères plus petits qui indiquent la date de fabrication et le numéro de lot.

Au vu du nombre de composants différents sur le marché et du peu de données sur les boîtiers à cause du manque de place, une identification directe et sans ambiguïté est tout simplement impossible. Divers sites donnent des vues d'ensemble et des tableaux dans lesquels on retrouve les différentes combinaisons de caractères avec les types et fabricants possibles ; mais attention : à une combinaison donnée peut parfois correspondre un choix de vingt possibilités ! Il est alors nécessaire de bien comparer le boîtier et le marquage avec les feuilles de caractéristiques des fabricants, et éventuellement d'étudier le circuit où le composant est utilisé, afin d'en déterminer le type exact. Les sites ci-après constituent une aide précieuse pour cette quête.

Codes beginning with 'C'					
Code	Device	Manufacturer	Base	Package	Leaded Equivalent/Data
C	BB365	Sic	I	SCD80	uhf varicap 2-20pF
C white	BAT165	Sie	I	-	40V 750mA sw Schottky
C	KV1832E	Tok	I	URD	uhf varicap 4-17pF
C0	HSMS-2820	HP	C	SOT23	HP2835 schottky
C0	HSMS-282B	HP	C	SOT323	HP2835 schottky
C1	HSMS-2821	HP	K	SOT23	HP2835 schottky
C1	BCW29	Phi	N	SOT23	BC178A
C1p	BCW29	Phi	N	SOT23	BC178A
C1t	BCW29	Phi	N	SOT23	BC178A
C1	BFQ51C	Phi	CX	SOT173	pnp BF90A complement
C2	BCW30	Phi	N	SOT23	BC178B
C2p	BCW30	Phi	N	SOT23	BC178B
C2t	BCW30	Phi	N	SOT23	BC178B
C2	BFQ32C	Phi	CX	SOT173	pnp 4.5GHz 15V 100mA
C2	HSMS-2822	HP	D	SOT23	dual HP2835 schottky
C2	HSMS-282C	HP	D	SOT323	dual HP2835 schottky
C2	SST112	Tem	F	SOT23	J112 analog sw n-ch jfet
C2A	ZDC833A	Zet	B	SOT23	dual cc 28V varicap 15pF @2V
C3	HSMS-2823	HP	A	SOT23	dual HP2835 schottky
C3	BFQ23C	Phi	CX	SOT173	pnp complement BF91A
C3	SMBT4126	Sie	N	-	2N4126
C3	SST113	Tem	F	SOT23	J113 analog sw jfet
C4	BCW29R	Phi	R	SOT23R	BC178A
C4	HSMS-2824	HP	B	SOT23	dual HP2835 schottky
C5	MMBA811C5	Mot	N	-	2N5086 pnp hfe 135,-20

Après quelques pages d'explications générales, le document de quatre-vingts pages *The SMD Codebook* (le livre des codes CMS) [5], nous confronte à de nombreux tableaux remplis de codes. À chaque code correspond le type de composant possible, le fabricant, le modèle de boîtier, et quelques caractéristiques. Plusieurs sites dont [5] proposent ce document en version PDF ; pour ceux qui préfèrent une version HTML, il y a le site de Roger Blackwell, GM4PMK [6].

Une autre aide utile est fournie par le site *S-Manuals* [7]. Dans la section *Marking SMD* (marquage CMS), on retrouve les diverses combinaisons de deux caractères. Un clic sur une combinaison mène à un récapitulatif des composants possibles, c'est-à-dire ceux qui ont une relation avec ladite combinaison. Le type de composant, le modèle de boîtier et le fabricant sont bien entendu repris ; un plus intéressant : un lien direct vers la feuille de caractéristiques !

Pour les plus gros circuits intégrés, avec quelques dizaines de broches, la tâche n'est heureusement pas si ardue : il y a assez de place sur le boîtier pour y indiquer lisiblement le type de composant.

Bonne chance pour l'identification de CMS inconnus !

(150339 – version française : Jean-Louis Mehren)

Liens

- [1] www.hobby-hour.com/electronics/smdcalc.php
- [2] [www.resistorguide.com/resistor-smd-code/](http://resistorguide.com/resistor-smd-code/)
- [3] www.radio-electronics.com/info/data/capacitor/smd_capacitor.php
- [4] https://en.wikipedia.org/wiki/Surface-mount_technology
- [5] www.sos.sk/pdf/SMD_Catalog.pdf
- [6] www.marsport.org.uk/smd/mainframe.htm
- [7] www.s-manuals.com/smd

Erratum : dans l'article « Hackaday, pour quoi faire ? » du numéro de mai 2015, p. 78, le projet *Red Pitaya comme SDR* n'est pas développé par Brian Benchoff mais par Pavel Demin.

The screenshot shows the homepage of S-Manuals.com. At the top, there's a search bar with 'Search' and a 'Google' button. Below the search bar is a sidebar with links to various sections like Home, Components, Books, and Downloads. The main content area features a large table titled 'The first 2 symbol marking SMD'. The table has columns for symbols and their corresponding codes. The first few rows show examples like '0 0' through '9 9' and various letters (A-Z). The table continues with many more entries, each with a small image of the symbol.

Compilées par **Beatriz Sousa**

Chaîne logistique renforcée



Nous nous attendions à ce que notre campagne de fin d'an-née *Le Père Noël est devenu fada* vous enthousiasme, mais pas à ce que notre campagne d'été *Tropicool* déclenche une telle affluence autour de l'e-choppe ! Nous avons donc décidé de renforcer notre chaîne logistique, ce qui se traduira à partir de ce mois-ci par des délais de livraison plus courts. Pardon à ceux d'entre vous dont les commandes n'auraient pas été livrées aussi vite qu'ils l'espéraient.

Un cahier de vacances écrit au fer à souder

Cet été, notre expert Bart Huyskens a animé en Belgique un atelier de robotique pour enfants. Stimulés par l'inimitable enthousiasme pédagogique de Bart et émerveillés par les cartes de commande d'Elektor.LABS, voici à n'en pas douter de futurs membres d'Elektor !



READ ONLY MEMORY

Le magazine Elektor et son éditeur sont fiers de leur histoire. Cette rubrique montre pourquoi.

Le **microcontrôleur 8052AH-BASIC d'Intel** aura été l'Arduino des années 1980. Il inspira deux projets Elektor populaires : la mono-carte SCALP, basée sur le 8052AH-BASIC (1987), et la mono-carte fondée sur les 8032 et 8052 (1993). De nombreux lecteurs s'initieront à l'embarqué grâce à elles. La SCALP (Système de Conception Assisté par un Langage Populaire) a été rétronisée dans le numéro de mars 2007.



- INDISCRÉTIONS** • C.J. Abate, de notre revue sœur Circuit Cellular, a aidé Elektor à lancer la Muhammed Söküt, en charge des membres allemands, proposera des partenariats et des produits germanophones DRONE • Margriet Debeij, responsable de la clientèle allemande, invite les des « rapports spéciaux » dans le prochain numéro d'**Elektor Business...**



The screenshots show:

- Screenshot 1:** The homepage featuring a search bar with 'Arduino' highlighted, a sidebar with categories like 'Arduino', and a news article about the 'Arduino en esclave I²C' project.
- Screenshot 2:** A search results page for 'Arduino en esclave I²C' showing the project details and a thumbnail image of the hardware.
- Screenshot 3:** An article page for 'Arduino en esclave I²C' with a sidebar for 'COMPILE THÉMATIQUE Arduino'.
- Screenshot 4:** A compilation page titled 'ARDUINO PAGE DE COMPILE THÉMATIQUE' featuring a large image of a multi-tube vacuum fluorescent display (VFD) and several news items related to Arduino.

Compilation thématique

La version rénovée et améliorée du site **elektormagazine.fr** a été très bien accueillie. Vous êtes actifs sur cette nouvelle plaque tournante et en utilisez les nombreuses nouvelles options. Outre l'accès aux magazines, nouvelles et articles, vous disposez désormais de nouveaux moyens d'exploration, comme les récentes compilations thématiques.

Une compilation thématique est une page où ont été rassemblées pour vous des informations et des publications relatives à un thème donné, Arduino ou Raspberry Pi par exemple. La page de compilation thématique devrait vous permettre de trouver plus facilement certains types de contenus depuis une même page, comme tous les projets inédits .POST de fin de semaine ou encore le meilleur d'Elektor.TV. C'est un moyen novateur et efficace pour explorer le meilleur d'Elektor !

Les compilations thématiques sont construites à partir des différents mots-clés attachés aux publications. Une partie de ces mots-clés apparaît déjà dans un nuage affiché dans la colonne de droite de toutes les pages du site.

Pour accéder à la compilation thématique d'Arduino par exemple, entrez Arduino dans le champ réservé à la recherche, validez (**écran 1**). Les résultats de la recherche sont affichés (**écran 2**).

Cliquez sur l'un des articles proposés (**écran 3**). Les deux bannières précédées de  permettent d'accéder à la compilation effectuée autour du mot-clé Arduino (**écran 4**).

Tout cela est pratique et facile à utiliser. Nous sommes ouverts à toute suggestion, et souhaiterions connaître les thèmes, sujets ou contenus que vous aimeriez voir associés à une compilation thématique.

**Envoyez-nous vos suggestions à
webmaster@eimworld.com !**

campagne **Tropicool** aux États-Unis et au Canada • aux réseaux périphériques tels que les communautés entreprises situées en et hors Allemagne à publier

PROFIL D'EXPERT

Elektor est au cœur d'un réseau de plus de 1 000 experts et d'auteurs engagés dans la publication de livres, d'articles, de DVD, de webinaires et autres événements. Coup de projecteur !



Nom : **Jason Long**

Âge : **37**

Études : **génie électrique** (licence)

Publications : **vaste collection de tutoriels en ligne**

Enseignement : **PIC, ARM, ANT+, BLE, LabVIEW, conception de systèmes embarqués basse consommation et sécurité intrinsèque**

Qui êtes-vous, Jason Long ?

Un Canadien passionné par les systèmes embarqués. J'ai fondé la société Engenuics Technologies en 2010 et cela fait 15 ans que j'enseigne bénévolement l'embarqué. J'ai deux fils déjà fascinés par les LED clignotantes !

D'où viendra l'évolution majeure de l'électronique ?

Je pense que les réseaux personnels et leur accès par NFC et des protocoles radio à très basse consommation comme ANT+ et BLE vont faire exploser l'Internet des Objets. Les spécialistes de l'embarqué (matériel et logiciel) seront très recherchés.

Et récemment, qu'est-ce qui a fait progresser l'électronique ?

La famille ARM Cortex a joué un rôle significatif ces dernières années, en particulier pour les systèmes sur puce comme le nRF51 de Nordic, qui offre tout le nécessaire et du sans-fil à basse consommation (non, je ne travaille pas pour Nordic !)

En matière d'innovation, qu'est-ce qui distingue le Canada des États-Unis ?

Peut-être sommes-nous mus par la passion, alors que les Américains auraient pour muse le capitalisme ? Quoi qu'il en soit et quel que soit le pays, l'innovation vient du désir de relier les gens au moyen de bons produits et services.

Si je vous donnais un bon d'achat de 500 € pour l'e-choppe, qu'achèteriez-vous ?

Oh, ce serait un super cadeau ! Les quadrirotors figurent en haut de ma liste des trucs à construire un jour, donc je choisirais le Crazyflie 2.0.

Qui est le concepteur que vous admirez le plus et pourquoi ?

Je dois être un peu bizarre parce que je n'ai jamais eu de modèle. J'admire simplement les amateurs et les ingénieurs qui mettent tout leur cœur à comprendre comment fonctionnent les choses et à les réaliser.

Quel est le projet dont vous êtes le plus fier et pourquoi ?

Je travaille depuis plusieurs années à l'évolution de nos cartes de développement Razor et Blade, et je pense qu'elles sont solides, qu'elles comblent le fossé entre le monde amateur ou universitaire et celui de l'industrie. ►

(150344 – version française : Hervé Moreau)

bienvenue dans votre e-choppe

Elektor recommande



Quadrioptère CrazyFlie 2.0

Le CrazyFlie 2.0 est un petit quadrioptère à code source ouvert. Son intérêt n'est pas seulement dans sa courbe d'apprentissage raide, spécialement pour ceux qui ont déjà leur brevet de nullité (autrement dit, on apprend vite), mais justement parce qu'il vous invite aussi à rentrer dans le code.



Ses dimensions (14 cm entre la pointe de ses pales) sont à elles seules un défi pour la stabilité du vol, mais dès que vous aurez capté les astuces élémentaires de pilotage avec votre tablette ou votre téléphone tactile, vous ne le lâcherez plus. L'autre défi, c'est la vitesse : 30 km/h, c'est rapide, surtout à l'intérieur !

Le site www.bitcraze.io. donne toutes les infos. Vous y trouverez même à télécharger l'image complète d'une machine virtuelle. Tous les paramètres sont personnalisables. Ne commencez peut-être pas par là, mais un jour vous pourrez vous attaquer au firmware !

Thijs Beckers, Elektor Labs



www.elektor.fr/crazyflie-2-combo

Vos favoris :

1. Smartscope
www.elektor.fr/top1

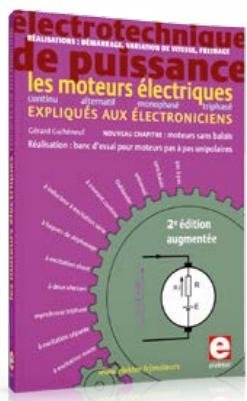


2. Raspberry Pi Mod. B+
www.elektor.fr/top2
3. Arduino Sensor Kit
www.elektor.fr/top3
4. clavier capacitif à 12 touches
www.elektor.fr/top4
5. Maîtrisez les µC à l'aide d'Arduino
www.elektor.fr/top5
6. Trousse d'outils pro IFIXIT
www.elektor.fr/top6
7. Arduino Uno
www.elektor.fr/top7
8. L'électronique pour les débutants
www.elektor.fr/top8

Les moteurs électriques expliqués aux électroniciens

DVD Elektor 2014

Extension ISP pour Arduino

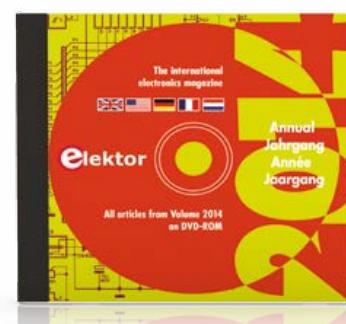


Les électroniciens découvriront ici les moteurs électriques, tandis que les électrotechniciens admettront qu'électronique n'est pas synonyme de complexité. L'information est digeste : constitution, fonctionnement, caractéristiques, domaines d'utilisation, réalisations simples et concrètes. Le lecteur mesurera ses connaissances grâce à des questionnaires d'évaluation (avec corrigés).



Prix (membres) : 39,50 €

www.elektor.com/moteurs

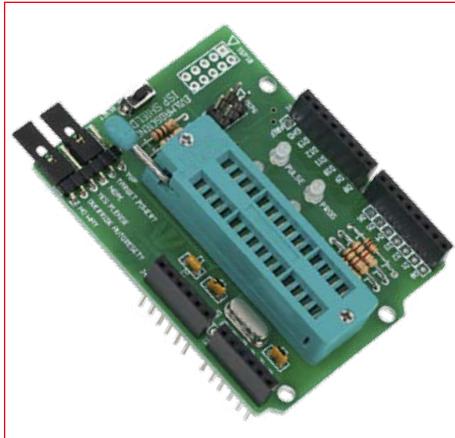


Ce DVD-ROM réunit tous les articles d'ELEKTOR parus au cours de l'année 2014, en français (mais aussi en anglais, allemand et néerlandais). Il contient le texte des articles ainsi que les schémas, et tous les dessins des circuits imprimés, sous forme de fichiers à haute résolution. Une fonction de recherche dans la table des matières vous permet de trouver immédiatement l'article souhaité.



Prix (membres) : 24,75 €

www.elektor.fr/dvd-2014



Ce kit de composants à souder comporte un circuit imprimé ainsi que tous les composants requis pour assembler un programmeur ISP (*in-system programmer*) pour cartes Arduino (et compatibles, comme Diavolino). Ce shield permet de programmer entre autres des microcontrôleurs AVR vierges directement depuis l'environnement Arduino pour y écrire un programme d'amorçage (bootloader). Le kit est fourni avec un support à force d'insertion nulle (ZIF) à 28 broches pour



Prix (membres) : 17,96 €

www.elektor.fr/isp-shield-arduino



I'intégrale des articles d'Elektor en format PDF

249 pages bourrées d'idées, d'explications, d'astuces, de schémas, de programmes, de circuits imprimés...

Des heures de lecture instructive et stimulante !

Cette compilation réunit l'intégrale des articles parus dans Elektor entre juillet 2012 et novembre 2014.

Les liens dans la table des matières de ce document numérique (PDF) permettent de naviguer facilement vers les articles qui vous intéressent.

Compilation Arduino

Imagine ce que tu désires, souhaite ce que tu imagines, tu finiras par créer ce que tu veux.

Avec Arduino. Avec Elektor.

interface de diagnostic OBD USB KKL

kit de diagnostic Red Pitaya

Prix (membres) : 9,50 €

www.elektor.fr/compilation-arduino-e-book

Kit *Three fives* – un modèle agrandi du temporisateur 555



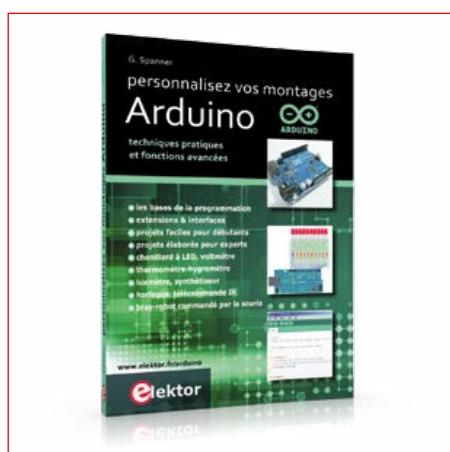
Le kit *Three fives* est une réplique exacte et fonctionnelle du temporisateur intégré NE555, l'une des puces les plus populaires de tous les temps. Il ressemble fidèlement à un circuit intégré (agrandi), avec un support en aluminium qui imite les broches du boîtier DIP original à huit broches. Ce kit permet surtout de reproduire, avec des transistors, le fonctionnement du célèbre temporisateur et de l'étudier pour en comprendre la structure interne et d'expérimenter.



Prix (membres) : 35,96 €

www.elektor.fr/trois-555

Personnalisez vos montages Arduino



L'objectif de ce livre est de vous emmener à pas guidés vers la maîtrise d'Arduino. Les projets sont regroupés par thème, avec des bases théoriques. Vous apprendrez à exploiter des techniques essentielles (conversion analogique-numérique, modulation de largeur d'impulsion, pilotage de différents types d'afficheurs, interface I2C, interruptions).



Prix (membres) : 31,05 €

www.elektor.fr/personnalisezArduino

kit d'extension Arduino (notamment RFID)



Ce kit réunit des composants électroniques qui vous permettront d'enrichir vos projets à partir d'Arduino. Une des possibilités offertes est la communication sans fil en mode RFID. Rien de bien sorcier, mais une fonction aux applications innombrables.

La liste des composants montre que RFID n'est pas la seule extension possible avec ce kit.



Prix (membres) : 44,96 €

www.elektor.fr/rfid-starter-kit-for-arduino-



Le matériel et les spécifications de Red Pitaya m'impressionnent. Ce que je voudrais encore, c'est du logiciel pour mettre davantage en valeur ce potentiel et ces qualités. Le logiciel fourni est très bien mais il me manque encore des réglages pour effectuer certaines mesures. Je trouve qu'Elektor devrait proposer des applications et même un cours sur l'utilisation et la programmation de Red Pitaya. Ce serait tout à fait dans la ligne éditoriale du magazine ! J'échangerais bien certains articles plus généraux sur les µC contre une série consacrée rien qu'au RedPit. Du concret, svp !

Luc Henderieckx



Bonjour Luc,

ton idée tombe à pic ! Nous aussi on est impressionnés par le RedPitaya, ses caractéristiques et son succès. Il ne reste plus qu'à trouver un ou plusieurs auteur(s) pour lancer une série d'articles. S'il y a parmi vous des auteurs en puissance, faites-nous signe.

Thijs Beckers | Elektor.LABS

www.elektor.fr/redpitaya

... vous aussi envoyez-nous votre avis sur votre produit favori d'Elektor ; si nous le retenons pour une publication, vous recevrez en échange un bon d'achat de 100 € à valoir dans l'e-choppe d'Elektor...

J'essaye : www.elektor.fr/rotm



DE NOUVEAU DISPONIBLE ! AVEC, EN CADEAU, UN BON D'ACHAT DE 10 € (OFFRE TEMPORAIRE)

**OSCILLOSCOPE USB
OPEN SOURCE
100 MÉCH/S
POUR SMARTPHONE,
TABLETTE ET PC**

229 € MEMBRES 206,10 €

JE SAISIS L'OCCASION

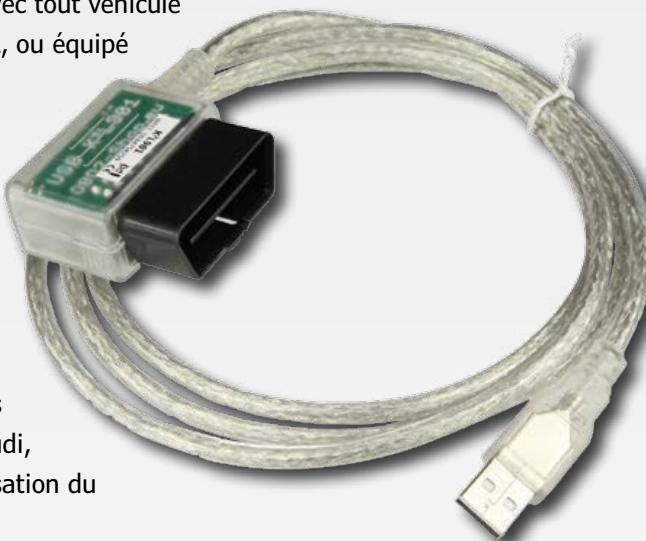
bon d'achat **VALEUR** **10 €**

Ce bon n'est pas utilisable en magasin.
Il est valable uniquement sur un achat dans l'e-choppe www.elektor.fr avant le 31-12-2015.
Seul le bonneur peut être déduit du montant d'une carte de membre ou d'adhérent.
Un bon ne peut être utilisé qu'une seule fois ; si la valeur de l'achat est inférieure à 100 €, la différence est perdue.



Le meilleur diagnostiqueur OBD conçu pour les véhicules du groupe VAG tels que VW, Audi, Seat, Skoda.

L'interface OBD 2 VAG-COM vous permet de connecter votre PC à l'ordinateur de bord de votre voiture et de lire les codes de défauts qui apparaissent lorsque les voyants d'alerte de votre véhicule s'allument. Vous pourrez également faire une remise à zéro, modifier des paramètres, etc... Il est compatible avec tout véhicule utilisant la simple *K line* ou la *K Plus L*, ou équipé d'un connecteur de données OBD 2 à 16 broches, ou d'un connecteur de données 2x2 à 4 fils des marques Volkswagen, Audi, Seat et Skoda, produit entre 1996 et aujourd'hui. Il dispose d'une base de données de plus de 7500 codes d'erreur et peut utiliser le nouveau code à 7 chiffres PIN/SKC pour se synchroniser avec les nouvelles clefs des modèles de VW, Audi, Seat et Skoda avec fonction immobilisation du véhicule.



Prix (membres) : 16,16 €

www.elektor.fr/OBD-USB-interface

Compilation Arduino

Imagine ce que tu désires, souhaite ce que tu imagines, tu finiras par créer ce que tu veux.

Avec Arduino. Avec Elektor.

interface de diagnostic OBD USB KKL

interface de diagnostic Red Pitaya

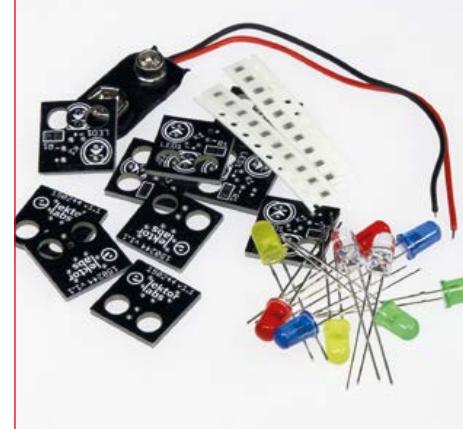
Les microcontrôleurs PIC pour les débutants

offre mystère

Kit LEGOLED avec circuit imprimé spécial LEGO



oserez-vous ?



Rémy Mallard initie les débutants à la programmation des PIC au moyen d'exemples pratiques. Il commence par les principes essentiels de programmation, puis regroupe par chapitre les informations nécessaires à la réalisation de chaque exemple. Il fait la part belle aux « petits » contrôleur à 8 bits, qui disposent entre autres de comparateurs, d'un oscillateur interne, de convertisseurs A/N, de communication à deux fils ou série, et bien davantage.



Prix (membres) : 34,16 €

www.elektor.fr/debutpic

Ce minuscule circuit imprimé compatible avec les briques LEGO permet d'incorporer des LED aux constructions en Lego. Le seul composant requis (outre la LED) est une résistance talon pour limiter le courant. La configuration du circuit permet de l'adapter à toutes les situations. Le kit comporte 10 circuit imprimés (noir), 10 LED de différentes couleurs (rouge, jaune, vert, bleu et blanc), 10 R de 390 Ω (5 V), 10 R de 820 Ω (9 V), un connecteur pour pile ou batterie de 9 V



Prix (membres) : 8,96 €

www.elektor.fr/legoled

www.elektor.fr/mystery

p.32 - carte d'E/S pour Android

- circuit imprimé avec CMS 150075-91
- circuit imprimé sans CMS 150075-1
- PIC16F1938-I/SO programmé 150057-41
- description des fonctions
(téléchargement gratuit) 150057-W

p.40 - USB déguisé en pile

- circuit imprimé 150245-1

p.42 - wattmètre CA/CC

- shield d'expérimentation 140009-91
- e-BoB ADS1115 140169-91
- circuit imprimé du wattmètre 140409-1
- circuit imprimé du module des filtres 140169-2

p.50 - e-BoB BL600 - 6e partie

- e-BoB FT232 assemblé (MOD1) 110553-91
- e-BoB BL600 assemblé (MOD2) 140270-91
- circuit imprimé 140270-1

p.56 - shield à VFD

- kit complet avec 4 tubes VFD 150064-71

Maîtrisez les microcontrôleurs à l'aide d'Arduino



Des montages Arduino inédits, étudiés spécialement pour se débarrasser définitivement de ses amis et de sa famille, et pour se retrouver enfin seul et libre de passer tout son temps à apprendre la programmation des microcontrôleurs ! Le seul livre sérieux de micro-électronique et de micro-informatique dans lequel il soit question d'Arduino mais aussi de Blanche-Neige, de la Cucaracha (en stéréo), de Saint Augustin, de Scarlatti et de Pindare (pas Pandore).



Prix (membres) : 36,85 €

www.elektor.fr/arduino

Elektor post : votre lettre d'information électronique !

The newsletter features several articles:

- Trop chaud pour s'arrêter ou quand les puits de chaleur ne suffisent plus**
- Garder la fraîcheur**
- Nouvelles** (Compilation Arduino : l'intégrale des articles d'Elektor, PIKasa, un écran pour votre précieux Raspberry Pi, Hector le phasma, premier robot conscient de lui-même, L'aluminium plus fort que le lithium-ion ?)
- Elektor.LABS**

Chaque semaine, plus de 100.000 lecteurs trouvent **elektor.post** dans leur boîte de réception le vendredi matin, avec des informations passionnantes sur le monde de l'électronique. Vous ne recevez pas **elektor.post** ? Savez-vous ce que vous manquez ?

- toutes les deux semaines un projet électronique Elektor inédit et gratuit sous forme de PDF (d'une valeur de 2,50 €) est joint à cette lettre
- des offres de réduction spéciales dans l'e-choppe d'Elektor
- téléchargement gratuit de 3 numéros parus (valeur d'au moins 3 x 7,80 €)

Inscrivez-vous, c'est gratuit !
www.elektor.fr/elektorpost

Carte Atmel SAM D20 Xplained pro



La carte Atmel SAM D20 Xplained Pro est conçue pour le prototypage avec les microcontrôleurs Cortex-SAM D20-M0+. Elle bénéficie d'un riche environnement d'outils pratiques et gratuits. Elle est utilisée notamment dans notre nouveau cours d'initiation aux µC à 32 bits dont le premier épisode a été publié dans le n° de janvier-février 2015, p. 112, et se poursuivra dans chaque numéro tout au long de l'année. Ce module est livré monté et testé. Lancez-vous !



Prix (membres) : 29,96 €

www.elektor.fr/samd20-board

Testeur de transistors avec Platino

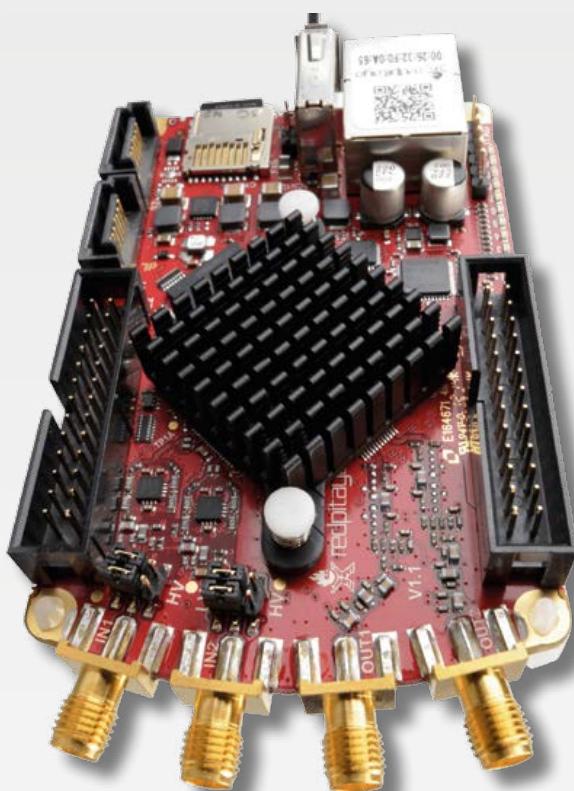


Ce testeur permet de trier comme bons ou mauvais les transistors sans référence ou douteux, de déterminer leur type (bipolaires PNP et NPN, MOSFET à canal N ou P) et de mesurer leur gain. L'instrument de mesure repose sur la carte Platino d'Elektor. Sont fournis : circuit imprimé, microcontrôleur programmé, circuit imprimé polyvalent pour AVR-Platino, afficheur et boîtier.



Prix (membres) : 66,95 €

www.elektor.fr/platino-transistor-tester



Kit Red Pitaya complet à petit prix

Red Pitaya est un outil de mesure et de commande à source ouverte, de la taille d'une carte crédit. À lui seul, il remplace avantageusement de nombreux instruments de laboratoire coûteux, tout en stimulant la créativité de ses utilisateurs. C'est un instrument de mesure de choix aussi bien pour les professionnels et les chercheurs que pour les enseignants et les étudiants.

Le kit de diagnostic Red Pitaya comprend un instrument de mesure à code source ouvert Red Pitaya V1.1, deux sondes d'oscilloscope, deux adaptateurs SMA(M) à BNC(F), un adaptateur d'alimentation micro USB et une carte Secure Digital Kingston MicroSHDC de 4 Go.



Prix (membres) : 283,50 € = Elektor casse la baraque !

www.elektor.fr/red-pitaya-kit

Compilation Arduino

Imagine ce que tu désires, souhaite ce que tu imagines, tu finiras par créer ce que tu veux.

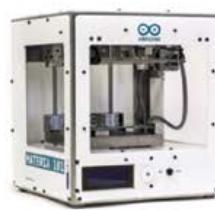
Avec Arduino. Avec Elektor.

interface de diagnostic OBD USB KKL

kit de diagnostic Red Pitaya

DANS LA FAMILLE
ARDUINO JE DEMANDE...

E-BOOK GRATUIT
À CHAQUE
COMMANDE ARDUINO



LE MOINS CHER @ ELEKTOR

hexadoku casse-tête pour elektorniciens

Si les vôtres sont déjà passées pour cette année, que diriez-vous de prolonger le plaisir de vos vacances par une séance de gymnastique cérébrale ! Ça détend, ça dérouille, ça ne peut faire que du bien. Prenez votre temps, peu importe que vous résolviez cette grille d'hexadoku en un tournemain ou que ça vous prenne trois jours, le bilan sera forcément positif.

Et vous en redemanderez, car toute grille (bien) remplie à un goût de revenez-y.

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 tirerons au sort **cinq bonnes réponses** internationales reçues dans les délais ; leurs auteurs recevront chacun un chèque-cadeau d'une valeur de **50 €** à valoir dans l'e-choppe d'Elektor.

À vos crayons !

Envoyez votre réponse (les chiffres sur fond grisé) avant le **1^{er} novembre 2015** avec vos coordonnées par courrier électronique exclusivement à **hexadoku@elektor.fr**

Les gagnants

La solution de la grille du numéro de juin est **458EF**

Les cinq bons Elektor d'une valeur de 50 € vont à : Gerard Lodder (Pays-Bas), Karl-Josef Wernet (Allemagne), Olivier Chaufouraux (Belgique), Pascal Jordil (Suisse) et Dirk Petig (Allemagne). Bravo à tous les participants et félicitations aux gagnants !

1	3	5						0	4	6					
9			F	0				8	D						5
7		E		C	9	F		0		D					A
C	F		4	8			3	7		2					E
9	1	B	D	E			7	4	5	6					0
0	4		C		6	A		8		F	5				
		3	2					B	4						
	A		4	5	C	2					3				
	D		9	0	8	3				C					
	2	A						E	5						
7	5	D		4	B			9		6	8				
A	C	7	8	2			5	F	4	1	9				
A	8		1	B			9	3		0	6				
0		9	5	3	4		D		8		2				
4		9	F				E	A							C
	F	E	7						3	5	D				

F	5	7	3	6	A	9	C	2	8	B	D	4	1	0	E
B	0	1	9	E	F	D	2	3	7	4	A	8	5	6	C
2	4	8	E	0	1	3	7	9	5	6	C	B	D	A	F
6	A	C	D	B	4	5	8	E	F	0	1	2	7	9	3
0	F	4	7	C	2	1	6	A	D	8	9	E	3	5	B
5	E	D	1	9	7	8	0	6	B	F	3	A	4	C	2
3	2	9	8	4	B	A	5	C	E	1	7	F	6	D	0
A	6	B	C	D	3	E	F	5	0	2	4	7	8	1	9
C	7	E	4	8	D	F	1	B	9	3	5	0	A	2	6
1	D	F	5	2	9	0	4	7	A	C	6	3	B	E	8
8	B	A	0	3	6	7	E	F	1	D	2	9	C	4	5
9	3	2	6	A	5	C	B	8	4	E	0	D	F	7	1
D	8	5	2	F	E	6	A	0	C	7	B	1	9	3	4
4	C	6	A	1	8	B	3	D	2	9	E	5	0	F	7
E	9	3	B	7	0	4	D	1	6	5	F	C	2	8	A
7	1	0	F	5	C	2	9	4	3	A	8	6	E	B	D

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.

KCS TraceME

2G 3G 4G LBS

LoRa™ BLE M2M

Iridium Sensor



iBeacon™

Glonass GPRS

RF GPS

Bluetooth®

SMS

Internet of Things



L'Internet des Objets avec LoRa™

KCS enrichit sa gamme de produits TraceME avec un puissant module conçu pour la mobilité planétaire et l'Internet des Objets. Dans sa dernière version, ce module de suivi et de traçage par GPS/GPRS allie géolocalisation RF et technologie LoRa™. Cette combinaison ajoute de l'intelligence aux objets communicants grâce aux longues portées et faibles consommations que permet la technologie LoRa™ dans les applications de communication entre machines. La prise en charge des protocoles GPRS/SMS et celle, optionnelle, des standards 3G, Wi-Fi, Bluetooth LE, ANT/ANT+ et iBeacon™ facilite l'intégration du module dans les réseaux sans fil et les applications mobiles. Nous le déclinerons bientôt en versions haut de gamme et gamme intermédiaire.

Module ANTIVOL à RF

Avec sa gamme de produits TraceME, KCS offre une solution de géolocalisation intelligente pour les applications antivol et à usage en intérieur. Elle utilise la RF et un algorithme intelligent pour mesurer le temps de propagation des signaux transmis (selon un protocole propriétaire). Notre module présente des caractéristiques uniques: petite taille (46x21x6,5 mm), légèreté (7 g pour une carte équipée) et longévité (durée de vie supérieure à 10 ans en mode veille). Notre algorithme « Écouter avant de communiquer » rend pratiquement impossible la localisation du module et sécurise ainsi le véhicule ou le bien à protéger. L'intégration dans les réseaux sans fil et les applications mobiles est facilitée par la prise en charge des protocoles GPRS/SMS et celle, optionnelle, des standards 3G, Wi-Fi, Bluetooth LE, ANT/ANT+ et iBeacon™.

www.Trace.ME

Toutes les marques mentionnées ici sont la propriété de leurs propriétaires respectifs.

Une qualité professionnelle à prix discount!



S'ABONNER MAINTENANT !

Newsletter

Recevez chaque semaine les toutes dernières informations

- Nouveautés
- Les meilleures offres
- Réductions des prix



TEMPÉRATURE
DÉTECTEUR DE PROXIMITÉ
ENREGISTREUR DE TRAJECTOIRE
DÉBIT
ULTRASONS
PRESSION
HUMIDITÉ
EFFET HALL
FORCE

- Plus de 45 ans d'expérience
- Envoi en 24 heures
- Plus de 50 000 produits

Les langues de notre boutique:

Capteurs effet Hall

Capteur radiométrique, linéaire

Honeywell

- Tension de service : 4,5 ... 10,5 VDC
- Consommation de courant : 8,7 mA (@ 5 VDC)
- Température de service : -40 ... +150 °C
- Linéarité : 1% typ.
- Temps de réponse : 3 µs



Boîtier : TO-92,
Sortie : 0,2 V / 1,5 mA

Numéro de commande	Magnetic-Range	Sensibilité
SS 496 A1	2,75	±84 mT
SS 495 A	1,40	±67 mT
SS 495 A1	2,65	±84 mT

Capteur radiométrique, linéaire, SMD

Honeywell

- Tension de service : 2,7 ... 6,5 VDC
- Consommation de courant : 10 mA
- Température de service : -40 ... +100 °C
- Linéarité : 1% typ.
- Temps de réponse : 3 µs



Boîtier : SOT-89
Sortie : 1,0 V / 1,5 mA

Numéro de commande	Magnetic-Range	Sensibilité
SS 59 ET	1,40	±65 mT

Capteur effet Hall, numérique, SMD

Honeywell

Capteurs effet Hall numériques compensés en température



Unipolaire, boîtier : SOT-89
Sortie : 0,4 V / 20 mA

Numéro de commande	Magnetic-Range
SS 543 AT	1,40
SS 549 AT	2,65

Capteurs Hall

SIEMENS



Boîtier : P-SSO-3-2

Numéro de commande	Magnetic-Range
TLE 4905L	0,60
TLE 4935L	0,63

Prix du jour ! Prix à la date du : 27.07.2015

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

Modes de paiement internationaux:



Commander maintenant! www.reichelt.fr

Assistance téléphonique en anglais : +49 (0)4422 955-360



Détecteur de proximité inductif

CONTRINEX



<http://rch.it/kj>



Excellent capteurs inductifs pour détecter sans contact les pièces métalliques dans le quotidien de l'automatisation. Idéal pour une utilisation dans des conditions environnementales difficiles.

Extrait de notre gamme.
Vous trouverez la gamme complète à l'adresse suivante :

Fermature PNP

Numéro de commande	Ø x longueur	Distance de commutation	Pose
DWAD 623 03	66,80	Ø 3,0 x 22 mm	1,0 mm
DWAD 623 04	51,40	Ø 4,0 x 25 mm	1,5 mm
DWAD 623 M5	51,40	M5 x 25 mm	1,5 mm
DWAD 503 M12	47,90	M12 x 50 mm	6,0 mm
DWAS 623 M 001	32,90	M8 x 45 mm	2,0 mm
DWAS 613 M 001	30,80	M8 x 45 mm	2,5 mm
DWAS 513 M12	47,80	M12 x 60 mm	10,0 mm

Barrière photo-électrique Ø x long. 18 x 50 x 50 mm

CONTRINEX



Barrière photo-électrique à une voie, câble de 2 m

Numéro de commande	Distance de commutation	Type de commutation
LLK 5050 000	23,30	15000 mm Emetteur
LLK 5050 003	35,99	15000 mm antivalent, PNP

Barrière photo-électrique à réflexion, câble de 2 m

LLK 5050 103	42,40	800 mm antivalent, PNP
--------------	-------	------------------------

Capteurs de distance

SHARP

Très haute fiabilité ainsi que précision plus élevée par rapport aux capteurs classiques.



Types : GP2Y0A

Numéro de commande	Portée (cm)	Long.xLarg.xProf. (mm)
GP2-0215	6,70	29,5 x 13,0 x 21,6
GP2-0430	5,10	4 ... 30 37,0 x 18,9 x 13,5
GP2-1080	5,65	10 ... 80 29,5 x 13,0 x 21,6

Numéro de commande

DMC01-SC150	2,15	3-pôles	Câble
-------------	------	---------	-------

Capteur d'humidité 0 ... 100 % rF, TO 39



Capteur d'humidité numérique avec interface IC dans le boîtier TO39 résistant à la pression (jusqu'à 16 bars), convient pour les mesures du point de rosée.



Numéro de commande

HYT 939 29,90 TO 39

Capteurs de pression, UsV 4,75-5,25

freescale semiconductor



Numéro de commande	PR kPa	Sens mV/kPa	Lin %
MPX 5010DP	11,60	0 - 10	450 ±5
MPX 5050DP	14,90	0 - 50	90 ±2,5
MPX 5100DP	14,90	0 - 100	45 ±2,5
MPX 5500DP	9,35	0 - 500	9 ±2,5

Capteurs à ultrasons

Récepteur et émetteur céramique à ultrasons de 40 kHz



Ø 9,9 mm, H 7,1 mm

Numéro de commande	Récepteur
MUS-40E	3,05

Numéro de commande	Emetteur
MUS-40S	3,05