

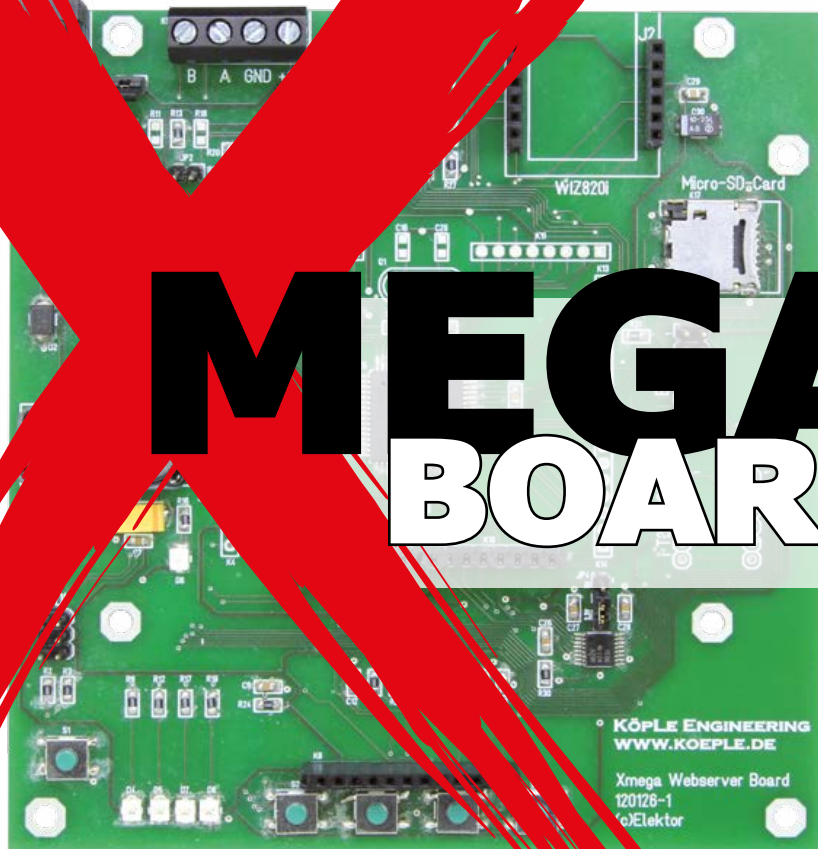
# elektor

mesure  
pilotage  
régulation

domo-  
tique

éclairage  
de scène

serveur  
web



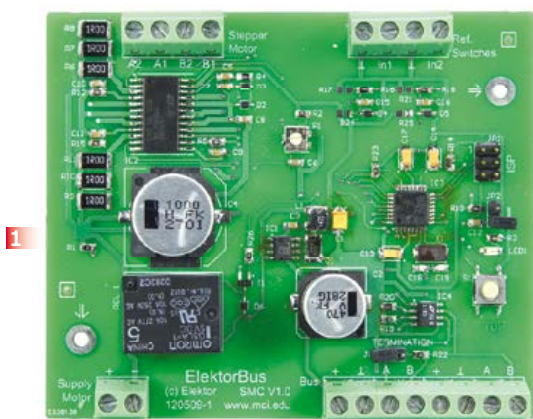
# MEGA BOARD

µC **Xmega** sur carte polyvalente  
avec afficheur, mémoire SD, réseau, RS485, boutons et LED

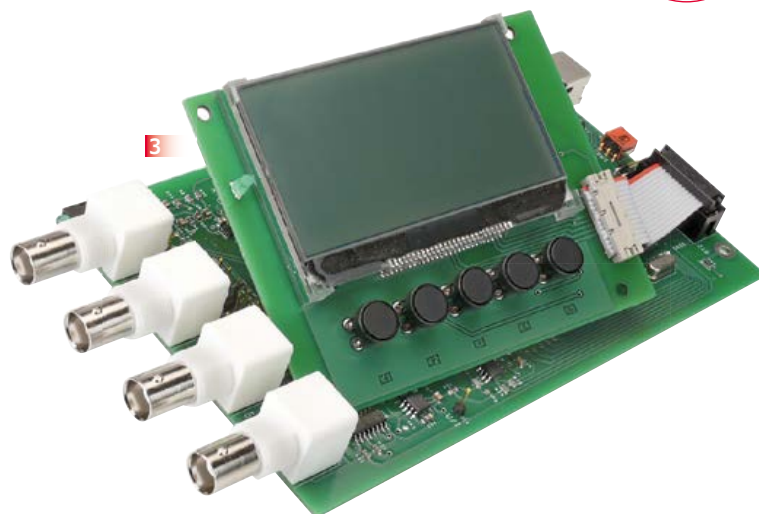
- **matrice de 8x8 LED** bicolores avec ATmega328P | horloge thermomètre à Numitron
- elektorcardi♥scope Android 3<sup>e</sup> partie | **carte Linux Elektor v.2**
- liaison radio modulaire à code Manchester 2<sup>e</sup> partie : le logiciel | DS : 4<sup>e</sup> jour - un projet simple
- lanceurs d'alerte, il vous faut une infrastructure !

L 19624 -424- F: 7,80 €-RD





1



3



2



4



5

## 1 Pilote de moteur pas à pas Pour ElektorBus (module)

Voyager avec l'ElektorBus ouvre de nouveaux horizons. Grâce à sa modularité logicielle autant que matérielle, ce bus accélère le développement d'applications. Exemple pratique : la mise au pas des moteurs !

Réf 120509-91

## 2 Barostick

Clés USB, clés du succès ? Elles sont partout, elles sont notre album d'images, de vidéos, de musiques préférées, d'articles, de fichiers et même de températures. Et la pression atmosphérique, y aviez-vous pensé ? C'est fait : sur un baromètre sans mercure, avec un capteur Bosch, hectopascals et degrés Celsius rejoignent Windows pour se faire tirer le portrait.

Réf 120481-91

## 3 LCR-Mètre 0,05 % Le luxe de la précision accessible à tous

La remarquable précision de cet appareil et son étonnant confort d'utilisation sont le résultat d'une étude soignée. Il marche si bien derrière sa façade dépouillée, qu'on en oublierait presque la subtilité des techniques de mesure mises en œuvre. L'occasion rêvée, pour nos lecteurs passionnés par la mesure, de se faire plaisir. Si, comme nous, les prodiges que les techniques modernes mettent à notre portée vous émerveillent, venez palper le poudrier de volt.

Module carte principale réf 110758-91

Module de l'extension afficheur et clavier réf 110758-92

Les deux modules réf 110758-93

## 4 Module réseau universel Connectivité internet pour les nuls

Le raccordement de sa propre électronique par une

liaison internet est une fonction impressionnante. Ce module réseau universel - composé d'un circuit compact, d'une bibliothèque de logiciels libres et d'un microcontrôleur utilisable immédiatement comme serveur web - s'adresse même aux débutants. Quant aux électroniciens plus chevronnés, ils apprécieront à leur juste valeur les caractéristiques comme la communication SPI, PoE, et plus encore !

Réf 100552-91

## 5 Accéléromètre DIPLOmé

Accessoire d'adaptation pour puce d'accéléromètre (dont les surfaces de contact se trouvent sous le boîtier de type LGA) à une carte de prototypage ordinaire.

Réf 090535-91

Informations et gamme complète sur :

[www.elektorpcbservice.com](http://www.elektorpcbservice.com)



# POUR L'AVENIR, IL VAUT MIEUX VOIR GRAND.

Du mardi 7 au vendredi 10 janvier 2014  
Las Vegas, Nevada • CESweb.org • #CES2014

## ÇA TOMBE BIEN, À LAS VEGAS ON A LES IDÉES LARGES.

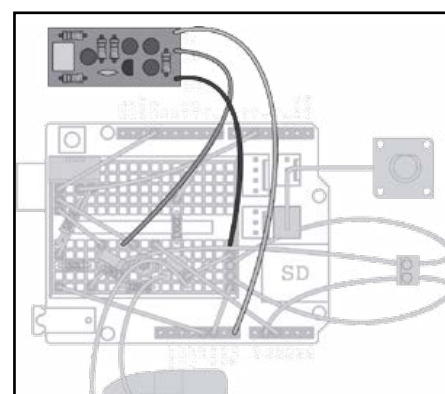
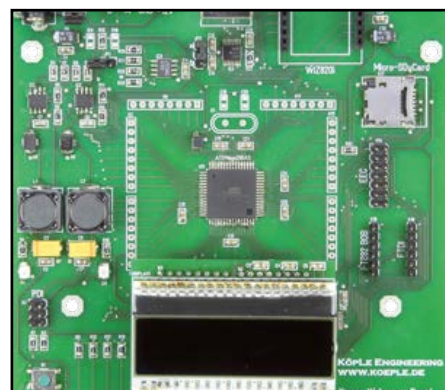
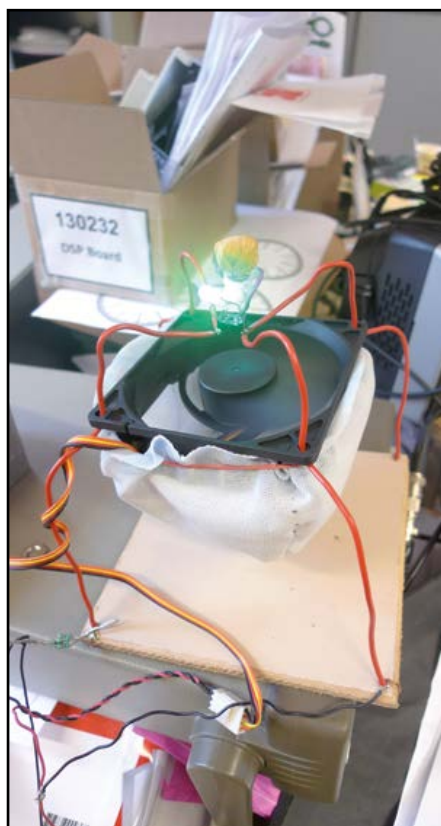
Quatre jours durant, les bâtisseurs du futur se retrouvent dans une ville elle-même symbole de réinvention. Là convergeront marques, marchés et économies, au sein de ce qui sera bien plus qu'un salon professionnel. En 2014, plus que jamais, les possibilités se multiplieront de rencontrer ceux qui font la différence. La seule question : quel sera votre rôle dans l'élaboration de votre futur ?

**Inscrivez-vous maintenant sur CESweb.org**



THE GLOBAL STAGE FOR INNOVATION

PRODUCED BY  CEA®



## ● e-communauté

### 6 de nous à vous

### 8 le monde d'elektor

Nous parlons ici des rencontres fécondes que l'on fait dans le *grand carrefour international de l'électronique* autour d'Elektor.

## ● DesignSpark

### 36 liaison radio modulaire à code Manchester

#### 2<sup>e</sup> partie : le logiciel

Après avoir présenté le mois dernier une carte habilement routée, nous abordons maintenant la question du micrologiciel.

### 54 4<sup>e</sup> jour – un projet simple

Je propose de réaliser un simple pilote de LED bicolore pour apprendre à utiliser les éditeurs de schémas et de circuits imprimés.

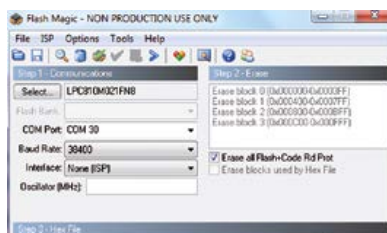
## ● e-labs

### 57 énergie négative

C'est l'histoire du mec qui arrive chez ses beaux-parents et appuie sur le bouton de la sonnette et qui... ah, je vous l'ai déjà racontée ?

### 58 premiers pas avec le Mini Kit LPC800

Que croyez-vous qu'il se soit passé avec ce Mini Kit LPC800 qui comporte un  $\mu$ C à 32 bits ARM Cortex-M0+ LPC810, proposé gratuitement par Elektor à ses lecteurs ?



## ● e-projets

### 10 matrice de 8x8 LED bicolores avec ATmega328P

Voici la matrice charliplexée, une méthode simple et astucieuse pour commander des LED, agrémentée ici d'une manette de jeu pour animer la matrice lumineuse.

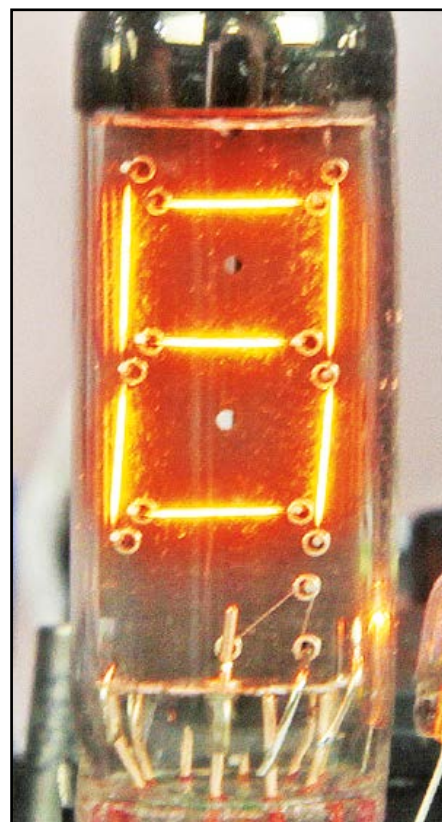
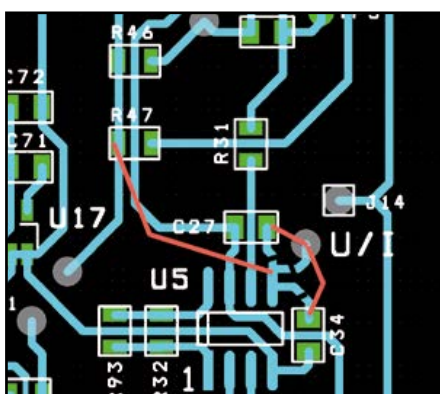
### 18 Xmega sur carte polyvalente avec afficheur, mémoire SD, réseau, RS485, boutons et LED

Pour tirer le meilleur du  $\mu$ C Xmega, il y a plein d'interfaces sur cette carte qui peut devenir serveur internet ou servir pour d'autres applications en réseau.

### 30 horloge thermomètre à Numitrons développée avec Arduino

Sortons l'Arduino de son monde de *shields* et de plaques d'essais pour l'intégrer à un projet qui plaise aussi





aux fans d'AVR. Ces deux mondes sont souvent perçus comme inconciliables, tout comme les  $\mu C$  et les tubes soviétiques des années 50.

#### 42 **elektorcardi♥scope Android** 3<sup>e</sup> partie : réalisation, réglages & mode d'emploi

Voici le dernier volet de cette réalisation qui fera date. Au moment de sa publication, il n'existe dans le commerce rien de comparable, ni par les performances, ni par le prix.

**51 nouvelle version de la carte Linux Elektor avec LAN et horloge en temps réel**  
Nous avons tenu compte des souhaits des (très) nombreux utilisateurs de la première version.

**62 importante mise à jour du LCR-mètre**  
Version améliorée du logiciel et du matériel

#### 66 **mettez le cap sur Arduino (7): à l'école d'Éole**

Nous avons invité des étudiants à utiliser une station météo mobile pour collecter des données en ville, dans des parcs et le long de cours d'eau, afin de comparer des ensembles de données provenant d'environnements différents.

#### ● e-tech the future

#### 72 **lanceurs d'alerte, il vous faut une infrastructure !**

Comment fonctionne la démocratie quand les électeurs sont maintenus dans l'ignorance ? Quelle est la portée de la dénonciation de ces pratiques par de simples citoyens ? Les lanceurs d'alerte ont-ils un rôle clé à jouer dans le processus démocratique ?

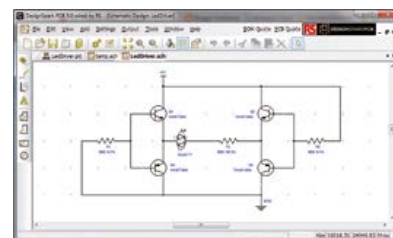
#### ● e-magazine

#### 76 **rugosimètres Philips PR9150/PR9151**

Gratter un peu la surface des métaux comme dans les années 50  
La plupart des électroniciens détestent l'usinage et le travail mécanique en général, et la plupart des accros au travail du métal ont une piètre opinion de l'électronique.

**81 hexadoku**  
casse-tête pour électroniciens

**82 bientôt dans Elektor**



**ELEKTOR / PUBLITRONIC SARL**

c/o Regus Roissy CDG

1, rue de la Haye

BP 12910

FR - 95731 Roissy CDG Cedex

Tél. : (+33) 01.49.19.26.19

lundi, mardi et jeudi de 8h30 à 12h30

Fax : (+33) 01.49.19.22.37

www.elektor.fr

Banque ABN AMRO : Paris

IBAN : FR76 1873 9000 0100 2007 9702 603

BIC : ABNAFRPP

**DROITS D'AUTEUR :**

© 2013 Elektor International Media B.V.

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

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

Elektor est édité par Elektor International Media B.V.  
Siège social : Allee 1 - 6141 AV Limbricht, Pays-Bas

Imprimé aux Pays-Bas  
par Senefelder Misset - Doetinchem

Distribué en France par M.L.P.  
et en Belgique par A.M.P.

## Éditorial

On ne s'arrête que rarement au nombre ordinal sur la couverture d'un magazine. À moins que ce soit le 100<sup>e</sup>, le 300<sup>e</sup> ou tout autre chiffre rond qui frappe l'imagination et fournisse le prétexte de parler du temps qui passe, si vite, du monde qui change, si vite aussi, ou d'autres lieux communs comme les frasques d'un acteur pathétique ou... une guerre qui se prépare.

Or, ce n'est pas parce qu'il est le 424<sup>e</sup> du rang que ce numéro d'Elektor est remarquable, mais évidemment par son contenu. Ce qui est

aussi un sujet bateau pour un édito... Encore

que, vous pouvez toujours courir les kiosques de France, de Navarre et même du reste du monde pour y dénicher un magazine qui vous serve, sur un plateau, une carte avec le **µC ATXmega256A3** à la pointe de la technologie des micro-contrôleurs, une **nouvelle carte Linux pour des applications embarquées** et un **électrocardi♥scope** de qualité pour tablette ou téléphone Arduino.

Mais je m'égare.

Mes collègues néerlandais, eux, ils en sont au 600<sup>e</sup> numéro de la version hollandaise d'Elektor. Cinquante-sept ans d'activité ininterrompue et se trouver encore et toujours à la pointe de la technologie, pour la rendre accessible au plus grand nombre, ça mérite bien **un grand coup de chapeau**.

Tiens, essayez donc d'imaginer où en sera l'électronique quand l'édition française aura atteint cet âge vénérable, dans... 176 numéros !

Qu'au même moment, précisément à l'heure où sont écrites ces lignes, on annonce l'imminence d'une guerre que je répugne à dire nouvelle (car n'est-ce pas toujours la même guerre qu'ils recommencent ?), qu'au même moment, précisément à l'heure où il faut en finir avec cet éditorial, le 28 août 2013 à 16 h, on commémore le célèbre discours de Martin Luther King et son rêve de paix, cette coïncidence a de quoi en désarçonner plus d'un, même parmi les plus solides, au point qu'ils en oublient qu'il fallait boucler cet édito avant 16 h.

Voilà qui est fait. Bonne lecture.

**Denis Meyer**



## Notre équipe

Rédacteur en chef :	Denis Meyer (redaction@elektor.fr)
Directeur éditorial :	Wisse Hettinga
Rédaction internationale :	Harry Baggen, Jan Buiting, Thijs Beckers, Eduardo Corral, Jens Nickel, Clemens Valens
Laboratoire :	Thijs Beckers, Ton Giesberts, Luc Lemmens, Tim Uiterwijk, Jan Visser
Ont coopéré à ce numéro :	Jean-Paul Brodier, Robert Grignard, Hervé Moreau, Kévin Petit, NN
Service de la clientèle :	Jolanda van Kruchten
Graphistes :	Giel Dols, Jeanine Opreij, Mart Schroijsen
Elektor online :	Daniëlle Mertens & Patrick Wielders





### France

Denis Meyer  
+31 46 4389435  
d.meyer@elektor.fr



### United Kingdom

Wisse Hettinga  
+31 (0)46 4389428  
w.hettinga@elektor.com



### USA

Hugo Van haecke  
+1 860-289-0800  
h.vanhaecke@elektor.com



### Germany

Ferdinand te Walvaart  
+49 241 88 909-17  
f.tewalvaart@elektor.de



### Netherlands

Harry Baggen  
+31 46 4389429  
h.baggen@elektor.nl



### Spain

Eduardo Corral  
+34 91 101 93 95  
e.corral@elektor.es



### Italy

Maurizio del Corso  
+39 2.66504755  
m.delcorso@inware.it



### Sweden

Wisse Hettinga  
+31 46 4389428  
w.hettinga@elektor.com



### Brazil

João Martins  
+31 46 4389444  
j.martins@elektor.com



### Portugal

João Martins  
+31 46 4389444  
j.martins@elektor.com



### India

Sunil D. Malekar  
+91 9833168815  
ts@elektor.in



### Russia

Nataliya Melnikova  
+7 (965) 395 33 36  
Elektor.Russia@gmail.com



### Turkey

Zeynep Köksal  
+90 532 277 48 26  
zkoksal@beti.com.tr



### South Africa

Johan Dijk  
+31 6 1589 4245  
j.dijk@elektor.com



### China

Cees Baay  
+86 21 6445 2811  
CeesBaay@gmail.com

## Notre réseau



VOICE & COIL

CIRCUIT CELLAR



audioXpress



## vous connecte à



## Nos annonceurs



### Beta Layout

[www.pcb-pool.com](http://www.pcb-pool.com) . . . . .17



### CadSoft

[www.cadsoft.de](http://www.cadsoft.de) . . . . .29



### CES International 2014

[www.CESweb.org](http://www.CESweb.org) . . . . .3



### Eurocircuits

[www.elektorpcbservice.com](http://www.elektorpcbservice.com) . . . . .35



### Pico

[www.picotech.com/PS250](http://www.picotech.com/PS250) . . . . .83



### Reichelt

[www.reichelt.fr](http://www.reichelt.fr) . . . . .84



### Schaeffer AG

[www.schaeffer-ag.de](http://www.schaeffer-ag.de) . . . . .82

## Pour placer votre annonce dans le prochain numéro d'Elektor

veuillez contacter Mme Ilham Mohammadi par téléphone au (+31) 6.41.42.25.25  
ou par courrier électronique : i.mohammadi@elektor.fr

## Vos correspondants

Nous sommes à votre service pour toute question relative à votre commande ou votre abonnement  
par téléphone au (+33) 01.49.19.26.19 lundi, mardi et jeudi de 8h30 à 12h30  
ou par courriel : service@elektor.fr

# le monde d'Elektor

compilé par  
**Wisse Hettinga**

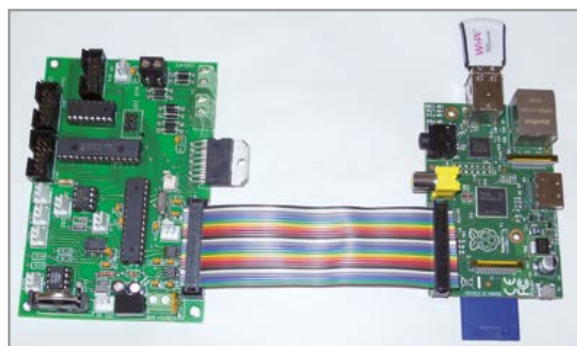
Le monde d'Elektor est un grand carrefour international où se croisent les voies les plus diverses et se créent les liens les plus improbables. Nous parlons ici de ces rencontres fécondes, de leurs fruits parfois inattendus. L'électronique en sort enrichie, plus passionnante que jamais.



## Séparation galvanique pour Iso-Pi

Quand Brian Millier, qui écrit pour notre confrère *Circuit Cellar*, a reçu fin 2012 son premier Raspberry Pi, il a aussitôt lancé un projet qui allait inspirer ses deux articles *Raspberry Pi I/O Board* (CC août et sept. 2013). Millier, ancien ingénieur d'instrumentation de l'Université Dalhousie, à Halifax, relate son propre apprentissage de Pi et propose une carte E/S polyvalente conçue par lui. « Entretemps, l'un des concepteurs de R.P. a lui aussi propose une telle carte d'E/S appelée Gertboard » constate Millier, « mais la mienne est très différente et offre quelques notables avantages » Millier cite p. ex. l'isolation galvanique totale entre les périphériques d'E/S embarqués et le Raspberry Pi lui-même. Il insiste pour que vous regardiez de près comment il configure son *Iso-Pi* dans le numéro de sept. de *Circuit Cellar*.

Un câble en ruban relie la carte Raspberry Pi (à droite) à l'Iso-Pi de Millier.



## xpressément audiophile

Rebondissement dans la longue histoire d'amour entre Elektor et l'audio : lancement de la nouvelle formule d'*audioXpress*. Ce titre est l'un des quatre dont Elektor avait fait l'acquisition en 2011, avec *Voice Coil*, *Loudspeaker Industry Sourcebook* et *World Tube Directory*, ainsi que les fonds d'ouvrages spécialisés associés. Ces titres fondés aux Etats-Unis par Edward T. Dell (1923-2013) et publiés par lui et son équipe pendant 35 ans, avaient pour mission de promouvoir l'œuvre de spécialistes non professionnels, érigée en modèle à suivre par l'industrie sur la voie de l'excellence de la conception et de l'exigence de qualité dans les réalisations. Cette passerelle jetée entre deux univers réputés inconciliables a permis aux amateurs qui construisent eux-mêmes de dialoguer avec l'industrie. Et vice versa ! Certains des ouvrages publiés par E. Dell ont d'ailleurs été publiés en français par Elektor. Une nouvelle équipe, renforcée par des auteurs du réseau d'Elektor, prépare le lancement d'un format *augmenté* d'*audioXpress* destinés aux ingénieurs et aux passionnés de l'audio. Sa présentation aura lieu à la prochaine Convention de l'AES, à New-York, du 17 au 20 octobre 2013.

[www.audioxpress.com](http://www.audioxpress.com)





## Prochaine étape : Arduino.next

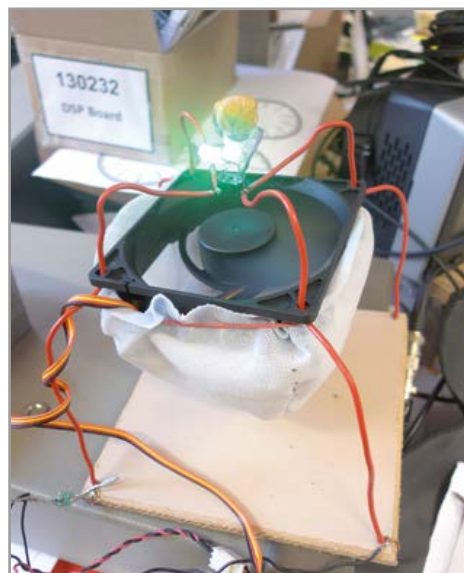
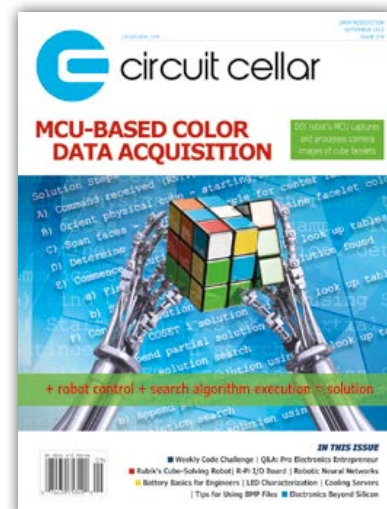
Pour de nombreux jeunes et moins jeunes, Arduino est devenu l'une des principales voies d'accès à la technique. Clé principale de ce succès planétaire, la facilité avec laquelle on crée une application avec Arduino en quelques minutes, sans connaissances approfondies de l'électronique ni de la programmation. Presque tout est sur l'internet. Il suffit de savoir copier-coller du code, reproduire un circuit simple et vous voilà parti.

Une fois arrivé, vous vous demanderez comment aller plus loin. Comment p. ex. changer le comportement de votre application ? Que faire pour obtenir un fonctionnement différent, adapté à vos besoins ? Ces questions, nous nous les sommes posées et avons cherché des réponses appropriées, simples et faciles à comprendre, pour vous aider à bien négocier la prochaine étape (et les suivantes, d'ailleurs) sur votre plate-forme embarquée préférée. Cette étape, nous l'avons baptisée *Arduino.next*, un nouveau service, bientôt proposé par Elektor. Restez à l'écoute ! Suivez-nous sur Facebook, [www.facebook.com/arduino.next](http://www.facebook.com/arduino.next), et sur Twitter, @arduino.next. En attendant, certains produits Arduino sont déjà en vente dans l'e-shoppe sur [www.elektor.fr/arduinoshop](http://www.elektor.fr/arduinoshop)

## Circuit Cellar : nouveau tonus

Dans son numéro de sept. 2013, le magazine *Circuit Cellar* apparaît ressourcé, dans la forme et le fond. Son ambition est toujours de s'adresser aux ingénieurs, aux universitaires et aux concepteurs ambitieux de systèmes embarqués, mais désormais partout dans le monde. En coopération étroite avec *Elektor International Media*, l'équipe de CC a imaginé une mise en page rafraîchie et modernisée, destinée à mieux mettre en valeur notamment les illustrations. Des liens explicites vers les sources et les références essentielles sont incorporés aux articles (codes QR inclus) pour faciliter l'accès au code source, aux vidéos et aux listes de composants.

Deux nouvelles rubriques d'information enrichissent le contenu de CC : *Green Computing* (ou l'informatique verte) par Ayse Coskun (n° 278) et la pratique de la logique programmable par Colin O'Flynn (n° 279). Vous remarquerez aussi *CC World* (= le monde de CC), une rubrique qui ressemble à celle que vous êtes en train de lire. Il y est question d'un captivant *Défi hebdomadaire de programmation* (<http://bit.ly/1brGEIU>).

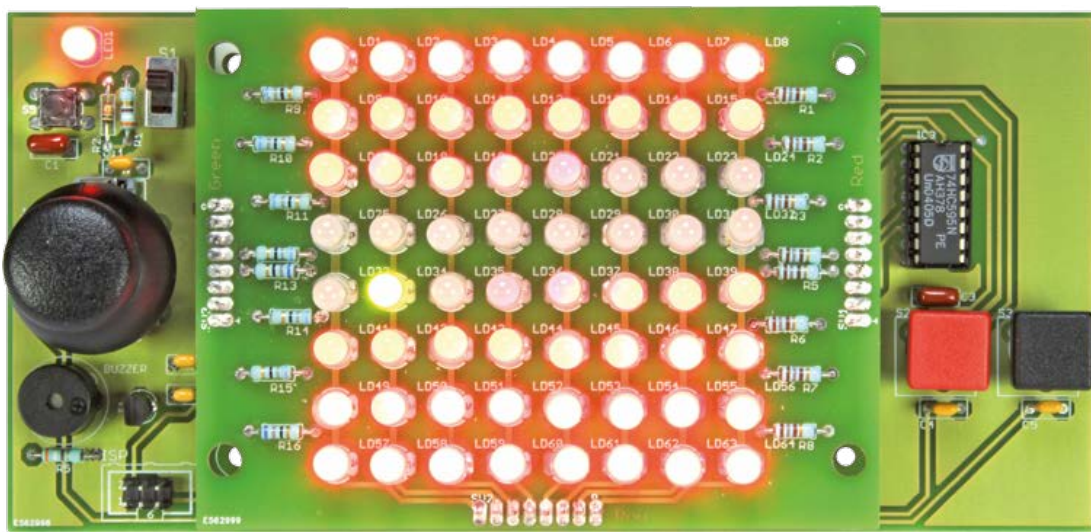


## Vortex anti-moustiques

Quand vous entrez dans le coin du labo occupé par Aurélien Moulin, stagiaire cet été chez Elektor, vous ne savez pas sur lequel de ses montages en cours fixer votre attention ; ils sont nombreux, variés, mais tous également intéressants. Il y en a un qui pourtant captive aussitôt tous les visiteurs de passage, y compris les moustiques, nombreux au-dessus des douves du château. Et c'est précisément pour cela qu'il est là, *l'aspirateur à moustiques* d'Aurélien : une ou deux LED surmontent un ventilateur de récup, lui-même posé horizontalement sur une espèce de poche en tissu. Son principe est simple, mais d'une efficacité redoutable pour les insectes : les LED attirent les moustiques, le ventilateur les aspire vers le bas et les souffle dans le sac en tissu où ils passent ensuite le reste de leur vie à jouer à la belote. Oui, le comble, c'est qu'ils sont encore vivants ! Il paraît que le premier jour, l'engin a attiré dans son vortex pas moins de 120 moustiques dûment recensés par Aurélien (qui n'aime pas les moustiques, mais les comptes précis).

(130187)

# matrice de 8x8 LED bicolores avec ATmega328P



Ruben van Leeuwen  
& Cederique Prevoo  
(Pays-Bas)

Vous avez plus de LED à allumer que de pattes au microcontrôleur ? Vous voulez faire simple et astucieux ? Voici *la matrice charliplexée*, une méthode particulièrement rentable. Et agrémentée ici d'un petit divertissement : une manette de jeu pour animer la matrice lumineuse.

Pour expérimenter sur la commande de LED, rien de tel que de se frotter à la construction d'une matrice de LED. Avec pareil réseau, on est vite confronté à la complexité de l'écheveau de fils d'alimentation et au manque de lignes de port d'un honnête microcontrôleur pour avoir accès à chacune des LED. La technique du multiplex

réduit les exigences dans les cas simples, mais elle a ses limites. Il nous fallait autre chose et avons choisi une variante méconnue du multiplex, à savoir le *charlieplexing*, conçu en 1995 par Charlie Allen, de chez Maxim. En multiplexage normal : les rangs de la matrice sont attaqués par les signaux de commande, bas ou hauts, tandis que les colonnes sont activées à tour de rôle au rythme de la fréquence de rafraîchissement. Le *charlieplexing* fait alterner les fonctions des signaux de rangs et de colonnes, ce qui permet de commander plus de LED avec le même nombre de lignes d'E/S. La **figure 1** donne un aperçu du procédé. Au lieu d'une LED ; il y en a deux, montées tête-bêche entre chaque nœud de la matrice. Il suffit alors de changer le niveau du signal de rang ou de colonne pour allumer les LED, ou bien les éteindre en mettant les deux signaux simultanément haut ou bas.

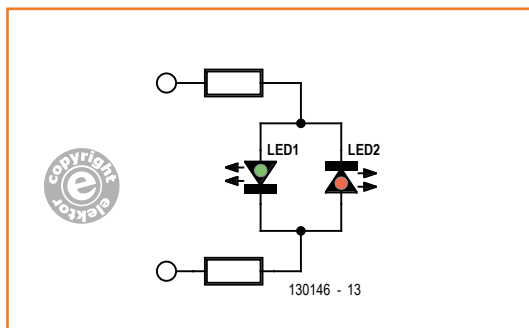


Figure 1.  
Le principe de la méthode  
du *charlieplexing*.



## La matrice

Les LED utilisées sont bicolores, en rouge et vert, leur boîtier contient en fait deux puces raccordées par trois fils, deux d'anode et celui des cathodes communes, comme le montre la **figure 2**. L'anode de la puce rouge est repérée par le méplat du boîtier. La matrice se compose de 8 x 8 de ces LED bicolores, elle fait l'objet d'un circuit imprimé à part dont le schéma est représenté à la **figure 3**.

## L'électronique de commande

C'est un ATmega328P qui est aux commandes, un microcontrôleur AVR à 8 bits avec 32 Ko de mémoire flash pour le programme, 1 024 octets d'EEPROM et 2 Ko de SRAM. Ce projet utilise l'oscillateur interne de l'ATmega, on économise de la sorte un quartz externe. Le contrôleur

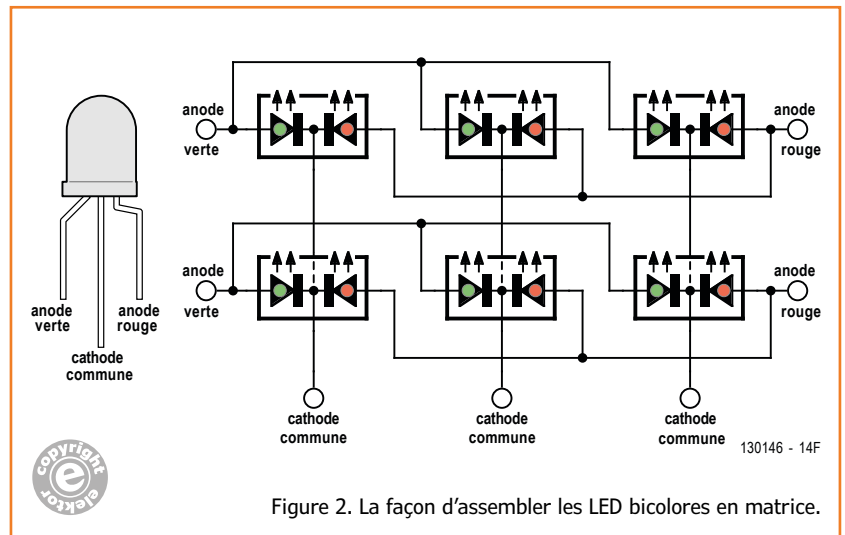


Figure 2. La façon d'assembler les LED bicolores en matrice.

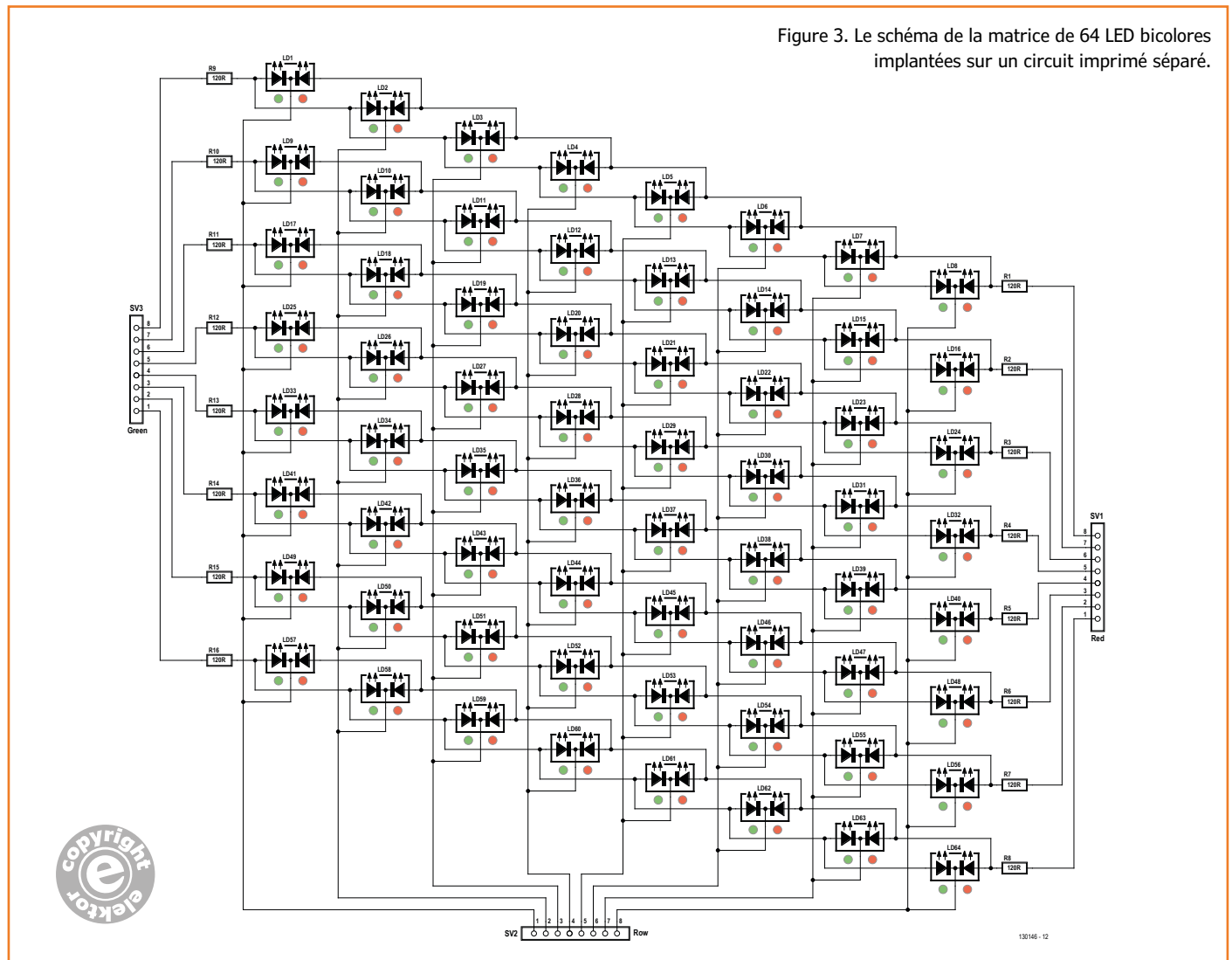


Figure 3. Le schéma de la matrice de 64 LED bicolores implantées sur un circuit imprimé séparé.

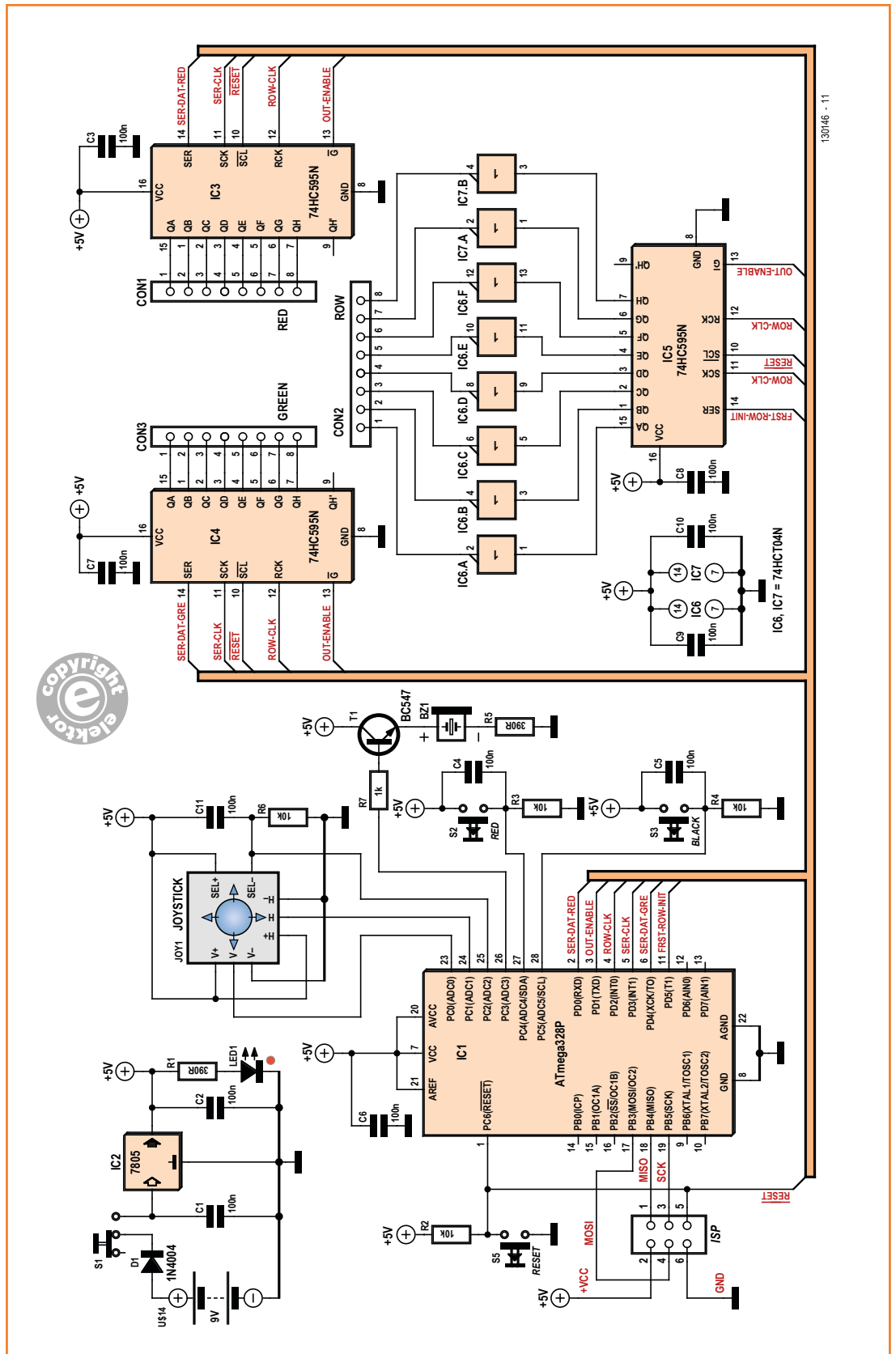


Figure 4.  
L'électronique de commande  
se fonde sur un  $\mu\text{C}$  ATmega  
et 3 registres à décalage.



commande trois registres à décalage à 8 bits du type 74HC595, un pour les LED vertes, un pour les rouges et un pour les cathodes communes, il s'agit d'IC3, IC4 et IC5 dans le schéma (**fig. 4**). Une combinaison qui épargne une quantité de lignes de port du  $\mu\text{C}$  ainsi disponibles pour d'autres tâches.

Le  $\mu\text{C}$  envoie toujours 8 bits en série aux registres des deux couleurs pour y stocker les informations. Les 8 bits une fois transmis, il donne une impulsion aux entrées de validation vers les sorties des deux registres de couleur et vers celui des cathodes communes pour allumer un rang de LED complet. Quand il active le registre des cathodes pour allumer un rang, cela implique que le rang acquiert un niveau bas ; en revanche, un niveau haut rend le rang inactif. Il s'ensuit que la différence de potentiel aux bornes des LED est nulle sur un rang inactif et ces LED restent éteintes. Sur les rangs actifs, la différence de potentiel sur les LED est de 5 V, de quoi les allumer. Tout ce processus est renouvelé huit fois par trame pour couvrir tous les rangs. Pareille trame représente le code programmé par l'utilisateur pour une image complète exposée sur 8 x 8 LED rouges et vertes.

L'allumage des LED se produit à une fréquence supérieure à 60 Hz, si bien que l'œil humain ne perçoit pas les extinctions successives des rangs. En reproduisant 60 images à la seconde, on obtient une image complète

en 0,017 s.

Le

**Tableau 1. L'attribution des ports**

entrées :	ports $\mu\text{C}$
bouton noir	PC5
bouton rouge	PC4
axe horizontal joystick	PC0
axe vertical joystick	PC1
sorties :	ports $\mu\text{C}$
ronfleur	PC3
impulsion 1er rang	PD5
données sérielles rouges	PD0
données sérielles vertes	PD4
horloge sérielle	PD3
horloge de rang	PD2

**tableau 1** indique les attributions des lignes de port. On y voit qu'une manette de jeu est raccordée à PC0 et PC1. Un convertisseur A/N servira à lire en analogique l'état du manche de commande. Pour produire les bips, le circuit est équipé d'un ronfleur à courant continu, donc avec oscillateur incorporé, dont l'inconvénient est la fréquence fixe du signal sonore. On remarque aussi dans le schéma la présence d'un connecteur ISP pour programmer l'ATmega en circuit. Il est ainsi possible d'apporter des modifi-

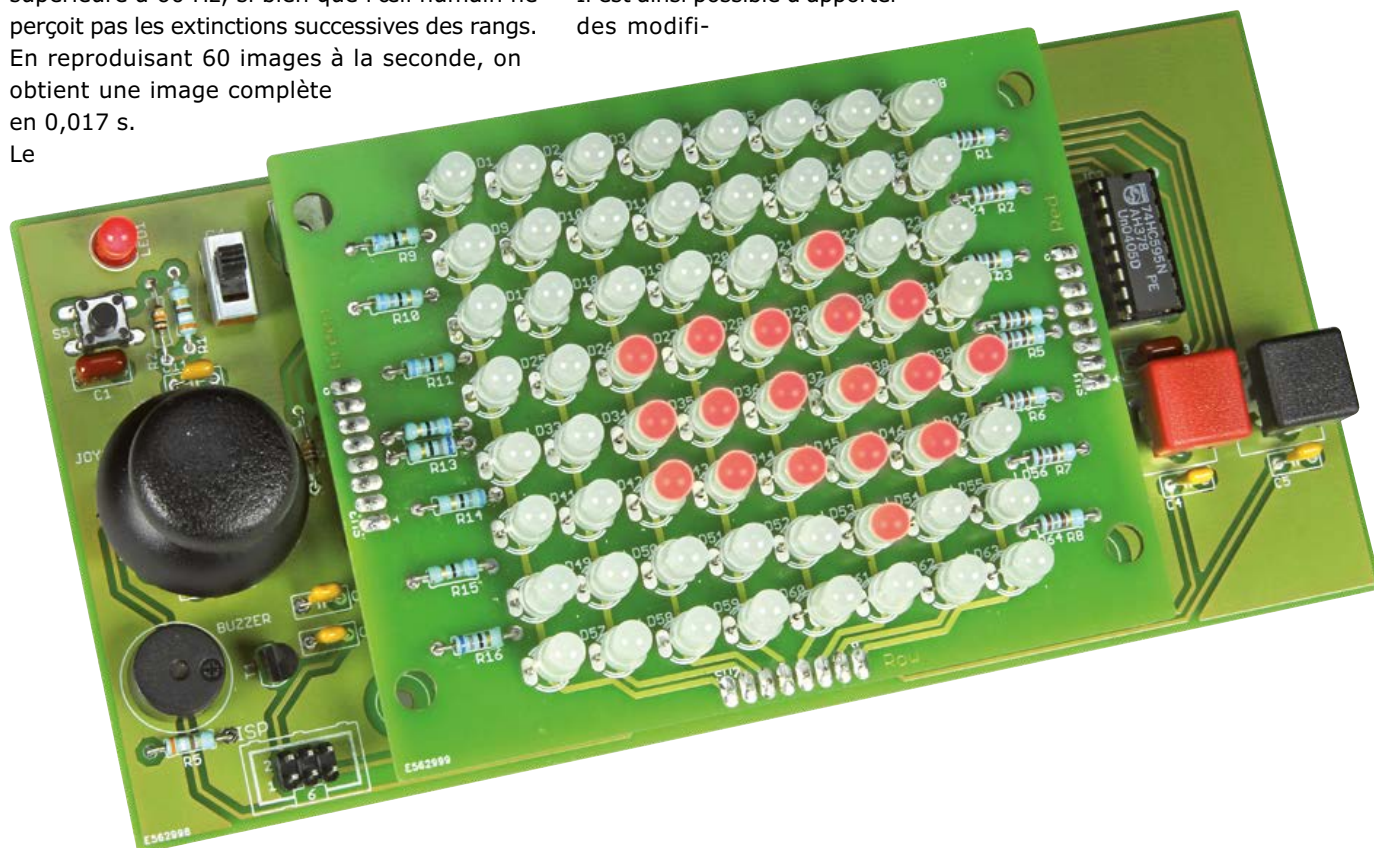
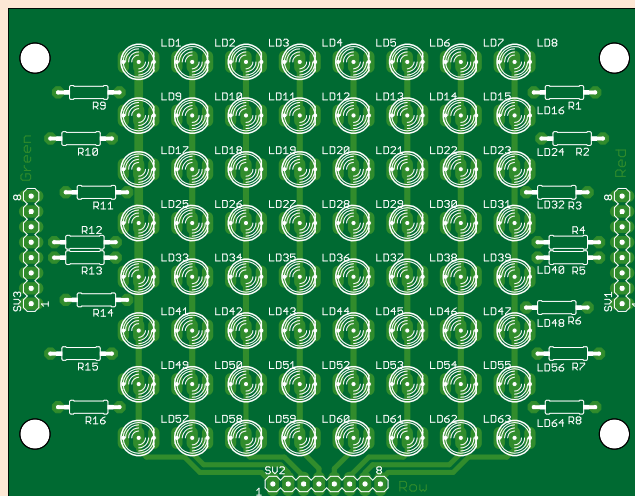


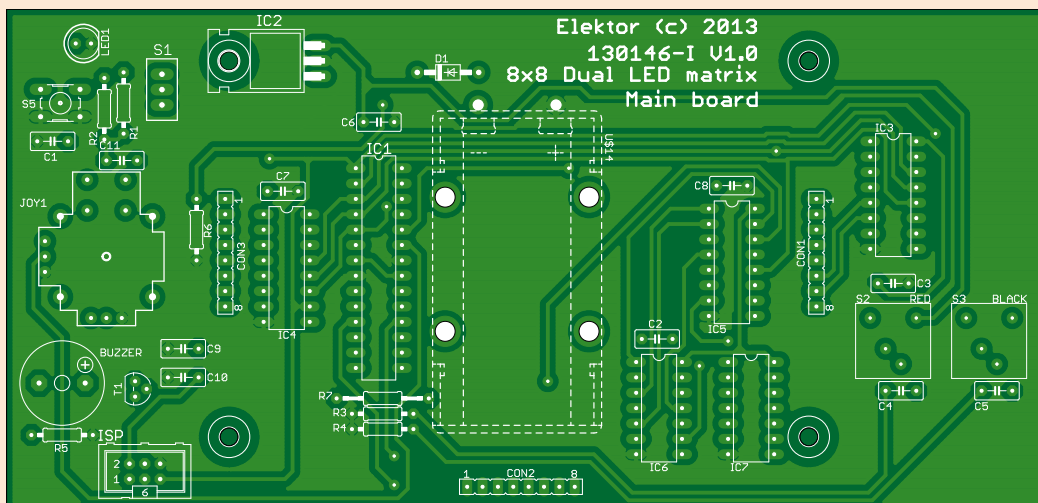


Figure 5.  
Le circuit imprimé pour la matrice de LED. Veuillez à les aligner soigneusement en 3D : rang, colonne et hauteur.



Ces deux circuits imprimés sont réduits ici à 80% de leur taille réelle.

Figure 6.  
La carte mère est conçue de telle sorte que le circuit imprimé de la matrice puisse se fixer par-dessus, centré sur elle.



## Liste des composants

### Résistances :

R1, R5 = 390  $\Omega$   
R2, R3, R4, R6 = 10 k $\Omega$   
R7 = 1 k $\Omega$   
JOY1 = joystick à 2 axes 2x10 k $\Omega$  + poussoir  
(p.ex. Conrad n° 425637-89 et 710047-89)

### Condensateurs :

C1 à C11 = 100 nF, au pas de 5 mm

### Semi-conducteurs :

D1 = 1N4004  
LED1 = LED rouge, 5 mm  
T1 = BC547  
IC1 = ATmega328P-PU (EPS 130146-41)  
IC2 = 7805  
IC3, IC4, IC5 = 74HC595  
IC6, IC7 = 74HCT04

### Divers :

ronfleur à cc 30 mA/5 V  
ISP = embase à 2x3picots à collerette  
S1 = interrupteur à glissière à contact travail  
S2 = bouton-poussoir à contact travail, capuchon rouge  
S3 = bouton-poussoir à contact travail, capuchon noir  
S5 = poussoir miniature à contact travail  
CON1, CON2, CON3 = connecteur SIL à 8 broches  
pile bloc 9 V avec coupleur  
circuit imprimé 130146-1 [1]

### Circuit imprimé pour la matrice

R1 à R16 = 120  $\Omega$   
LD1 à LD64 = LED bicolore vert/rouge,  $I_F$  = 10 mA  
(p.ex. Conrad n° 156269-89)  
SV1, SV2, SV3 = embase à 8 picots  
circuit imprimé 130146-2 [1]

cations au logiciel et de les tester rapidement. Un stabilisateur du type 7805 assure l'alimentation de tout le circuit en 5 V. La consommation totale, LED comprises, est de 50 mA au maximum. La tension d'entrée par un adaptateur secteur peut aller de 7 à 12 V<sub>CC</sub>, mais, pour jouer de temps en temps, une pile bloc de 9 V suffit, que l'on peut monter à l'arrière du circuit imprimé.

### Les circuits imprimés

L'ensemble du circuit se répartit sur deux platines distinctes, l'une pour la matrice de LED, à la **figure 5**, l'autre pour le circuit de commande, à la **figure 6**. La construction est simplifiée par l'usage exclusif de composants traversants. Pour les puces logiques, il est recommandé d'employer des supports. Si vous alimentez le montage sur piles, il est possible de visser un support pour une pile de 9 V sur le côté des soudures. La matrice de LED n'aura bel aspect que si elles sont toutes implantées en parfait alignement horizontal comme vertical et toutes à la même hauteur. Aidez-vous d'un morceau de carton à glisser entre les pattes des LED avant le soudage. Évidemment, il faut aussi les monter toutes dans le même sens, la connexion pour les rouges (dans la direction de l'inscription *Red*) se trouve du côté aplati.

### Le programme

Le logiciel s'étale sur plusieurs composantes, nous allons en présenter les plus importantes : la commande de la matrice par données sérielles, la lecture analogique de la manette et le jeu de démonstration pour la mise à l'ouvrage du circuit. Trois entrées du registre à décalage 74HC595 sont utilisées : *Serial-Data*, *Serial-Clock* et *Output-Enable*. Au rythme de l'entrée sérielle d'horloge, les huit bits présents dans le registre sont décalés d'un rang, de sorte que le bit de poids faible disparaît, tandis que le niveau logique actuel de l'entrée *Data* s'inscrit dans la cellule comme bit de poids fort.

Quand huit nouveaux bits ont ainsi été avalés par le registre, on active la ligne de validation de sortie, *Output Enable*, pour les copier dans les tampons de sortie du registre à décalage. Pour chaque trame, on utilise un tableau de 8 caractères que l'on analyse pour savoir quelles LED doivent s'allumer. Le programme vérifie d'abord s'il s'agit du premier rang, auquel cas une impulsion est envoyée au registre à décalage

#### Listage 1.

```
if(RowNbr!=0) //check if the current Row is not the first
{
    //Set First row out to high = PD5 (DDRD 0b00100000)
    PORTD&=0b11011111;
}
else if(RowNbr==0)
{
    //Set First row out to low = PD5 (DDRD 0b00100000)
    PORTD|=0b00100000;
}

//give a row clock pulse
PORTD|=0b00000100;
PORTD&=0b11110111;

for(I=0;I<8;I++) //repeat the routine 8 times to get all the bits
{

    //get the right bit value by bit shifting the bit
    if(((RedRowData>>(7-I))%2)==1)
    {
        //set the serial Red output high = PD0 (DDRD 0b00000001)
        PORTD|=0b00000001;
    }
    else
    {
        //set the serial Red output low = PD0 (DDRD 0b00000001)
        PORTD&=0b11111110;
    }

    //get the right bit value by bit shifting the bit
    if(((GreenRowData>>(7-I))%2)==1)
    {
        //set the serial Green output high = PD4 (DDRD 0b00010000)
        PORTD|=0b00010000;
    }
    else
    {
        //set the serial Green output low = PD4 (DDRD 0b00010000)
        PORTD&=0b11101111;
    }

    //give a serial clock pulse = PD3 (DDRD 0b00001000)

    PORTD|=0b00001000;
    PORTD&=0b11110111;
}
```



des cathodes communes. Chaque bit du rang est ensuite vérifié, la valeur 1 signifie actif, 0 représente inactif. Pour chaque bit actif, on active l'entrée *Serial-Data* de la couleur correspondante ainsi que l'entrée *Serial-Clock*. Dans le cas d'un bit inactif, seule l'entrée d'horloge est activée.

Le **listage 1** reproduit un fragment du code de la commande série.

La lecture de l'état de la manette de jeu analogique s'opère avec des fonctions existantes de la bibliothèque *avr/io.h* du logiciel de programmation AVR Studio 5. Voici les variables globales déclarées dans l'en-tête du fichier *program.h* pour la lecture analogique :

- *PortJoyV* – la valeur actuelle en sens vertical de la manette
- *PortJoyH* – la valeur actuelle en sens horizontal de la manette

Le **listage 2** reprend le code de *ProgramInit.ccp*. La tension de référence pour le convertisseur A/N est prise sur le 5 V qui alimente le microcontrôleur (AVcc). Le facteur de division préalable (*prescale div factor*) est choisi pour que les conversions se produisent à une vitesse adéquate. On lit ensuite dans le fichier *DataIn.ccp* l'état de chaque axe de

la manette (*joystick*). Le **listage 3** donne comme exemple la lecture de l'état de l'axe vertical.

Le programme de démonstration pour ce projet est disponible gratuitement sur [1], c'est un jeu qui consiste à diriger avec la manette une LED verte allumée dans la matrice de manière à éviter une collision avec les LED rouges qui arrivent en sens inverse.

Après une petite animation d'introduction, on lance le jeu en poussant sur le bouton rouge. Le premier niveau est alors chargé du fichier *LoadLevel.cpp* dans un tableau de 117 caractères, c'est un niveau de 8 bits de hauteur – toute la hauteur de la matrice – et 117 octets de large. La position actuelle de la LED verte est aussi mémorisée dans une variable d'un caractère que l'on peut modifier en inclinant la manette vers le haut ou vers le bas. Après chaque trame, l'état de la LED verte est comparé dans une porte ET au rang suivant du niveau. Si les deux coïncident, la boucle prend fin et un nouveau jeu commence avec le même niveau que le précédent.

Il y a aussi une fonction de test intégrée. À l'issue d'un jeu ou lors de la mise sous tension, si l'on appuie sur le bouton noir, la manette permet de promener un bloc de quatre LED sur la totalité de la matrice.

Bon amusement avec ce jeu, après quoi vous aurez tout loisir de programmer votre propre application.

(130146 – version française : Robert Grignard)

Ce projet s'inscrit dans le cadre d'un exercice pratique d'électrotechnique au *Centre régional de formation (ROC) Leeuwenborgh* à Sittard (Pays-Bas).

## Lien

[1] [www.elektor.fr/130146](http://www.elektor.fr/130146)

### Listage 2.

```
//analog reading setup
ADMUX=0b01000000; // For Aref=AVcc;
ADCSRA=0b10000110; // prescale div factor = 64
```

### Listage 3.

```
//Reading the analog signal of the joystick Vertical-Axis


//Reads vertical joystick (port 0)
ADMUX&=0b11111100;

//Start Single conversion
ADCSRA|=0b01000000; //ADSC = high

//Wait for conversion to complete
while(!(ADCSRA&0b00010000));

//Clear ADIF by writing "one" to it
ADCSRA|=(1<<ADIF);
ADCSRA&=0b10111111;

PortJoyV=ADC;
```

  
L'ORIGINAL DEPUIS 1994  
**PCB-POOL**  
Beta LAYOUT

create : electronics

**Pochoir gratuit**  
avec chaque commande  
"Prototype"

**Service Assemblage**  
A partir d'un composant

**Embedded RFID**  
authentifiez, suivez et  
protégez votre produit  
[www.magic-pcb.com](http://www.magic-pcb.com)

**NOUVEAU!**

Appel Gratuit : FR 0800 90 33 30  
[sales@pcb-pool.com](mailto:sales@pcb-pool.com)



[www.pcb-pool.com](http://www.pcb-pool.com)

**Beta**  
LAYOUT  
create : electronics

PCB-POOL® est la marque déposée de Beta LAYOUT GmbH  
chaque marque déposée appartient à son propriétaire respectif

  
**eSTORE**  
Beta LAYOUT

Développer, assembler, souder

  
**Raspberry Pi Modèle B**  
RAM 512 MO  
**€ 38,95**

**Reflow Controller**  
  
**€ 129,00**

**Starter Kit RFID UHF Basic**  
  
**€ 279,00**

**Le grand Beta-Reflow-Kit**  
  
**€ 129,00**

[www.beta-eSTORE.com](http://www.beta-eSTORE.com)

**Beta**  
LAYOUT  
create : electronics

eSTORE® est la marque déposée de Beta LAYOUT GmbH

**le best-seller de C. Valens, le chef du labo d'Elektor :**  
3<sup>e</sup> tirage en 6 mois



**maîtrisez les microcontrôleurs à l'aide d'Arduino**

352 pages – 10 chapitres – 13 réalisations inédites  
sommaire et extraits : [www.elektor.fr/enervuino](http://www.elektor.fr/enervuino)

Ce livre n'est pas banal, pas superficiel, pas barbant, pas soporifique, pas du genre *30 applications rigolotes qui servent à rien*, pas lâche, pas infantilisant, pas appuyé sur du matériel introuvable ou hors de prix,

**logiciel gratuit téléchargeable**

**mais il est** original, par le fond et la forme, consistant, profond, complet, plaisant, souvent drôle, parfois hilarant ou déconcertant. Il est généreux, car il ne se contente pas de donner des envies, mais donne les moyens de les satisfaire ; il est courageux, car il aborde les sujets laissés savamment dans l'ombre par les autres, par exemple les interruptions...

**changez de loisirs**  
**devenez dresseur de puces !**

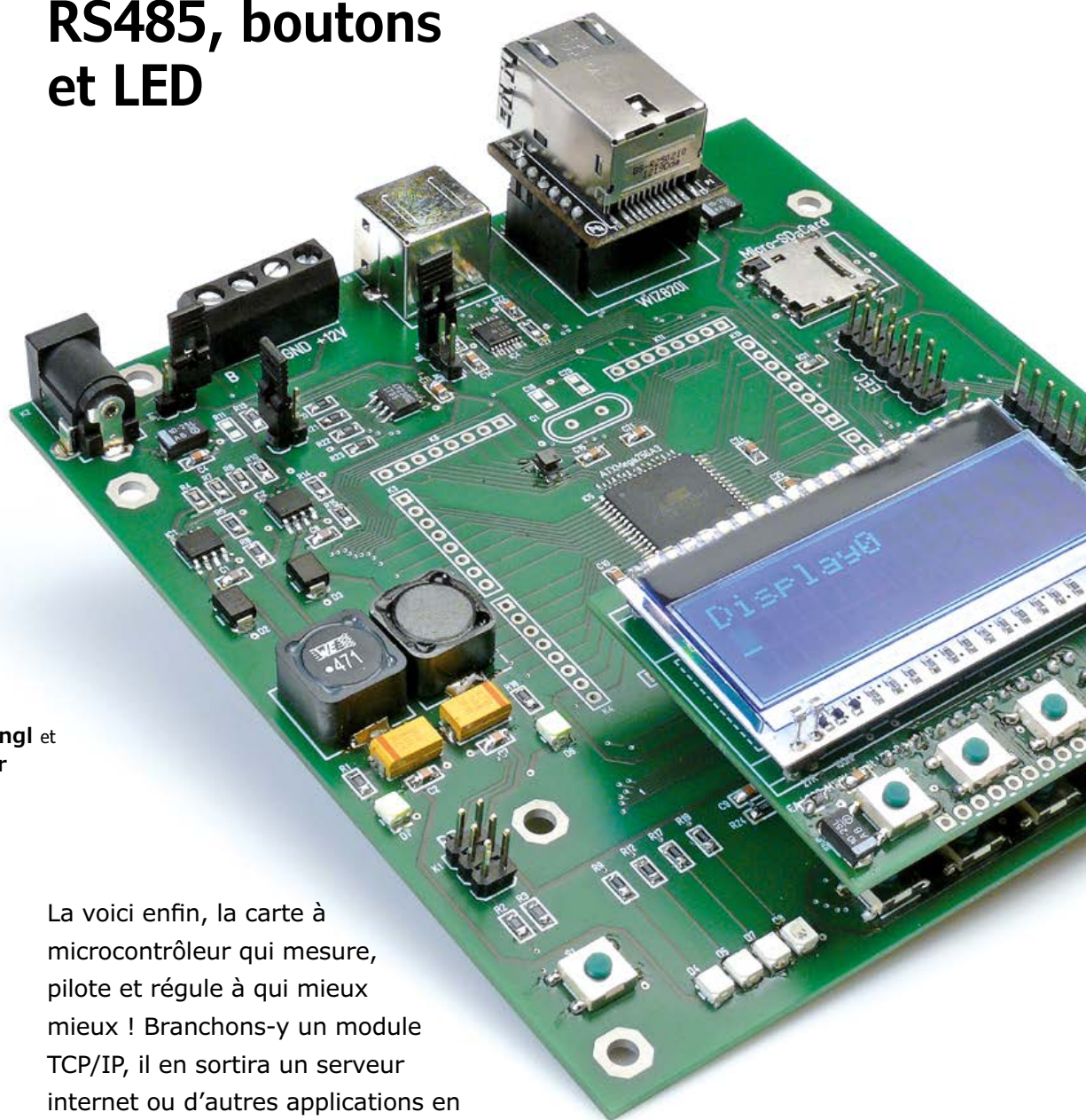
**elektor**

« À la fin du livre, le lecteur qui n'aura rien sauté sera capable de mettre en oeuvre n'importe quel microcontrôleur. »  
**Clemens Valens**



# Xmega sur carte polyvalente

avec afficheur, mémoire SD, réseau,  
RS485, boutons  
et LED



**Jens Nickel**

concepteurs : **Achim Lengl** et

**Bernd Köppendörfer**

(KöpLe Engineering)

La voici enfin, la carte à microcontrôleur qui mesure, pilote et régule à qui mieux mieux ! Branchons-y un module TCP/IP, il en sortira un serveur internet ou d'autres applications en réseau. Sa mémoire de masse réside dans une micro SD. Pour s'en servir, il y a quatre boutons, autant de LED et un afficheur amovible. Et surtout plein d'interfaces sur la même carte.



Les échos suscités par notre série d'articles sur l'ElektorBus prouvent que la mesure, la commande et la régulation ont la cote auprès de nos lecteurs. Un PC pourrait tout aussi bien servir de centrale de commutation, mais pourquoi faire appel à la grosse artillerie, coûteuse et bruyante ? Pour d'autres projets, la carte Linux d'Elektor serait un bon choix, mais nous sommes nombreux à n'être pas (encore) familiarisés avec ce système d'exploitation. Pour bien des fonctions, un contrôleur à 8 bits suffit, un de la série AVR, par exemple.

Pour le dialogue à distance avec des capteurs et des actionneurs, l'interface RS485 fait merveille, avec bien sûr l'accès à un réseau, et pour le stockage des valeurs mesurées ce sera une carte mémoire SD. Reste à régler l'interface avec l'humain : les entrées et sorties, éventuellement via un menu, un affichage de texte, quelques boutons bien placés et des voyants composeront le tableau de bord.

### Xmega à la base

À partir de cette petite liste, nous avons entamé la conception de la carte avec le diagramme de la **figure 1**. Le bureau d'ingénieurs KöpLe [1] y a apporté d'autres idées et repris le développement du montage et du circuit imprimé. Le résultat final est une carte assemblée et testée, prête à l'emploi, ainsi qu'un circuit imprimé (**fig. 2**) disponibles l'un et l'autre auprès d'Elektor [2].

Dans le choix du contrôleur, nous avons veillé à disposer de beaucoup de flash, d'autant que nous visions l'application comme serveur internet. Au populaire ATmega2560, utilisé entre autres sur la grande carte Arduino, nous avons préféré l'ATXmega256A3 qui, en plus de ses 256 Ko de flash et 16 Ko de RAM, offre de belles fonctions potentielles telles que le système événementiel perfectionné [3]. Il permet par exemple de compter facilement tous les changements de niveau sur chaque broche du GPIO (les E/S numériques polyvalentes). Sur des applications complexes, on aime disposer d'interruptions à priorité graduée. Malheureusement, le fait que les registres ne soient plus compatibles avec

## Caractéristiques techniques

- ATXmega256A3 avec 256 Ko de Flash et 16 Ko de SRAM
- 4 boutons et 4 LED
- module d'affichage de texte enfichable 3 lignes de 16 caractères avec éclairage à LED
- pilote RS485, bornes à vis pour A, B, 12 V et GND (ElektorBus)
- embases à picots pour câble USB/TTL de FTDI
- embase à picots pour convertisseur USB/TTL BOB
- autres lignes UART disponibles à l'extérieur, mini-DIN
- accès à quasi toutes les broches du contrôleur sur embases facultatives
- chevalet pour carte micro-SD, commandée par SPI
- socle pour module réseau WIZ820io, disponible chez Elektor (130076-91)
- EEC (Embedded Extension Connector) avec 3 x CA/N, 2 x GPIO, SPI, I<sup>2</sup>C sur embase à 2x7 picots, diverses cartes d'extension disponibles chez Elektor
- circuit imprimé adapté aux boîtiers Hammond 1598REGY et RS 220-995
- programmable par programmeur AVRISP et le Studio d'Atmel gratuit
- bibliothèque de logiciels en C pour toutes les unités périphériques (téléchargement gratuit)

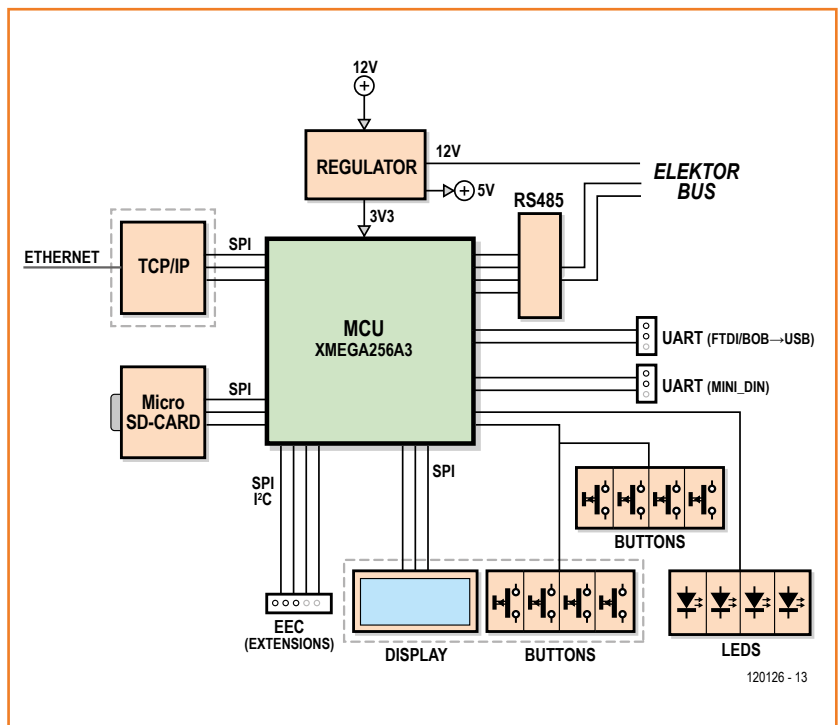


Figure 1. Le diagramme fonctionnel de la carte Elektor Xmega.

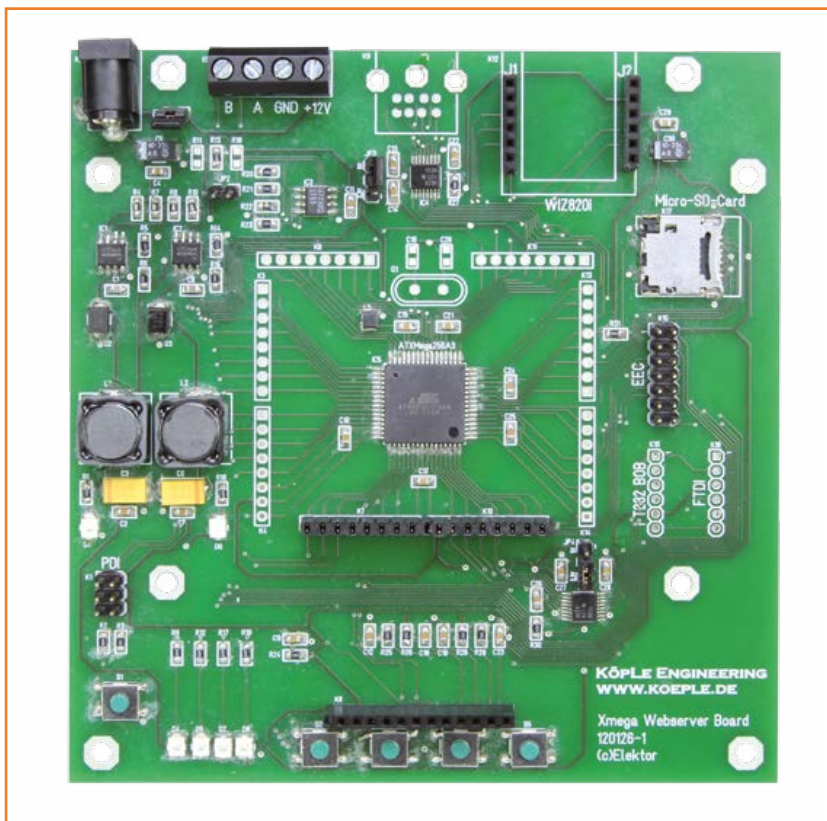


Figure 2.  
La carte Elektor Xmega est disponible assemblée et prête à l'emploi. Vous pouvez encore la doter de modules complémentaires et d'embases supplémentaires.

ceux des ATmega traditionnels apparaît comme un inconvénient, mais sachez que même les programmeurs chevronnés sur AVR passent des heures à déchiffrer les feuilles de caractéristiques.

On aura plus vite fait de boucler son code avec les pilotes pour UART, SPI et autres du fabricant, disponibles gratuitement sur l'internet, tout comme l'environnement de développement Studio 6 d'Atmel. Chez Elektor, c'est encore plus facile : avec une API (interface de programmation) pour le contrôleur et un fichier de pilote unique pour toute la carte, toutes les unités périphériques sont couvertes. Nous allons y revenir.

### Alimentations

Il faut au Xmega une tension de service de 3,3 V, c'est la tendance actuelle dans le monde de l'électronique. Il est évident que nous avons besoin aussi d'une tension de 5 V, pour l'électronique de communication conventionnelle. Les deux tensions, en provenance chacune d'un régulateur à faible dissipation MC34063A [4] sont donc distribuées sur la carte (**fig. 3**). On trouve sur internet [5] un outil fort utile pour déterminer les valeurs des composants associés.

Leurs entrées sont toutes deux alimentées sous 12 V. On peut l'appliquer sur des bornes à vis (voir ci-dessous) ou bien sur une prise jack (attention à la polarité !), selon la position de JP1. K1 est le connecteur à 6 contacts PDI pour la programmation en circuit du contrôleur. Son brochage est différent de l'ISP bien connu pour la série ATmega, mais celui-ci permet d'utiliser le programmeur AVR ISP mkII [6], relativement bon marché, qui détecte automatiquement la tension de fonctionnement de 3,3 V du Xmega. On notera encore la présence de l'indispensable bouton d'initialisation.

### Interfaces

Pour la communication entre cartes, c'est l'UART qui reste l'interface favorite. Sa programmation est facilitée par la présence, dans le contrôleur, d'UART matériels ; on y trouve aussi plusieurs convertisseurs, entre autres pour RS485 et USB. Notre contrôleur possède six de ces UART, nous en utilisons trois.

Les signaux des UART PC2/PC3 passent par une puce pilote de RS485 dont les lignes A et B sont reliées à un bornier à vis. Comme la masse est aussi disponible sur une borne à vis, il est facile de relier la carte à d'autres, équipées en RS485, comme l'ElektorBus. C'est ainsi qu'on peut, avec trois fils, raccorder notre fameux convertisseur RS485/USB [7] et de là, joindre un PC à distance. À travers le module Andropod [8], on fait pareil avec les tablettes ou les téléphones tactiles. Au moyen d'une quatrième borne à vis, on peut encore alimenter la carte en 12 V, comme l'ElektorBus.

Ce sont deux signaux numériques, DE et  $\overline{\text{RE}}$ , qui activent l'émission et la réception de la puce RS485. Le cavalier JP1 permet de terminer la ligne de bus sur 120  $\Omega$ . Les résistances facultatives R11 et R18 peuvent ramener à un niveau stable les lignes A et B, quand aucun nœud ne serait actif, et réduire ainsi la sensibilité aux interférences. En pratique, lors de nos nombreuses expérimentations, nous n'avons jamais eu besoin de ce genre de polarisation. On voit dans le schéma la prise mini DIN K9, sur laquelle se connectent un autre UART ainsi que quatre lignes de GPIO. Cette prise est facultative, on peut aussi bien souder directement un câble pour établir une liaison permanente avec un autre appareil. La même connexion par mini DIN est

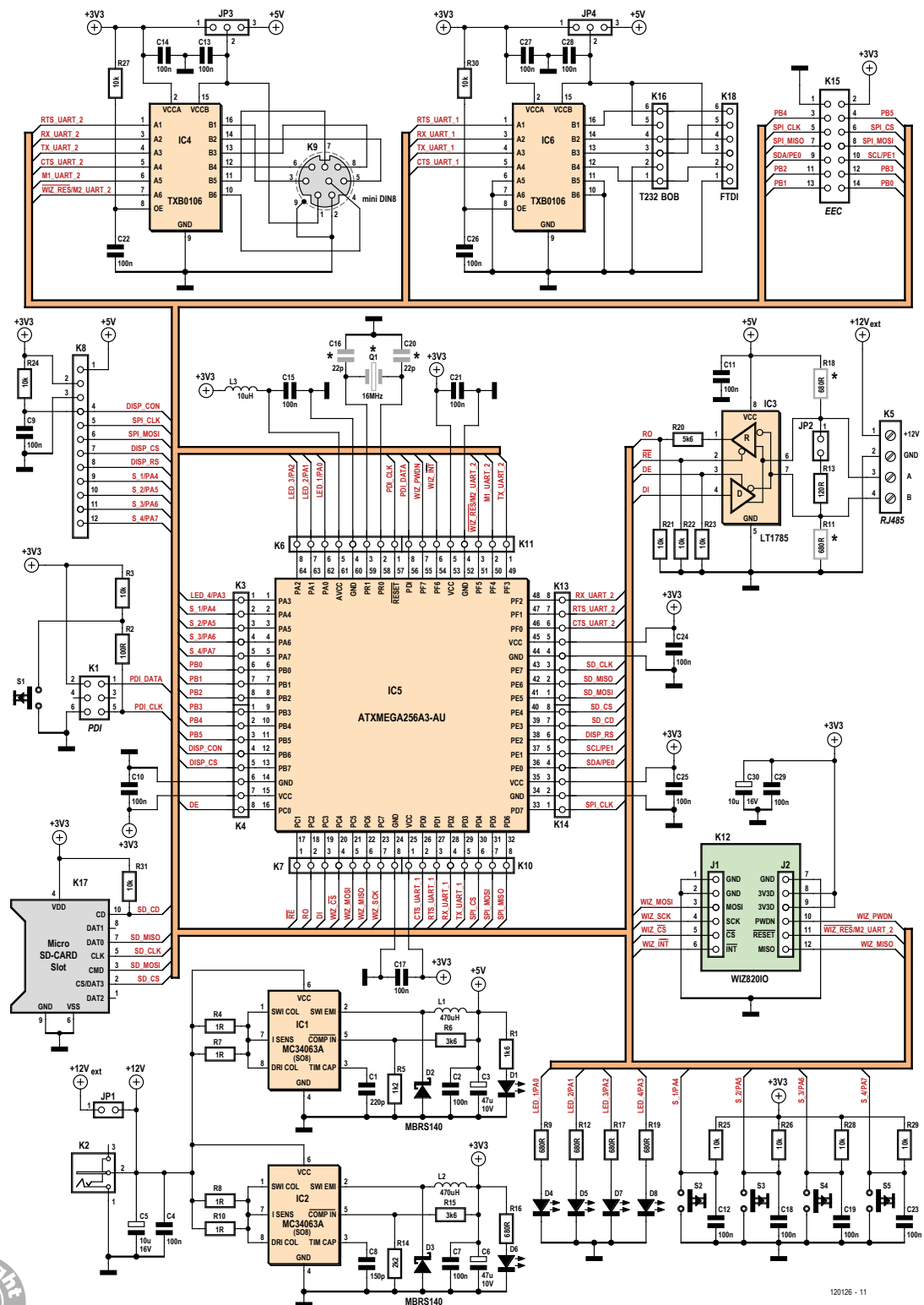
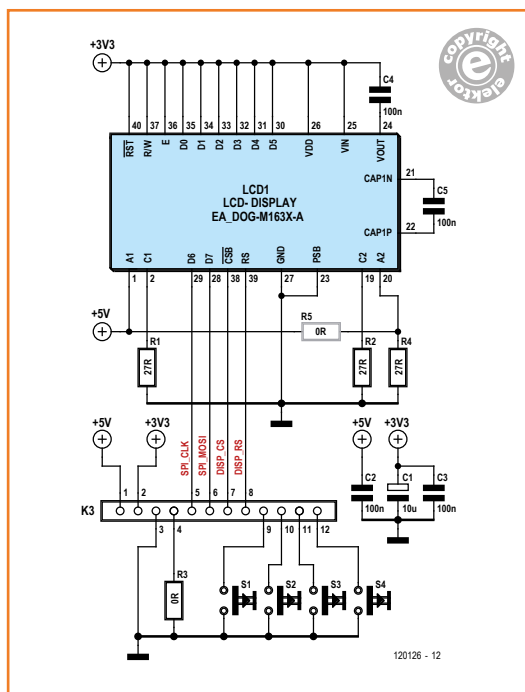


Figure 3. Le schéma de la carte Xmega. Utilisation intensive des possibilités du contrôleur avec, ntre autres, trois unités SPI, trois UART et une I2C.



Figure 4.  
Le schéma du module d'affichage : quatre boutons-poussoirs permettent de réaliser une commande par menu.



Extension Connector (EEC) à 14 contacts qui s'adapte à toutes les cartes d'extension conçues par l'équipe d'*Embedded Projects*. On y trouve un afficheur avec extension de port et horloge en temps réel [9], une carte à relais [10], un circuit imprimé pour moteur pas à pas, un capteur de température et d'autres encore [11] [12].

Pour les essais, le débogage et les extensions, on peut encore installer autour du contrôleur d'autres connecteurs mâles ou femelles pour gagner l'accès à presque toutes les broches du Xmega. Sur la carte livrée par Elektor, des connecteurs femelles sont soudés sur la face inférieure, de quoi fixer solidement le module d'affichage décrit plus loin ; ils n'ont aucune fonction électrique.

### L'interface utilisateur

Si la carte est employée comme centrale de mesure, de commande ou de régulation, un afficheur sera très précieux, mais tout le monde

## facile comme bonjour : un serveur internet et d'autres systèmes en réseau

aussi prévue sur le module Andropod. Pour ne rien laisser au hasard, JP3 permet de sélectionner l'une des tensions, 3,3 V ou 5 V aux fins de compatibilité.

Un troisième UART (PD2/PD3) est attribué à un convertisseur USB/UART. À vous de choisir entre le câble USB/TTL de FTDI ou la petite passerelle USB/série BOB-FT232R, tous deux disponibles auprès d'Elektor [2]. Ici aussi, il est possible avec JP4 d'adapter le niveau de tension, 3,3 V ou 5 V, pour rester compatible. Chacun choisira, selon ses besoins, une embase à picots droite ou coudée.

### Extension

Notre carte est déjà pas mal équipée, mais certaines applications pourraient requérir une extension de ses capacités par le biais d'autres puces. Lesquelles sont pour la plupart dotées de SPI et d'I<sup>2</sup>C, aussi avons-nous conduit les lignes de port correspondantes du contrôleur vers l'extérieur en passant par une embase. Notre idée était de réutiliser le brochage adopté par Benedikt Sauter sur la carte Linux d'Elektor : un *Embedded*

n'en a pas nécessairement l'usage. Aussi l'avons-nous conçu comme module séparé à brancher sur la carte du contrôleur. Cela laisse aussi le loisir de le placer autre part, peut-être à l'extérieur d'un boîtier.

Le connecteur K8 pour la liaison à l'afficheur dispose de 12 contacts, dont trois sont réservés aux deux tensions de fonctionnement et à la masse. L'afficheur est commandé par SPI. Aux trois lignes MOSI, SCK et CS vient s'ajouter RS qui détermine si l'octet transmis est une commande ou une donnée [13].

Si le module d'affichage est branché sur la carte, une autre broche est mise à la masse. Il s'agit de la broche 4 qui est reliée à la ligne de port PB6 du contrôleur. Si PB6 est commutée en entrée avec la résistance de polarisation haute activée, on peut déterminer en fonction du niveau si l'afficheur est branché ou pas.

Quatre autres lignes sont consacrées aux quatre boutons situés également sur le module d'affichage (figure 4). Ils ont été placés juste sous l'afficheur pour servir dans un menu. Sans afficheur, on n'a pas besoin de ces boutons.

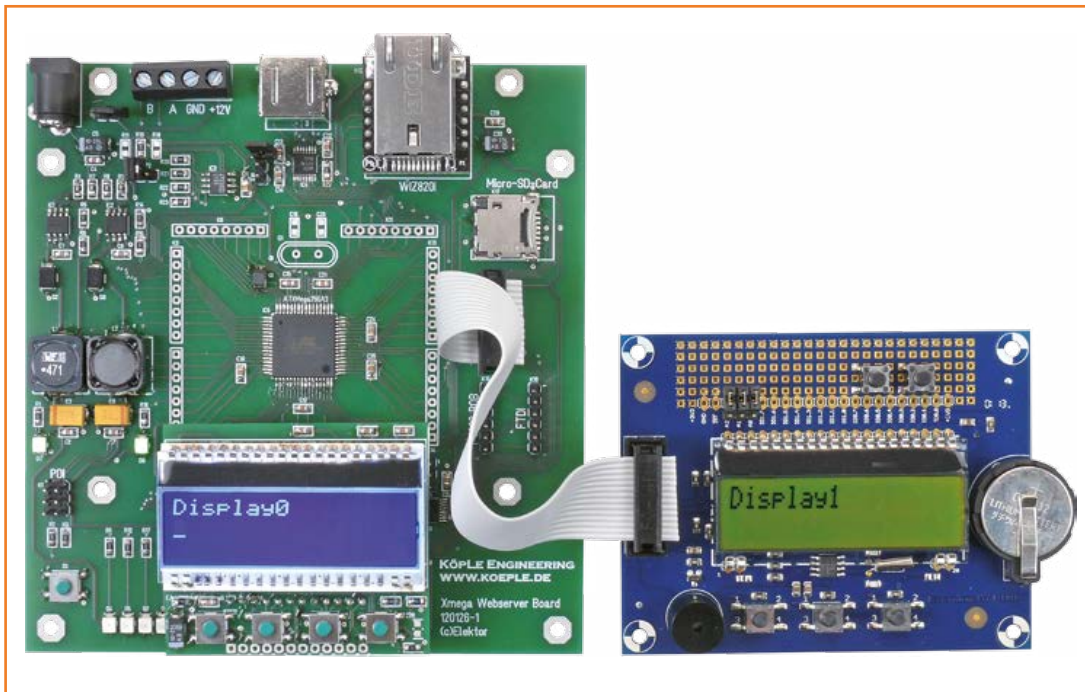


Figure 5.  
Un connecteur « Embedded Extension » (EEC) permet de raccorder différentes cartes d'extension.

Il y en a quatre aussi sur la carte, reliés aux mêmes lignes de port du contrôleur. Les condensateurs C12, C18, C19 et C23 servent d'anti-rebond. Finalement, mentionnons les quatre LED sur la carte, indispensables au débogage et à la visualisation aisée d'événements

## Carte SD et réseau

Sur notre circuit imprimé, le chevalet de carte SD accepte les micro-SD ; elles servent de mémoire de masse et sont commandées ici en mode SPI. Les quatre lignes MISO, MOSI, SCK et CS sont reliées à une unité matérielle SPI, le port E du Xmega. Comme pour les autres périphériques, nous fournissons une petite bibliothèque, du moins pour les données brutes (celles-ci ne pourront donc être lues ou écrites sur la carte SD que montée sur ce circuit imprimé). Si vous envisagez de programmer un pilote vous-même, vous trouverez de l'inspiration sur [14]. Dès qu'une carte y est introduite, la broche CD du chevalet de carte SD, reliée à la ligne de port PE3 du Xmega, est ramenée à la masse.

La vedette du projet revient certainement au module de réseau Wiz820io qui s'invite sur notre carte, il est aussi disponible [2] auprès d'Elektor (réf. 130076-91). Il s'agit d'une petite carte à processeur avec pile TCP/IP intégrée. Du coup,

notre Xmega n'a plus à se soucier de la couche de protocole du trafic réseau. Il se contente de communiquer avec le petit module pour savoir s'il faut établir un socket (une adresse IP et un port spécifiques) et s'il faut envoyer des caractères, par exemple. De cette façon, il est en mesure de récupérer les caractères reçus d'un socket. La communication entre Xmega et le module de réseau s'opère également par SPI, ici par l'unité SPI du port C. Cette fois encore, pour éviter de déborder du cadre de cet article, nous vous renvoyons à la documentation du fabricant de WizNet [15]. Les Coréens offrent également des pilotes en C pour un certain nombre de contrôleurs avec le code de l'application (p. ex. un serveur internet) pour des fonctions comme `SocketOpen(...)`. Nous avons adapté ce pilote à la carte et au Xmega vers le bas et étendu vers le haut quelques fonctions simples. Un prochain article d'Elektor parlera de l'utilisation de la carte dans un réseau domestique et sur l'internet.

## Le logiciel

La bibliothèque existante de l'ElektorBus, une petite bibliothèque d'affichage de KöpLe et le pilote de WizNet ont permis l'accouchement dans cette fameuse EFL, la bibliothèque de micrologiciels embarqués, qui a fait l'objet de

### Listage 1. Démonstration avec LED, boutons et affichage.

```
int main(void)
{
    Controller_Init();
    Board_Init();
    //Extension_Init();

    ApplicationSetup();

    while(1)
    {
        ApplicationLoop();
    }
};

void ApplicationSetup(void)
{
    LEDButton_LibrarySetup(ButtonEventCallback);

    Display_LibrarySetup();
    Display_WriteString(0, 0, "Display0");
    //Display_WriteString(1, 0, "Display1");
}

void ApplicationLoop()
{
    ButtonPollAll();
}

void ButtonEventCallback(uint8 BlockType, uint8 BlockNumber,
uint8 ButtonPosition, uint8 Event)
{
    //Buzzer(BuzzerBlockFirstIndex, 1000,
BUZZER_TONEMODE_RAMP);

    if (Event == EVENT_BUTTON_PRESSED)
    {
        ToggleLED(0, 0);
        Display_WriteNumber(0, 1, BlockNumber);
        Display_WriteNumber(0, 2, ButtonPosition);
    }
}
```

deux articles d’Elektor [16] et [17]. Depuis, le travail s’est poursuivi dans ce cadre, en particulier pour la carte Xmega-Webserver, et nous avons le plaisir d’offrir des modules de bibliothèque pour tous les périphériques de la carte.

Vous pouvez télécharger sur [2] et [18] la base de code EFL actuelle, contenant les modules de code individuels et une application de démonstration pour notre carte. L’API du contrôleur pour le Xmega se trouve comme toujours dans une paire de fichiers sous les noms ControllerEFL.h et .c. Les deux fichiers sont dans ce cas-ci dans le dossier Xmega256A3. Il contient des fonctions pour commuter et lire les E/S, interroger les broches du CA/N, émettre et recevoir par l’unité UART et bien d’autres choses. Plus besoin d’éplucher les feuilles de caractéristiques !

Le fichier de la carte contient le code qui permet d’appeler ces mêmes fonctions de contrôleur. Pour les couches supérieures d’EFL, il y a des fonctions de bas niveau pour communiquer avec les périphériques. Par exemple :

```
void Display_SendByte(uint8
DisplayBlockIndex, uint8 ByteToSend,
uint8 DATABYTE_COMMANDBYTE)
```

une fonction qui envoie un octet sur l’unité SPI indiquée, sur laquelle est branché l’afficheur ; dans le tableau interne du périphérique, il y a une référence à cette unité. Simultanément, le signal numérique RS permet de distinguer entre commande et octet de données et la fonction peut dès lors vérifier quelle broche du contrôleur y est reliée. Les couches supérieures ne connaissent pas le câblage de la carte, donc on peut programmer la bibliothèque d’affichage et l’application elle-même en toute indépendance par rapport au matériel. Peu importe à l’utilisateur de savoir si l’afficheur est piloté par SPI ou en parallèle sur quatre lignes de données, il n’a qu’à appeler la fonction Display\_SendByte().

Des fonctions similaires de bas niveau sont aussi prévues dans le fichier de la carte pour la commande de la carte SD et du module réseau. Vous trouverez la documentation des fonctions, rédigée en *Doxygen*, dans la base de code. Il y a dans le répertoire *Manuals* un document supplémentaire en allemand, anglais et français qui explique le Contenu d’EFL.



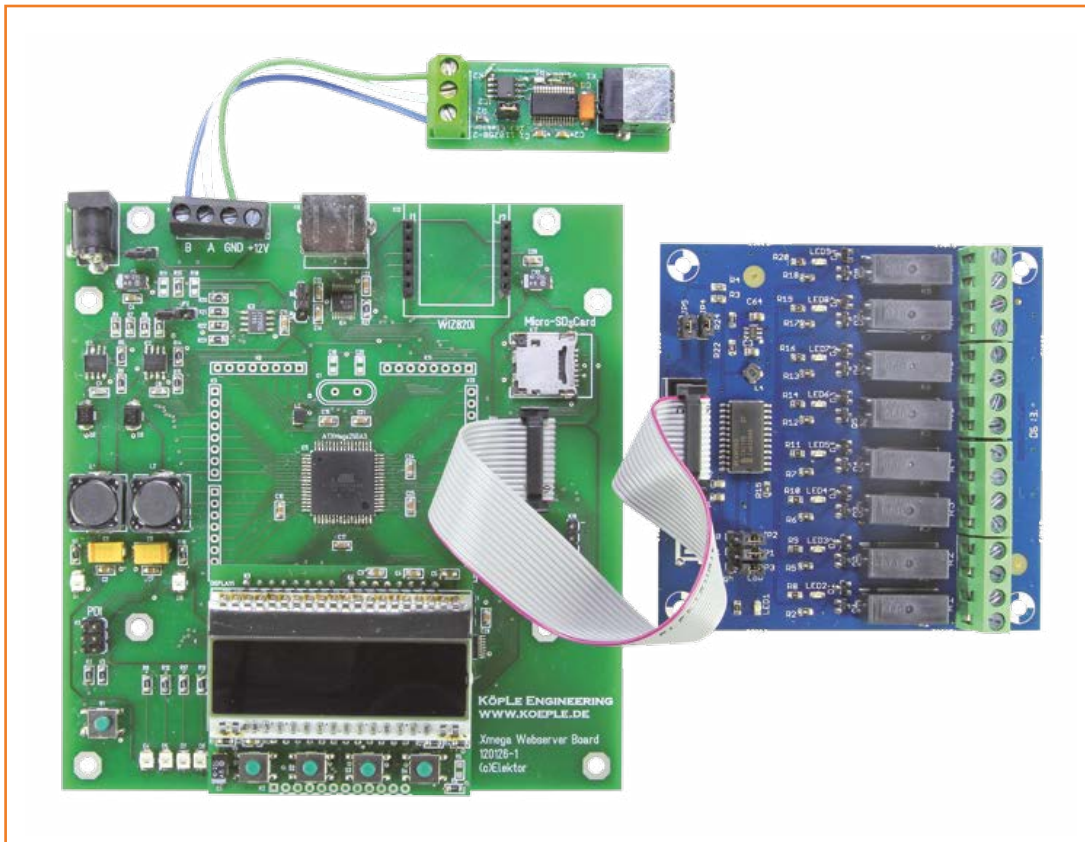


Figure 6.  
Avec huit relais, on dispose d'une puissante centrale de commutation, avec possibilité d'une commande par PC via RS485.

### Le premier programme

La bibliothèque *Display* indépendante du matériel contient le code de base dans le dossier *Libraries* : *DisplayEFL.h* et *.c*.

On initialise la bibliothèque dans le programme d'application par la commande :

```
Display_LibrarySetup();
```

Il y a quatre afficheurs et des extensions que l'on adresse alors par un numéro entre 0 et 3. Nous pouvons le tester avec l'application de démonstration *XmegaDemo* en allant dans la base de code du dossier *Applications* vers le fichier *XmegaDemo.atsIn*. Un double clic ouvre le projet dans Studio 6 d'Atmel. Quand on flashe le fichier hexadécimal correspondant dans le Xmega, il faut qu'apparaisse *Display0* dans la première ligne de l'afficheur. En poussant sur un des boutons, on fait apparaître son numéro et, sur la carte, la première LED clignote.

Le **listage 1** reproduit le code source. À côté de la bibliothèque *Display*, nous initialisons dans la fonction *ApplicationSetup* la bibliothèque

*LEDbutton* pour lui notifier vers quelle fonction sauter dans notre application quand on a appuyé sur un bouton.

Comme on interroge régulièrement les boutons, la fonction *ApplicationLoop* doit contenir la ligne de code *ButtonPollAll()* ;

On trouve dans la fonction *ButtonEventCallback(...)* le code à exécuter lors d'un événement sur un bouton. Les numéros des boutons sur la carte Xmega-Webserver (0 à 3) sont dans la variable *ButtonPosition*. La variable *Event* peut prendre la valeur *EVENT\_BUTTON\_PRESSED* (= 1) ou *EVENT\_BUTTON\_RELEASED* (= 2). On peut ainsi faire réagir différemment l'application selon que le bouton est enfoncé ou relâché.

### Extension

Si vous disposez d'une carte d'extension Linux, c'est le moment de la raccorder avec un câble plat au EEC (connecteur d'extension embarqué) comme à la **figure 5**. Spécifiques à cette carte, il y a une paire de fichiers appelés *ExtensionEFL.h* et *.c* qui apportent les fonctions de bas niveau nécessaires au bon fonctionnement des périphériques de la carte d'extension. Les fichiers

## Les concepteurs

Bernd Köppendörfer et Achim Lengl ont terminé en 2009 leurs études d'ingénieur en électrotechnique et en techniques de l'information à la Grande École Georg Simon Ohm de Nuremberg. Ils ont fondé en 2010 leur propre entreprise, la *KöpLe Engineering GbR*, dont le siège est à Oberasbach (en Bavière). Depuis lors, ils soutiennent d'autres firmes comme consultants et se chargent de projets de développement dans le domaine des circuits analogiques et numériques, depuis des ensembles simples jusqu'à la mise en oeuvre de FPGA pour des systèmes complexes de traitement d'images en temps réel.

### Listage 2. Commande par RS485/UART.

```
int main(void)
{
    Controller_Init();
    Board_Init();
    Extension_Init();

    ApplicationSetup();

    while(1)
    {
        ApplicationLoop();
    }
};

void ApplicationSetup(void)
{
    UARTInterface_LibrarySetup();
    UARTInterface_SetBaudrate(0, 38400);

    BlockProtocol_LibrarySetup(UARTInterface_Send, 0,
    UARTInterface_GetRingbuffer(0));
}

void ApplicationLoop()
{
    BlockProtocol_Engine();
}
```

sont inclus dans le projet, il n'y a plus qu'à effacer les commentaires avant la ligne `Extension_Init()`. À faire en même temps que les autres lignes de commentaires dans le code du fichier principal. Après compilation et flashage du programme dans le Xmega, les trois boutons de la carte d'extension sont aussi interrogés. Au moyen de la variable `BlockNumber`, nous pouvons distinguer dans la fonction `ButtonEventCallback(...)` le bloc de boutons dans lequel l'événement s'est produit :

0 = carte contrôleur, 1 = carte d'extension.

On peut voir que pour l'application, cela ne fait aucune différence si ce sont les boutons ou l'afficheur de la carte contrôleur ou de celle d'extension. C'est remarquable dans la mesure où les boutons que l'on interroge sur la carte d'extension Linux ne sont pas branchés sur des entrées numériques du contrôleur, mais bien analogiques (voyez le passage sur les broches de port virtuelles dans le document annexe à EFL).

### Commuter pour régner

Comme dans les précédents articles, connectons-nous à un PC par RS485 sur le convertisseur RS485/USB [6] bien connu. Branchons sur le connecteur EEC la carte « huit relais et plus encore » [10] du numéro de juillet/août 2013 ; l'ensemble devrait correspondre à la figure 6. Comme application, appelons cette fois `XmegaRelay.atsln`. L'application proprement dite est particulièrement courte (**listage 2**). Ce qui se passe dans les fonctions `ApplicationSetup` et `ApplicationLoop` est détaillé dans « commande de cartes par le PC » [17] du même numéro. Préparons sur la carte les interfaces UART pour RS485 et FTDI/BOB, initialisons aussi une bibliothèque qui fournit un protocole simple de commande du nom de `BlockProtocol`.

Dans la fonction `ApplicationLoop`, avec la ligne

```
BlockProtocol_Engine();
```

nous demandons si une nouvelle commande du PC est arrivée à la carte.

Après flashage du fichier hexadécimal correspondant, utilisons un programme de terminal sur le PC pour ouvrir le port COM adéquat et fixer un débit binaire de 38400 bauds. Il peut être nécessaire de configurer le programme de terminal choisi pour qu'après un appui sur la touche Entrée,

le caractère soit suivi d'un CR (ASCII 13).  
La saisie de

R 0 0 + <ENTER>

devrait activer le premier relais, tandis que

R 0 0 - <ENTER>

aura l'effet inverse. Les sept autres relais se commandent de la même manière avec R 0 x ... (x = 1 à 7)

Si vous n'avez pas de convertisseur USB/RS485, vous pouvez vous relier au PC avec un câble FTDI ou une passerelle BOB. Au lieu de la première, il faut alors choisir la deuxième interface UART sur

## Liste des composants de la carte Xmega

### Résistances :

(CMS 0805)

R1 = 1,6 k $\Omega$

R2 = 100  $\Omega$

R3, R21 à R31 = 10 k $\Omega$

R4, R7, R8, R10 = 1  $\Omega$

R5 = 1,2 k $\Omega$

R6, R15 = 3,6 k $\Omega$

R9, R12, R16, R17, R19 = 680  $\Omega$

R11, R18 = 680  $\Omega$  (facultatives)

R13 = 120  $\Omega$

R14 = 2,2 k $\Omega$

R20 = 5,6 k $\Omega$

### Condensateurs :

C1 = 220 pF (0805)

C2, C4, C7, C9 à C15, C17, C18, C19, C21 à C29 = 100 nF (0805)

C3, C6 = 47  $\mu$ F (tantale, 10 V, CMS-D/E)

C5, C30 = 10  $\mu$ F (tantale, 16 V, CMS-C)

C8 = 150 pF (0805)

C16, C20 = 22 pF (facultatif)

### Inductances :

L1, L2 = 470  $\mu$ H (ferrite, PIS4728)

L3 = 10  $\mu$ H (LQH3C)

### Semi-conducteurs :

D1, D4 à D8 = LED LG T67K (PLCC2)

D2, D3 = MBR5140 diode Schottky

IC1, IC2 = MC34063A régulateur à commutation (SO8)

IC3 = LT1785CS8 pilote RS485 (SO8)

IC4, IC6 = TXB0106 changeur de niveau (TSSOP16)

IC5 = ATXmega256A3-AU (TQFP64)

### Divers :

JP1, JP2 = embase 1x2, avec cavalier, au pas de 2,54 mm

JP3, JP4 = embase 1x3, avec cavalier, au pas de 2,54 mm

K1 = embase 2x3, au pas de 2,54 mm

K2 = prise jack 2,5 mm encartable

K5 = borne à 4 vis, au pas de 5,08, encartable

K8 = embase femelle 1x12, au pas de 2,54 mm

K9 = prise mini-DIN à 8 contacts (facultative)

K12 = 2x embase femelle 1x6, au pas de 2,54 mm, pour module Wiz820io

K15 = embase 2x7, au pas de 2,54 mm

K16 = embase 1x6, au pas de 2,54 mm, pour convertisseur USB/TTL BOB (facultative)

K17 = chevalet pour carte micro-SD

K18 = embase 1x6, au pas de 2,54 mm, pour câble USB/TTL de FTDI (facultative)

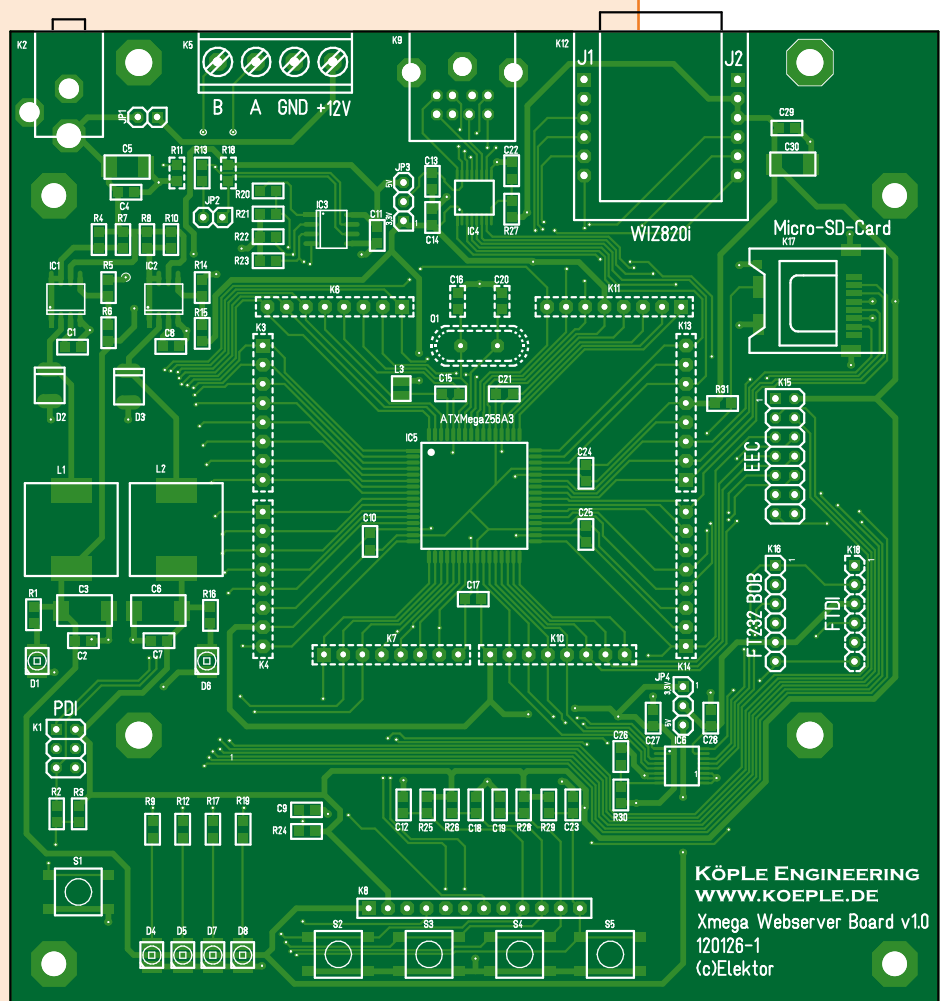
Q1 = quartz 16 MHz (facultatif)

S1, S2, S3, S4, S5 = poussoir bas CMS

circuit imprimé 120126-1

ou

120126-91 carte de contrôleur assemblée, testée, prête à l'emploi



KÖPPE ENGINEERING  
WWW.KOEPLER.DE  
Xmega Webserver Board v1.0  
120126-1  
(c)Elektor





## Liste des composants du module d'afficheur

### Résistances :

(CMS 0805)

R1, R2, R4 = 27  $\Omega$

R3 = 0  $\Omega$

R5 = 0  $\Omega$  (facultatif)

### Condensateurs :

C1 = 10  $\mu$ F (tantale, 16 V, CMS-C)

C2 à C5 = 100 nF (0805)

### Divers :

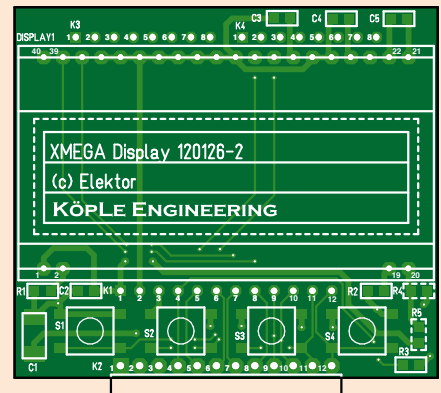
DISPLAY1 = afficheur EA-DOG-M163X-A  
avec LED 55X31 + embases femelles 1x20, 1x2, 1x2  
pour l'assemblage

K1 = embase 1x12, au pas de 2,54 mm

K2 = embase 1x12, au pas de 2,54 mm (facultative)

K3, K4 = embase 1x8, au pas de 2,54 mm

S1, S2, S3, S4 = poussoir bas CMS



circuit imprimé 120126-2

ou

module d'affichage complet et testé 120126-92

la carte, celle qui porte le numéro #1. Il n'y a qu'une ligne de code du programme à changer :

```
BlockProtocol_LibrarySetup(UARTInterface_
Send, 0, UARTInterface_GetRingbuffer(0));
```

en

```
BlockProtocol_LibrarySetup(UARTInterface_
Send, 1, UARTInterface_GetRingbuffer(1));
```

Et du coup, les messages empruntent l'autre

canal UART. D'ailleurs, on peut aussi recevoir les commandes par TCP/IP avec :

```
BlockProtocol_LibrarySetup(IPInterface_
Send, 0, IPInterface_GetRingbuffer(0));
```

C'est ce que nous essaierons de faire dans le deuxième article sur cette carte !

(120126 - version française : Robert Grignard)

## Liens

[1] [www.koeple.de](http://www.koeple.de)

[2] [www.elektor.fr/120126](http://www.elektor.fr/120126)

[3] [www.atmel.com/Images/Atmel-8331-8-and-16-bit-AVR-Microcontroller-XMEGA-AU\\_Manual.pdf](http://www.atmel.com/Images/Atmel-8331-8-and-16-bit-AVR-Microcontroller-XMEGA-AU_Manual.pdf)

[4] [www.onsemi.com/pub\\_link/Collateral/MC34063A-D.PDF](http://www.onsemi.com/pub_link/Collateral/MC34063A-D.PDF)

[5] [www.a-z-e.de/index.php?file=mc34063.php](http://www.a-z-e.de/index.php?file=mc34063.php)

[6] [www.atmel.com/tools/AVRISPMKII.aspx](http://www.atmel.com/tools/AVRISPMKII.aspx)

[7] [www.elektor.fr/110258](http://www.elektor.fr/110258)

[8] [www.elektor.fr/110405](http://www.elektor.fr/110405)

[9] [www.elektor.fr/120596](http://www.elektor.fr/120596)

[10] [www.elektor.fr/130157](http://www.elektor.fr/130157)

[11] [www.elektor.fr/130212](http://www.elektor.fr/130212)

[12] [www.elektor.fr/gnublin](http://www.elektor.fr/gnublin)

[13] [www.lcd-module.de/eng/pdf/doma/dog-me.pdf](http://www.lcd-module.de/eng/pdf/doma/dog-me.pdf) (en anglais)

[14] [http://elm-chan.org/docs/mmc/mmc\\_e.html](http://elm-chan.org/docs/mmc/mmc_e.html)

[15] [www.wiznet.co.kr/WIZ820IO](http://www.wiznet.co.kr/WIZ820IO)

[16] [www.elektor.fr/120668](http://www.elektor.fr/120668)

[17] [www.elektor.fr/130154](http://www.elektor.fr/130154)

[18] [www.elektor-labs.com/efl](http://www.elektor-labs.com/efl)

# OBTÉNEZ LA DERNIÈRE VERSION !

## Nouvelles fonctionnalités de la version 6.4:

- La simulation de schémas EAGLE dans LTspice IV
- L'affichage et la fonction de recherche des attributs pour le dialogue ADD- et REPLACE-
- L'importation de fichiers de design depuis P-CAD, Altium et Protel dans ACCEL-ASCII
- Un guide utilisateur et le pré-réglage (astuces sur les outils, raccourcis) optimisés

# EAGLE

# V6



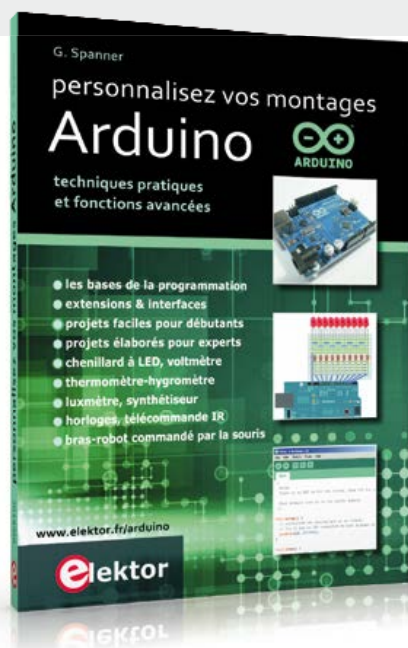
[www.cadsoft.de](http://www.cadsoft.de)

25 years CadSoft



## Personnalisez vos montages **Arduino**

techniques **pratiques** et fonctions **avancées**



# NOUVEAU

L'objectif de ce livre est de vous emmener à pas guidés vers la maîtrise d'Arduino. Les projets, regroupés par thème, accompagnés de bases théoriques, sont des applications concrètes : chenillard à LED, voltmètre, thermomètre numérique, horloges sous différentes formes, ou encore bras de robot commandé par la souris.

Vous apprendrez ainsi à exploiter des techniques essentielles comme la conversion analogique-numérique, la modulation de largeur d'impulsion, ou encore les interruptions. Après avoir mené à bien tous ces projets vous maîtriserez les fondamentaux de la technique des microcontrôleurs.

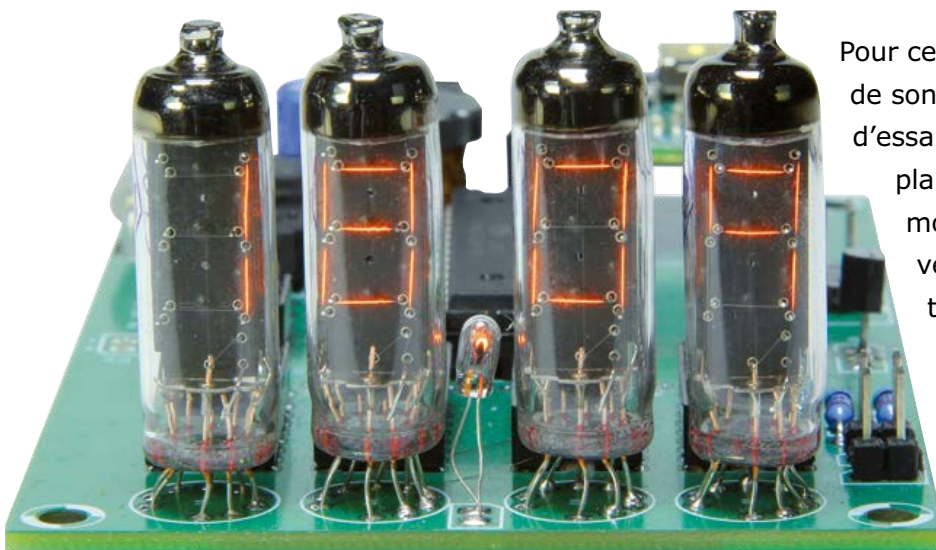
272 pages | ISBN 978-2-86661-191-0 | 34,50 €

# elektor

[www.elektor.fr/arduino](http://www.elektor.fr/arduino)

# horloge thermomètre à Numitrons

## développée avec Arduino



Pour cet article, j'ai tenu à sortir l'Arduino de son monde de *shields* et de plaques d'essais pour l'intégrer à un projet qui plaise aussi aux fans d'AVR. Ces deux mondes, bien que proches, sont souvent perçus comme inconciliables, tout comme les microcontrôleurs modernes et les tubes de l'Union Soviétique des années 50. Voici la preuve du contraire.

**Paul Court**  
(Royaume-Uni)

Las des horloges à afficheurs 7 segments à LED, je me suis tourné vers des afficheurs bien plus jolis : les tubes Numitron. Les Numitrons sont des cousins des Nixies que les habitués d'Elektor connaissent bien. La seule différence c'est qu'ils se contentent d'une alimentation de 4 V au lieu des 150 V (et plus) demandés par leurs cousins. Expérimenter avec ces tubes est donc très facile et sans danger. Tout comme les Nixies, on trouve les Numitrons à des prix raisonnables sur des sites comme eBay. Notre projet, basé sur un Arduino UNO, sera construit sur une seule carte.

Grâce à la compatibilité Arduino, vous pourrez tout développer depuis l'EDI du même nom. Le projet ne comporte ni shields, ni modules ou autres cartes enfichables : juste un ATmega328 entouré des composants nécessaires.

### Ainsi tron, tron, tron ...

*Dekatron, Klystron, Thyatron, Magnetron, Trochotron.* Et quoi après ? Un *Ennuitron* ou un *Patron* ? Ceux qui n'ont connu que l'électronique sous basse tension approfondiront la question sur [www.radiomuseum.org](http://www.radiomuseum.org) et admireront d'autres étonnants tubes en *-tron* datant de cette période lointaine où l'internet n'existait pas encore.

Nul besoin d'être un cador en latin pour comprendre qu'un *Numitron* a quelque chose à voir avec les nombres. En effet, un tube Numitron est capable d'afficher les chiffres de 0 à 9 ainsi que quelques lettres à l'aide de sept segments qui sont en fait les filaments de petites lampes. Beaucoup de Numitron comportent un genre de virgule (symbole décimal) sous la forme d'un huitième segment dans leur coin inférieur droit. Les segments d'un Numitron partagent le positif de l'alimentation et sont reliés à des fils, pas

### Caractéristiques

- quatre tubes Numitron IV-9
- affiche l'heure avec les secondes, la date, l'année et la température
- microcontrôleur ATmega328
- développé sur une plateforme Arduino
- alimentation à découpage réglable intégrée pour un réglage optimal de la luminosité
- alimentée par un adaptateur 9-12 VDC 500 mA
- pas de haute tension
- code source gratuit
- conception par Elektor.Labs



des broches, de l'autre côté. Comme beaucoup de Nixies, les Numitrons sont des tubes à fils. La lumière des Numitrons est plus chaude que celle des afficheurs 7 segments à LED. Même s'ils datent de la guerre froide, quand les tubes étaient rois, on peut encore les trouver sur l'internet parmi les vieux stocks vendus par des russophones, souvent ukrainiens. C'est sur eBay (à n'en pas douter américain) que Luc Lemmens, du laboratoire Elektor, a trouvé les Numitrons IV-9 qui lui ont permis de reproduire ce montage. Ils sont arrivés intacts malgré l'emballage amusant, mais plus que douteux d'un vendeur pourtant très bien noté. Je vous laisse vous délecter de la petite histoire en photos (**fig. 1**).

Les principales caractéristiques des IV-9 sont données dans le **tableau 1**. Vous aurez maintenant compris (nous vous l'avions déjà dit dans l'article sur le VU-mètre à tubes Nixie IN-9 [1]), en tout cas nous aimons à le penser, que ces tubes soviétiques répondaient au cahier des charges suivant : (1) doit être sujet à une large interprétation, (2) les tolérances doivent être encore plus larges, et (3) les marquages au pochoir en caractères cyrilliques sont obligatoires.

## Le circuit

Après un coup d'œil sur le schéma (**fig. 2**), vous serez sans doute surpris de constater que l'affichage n'est pas multiplexé et que chacun des chiffres de l'horloge est piloté par son propre SN74LS47D. Comme les IV-9 sont des lampes à filament, pas rapides donc, on ne peut pas vraiment les multiplexer sans que l'affichage ne scintille ou soit trop peu lumineux. Chaque voie possède donc son propre circuit de pilotage commandé par quatre entrées en BCD excepté V1, le chiffre le plus significatif, qui n'en nécessite que deux. Utiliser quatre 74LS47 et aucun verrou ou multiplexeur va mobiliser plus de broches sur l'ATmega, mais l'écriture du code sera plus facile. Étant donné que chaque tube est commandé par ses propres entrées BCD, il suffit de changer leur état pour changer l'affichage.

Côté alimentations, le circuit en possède deux : le +5 V provient d'un ordinaire régulateur 78L05 (IC1) et le +4,00 V (réglable avec P1) d'un petit régulateur à découpage réalisé autour du classique MC34063 (IC2). Les filaments des IV-9 acceptent une tension dans la gamme de 3,15 à 4,50 V ; en ajustant P1, vous obtiendrez la luminosité qui vous convient. Chaque filament consomme environ 19,5 mA. L'ensemble de l'hor-

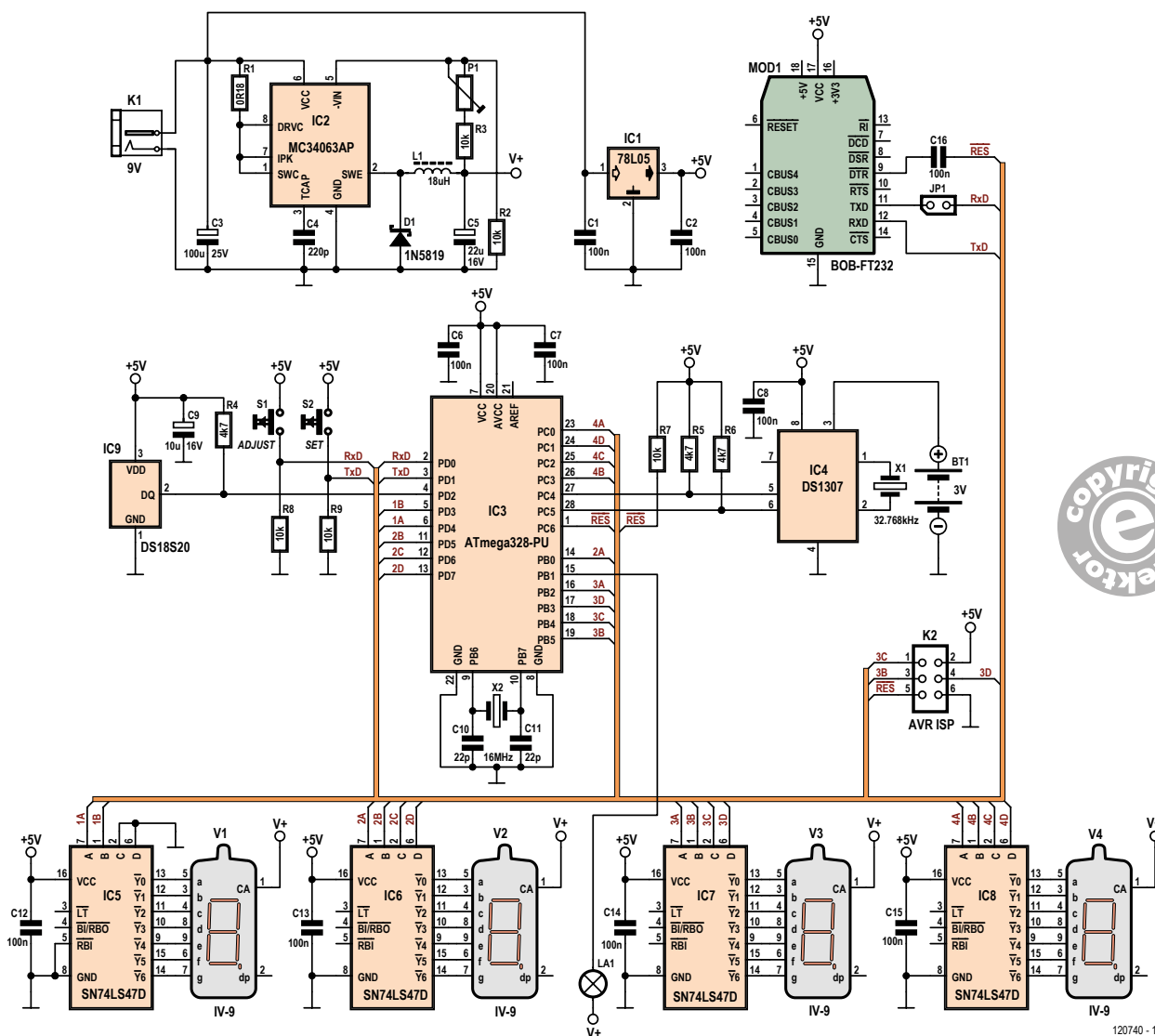


Figure 1.  
Voici les Numitrons arrivés droit d'Ukraine, avec de quoi lire en prime ! Je les ai payés en dollars et ils ont passé la douane sans problème.

Figure 2.  
Schéma de notre horloge  
thermomètre à Numitrons.  
Vous remarquerez que  
chacun des Numitrons  
V1 à V4 est piloté  
individuellement, sans  
multiplexage.

Tableau 1. Tentative de caractéristique des IV-9

Type	IV-9 (IB-9)
marque	Reflector/ Sovtec
substitut(s)	-
symboles affichés	7 segments, séparateur décimal
hauteur des symboles	10 mm
dimensions	11 mm dia. X 35 mm
diamètre des broches	0,5 mm
tension d'alimentation (max ?)	4,5 V
courant par segment	19,5 mA
base	TO-100, broche 1 sur le devant
Connecteur	NJET (= aucun) : connexion par fils



copyright  
elektor

120740 - 11

loge nécessite une alimentation en 7,5 V continus capable de fournir 500 mA, mais qui n'aura pas besoin d'être régulée.

Le choix de deux tensions d'alimentation à 5 V et 4 V (3,00 V-4,50 V) plutôt qu'une unique valant 3,3 V a été fait pour deux raisons :

- la plus haute tension permet à l'ATMega de fonctionner à 16 MHz
- 3,3 V serait trop peu pour que les sorties LS-TTL des 74LS47D puissent s'approcher suffisamment de 0 V à l'état bas pour alimenter les filaments depuis le V+.

Au cœur de l'horloge se trouve un ATmega328P, la même puce que sur les cartes UNO et Duemilanove de la série Arduino. Si vous achetez une puce programmée avec le chargeur de démarrage Arduino, préférez la version Duemilanove. La programmation de l'ATmega328P se fera via le connecteur ISP (*In-System Programming*) à 6 broches habituel à l'aide d'AVR Studio ou par l'intermédiaire de la platine Elektor BOB FT232. Dans ce dernier cas, c'est l'EDI Arduino (V1.0 ou ultérieure) qui utilisera le chargeur de démarrage Duemilanove se trouvant dans la puce. L'EDI Arduino ne fera pas la différence avec une carte Arduino *Duemilanove*.

L'horloge en temps réel (RTC) est un DS1307, le capteur de température un DS18B20 ; ces deux composants sont courants et vous n'aurez pas de mal à trouver de l'aide dans les communautés Arduino et AVR. Les données sont lues par l'ATMega en utilisant le protocole One-Wire sur les deux dernières broches disponibles : PD2 (température) et PC4 (RTC).

Le classique connecteur ISP K2 permet de reprogrammer l'ATMega sur son support. MOD1, une platine Elektor BOB-FT232, est optionnelle. Elle permet de raccorder le montage à un bus USB et vous sera utile si vous avez l'intention de bidouiller le logiciel depuis votre environnement Arduino.

## Le logiciel

Vous trouverez code source et les fichiers hexadécimaux sur le site d'Elektor [1]. Des puces programmées sont également disponibles dans l'e-shoppe (réf. 120740-41).

Le code a été écrit pour boucler toutes les 500 ms, ce dont témoigne la lampe La1 qui s'allume et s'éteint une fois par seconde. Cela permet aussi à la RTC de récupérer entre deux lectures : si vous l'interrogez trop souvent, elle renvoie des données erronées ! La1 est une ampoule minia-

ture de type grain de blé (GOW) que l'on utilise également dans les maisons de poupées et les modèles réduits ferroviaires.

À chaque tour de boucle, le code commence par vérifier si le bouton SET (S2) a été actionné avant de changer, si besoin, l'état de l'ampoule. Puis la RTC est lue et, à l'aide d'une simple division d'entiers, les heures et minutes sont chacune séparées en unités et dizaines qui seront affichées sur les Numitrons adéquats.

Par souci de simplicité, chacun des chiffres est piloté par sa propre routine. Elles fonctionnent toutes de la même manière à l'exception de celle pilotant les dizaines d'heures qui ne produit que deux bits (pour IC5/V1). Enfin, avec un peu de magie et surtout quelques ET bit à bit, les nombres décimaux sont convertis en binaire compréhensible par les 74LS47. Vous trouverez plus d'informations sur les opérations binaires sur [2]. Ça peut vous paraître tordu, mais c'est en fait très simple : il s'agit là d'une technique très utile pour convertir des nombres décimaux en binaire. Deux fonctions supplémentaires viennent égayer le montage. La première est appelée 10 s après le début de chaque minute, la deuxième après 30 s. Ces fonctions sont chargées respectivement de lire et d'afficher la température en degrés Celsius. Comme pour l'heure, modules et opérations bit à bit sont utilisés pour prendre la mesure, la convertir en binaire puis l'afficher via les 74LS47. L'exception cette fois-ci se trouve sur le chiffre de poids le plus faible (V4) qui, recevant le nombre décimal 10, affiche le 'c' venant compléter l'affichage ; par exemple '22.5c'. Comme le 74LS47 est incapable d'afficher 'f', il n'était pas possible de prendre en charge l'affichage en degrés Fahrenheit.

Enfin, la fonction *showdate()* est appelée à la cinquantième seconde de chaque minute pour afficher la date, le mois puis l'année.

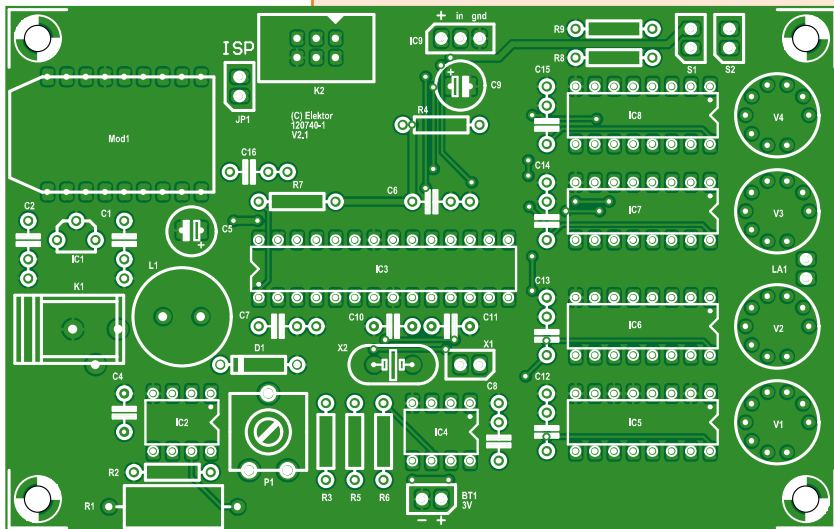
Le code est rudimentaire et je suis sûr que vous trouverez d'autres fonctions à rajouter à l'horloge (un temporisateur pour les œufs à la coque ?). Je vous invite à rejoindre le projet sur Elektor Labs [4] et à nous montrer votre code, vos mises à jour et vos bidouilles diverses. Venez partager vos idées et n'hésitez pas à faire des suggestions.

## Assemblage et utilisation

Comme il ne comporte aucun CMS, ce projet devrait être facile à assembler, que ce soit dans votre cuisine ou dans votre atelier suréquipé. La carte présentée en **figure 3** correspond à la liste des com-



Figure 3.  
Le circuit imprimé double  
face conçu par Elektor.Labs.



## Liste des composants

### Résistances

R1 = 0,18  $\Omega$  3 W  
R2,R3,R7,R8,R9 = 10 k $\Omega$   
R4,R5,R6 = 4,7 k $\Omega$   
P1 = 4,7 k $\Omega$  aj. hor.

### Condensateurs

C1,C2,C6,C7,C8,C12,C13,C14,C15,C16 = 100 nF  
C3 = 100  $\mu$ F 25 V radial

C4 = 220 pF  
C5 = 22  $\mu$ F 25 V radial  
C9 = 10  $\mu$ F 50 V radial  
C10,C11 = 22 pF

### Inductances

L1 = bobine d'arrêt 18  $\mu$ H, 3,4 A, 36 m $\Omega$  (Panasonic ELC10D180E)

### Semiconducteurs

D1 = 1N5819  
IC1 = 78L05  
IC2 = MC34063  
IC3 = ATmega328-PU programmé, réf. 120740-41 [2]  
IC4 = DS1307  
IC5,IC6,IC7,IC8 = 74LS47  
IC9 = DS1820

### Divers

V1,V2,V3,V4 = tubes Numitron IV-9  
JP1 = barrette sécable deux voies au pas de 2,54 mm + cavalier  
X1 = résonateur à quartz 32,768 kHz  
X2 = résonateur à quartz 16 MHz  
MOD1 = BOB-FT232R (optionnel) réf. Elektor 110553-91  
Bt1 = pile CR2032  
S1,S2 = poussoirs SPST pour face avant  
K1 = jack d'alimentation 2,1 mm encartable  
K2 = connecteur 6 voies deux rangées au pas de 2,54 mm  
La1 = ampoule 5 V 300 mW GOW  
circuit imprimé réf. 20740 [2]

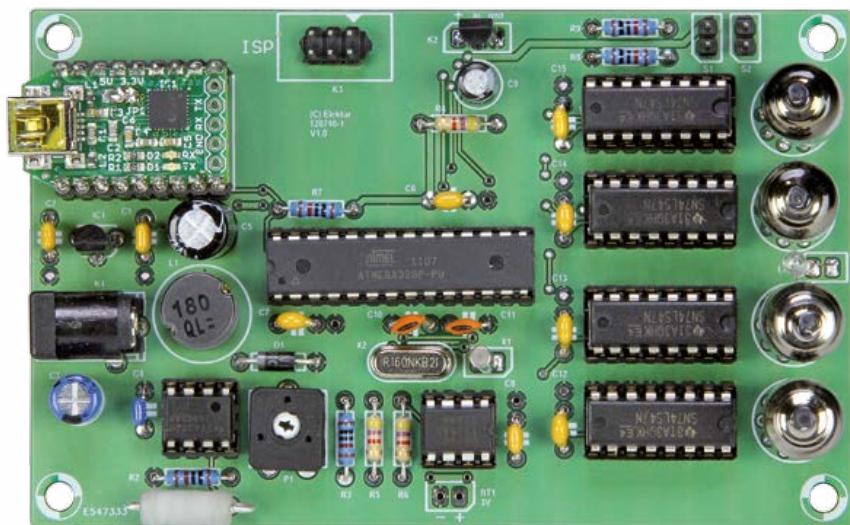


Figure 4.  
La carte assemblée et  
testée, prête à l'emploi.

posants. La version assemblée (**fig. 4**) pourra vous servir de modèle pour le choix des composants, l'assemblage et la soudure. En première approche, réglez P1 pour obtenir +4,00 V sur le V+. Pour régler l'horloge, commencez par retirer le cavalier JP1 (mode Run). Actionnez ensuite le bouton SET S2 qui appelle la routine *setclock*(.).

Les minutes s'affichent alors. Une action sur le bouton ADJUST S1 incrémentera de 1 puis ajoutera 1 toutes les demi-secondes si vous le maintenez enfoncé. De petits délais dans le code se chargent d'éliminer les rebonds et faire fonctionner le tout sans anicroches.

Une fois les minutes réglées, appuyez à nouveau sur SET pour passer respectivement aux heures, jour, mois et enfin à l'année. À la fin d'un cycle de réglage, les nouvelles valeurs sont envoyées à la RTC et l'exécution de la boucle principale reprend.

(120740 – version française : Kevin PETIT)

## Liens

- [1] Vu-mètre à tubes Nixie, Elektor nov. 2012, [www.elektor.fr/110744](http://www.elektor.fr/110744)
- [2] [www.elektor.fr/120740](http://www.elektor.fr/120740)
- [3] [http://playground.arduino.cc/Code/BitMath#bitwise\\_and](http://playground.arduino.cc/Code/BitMath#bitwise_and)
- [4] [www.elektor-projects.com/project/numitron-arduino-clock-and-thermometer-120740.12460.html](http://www.elektor-projects.com/project/numitron-arduino-clock-and-thermometer-120740.12460.html)

Professional Quality  
Trusted Service  
Secure Ordering



**Elektor PCB Service at a glance:**

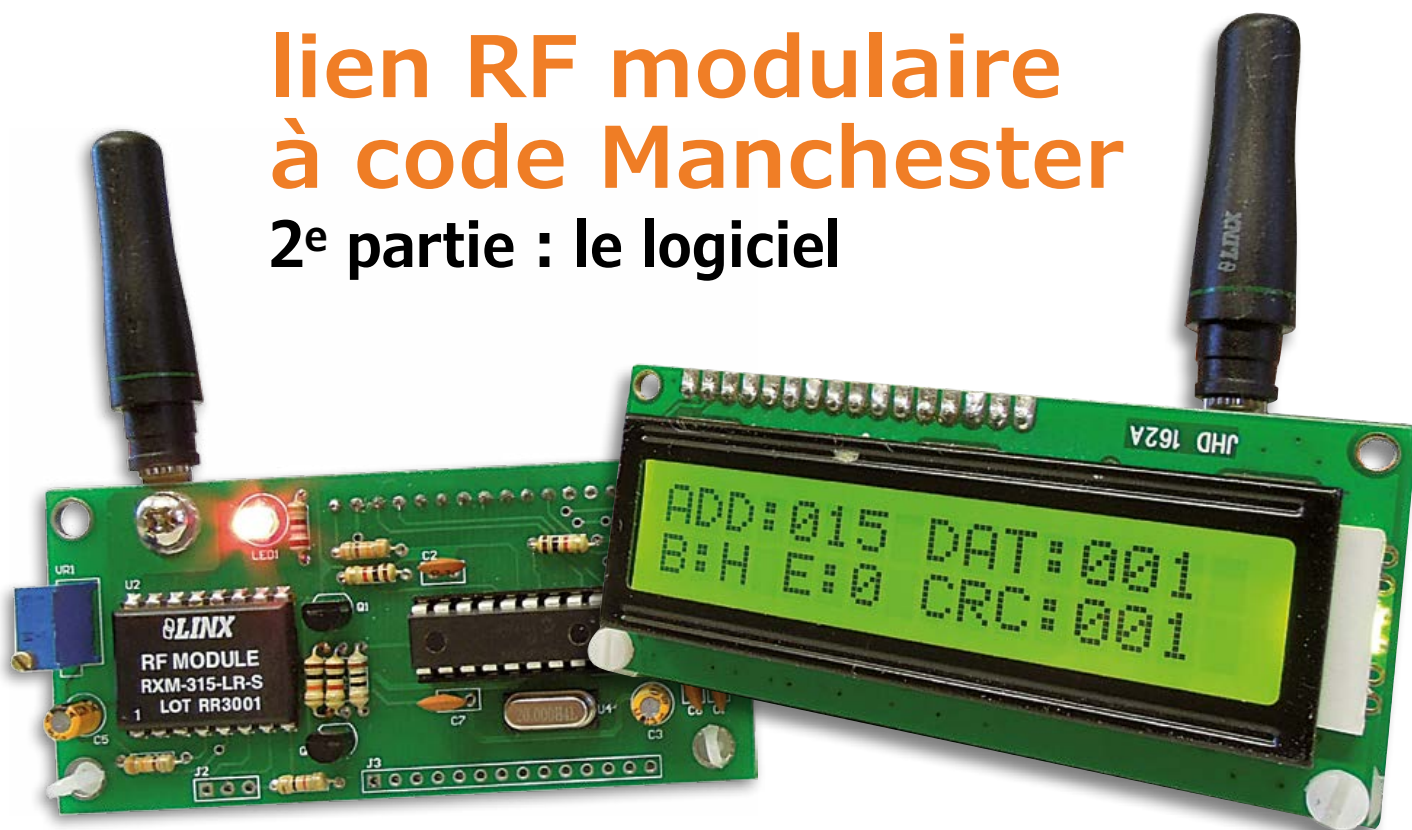
- ➔ 4 Targeted pooling services and 1 non-pooling service
- ➔ Free online PCB data verification service
- ➔ Online price calculator available
- ➔ No minimum order value
- ➔ No film charges or start-up charges

Delivery  
from 2  
working  
days



# lien RF modulaire à code Manchester

## 2<sup>e</sup> partie : le logiciel



Le mois dernier nous avons parlé du matériel. Une carte bien conçue et habilement routée est indispensable à la bonne transmission des signaux RF. Cette fois-ci il sera question du logiciel (ou micrologiciel si vous préférez) qui joue un rôle fondamental dans la fiabilité des transmissions portées par ces signaux.

**Marcelo Maggi**  
(États-Unis)

Les modules RF, je vous l'avais dit le mois dernier, ne comportent aucune intelligence et si nous voulons que notre système radio en ait, ça sera au logiciel du microcontrôleur de s'en charger. Ce que j'appelle intelligence ici va du choix du protocole de codage et de la vitesse de transmission (*bit rate*) à l'organisation des données en passant par les mécanismes de détection/correction d'erreurs.

Le logiciel a été conçu de manière modulaire et ses fonctions de base se trouvent dans des routines que vous n'aurez pas besoin de modifier. Vous aurez toute latitude pour incorporer vos routines dans le programme principal. Un appel à la routine d'émission se chargera de l'envoi de vos données. Côté récepteur, les routines se chargent également des tâches compliquées et ne renverront que les données utiles au programme principal.

### Le code Manchester

Le code Manchester, même s'il est basique, a prouvé qu'il était très fiable pour envoyer des données sur une liaison radio. Dans ce projet, les vitesses allant jusqu'au maximum de 5000 b/s donnent une bonne stabilité de la transmission. Polyvalence oblige, une vitesse plus basse est aussi disponible.

Les données à transmettre sont organisées dans une structure appelée *trame*. Un octet de correction d'erreur y est ajouté. Le code Manchester est très simple : chaque bit est représenté par une transition au lieu d'un niveau logique (**fig. 1**), indépendamment de la convention choisie. Nous utiliserons la convention de l'IEEE 802.3 : un 1 logique sera représenté par un front montant, un 0 par un front descendant. Les avantages de ce type de codage sont plutôt évidents :



- l'horloge du signal est présente dans chaque transition, peu importe les données, et est facile à extraire pour le récepteur
- le niveau moyen du signal est constant, autour de 50 %
- avec la modulation tout ou rien (OOK, *On-Off Keying*) de notre liaison RF, la puissance moyenne d'émission est réduite. C'est utile non seulement pour économiser l'énergie dans les applications portables, mais également pour rester conforme aux limites de puissance imposées par les différentes réglementations locales sur les bandes ISM tout en pouvant utiliser une puissance de pointe supérieure afin d'augmenter la portée.

Les modules Linx sont spécifiés pour 10000 b/s au maximum. Même si cette vitesse est atteignable, notre liaison se limitera à 5000 b/s à cause du codage Manchester qui, parce qu'il utilise deux niveaux logiques pour chaque bit, double le nombre de bits à transmettre. Cela vaut cependant le coup de sacrifier la vitesse pour la simplicité et la stabilité. J'ai été en mesure d'obtenir des transmissions fiables à 5000 b/s sur des distances supérieures à 180 m.

Il est possible d'utiliser une vitesse inférieure (2,500 b/s) pour les environnements très bruyants ou pour augmenter la portée. Linx prétend que l'on peut presque atteindre 1 km de portée avec le bon matériel.

## Format des données

Les données à transmettre sont composées de trois éléments d'un octet chacun : *adresse*, *donnée* et *CRC*. L'octet *adresse* permet d'indiquer quel est le récepteur destinataire de la transmission. Le codage sur un octet permet 256 adresses différentes. Dans la plupart des cas, il s'agit d'un gaspillage de bits et il serait plus efficace d'utiliser l'octet d'adresse pour coder l'adresse et une fonction :

- le quartet de poids fort (4 bits) pour sélectionner le récepteur (16 adresses)
- le quartet de poids faible pour indiquer au récepteur la fonction choisie (16 options).

Cela permet par exemple d'indiquer au récepteur n° 5 qu'il doit exécuter la commande n° 9 sur les données, par exemple pour mettre sous tension et positionner un servomoteur en fonction de la valeur du champ *donnée*.

Dans notre code d'exemple, le champ *adresse* ne sera utilisé que pour sélectionner le récep-

## Caractéristiques

- un émetteur, plusieurs récepteurs possibles
- portée jusqu'à 180 m
- débit sélectionnable par logiciel entre 2500 b/s et 5000 b/s
- correction d'erreurs facile à rajouter
- fichiers CAO du circuit imprimé et logiciels d'exemple disponibles gratuitement [2]

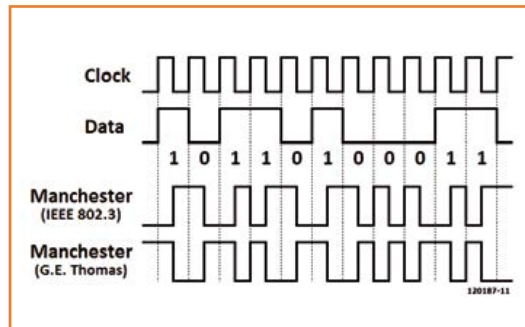


Figure 1.  
Les deux conventions du code Manchester.

teur ; il contient la valeur 15 à titre indicatif. Côté récepteur, lorsque la valeur 15 est lue dans le champ *adresse*, l'octet *donnée* sera considéré valide et traité. En binaire, 15 est représenté 00001111 ; facile à repérer avec un oscilloscope. Après codage Manchester, la séquence devient 1010101001010101.

L'octet *donnée* contient l'information utile et pourra être une valeur fixe, codant l'ordre d'exécuter une tâche définie par exemple, ou variable comme la valeur issue d'un convertisseur A/N. Le code d'exemple envoie un octet de données différent en fonction de l'état de la broche B3 (RB3/CCP1, br. 9) : si elle se trouve à l'état bas, c'est la valeur fixe 1 qui est envoyée, sinon c'est un 0 ou un 1 suivant l'état d'un temporisateur interne. Côté récepteur, la donnée est utilisée pour allumer (1) ou éteindre (0) D1. C'est un moyen simple de tester la liaison et montrer comment utiliser l'octet *donnée*. Ne vous fiez pas à l'apparente simplicité du code. En utilisant astucieusement l'octet *adresse* il est facile d'envoyer plusieurs octets de données à la fois. Le quartet le moins significatif peut être utilisé pour indiquer l'octet en cours d'envoi, permettant au récepteur de recomposer le message. Envoyer plusieurs octets de données est bien entendu plus efficace que d'envoyer un seul octet. Cet exemple se veut simple, mais moyennant quelques changements du code, il serait possible d'envoyer un nombre quelconque d'octets par trame.

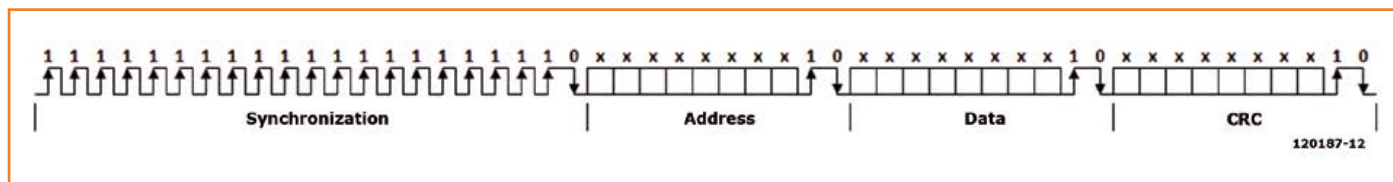


Figure 2.  
Anatomie d'une trame  
complète.

L'octet *CRC* ou *Cyclic Redundancy Check*, est utile pour valider le ou les octet(s) de données reçu(s) précédemment et peut aider à retrouver un bit manquant. Expliquer les différents algorithmes de détection et correction d'erreurs sortirait du cadre de cet article, mais comme tout protocole de transmission se doit d'inclure un tel champ, je l'ai ajouté. Dans le code d'exemple *CRC* est une copie de *donnée*. Si, côté récepteur, *CRC* n'est pas égal à *donnée*, vous savez que la trame n'est pas correcte. Par contre vous ne savez pas lequel des champs est erroné.

### Composition d'une trame

Nos trois champs doivent maintenant être rassemblés pour composer des *trames*. Toutes les liaisons radio sont sensibles au bruit et le récepteur doit donc être capable de déterminer où une trame commence et si elle est valide. C'est pourquoi je fais précéder chacune des trames par une séquence de synchronisation fixe qui ne doit pas pouvoir être émulée par une combinaison des trois champs précédemment décrits.

J'utilise la séquence 11111111111111111110 et, afin d'empêcher qu'elle soit émulée par une combinaison des trois champs utiles de la trame, lui adjoints une séquence d'espacement. Il s'agit d'une courte suite de bits, juste 10, ajoutée entre les trois champs utiles de la trame. La confusion qui se produirait si *adresse* et *donnée* valaient 11111111 et *CRC* 11110XXX ne sera plus un souci : 11111111 10 11111111 10 11110XXX 10. Nous sommes maintenant sûrs que le récepteur ne pourra pas confondre la synchronisation avec autre chose. Vous trouverez sur la **figure 2** une trame complète codée en Manchester.

### Débit de données

La vitesse de transmission est sélectionnée par l'émetteur. Elle est prédéfinie dans le micrologiciel, mais peut aussi être sélectionnée par l'entrée B2 (RB2/TX/CK, broche 8) du microcontrôleur, accessible sur la broche 11 de K2. Un niveau bas correspond à 2500 b/s, un niveau haut à 5000 b/s. Le récepteur détecte automatiquement la vitesse. Étant donné que la détection se fait

par des mesures de durée pour lesquelles l'oscillateur local sert de référence, il est très important que celui-ci soit le résonateur à quartz à 20 MHz décrit ici.

Pour que la liaison fonctionne, il faut que l'émetteur transmette régulièrement ses trames. Une trame isolée peut ne pas être reçue correctement, tout comme des trames espacées de plus de 10 à 15 ms. Pour qu'il soit capable de recevoir et démoduler correctement un signal, le récepteur doit être actif et son gain doit être réglé correctement. Après plus de 10 ms d'inactivité, le récepteur peut ne pas être prêt.

La solution simple est d'envoyer continuellement les trames en les espaçant de moins de 10 ms. Cela fonctionne, mais au prix de deux inconvénients majeurs :

- il n'est pas possible que le microcontrôleur soit occupé plus de 10 ms
- certaines réglementations peuvent interdire l'émission quasi continue sur une même fréquence de la bande ISM

La solution est simple : utiliser un temporisateur dans le programme principal pour émettre deux ou trois trames espacées de moins de 10 ms d'un coup. L'émetteur pourra ensuite se tourner les pouces pour quelques secondes.

Remarque : l'exemple téléchargeable sur [1] a pour unique but de montrer que la liaison fonctionne et comment l'utiliser. Le code du programme principal ne devrait pas être utilisé tel quel. Il est de **votre** responsabilité de vérifier que le comportement de votre montage est en accord avec la réglementation des bandes ISM en vigueur dans votre pays.

### Logiciel de l'émetteur

Que ce soit pour l'émetteur ou le récepteur, les routines de base ont été placées dans un fichier à part. Ces routines contiennent les parties les plus complexes du code que vous n'aurez pas à changer pour votre application. Un programme principal qui ne fait que quelques lignes est fourni pour tester la liaison. C'est là que vous devrez placer votre code.

## Routines de réception

Toute la complexité de cette liaison radio est masquée par ces routines de réception qui se chargent de recevoir et décoder les trames. Leur code comporte des commentaires pour expliquer chacune des sections. C'est la fonction de réception *mc\_rx*, qui fait le lien entre le programme principal et le reste des routines de bas niveau.

Tout d'abord, la détection du débit des données s'opère : *baud\_detect* sert à compter la durée de deux demi-bits. Cette détection est déclenchée par un front montant du signal reçu. Le premier bit d'une trame est un 1, un front montant en Manchester, et la première moitié du premier bit sera donc perdue au moment du premier déclenchement. C'est pourquoi *baud\_detect* mesure la deuxième moitié d'un bit et la première moitié du suivant.

Est-il suffisant de mesurer des moitiés et compter sur la symétrie ? Oui et non. Si l'envoi des données est continu, le récepteur sera actif en permanence et les niveaux DC internes seront stables. Et puis les bits sont plutôt symétriques. À l'inverse, si les trames sont espacées, disons de 10 ms, le récepteur ne sera pas forcément complètement prêt à l'arrivée d'une nouvelle trame et les premiers bits pas forcément symétriques (**fig. 3**). La liaison fonctionne encore, mais si l'on ne mesure pas un bit complet afin de déterminer le débit, il est tout à fait possible de mal estimer ce dernier et être donc incapable de recevoir les bits suivants. Dans notre exemple, les trames sont espacées de 1 ms et il n'y a donc pas de problèmes. Cela ne m'a pas empêché de prendre en compte ces cas extrêmes lorsque j'ai conçu la fonction *baud\_detect*. Après la première moitié du second bit, *timer0* contiendra la durée d'un bit, en quelque sorte.

Quelques calculs simples :

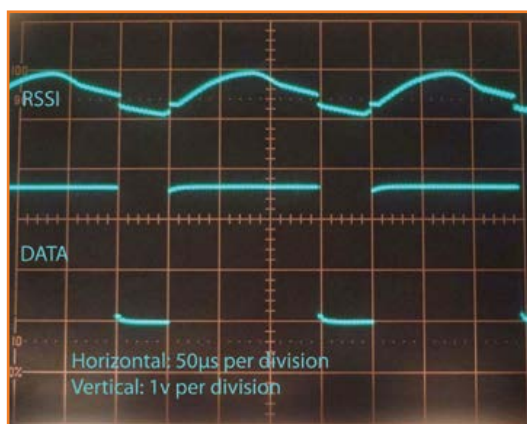
- un oscillateur à 20 MHz implique une horloge d'instruction à 5 MHz
- la durée d'un cycle est donc  $1/5000000 = 0,2 \mu\text{s}$
- *Timer0* augmente toutes les  $0,2 \mu\text{s} \times 16 = 3,2 \mu\text{s}$

- à 5000 b/s, un bit dure  $200 \mu\text{s}$  ; après un bit complet, on a donc  $\text{timer0} = 200/3,2 = 62,5$ . Comme *timer0* est un entier codé sur 8 bits il vaudra donc 62 ou 63, ce n'est pas vraiment important. La nature des signaux radio fait qu'ils seront impactés par un phénomène appelé gigue : une fluctuation des fronts d'un signal qui raccourcit ou étire les bits. Dire que *timer0* varie autour de 62,5 est suffisant et le programme s'accommodera des variations lors du calcul de la durée d'un bit en acceptant les valeurs allant de 55 à 70. Si *timer0* se trouve dans cette plage, la durée d'un bit sera choisie à  $200 \mu\text{s}$ , soit  $100 \mu\text{s}$  pour un demi-bit. C'est précisément ce qui est stocké dans la variable *semiperiod* renvoyée par la fonction. Pour 2500 b/s *semiperiod* vaut  $200 \mu\text{s}$  et la plage de détection de *timer0* va de 118 à 133.

Maintenant que nous connaissons leur débit, détecter les bits est facile : lire l'état du signal, attendre qu'il change, le lire à nouveau et comparer. Pour chaque bit le signal changera au milieu de la durée d'un bit. Avant de passer au bit suivant le programme attend pendant *semiandjitter* pour être sûr de lire le bit suivant et pas la fin du bit en cours. Si *status 1* est supérieur à *status 2* (front descendant, *in1* et *in2* dans le code), le bit reçu est un zéro, sinon c'est un 1. Ce bit est ensuite ajouté à la variable à 32 bits *three\_byte\_rx*, qui est décalée vers la gauche pour faire de la place pour le bit d'après. Si l'on élimine soigneusement les séquences d'espacement, *three\_byte\_rx* contiendra *adresse*, *donnée* et *CRC* et les 8 derniers bits seront inoccupés. S'il n'y a pas d'erreur en cours de traitement, *three\_byte\_rx* est copié dans *ad\_da\_cr\_rx* puis déplacé dans *frame\_rx* dans la fonction *mc\_rx*. Son contenu est décalé de 8 positions vers la gauche ( $\times 256$ ) puis *semiperiod* est ajoutée. *Frame\_rx* est maintenant prête à être renvoyée au programme principal. En cas d'erreur, la routine de réception renvoie zéro pour prévenir le programme principal. Une dernière remarque : faites attention à ne pas utiliser accidentellement dans le programme principal les variables utilisées par les routines de réception et d'émission.



Figure 3.  
Capture à l'oscilloscope  
du signal reçu lorsque les  
trames sont espacées de  
10 ms.



Le logiciel est écrit en C et a été compilé avec CCS. Les commentaires inclus dans le code apportent des informations supplémentaires. Examinons le code de l'émetteur, réparti sur trois fichiers :

- Manchester\_Link\_TX.c : le programme principal
- Manchester\_Link\_TX.h : initialisation du PIC
- MAN\_TX.c : les routines d'émission

Le programme principal est plutôt simple. Après les réglages initiaux le programme rentre dans une boucle infinie dans laquelle la broche B2 est lue pour sélectionner le débit des données, *adresse* est fixé à 15, la broche B3 est lue pour déterminer la valeur de *donnée* et *CRC* prend la même valeur que *donnée*. Une fois les quatre variables connues, la fonction d'émission est appelée avec `mc_tx(baud, address, data, crc);`. C'est comme ça que le programme principal passe les informations aux routines d'émission. Peu importe leur complexité, il ne faut qu'une ligne de code pour émettre des données. La dernière ligne est une temporisation de 1 ms avant de repartir pour un tour.

Ce sont les routines d'émission qui constituent le cœur de ce projet. Vous n'aurez pas à les modifier pour votre application. Vous devrez les inclure tout au début du programme principal, après la configuration du PIC. Elles seront capables de créer une trame codée en Manchester juste avec les quatre variables reçues du programme principal. Le débit de donnée est utilisé pour sélectionner la demi-durée d'un bit (*semiperiod*), valeur essentielle pour coder en Manchester. La séquence de synchronisation est construite en appelant 20 fois la fonction *one* puis une fois la fonction *zero*.

Chaque octet est analysé bit par bit, en commençant par le bit de poids fort. Les fonctions *one* et *zero* sont appelées pour envoyer les données et la séquence d'espacement après chaque octet. Et voilà ! Une trame complète est envoyée sur la broche B0 à la vitesse choisie. Il ne faudra que peu modifier ces routines si vous souhaitez envoyer plusieurs octets par trame. Les informations contenues dans les trames (*adresse*, *donnée* et *CRC*) seront affichées sur un afficheur à cristaux liquides.

### Logiciel du récepteur

Tout comme leur contrepartie matérielle, les routines de réception sont un peu plus complexes que celle utilisées pour l'émission. Pas d'inquiétudes cependant : vous n'aurez pas à les modifier. Le code est ici aussi réparti sur trois fichiers :

- Manchester\_Link\_RX.c : programme principal
- Manchester\_Link\_RX.h : configuration du PIC
- MAN\_RX.c : routines de réception

Encore une fois le programme principal est plutôt simple. Après les fonctions de gestion du LCD, vous trouverez les définitions standard. Vous remarquerez la routine d'interruption définie dans `void detection_isr();`. C'est cette routine qui sera appelée à chaque front montant du signal sur la broche B0 (RB0/INT, broche 6).

Une autre ligne importante qui ne doit pas être oubliée est celle chargée de configurer *timer0* dans la fonction *main* : `setup_timer_0(RTCC_INTERNAL|RTCC...` Elle sélectionne le mode temporisateur à 8 bits de *timer0* qui sera incrémenté tous les 16 coups d'horloge. C'est ce temporisateur interne est utilisé pour déterminer le débit des données du signal entrant en mesurant la durée d'un bit.

Comme nous utilisons les interruptions, il ne faut pas oublier de les activer avec `enable_interrupts(GLOBAL);`. Le programme rentre ensuite dans sa boucle principale qui surveille les données entrantes. C'est dans cette même boucle qu'il faudra ajouter votre code qui ne devra pas désactiver les interruptions. Certaines actions, comme afficher des données sur un LCD, peuvent désactiver les interruptions et devront donc être programmées dans l'ISR, après que l'information ait été reçue et validée. N'oubliez pas que le récepteur attend l'arrivée d'un signal\*.

À chaque front montant sur le signal en provenance du module radio, la routine d'interruption

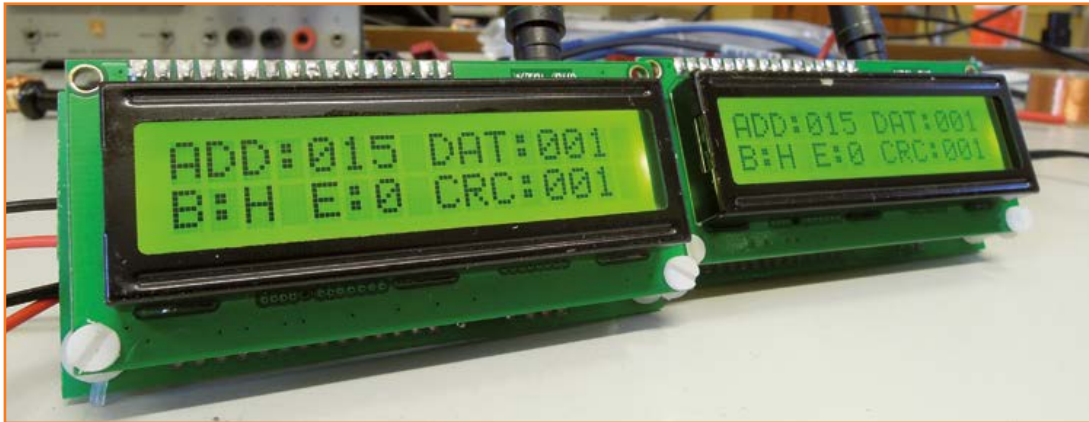


Figure 4.  
Premiers prototypes avec  
LCD.

est appelée. Celle-ci désactive les interruptions puis appelle la fonction de réception d'une trame `frame=mc_rx(frame);`. Cette fonction qui fait partie des routines de réception retourne un entier codé sur 32 bits. Un signal arrive, la fonction `mc_rx` est appelée et retourne toutes les informations reçues : impossible de faire plus simple.

Comme la variable `frame` fait 32 bits, elle contient 4 octets : *adresse*, *donnée*, *CRC* et *duration* de la trame reçue. Les octets sont extraits de la trame comme suit : la variable `frame` est copiée vers une variable à 8 bits qui ne conservera que les huit bits de poids le plus faible : `halfbyte=frame;`. *Halfbyte* contient maintenant le quatrième octet de `frame`. En décalant le contenu de `frame` de huit positions vers la droite, on peut extraire l'octet suivant en répétant la copie.

*Halfbyte* contient la durée d'un demi-bit de données et permet de calculer leur débit. Le reste des instructions montre une utilisation rudimentaire des informations reçues. Si la trame reçue comporte une erreur, la fonction renverra 0 et *Halfbyte* vaudra 0. C'est utilisé pour positionner un drapeau de signalisation d'erreur.

Si *adresse* vaut 15 (comme pour l'émetteur) et qu'il n'y a pas d'erreur, le contenu de *donnée* est utilisé pour choisir l'état imposé à la LED. *CRC* n'est pas utilisé dans cet exemple.

Le LCD affiche en permanence les informations reçues (*adresse*, *donnée* et *CRC*) dès que l'interruption est activée. Si aucune donnée n'est reçue, le LCD affiche « Error is '1' » et « Bit status is 'L' ».

### Améliorations et applications

Ce n'est pas parce que la liaison radio présentée ici est pleinement fonctionnelle et s'est montrée fiable lors de mes différents tests qu'il n'est pas

possible de l'améliorer ou la personnaliser. Vous pourriez par exemple ajouter l'envoi de plusieurs octets de données à la fois. Les octets supplémentaires pourraient facilement être ajoutés à une trame, il faut juste savoir combien en attendre pour ne pas en oublier. Bien évidemment une unique variable de 32 bits ne pourrait plus dans ce cas contenir tous les bits et il faudrait utiliser une structure.

Les possibilités d'applications sont infinies et ne seront limitées que par votre imagination : ce projet n'est qu'une brique de base à utiliser dans un projet plus grand.

Vous verrez en **figure 4** les prototypes assemblés au labo Elektor ; le LCD affiche *adresse* = 15, *donnée* et *CRC* = 1, Bit rate (B) est au max (H = 5,000 b/s) et Error (E) = 0 (pas d'erreur).

Cette application simple permet de montrer le potentiel de ces modules ; c'est également un outil précieux pour dépanner le code quand celui-ci ne se comporte pas comme prévu.

Nous voici arrivés au terme de ce projet. Si vous avez des questions, des inquiétudes ou des commentaires à faire sur le matériel ou le logiciel, n'hésitez pas à nous rejoindre sur [elektor.LABS](http://elektor.LABS) [2] ou visitez la page de l'auteur [3] ; les fichiers de CAO du circuit imprimé et les routines logicielles sont téléchargeables gratuitement.

(120187 – version française : Kévin PETIT)

### Liens

- [1] [www.elektor.fr/120187](http://www.elektor.fr/120187)
- [2] [www.elektor-projects.com/120049](http://www.elektor-projects.com/120049)
- [3] [www.magusporta.com](http://www.magusporta.com)

\* NDLR: Piloter un afficheur LCD depuis une routine d'interruption déclenchée par un signal à 10 kHz est un exemple à ne pas suivre !

# elektor cardi♥scope Android

## 3<sup>e</sup> partie

## réalisation, réglages et mode d'emploi

Voici le dernier volet de cette réalisation qui fera date. Au moment de sa publication, il n'existe dans le commerce rien de comparable, ni par les performances, ni par le prix. Méfiez-vous des imitations !

**Marcel Cremmel**

marcel.cremmel.llc@free.fr  
(en coopération avec  
Raymond Vermeulen)

### Réalisation : quelques remarques

Pour la réalisation de l'elektorcardi♥scope (que nous appelons aussi l'interface ECG), le travail a été bien mâché, puisque le module est disponible assemblé, testé et prêt à l'emploi [8]. Tous les composants du schéma publié dans le premier article [9] tiennent sur une carte de dimensions modestes (10 x 6 cm) (**fig. 19**) mais étudiée pour tenir (sans vis !) dans un modèle de boîtier courant avec compartiment pour piles (cf liste des composants). La seule chose à faire pour la carte d'interface elle-même est d'y brancher les deux fils d'alimentation.

Si d'aventure vous assemblez le circuit vous-même au lieu d'acheter le module prêt à l'emploi, ne le faites que si vous jouissez d'une solide expérience en la matière. Le régulateur IC12 doit être un MCP1640BT (moins bruyant). Il est possible d'utiliser un autre module *Bluetooth* que le modèle proposé (fig. 4c du 1<sup>er</sup> article) à condition qu'il utilise le protocole SPP. L'avantage de celui-ci est d'offrir une possibilité de remise à zéro (S3). En n'implantant pas ou en dessoudant L3, la self de déparasitage à perle de ferrite, il est facile de tester séparément différentes parties du circuit. La barrette sous le PCB est optionnelle. Elle permet de reprogrammer le  $\mu C$  *in circuit* (ICSP) avec un outil compatible (PICKit2 ou PICKit3 de *Microchip* p. ex.).

La fonction des trois boutons et des deux LED apparaît sur le dessin de face avant (**fig. 20**) : S1 = arrêt (et r.à.z. du  $\mu C$ ), S2 = marche, S3 =



r.à.z. *Bluetooth*, D3 = transmission des données vers le terminal Android (TX) et D4 = statut du module *Bluetooth* (BT).

Il faudra encore confectionner vous-même un connecteur pour vos électrodes et peut-être vos électrodes elles-mêmes, nous y reviendrons. D'abord nous allons mettre en service notre interface ECG. Une fois alimenté par les deux piles AA, il suffit d'appuyer brièvement sur S2 pour que le module assemblé se manifeste en faisant clignoter D4 (2 Hz), ce qui indique que le module *Bluetooth* est identifiable.

### L'application Android

L'installation est classique : téléchargez l'application Android empaquetée (fichier ANDROECG.

Que soit remercié Aurélien Moulin, étudiant I2 à l'ESEO d'Angers et stagiaire au labo d'Elektor, pour sa participation active au test et au débogage des premiers modules assemblés proposés par elektorPCBservice.



## Liste des composants

### Résistances (0603 1%) :

R1, R13, R15, R18, R19, R20, R33, R34, R54 = 10 k 0,25 W  
 R2, R12 = 3k3  
 R3 = 523 k  
 R4 = 300 k  
 R6 = 150  $\Omega$   
 R7...R11, R23 = 1 k  
 R14, R53, R55, R56, R57, R59...R62 = 100 k  
 R5, R16, R17, R35 = 1 M  
 R21, R36, R37, R38 = 330 k  
 R22 = 100  $\Omega$   
 R24, R65 = 390 k  
 R25, R29 = 47 M 5%  
 R26, R30 = 10 M  
 R27, R31 = 2M2  
 R28, R32 = 470 k  
 R39, R40, R49...R52 = 47 k  
 R41, R42, R45, R46 = 28k7  
 R43, R44 = 1M4  
 R47, R48 = 45k3  
 R58 = 9k1  
 R63, R64 = 27  $\Omega$   
 P1, P2 = aj. 5 k 20% (Vishay TS53YJ502MR10)  
 P3 = aj. 2 k 20% (Vishay TS53YJ202MR10)

### Condensateurs (0603) :

C1, C13 = 33  $\mu$ F / 6V3, tant. (case A)  
 C2, C4, C7, C9, C12, C14 = 10  $\mu$ F / 6V3, X5R  
 C3, C5, C6, C8, C10, C15, C21, C22, C23, C24, C39, C40, C41, C42, C43, C44 = 100 nF / 25 V, X7R  
 C11, C16, C25, C26, C31...C38 = 1  $\mu$ F / 10 V, X5R  
 C17...C20\* = 1 nF / 50 V, X7R  
 C27, C29 = 470 pF / 50 V, NP0, 5%  
 C28, C30 = 47 nF / 25 V, X7R, 5%

\* La valeur de C18, C19 et C20 est 1 nF/50 V et non 100 nF/50 V comme dans le schéma (fig. 4).

### Inductances :

L1 = 4,7  $\mu$ H 20% 0,5 A (Wuerth 744032004)  
 L2...L9 = perle de ferrite, 30  $\Omega$ @100 MHz (Murata BLM18PG330SN1D)

### Semi-conducteurs :

D1, D2 = BAV99S  
 D3, D4 = LED rouge (PLCC-4)  
 T1 = NMOSFET PSMN6R5-25YLC  
 IC1 = PIC24FJ32GA002-I/SS **EPS 120107-41**  
 IC2 = TPS60403DBVT  
 IC3...IC7 = TLC2252AIDRG4  
 IC8 = LMC6482AIMX/NOPB  
 IC9 = DG4053 AEQ-T1-E3  
 IC10, IC11 = CD74HC4052PW  
 IC12 = MCP1640BT-I/CHY  
 IC13 = LTC1981ES5#TRMPBF

### Divers :

K1 = embase coudée 1x5 2,54 mm  
 K2 = barrette 1x6 2,54 mm  
 K3 = embase coudée 1x2 2,54 mm  
 MOD1 = module Bluetooth RN-42 *Roving Networks*  
 S1, S2, S3 = poussoir (Omron SPNO B3FS-1052) avec cabochon (Omron B32-2010)  
 boîtier Pactec PPL-2AA  
 circuit imprimé 120107-1 ou  
 module assemblé prêt à l'emploi 120107-91

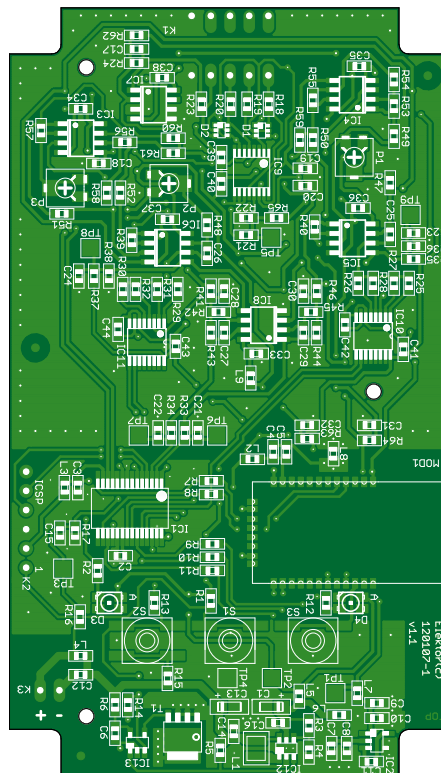


Figure 19.

L'interface ECG tient dans le creux de la main. Les connaisseurs apprécieront dans le routage la bonne séparation, essentielle ici, des parties analogique et numérique du circuit.

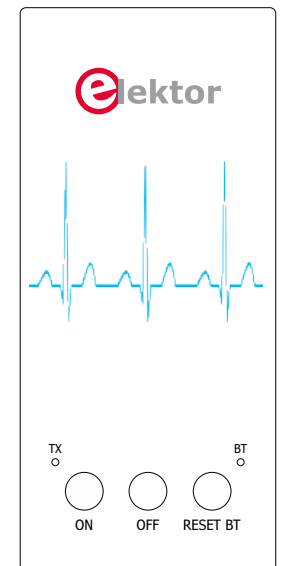


Figure 20.

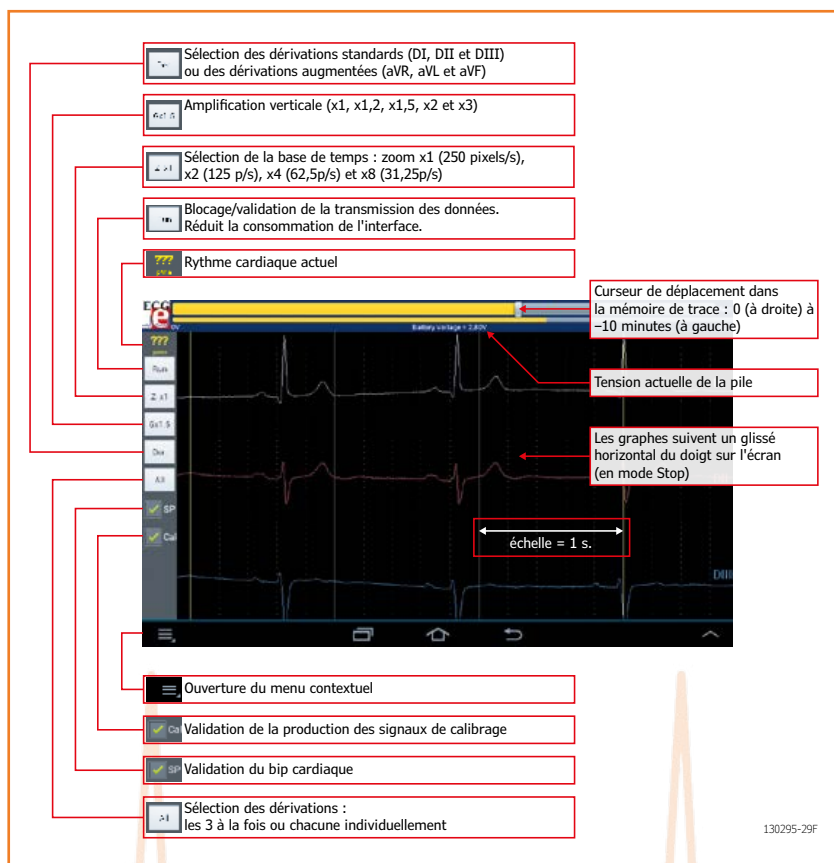
Dessin de la face avant avec 3 boutons et 2 LED (PDF téléchargeable).

La numérotation des figures et des liens commence là où finissait celle des deux articles précédents.

Figure 21.  
Menu de sélection du lien Bluetooth à établir entre l'interface ECG et un terminal Android du voisinage.



Figure 22.  
Vue d'ensemble des fonctions de l'interface tactile.



apk) [10], copiez-la dans la racine du terminal Android et sélectionnez-la avec l'explorateur de fichiers. Elle s'installera automatiquement (si vous avez au préalable autorisé, dans le menu Sécurité de votre terminal Android, l'installation d'applications *non-Market*).

Placez alors un raccourci à votre convenance. Si au lancement de l'application l'interface *Bluetooth* du terminal n'est pas activée, l'application le signale. Il faut bien entendu l'activer.

À ce stade, la liaison radio BT entre l'interface ECG et le terminal Android n'est pas encore établie. Pour ce faire, il faut ouvrir le menu et demander les périphériques BT enregistrés (*Paired BT Devices*) (**fig. 21**). La liaison devrait s'établir aussitôt après le choix du module *Bluetooth* de l'interface ECG. Dans ce cas, le graphe commence à défiler et la LED BT (*State*) sur l'interface reste allumée. À la première connexion, il faudra saisir le code PIN (ici : 1234) du module BT. Le périphérique BT est alors enregistré dans le terminal Android et le code PIN ne sera plus demandé. Le  $\mu C$  de l'interface n'est pour rien dans ce protocole. Le code MAC imprimé sur le module BT et affiché sur le terminal Android peut faciliter l'identification de l'interface ECG si la liste des appareils BT est longue.

Si aucune liaison BT n'est établie dans les 5 min, l'interface se mettra automatiquement hors tension. Vous aurez tôt fait de vous familiariser avec les menus et les fonctions du logiciel de l'elektorcardi♥scope sur le terminal Android grâce au « mode d'emploi » intuitif de la figure 22. Il existe également une vidéo de démonstration [11].

Il reste maintenant à procéder aux réglages de l'interface qui se résument à deux opérations simples : le taux de réjection du mode commun (TRMC) et l'équilibrage des gains.

### Réglage du TRMC de chaque voie

Le premier réglage consiste à optimiser le TRMC de chaque amplificateur différentiel à l'aide d'un générateur de fonction. Commencez par réaliser l'accessoire de calibrage à gauche de la **figure 23** (BNC + barrette sans résistance). Enfichez-le dans l'embase K1 en respectant l'orientation : la broche 1 est à droite quand on tient l'interface les boutons vers soi. Les entrées RA, LA et LL sont reliées ensemble pour y injecter un signal en mode commun par les broches 1, 2 et 3 de l'accessoire de calibrage ; sa broche 4 est reliée au corps de la fiche BNC et donc à la masse du

géné BF tandis que la broche 5 (masse de l'interface ECG) reste en l'air.

Réglez le générateur pour obtenir un sinus alternatif 50 Hz d'amplitude 1 V. Mettez l'interface ECG sous tension et lancez l'application ANDROECG. Établissez la liaison *Bluetooth* et observez les dérivations DI et DII avec l'amplification maximale. Ajustez alors P1 et P2 pour réduire autant que possible les amplitudes crête à crête. Celles-ci doivent être à peine visibles avec l'amplification x10 (**fig. 24**). À défaut de géné BF, vous pouvez opérer comme suit :

- touchez le GND de l'interface avec un doigt d'une main ;
- avec un doigt de l'autre main, touchez le point commun des trois électrodes RA, LA et LL : on injecte ainsi un signal de mode commun issu du secteur 50 Hz capté par votre corps ;
- observez les signaux DI et DII sur le terminal Android ;
- avec votre troisième main (vous aurez peut-être besoin d'un cobaye !), ajustez P1 et P2 pour obtenir des courbes aussi plates que possible.

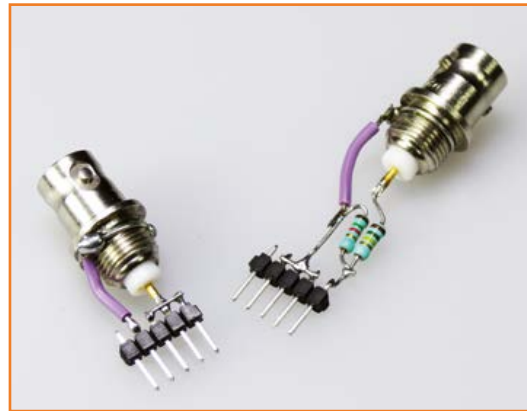


Figure 23.  
Vous n'aurez à allumer votre fer à souder que pour confectionner ces deux accessoires de calibration.

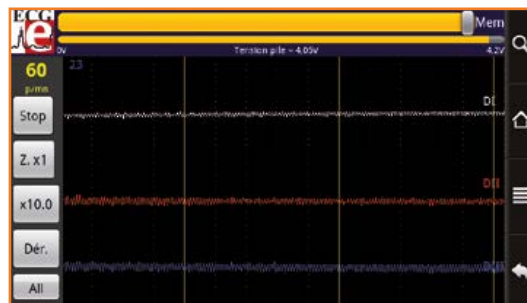


Figure 24.  
Il faut réduire autant que possible l'amplitude des résidus de 50 Hz.

### Équilibrage des gains

Le gain total des amplificateurs de chaque voie doit être identique, car les signaux DI et DII sont utilisés pour calculer les autres dérivations (encadré **Les dérivations** [9]). Un générateur de signal est recommandé pour ce réglage, avec un atténuateur car le signal à injecter doit être très faible :  $1,4 \text{ mV}_{cc}$  (à droite sur la **fig. 23**, BNC + barrette + résistances). Cet atténuateur, constitué de deux résistances de 1k5 et 150 k connectées comme l'indique la **fig. 25**, doit être inséré entre le géné BF et l'interface ECG dans l'embase K1 : la broche 1 est à droite quand on tient l'interface les boutons vers soi.

Réglez le générateur pour qu'il produise une sinusoïde de 1 Hz et  $140 \text{ mV}_{cc}$ . Si ce n'est déjà le cas, connectez votre terminal Android à l'interface ECG et observez les dérivations DI, DII et DIII avec une amplification x2. Vous devriez observer sur DI et DII des sinusoïdes de 1 Hz comme sur le relevé 1 (**fig. 26**). Ajustez alors P3 pour réduire au minimum le tracé de la dérivation DIII. En effet, le terminal Android calcule  $DIII = DII - DI$  : DIII doit être nulle quand  $DI = DII$ . On peut augmenter l'amplification pour obtenir plus de sensibilité et n'afficher que DIII. Les relevés 2 et 3 représentent DIII avant et après le réglage.

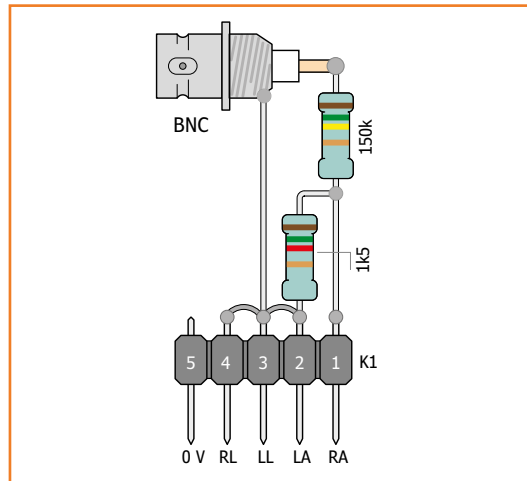


Figure 25.  
Cet accessoire de réglage assure une division par 100 du signal de calibration injecté.

### ATTENTION

La fabrication et l'utilisation d'appareillage médical sont régies par des lois nationales et internationales [14]. L'elektorcardi♥scope n'a reçu aucun agrément médical et n'est donc pas destiné à un usage professionnel.

Il doit être alimenté exclusivement par piles pour respecter la classe de protection III et ne peut servir qu'à des fins d'étude et d'expérimentation personnelle. La responsabilité de l'auteur de cet article ni celle de l'éditeur ne sauraient être engagées d'aucune manière pour les conséquences de l'utilisation de cette interface.



Pour l'équilibrage des gains, il faut injecter le niveau le plus fort possible pour augmenter la sensibilité du réglage, mais surtout ne pas saturer les amplificateurs, car le réglage devient alors impossible. Avant le réglage, assurez-vous de l'absence de saturation ou de déformation des sinusoïdes affichées sur le terminal (notamment DI) et le cas échéant, diminuer le niveau. En principe cela ne devrait pas se produire, car il y a un peu de marge en injectant 1,4 mV.

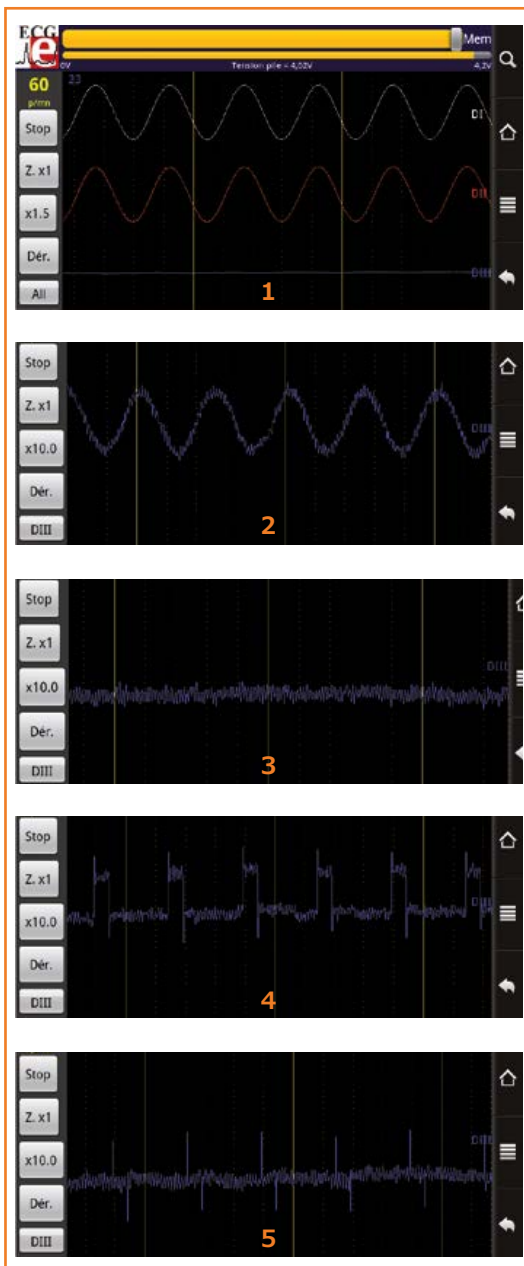


Figure 26. Ces relevés vous seront utiles pour l'équilibrage des amplis IC3 et IC4 (voies DI et DII) : il s'agit de réduire au minimum le tracé de la dérivation DIII.



Figure 27. Chaque minute, l'interface produit pendant 10 s un signal CAL de 2 Hz/1 mV, utilisable pour l'équilibrage des gains (cf. fig. 26).

À défaut de généré BF, attendez l'injection périodique à l'entrée de l'interface ECG du signal CAL de calibrage (fig. 27, voir aussi fig. 9 dans le 2<sup>e</sup> article) et ajustez P3 pour obtenir sur DIII un résidu de signal CAL aussi faible que possible. Les relevés 4 et 5 (fig. 26) illustrent les résultats avant et après le réglage ; notez qu'il est impossible d'éliminer les impulsions fines visibles sur le relevé 5.

L'elektorcardi♥scope est maintenant opérationnel et on peut, sur l'écran du terminal Android, agir sur les différents boutons tactiles (fig. 22) pour choisir les graphes, amplifier les amplitudes, changer la base de temps et se déplacer dans la mémoire de trace. Mais pour pouvoir observer vos électrocardiogrammes sur l'écran de votre terminal Android, il vous reste à vous équiper d'électrodes et à trouver un patient.

### Les électrodes

Les signaux électriques que nous capterons à l'aide des électrodes sont **infiniment plus faibles** que ceux dont nous sommes familiers. Un bon électrocardiogramme ne pourra donc être obtenu qu'avec de bonnes électrodes, bien placées et bien câblées. Pour mémoire, sur le connecteur des électrodes (fig. 28a) la masse est à gauche quand on tient l'interface ECG les boutons vers soi. Côté patient, prenez la bonne habitude de respecter le code de couleurs [9] :  
rouge = bras droit => RA (*right arm*)  
jaune = bras gauche => LA (*left arm*)  
vert = jambe gauche => LL (*left leg*)  
noir = jambe droite => RL (*right leg*)

Si vous ne souhaitez pas faire les frais de quatre électrodes de fabrication industrielle -- les pinces, bien pratiques, surtout avec les enfants, ne sont pas bon marché -- vous pouvez aisément en fabriquer vous-même. Toute électrode avec son câble de liaison est aussi une formidable antenne, or là vous en aurez quatre : l'usage de câble blindé s'impose donc pour limiter l'influence des signaux indésirables. Le blindage n'est relié que du côté de l'interface ; du côté de l'électrode, il faut l'isoler pour éviter tout contact avec la peau. Attention ! Si le câble audio convient électriquement, il est fragile mécaniquement. Des serre-câbles atténuent le risque de rupture (fig. 28b). Les fiches bananes de 4 mm permettent d'utiliser des accessoires de fabrication industrielle (fig. 29) tels les adaptateurs à fiche banane pour bouton-pression [12] ou des bananes à visser (RS ref. 641-8053).

S'il vous reste des pièces de monnaie d'avant l'euro, à base de nickel (1 franc, 1 florin ou 1 deutschmark p. ex.), vous pouvez aussi réaliser ces accessoires vous-même à moindres frais. La seule difficulté est de souder une douille de 4 mm sur la pièce (**fig. 30**) que quatre bracelets élastiques (ruban pour bretelles et bande velcro) maintiendront en place sur les poignets et les chevilles. On peut aussi utiliser des sections de chambres à air de moto ou de scooter. Placez les électrodes sur les poignets et les chevilles, si possible avec du gel conducteur qui améliore la qualité des ECG par une réduction notable du bruit et des tensions de contact.

### Sauvegarde et lecture des ECG

Ces opérations sont simples, aucune option n'est demandée : on sauvegarde ou on lit les 10 min d'échantillons des dérivations DI, DII et DIII. L'indice du dernier échantillon (l'actuel) dans la mémoire ECG circulaire est également sauvegardé pour retrouver la même apparence des tracés à la lecture (dernier échantillon représenté à droite de l'écran : voir fig. 17 dans le 2<sup>e</sup> article). Les opérations de sauvegarde/lecture sont proposées dans le menu. Une fenêtre vous propose le choix entre un fichier existant ou un nouveau fichier dont il suffit de saisir le nom (fig. 18 dans le 2<sup>e</sup> article [13]).

### Autres fonctions

La case à cocher «Cal» (fig. 22) permet de choisir l'injection périodique d'un signal de calibrage à la place des signaux ECG. Bien connu des cardiologues, le signal de calibrage, dont l'amplitude est d'exactement 1 mV (**fig. 27**), est utilisé comme référence de comparaison avec les ECG.

Astuce pour les impatientes : vous pouvez forcer l'injection immédiate du signal de calibrage en décochant puis en recochant le bouton «Cal». L'application Android calcule le rythme cardiaque avec un algorithme basé sur la dérivée du signal DI. Cet algorithme peut parfois être mis en défaut ainsi que son affichage sur l'écran et le *bip* sonore produit. Ce bruit peut devenir agaçant, on le coupe alors avec le bouton SP.

### Fonctions futures

Un tel projet est en perpétuelle évolution. Dans l'état actuel, il n'existe pas d'autre application d'affichage que la version Android. Cependant, Elektor se ferait un plaisir de diffuser les versions pour iPhone, LINUX, MAC ou PC que des lecteurs

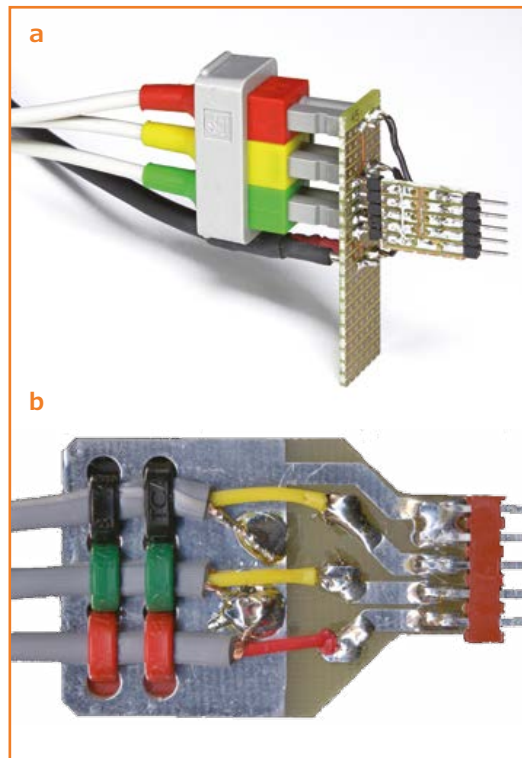


Figure 28.

Nous avons confectionné cet adaptateur pour un connecteur d'électrodes de fabrication industrielle (fig. 28a). Remarquez les serre-câble du robuste connecteur maison (28b) réalisé en 2006 pour la version Gameboy qui avait une électrode de moins.

motivés voudront bien nous proposer. En attendant, voici les fonctions de l'application Android en cours de développement :

- filtre numérique de réjection du secteur 50 Hz (ou 60 Hz)
- dépôt des relevés sur le *cloud* via une API



Figure 29.

Les accessoires de fabrication industrielle ne sont pas bon marché, mais ils présentent bien.



Figure 30.

On obtiendra des électrodes satisfaisantes à l'aide de pièces de monnaie d'un franc à condition de parvenir à y souder une douille pour fiche banane.

L'auteur, Marcel Cremmel, est professeur agrégé en Génie Électrique, option électronique, au lycée Louis Couffignal de Strasbourg, dans la section BTS électronique.  
<http://electronique.marcel.free.fr/>

(Application Programming Interface) de Google

Je travaille aussi sur une application *Windows* pour consulter les ECG enregistrés sur une carte SD et même les afficher en direct via une interface *Bluetooth*.

Dans un avenir plus lointain, je proposerai du code à copier sur votre site personnel permettant de recevoir et afficher les ECG relevés par votre terminal Android et transmis vers ce site. Votre médecin traitant ou votre cardiologue pourra, si vous le souhaitez, régulièrement consulter cette page pour suivre votre état de santé.

(130295)

Ci-contre une électrode dite poire précordiale (vendues généralement par jeu de six) couplée par une fiche banane à un robuste connecteur fait maison.



## Liens

[8] [www.elektor.fr/120107-91](http://www.elektor.fr/120107-91)

[9] 1<sup>ère</sup> partie :

elektor n°421-422 juil./août 2013, p.36  
[www.elektor.fr/120107](http://www.elektor.fr/120107)

[10] application ANDROECG.apk  
[www.elektor.fr/130295](http://www.elektor.fr/130295)

[11] Vidéo de démonstration Version FR 1  
[www.youtube.com/watch?v=wCGXK-kV7EJ4](http://www.youtube.com/watch?v=wCGXK-kV7EJ4)

[12] Électrodes & accessoires

[www.praxisdienst.fr/fr/home/](http://www.praxisdienst.fr/fr/home/)  
[www.cardiostore.fr](http://www.cardiostore.fr) (ne vend qu'aux professionnels)

[13] 2<sup>e</sup> partie :

elektor n° 423 sept. 2013, p. 36  
[www.elektor.fr/130227](http://www.elektor.fr/130227)

[14] *dura lex sed lex*

<http://homeusemedicaldevices.com/humd.html>

Coupleur à bouton pression et fiche banane (*banana to snapon adapters*)

<http://goo.gl/5WWYFu> ou

[www.praxisdienst.com/en/Diagnosis/Specialised+diagnosis/ECG+devices+and+accessories/oxid+oxid/Press+stud+adapter+for+ECG+red.html](http://www.praxisdienst.com/en/Diagnosis/Specialised+diagnosis/ECG+devices+and+accessories/oxid+oxid/Press+stud+adapter+for+ECG+red.html)

Pinces (*limb clamps*)

<http://goo.gl/bZIKXf> ou

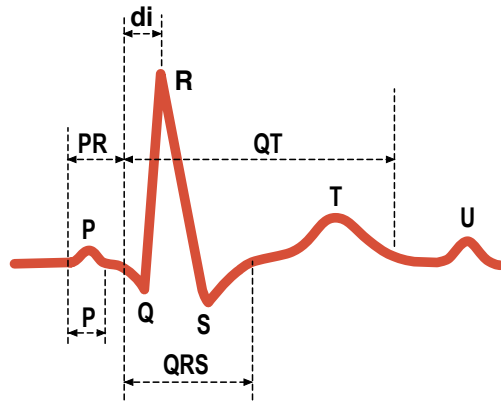
[www.praxisdienst.com/en/Diagnosis/Specialised+diagnosis/ECG+devices+and+accessories/oxid+oxid/Limbs+Clip+electrode+Adult+red.html](http://www.praxisdienst.com/en/Diagnosis/Specialised+diagnosis/ECG+devices+and+accessories/oxid+oxid/Limbs+Clip+electrode+Adult+red.html)



## L'électrocardiogramme normal

### Rythme et fréquence

La fréquence normale au repos est comprise entre 50 et 100 pulsations par minute. Inférieure à 50, c'est une bradycardie sinusale, supérieure à 100, c'est une tachycardie sinusale.



### Dépolarisation auriculaire : onde P

Sa durée normale est inférieure ou égale à 0,1 s.

Son amplitude normale est inférieure ou égale à 0,25 mV, soit  $\frac{1}{4}$  de l'amplitude du signal de calibrage CAL.

L'onde P est généralement maximale en DII, DIII et aVF. L'onde P est toujours positive en DI et DII et négative en aVR.

### Espace PR ou PQ

La durée normale de l'intervalle PR est comprise entre 0,12 et 0,20 s. Il se mesure du début de l'onde P au début du complexe QRS. Il correspond au temps de conduction de l'influx de l'oreillette aux ventricules. Il peut diminuer quand la fréquence cardiaque s'accélère à l'effort. Au-delà de 0,20 s, il témoigne d'un trouble de conduction auriculo-ventriculaire.

### Amplitude de QRS

Dans les dérivations frontales, l'amplitude est très variable. Une amplitude inférieure à 0,5 mV ( $\frac{1}{2}$  CAL) dans l'ensemble de ces dérivations fait parler de micro voltage.

### Durée du complexe QRS

Elle est en moyenne de 0,08 s ; elle doit rester inférieure à 0,12 s. Au-delà il s'agit le plus souvent d'un asynchronisme de dépolarisation des deux ventricules lié à un trouble de conduction intra ventriculaire.

### Repolarisation ventriculaire : segment ST – onde T – onde U

**Le segment ST** sépare le complexe QRS de l'onde T. Son origine est prise à la fin du QRS.

**L'onde T** est habituellement de faible amplitude, asymétrique avec une pente ascendante plus faible que la pente descendante, et de même sens que QRS. Elle est normalement positive en DI, DII, DIII et aVF. Une onde T diphasique ou négative en DIII doit être considérée comme physiologique.

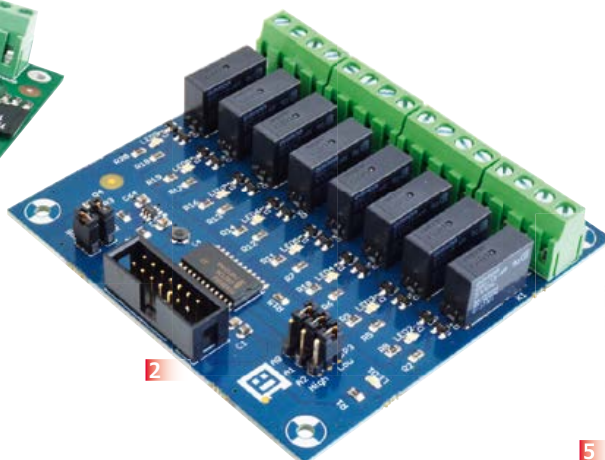
**L'onde U**, inconstante, fait suite à l'onde T. Elle est de même sens, mais d'amplitude moindre. Sa signification est discutée.

**L'intervalle QT** (début de QRS, fin de T) varie en fonction de la fréquence cardiaque. Pour une fréquence voisine de 60/mn, la durée de l'intervalle QT avoisine 0,4 s.

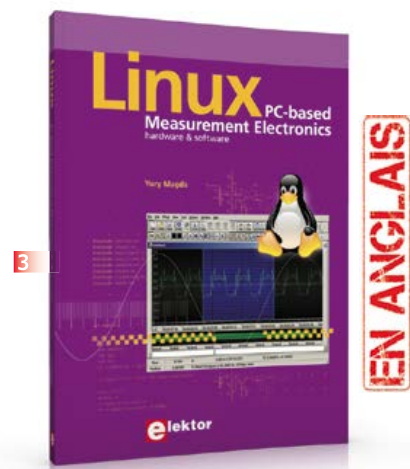
# •Elektor e-CHOPPE



1



2



3



5



4

Une carte compacte et bon marché qui vous initiera tout en douceur !

## 1 Embarquez Linux

Linux est partout, même dans certaines machines à café. Souvent, l'électronicien tenté d'adopter ce système d'exploitation est arrêté par sa complexité et par le prix des cartes de développement.

Voici Linux pour les électroniciens, sous la forme d'une carte compacte et bon marché qui vous initiera tout en douceur !

**Carte Linux Elektor (montée et testée)**

Réf. : 120026-91 • 64,95 €

## 2 Module d'extension pour les cartes à µC Linux et autres

La platine d'extension pour la carte Linux d'Elektor peut aussi se connecter à d'autres cartes à µC. Par exemple cette carte à huit relais. La carte d'extension conviendra aux débutants pour qui Linux est (encore) trop compliqué ; elle conviendra aussi aux utilisateurs chevronnés qui préfèrent sculpter leurs programmes « dans le métal brut » (sans l'aide d'un système d'exploitation).

Réf 130212-91 • 42,95 €

## 3 Linux – PC-based Measurement Electronics matériel et logiciel

Apprenez rapidement à construire des applications sous Linux, pour collecter, traiter et afficher des données sur un PC à partir de capteurs analogiques et numériques. Ne vous contentez pas de commander des circuits raccordés à un ordinateur. Apprenez aussi à transmettre les données via un réseau ou à communiquer sans fil avec des systèmes embarqués ! Aucun appareillage spécial ni logiciel coûteux n'est requis. Les composants électroniques et outils logiciels utilisés sont gratuits ou à faible coût.

**Langue : anglais • 263 pages**

**ISBN 978-1-907920-03-5 • 34,50 €**

**Kit (circuits imprimés & composants)**

## 4 Tapir – détecteur ultrasensible d'électrosmog

Fin limier de la pollution électromagnétique qu'il traque et rend audible sur casque, le TAPIR est (aussi) un beau projet à construire, un kit avec tout ce qu'il faut, même le boîtier, qui est fait de quatre circuits imprimés ingénieusement assemblés. Le TAPIR, sigle de Totally Archaic but Practical Interceptor of Radiation

(totalement archaïque mais pratique intercepteur de radiations), détecte et localise (en la faisant entendre) toute source de champ électrique E ou, avec l'antenne appropriée, de champ magnétique H.

Réf. : 120354-71 – 14,95 €

**Tous les articles d'ELEKTOR de l'année 2012**

**en français, anglais, allemand, espagnol et néerlandais**

## 5 DVD Elektor 2012

Ce DVD-ROM réunit tous les articles d'ELEKTOR, le mensuel d'électronique et de micro-informatique appliquées, parus au cours de l'année 2012. Il contient non seulement le texte des articles ainsi que les schémas, mais aussi tous les dessins des circuits imprimés, sous forme de fichiers à haute résolution.

**ISBN 978-90-5381-273-0 • 27,50 €**

**Informations complémentaires et gamme complète sur :**

**[www.elektor.fr/e-choppe](http://www.elektor.fr/e-choppe)**

Elektor/Publitrone SARL

1, rue de la Haye – BP 12910 – 95731 Roissy CDG Cedex

Tél. : +33(0)1.49.19.26.19 – Fax : +33(0)1.49.19.22.37

@ : ventes@elektor.fr

# carte Linux Elektor : version 2 !

## avec LAN et horloge en temps réel

Elektor propose depuis six mois une carte Linux compacte et bon marché, complétée par une série d'articles qui permet à chacun de faire ses premiers pas ou d'avancer dans le monde de l'embarqué Linux. Il était temps de sortir une nouvelle version de la carte qui prît en compte les retours de la communauté.

**Benedikt Sauter** [1]

De nombreux utilisateurs se servent des cartes Linux Elektor et GnuBLIN pour enregistrer des données (température ou autre consommation de courant), qu'ils interrogent ensuite via un réseau. Il était donc naturel d'équiper la nouvelle carte d'un accès réseau et d'une horloge temps réel (**fig. 1**). Nous avons également retenu d'autres propositions, comme l'agrandissement des trous de la carte afin qu'elle puisse être fixée avec de plus grosses vis.

### Puces supplémentaires

Le circuit de base de la carte ne diffère pas de celui de la version précédente [2]. Nous retrouvons le processeur, 32 Mo de RAM, un adaptateur USB/série (CP2102) et l'alimentation. Comme sur la première carte Linux Elektor, le connecteur GnuBLIN à 14 points EEC permet de relier plusieurs cartes d'extension, cartes que vous pouvez également vous procurer auprès d'Elektor [5]. Schéma et dessins de la carte au format Eagle peuvent être téléchargés gratuitement depuis la page [6] du site Elektor. Le lien [7] propose une version gratuite d'Eagle.

L'accès réseau a nécessité l'ajout d'un composant particulier, car le processeur LPC3131 ne dispose d'aucune interface réseau intégrée. Il s'agit de l'adaptateur réseau ENC28J60 de Microchip [8],

un contrôleur pour processeurs à 8 bits. L'accès se fait via SPI et par une ligne d'interruption supplémentaire.

L'archive du noyau contient depuis longtemps déjà un pilote pour ce contrôleur. Pour l'horloge temps réel (RTC), nous avons choisi le CI MCP7940 [9]. Il ne nécessite aucun quartz externe. Vous pouvez utiliser une pile bouton si vous souhaitez que l'heure soit conservée lorsque la carte est hors tension.

### Configuration du réseau

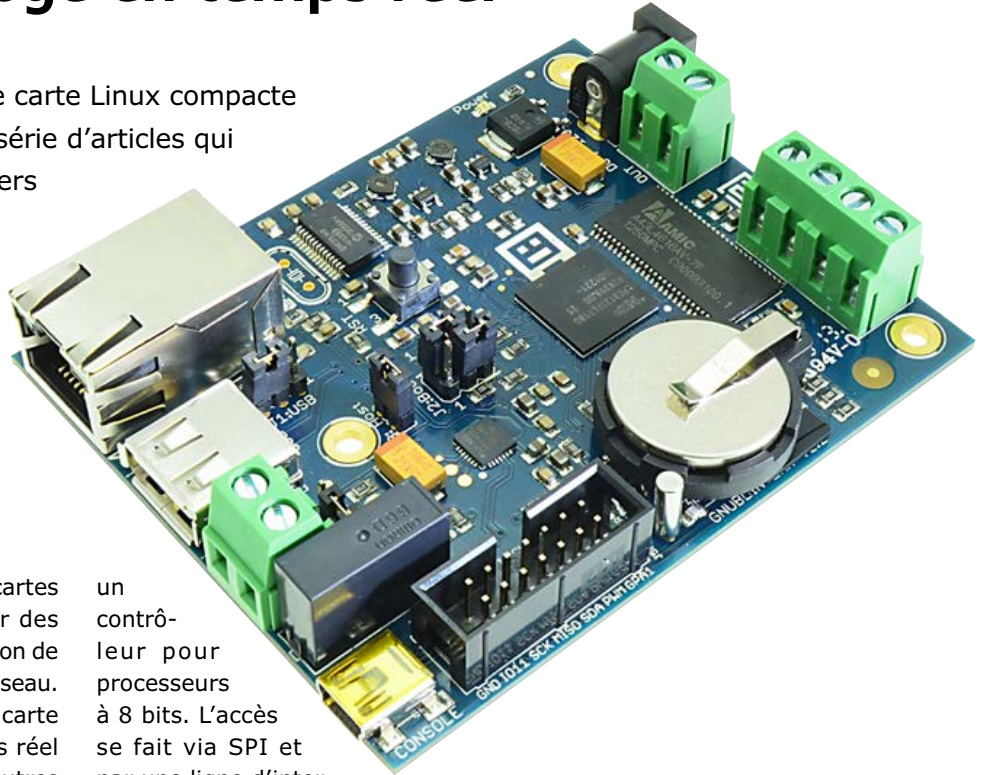
Voici les étapes à suivre avant de pouvoir utiliser la puce LAN ENC28J60. D'abord charger le pilote :

```
modprobe enc28j60 irq_pin=12 cs_pin=19
```

Entrez la commande suivante pour vérifier qu'il a bien été chargé :

```
ifconfig -a
```

La sortie devrait contenir « eth0 » parmi les





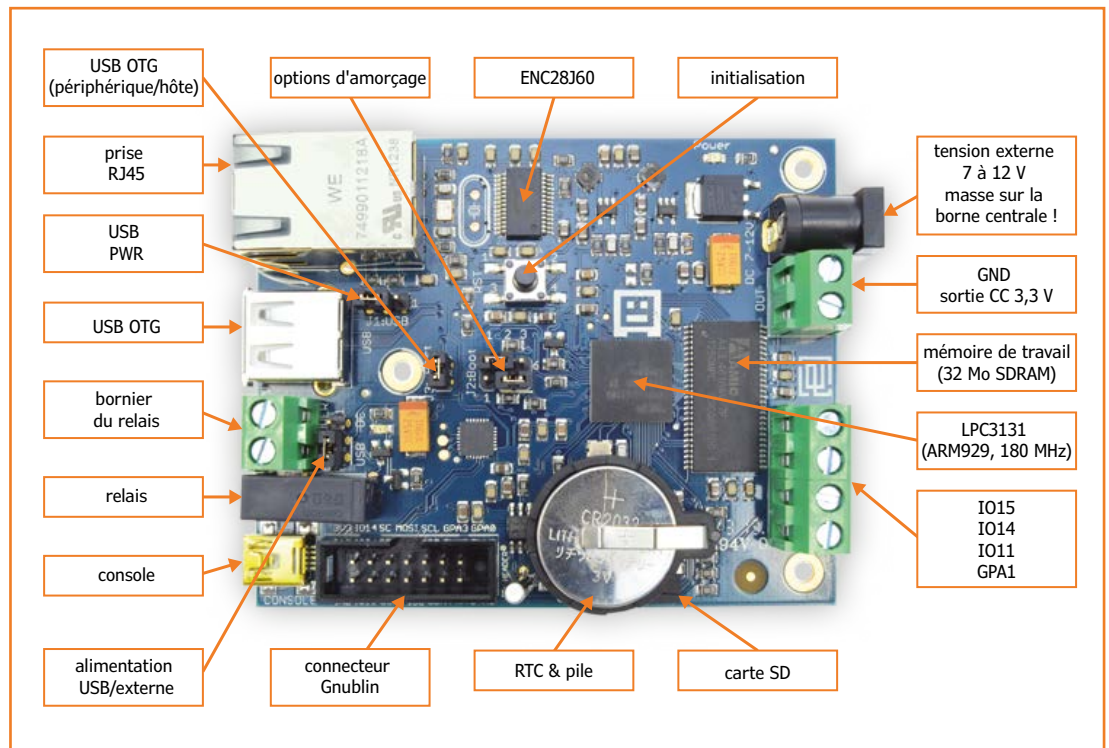


Figure 1.  
La carte Linux Elektor GnuBlin LAN est équipée d'un accès réseau et d'une horloge en temps réel.

interfaces détectées.

Entrez ensuite, pour obtenir une adresse IP du serveur DHCP :

```
dhclient eth0
```

Testez la connexion en envoyant une requête au PC d'un réseau, ou encore à un serveur Internet. Pour cela utilisez l'utilitaire *ping* :

```
ping google.fr
```

Vous devriez recevoir une réponse semblable à celle de la **figure 2**. Comme toujours sous console, un utilitaire se ferme avec Ctrl-C.

### Chargement automatique du pilote

Si vous souhaitez que le pilote soit chargé automatiquement au démarrage du système, ajoutez la ligne ci-dessous au fichier */etc/modules* du système de fichiers de la carte :

```
enc28j60 irq_pin=12 cs_pin=19
```

Le plus simple est d'effectuer cet ajout en ligne de commande :

```
echo "enc28j60 irq_pin=12 cs_pin=19" >> /
```

*etc/modules*

Les deux caractères >> demandent à echo d'ajouter la ligne à la fin du fichier sans en écraser le contenu. Vous pouvez aussi éditer le fichier, par exemple avec *nano* :

```
nano /etc/modules
```

### Adresse MAC fixe

Une nouvelle adresse IP est affectée à la carte après chaque démarrage. La puce ENC28J60 ne possède alors aucune adresse MAC fixe et renvoie une adresse différente à chaque chargement du pilote. Pour lui attribuer une adresse MAC fixe, nous pouvons ajouter la ligne suivante au fichier */etc/network/interfaces* :

```
hwaddress ether MAC-ADRESSE
```

Comme adresse MAC, vous pouvez choisir celle que le module LAN a reçue lors du premier chargement du pilote. Pour la connaître, servez-vous à nouveau de la commande *ifconfig* :

```
ifconfig
```

La sortie indiquera par exemple, pour *Hwaddr* :

```
eth0      Link encap:Ethernet  HWaddr
ba:07:1b:0c:64:0
```

## Horloge RTC

Une courte ligne de commande suffit pour configurer l'horloge en temps réel :

```
gnublin-rtc -s "2013/01/20 11:23:12"
```

Pour lire l'heure de l'horloge, entrez :

```
gnublin-rtc -g
```

La commande suivante met la date Linux à la date renvoyée par l'horloge :

```
gnublin-rtc -x
```

Quelques modifications sont nécessaires si l'on souhaite qu'à chaque démarrage la date et l'heure du système soient réglées en fonction de l'horloge RTC. Il faut d'abord ajouter les lignes suivantes dans le fichier */etc/rc.local*, avant la ligne « exit 0 » :

```
echo mcp7940 0x6f > /sys/bus/i2c/devices/
i2c-1/new_device
echo "Now setting the date and time."
sleep 1
hwclock --hctosys
```

Ajoutez ensuite l'entrée suivante au fichier */etc/modules* :

```
rtc-mcp7940
```

Le script *hwclock* doit être désactivé, car avec ce type d'horloge en temps réel il peut être responsable de certaines complications :

```
update-rc.d hwclock remove && update-rc.d
hwclock.sh remove
```

Après ces modifications, à chaque démarrage le

système de la carte affichera l'heure fournie par l'horloge matérielle. Si ce thème vous intéresse, consultez le wiki GnuBlin [10].

## Perspectives

La dernière version de l'installateur GnuBlin [11] permet de créer depuis Linux une carte SD qui contient un chargeur d'amorçage, un noyau, et un système de fichiers. Ce nouvel installateur graphique vous permet de choisir entre une image de 8 Mo ou de 32 Mo.

Autre changement, le système de fichiers utilise désormais ext4, système qui offre une protection presque totale en cas de panne de courant soudaine. Avec ext4 vous devriez ainsi éviter les interminables vérifications du système de fichiers (qui devaient être lancées après l'ancienne procédure de « réparation » de la carte SD).

Nous reviendrons sur ces nouveautés dans le prochain article.

(130214 – version française : Hervé Moreau)

## Liens

- [1] [sauter@embedded-projects.net](mailto:sauter@embedded-projects.net)
- [2] [www.elektor.fr/120181](http://www.elektor.fr/120181)
- [3] [www.elektor.fr/120596](http://www.elektor.fr/120596)
- [4] [www.elektor.fr/130212](http://www.elektor.fr/130212)
- [5] [www.elektor.fr/gnublin](http://www.elektor.fr/gnublin)
- [6] [www.elektor.fr/130214](http://www.elektor.fr/130214)
- [7] [www.cadsoft.fr/download-eagle/freeware](http://www.cadsoft.fr/download-eagle/freeware)
- [8] <http://ww1.microchip.com/downloads/en/devicedoc/39662a.pdf>
- [9] <http://ww1.microchip.com/downloads/en/DeviceDoc/22266D.pdf>
- [10] [http://en.gnublin.org/index.php/RTC\\_DS1307](http://en.gnublin.org/index.php/RTC_DS1307)
- [11] [http://en.gnublin.org/index.php/Gnublin\\_Installer](http://en.gnublin.org/index.php/Gnublin_Installer)

```
PING google.de (173.194.69.94) 56(84) bytes of data.
64 bytes from bk-in-f94.1e100.net (173.194.69.94): icmp_req=1 ttl=49 time=44.3 ms
64 bytes from bk-in-f94.1e100.net (173.194.69.94): icmp_req=2 ttl=49 time=44.7 ms
64 bytes from bk-in-f94.1e100.net (173.194.69.94): icmp_req=3 ttl=49 time=43.3 ms
64 bytes from bk-in-f94.1e100.net (173.194.69.94): icmp_req=4 ttl=49 time=43.4 ms
--- google.de ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 43.308/43.968/44.784/0.673 ms
```

Figure 2.  
Réponse à une requête *ping* : nous sommes sur le réseau !



# 4<sup>e</sup> jour – un projet simple

Neil Gruending  
(Canada)

La dernière fois, j'ai évoqué l'utilisation de bibliothèques (non, ce ne sont pas des *librairies*) dans *DesignSpark*. Cette fois je propose de réaliser un simple pilote de LED bicolore pour apprendre à utiliser les éditeurs de schémas et de circuits imprimés. Le circuit de commande LED bicolore que j'ai retenu est une variante du pont en H câblé pour que l'une des couleurs s'allume.

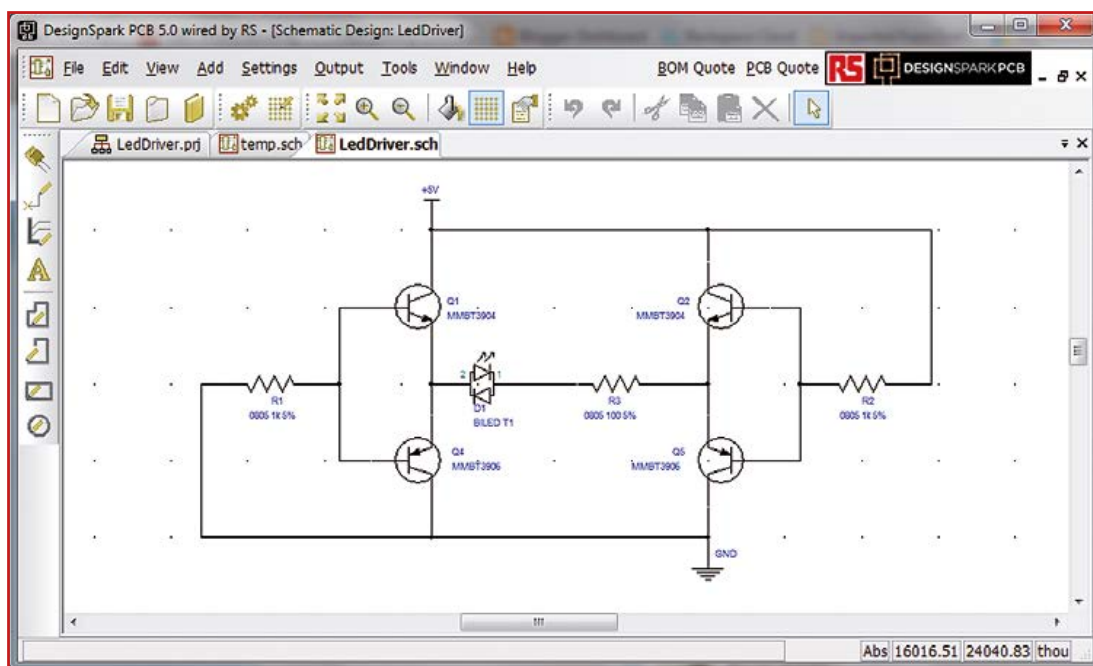


Figure 1.  
Schéma d'un étage de  
commande de LED bicolore.

## Dessiner le schéma

Il faut commencer par créer à l'aide du menu *File->New* un nouveau projet qui combinera schéma et circuit imprimé. Puis il faut ajouter le schéma à ce projet à l'aide de la commande *File->New* une fois de plus, mais en veillant à activer le bouton *Add To Open Project* (ajouter au projet en cours). À ce stade, vous pouvez aussi choisir le fichier à utiliser pour le schéma, comme nous l'avons vu dans un article antérieur. La **figure 1** montre le schéma que nous utiliserons. *Designspark* propose un tutoriel [1] de saisie de schéma.

La manipulation dans un schéma de DS de champs de composants visibles, comme p. ex. leurs références, diffère des opérations similaires dans d'autres applications, car tous les champs sont déplacés en bloc. Dans le schéma qui nous

sert ici d'exemple, il y a ainsi pour chaque transistor un numéro de composant et une référence. Que vous cliquiez sur l'un ou sur l'autre, les deux champs seront activés. Ils peuvent être déplacés ailleurs en groupe. Ce qui peut poser des problèmes d'alignement du texte avec la fonction de miroir que DS appelle d'ailleurs *flipping*. Il est heureusement très facile de modifier l'alignement du texte au moyen d'un clic droit et du menu *Properties*. L'onglet *Text* donne le choix de l'alignement : *Left*, *Center* et *Right*.

N'oubliez pas que DS considère les symboles de tension d'alimentation et de masse comme des composants. La bibliothèque de schémas de DS en propose de tout faits, mais vous pouvez créer votre propre bibliothèque avec les symboles qui vous conviennent. Remarquez que si vous connectez



tez un symbole d'alimentation à une liaison électrique existante, DS vous avertira qu'il remplace le nom de cette liaison même si ce n'est pas ce que vous souhaitez.

### Le composant LED

Pour les transistors et les résistances, je me suis contenté de symboles et d'empreintes de la bibliothèque standard de DS. Pour la LED, j'ai modifié un symbole de LED existant dans la bibliothèque de DS puis j'ai créé ma propre empreinte.

L'utilisation de la fonction d'assistance (*foot-print wizard*) facilite beaucoup la création d'une empreinte. Pour y accéder avec le gestionnaire de bibliothèques, ouvrir la bibliothèque de circuits imprimés là où vous voulez sauvegarder l'empreinte et cliquez sur le bouton *Wizard*...

Cet outil vous pose alors une série de questions pour la création de l'empreinte. Il importe de bien choisir le modèle le plus proche possible de ce que vous souhaitez afin de réduire le nombre de changements à faire. Pour ma LED, j'ai utilisé un composant axial au pas de 2,54 mm, de sorte que je n'ai eu à éditer que le masque de sérigraphie et à repérer la polarité sur la broche 1. DS propose une fonction d'assistance similaire pour les symboles de schémas et de composant.

### Préparer le tracé

Nous voici prêts pour tracer notre circuit imprimé qu'il faut donc créer avec les commandes du menu *Tools->Translate to PCB*. L'assistant *New PCB* est lancé. Créons un double face carré de 20 mm. Si vous demandez à l'assistant d'implanter les composants hors du circuit, vous obtiendrez quelque chose comme la **figure 2**. Je préfère travailler sur une grille de 0,25 mm ; j'en change donc le pas avant de placer les composants (**fig. 3**).

Avant de passer au routage, je voudrais parler de l'espacement des pistes de cuivre sur la platine. DS n'offre pas de fonction interactive lors du routage automatique, c'est donc à vous de fixer la largeur de piste. Ainsi, quand deux pistes se touchent l'espacement est 0 mm, et quand il y a un espace, il est (au moins) de la largeur de la piste. Ceci est lié au fait que la grille de routage s'applique au centre de la piste et pas sur ses bords. Ainsi, pour le routage d'une piste de 0,2 mm vous fixerez la grille de routage à 0,2 mm

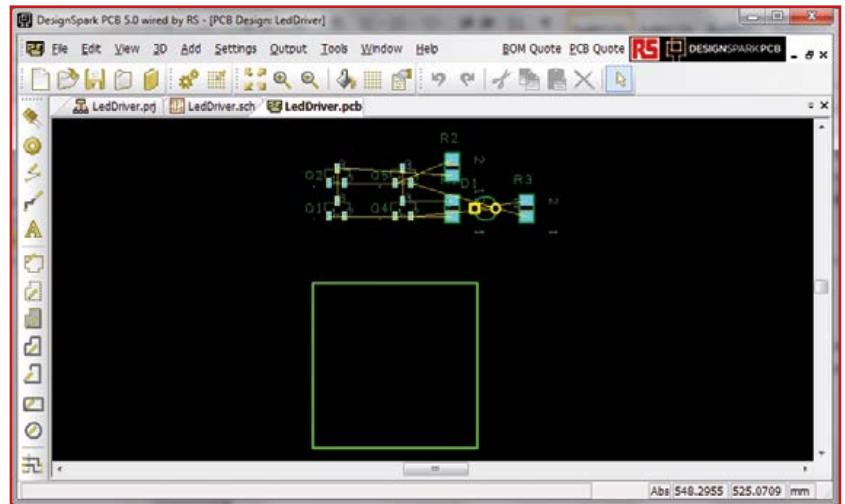


Figure 2. Je commence par mettre les composants à côté de la carte.

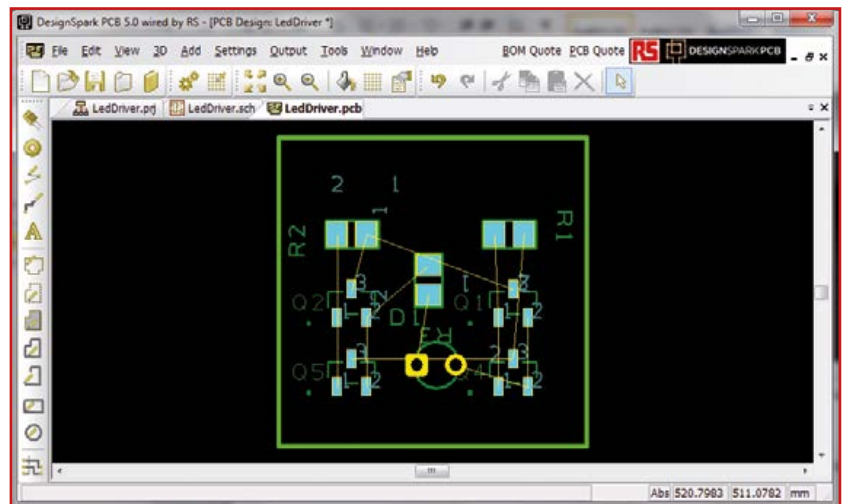


Figure 3. Puis je mets les composants sur la carte dont la grille est espacée de 0,25 mm.

pour obtenir cet espacement. L'inconvénient de cette façon de faire est d'imposer que la largeur des pistes soit toujours un multiple de la plus fine d'entre elles. Avec par exemple 0,2 mm et 0,6 mm, ça marche, mais pas avec 0,2 mm et 0,35 mm.

N'omettez pas de créer dans les préférences un style pour chaque largeur de piste souhaitée (*Settings->Design Technology...* puis choisir *Track Styles*). Cela facilite la gestion des différentes largeurs de piste dans les circuits plus

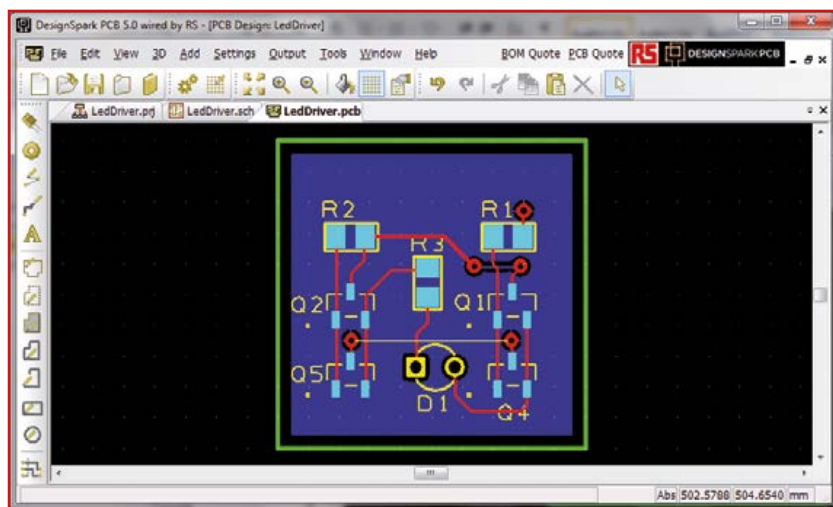


Figure 4. Et voilà le résultat.

compliqués, car vous pourrez ainsi changer la largeur d'une piste en changeant simplement de style. Pour cela, pendant le routage d'une piste, il suffira de taper "s" et de choisir le nouveau style souhaité.

Ceci vaut aussi pour les vias. Dans DS, vous configurez les styles des vias avec l'onglet *Pads Styles* dans la fenêtre *Design Technology*. Je vous conseille de créer au moins un style *SignalVia* et tout autre style supplémentaire qui pourrait correspondre à vos besoins. Ici j'ai opté pour des vias de signal de 0,45 mm pour le trou et 0,95 mm pour la pastille. Pour changer de style de via pendant le routage, un clic droit vous propose le menu *Change Via Style*.

DS utilise le menu *Settings->Defaults* pour fixer les valeurs par défaut des pistes et des vias, mais je n'ai pas réussi à obtenir qu'il reconnaisse mes nouveaux réglages, même après redémarrage. Le site [2] donne toutes les informations.

## Routage

Cette fois il est temps de router (**fig. 4**). J'ai mis les signaux sur la face supérieure, avec un plan de masse sur l'autre face. Pour le routage d'une liaison, il importe de double cliquer sur la ligne correspondante dans le chevelu. Chemin faisant, le clic droit vous permet d'influer sur la manière dont DS trace les angles et les segments. En regardant bien, vous verrez une liaison entre Q4 et Q5 pour la connexion de masse ce qui indique que DS ne voit pas que ces deux tran-

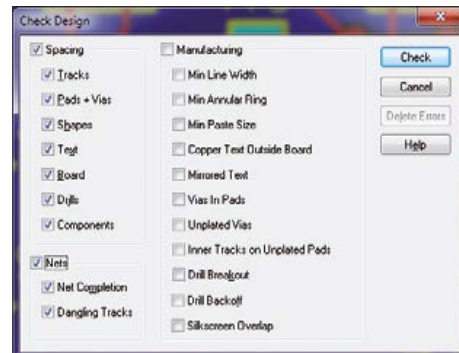


Figure 5. Une ultime vérification avec l'outil *Design Rule Check (DRC)*.

sistors sont (déjà) à la masse. Le programme connaît une fonction DRC (*design rule check*) de vérification (menu *Tools*), avec laquelle on peut vérifier toutes les connexions (**fig. 5**).

Cliquez sur le bouton *Check* pour que DS vérifie que le tracé du circuit imprimé est conforme aux critères et établisse un rapport détaillé. Il signalera les erreurs éventuelles. Celles-ci apparaissent aussi dans le tracé à l'écran. Tous les critères d'espacement entre éléments conducteurs non connectés (*clearance rules*) sont fixés dans l'onglet *Spacing* sous forme de tableau. Ne vous privez pas du tutoriel préparé pour vous [3].

## Conclusion

Pour cet article, nous nous sommes cantonnés à un circuit simple, à partir d'un schéma, et nous l'avons vérifié avec l'outil de vérification de *DesignSpark*. Pour le prochain, nous passerons par une BOM (= liste de composants) et des fichiers Gerber.

(130230)

## Liens

- [1] [www.designspark.com/eng/tutorial/schematics-entry](http://www.designspark.com/eng/tutorial/schematics-entry)
- [2] [www.designspark.com/tutorial/pcb-setup-placing-components](http://www.designspark.com/tutorial/pcb-setup-placing-components)
- [3] [www.designspark.com/tutorial/pcb-routing](http://www.designspark.com/tutorial/pcb-routing)

# énergie négative

Vous connaissez la nouvelle la plus courte de la SF ? « Le dernier homme vivant sur la Terre se trouvait chez lui. On frappa à la porte. » Terrifiant, non ? Moins éprouvant pour vos nerfs, voici un autre mystère, celui de la porte de chez mes beaux-parents. La semaine dernière, j'arrive chez eux, j'appuie sur le bouton de la sonnette et... rien. C'est moi qui avais installé cette sonnette sans fil il y a deux ans ; j'en avais même changé les piles il y a deux mois. Bref, me voilà moi aussi face à une porte, à du mystère, du suspense et même de la tension.

Enfin, pour la tension, ça reste à voir. Je demande à mon beau-père de bien vouloir sortir son multimètre. Il me tend un jeu de piles neuves... Le sourcil noir, je lui décoche une œillade peu amène, l'air de dire : « Laissez donc faire les professionnels ». Il n'insiste pas et me ramène son antiquité à cadre mobile portant l'inscription « testeur de piles ». A priori, ça devrait faire l'affaire. La LED rouge de contrôle s'allume quand on appuie sur le bouton de la sonnette. Côté carillon, par contre, rien ne s'allume, et *Big Ben* reste muet. C'est donc la cloche qui cloche.

Plusieurs hypothèses défilent alors dans mon esprit :

1. Farce de beau-papa (possible, mais improbable)
2. Piles déjà déchargées (probable – prévoir sarcasmes de beau-papa)
3. Sonnette fichue (ça arrive aux meilleures)
4. Autre (comme dans les sondages)

Et vous, des soupçons ? Une idée ? Une intuition ? J'ouvre le compartiment des piles, en sors les trois AA remplacées par bibi deux mois auparavant, puis je mesure la tension entre les piles en me servant de la position 4,5 V du testeur. Résultat : 1,5 V.

L'hypothèse n° 2 se confirmerait-elle ? Sourire en coin de Beau-papa. Je positionne le testeur sur 1,5 V et mesure la tension des piles une par une. Pour la première je trouve... 1,5 V ! Les deux autres seraient donc défectueuses ? Je mesure la deuxième qui donne... encore 1,5 V ! Là je vous arrête tout de suite : oui, les piles avaient été insérées dans le bon sens. Oui, je restais zen, parfaitement maître de moi, même si d'habitude,



quand je rends visite à mes beaux-parents, il ne se passe pas autant de trucs bizarres, surtout dans un domaine où je suis réputé compétent. Là, j'avais deux piles de 1,5 V et une tension totale de 1,5 V. Qu'allait me réserver la troisième ? Je la branche, l'aiguille part dans le mauvais sens. Je change les pointes de test, rouge sur le moins, noir sur le plus : 1,5 V !!

$1,5 \text{ V} + 1,5 \text{ V} - 1,5 \text{ V} = 1,5 \text{ V}$ . Logique ! Je remplace la pile « négative ». Aussitôt *Big Ben* sonne et Beau-papa, goguenard, sifflote. Reste la question : d'où sort cette pile avec une tension négative ? Vous saviez que ça existe ? Non, il n'y a pas de piles négatives ; d'ailleurs, je suis sûr que deux mois auparavant, lorsque j'avais changé les piles, le récepteur (le côté « cloche ») fonctionnait.

Rentré chez moi, je remesure la tension de cette *anti-pile* à l'aide d'un multimètre numérique de précision. Elle possédait toujours autant d'énergie *négative*, mais plus tard, après quelques heures de stockage, sa tension était déjà retombée à -35,9 mV.

L'explication est finalement assez simple : un défaut de fabrication de cette cellule en avait probablement affaibli la capacité d'origine. Une fois insérée avec les deux autres dans le récepteur (celui-ci reste sous tension en permanence), elle s'est très vite épuisée ; puis, sous la tension de 3 V des deux piles en bon état, un courant a continué de circuler et a inversé la polarité de la pile défectueuse.

(130108 – version française : Hervé Moreau)

**Thomas Scherer**  
(Allemagne)



# premiers pas avec le Mini Kit LPC800



Juste avant l'été, Elektor a proposé à ses lecteurs de recevoir gratuitement une petite carte à microcontrôleur. Cette carte est un Mini Kit LPC800 qui comporte un  $\mu\text{C}$  à 32 bits ARM Cortex-M0+ LPC810, en boîtier DIP à 8 broches, un régulateur de tension, deux boutons-poussoirs, une LED et deux petites plages de prototypage. La campagne a été un énorme succès : toutes les cartes sont parties en trois quarts d'heure...

**Clemens Valens**  
(elektor.LABS)

Si vous étiez parmi les chanceux qui ont réussi à mettre la main sur une de ces cartes, vous avez eu tout l'été pour l'expérimenter. Maintenant, nous savons ce qu'il advient de ces choses : vous recevez le kit, vous ouvrez la boîte, vous examinez la carte, vous la connectez à un PC pour voir la LED clignoter et vous la posez sur votre bureau pour une utilisation prochaine. Pour beaucoup de ces kits, c'est là qu'ils sont encore. C'est pourquoi, afin de le sauver de l'e-oubli, voici un court tutoriel pour vous expliquer comment démarrer avec ce petit kit.

Pour jouer avec le Mini Kit, il vous faudra :

- un Mini Kit ;
- un PC avec un port sériel compatible avec les niveaux logiques 3,3 V ;
- une alimentation 5 V ;
- une interface SWD (*Serial Wire Debug*) optionnelle.

Vous pouvez cocher la première ligne – pour cocher la deuxième, il vous faut soit un PC avec un vrai port sériel connecté à un adaptateur de niveau MAX3232 alimenté en 3,3 V, soit (plus pratique parce qu'il règle aussi le troisième point)

un câble « FTDI » (disponible à l'e-choppe Elektor, réf. 080213-72). Vous pouvez aussi utiliser notre pont USB/Série « BOB » (e-choppe, réf. 110553-91), plus souple mais qui a besoin d'un fer à souder chaud pour souder le cavalier en position 3,3 V. Le câble FTDI et le BOB peuvent l'un et l'autre fournir la tension d'alimentation de 5 V nécessaire au Mini Kit.

Le Mini Kit est muni d'un connecteur compatible avec le câble FTDI 3,3 V, mais je préfère utiliser un BOB monté sur une petite carte d'adaptation sur laquelle j'ai installé quelques cavaliers qui me permettent de déconnecter les lignes de signal sériel sans couper l'alimentation (**figure 1**). C'est utile pour les essais parce que certaines fonctions du micro partagent des broches avec le port sériel. Il est possible aussi de choisir la tension de la ligne VCC (0 V, 3,3 V ou 5 V). Si vous utilisez ce circuit adaptateur, placez un cavalier entre les broches 1 et 2 de JP1 pour fixer VCC à 5 V. C'est nécessaire pour le fonctionnement du régulateur de tension du Mini Kit.

Le port sériel est nécessaire pour la programmation du  $\mu\text{C}$  – un vrai débogage n'est pas possible. Si vous avez une interface SWD (*Serial Wire*

*Debug*), vous pouvez l'utiliser à la place du port sériel pour charger vos programmes dans le  $\mu$ C. L'interface permet aussi de les déboguer. Comme – malheureusement – je n'ai pas d'interface SWD, j'utiliserai le port sériel dans la suite de cet article. Une fois le matériel rassemblé, vous avez un peu de logiciel à installer. Tout le nécessaire est gratuit, il suffit de le télécharger. Vous irez donc chercher :

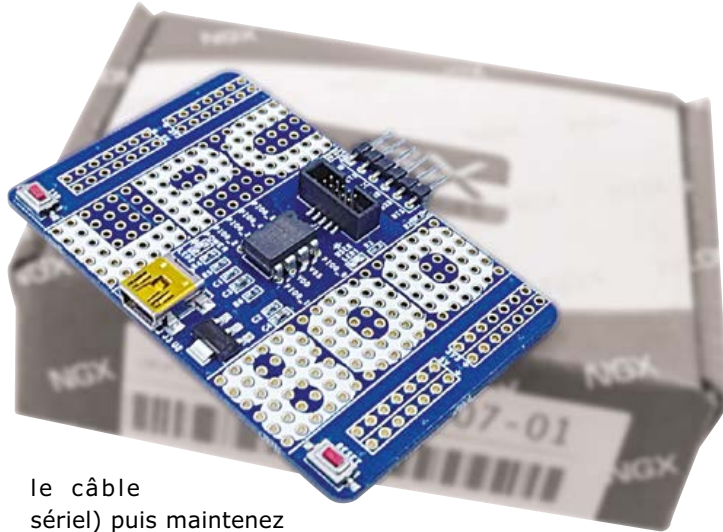
- LPCXpresso IDE (gros, nécessite une inscription) [1] ;
- *Flash Magic* [2] ;
- LPC800 Mini Kit code base [3].

Inutile de préciser qu'il vous faut la dernière version de chaque outil et bibliothèque. Installez d'abord les deux premiers outils dans n'importe quel ordre (ceux qui comptent utiliser une interface SWD n'ont pas besoin de *Flash Magic*). Décompactez le code de base quelque part sur un disque – le répertoire de votre projet serait un bon choix.

Avec *Flash Magic* installé, vous pouvez vérifier la communication entre le Mini Kit et le PC. Connectez la carte au port sériel (par un BOB, un câble FTDI ou un adaptateur de niveau maison) et vérifiez qu'elle est alimentée. Vous pouvez aussi alimenter la carte par le connecteur USB si vous préférez (ce connecteur ne sert qu'à l'alimentation, les lignes de données sont en l'air). Si votre carte est neuve, la LED témoin d'alimentation devrait s'allumer et la LED utilisateur commencer à clignoter.

Appuyez sur le bouton de programmation *in situ* (ISP) pendant que vous appuyez sur *Reset*. La LED qui clignote devrait cesser. Lancez *Flash Magic*, choisissez (*Select...*) le bon type de  $\mu$ C (LPC810M021FN8) et le port sériel (*COM Port*). Le débit (*Baud rate*) peut aller jusqu'à 115 200 bauds, mais j'ai eu des problèmes à cette vitesse ; je n'en ai jamais eu à 38 400 bauds. Le champ *Interface* sera *None (ISP)* et le champ *Oscillator* peut rester vide (**fig. 2**). Dans le menu *ISP*, sélectionnez *Read Device Signature*. Une fenêtre s'ouvre et devrait se remplir de données si tout va bien. Si la réponse est *autobaud error*, il est probable que votre câble ne fonctionne pas correctement ou que le  $\mu$ C n'est pas en mode ISP. Recommencez après avoir vérifié votre câble et peut-être à une vitesse différente.

Il est possible que la broche *Reset* du  $\mu$ C ait été inhibée (lors d'expériences précédentes). Dans ce cas, coupez l'alimentation (ou déconnectez



le câble sériel) puis maintenez le bouton ISP pendant que vous la rétablissez. Ce truc remet toujours le  $\mu$ C en mode ISP. Si vous ne pouvez toujours pas lire l'identification de la puce, vous avez un problème de connexion. Il est temps de lancer l'environnement de développement intégré (*IDE*) LPCXpresso. Quand on vous demande de désigner un espace de travail (*workspace*), choisissez le répertoire où vous voulez ranger vos projets. Rappelez-vous le chemin

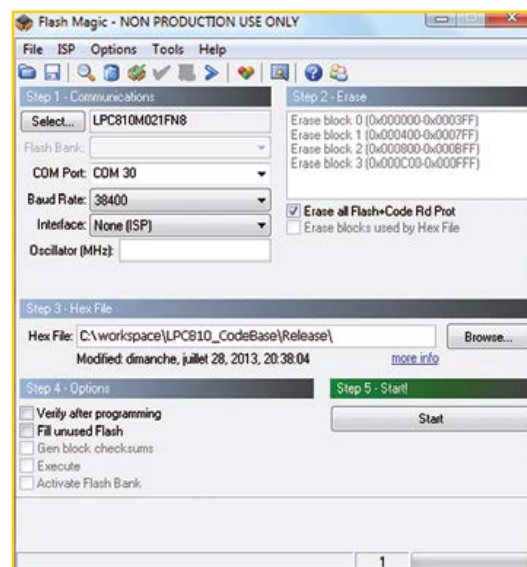
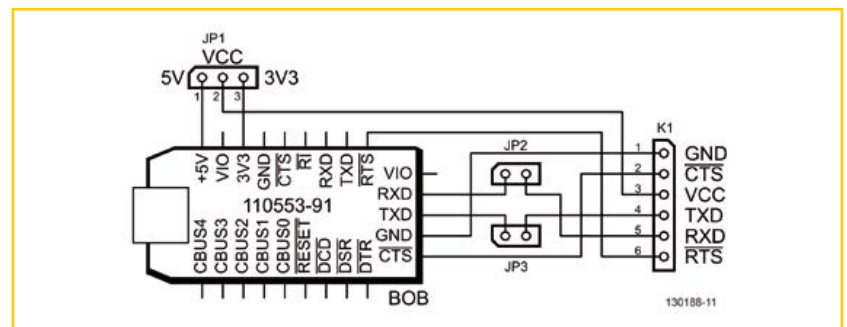


Figure 1.  
La carte d'adaptation BOB d'Elektor, utilisée pour programmer le Mini Kit LPC800.

Figure 2.  
*Flash Magic* et les réglages qui fonctionnent pour moi.



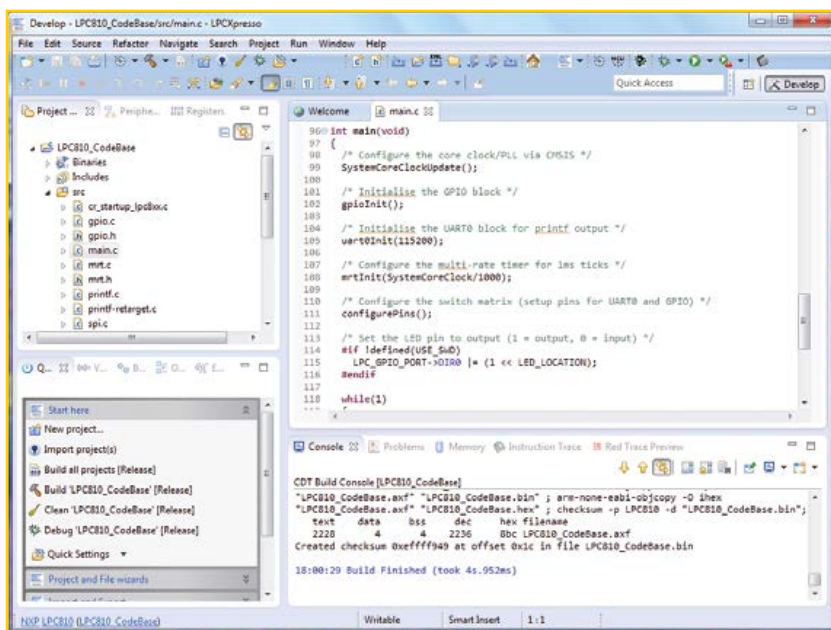


Figure 3. L'IDE LPCXpresso rend compte d'une construction réussie du projet LPC800\_CodeBase.

parce que vous en aurez besoin plus tard. L'IDE met du temps à démarrer mais quand il finit par être prêt, il propose un menu à accès rapide, nommé *Start here*, qui contient les fonctions les plus importantes (et quelques autres), utilisées souvent, comme *new project*, *build* et *debug*. Vous y trouverez aussi l'option d'importer des exemples de projets. Cliquez sur le lien *Import project(s)* pour ouvrir la boîte de dialogue *ad hoc*, puis cliquez sur *Browse* (parcourir) à droite du champ *Root directory* si vous avez décompacté le code de base (si vous

[www.elektor-labs.com](http://www.elektor-labs.com)

l'avez gardé sous forme d'archive, vous pouvez cliquer sur l'autre bouton *Browse*) et naviguez jusqu'au répertoire *LPC800 Mini Kit Code Base*. Une fois que vous l'avez trouvé, cliquez sur *OK* puis *Next*. Assurez-vous que le projet *LP810\_CodeBase* est coché avant de cliquer sur *Finish*.

Vous avez maintenant un projet nommé *LPC810\_CodeBase* dans la fenêtre *Project Explorer*. Sélectionnez-le pour le construire (*build*) à partir du menu *Start here*. Observez les messages qui défilent dans la fenêtre *Console* ; il ne devrait y avoir ni erreurs ni avertissements (fig. 3). Si pour quelque raison vous voyez une erreur ou un avertissement, cliquez sur l'onglet *Problems* pour avoir davantage d'informations. Un double clic sur une ligne de cette fenêtre vous amènera sur la portion de code incriminée.

Après une construction réussie, vous trouverez un fichier HEX dans le répertoire *Release* de votre projet, copié dans votre espace de travail lors de l'importation. Cliquez sur le bouton *Browse* de *Flash Magic* pour naviguer jusqu'à ce fichier HEX. Placez votre Mini Kit en mode ISP avant de cliquer sur *Start*. Si tout est correct, le µC sera programmé avec votre nouveau fichier HEX. Appuyez sur le bouton *Reset* de la carte pour lancer le programme.

Vous voilà prêt à développer vos propres programmes. Le site [elektor.LABS](http://www.elektor.LABS) [4] propose quelques projets pour vous mettre le pied à l'étrier avec cette carte. Si vous faites quelque chose de terrible, utile, intéressant ou autre avec le Mini Kit, nous aimerions en entendre parler. Pour cela, veuillez envoyer vos projets LPC800 à [elektor.LABS](mailto:elektor.LABS).

(130188 – version française : Jean-Paul Brodier)

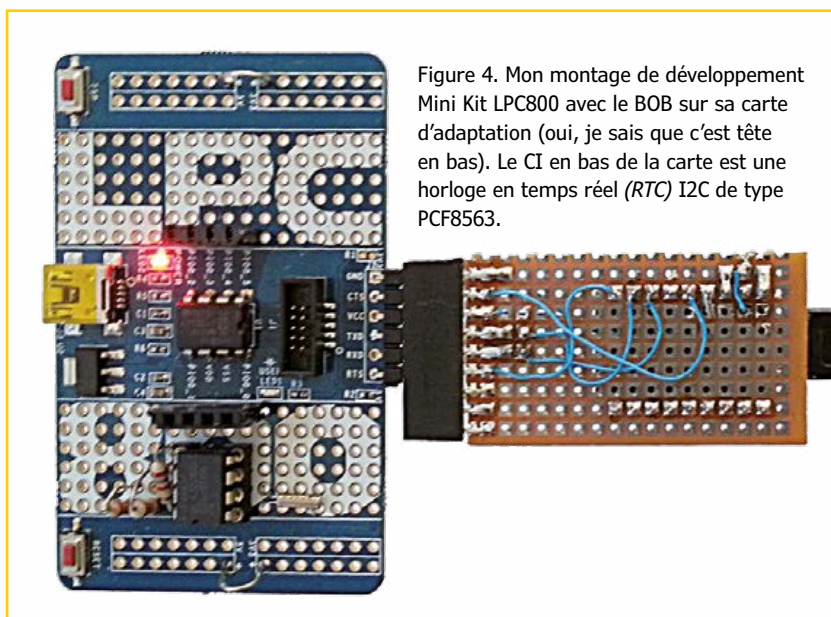


Figure 4. Mon montage de développement Mini Kit LPC800 avec le BOB sur sa carte d'adaptation (oui, je sais que c'est tête en bas). Le CI en bas de la carte est une horloge en temps réel (RTC) I2C de type PCF8563.

## Liens

- [1] LPCXpresso IDE:  
<http://lpcxpresso.code-red-tech.com/>  
LPCXpresso
- [2] Flash Magic: [www.flashmagictool.com](http://www.flashmagictool.com)
- [3] LPC810 Code Base:  
<http://lpcware.com/lpc800-mini-kit>
- [4] [elektor.LABS](http://www.elektor-labs.com): [www.elektor-labs.com](http://www.elektor-labs.com)



# Guide de démarrage pour Eagle V6

**NOUVEAU**

## Trucs & astuces

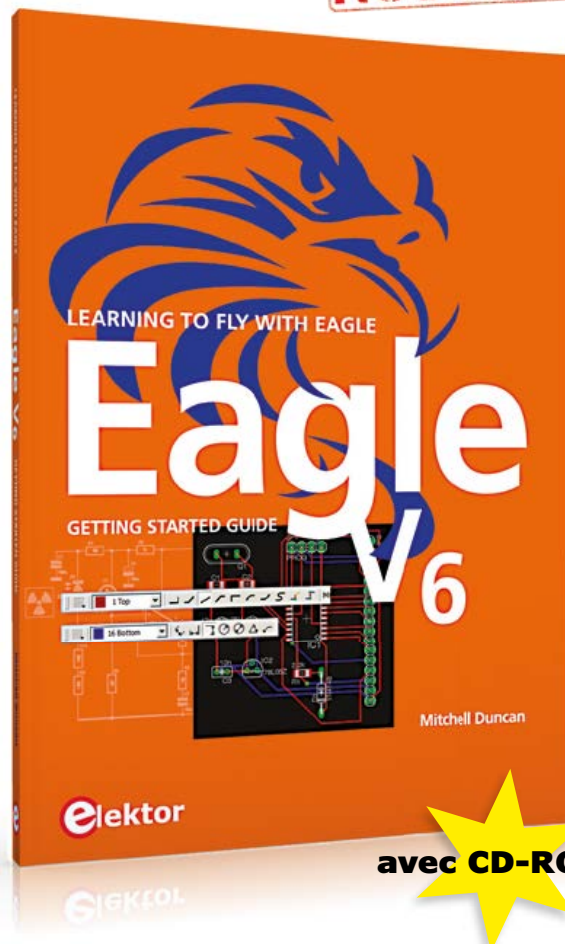
L'IDE LPCXpresso repose sur Eclipse, un outil répandu et pourtant – à mon humble avis – horrible. C'est pourquoi je recommande à l'utilisateur débutant de commencer par le projet donné en exemple pour les nouveaux projets. Vous pouvez copier et coller un projet dans la fenêtre *Project Explorer* par des clics à droite – ce qui vous assurera des réglages corrects.

Les projets construits de zéro ne produiront pas de fichier HEX. Pour y remédier, copiez les réglages du projet exemple : sélectionnez le projet exemple ; dans le menu, cliquez sur *Project* puis *Properties*. Déroulez *C/C++ Build* ; cliquez sur *Settings* puis sur l'onglet *Build Steps*. Copiez le contenu du champ *Command* de la zone *Post-build steps* et collez-le dans Notepad ou quelque chose de similaire, pour éviter de le perdre dans les étapes suivantes. Fermez la boîte *Properties*. Sélectionnez maintenant le nouveau projet et répétez les étapes ci-dessus pour revenir à *Post-build steps*, mais cette fois dans le nouveau projet. Remplacez-y la ligne de commande par celle que vous venez de copier. Cliquez sur *OK* pour sauvegarder les réglages. Répétez cette procédure pour toutes les configurations que vous aurez créées (*release*, *debug*, autres).

Construisez seulement pour *Release*, sinon vous serez vite à court de mémoire de programme (le LPC810 n'en a que 4 Ko). Naturellement, si vous avez une interface SWD, vous pouvez construire aussi des versions *debug*, mais les utilisateurs de *Flash Magic* n'ont rien à y gagner.

L'ajout de fichiers sources à un projet est plutôt contre-intuitif (ou devrais-je dire contre-productif ?) : *File* → *Import...* → choisissez *Filesystem*, cliquez sur *Next*, naviguez jusqu'à l'emplacement du fichier, cochez le fichier voulu et cliquez sur *Finish*. Comme l'IDE ne retient pas le dernier chemin emprunté, un grand nombre de clics de souris peut être nécessaire pour arriver au but. Vous pouvez aussi bien copier le fichier directement dans le répertoire avec votre outil préféré (Total Commander dans mon cas). Après avoir copié les fichiers, appuyez sur F5 pour rafraîchir votre projet et faire apparaître les nouveaux fichiers.

La fonction de remise à zéro du µC peut être déconnectée de la broche, ce qui la libère pour un autre usage (voir la matrice de commutations du µC). Le bouton Reset de la carte devient inopérant et le passage en mode ISP est plus difficile. Dans ce cas, vous devrez couper l'alimentation de la carte et tenir enfoncé le bouton ISP pendant la remise sous tension. Cette manœuvre passe toujours le µC en mode ISP et de plus elle réactive l'interface SWD si vous l'aviez déconnectée.



## LEARNING TO FLY WITH Eagle V6 GETTING STARTED GUIDE

Que vous soyez électronicien amateur ou professionnel, ce guide de prise en main vous initiera rapidement au logiciel de conception de circuits imprimés EAGLE de CadSoft.

Ce livre (en anglais) a été écrit pour tous ceux qui souhaitent découvrir les possibilités d'EAGLE. L'auteur s'adresse aussi bien au novice en conception de circuits imprimés qu'au professionnel déjà familier d'un autre logiciel de CAO, mais qui envisage de passer à EAGLE.

Ce livre est accompagné d'un CD-ROM gracieusement fourni par CadSoft Computer GmbH ([www.cadsoft.de](http://www.cadsoft.de)). Il contient une version d'Eagle pour Windows, Linux et Mac.

ISBN 978-1-907920-20-2

206 pages – 34,50 €

**elektor** [www.elektor.fr/eagle](http://www.elektor.fr/eagle)

# Importante mise à jour du LCR-mètre 0,05 % version 3.0.0

Jean-Jacques Aubry,  
Ollioules

Depuis la publication de ce projet en trois articles [1], plusieurs centaines d'exemplaires de l'appareil ont été fabriquées, pour la plus grande satisfaction de leurs utilisateurs et de l'auteur. Celui-ci a remédié à deux imperfections susceptibles de perturber le fonctionnement dans certaines conditions.

Les deux points dignes d'attention sont d'une part le fait que l'appareil semble figé lors de la mesure de faibles résistances ( $< 1 \Omega$ ) et d'autre part l'impossibilité de réaliser les *Trim*. Ces problèmes ont déjà été commentés par l'auteur sur les forums [2] [3], mais il n'est pas inutile d'y revenir ici.

Lors de la mesure de résistances de valeur inférieure à  $1 \Omega$ , afin d'avoir une tension de mesure conséquente, le logiciel interne (*firmware*) règle au maximum le **gain pour la mesure « tension »** :

- gamme 1 ( $R_{\text{sense}}$  de  $100 \Omega$  et gain du PGA103 de 100)
- gain de l'amplification finale proche du maximum (palier E ou F).

Malheureusement, la tension de décalage à l'entrée est fortement amplifiée elle aussi et sa compensation n'est plus assurée avec le schéma publié en mars 2013. D'où un possible dépassement permanent de la tension maximale à l'entrée du convertisseur analogique-numérique. Ce phénomène erratique n'est pas systématique et dépend beaucoup de la tension d'*offset* cumulée de U6 et de U4 et de la valeur de la (faible) résistance (ou inductance) mesurée. Et d'une !

D'autre part, quand aucun composant n'est connecté lors de l'opération de « TRIM – OPEN-CIRCUIT », là aussi le **gain pour la mesure « courant »** est maximum à la fréquence de 100 Hz (ou 120 Hz) :

- gamme 8 ( $R_{\text{sense}}$  de  $100 \text{ k}\Omega$  et gain du PGA103 de 100)
- gain de l'amplification finale proche du maximum (palier E ou F).

À ce moment, le circuit de mesure « courant » est très sensible aux parasites captés par les cor-

dons de mesure, particulièrement ceux du secteur (50 Hz ou 60 Hz). Le résultat de la mesure est alors si instable que le *firmware* refuse de l'afficher et il ne valide pas le « TRIM ». Et de deux !

## 2 problèmes, 2 solutions

Selon que vous êtes un adepte du fer à souder ou pas, il existe deux remèdes :

- Intervention sur le matériel – *la meilleure solution* – pour modifier le schéma électrique de la compensation de la **tension de décalage** de l'entrée et mise à jour du *firmware* (V. 3.0.0) et du programme AU2011 (V. 3.0.0). La modification du matériel sera détectée automatiquement par le logiciel, à la mise en route, du fait de la présence d'une résistance sur la ligne de port P2.2 (LCD\_SI) (fig. 1).
- la mise à jour avec les mêmes versions du *firmware* (V. 3.0.0) et du programme AU2011 (V 3.0.0), sans modification du matériel.

**Important : La mise à jour du logiciel en version 3.0.0. est donc à faire dans tous les cas. La nouvelle version est compatible aussi bien avec le matériel en l'état qu'avec la version modifiée décrite ci-dessous.**

## Modification du matériel

Telle que conçue à l'origine, la compensation de la tension de décalage à l'entrée est réalisée par injection d'un courant à l'entrée de U6. Le résultat est malheureusement trop dépendant de la résistance (en continu) du DUT.

Le nouveau schéma (**fig. 1**) applique la tension de correction au niveau de la sortie de U5 (INA128)

et l'impédance du DUT n'a donc plus d'influence ; cela permet aussi de séparer la compensation pour les mesures « courant » et « tension ». Pour cela, la broche 5 de U5 est reliée non plus à la masse analogique mais, à travers une résistance de faible valeur, à une tension ajustable par logiciel.

La modification comporte quatre étapes :

- retirer R34 afin de désactiver l'ancienne compensation. R42 et C35 ne servent plus et peuvent aussi être retirées ;
- remplacer R46 par une résistance (0805 de préférence) de 10  $\Omega$ . On peut aussi souder la résistance de 10  $\Omega$  directement sur la résis-

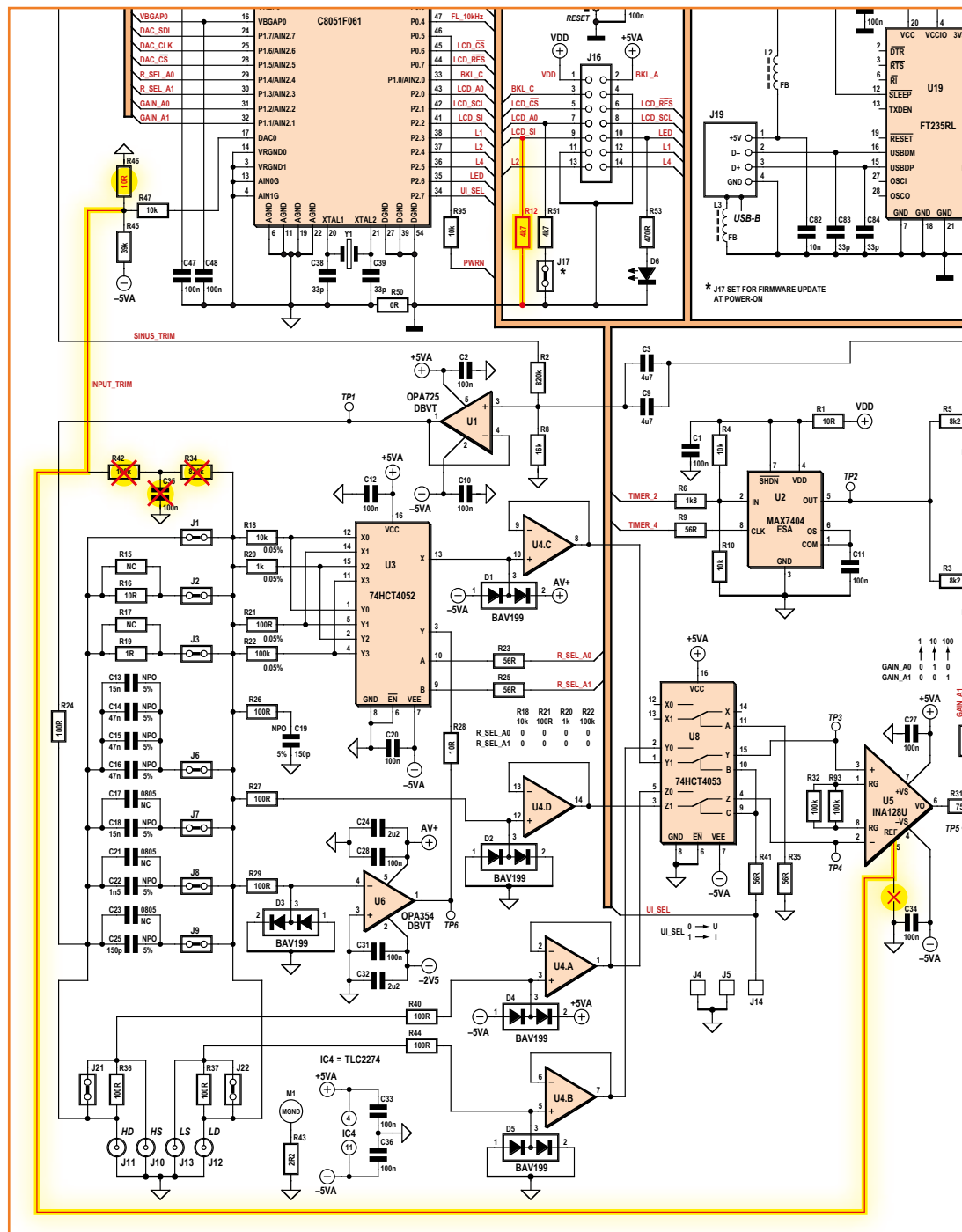
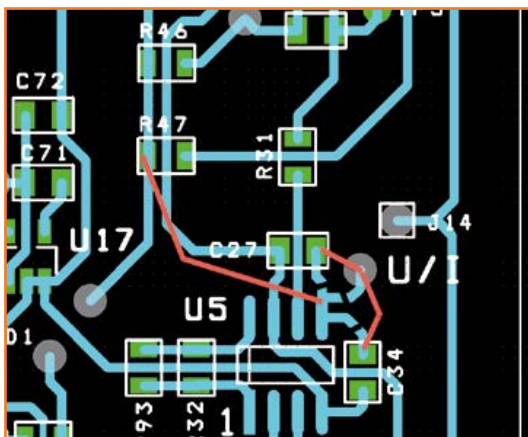


Figure 1.  
Schéma partiel avec les corrections à faire pour améliorer le LCR-mètre.



Figure 2.  
Modification du circuit de compensation de la tension de décalage autour de U5.



- tance de 680  $\Omega$  déjà en place ;
- la troisième intervention est plus délicate : il faut isoler de ses connexions actuelles la broche 5 de U5 et la relier au point commun à R45, R46 et R47. C'est faisable en soulevant la broche 5, mais nous le **déconseillons, car il ne faut pas prendre le risque d'abimer U5**. Nous préconisons donc de

Figure 3.  
Ne cherchez pas à dessolder la broche 5 de U5, vous risquez le pire. Travaillez au scalpel pour isoler l'îlot du reste du circuit, puis rétablissez avec du fil les liaisons à modifier.

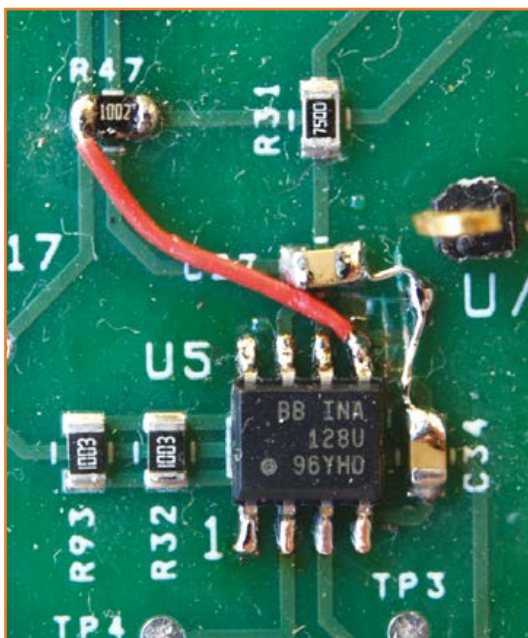
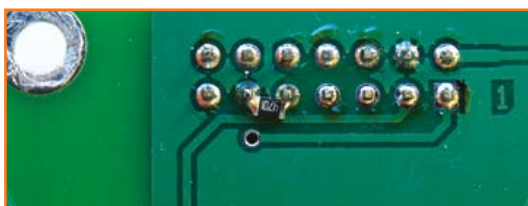


Figure 4.  
Cette résistance indique au *firmware* que le circuit a été modifié conformément aux indications des figures 1 à 3.



couper proprement les pistes (visibles) entre U5.5 et C27, U5.5 et C34 et U5.5 et le via à côté de J14 (**fig. 2**). Il faut ensuite rétablir les liaisons de C27 et C34 avec le via (masse analogique) par de petits fils (**fig. 3**) ; et établir par un fil court la nouvelle liaison entre la broche 5 de U5, et le point commun à R45, R46 et R47 ;

- souder une résistance de 4,7 k $\Omega$  à 10 k $\Omega$  entre les broches 9 et 11 de J17, sur la face opposée du circuit imprimé (**fig. 4**). La présence de cette résistance permettra au *firmware* de détecter le circuit modifié.

### Nouveaux programmes [4]

Le *firmware* version 3.0.0 (*LCR3A\_firmware\_V300.hex*) prend en charge le nouveau schéma de compensation de la tension de décalage d'entrée si, et seulement si, une résistance à la masse est détectée en broche 9 de J17.

Cela se traduit par un réarrangement des menus du programme *AU2011* qui passe lui aussi en version 3.0.0 : au menu initial *Réglage input\_offset...* se substituent deux menus *Réglage input\_offset\_U...* et *Réglage input\_offset\_I...*, le premier avec un court-circuit à l'entrée, le second avec un circuit ouvert. Pour les utilisateurs qui ne veulent (ou peuvent) pas effectuer la modification du matériel, la solution consiste à limiter le gain global en gamme 1. Dans la foulée, une limitation semblable peut être effectuée en gamme 8, si l'environnement est très perturbé et empêche d'effectuer correctement la compensation *TRIM - OPEN-CIRCUIT*.

### La solution de limitation du gain est valable que l'appareil ait été modifié ou pas.

La valeur maximum *initiale* sera de 5 pour un appareil non modifié et du maximum 15 (palier F) pour un appareil modifié. Les modifications de valeur sont bien entendu sauvegardées dans la mémoire de l'appareil, tout comme les autres réglages/options de la fenêtre des Préférences. Il en découle l'ajout de deux menus en mode autonome, et la modification de la fenêtre des préférences en mode PC quand l'option *menus de calibration accessibles* est cochée (**fig. 5 & 6**).

D'autres modifications ont été effectuées pour augmenter le confort d'utilisation, comme l'apparition d'un menu « **Port / Fermer le port** », pratique si un mauvais port a été sélectionné

initialement ; il suffit ensuite de sélectionner le bon port et de cliquer sur le bouton « Ouvrir le port » de la fenêtre principale.

### Mesure d'impédances élevées

Avant échantillonnage, les signaux « tension » et « courant » sont amplifiés sans aucun filtrage des basses fréquences. Ce n'est qu'après numérisation, en effectuant la mesure pendant un nombre entier de périodes du secteur et en effectuant la moyenne de plusieurs mesures, qu'il est possible de diminuer l'influence des signaux parasites captés par le dispositif de mesure.

Cela veut dire qu'il est possible que le signal appliqué au convertisseur analogique-numérique (ADC) dépasse, de courts instants, la plage d'entrée de cet ADC et invalide la mesure.

Il faut donc prendre un soin particulier pour la mesure des impédances élevées, quand le LCR-mètre se règle en gamme 7 et surtout 8 ; c'est le cas lors du *TRIM CIRCUIT-OUVERT*.

- Mettez l'électronique dans un boîtier métallique (le fer étant préférable à l'aluminium aux basses fréquences) réuni à la terre. Prenez garde si vous utilisez le LCR-mètre en mode autonome avec une alimentation USB, ou en mode PC avec un ordinateur portable : dans ces cas, la liaison à la terre n'est pas établie et c'est à vous le faire. En effet, la fiche secteur d'une alimentation USB n'a pas de broche de terre de protection qui permettrait de relier le boîtier du LCR-mètre à la terre du réseau électrique ; un ordinateur portable sur batterie n'est pas connecté à la terre non plus !
- Réduisez autant que possible la longueur des câbles de mesure et éloignez-vous des cordons secteur. Pour protéger l'appareil et les cordons de mesure des rayonnements du secteur, intercalez une plaque métallique (fer de préférence), reliée à la terre, de surface suffisante.
- Si la perturbation existe toujours, réduisez le gain de la chaîne de mesure en gamme 8 (*Max DACIndex I* en mode autonome).

Pour conclure, précisons que l'usage a montré que la solution du boîtier de mesure à 4 bornes BNC (TONGHUI TH26001A ou HAMEG HZ181) était nettement préférable à la pince Kelvin.

(130307)



Figure 5.  
Deux nouveaux menus en mode autonome.

### Liens

[1] LCR-mètre 0,05%

1ère partie Elektor n°417, mars 2013

[www.elektor.fr/110758](http://www.elektor.fr/110758)

2e partie Elektor n°418, avril 2013

[www.elektor.fr/130022](http://www.elektor.fr/130022)

3e partie Elektor n°419, mai 2013

[www.elektor.fr/130093](http://www.elektor.fr/130093)

[2] [www.elektor.fr/forumLCR](http://www.elektor.fr/forumLCR)

[3] [www.elektor.com/forum/elektor-forums/fields-of-interest/test-measurement.1543743.lynkx](http://www.elektor.com/forum/elektor-forums/fields-of-interest/test-measurement.1543743.lynkx)

[4] [www.elektor.fr/130093](http://www.elektor.fr/130093)

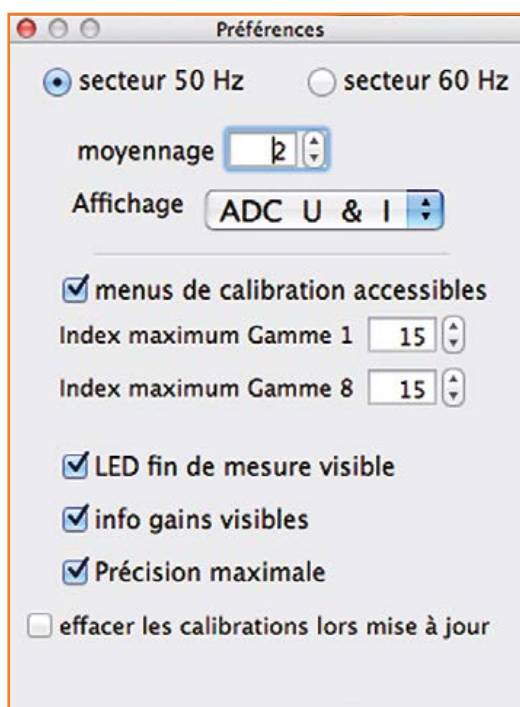


Figure 6.  
L'option *menus de calibrage accessibles* dans les préférences en mode PC.

# mettez le cap sur Arduino (7)



## station météo : à l'école d'Éole

L'équipe d'*Arduino Verkstad* (« atelier » en suédois) a été chargée de créer des supports d'initiation à l'électronique. Destinés à des lycéens, ces supports devaient être peu onéreux, mais néanmoins assez intéressants pour stimuler et impliquer des adolescents. L'équipe a ainsi fabriqué une station météo mobile pour des expériences à la croisée des sciences naturelles, de la physique et de la technologie.

**David Cuartielles,  
Clara Leivas,  
Tien Pham**  
(Arduino Verkstad, Malmö)

Nous voulions que les étudiants utilisent une station météorologique mobile pour collecter des données en ville, dans des parcs et le long de cours d'eau. L'idée était qu'ils puissent comparer des ensembles de données provenant d'environnements différents.

Nous avons décidé d'utiliser des ballons captifs : leur position est facile à contrôler, ils permettent de recueillir des données à des altitudes élevées, et il est possible d'y embarquer une caméra. Les images en accéléré obtenues ont d'ailleurs servi à visualiser les données recueillies et à les corrélérer avec les environnements de mesure.

### Les préparatifs

Nous avons réuni l'électronique nécessaire, à savoir un capteur d'humidité et de température, un microphone, une mini-caméra, et un capteur de CO<sub>2</sub>. À notre connaissance, aucun capteur anémométrique prêt à l'emploi ne convenait : nos relevés devaient être faits depuis un ballon, mais pour cela la plupart des anémomètres ont

besoin d'un support stable. Nous devions donc trouver une méthode adéquate. D'autres imprévus allaient vite compliquer ce qu'au départ nous pensions être un projet assez simple.

Pour réduire les coûts, nous avons choisi d'utiliser une grappe de ballons plutôt qu'un seul gros ballon-sonde. Nous avons acheté une bouteille d'hélium ainsi que des ballons gonflables ordinaires. Après quelques recherches en ligne, nous avons estimé qu'environ 25 ballons seraient nécessaires pour soulever l'Arduino, une carte d'extension, les capteurs, et un conteneur léger. Côté anémomètre, notre objectif n'était pas tant de mesurer des vitesses de vent précises que de pouvoir comparer entre elles différentes vitesses. Nous avons donc décidé de ne mesurer que des rotations par seconde à l'aide d'un capteur maison fait d'une photorésistance (LDR) et d'une LED. Nous pensions intercaler entre la LED et la LDR un disque de papier pourvu d'une fente. Avec la rotation des pales, la lumière aurait ainsi traversé le disque par intermittence et déclenché la LDR.



### Nuages en vue

Nous avons d'abord placé la LED très près de la LDR et espéré, sans trop y croire, que l'amplitude des valeurs relevées rendrait inutile la présence d'un écran. L'échec fut sans surprise, et nous nous sommes mis en quête d'une boîte noire pour loger l'électronique.

La présence de cette boîte souleva hélas des problèmes mécaniques d'assemblage. Nous avons donc opté pour un capteur infrarouge (IR) et un disque de papier noir et blanc : au moins aurions-nous moins de chance de rencontrer un problème d'ordre mécanique avec un seul composant électronique.

Cette fois-ci notre montage fonctionnait et nous sommes passés à la suite, plus délicate : comment capturer le vent pour faire tourner le disque ? Puisque nous voulions une station légère, nous ne pouvions pas profiter de son poids pour la stabiliser et ainsi disposer d'un point de référence stationnaire pour l'anémomètre.

Pour que n'importe quel étudiant puisse reproduire notre assemblage, nous avons là aussi choisi le papier et le ruban adhésif pour fabriquer une aube de turbine (**fig. 1**). Nous l'avons montée horizontalement afin qu'elle tourne quelle que soit la direction du vent. Nous avons groupé les ballons aux quatre angles, en espérant qu'ils restent stables et que la station ne tourne pas sur elle-même.

### Les épines de la rose des vents

Notre maquette en papier laissait supposer que notre approche était bonne. En tout cas le capteur IR et le disque de papier fonctionnaient, donc nous étions satisfaits. Restait à démontrer la faisabilité de l'ensemble et à le réaliser — nous n'avions que le capteur IR, le capteur de température/humidité et la mini-caméra.

Nous avons d'abord testé le capteur de température/humidité et la caméra dans un bâtiment de l'université voisine. Bonne surprise, nous n'avions pas besoin d'autant de ballons que prévu (**fig. 2**). L'essai terminé, nous avons chargé les résultats enregistrés sur la carte SD avec un large sourire. Nous allions pouvoir tester notre anémomètre de fortune.

Le port de Malmö est très venteux, et nous pensions qu'il ferait un excellent terrain d'essai. Malheureusement, dès que les portes de la fac se sont ouvertes, les ballons ont été aspirés, la longe s'est raidie, les ballons ont commencé à vriller et à emmêler leurs ficelles, et le vent était si fort



Figure 1. Une turbine réalisée à l'aide de papier et d'adhésif pour servir d'anémomètre. On y croit !

Figure 2. Nous pensions que 25 ballons seraient nécessaires pour soulever la station Arduino. La preuve qu'une douzaine suffisait





Figure 3. Notre cerf-volant de type *rokkaku*.

Figure 4. Le schéma de connexion pour aider les lycéens à câbler les composants de la station.

que l'ensemble se retrouva emporté à l'horizontale, dans une position où la longe menaçait de rompre.

C'est à ce moment-là que nous avons réalisé l'évidence : notre anémomètre ne pourrait jamais fonctionner car des ballons attachés ne seraient jamais verticaux, au mieux légèrement inclinés. Sans doute la conception traditionnelle des bal-

lons météo sans attache avait-elle influencé notre jugement, toujours est-il que la présence d'une longe changeait complètement la dynamique de la station.

Nous avons alors essayé d'autres conceptions, dont un montage vertical de la turbine. Le problème était de réussir à ce qu'elle s'aligne avec le vent. Nous avons tenté l'ajout d'une queue semblable à celle des girouettes, mais elle déplaçait le centre de gravité et l'unité oscillait dans le vent de façon imprévisible.

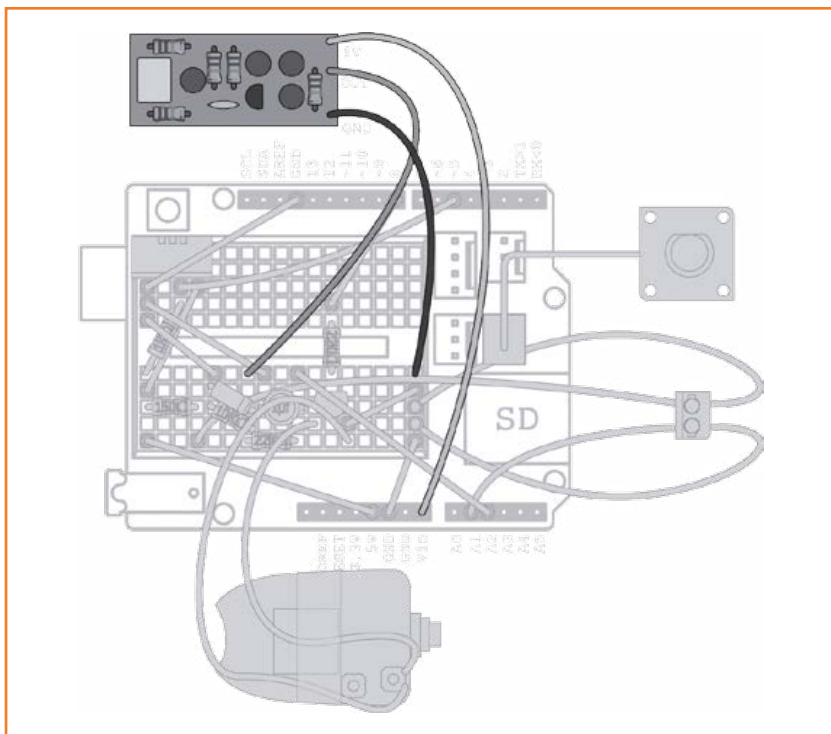
### Turbine bidirectionnelle

Nous avons finalement décidé de fabriquer par impression 3D une turbine capable de souffler dans les deux sens. Pour cela nous avons torsadé les pales le long de l'axe de rotation. Cette solution 3D représentait un compromis : nous n'avions plus assez de temps pour trouver mieux, mais au moins le coût d'impression serait raisonnable, car le centre de formation avec lequel nous collaborions possédait déjà une imprimante 3D.

David, qui était chargé de la partie mécanique, n'avait aucune expérience de l'impression 3D. Après quelques essais il sortit un prototype mal dégrossi mais utilisable.

Nous l'avons testé et constaté que l'anémomètre fonctionnait sous presque tous les angles. Tout allait bien donc. Nous avons alors conçu un support prévu pour être découpé au laser dans un panneau MDF (fibres de bois à densité moyenne). Il donnerait un montage propre et homogène, et permettrait aux capteurs, en particulier au capteur IR, de bien fonctionner (encore un compromis, mais la machine à découpe laser était commune, donc peu onéreuse, et la production de kits légers réduirait les frais d'envoi).

Nous avons déchanté lorsque nous sommes sortis faire un nouvel essai (sans les ballons) : les capteurs IR étaient d'une façon ou d'une autre perturbés par de nombreuses interférences. La seule solution trouvée fut de peindre en noir le boîtier en plastique. Ce boîtier devrait donc être imprimé avec du plastique noir uniquement, puisque les autres couleurs provoquaient des interférences. Le capteur IR se montra aussi capricieux que la LDR. L'éloigner ou le rapprocher du disque de quelques millimètres suffisait à empêcher une lecture correcte des valeurs. Ce comportement était dû à une plage de tolérance étroite, mais finalement nous avons réussi à le faire fonctionner



correctement.

### Des ballons pour une autre fête

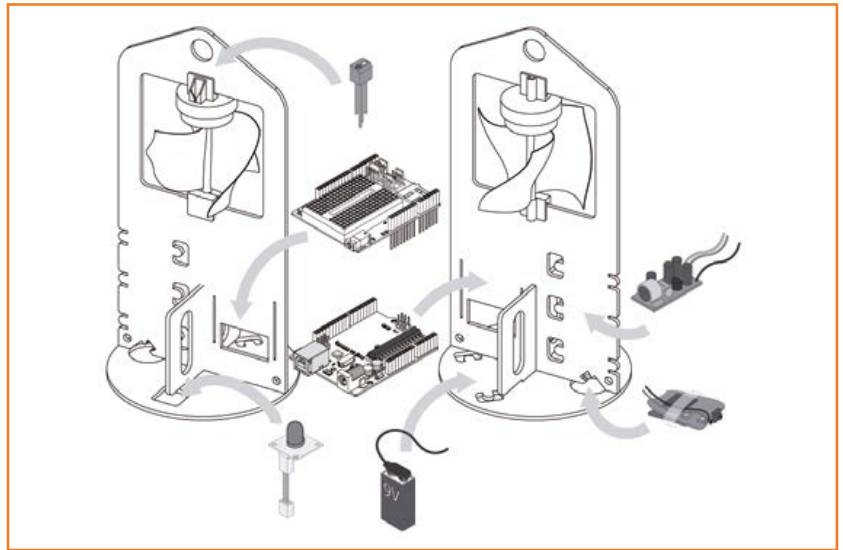
Nous avons trouvé les imprimantes 3D formidables pour le prototypage, mais nous avons aussi regretté leur manque de précision (notre inexpérience y est sans doute pour quelque chose). L'impression avait laissé des petites bosses, donc des points de friction qui se traduiraient par un besoin en vent accru. Encore un compromis à accepter.

Nous pensions toucher au but... avant que le problème des ballons ne nous revienne en mémoire. Quelqu'un a alors soulevé l'idée d'un cerf-volant. Pourquoi pas ?

L'un de nous raconta avoir vu un cerf-volant cellulaire doté d'une poussée suffisante pour soulever un homme. Impressionnant. Nous avons tenté d'en fabriquer un à l'aide de goujons et de sacs poubelle. Pas très esthétique, mais il vola. Enfin, il vola dix secondes avant de piquer du nez et de s'écraser. L'ajout d'ailes stabilisatrices n'y changea pas grand-chose. Le vent était imprévisible et finissait toujours par envoyer au sol notre modèle — des heures de travail... en l'air.

Nous avons alors choisi un modèle japonais appelé *rokkaku* (« à six côtés »), réputé être le plus stable des cerfs-volants guidés par une seule ficelle. Nous avons utilisé du bambou et un film plastique pour la voilure, et réussi à le faire voler (**fig. 3**). Une joie et une première, aucun de nous n'avait construit de cerf-volant jusqu'alors. C'est au moment où nous nous apprêtions à assembler l'ensemble pour effectuer un essai que nous avons réalisé que le comportement des constituants serait trop imprévisible par rapport à nos objectifs. Nouveau compromis donc, pour obtenir un engin moins sensible aux aléas de l'environnement nous avons accroché la station au bout d'un long bâton, à l'aide d'un crochet. Notre décision se révéla à la fois bonne et intéressante, car ce montage allait permettre aux étudiants de placer la station dans un arbre pour étudier l'effet du feuillage, ou encore de la suspendre au-dessus de l'eau depuis un pont.

Retour à la partie électronique. Nous avons ajouté le capteur sonore et il fonctionna tout de suite. Ça a été plus compliqué avec le capteur de CO<sub>2</sub>. Il fallait le réchauffer, et l'écriture du code source demandait beaucoup d'attention. Pour ces raisons il était difficile d'expliquer son fonctionnement aux élèves qui participaient à l'atelier. La



**figure 4** donne une idée de la façon dont tous les éléments étaient reliés à l'Arduino.

Nous n'avons pas toujours été satisfaits du déroulement du projet, mais au moins nous avons beaucoup appris. Les choses ont parfois tourné au désastre, et nous avons souvent choisi la mauvaise route, mais au final tout fonctionne. Parfois l'échec mène au succès.

L'assemblage complet de la station météo mobile est représenté en vue éclatée sur la **figure 5**. Vous pouvez télécharger gratuitement depuis le lien [1] d'autres dessins montrant l'électronique.

### Les kids de Zaragoza

Nous avons effectué des essais avec la station suspendue à des ballons, et aussi portée par un cerf-volant. Les photos prises sont trop mauvaises pour être reproduites ici. La **figure 6** montre la station accrochée à une perche et à un crochet. Les premières mesures pratiques ont été faites cet été au camp de découverte *Etopia Kids*, à Zaragoza, en Espagne. Les enfants ont profité durant 5 jours de différentes technologies interactives pour apprendre à compter en binaire, à programmer des cartes Arduino, à déplacer des robots, à lire et à représenter sur écran des données environnementales.

Les enfants ont d'abord assemblé la station, puis sont partis en randonnée recueillir des données. Température, hygrométrie, niveau sonore et vitesse du vent ont ainsi été mesurés régulièrement, tandis qu'étaient prises des vues aériennes des points de mesure. Une fois rentrés, les enfants utilisaient le programme

Figure 5. Assemblage final de la station météo. Température, humidité, altitude et niveau de CO<sub>2</sub> sont mesurés, traités par Arduino, puis écrits sur une carte SD. Le système prend également des vues aériennes. La turbine fabriquée par impression 3D est incorporée à la carte.



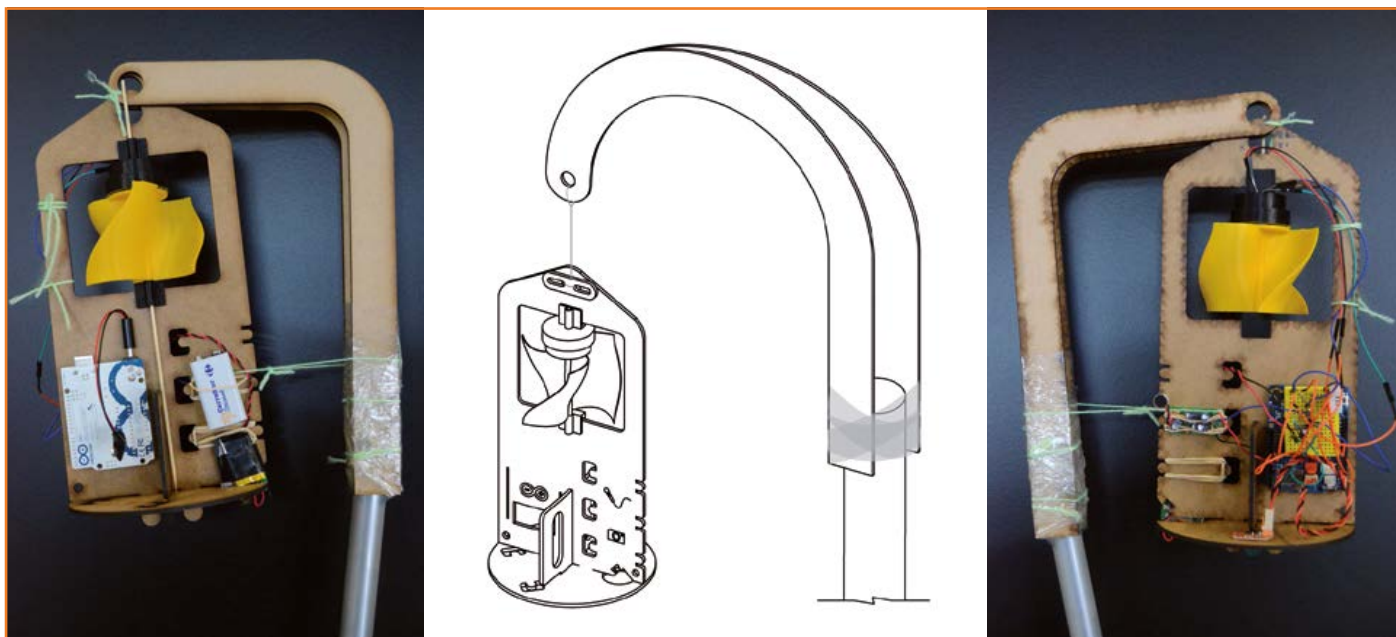


Figure 6. Après les échecs des ballons et des cerfs-volants, la station a finalement été accrochée à une perche. Voici ce que les formateurs avaient en tête et... la réalisation d'un des enfants. Les composants de la turbine ont été fabriqués en plusieurs couleurs.

*Processing* pour représenter graphiquement les données. Ils présentaient ensuite leurs résultats aux autres groupes et expliquaient ce que les mesures disaient de l'endroit où elles avaient été faites. Le croquis *Processing* du projet est disponible en [1].

### Leçons apprises

Au moment d'écrire cet article, nous avons mené l'expérience météo avec des enfants durant trois jours. Elle sera recommencée 15 fois sur une période de trois semaines, au bout desquelles 150 enfants auront joué avec la station météo. Nous avons remarqué que notre conception comportait beaucoup trop de fils. Pour que l'expérience reste un succès, voire s'étende, il serait pertinent de souder sur une carte les composants de l'actuelle conception Arduino. Autre point à revoir, la caméra. Si sa robustesse et la qualité des images étaient (sans surprise) au niveau du prix payé, nous avons dû ressouder plusieurs fois les fils qui la commandaient.

Nous allons poursuivre cette expérience avec le même enthousiasme. Si vous avez des idées d'améliorations, ou si vous connaissez des expériences semblables qui pourraient nous servir de modèles, vous pouvez nous écrire à [experiment.design@arduino.cc](mailto:experiment.design@arduino.cc).

### L'histoire en images

Les dessins techniques qui ont aidé les étudiants

à assembler la station météo mobile Arduino peuvent être téléchargés gratuitement [1]. Un groupe de dessins montre la connexion de l'électronique externe à la carte principale Arduino (*BalloonExperiment-xx.png*), l'autre l'assemblage complet de l'unité, y compris la perche à laquelle elle est reliée (*MDF\_WeatherStation\_Instructions-xx.png*).

(130043 – version française : Hervé Moreau)

[1] Croquis pour le programme *Processing* (fichier .zip) ; schémas électriques et mécaniques (fichier .zip) : [www.elektor.com/130043](http://www.elektor.com/130043) (gratuits).

### À propos d'Arduino Verkstad

Récemment formée, l'équipe d'*Arduino Verkstad* (« atelier » en suédois) compte des spécialistes de différents domaines, qui parfois se recoupent. Tous les membres n'ont pas été directement impliqués dans le projet, mais chacun a su conseiller Tien et Clara, les développeurs principaux.



Nous fêtons un look nouveau - de même que les 25 années à venir.



## 25% de réduction sur Circuit Cellar

Qu'il s'agisse de conseils pour la programmation ou d'application de conception, vous pouvez compter sur Circuit Cellar pour trouver des solutions à tous vos défis électroniques. Raspberry Pi, Linux embarqué, design de faible consommation, réduction de l'empreinte mémoire et bien plus encore ! Devenez membre, et voyez de vos propres yeux comment sont passées à la pierre de touche les toutes dernières nouveautés technologiques.

**Venez nous rejoindre aujourd'hui !**

[www.circuitcellar.com/sepN13](http://www.circuitcellar.com/sepN13)



Tessel Renzenbrink (elektor)

# lanceurs d'alerte, il vous faut une infrastructure !



Arturo Filastò.  
(Photo: Matteo G.P. Flora [5].  
CC license: BY-NC-SA 3.0)

Les dénonciateurs, divulgateurs ou lanceurs d'alerte comme on dit maintenant, s'exposent à de sévères poursuites puisqu'ils enfreignent sciemment les lois d'un pays faites justement pour en protéger les secrets ; on connaît le cas de Bradley Manning qui, motivé selon ses déclarations par la volonté de déclencher un débat public autour des guerres américaines en Iran et Afghanistan, a rendu publics des documents militaires et diplomatiques secrets. Maintenu dans l'isolement neuf mois durant avant son procès, reconnu coupable de 20 chefs d'accusation, il n'a été condamné finalement qu'à 35 ans de prison alors qu'il en risquait 90. Quelles que soient les connaissances et les opinions que l'on a de ces sujets et du droit en particulier, l'arithmétique absurde et la sévérité de telles peines ne manquent pas de frapper.

Dans ce contexte, la dénonciation ou le lancement d'alerte (*whistleblowing* en anglais) sont devenus un sujet phare de l'OHM2013, grand rassemblement de hackers d'Europe. J'y ai parlé avec des membres de plusieurs organisations qui ont lancé des initiatives pour une meilleure

Les divulgations d'Edward Snowden ont mis sous les projecteurs de l'actualité de vastes collectes secrètes de données sur l'internet par des services de renseignement. La légitimité de ces programmes de surveillance à grande échelle est discutée en raison notamment de leur manque de transparence. Autrement dit : Faut-il surveiller la surveillance ? Comment fonctionne la démocratie quand les électeurs sont maintenus dans l'ignorance ? Quelle est la portée de la dénonciation de ces pratiques par de simples citoyens ? Les lanceurs d'alerte ont-ils un rôle clé à jouer dans le processus démocratique ?

infrastructure du lancement d'alerte. *Globaleaks* (≈ fuites globales) fournit l'infrastructure technique permettant à n'importe qui de mettre en place sa propre plateforme. *L'Associated Whistleblowing Press* est une plateforme de lancement d'alerte décentralisée et *l'International Modern Media Institute* prêche pour une meilleure protection légale universelle des lanceurs d'alerte.

## International Modern Media Institute

*L'International Modern Media Institute* veut protéger la liberté de la presse en militant pour l'adoption d'un cadre légal protecteur [1]. Smári McCarthy, son directeur, nous dit : « l'IMMI travaille à la création d'un cadre légal qui protégerait lanceurs d'alerte et journalistes. Nous usons pour cela d'un lobbying classique : écriture de rapports, communication avec les parlementaires et sensibilisation du public. Nous avons avancé là-dessus en Islande et un premier brouillon d'une loi de protection y fait son chemin dans les ministères. C'est un travail de longue haleine, comme toujours en politique. Nous essayons de relever le niveau de protection dans les législations sur l'information à tra-



vers le monde. Nous discutons avec les parties de différentes juridictions telles que l'Europe ou celles des pays d'Afrique du Nord. Les avancées obtenues en Islande serviront de modèle : il est plus facile de propager une idée à partir d'une situation exemplaire. »

**Tessel :** *Quels sont les problèmes légaux liés au lancement d'alerte ?*

McCarthy : « Ils sont nombreux. La plupart des pays n'ont aucune loi pour protéger les lanceurs d'alerte. Certains pays disposent d'un niveau de protection rudimentaire, mais leurs gouvernements n'hésitent pas à s'en affranchir lorsqu'ils en ressentent le besoin. Comme le montre p. ex. la condamnation de Bradley Manning en vertu de la loi sur l'espionnage.

Nous avons également été témoins de tentatives de museler la presse. C'est le cas du journaliste Barrett Brown p. ex. qui, pour ses recherches sur l'état de la sécurité nationale étatsunienne, a utilisé des données récupérées dans les entreprises privées de sécurité *Stratfor* et *HBGary* piratées par les *Anonymous* et publiées en ligne. En prison dans son pays où l'attend un procès, il encourt le risque d'une condamnation à 105 années de prison, alors qu'il ne s'est introduit dans aucun système et qu'il n'a utilisé qu'une copie de données en ligne. »

### Fuites générales

Écoutons Arturo Filastò, de l'équipe de *Globaleaks* [2] : « *Globaleaks* est un logiciel conçu pour permettre à n'importe qui de mettre en place facilement un site de lancement d'alerte. N'importe qui peut télécharger son code à source(s) ouverte(s), l'installer et mettre en place son site. Nous fournissons l'infrastructure technique, mais ne sommes pas nous-mêmes une plateforme de lancement d'alerte. La technique n'est qu'un des aspects. Il faut aussi militer, étudier les documents divulgués et collaborer avec une plateforme de publication existante ou en créer une. Nous contribuons à cet écosystème en permettant à d'autres de réussir leurs projets. »

L'aspect visible du logiciel est une interface facile d'utilisation pour qui souhaite divulguer des informations. Il suffit de quelques clics pour envoyer des documents anonymement et de manière sécurisée à un ou plusieurs destinataires de son choix, des journalistes ou des organisations de droits de l'homme, à condition qu'ils se soient eux-mêmes rendus accessibles sur la plateforme. Lorsque le divulgateur choisit d'adresser son envoi

à un destinataire particulier, celui-ci reçoit les documents par messagerie électronique.

**Tessel :** *Quelles sont les mesures de sécurité offertes par Globaleaks pour protéger les protagonistes ?*



Pedro Noel.

(Photo: Matteo G.P. Flora.  
CC license: BY-NC-SA 3.0).

**Filastò :** « Pour anonymiser le trafic par l'internet, *Globaleaks* utilise *Tor*. » [Un logiciel libre et un réseau ouvert grâce auquel ni le récepteur ni quiconque interceptant les paquets ne peut identifier l'émetteur. Le trafic passe par une série de connexions chiffrées sur un réseau de volontaires.] « Il utilise une session *Tor* cachée afin d'assurer l'anonymat du lanceur d'alerte ainsi que celui de la personne ou l'organisation qui entretient le serveur. L'emplacement du serveur est inconnu et celui-ci ne peut donc pas être saisi. Les destinataires ne devraient pas être anonymes : personne ne va divulguer des informations sans savoir où elles vont.

L'envoi du fichier, lui-même chiffré, est effectué avec PGP [*Pretty Good Privacy*, un programme utilisé pour chiffrer la messagerie électronique]. »

**Tessel :** *Quand avez-vous commencé ce projet ?*

« Après l'affaire des câbles diplomatiques publiés par *Wikileaks*. » [En 2010, *Wikileaks* a commencé à publier au compte-goutte des câbles diplomatiques américains. Ces documents ont été modifiés pour supprimer les informations sensibles



Smári McCarthy.  
(Photo: SHAREconference.  
CC license: BY-SA 2.0).

comme les noms. Suite à une faille de sécurité, la base de données tout entière a été mise en ligne.] « Après ça, beaucoup de sites de divulgation ont germé, avec une sécurité souvent douteuse ; beaucoup ont échoué. » La fragile sécurité de *SafeHouse*, la boîte de dépôt du *Wall Street Journal*, a p. ex. été révélée quelques heures seulement après sa mise en ligne. Filastò poursuit : « Il y avait une demande, mais les développeurs s'y prenaient mal. On s'est dit : nous sommes experts en sécurité, faisons mieux. Il y a deux ans, nous avons un prototype avancé (*Globaleaks 0.1*), nous en sommes à la version 2.24. *Globaleaks* fait partie de l'organisation à but non lucratif *Hermes*, pour la transparence et les droits de l'homme [3]. »

### Associated Whistleblowing Press

Le logiciel *Globaleaks* est utilisé p. ex. par la plateforme de lancement d'alerte *Associated Whistleblowing Press* [4]. Son cofondateur, Pedro Noel, activiste des médias de l'internet, explique : « AWP est une organisation à but non lucratif qui se bat, à l'aide du lancement d'alerte, pour la liberté d'expression et contre les violations des droits de l'homme. Notre but est la révélation de données confidentielles prouvant des méfaits : nous pensons que la transparence est la meilleure arme contre la corruption et l'injustice. AWP est situé en Belgique, un des rares pays à fournir un cadre légal pour la protection des sources, même si elle n'est pas totale : en effet, si le fait de connaître la source est considéré par les autorités belges comme une question de sécurité nationale et qu'il n'y a pas d'autre moyen de trouver la source, le gouvernement belge peut nous obliger à la révéler. C'est pour-

quoi nous avons mis en place des mécanismes qui nous permettent de ne pas connaître l'identité de la source. Nous demandons aux sources de ne pas révéler leur identité et nous utilisons des logiciels tels que *Tor* ou *Globaleaks* pour nos communications sécurisées.

Notre plateforme est décentralisée, elle travaille avec des antennes locales qui possèdent leurs propres sites de lancement d'alerte, autonomes dans leur organisation et leurs décisions. Si l'on souhaite que les individus participent, il faut le faire au niveau local. Si quelqu'un possède des informations à propos du déversement de déchets toxiques dans une rivière et qu'il les envoie à un site d'envergure mondiale, leurs chances d'être utiles sont faibles. C'est au niveau local que leur importance est primordiale. Les plateformes centralisées grossissent mal. Les personnes qui s'en occupent sont submergées par le volume d'informations ou harassées par les pressions politiques et le stress. Le seul aspect centralisé d'AWP est sa structure juridique qui offre aux antennes un chapeau international. »

Tessel : *Peut-on considérer AWP comme un organe de presse ?*

« AWP travaille aussi à l'analyse des données envoyées, afin de les partager localement dans une lettre d'information. Nous voulons créer une agence mondiale de dépêches qui publierait des nouvelles impartiales, basées sur un journalisme scientifique. Il est important de disposer de nos propres canaux de diffusion : s'appuyer sur les médias grand public serait problématique, puisque ces entreprises à la recherche de bénéfices ménagent la chèvre et le chou, du point de vue économique et politique. Des coopérations ponctuelles permettent cependant d'augmenter l'impact.

Comme c'est la rédaction qui porte les responsabilités, la structure juridique centralisée d'AWP fournit une protection aux journalistes. Ce sont les rédacteurs en chef qui sont responsables de l'article écrit par un journaliste. Ils sont protégés par le cadre légal belge. Tout du moins, nous l'espérons. »

(130192 – version française : Kévin PETIT)

### Liens

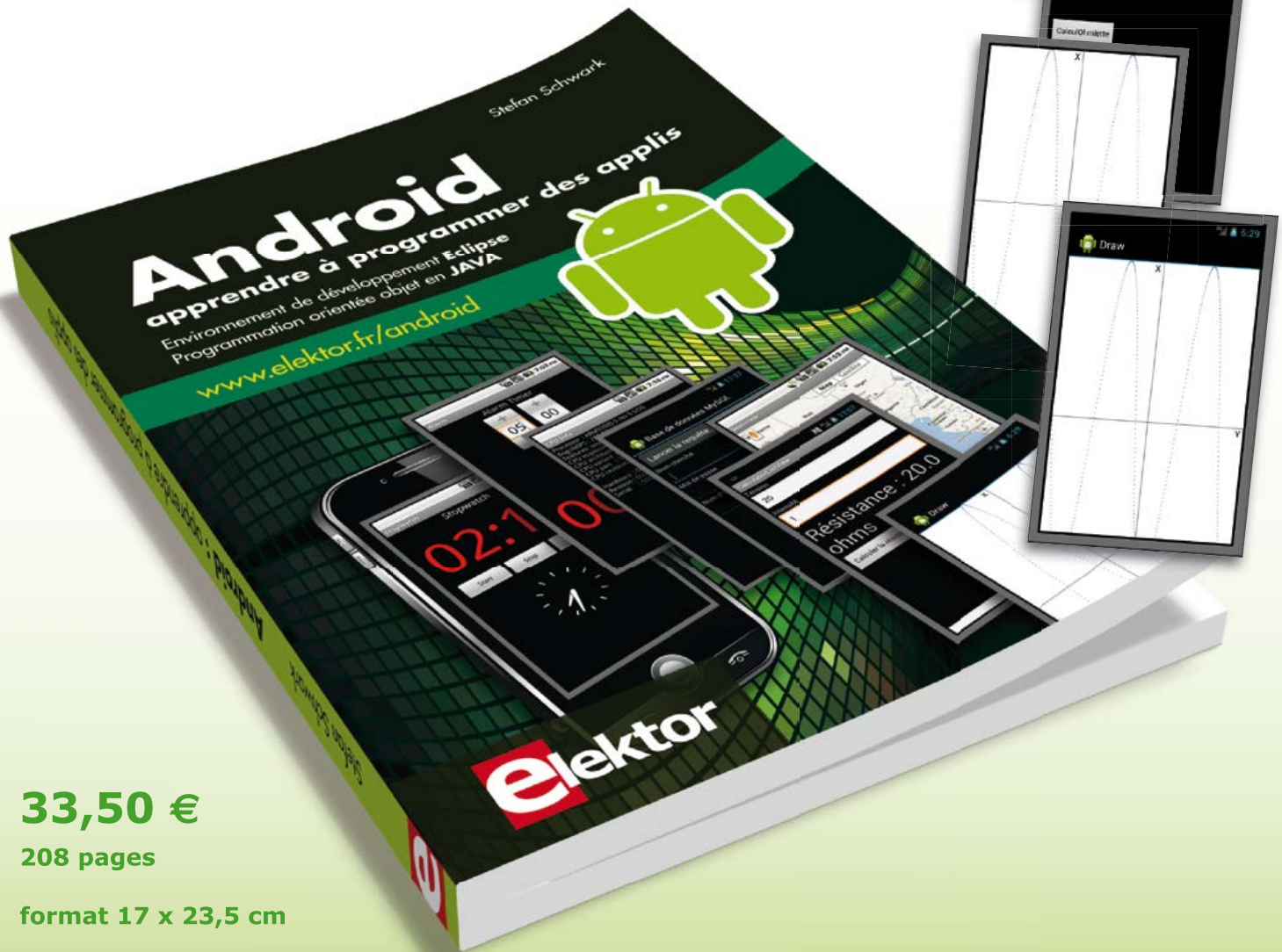
- [1] <https://immi.is/>
- [2] <https://globaleaks.org/>
- [3] <http://logioshermes.org/>
- [4] <http://awp.is>
- [5] <http://mgpf.it>



# Télécharger des applications existantes n'est pas le seul moyen de **rendre votre téléphone tactile encore plus intelligent !**

Voici un nouveau livre, publié par Elektor, pour apprendre à construire des applications sous Android et à tirer le meilleur de ce système d'exploitation.

C'est un ouvrage d'initiation, avec des exemples simples, variés et concrets, qui montre de façon progressive comment la combinaison de briques de code permet de créer toutes sortes d'applis dans l'environnement de développement Eclipse :  
calculatrice simple, interrogation des capteurs, exploitation des données GPS, communication par l'internet etc.



**33,50 €**

208 pages

format 17 x 23,5 cm

ISBN 978-2-86661-187-3

**elektor**

[www.elektor.fr/android](http://www.elektor.fr/android)



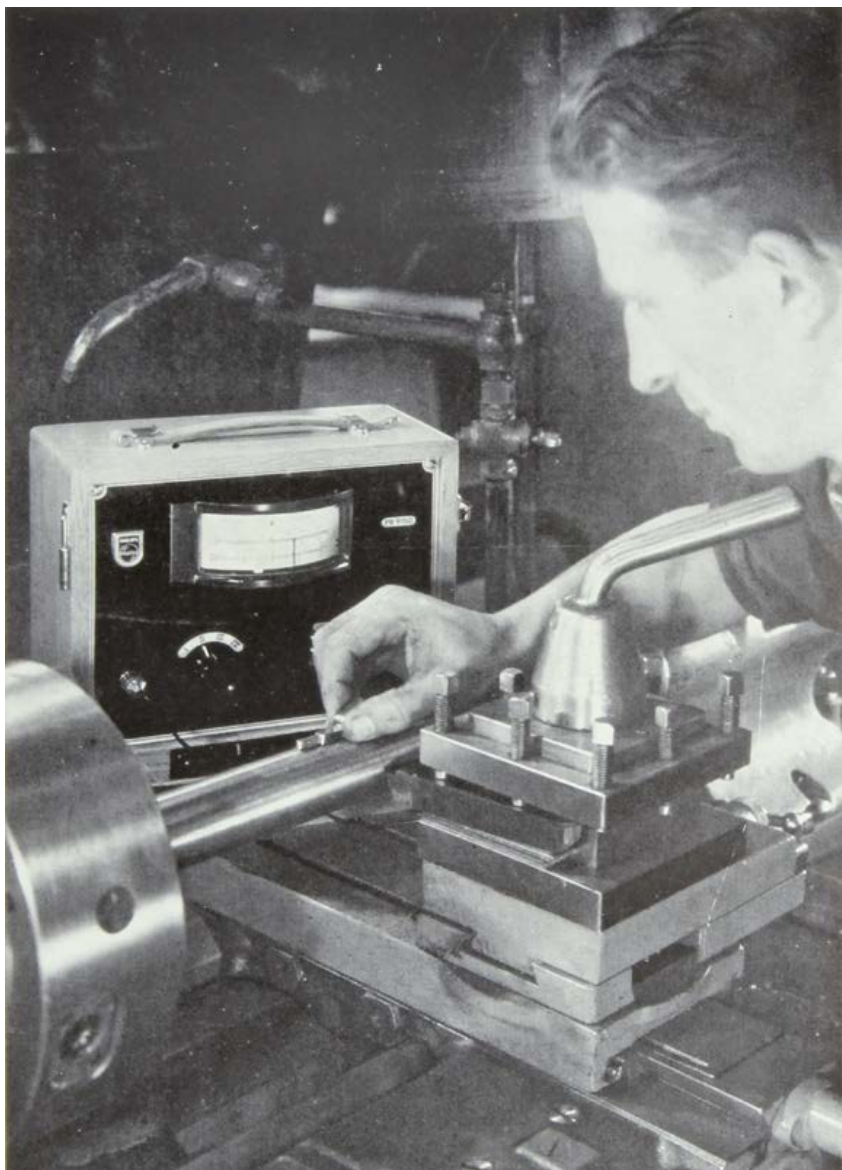
# rugosimètres Philips PR9150/PR9151

## Grattons un peu la surface des métaux comme dans les années 50

**Jan Buiting** (Elektor)

La plupart des électroniciens de ma connaissance détestent l'usinage et le travail mécanique en général, et la plupart des accros au travail du métal de ma connaissance ont une piètre opinion de l'électronique. Historiquement, ce sont la conception et l'usinage assistés par ordinateur

(connus par les acronymes anglais de *Computer Assisted Design* et *Computer Assisted Machining*, CAD-CAM) qui ont réuni les deux domaines. Récemment, les électroniciens et les mécaniciens sont devenus amis dans le domaine « chaud » de l'impression 3D.



Au milieu des années cinquante, la compagnie *Philips Pays-Bas* était renommée pour la qualité exceptionnelle de tous les assemblages et composants mécaniques de leur matériel de test, mesure et communication. Cela devait résulter en grande partie des fondations scientifiques posées par les laboratoires renommés *Physics Labs* de *Philips*, où quelques-uns des physiciens et ingénieurs mécaniciens les plus brillants jouissaient de la liberté totale de perfectionner leur art – et sans difficulté de financement.

Toujours au milieu des années 50, *Philips* était un acteur puissant dans le domaine du matériel de test et mesure para-scientifique. Leur série d'appareils « PR » était électronique, bien sûr, mais pas seulement électronique – utilisée qu'elle était dans les instituts de recherche et les industries chimique, médicale et mécanique : testeurs d'acidité et mesureurs de fluides en matière de conductivité, pression acoustique, mégavolts, toutes sortes de gaz, vibration, rayons X...

En 1952, G. W. van Santen, employé des *Physics Labs* de *Philips*, déplorait le fouillis de normes des « mécanos » qui parlaient de la rugosité de surface (RS) des produits métalliques finis, sans unités compréhensibles par chacun. On trouvait aussi bien des appareils extrêmement coûteux manipulés par quelques scientifiques privilégiés que de vieux praticiens du travail du métal qui, les yeux fermés, se servaient de leurs ongles pour jauger le toucher lisse après usinage, affûtage et polissage de leurs pièces. Il est remarquable que beaucoup d'entre nous perçoivent la rugosité jusqu'à 40  $\mu\text{m}$  environ, à peu près l'épaisseur d'un cheveu humain.



## Rugosité : définition et confusion

Ce qui suit est un film d'horreur sur les normes, façon Tour de Babel, qui nous emmène en 1952. Le **figure 1** montre le profil d'une surface usinée. Le dessin est dilaté verticalement par rapport à la longueur de la pièce usinée. Pour que la rugosité soit notée sans équivoque, il faudrait que la quantification soit normalisée internationalement. Malheureusement, ce n'est pas encore le cas. La rugosité est définie différemment suivant les pays :

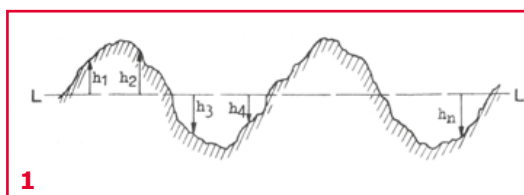
- **la hauteur de rugosité maximale**,  $H_{\max}$ , c-à-d la différence entre la pointe la plus élevée et le creux le plus profond. Cela forma la base de la norme allemande DIN 4762. La hauteur  $H_{\max}$  est exprimée en microns.
- **la hauteur de rugosité effective**,  $h_{\text{eff}}$  ou  $h_{\text{rms}}$ , c-à-d la moyenne quadratique (racine carrée de la moyenne des carrés) des valeurs absolues des écarts  $h$  par rapport à la ligne de référence (L, niveau moyen), sur une longueur donnée :

$$h_{\text{eff}} = \sqrt{\frac{h_1^2 + h_2^2 + h_3^2 + \dots + h_n^2}{n}}$$

la norme américaine ASA B46 se réfère à  $h_{\text{rms}}$  d'un bout à l'autre, avec les valeurs en micro-pouces.

- **hauteur de rugosité moyenne**,  $h_{\text{avg}}$ , c-à-d moyenne arithmétique des valeurs absolues des écarts par rapport à la ligne de base, mesurés sur une longueur définie :

$$h_{\text{avg}} = \frac{h_1 + h_2 + h_3 + \dots + h_n}{n}$$



La norme britannique BS 1134 définit la hauteur de rugosité moyenne (*center-line average*, C.L.A.) comme la norme, avec les valeurs exprimées en micro-pouces, comme avec  $h_{\text{rms}}$ .

La commission néerlandaise de normalisation a recommandé aussi la hauteur de rugosité moyenne comme norme, arguant que (a) la définition de  $h_{\text{avg}}$  ne suscite pas de discussion, (b) le terme de valeur moyenne est plus facile à expliquer dans un atelier et (c) la grandeur peut être mesurée directement avec des moyens électroniques. De plus, la différence entre  $h_{\text{avg}}$  et  $h_{\text{eff}}$  est négligeable dans la pratique.

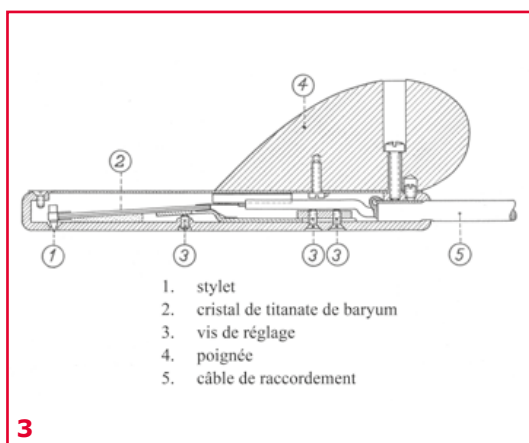
## Unités de rugosité

Pour des surfaces (métalliques) lisses, le micron (millième de millimètre,  $10^{-6}$  m) est une unité un peu grossière, susceptible d'embêter les utilisateurs avec de tout petits nombres. Le milli-micron, lui, est trop petit et il donne de grands nombres peu pratiques. Le micro-pouce est une bonne unité, entre grande et petite, puisque la rugosité de la plupart des surfaces finies se chiffrera entre 1 et 500 micro-pouces.

Par exemple :

surface lisse :  $0,05 \mu = 2 \mu'' = 50 \text{ m}\mu$

surface rugueuse :  $6,3 \mu = 250 \mu'' = 6300 \text{ m}\mu$



Hélas, dans des pays comme la Hollande, où le système métrique domine, il est peu probable que soit adoptée une norme basée sur les pouces (unité non métrique). Comme cote mal taillée, on a proposé l'unité de rugosité « ru », avec des ruses linguistiques : rugosité, *ruwheid*, *roughness unit* de nature à convaincre les utilisateurs anglophones, néerlandophones et francophones (chiasme pour les amateurs). La rugosité d'une surface est de 1 ru si la hauteur moyenne de rugosité  $R_{avg} = 1/40$  micron  $\approx 1$  micro-pouce. Ce problème évacué (avec un petit hic sur la récurrence persistante du pouce), des classes de rugosité de 1 à 6 ont été créées, avec un jeu de symboles que les tourneurs et fraiseurs étaient censés reconnaître sur les plans d'exécution. C'était une longue série de triangles creux et pleins associés à des cercles creux et pleins et je suppose qu'on pouvait les apprendre tout comme nous avons appris nos kilohms et milliampères.

### Modèles PR9150/PR9151

La seule différence entre les deux est l'utilisation de tubes américains dans le PR9151 (comme un 6X4 au lieu d'un AZ41).

Je n'ai aucun schéma des PR9150 et 9151 à vous présenter, mais cela ne m'a pas empêché de les ouvrir (**fig. 2**). Intérieur impeccable après soixante ans. Faisons une supposition éclairée. L'électronique se compose probablement d'un

Tableau 1. Versions du PR9150 / 9151			
Type	Échelle	Gammes de mesure	Échantillons de rugosité
PR9150/01 PR9151/01	Échelle /01 (fig. 5a) pour <ul style="list-style-type: none"> <li>comparaison ; rouge = accepté, vert = rejeté</li> <li>multiplication (avec le facteur indiqué, pour SR par rapport à la pièce de référence)</li> </ul>	1 à 300 ru, divisée en 4 sous-gammes	Pièce de référence ou PR9180/00 (Ra valeur de calibrage : 125 – 32 – 8 et 2 ru ; 3,2 – 0,8 – 0,2 et 0,05 $\mu$ )
PR9150/02 PR9151/02	Échelle /02 (fig. 5b) calibrée pour Ra en ru (=C.L.A. en $\mu''$ ); l'échelle montre les valeurs de SR recommandées en standard	I: 50 à 280 ru II: 10 à 70 ru III: 3 à 16 ru IV: 1 à 4 ru	PR9180/00 (voir ci-dessus)
PR9150/03 PR9151/03	Échelle /03 (fig. 5c) avec division décimale, calibrée pour Ra en ru.	I: 50 à 250 ru II: 10 à 60 ru III: 3 à 16 ru IV: 1 à 4 ru	PR9180/00 (voir ci-dessus)
PR9150/04 PR9151/04	Échelle /04 (fig. 5d) calibrée pour $R_t = H_{max}$ en $\mu$	I: 5 à 25 $\mu$ II: 1 à 6 $\mu$ III: 0,3 à 1,6 $\mu$ IV: 0,1 à 0,4 $\mu$	PR9180/02 (Ra valeur de calibrage: 12,5 – 3,2. – 0,8 et 0,2 $\mu$ )
PR9150/05 PR9151/05	Échelle /05 (fig. 5e), calibrée pour Ra en $\mu$	I: 1 à 6 $\mu$ II: 0,3 à 1,6 $\mu$ III: 0,1 à 0,4 $\mu$ IV: 0,02 à 0,1 $\mu$	PR9180/00 (voir ci-dessus)



atténuateur d'entrée réglable (pour les classes de rugosité de 2 à 5), d'un amplificateur, d'un redresseur et d'un galvanomètre à cadre mobile. Avec le coffret en bois de belle facture et sa courroie de transport en cuir et un couvercle dégonflable, le tout pèse environ six kilos.

### Sonde

Le clou de l'appareil est la sonde schématisée sur la **figure 3**. Elle contient un cristal piézoélectrique en titanate de baryum, fixé sur le boîtier de la sonde d'un côté et terminé par un stylet de l'autre. Le stylet est une aiguille en saphir synthétique avec une dureté de 2 000 VPN (*Vickers Pyramid Number*) et un rayon d'environ 60  $\mu\text{m}$  à la pointe. Cela permet de « palper » la rugosité de surfaces finies jusque dans les vallées les plus profondes (les pentes dépassent presque toujours 150 °). On voit et on sent facilement la pointe du stylet. Si les PR9150/9151 avaient un haut-parleur, on pourrait les utiliser pour jouer des disques de Joe Cocker ou Janis Joplin.

En pratique, la sonde est si petite qu'on peut l'utiliser pour mesurer dans des trous de diamètre aussi petit que 8 mm. Elle est connectée à l'appareil par une bonne longueur de câble blindé et une fiche massive de très bonne qualité. La sonde est munie (**fig. 4**) d'une élégante poignée en bakélite.

### Une échelle pour chaque norme

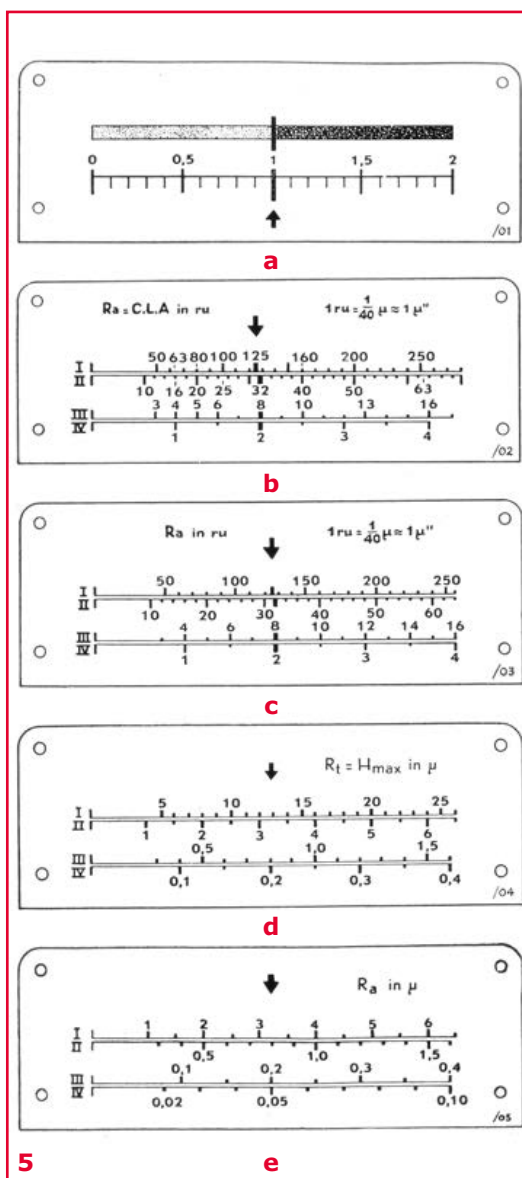
Apparemment Philips n'avait pas confiance dans un appareil « à taille unique » avec une échelle en ru comme le proposaient, probablement, de nombreux comités et bureaucrates. Clairement un exemple d'une industrie qui cherche à satisfaire « Washington et Bruxelles ». J'ai été étonné et intrigué en trouvant cinq types différents de PR9150/9151 cités dans le manuel du premier appareil que j'ai obtenu (tableau 1). Chaque version a sa propre échelle pour s'adapter à différents marchés, différentes préférences des chefs de service de contrôle, différentes applications, je suppose. La **figure 5a-d** est une tentative de reproduction des diverses échelles d'après mon unique exemplaire du manuel. Je possède des PR9151 en versions /01, /02 et /05, de même que trois sondes (dont une avec le stylet cassé), un manuel et une... (roulement de tambour) :

### Boîte de calibrage originale !

Alors que les PR9150 et 9151 ne sont pas des oiseaux rares, la boîte de calibrage mentionnée



4



5



exactes de rugosité sont écrites à la main. J'ai pu voir facilement le degré de finition des quatre échantillons de métal dans la boîte et sentir les sillons dus au fraisage sur les exemplaires les plus « grossiers ». La distinction entre les deux autres pièces demande un peu plus d'efforts. Toutes les pièces ont un plaquage de chrome mince qui n'affecte pas la rugosité de leur surface. Quelqu'un aurait-il une boîte PR9180/02 ?

### En pratique

Après avoir calibré votre instrument avec les échantillons de la boîte, vous êtes censé poser la sonde sur la surface à tester à la perpendiculaire de la direction de la finition. Vous pouvez déplacer la sonde de haut en bas deux ou trois fois par seconde sur une distance de 1,5 à 2 cm, ou faire tourner la pièce (p. ex. sur un tour ou une fraiseuse). La vitesse minimale de la sonde ou de la pièce est d'environ 4 cm/s. Le dessous de la sonde doit toucher la pièce par toute sa longueur. Bien que le PR9150/9151 ait un filtre passe-haut pour éliminer les effets de mouvements involontaires de la main, il faudra un peu de pratique pour éviter les changements brusques de déviation de la sonde. Et oui, le test est préjudiciable – le stylet laisse des rayures. Dans un chapitre plutôt long du manuel, Philips conclut que leurs appareils offrent une précision de quelque 20%, ce qui est « exceptionnel, si on considère que des rugosimètres extrêmement précis et coûteux atteignent une précision de 15%, du fait de l'hétérogénéité de la structure des surfaces finies ordinaires ».



6

**EST<sup>®</sup> 2004**

Rétronique est une rubrique mensuelle sur les pages glorieuses et jaunies de l'électronique, avec occasionnellement des montages de légende décrits dans Elektor.

Si vous avez des suggestions de sujets à traiter, merci de les télégraphier à [redaction@elektor.fr](mailto:redaction@elektor.fr)

dans le manuel en est un. J'ai pu en trouver une finalement, avec le troisième PR9151 qui m'est tombé sous la main, et qui était aussi le plus miteux.

Même si vous pouvez faire des mesures comparatives avec les PR9150/9151, les commerciaux, le PDG et le responsable de la communication exigeront d'avoir des chiffres absolus et des bancs d'essai ! C'est pourquoi vous devrez calibrer votre appareil avant de prendre des mesures sur des échantillons.

La boîte de calibrage PR9128/00 de la **figure 6** est faite de bakélite massive. Remarquez aussi les symboles en triangles pleins et creux qui repèrent quatre classes de rugosité de surface. Les valeurs

Aujourd'hui, on trouve de nombreux rugosimètres sur Ebay & Co. Ils utilisent des technologies largement améliorées par rapport à l'enregistrement des chocs avec un stylet piézoélectrique et un ou deux tubes, comme on le faisait en 1955. Pourtant, vous n'en verrez pas des masses dans les ateliers qui travaillent le métal alentour, ni dans l'alésage de cylindres ou le polissage de précision. Je suppose que la bonne vieille méthode de l'ongle du pouce, couplée au métier et à l'expérience, tient toujours la route en face des kilodollars d'électronique et de l'internet.

(130057 – version française : Jean-Paul Brodier)

# hexadoku casse-tête pour elektorniciens

Assez de schémas, de programmes, de soudage et de dessoudage ! Posez la pince coupante, repoussez le fer à souder. Faisons un peu de calcul mental maintenant, ça détend. Remplissez la grille selon les règles, envoyez-nous votre solution, et gagnez peut-être un bon-cadeau de 100 € offert par Elektor et Eurocircuits.

Après ça, vous ne vous sentirez que mieux pour reprendre vos cogitations électroniques.

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 ran-

gée, 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 envoyer la série de chiffres sur fond grisé.

## Participez et gagnez !

Nous tirons au sort l'une des réponses internationales correctes reçues dans les délais ; son auteur recevra un chèque-cadeau d'une valeur de **100 €** à valoir sur des circuits imprimés **elektorPCBservice (Eurocircuits)**.

Nous offrons en outre 3 chèques-cadeaux à valoir sur des **livres d'Elektor** d'une valeur de **50 €** chacun.

## Où envoyer ?

Envoyez votre réponse (les chiffres sur fond grisé) avec vos coordonnées par courriel, télécopie ou courrier avant le **1<sup>er</sup> novembre 2013** :

Elektor c/o Regus Roissy CDG – Le Dôme – 1, rue de La Haye  
BP 12910 – 95731 Roissy CDG

Courriel : [hexadoku@elektor.fr](mailto:hexadoku@elektor.fr) | [www.elektor.fr/hexadoku](http://www.elektor.fr/hexadoku)

## Les gagnants

La solution de la grille du numéro de juillet-août (421-422) est : **3B4CD**

Le gagnant des **100€** à valoir sur des circuits imprimés **Eurocircuits** est József Nagy (Hongrie)..

Les 3 chèques-cadeaux Elektor d'une valeur de 50 € chacun vont à :

Jacqueline Deletombe (France), Mary Chang (USA) et Olavi Parkka (Finlande)..

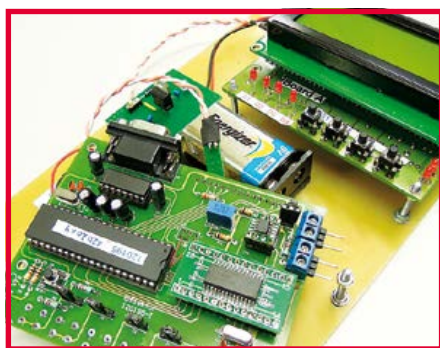
Bravo à tous et félicitations aux gagnants !

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

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

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.





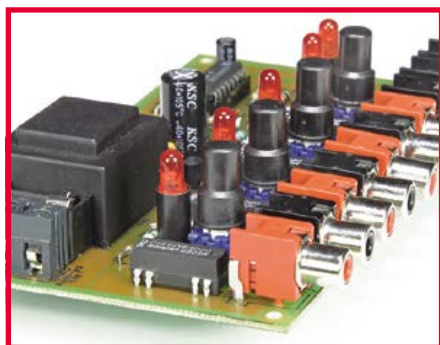
testeur CAN

Nos autos sont devenues de complexes réseaux ambulants. Diverses unités de commande et de régulation y communiquent par un réseau qui chez beaucoup de constructeurs est le fameux bus CAN. Sa réputation de robustesse n'est pas usurpée. Et pourtant, qui dit complexité dit fragilité. Et donc perturbations. Voici de quoi tester et évaluer.



enregistreur de températures multicanal avec afficheur 4x20 signes et clavier

Cet enregistreur à 6 voies, avec horloge en temps réel permet d'utiliser en même temps plusieurs capteurs de température numériques du genre DS18B20 pour enregistrer des relevés effectués à plusieurs endroits, p. ex. à des hauteurs différentes du sol, ou en différents points d'un appareil dont on veut surveiller le fonctionnement.



commutateur audio

Les sélecteurs de voie (ou d'entrée) des amplificateurs audio s'usent et ça s'entend, et ils sont souvent trop limités. Voici un commutateur audio à composants traversants courants (donc pas de CMS) qui permet de choisir une voie parmi quatre. Il est même possible de sélectionner deux voies en même temps sans les court-circuiter.

Informations préliminaires non contractuelles  
Parution du numéro de novembre : 23 octobre



## FACES AVANT ET BOÎTIERS

Pièces unitaires et petites séries à prix avantageux.

A l'aide de notre logiciel – Designer de Faces Avant \* – vous pouvez réaliser facilement votre face avant individuelle. GRATUIT: essayez-le! Pour plus de renseignements, n'hésitez pas à nous contacter, des interlocuteurs français attendent vos questions.

\*Vous en trouverez la dernière version sur notre site internet.

- Calcul des prix automatique
- Délai de livraison: entre 5 et 8 jours
- Si besoin est, service 24/24



Exemple de prix: 34,93€  
majoré de la TVA/  
des frais d'envoi

Schaeffer AG · Nehmitzer Damm 32 · D-12277 Berlin · Tel +49 (0)30 805 86 95-0  
Fax +49 (0)30 805 86 95-33 · Web info@schaeffer-ag.de · www.schaeffer-ag.de

## ECD7

NOUVELLE EDITION

### Base de composants d'ELEKTOR

Cet ensemble consiste en une quadruple banque de données (circuits intégrés, transistors, diodes et optocoupleurs) complétée par neuf applications satellites, au nombre desquelles on trouvera notamment de quoi calculer la valeur de la résistance associée à une diode zener, à un régulateur, à un diviseur, ou un multivibrateur astable, mais aussi le code de couleur de la résistance et de l'inductance. Avec ce CD-ROM, vous disposez donc de données fiables sur plus de 7.800 circuits entiers ; plus de 35.600 transistors, FET, thyristors et triacs ; environ 25.000 diodes et plus de 1.800 optocoupleurs. Le clou, c'est que vous allez pouvoir rajouter dans la base de données ce qui y manque encore, car elle est interactive ! Ainsi chaque utilisateur pourra lui-même rajouter des composants, en modifier les caractéristiques déjà enregistrées ou les compléter.



ISBN 978-90-5381-298-3 • 29,50 €

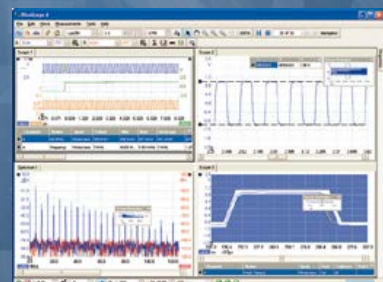
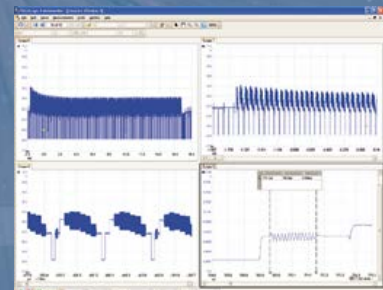
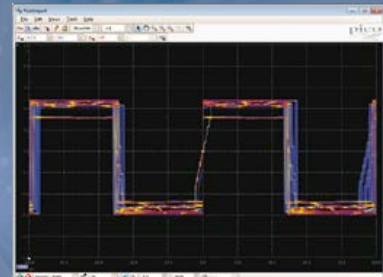
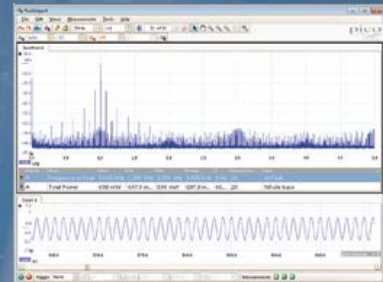


Pour commander en ligne :

[www.elektor.fr/ecd7](http://www.elektor.fr/ecd7)

# PicoScope<sup>®</sup> 6000 Série

- Oscilloscopes PC hautes performances
- 500 MHz avec 5 GS/s
- Mémoire tampon de premier plan pouvant accueillir 2 G d'échantillons
- Haute performance fonction / générateur de signaux arbitraires
- SuperSpeed USB 3.0 interface



PicoScope	6402C	6402D	6403C	6403D	6404C	6404D
Canaux	4					
Bande passante	250 MHz		350 MHz		500 MHz	
Taux d'échantillonnage	5 GS/s					
Générateur de signal	FG	AWG	FG	AWG	FG	AWG
Mémoire	256 MS	512 MS	512 MS	1 GS	1 GS	2 GS
Prix	£1995	£2495	£2995	£3495	£3995	£4495
Compatibilité	USB 2.0 & 3.0, Windows XP, Windows 7 & Windows 8					

TOUS LES MODÈLES INCLUENT SONDAS, LOGICIEL COMPLET, AINSI QUE 5 ANS DE GARANTIE. LE LOGICIEL COMPREND DES: MESURES, ANALYSEUR DE SPECTRE, FULL SDK, DÉCLENCHEMENTS AVANCÉS, PERSISTANCE DE COULEURS, DÉCODAGE SÉRIE (CAN, LIN, RS232, I<sup>2</sup>C, I<sup>2</sup>S, SPI), MASQUES, MATHS, LE TOUT EN STANDARD. MISES À JOUR GRATUITES.

[www.picotech.com/PS250](http://www.picotech.com/PS250)



Votre partenaire en  
ligne compétent pour

Eléments de construction  
Alimentation électrique  
Technique de mesure

Technique atelier & de brasage  
Technique domotique & sécurité  
Technique réseau

Technique PC  
Technique Sat/TV  
Communication

COMMANDER  
CONFORTABLEMENT !

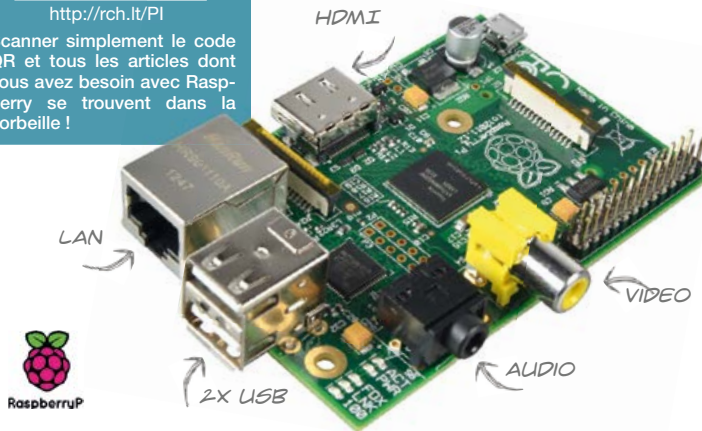
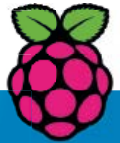


<http://r.ch.it/PI>

Scanner simplement le code  
QR et tous les articles dont  
vous avez besoin avec Rasp-  
berry se trouvent dans la  
corbeille !

## « Raspberry Pi » -Mini-PC

Réalisez votre projet !



### Raspberry Pi — Modèle B avec 512 Mo RAM

- Broadcom BCM2835
- 700 MHz ARM, Dual Core
- Open GL ES 2.0, OpenVG
- Prise Ethernet 10/100 BaseT
- Prise vidéo Composite HDMI-/RCA
- Slot carte SD
- 2x USB 2.0

€ **38.95**

RASPERRY PI B

BE INVENTIVE!

Raspberry Pi – une platine grande  
comme une carte de visite avec  
des possibilités infinies :

Vous pouvez après les première  
installation et configuration par  
exemple regrouper à la maison  
tous les fichiers médias puis utili-  
ser le Raspberry Pi comme lecteur  
HD dans votre salon dans la sphère  
dédiée aux divertissements. Ou bien  
vous utilisez le Raspberry Pi comme  
centrale de commande pour les ser-  
vices réseau.

Qu'il serve d'accès à des fichiers et  
à une imprimante, d'interface impi-  
mante réseau, d'impression AirPrint  
sans fil ou de haut-parleur sans  
câble via AirPlay – avec Raspberry  
Pi tout est pratiquement possible.

### Boîtier « Raspberry Pi »

- Exactement coordonné  
aux modèles A et B
- Snap-On-System
- Matériau ignifuge
- Fenêtre d'affichage de statut
- Trous d'aération



TEK-BERRY	blanc	€ 4.95
TEK-BERRY SW	noir	€ 4.95
TEK-BERRY TR	transparent	€ 4.95

adaptateur VESA (50/75/100) :		
TEK-BERRY VESA	transparent	€ 5.50

Fourniture sans Raspberry Pi, autres coloris en ligne !

### Raspberry Pi OS

- Système d'exploitation Linux  
pour Raspberry Pi A/B
- Pré-installé sur carte SD (4 Go)

RASPERRY PI OS

€ 18.50

### Adaptateur WLAN-USB, 150 Mbit/s

- Supporte QoS-WMM- & mode WMM-Power Save
- Puce : RTL8188CUS
- Supporte WEP, WPA, WPA2
- Compatible WPS

EDIMAX EW-7811UN

€ 8.95

EDIMAX



### Kit de refroidissement à 4 éléments

- Compatible avec Rev. 1 + 2
- Profilé très bas
- Montage aisé

TEK-BERRY COOL € 4.95

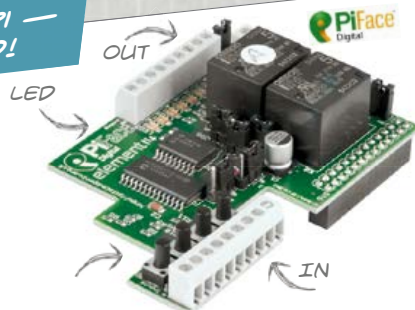
YOUR RASPBERRY PI —  
CONNECTED TO THE WORLD!

### PiFace Digital L'extension E/A pour le Raspberry Pi.

- Socle GPIO • 2 relais • 4 commutateurs
- 8 entrées numériques • 8 sorties
- Affichages 8 LED

RASPERRY PIFACE

€ 35.95



EXTENSION TOP :

### Caméra HD pour le Raspberry Pi

- 5 mégapixels
- Photo : 2592 x 1944 pixels
- Vidéo : 1080p (Full HD),  
jusqu'à 30 clichés/s
- Connexion :  
câble plat 15 pôles  
à l'interface série de  
caméra MIPI



RASPERRY PI CAM

€ 29.95

Commander maintenant ! [www.reichelt.fr](http://www.reichelt.fr)

Assistance téléphonique en anglais : +49 (0)4422 955-360

Prix du jour ! Prix à la date du : 12.08.2013

Pour les consommateurs : Les règles relatives à l'annulation s'appliquent. Tous les prix sont indiqués en € y compris la TVA, de l'entrepôt de Sande/Allemagne, et en plus des frais d'envoi pour l'ensemble du panier de produits. Nos conditions générales de vente s'appliquent de manière exclusive (sur [www.reichelt.com/agb](http://www.reichelt.com/agb)). Vente intermédiaire réservée. Tous les noms de produits et les logos sont la propriété des fabricants respectifs. Il en est de même pour les illustrations. Fautes d'impression, erreurs et modifications de prix réservées. reichelt elektronik GmbH & Co. KG, Elektronikring 1, 26452 Sande/Allemagne (HRA 200654 Oldenburg)

Modes de paiement internationaux :

