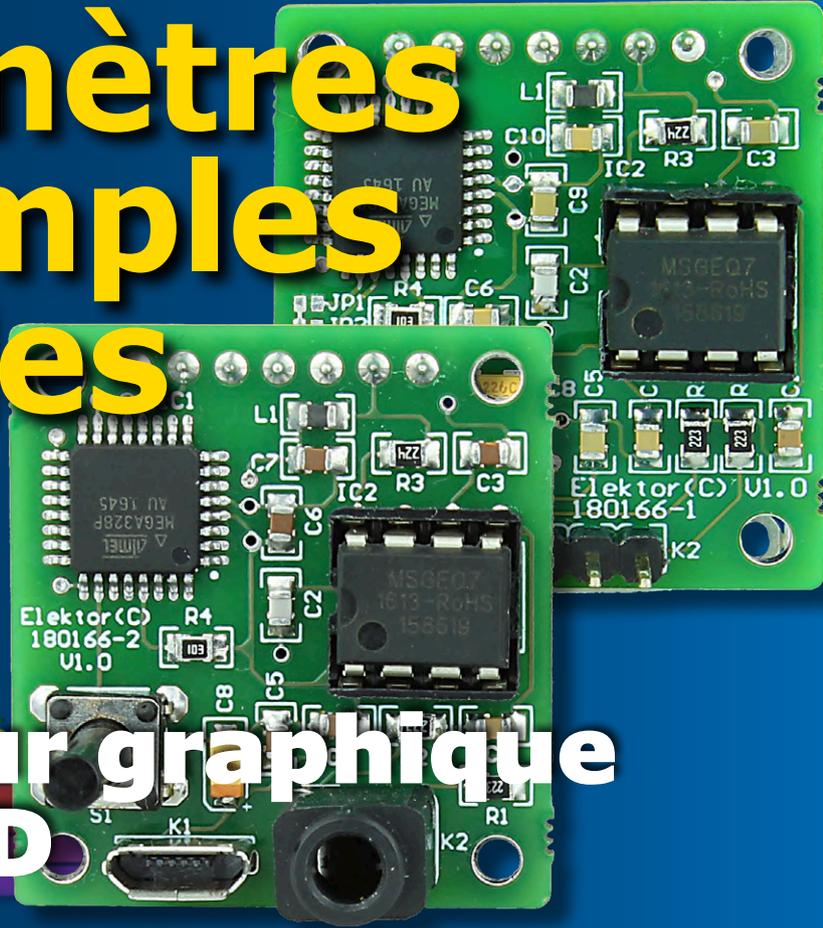


# spectromètres audio simples à 7 bandes

avec égaliseur graphique et écran OLED



## carte de commutation à relais à 9 canaux



avec commande locale ainsi qu'à distance



## horloge à LED géante avec Wi-Fi et mesures météo

chiffres de 10 cm, temps d'internet par ESP32 et capteurs d'ambiance sans fil

thyristor MOSFET à double anode ✂ commande pour la découpe de polystyrène ✂ cryptomonnaie IOTA (et FPGA pour RPi) ✂ kits à microcontrôleurs pour utilisateurs chevronnés ✂ surveillance de la consommation sur port USB ✂ du nouveau dans la famille des processeurs à 8 bits ✂ générateur d'impulsions de gâchette ✂ réveil à triple affichage : amélioration avec une horloge en temps réel précise ✂ projet SCCC (2) ✂ banc d'essai : station de soudage Weller WT 1014 ✂ introduction aux codes correcteurs d'erreurs ✂ Elektor Start-up Challenge – Paris 2019





AHEAD OF WHAT'S POSSIBLE™

LE PORTEFEUILLE ANALOG DEVICES DE PRODUITS DE PUISSANCE VOUS PERMET DE RÉSOUDRE LES PROBLÈMES D'ALIMENTATION LES PLUS DIFFICILES.

Maintenant vous pouvez être sur le marché plus rapidement avec tout ce qu'il vous faut pour offrir des solutions performantes qui possèdent une véritable avance concurrentielle.

Power by Linear™ vous offre :

- ▶ Une performance au top de l'industrie
- ▶ Une qualité et une fiabilité reconnue
  - ▶ La meilleure qualité
- ▶ Une politique de non obsolescence
- ▶ Un service et un support de première classe

# LA SOURCE DE PUISSANCE ULTIME

RENDEZ-NOUS VISITE À **pcim**  
EUROPE

07 - 09 Mai, Nuremberg, Allemagne  
HALL 6, STAND 6-442



#PowerByLinear

EXPLOREZ LES SOLUTIONS DE GESTION DE L'ALIMENTATION ADI  
[analog.com/power](http://analog.com/power)

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

@ : [service@elektor.fr](mailto:service@elektor.fr)

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

Télécopie : (+33) 01 70 61 27 42

[www.elektor.fr](http://www.elektor.fr) | [www.elektormagazine.fr](http://www.elektormagazine.fr)

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

#### Publicité :

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

#### DROITS D'AUTEUR :

© 2019 Elektor International Media B.V.

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

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

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



## Tût tût tût...

Fin 2018 : arrêt de l'installation des lignes de téléphone fixe en France. Le réseau téléphonique commuté (RTC) sera progressivement remplacé par un réseau de type IP. 2023 – 2025 : les lignes existantes seront définitivement coupées, zone par zone. Toutefois cette transition de l'analogique au numérique sera complexe, notamment pour les machines qui communiquent sur le RTC : terminaux de paiement, surveillance des ascenseurs, alarmes et télésurveillance, télérelève de compteurs...

### Disparitions en série : et la modulation de fréquence ?

La numérisation agite aussi le monde de la diffusion radio. Il y a quelques semaines, le Conseil supérieur de l'audiovisuel attribuait vingt-quatre fréquences de radio en DAB+ à des chaînes nationales et locales. Dans un premier temps ces chaînes ne seront disponibles que dans les grandes villes et sur les autoroutes. Couverture totale : 2030 ! La radio numérique terrestre ne remplace pas la diffusion en FM, mais la complète. La TNT avait réduit au silence les canaux analogiques, mais ce n'est pas le cas avec la RNT. Celle-ci permet d'écouter des radios qui ne sont pas diffusées sur la FM locale, mais elle souffre de la concurrence des radios par internet. Les webradios internationales ont le vent en poupe, les *podcasts* prolifèrent, de plus en plus de programmes sont personnalisables, sans parler du succès inattendu de la radio filmée... Le déploiement de la RNT est accompagné d'obligations pour les fabricants de récepteurs. L'Union européenne impose par exemple que tous les nouveaux autoradios puissent recevoir la RNT.

Cela suffira-t-il à réduire au silence la modulation de fréquence ?

Tous ces bouleversements stimulent l'innovation. Ils forment un terrain propice pour le développement de jeunes entreprises. C'est l'occasion de mettre en jeu leur potentiel de création de solutions inédites.

### Mariline Thiebaut-Brodier

PS Elektor lance en France un concours de startups. La finale aura lieu en septembre à Paris dans le cadre du salon *Forum de l'électronique*. Fort de son expérience avec des concours similaires dans d'autres pays, Elektor s'associe à CAP'TRONIC et GL Events Exhibitions pour une aventure similaire en France. Vous aussi ? (cf. p. 10 de ce numéro).

## Notre équipe

Rédactrice en chef :	Mariline Thiebaut-Brodier ( <a href="mailto:redaction@elektor.fr">redaction@elektor.fr</a> )
Rédaction internationale :	Thijs Beckers, Jan Buiting, Jens Nickel
Laboratoire :	Mathias Claussen, Ton Giesberts, Luc Lemmens, Clemens Valens (responsable), Jan Visser
Coordination :	Hedwig Hennekens
Ont coopéré à ce numéro :	Pascal Duchesnes, Pascal Godart, Yves Georges, Robert Grignard, Denis Lafourcade, Jean-Louis Mehren, Denis Meyer, Hervé Moreau, Helmut Müller, Kévin Petit, Xavier Pfaff, Guy Raedersdorf
Service de la clientèle :	Cindy Tijssen
Graphistes :	Giel Dols, Jack Jamar
Elektor en ligne :	Daniëlle Mertens

- 5 bientôt dans Elektor**
- 41 agenda**  
mai 2019
- 43 détecteur d'évènement nucléaire**  
drôle de composant n°40
- 47 agenda**  
juin 2019
- 84 vol tous azimuts**  
l'électronique par monts, maux et merveilles
- 104 Rétronique**  
instruments d'étalonnage des oscilloscopes  
Tektronix - le réglage fin le plus précis
- 110 questions d'éthique**  
qu'attendons-nous de l'intelligence artificielle ?
- 112 l'e-choppe d'Elektor**
- 114 hexadoku**  
casse-tête pour elektorniciens

## en coulisse

- 10 Elektor Start-up Challenge - Paris 2019**  
Une rampe de lancement internationale pour les start-ups françaises
- 12 la pratique des amplificateurs opérationnels**  
partie 3 : amplificateurs opérationnels avec étage d'entrée PNP et modèles de puissance
- 17 banc d'essai : station de soudage Weller WT 1014**
- 62 du nouveau dans la famille des processeurs à 8 bits**  
flash ultra-endurance et cellules logiques dans les contrôleurs PIC
- 74 kits à microcontrôleurs pour utilisateurs chevronnés**  
pour les projets exigeants
- 88 concours Fast Forward productronica 2019**
- 98 hors-circuits de R. Lacoste**  
introduction aux codes correcteurs d'erreurs syndrome de Huffman ? Pas de panique !

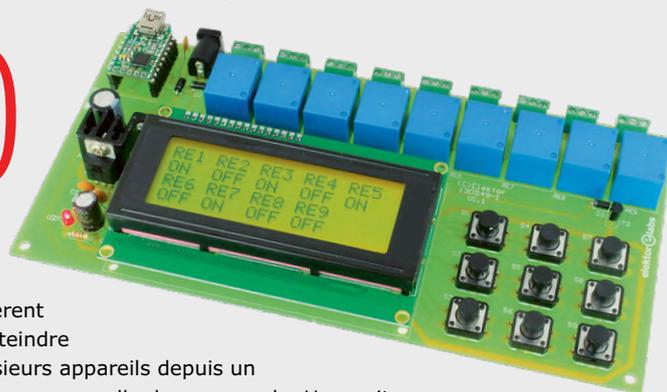
# spectromètres à 7 bandes

## eh oui, deux versions !

## carte de commutation à relais à 9 canaux

avec commande locale ainsi qu'à distance

# 70



De nombreuses situations requièrent d'allumer ou d'éteindre séparément plusieurs appareils depuis un point central comme une salle de commande. Un pupitre de commande avec un interrupteur pour chaque appareil semble une solution simple, mais elle ne permet pas la commande à distance, ni par ordinateur. Le circuit présenté ici apporte ces fonctions.

## sur la scène : les projets

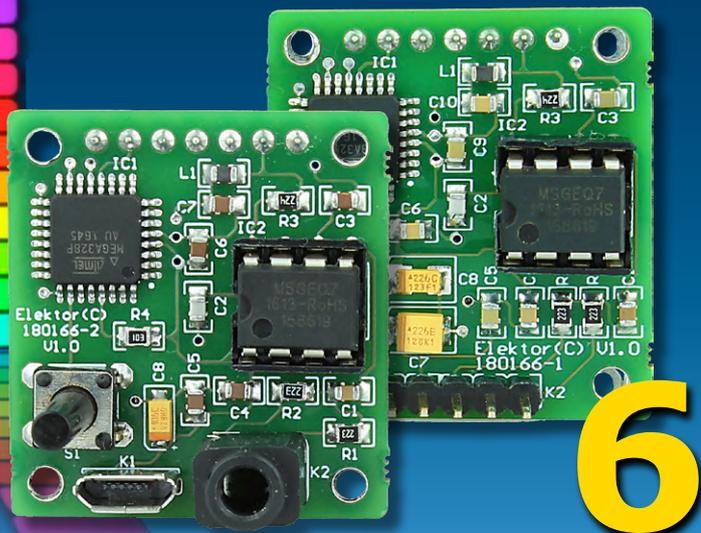
- 6 spectromètres audio simples à 7 bandes**  
eh oui, deux versions !
- 20 projet SCCC (2)**  
processeur *softcore* et compilateur C à construire soi-même
- 25 générateur d'impulsions de gâchette**  
amorçage précis des thyristors (SCR)

# 84

## vol tous azimuts

l'électronique par monts, maux et merveilles

# audio simples



**44 amélioration avec une horloge en temps réel précise**

retouche du réveil à triple affichage

**48 surveillance de la consommation sur port USB**

protection intelligente pour ports USB

**52 horloge à LED géante avec Wi-Fi et mesures météo**

des chiffres hauts de 10 cm, le temps d'internet par ESP32 et des capteurs d'ambiance sans fil

**70 carte de commutation à relais à 9 canaux**

avec commande locale ainsi qu'à distance

**90 commande pour la découpe de polystyrène**

alimentation à faible perte du fil chauffant grâce à un variateur à basse tension

**93 thyristor MOSFET à double anode**

plus de vitesse et moins de gaspillage qu'avec les vieux thyristors

## horloge à LED géante avec Wi-Fi et mesures météo

des chiffres hauts de 10 cm, le temps d'internet par ESP32 et des capteurs d'ambiance sans fil



**28 machine-outil à commande numérique Wolverine**

portique robotisé multifonctionnel – 2<sup>e</sup> partie : l'électronique et le logiciel

**36 configuration de la carte de liaison BME280 d'Elektor**

**38 cryptomonnaie IOTA (et FPGA pour RPi)**

1<sup>ère</sup> partie : les machines paient les machines

 **bientôt sur ces pages**

**Extrait du sommaire du prochain numéro :**

- Carte de liaison d'E/S MIDI
- Découverte du bus ASI
- Extensions pour le détecteur de coupure de courant
- Installation de Qt
- KiCad : dessiner de nouveaux composants
- Nouvelle carte CPLD avec MAX10
- Surveillance de bus CAN en PHP
- Surveillance de la consommation d'eau

Sous réserve de modification.

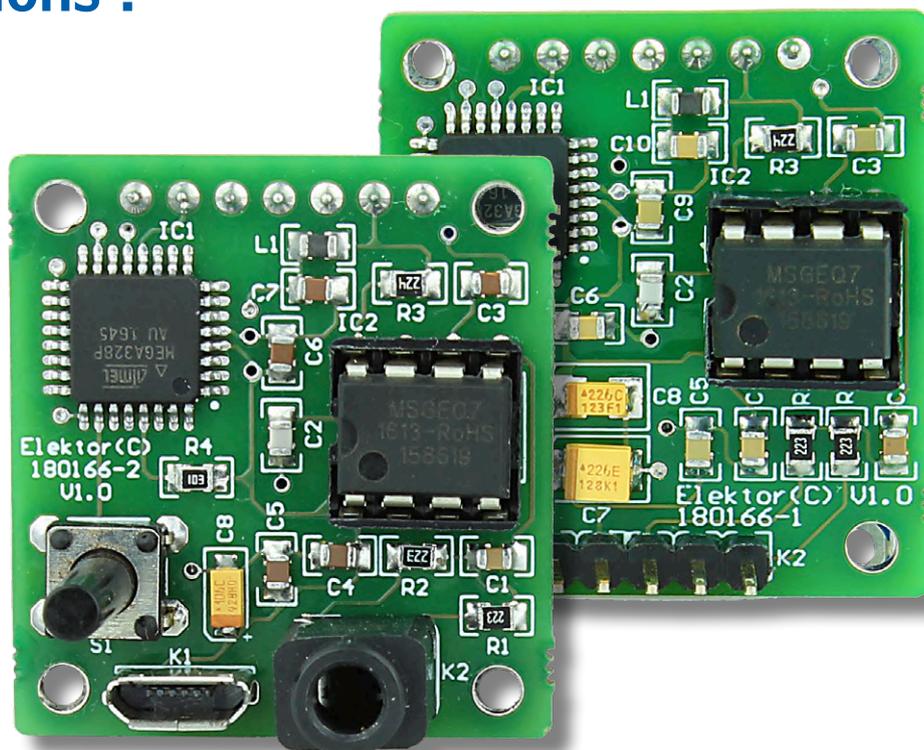
Le numéro de juillet-août 2019 paraîtra le 20 juin 2019.

# spectromètres audio simples à 7 bandes

## eh oui, deux versions !

Sunil Malekar (Inde)

Ce spectromètre audio expose sur sept barres verticales d'un écran OLED la densité du son dans les différents registres de la gamme. Vite fait, bien fait, avec juste un microcontrôleur et une puce d'égaliseur graphique, c'est attrayant, non ? Il existe en deux versions aux différentes fonctions selon l'alimentation et le niveau d'entrée.



### INFOS SUR LE PROJET



USB  
audio ATmega  
OLED



→ débutant  
connaisseur  
expert



env. 1 h



fer à souder,  
source audio



env. 25 €

Ses deux composants principaux sont l'égaliseur graphique MSGEQ7 de MSI [1] capable de répartir le spectre audio en sept bandes sur : 63 Hz, 160 Hz, 400 Hz, 1 kHz, 2,5 kHz, 6,25 kHz et 16 kHz. L'autre puce, c'est le microcontrôleur familial ATmega328P-AU, programmé ici pour convertir les signaux de sortie du MSGEQ7 en commandes pour un petit écran OLED en couleurs.

### Spectre et apparentés

Dans le domaine audio, un spectrogramme est l'analyse de l'amplitude

relative dans les bandes de fréquence contiguës d'un son composite ou d'un signal quelconque dont le volume varie au fil du temps. Ces spectrogrammes sont parfois appelés cascade spectrale, signature vocale ou audiogramme.

Il y en a de différents formats, le plus commun est un graphe à deux dimensions, en abscisse les fréquences, en ordonnée l'amplitude. Le sonagramme est la représentation tridimensionnelle de l'amplitude, du temps et de la fréquence d'un son, il permet aussi la reconnaissance vocale.

### Caractéristiques de la version 180166-1

- afficheur OLED RVB compact de 0,96 pouce
- spectrogramme audio sur 7 bandes de fréquences
- alimentation de 8 V à 12 V<sub>=</sub>
- conception compacte à faible consommation d'énergie
- choix entre 4 modes d'affichage par 2 cavaliers

## Pourquoi deux versions ?

La **figure 1** montre le schéma du spectromètre qu'il faut alimenter avec un bloc secteur donnant entre 8 et 12 V=. C'est le **version 180166-1**, on peut y adjoindre un haut-parleur ou un ampli-

ficateur audio. Le signal stéréo arrive par K2 et se transforme en mono avec les résistances R1 et R2 pour aboutir à l'analyseur de spectre à travers le condensateur C1. Dans le MSGEQ7, on détecte les valeurs

de pointe de chacune des bandes centrées sur les sept fréquences mentionnées. On les multiplie pour délivrer la valeur instantanée de l'amplitude dans chaque bande. Il ne faut aucun composant externe pour filtrer lesdites bandes.

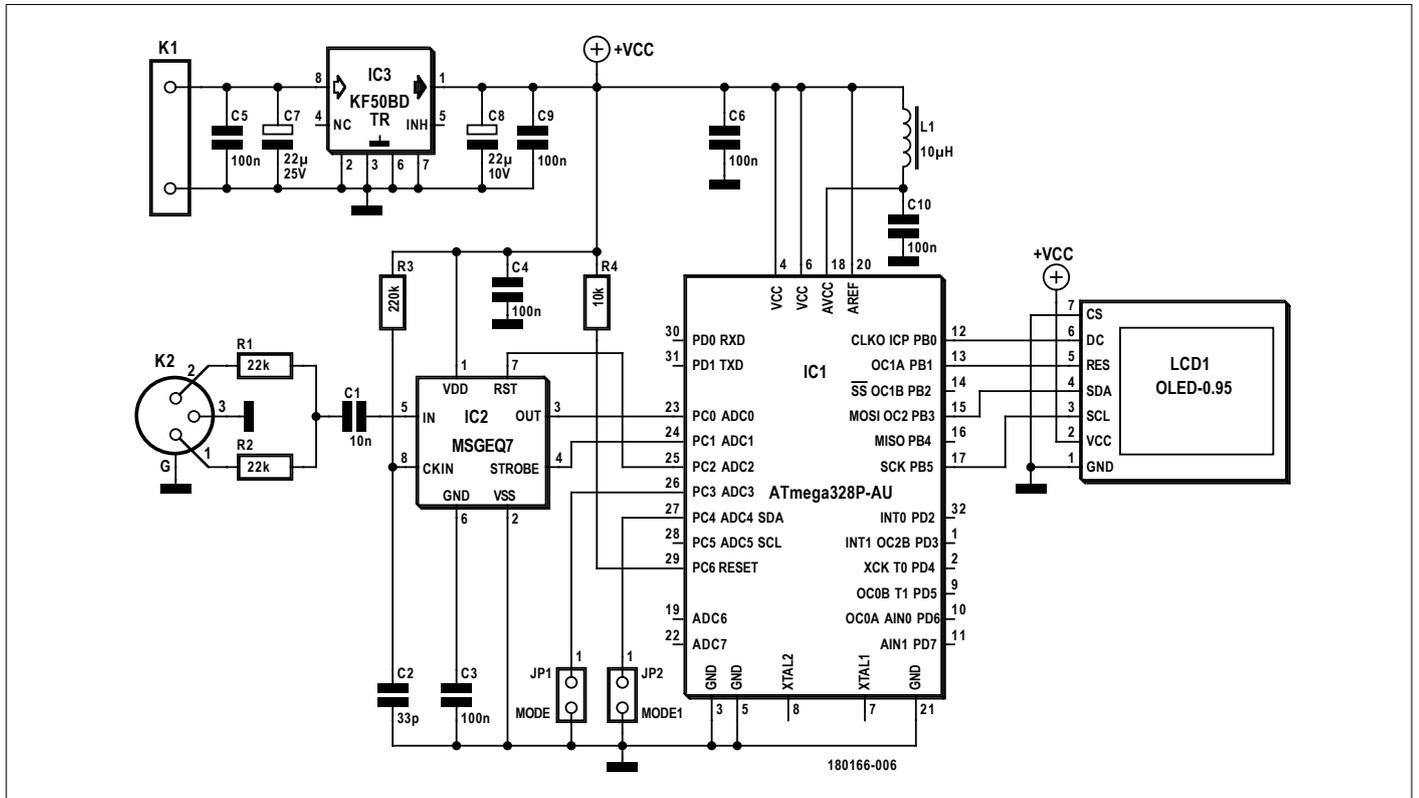


Figure 1. Schéma du spectromètre à 7 bandes avec adaptateur secteur de 8 à 12 V= ou équivalent. Cette version est prévue pour s'intégrer à un appareil audio existant.

### LISTE DES COMPOSANTS

Circuit imprimé 180166-1

**Résistances**  
R1, R2 = 22 kΩ, 1 %, 125 mW, 150 V  
R3 = 220 kΩ, 1 %, 125 mW, 150 V  
R4 = 10 kΩ, 1 %, 125 mW, 150 V

**Condensateurs**  
C1 = 1 nF, 50 V, 10 %  
C2 = 33 pF, 50 V, COG, 10 %  
C3 à C6, C9, C10 = 0,1 µF, 100 V, ±10 %  
C7 = 22 µF, 20 V, série TAJ, ±10 %  
C8 = 22 µF, 10 V, série TAJ, ±10 %

**Inductance**  
L1 = 10 µH

**Semi-conducteurs**  
IC1 = ATmega328P-AU, programmé  
IC2 = MSGEQ7  
IC3 = KF50BD-TR

**Divers**  
K1 = embase à 2 picots au pas de 2,54 mm  
K2 = embase à 3 picots au pas de 2,54 mm  
Support DIP au pas de 2,54 mm, série 2227MC, long. 7,62 mm

LCD1 = M605 0,95 pouce, 7 broches, couleurs RVB 65K, ssd1331 OLED  
Embase femelle à 7 contacts au pas de 2,54 mm  
Circuit imprimé 180166-1, dans l'e-choppe

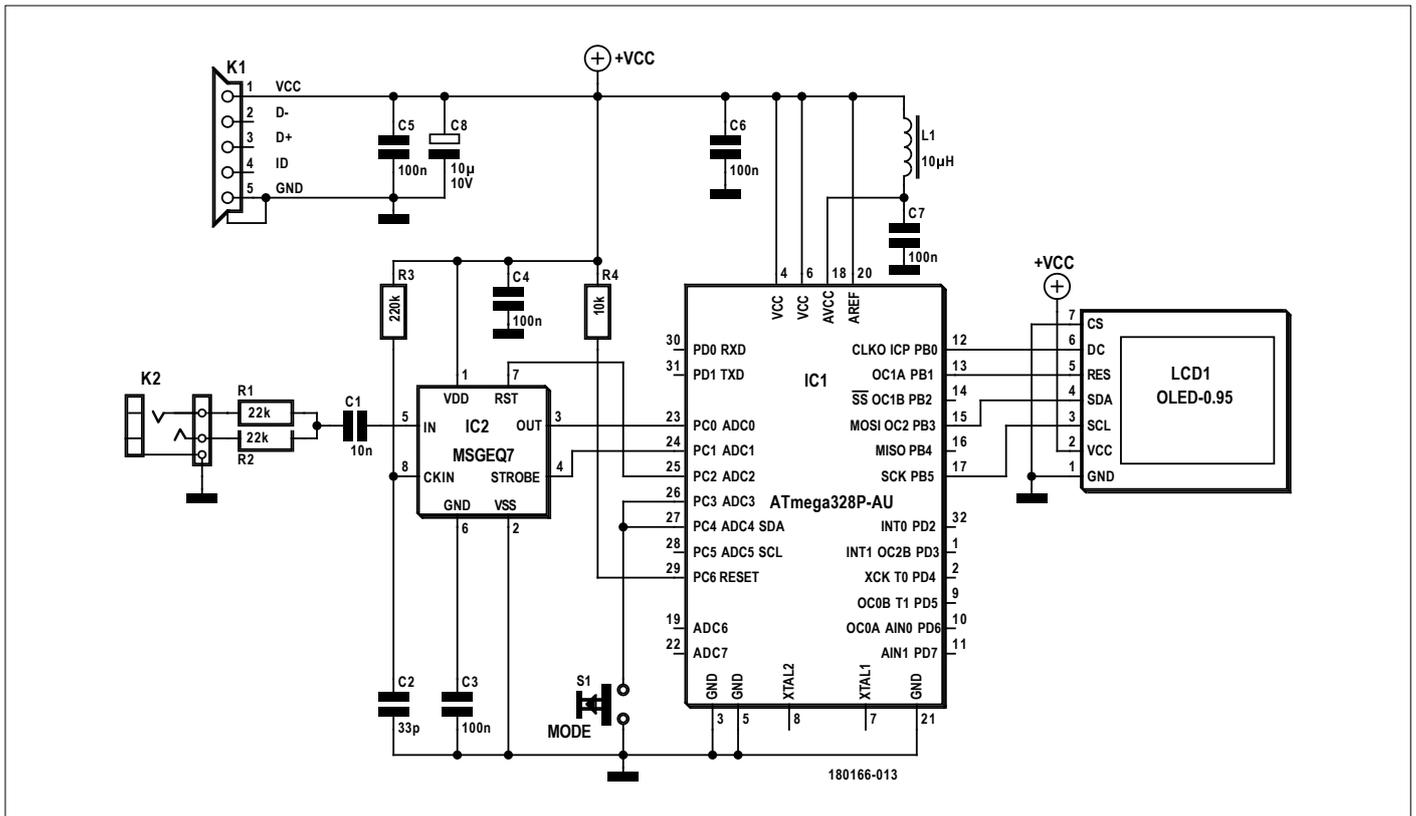


Figure 2. Observez les différences avec la figure 1 — en particulier autour de K1. Cette version est prévue pour servir d'appareil portable.

## Caractéristiques de la version 180166-2

- afficheur OLED RVB compact de 0,96 pouce
- spectrogramme audio sur 7 bandes de fréquences
- alimentation de 5 V<sub>=</sub> par USB
- conception compacte à faible consommation d'énergie

La fréquence d'horloge de l'oscillateur de la puce est fixée par R23 et C2. Chaque fréquence centrale des filtres se base sur celle de l'horloge.

Le constructeur MSI déclare que le MSGEQ7 donne ses meilleures performances à +5 V, exactement ce que délivre le régulateur IC3 entouré des habituels condensateurs de découplage

à la céramique et électrolytiques. Le MSGEQ7 se caractérise par un faible courant de repos typique, moins de 1 mA, parfait pour les appareils audio portables. On peut contrôler le multiplexeur embarqué par une broche RàZ (RST) et une broche d'échantillonnage pour lire le multiplexeur sur deux broches. La récurrence de lecture du multiplexeur com-

mande aussi la vitesse de décroissance, 10 % par lecture, sans nécessiter l'usage d'autres broches ou composants.

L'ATmega328P-AU et l'afficheur OLED sont tellement connus qu'ils rendent inutile toute description de leur matériel. Les cavaliers JP1 et JP2 suffisent à former l'une des quatre valeurs 00, 01, 10 et 11 pour sélectionner au choix l'une des quatre configurations de l'OLED, à savoir :

- bargraphe blanc
- bargraphe vert – jaune – rouge (chaque barre est tricolore)
- bargraphe fin bleu
- bargraphe fin rouge.



Un cavalier mis vaut un 0, ôté c'est un 1. Comme cette version-ci du spectro-mètre est conçue pour être incorporée à un système audio, la sélection du mode d'affichage par cavaliers est ce qu'il y a de plus simple, vu qu'elle ne requiert que deux cavaliers.



## LISTE DES COMPOSANTS

### Circuit imprimé 180166-2

#### Résistances

R1, R2 = 22 k $\Omega$ , 1 %, 125 mW, 150 V  
 R3 = 220 k $\Omega$  1 %, 125 mW, 150 V  
 R4 = 10 k $\Omega$ , 1 %, 125 mW, 150 V

#### Condensateurs

C1 = 1 nF, 50 V, 10 %  
 C2 = 33 pF 50 V, COG, 10 %  
 C3 à C7 = 0,1  $\mu$ F, 100 V, 10 %  
 C8 = 10  $\mu$ F, 10 V, série TAJ, 10 %

#### Inductance

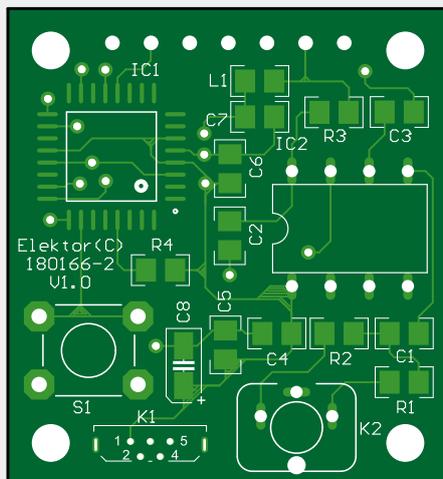
L1 = 10  $\mu$ H

#### Semi-conducteurs

IC1 = ATmega328P-AU, programmé  
 IC2 = MSGEQ7

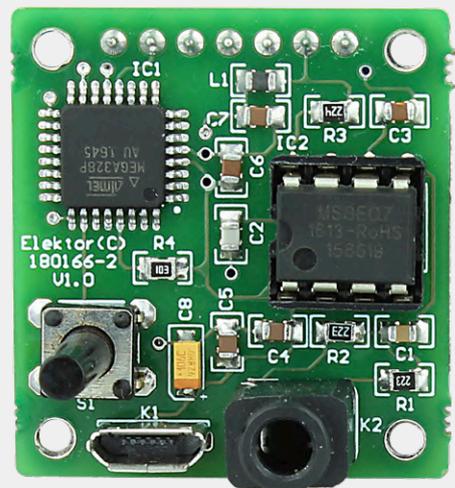
#### Divers

K1 = prise micro-USB type B, verticale



K2 = prise pour jack à 3 contacts, 3,5 mm, encartable

Support DIP au pas de 2,54 mm, série 2227MC, long. 7,62 mm



Embase femelle à 7 contacts au pas de 2,54 mm

S1 = bouton tactile, non éclairé, 24 V, 50 mA  
 Circuit imprimé 180166-2, dans l'e-choppe

Le schéma de la **version 180 166-2** est à la **figure 2**. Ce circuit s'insère aisément dans un petit boîtier portable avec de l'espace pour les connecteurs et un bouton-poussoir ainsi qu'un dégagement pour l'afficheur. On l'alimente en 5 V par USB sur le connecteur K1. Cette carte permet de sélectionner l'une des quatre configurations d'affichage en poussant sur S1.

### Le logiciel

Le logiciel de commande pour les deux versions a été rédigé avec Atmel Studio paramétré pour le microcontrôleur ATmega328p, rythmé par l'horloge interne à 8 MHz. Il est disponible gratuitement au téléchargement sur le site [2] du magazine. Il commande le multiplexeur de IC2 avec des impulsions sur les broches STROBE et RST. Les signaux appliqués sur les broches PC1 et PC2 de l'ATmega imposent au CA/N interne de lire sur la broche PC0 différentes fréquences, reproduites graphiquement sur l'écran OLED couleur de 0,96 pouce, connecté au PORTB du  $\mu$ C. La communication entre

le  $\mu$ C et l'écran s'effectue selon le protocole I<sup>2</sup>C, en tenant compte du mode d'affichage sélectionné par l'utilisateur sur les broches PC3 et PC4.

### Deux versions quand même

Chaque version du projet, que ce soit celle alimentée par un adaptateur 8 à 12 V= ou celle alimentée par USB, possède son propre circuit imprimé et sa liste de composants à part, donc il ne faut pas les mélanger. Aussi, vérifiez scrupuleusement votre assemblage et le choix des composants avant de passer au...

#### test de la version 180 166-1 :

- installez les cavaliers JP1 et JP2 pour choisir le mode d'affichage
- branchez le signal audio sur le connecteur K2
- alimentez le circuit par le connecteur K1 avec le bloc 8 à 12 V=
- vérifiez le spectre audio sur l'afficheur.

Pour changer de mode, éteignez l'alimentation, changez les cavaliers et remettez

le courant.

Si vous avez construit la version sur USB, vous pouvez commencer à :

#### tester la version 180166-2 :

- branchez le signal audio sur le connecteur K2
- connectez sur K1 l'USB du PC ou un adaptateur USB de 5 V
- vérifiez le spectre audio sur l'afficheur.

Pour changer de mode d'affichage, pressez S1. Il change le mode à chaque pression. ◀

(180166-03

version française : Robert Grignard)

### Liens

- [1] Feuille de caractéristiques du MSGEQ7 : [www.sparkfun.com/datasheets/Components/General/MSGEQ7.pdf](http://www.sparkfun.com/datasheets/Components/General/MSGEQ7.pdf)  
 [2] Page du projet : [www.elektormagazine.fr/180166-03](http://www.elektormagazine.fr/180166-03)



@ WWW.ELEKTOR.FR

→ Circuit imprimé 180166-1  
[www.elektor.fr/180166-1](http://www.elektor.fr/180166-1)

→ Circuit imprimé 180166-2  
[www.elektor.fr/180166-2](http://www.elektor.fr/180166-2)

→ Microcontrôleur programmé pour la version 180166-1  
[www.elektor.fr/180166-41](http://www.elektor.fr/180166-41)

→ Microcontrôleur programmé pour la version 180166-2  
[www.elektor.fr/180166-42](http://www.elektor.fr/180166-42)



# elektor start-up challenge

# Paris 2019

## Une rampe de lancement internationale pour les start-ups françaises

Mariline Thiebaut-Brodier, Denis Meyer

### Qui peut embarquer ?

Vous par exemple, qui développez un produit, un logiciel ou un service lié à l'électronique. *Elektor Startup Challenge* vous donne l'occasion de venir présenter votre projet aux professionnels de la branche.

Les précédents concours organisés par Elektor, notamment dans le cadre d'*electronica* à Munich, le plus grand salon d'électronique au monde, ont récompensé des projets remarquables dans différents domaines :

- Teiimo GmbH (Allemagne) pour son système textile de mesure et de transmission de paramètres biologiques vitaux.
- Wizama (France) pour son plateau de jeu à écran tactile et objets connectés.
- Querom Elektronik (Allemagne) pour ses solutions d'électronique de puissance sur mesure et ses convertisseurs CC/CC haute tension.
- Mowea (Allemagne) pour son prototype de sa pico-éolienne qui fournit aux domiciles équipés d'une installation photovoltaïque une source auxiliaire pour les périodes sans soleil (nuit, mousson, smog).
- BotFactory (Etats-Unis) avec son imprimante 3D Squink qui produit des circuits imprimés multicouches, avec fonctions intégrées de distribution de colle ou de soudure et de placement des composants, commandées directement par des fichiers Gerber.
- Wisebat (France) avec sa plate-forme de simulation pour les ingénieurs en électronique qui développent des dispositifs pour l'Internet des Objets.

### Pourquoi monter à bord de la fusée ?

Si vous vous inscrivez à l'*Elektor Start-Up Challenge* de Paris 2019 et que votre candidature est retenue, vous serez accueilli avec les autres concurrents sur l'aire de l'*Elektor Start-Up Challenge*, un grand stand au beau milieu du salon **Forum de l'électronique** (Paris Expo Porte de Versailles, du 24 au 26.09.2019).

Vous pourrez :

- y présenter votre projet sur votre propre pupitre
- concourir au côté des autres finalistes
- attirer l'attention de partenaires potentiels

Avant, pendant et après le salon, vous bénéficierez d'une large visibilité :

- vous et votre start-up serez présents dans les supports du Forum de l'électronique : catalogue des exposants, liste des exposants, site officiel...
- vous et votre start-up serez présentés à l'ensemble des lecteurs du magazine Elektor via tous nos supports (papier, web, réseaux sociaux, chaîne YouTube...) en France et dans les pays francophones
- vous et votre start-up serez présentés aux adhérents du programme Cap'tronic

Pour vous et pour votre start-up, le summum ce serait d'emporter **le gros lot**. En effet, le gagnant de l'*Elektor Start-up Challenge* à Paris recevra son ticket d'entrée pour le prochain salon **electronica** à Munich : vous serez sélectionné d'office pour y participer au concours international **Fast Forward Award** au cours duquel vous **présenterez votre projet sur le stand** et bénéficierez de la très large publicité donnée à cet événement.

*Elektor Startup Challenge* est organisé en collaboration avec :



**Forum de  
l'électronique**  
Salon de l'innovation et des solutions électroniques



**CAP'TRONIC**  
Systèmes électroniques et  
logiciel embarqué

## N'attendez pas qu'Elon Musk vous invite à bord de SpaceX !

Après l'Allemagne et les Pays-Bas, cette année Elektor fait escale en France avec un nouveau concours pour les jeunes entreprises dont l'activité est liée à l'électronique embarquée. Cela se passera à Paris dans le cadre du nouveau salon Forum de l'électronique en septembre 2019.

Ce concours baptisé Elektor Start-Up Challenge vise à stimuler la croissance de jeunes pousses après leurs débuts prometteurs.

Le choix des gagnants sera guidé par la viabilité industrielle des projets, leur utilité et bien sûr leur originalité. L'Elektor Start-Up Challenge est le fruit de notre coopération étroite avec Cap'tronic, le programme d'aide aux PME en matière de systèmes électroniques et de logiciel embarqué, et GL Events Exhibitions, l'organisateur du salon Forum de l'électronique.

Réunion de travail à Paris dans les locaux de GL Events Exhibitions pour l'organisation du prochain *Elektor Start-up Challenge*.

De gauche à droite :

**Michel Marceau** (directeur du programme CAP'TRONIC Paris Nord Est),

**Jérôme Letu-Montois** (directeur du salon Forum de l'électronique),

**Denis Meyer** (Elektor),

**Mariline Thiebaut-Brodier** (Elektor & MagPi).



### Le rendez-vous de votre start-up avec l'avenir !

**Où ?** Paris Expo Porte de Versailles

**Quand ?** Du 24 au 26 septembre 2019

**Comment ?** Sur inscription au plus tard le mercredi 31 juillet 2019.

**Rendez-vous sur :** [www.elektormagazine.fr/escparis2019](http://www.elektormagazine.fr/escparis2019)

### Attention :

### la date limite d'inscription est fixée au mercredi 31 juillet 2019.

En cas de forte affluence, il est possible que le nombre limité de places nous contraigne à avancer la date de clôture des inscriptions. Ne tardez pas à vous rendre sur la page [www.elektormagazine.fr/escparis2019](http://www.elektormagazine.fr/escparis2019) pour vous inscrire.

Pour tout renseignement supplémentaire sur ce concours, veuillez adresser votre question par courriel à l'adresse [redaction@elektor.fr](mailto:redaction@elektor.fr)

### Présenter sa candidature

Pour participer à *Elektor Start-Up Challenge* dans le cadre du *Forum de l'Électronique*, le candidat doit commencer par créer un compte (gratuit) sur le site [www.elektormagazine.fr](http://www.elektormagazine.fr).

Il devra fournir une documentation la plus complète possible de son projet afin que le jury puisse apprécier clairement la mission de la start-up, la technologie du projet, se faire une idée du produit et, le cas échéant, des services fournis. Toutes les propositions seront évaluées par le jury de l'*Elektor Start-Up Challenge*.

Pour être considérée comme une start-up, son entreprise doit avoir été créée il y a moins de cinq ans.

Si sa start-up est admise à participer à l'*Elektor Start-Up Challenge*, le représentant de l'entreprise accepte d'être personnellement présent à Paris lors du *Forum de l'Électronique 2019*.

Toutes les conditions générales de participation au *Elektor Start-Up Challenge* de Paris sont disponibles en ligne.



# la pratique des amplificateurs opérationnels

## partie 3 : amplificateurs opérationnels avec étage d'entrée PNP et modèles de puissance

Burkhard Kainka

Le dernier article de cette série succincte et élémentaire sur les AOP montre comment tirer parti des particularités des amplificateurs opérationnels. En second lieu, nous aborderons les AOP de puissance.

Les amplificateurs opérationnels travaillent pour la plupart sur une plage de tension limitée en deçà de la tension d'alimentation, la différence étant engendrée par la source de courant, la jonction base-émetteur et le miroir de courant de l'étage d'entrée. Certains modèles peuvent travailler presque jusqu'à la tension d'alimentation négative grâce à l'optimisation spécifique de leur circuit d'entrée : ils possèdent un chemin base-émetteur supplémentaire dans l'amplificateur différentiel d'entrée, de sorte qu'ils peuvent accepter des signaux d'entrée atteignant la limite de tension d'alimentation négative. Par ex. l'AOP double LM358 (**fig. 1**) ainsi que l'AOP quadruple LM324 peuvent être utilisés avec une tension d'alimentation unique de +3 V ; ils sont donc adaptés à un fonctionnement sur batterie. Grâce à l'utilisation d'étages d'entrée PNP, ces AOP parviennent même à faire descendre la tension d'entrée admissible à 300 mV *au-dessous* de la tension d'alimentation négative. La tension de sortie en revanche n'atteint pas tout à fait le zéro. Dans

tous les cas, vous avez intérêt à regarder de près les caractéristiques des entrées.

Le LM358 est ainsi l'AOP standard à tout faire qui, par sa production de masse, est disponible à bas coût et suffit dans la plupart des cas. Si plus de deux AOP sont mis en œuvre dans un circuit, le LM324, tout aussi avantageux, est tout indiqué. En fait, nous devrions toujours concevoir le circuit de sorte qu'il fonctionne correctement avec presque tous les amplificateurs opérationnels. Il est toutefois intéressant d'exploiter les caractéristiques spécifiques de certains modèles d'AOP pour réduire le nombre de composants ou offrir des solutions sortant des sentiers battus. Le schéma de la **figure 2** montre un capteur de luminosité simple utilisant une LED verte comme photodiode. La LED est polarisée en inverse, c.-à-d. avec l'anode à la masse. Si la LED est éclairée, un faible courant photonique se met à circuler. Sous cette forme, l'essai ne sera pas concluant avec tous les AOP, mais avec les LM358 ou LM324 ça fonctionne bel et bien. En effet, grâce à leurs étages d'entrée PNP, ces

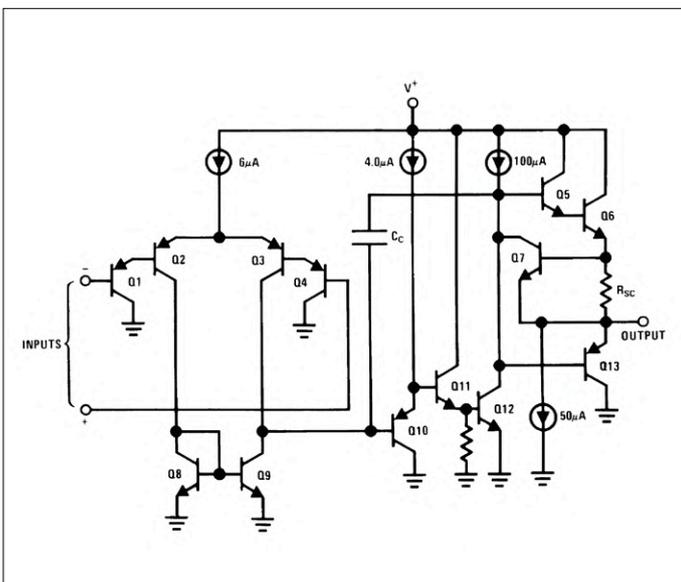


Figure 1. Schéma interne du LM358 avec étages d'entrée PNP (doc. : Texas Instruments).

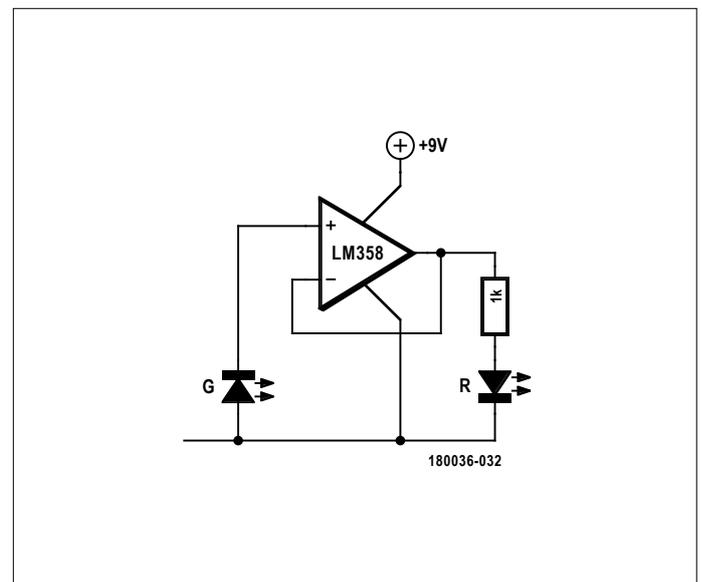


Figure 2. Exploitation du courant d'entrée.

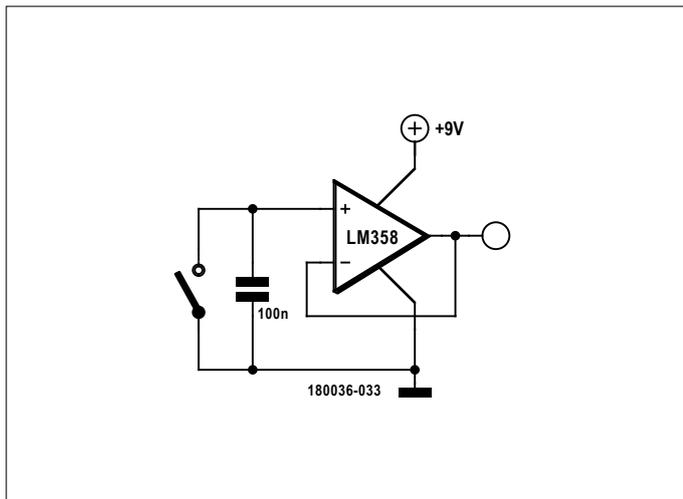


Figure 3. Générateur de rampe simple.

modèles produisent sur chaque entrée un faible courant (courant de repos sur chaque base) de 30 nA environ. Le potentiel d'une entrée ouverte est alors tiré vers le haut. Sous un faible éclairage, la LED rouge s'allume. Dès qu'une quantité suffisante de lumière tombe sur le capteur (LED verte), ce dernier détourne le courant d'entrée ce qui fait baisser la tension d'entrée. Lorsque l'éclairage est suffisant, l'AOP éteint la LED rouge. Le courant d'entrée bénéficie d'une faible dispersion et est pratiquement indépendant de la tension d'alimentation. Il détermine ici le seuil de la luminosité qui déclenche l'extinction de la LED.

Si l'on souhaitait réaliser le circuit avec des amplificateurs opérationnels CMOS, il faudrait produire un courant d'entrée via une résistance de 330 M $\Omega$ , difficile à trouver dans un magasin d'électronique. En ajoutant un condensateur sur l'entrée d'un LM358, on obtient un générateur de rampe très simple (fig. 3). Le courant d'entrée pratiquement constant de l'AOP charge lentement le condensateur. La pente de la courbe permet de calculer le courant

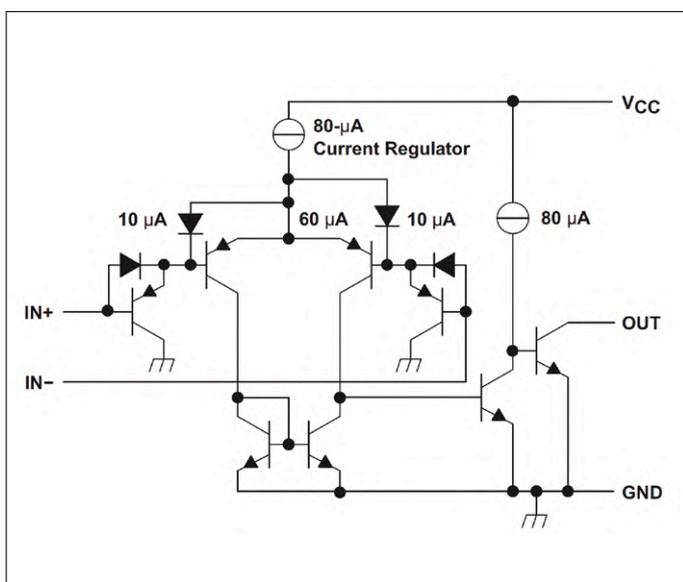


Figure 5. Schéma interne du comparateur LM339 (doc. : Texas Instruments).

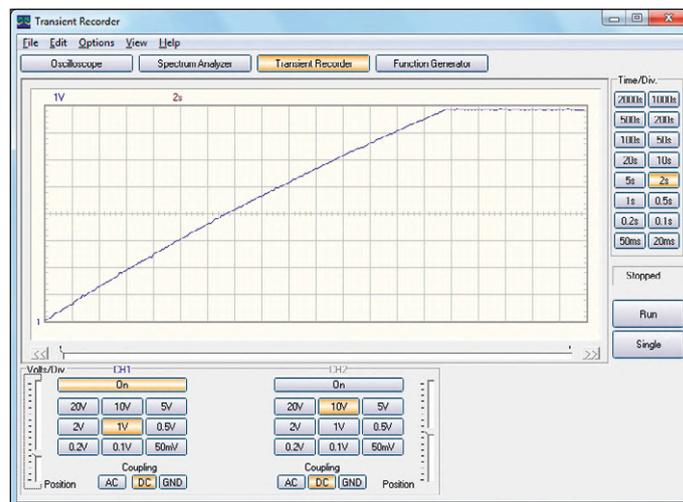


Figure 4. Rampe de tension en sortie.

d'entrée avec précision (fig. 4). La rampe montre une légère incurvation. Avec un courant de charge de 30 nA, la première dizaine de secondes permet d'atteindre une tension de 3 V.

### Comparateur LM339

Le comparateur LM339 utilise des étages d'entrée très similaires (fig. 5). Des diodes complémentaires améliorent le comportement en cas de forts dépassements du signal d'entrée. Il utilise une sortie à émetteur commun et collecteur ouvert de sorte qu'il faut monter une résistance de collecteur externe. Une autre différence par rapport à un amplificateur opérationnel classique est l'absence de limitation interne de la bande passante. Ainsi en sortie, la commutation est plus rapide, la fréquence admissible plus élevée et les flancs sont plus raides. Dans le cas d'un comparateur, il n'y a aucun problème de stabilité à craindre, car en pratique, il se trouve toujours dans un état de saturation des entrées pour lequel il n'y a effectivement pratiquement pas d'amplification.

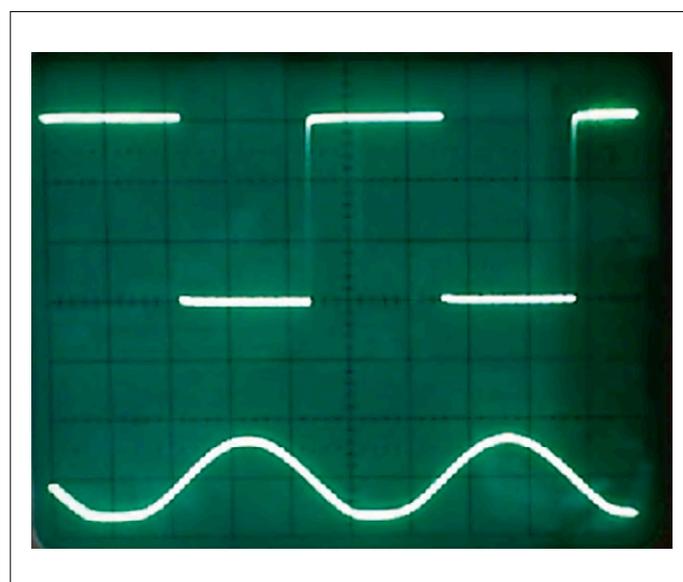


Figure 6. Synthèse de signaux carrés.

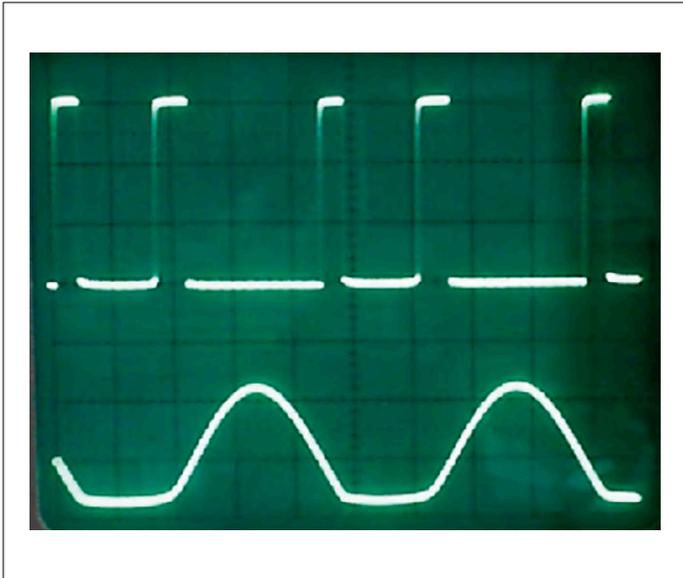


Figure 7. Saturation dans la zone négative.

On pourrait essayer de relier le comparateur avec une résistance de charge au  $V+$ , comme pour un amplificateur opérationnel. Avec une amplification plus élevée, le comparateur se comporterait comme un amplificateur opérationnel classique, mais avec une contre-réaction totale, il se mettrait à auto-osciller. Mieux vaut laisser le comparateur travailler dans le domaine pour lequel il a été conçu, c.-à-d. fournir des signaux carrés. La **figure 6** montre le cas type visant à transformer un signal sinusoïdal ou d'autres formes de signaux en signaux carrés. Pour ce faire, nous pourrions également utiliser un LM358, mais un véritable comparateur atteindra une fréquence limite plus haute.

La feuille de caractéristiques donne pour une tension d'alimentation de 5 V une plage de mode commun (plage acceptable pour un signal présent sur les deux entrées, sans déphasage) de 0 à 1,5 V. Des tensions d'entrée inférieures à zéro ne sont pas prévues ; cela fonctionne cependant jusqu'à  $-300$  mV environ, exactement comme avec le LM358. À partir de  $-500$  mV

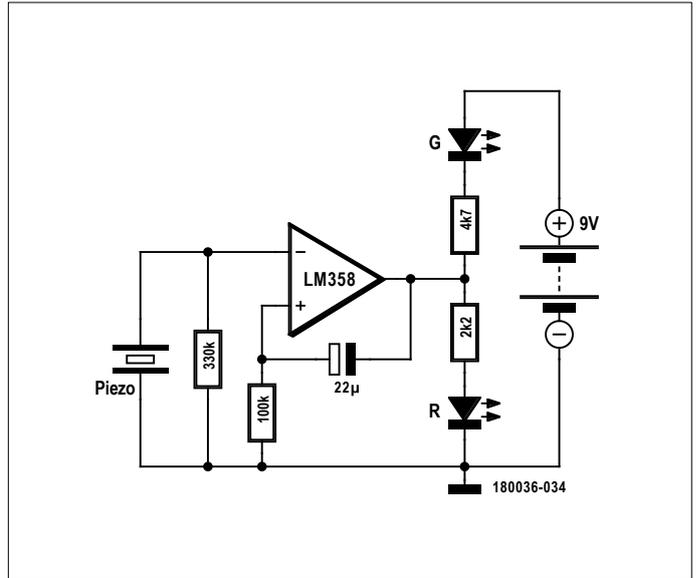


Figure 8. Monostable et intégrateur avec un LM358.

de sérieux problèmes apparaissent cependant. L'entrée inverse sa fonction subitement, le résultat de la comparaison est alors inversé. Avec une saturation dans le domaine négatif, on retrouve soudainement en sortie des signaux à fréquence doublée (**fig. 7**). Pour lutter contre ce phénomène, deux diodes Schottky montées tête-bêche sur les entrées sont efficaces. Elles limitent en effet la tension d'entrée à  $\pm 0,3$  V.

### Commutateur sonore

Le LM358 est affecté par le problème suivant : jusque  $-0,3$  V sur les entrées, tout se passe normalement, mais si une tension de  $-0,5$  V est appliquée sur l'entrée non-inverseuse, il se comporte soudain comme s'il s'agissait d'une entrée inverseuse. Heureusement, il est toujours possible de tirer parti d'un défaut : grâce à ce phénomène très gênant, on peut réaliser un commutateur sonore avec un disque piézoélectrique en guise de capteur pour commuter des LED au moyen d'un bruit assez fort ou en tapotant sur le capteur (**fig. 8**). Au repos,

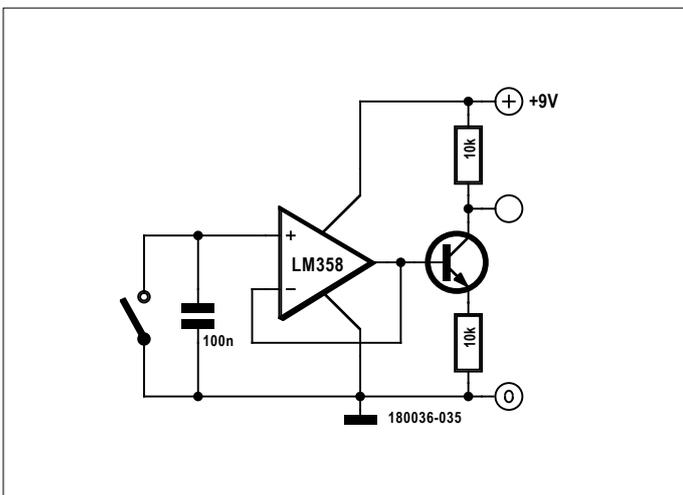


Figure 9. Montage d'essai pour l'inversion de la fonction d'entrée.

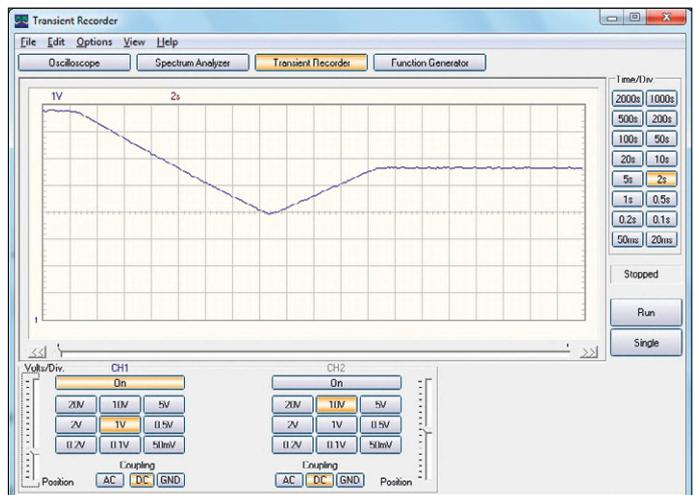


Figure 10. Inversion du signal par saturation.

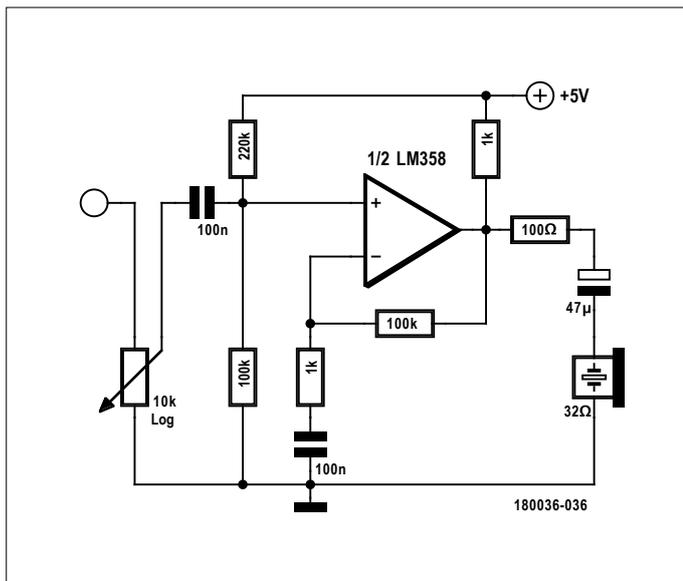


Figure 11. Amplificateur de casque.

la LED verte est allumée. Un bruit ou une secousse provoque la commutation rapide du vert au rouge. Il faut ensuite environ 30 s pour rebasculer sur la LED verte. Il s'agit visiblement d'un multivibrateur monostable.

Mais il se comporte bizarrement : le passage du rouge au vert n'est pas brutal, il s'agit plutôt d'un fondu-enchaîné lent. Un amplificateur opérationnel idéal se comporterait tout autrement. Nous utilisons ici les caractéristiques spécifiques des amplificateurs opérationnels à étages d'entrée PNP. Le courant d'entrée est de 30 nA environ de sorte qu'une chute de tension de 10 mV sur l'entrée inverseuse se produit tandis que l'entrée non-inverseuse ne voit que 3 mV. Cette différence suffit à établir un état de repos stable. Le capteur doit apporter au moins 7 mV pour modifier cet état. En basculant à l'état rouge, le condensateur électrochimique tire la tension de l'entrée + au positif et maintient cet état stable par la contre-réaction. Il se charge ensuite lentement jusqu'à ce que la tension d'entrée baisse au-dessous de 5 mV, ce qui prend environ 30 s. Le circuit bascule ensuite dans l'état initial. Cependant l'entrée descendrait alors de quelques volts au-dessous de zéro ce qui n'est pas dans la plage de fonctionnement normal de l'amplificateur opérationnel. Au-dessous de  $-0,5\text{ V}$ , la fonction du LM358 s'inverse. En raison de la contre-réaction, le circuit travaille alors en intégrateur pendant un certain temps. C'est la raison pour laquelle l'état des sorties ne se modifie que lentement. Si l'on veut comprendre ce phénomène plus précisément, il faut examiner l'étage d'entrée de plus près. Avec une entrée + trop négative, les deux émetteurs des transistors Darlington de l'amplificateur différentiel d'entrée sont suffisamment tirés vers une tension négative pour que les tensions sur les collecteurs soient encore plus négatives. Il est facile de s'en assurer au moyen d'un montage d'essai : nous pilotons ici un transistor NPN (fig. 9). Il est commandé par le générateur de rampe évoqué ci-dessus. La tension de sortie commence par descendre lorsque la tension d'entrée monte, comme chacun s'y attend avec un circuit à émetteur commun (fig. 10). Cependant dès que le transistor est saturé, le comportement s'inverse. La tension de sortie recommence à augmenter.

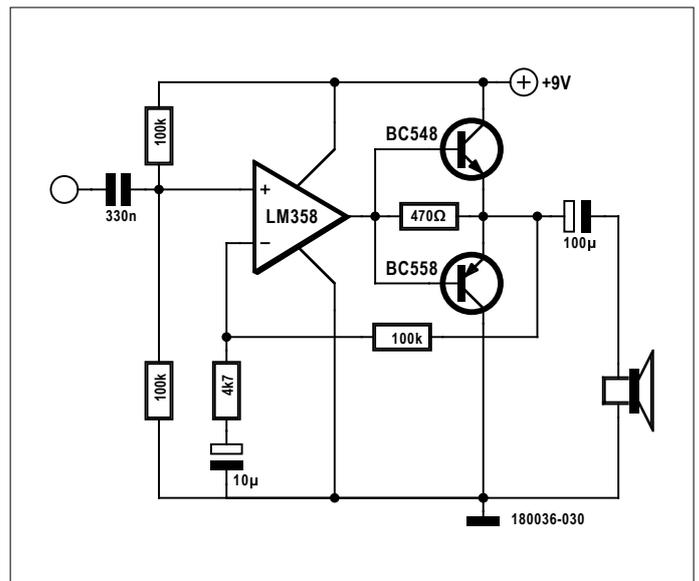


Figure 12. Amplificateur de puissance à transistors complémentaires.

### Amplificateur de puissance

La plupart du temps les amplificateurs opérationnels ne fournissent qu'un faible courant de sortie, tout au plus 10 mA. Il existe cependant des amplificateurs opérationnels de puissance spécialisés. Parfois un AOP standard peut fournir suffisamment de courant ou bien on utilise quelques transistors supplémentaires pour augmenter le courant de sortie.

Le petit amplificateur de casque de la **figure 11** est construit autour d'un AOP double, le LM358, pour utiliser directement la radio logicielle SDR d'Elektor sans passer par un PC. S'il est utilisé à gain élevé avec une charge de faible valeur ohmique, le LM358 tend à produire des distorsions. Ce problème est réduit avec une résistance de 1 kΩ intercalée entre la sortie et la tension d'alimentation, car pour de petits signaux, seule la branche inférieure de la sortie complémentaire est mise à contribution. En série avec le casque, nous ajoutons également une résistance de 100 Ω. Cela facilite la tâche de l'amplificateur opérationnel et exclut par ailleurs les dommages auditifs, car la puissance maximale dans le casque reste limitée. La tension de sortie du LM358 peut descendre presque jusqu'à la masse, mais ne monte pas tout à fait jusqu'à la tension d'alimentation. C'est pourquoi nous avons opté pour une tension de repos égale au tiers de la tension d'alimentation.

Les amplificateurs opérationnels permettent également de réaliser des amplificateurs de haut-parleurs (**fig. 12**). Cependant, le courant maximal de sortie limité à 10 mA environ ne peut convenir qu'à un amplificateur de casque simple. Deux transistors de plus conduisent à un montage abordable. Les transistors réalisent un étage *push-pull* à courant de repos nul. Les petits signaux sont prélevés directement en sortie de l'AOP. Les transistors de sortie ne commencent à amplifier qu'à partir de 10 mA. Le principe ressemble fort au légendaire amplificateur Edwin des débuts d'Elektor.

Un amplificateur *push-pull* à courant de repos nul produit par principe de fortes distorsions de raccourcement. Celles-ci sont cependant en grande partie compensées par la forte contre-réaction du montage. Il n'est cependant pas question d'amplificateur hi-fi. Ce schéma convient pour l'expérimentation

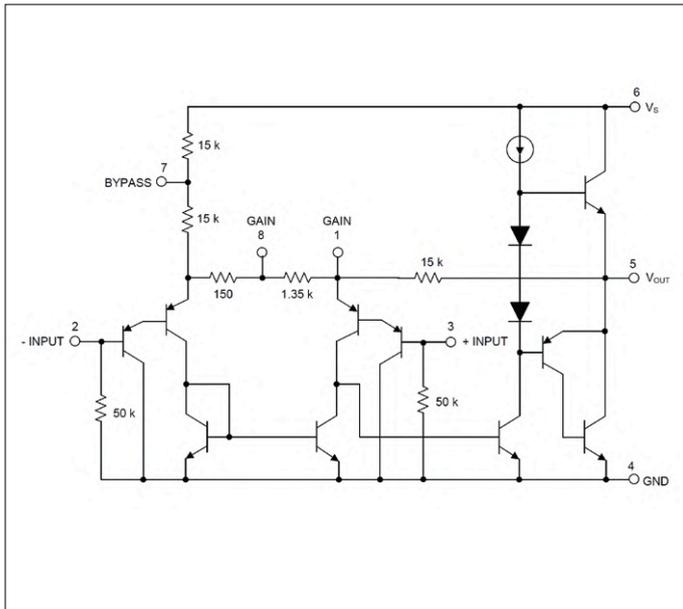


Figure 13. Schéma interne de l'amplificateur de haut-parleur LM386 (doc. : Texas Instruments).

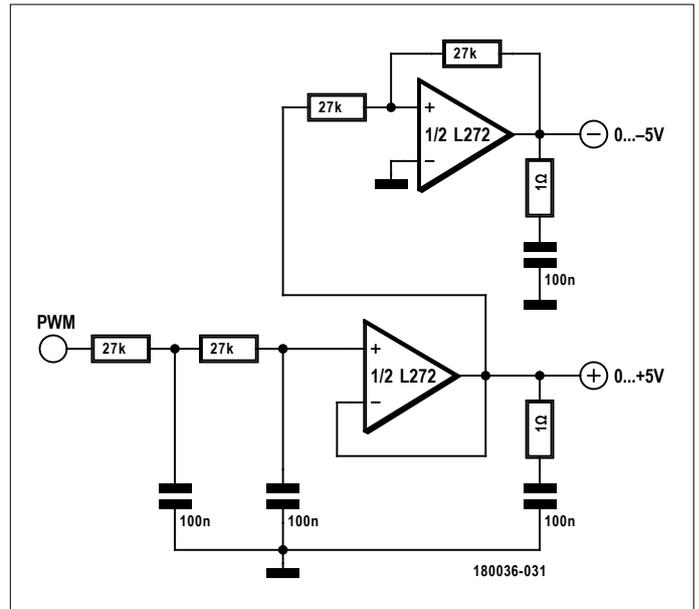


Figure 14. Mise en œuvre de l'AOP de puissance L272.

ou bien un simple avertisseur sonore. Pour des applications sérieuses, mieux vaudrait utiliser un amplificateur de puissance intégré comme le LM386.

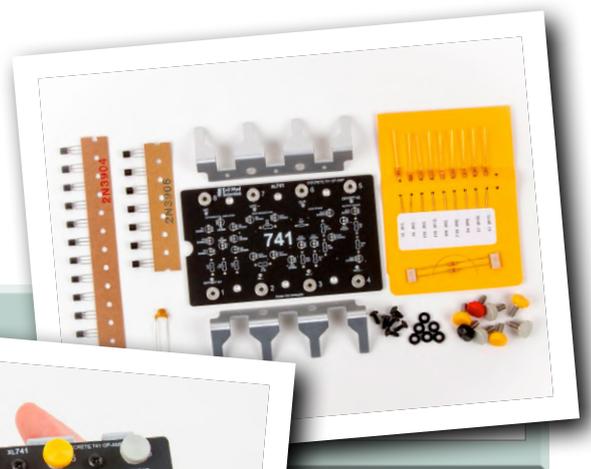
Le schéma interne du LM386 (**fig. 13**) ressemble fortement à celui d'un amplificateur opérationnel. Toutefois, il comporte une contre-réaction interne pour obtenir un gain initial de 20 environ. Les étages d'entrée PNP permettent des tensions d'entrée produisant une saturation autour du potentiel de masse. La tension de sortie médiane s'établit sans nécessiter d'autres composants.

Pour les applications générales, avec une plus grande puissance de sortie, nous pouvons utiliser le double AOP de puissance L272 capable de fournir jusqu'à 1 A. La **figure 14** montre le schéma de réalisation d'une alimentation délivrant une tension double réglable. Un microcontrôleur fournit un signal MLI (modulation en largeur d'impulsion) lissé par deux filtres passe-bas en une tension continue variable de 0 à 5 V. Le L272 travaille en tampon de puissance et en inverseur pour la sortie négative de l'alimentation. L'AOP tend par ailleurs à auto-osciller. C'est pourquoi des filtres de 100 nF / 1 Ω ont été ajoutés, comme

on en voit souvent sur les amplis audios. Ils assurent la stabilité du montage.

Le L272 possède comme le LM358 un étage d'entrée PNP. Mieux vaut le savoir ! Dans un test de fonctionnement simple, nous connectons brièvement l'entrée à la masse (sans raccorder cette entrée au signal MLI d'un microcontrôleur). Les deux tensions de sortie sont alors proches de 0 V. Dès que l'on ouvre l'entrée, le courant d'entrée du L272 charge lentement les condensateurs du filtre d'entrée. Nous nous retrouvons dans le cas du générateur de rampe simple et nous pouvons examiner le comportement en tension des deux sorties avec ou sans charge pour détecter d'éventuelles instabilités. ◀

(180036-C-04 - version française : Yves Georges)



@ WWW.ELEKTOR.FR

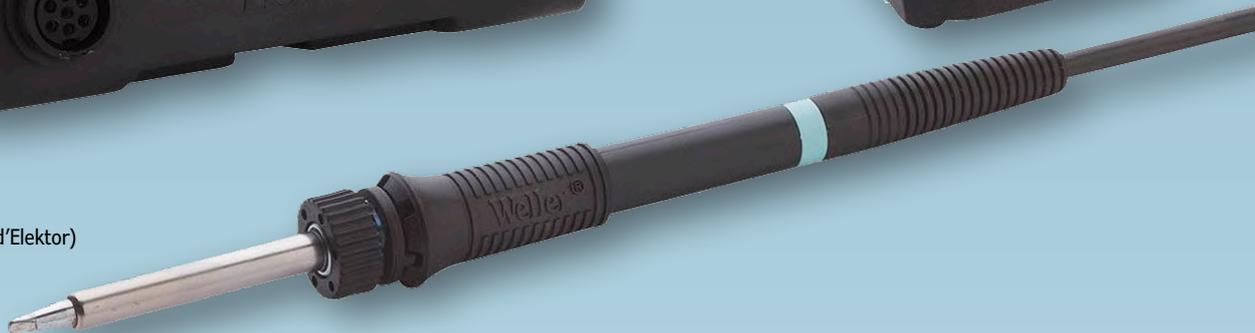
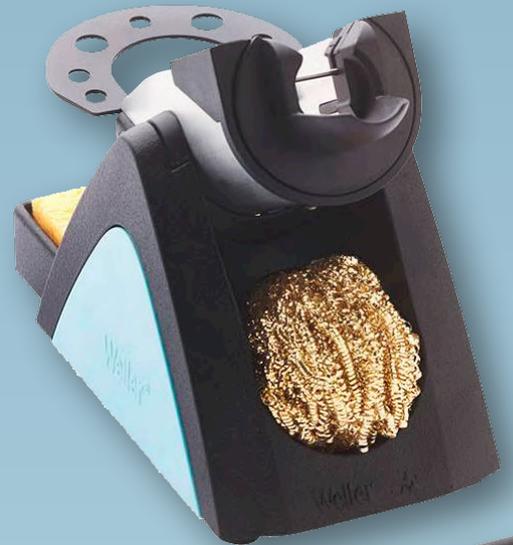
→ Kit de l'AOP 741 en format XL

[www.elektor.fr/the-xl741-discrete-op-amp-kit](http://www.elektor.fr/the-xl741-discrete-op-amp-kit)



banc d'essai

# station de soudage Weller WT 1014



Luc Lemmens (labo d'Elektor)

Dans la liste des appareils absolument indispensables à acquérir pour un labo d'électronique, les fers à souder et les stations de soudage à contrôle de la température sont incontournables. Si vous regardez de près ce que propose le marché, vous serez submergé par les marques et les types d'appareils disponibles, et même surpris par les énormes différences de prix des offres. La facture variera de quelques euros à peine pour un petit fer simple jusqu'à plus de 1000 € pour une station de soudage professionnelle. Que choisir ?

Pour ceux qui n'ont besoin de ce type d'outil que de temps en temps, un fer bon marché sera plus que suffisant. En revanche, pour les autres, plus assidus dans leurs réalisations électroniques, une station de soudage à température régulée s'imposera rapidement, avec un fournisseur connu et reconnu comme Weller. Elektor disposant en stock depuis quelque temps du produit WT 1014, j'ai été amené à évaluer cette station de soudage pour partager ensuite avec vous les enseignements que j'en ai tirés. Permettez-moi d'être franc : la marque Weller ne suscite pas chez moi un très grand enthousiasme. Je n'ai pas grand-chose à reprocher à ce fabricant en termes de qualité, mais

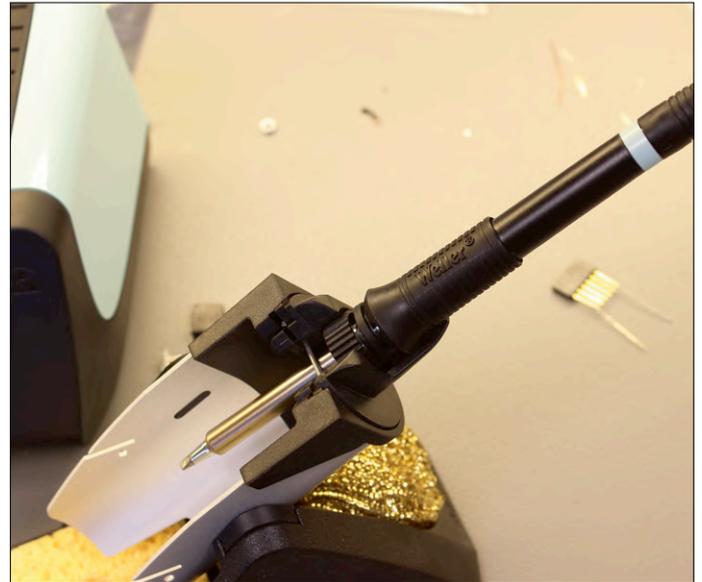
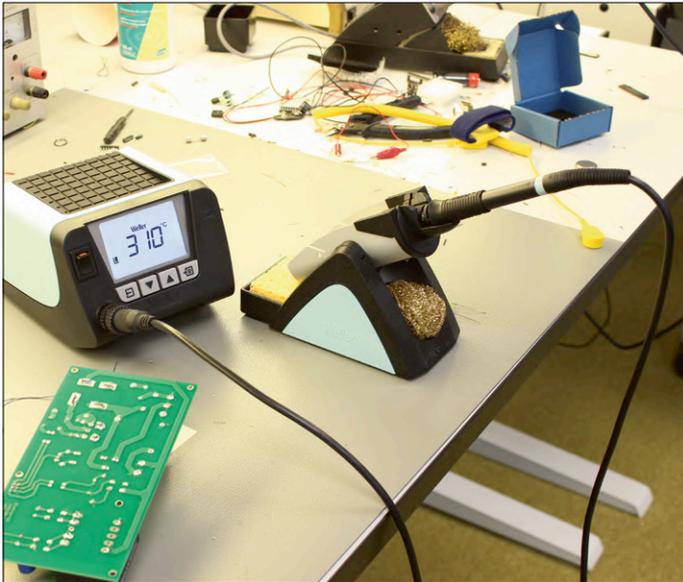
un coup d'œil aux prix peut rapidement inciter n'importe qui à chercher quelque chose de plus abordable.

En outre, les fers à souder de la marque que j'ai pu ou dû utiliser dans le passé étaient trop encombrants et lourds. Leur temps de chauffe relativement long m'agaçait. Mais, sait-on jamais, cette redécouverte m'amènera peut-être à changer d'avis...

La station de soudage WT 1014 se situe dans le milieu de gamme du catalogue de prix de Weller et se compose des éléments suivants :

- Station de soudage WT 1
- Fer à souder WSP-80
- Support de fer à souder WSR 201

La station WT 1 a été conçue pour six fers à souder différents dans la gamme Weller, parmi lesquels le WSP-80 est le plus performant en termes de puissance (80 W) et de taille. Il est également possible de connecter le fer à la station WT 1H de 120 W. Le support peut recevoir d'autres modèles de fers de cette gamme. La station WT 1014 vous est livrée dans une



grande boîte en carton où les trois éléments sont emballés séparément. Commençons donc par les déballer l'un après l'autre...

Vous trouvez d'abord le support, peut-être l'élément a priori le moins passionnant. Vous verrez pourtant qu'à l'utilisation il a une très grande importance. Le rôle d'un bon support est tout simplement de rester parfaitement en place lorsque vous déposez ou prenez en main votre fer. De plus, le fer à souder doit pouvoir entrer et sortir très facilement. Pour replacer votre fer dans le support, vous ne devez pas avoir à « viser ». Il est tout aussi pénible qu'un fer reste accroché lorsque vous voulez l'extraire pour vous en servir. Le WSR 201 a très bien réussi son examen de passage sur ces deux aspects. Le support est stable comme un roc et fonctionne parfaitement, sans restriction. Un logement idoine peut recevoir des pannes de réserve, une option commode pour éviter de les égarer. Pour les nettoyer, vous disposez à la fois d'une éponge et d'un tampon de laine de fer. Une configuration bien conçue, même si on ne pouvait en attendre moins d'un fabricant qui domine le marché depuis plus de 75 ans.

Le fer à souder lui-même (WSP-80) est étonnamment léger et maniable pour un modèle de 80 W. Son cordon est flexible, mince et d'une longueur suffisante. Parfait. Le WSP-80 porte une panne relativement large (type LT B), avec une pointe biseautée de  $2,4 \times 0,8$  mm, adaptée aussi bien pour les opérations de soudage méticuleuses ou plus lourdes. Néanmoins, pour le soudage plus délicat des composants CMS, j'envisagerais d'acheter immédiatement une panne étroite et pointue. L'arsenal des pannes de la série LT est impressionnant (voir en [1] les formes et dimensions). J'admire l'étendue de l'assortiment proposé par Weller, mais l'abondance de variantes fait perdre la vue d'ensemble.

Le prix des pannes est parfaitement acceptable, environ 10 € pour la plus chère. Si vous êtes prêt à prendre des risques, vous pouvez acheter, sur le marché asiatique, des jeux de dix pannes LT différentes pour 9 €, mais il est peu probable qu'il s'agisse d'accessoires d'origine Weller.

Pour changer de panne, aucun outil spécial n'est nécessaire. Pour la démonter sans vous brûler les doigts, vous devez dévis-

ser la grande bague de matière plastique, même si le fer est encore chaud. Il vous suffit ensuite de démonter la panne, de la remplacer par une autre et de revisser la bague ! Il me semble que pour cette opération de routine, un tampon de silicone n'aurait pas été superflu – d'autres marques le fournissent d'origine avec leur station de soudage. Comme les parties métalliques du fer mettent du temps à refroidir, la tentation est grande, pour éviter de se brûler, d'employer une pince. Même si vous y allez doucement, cela ne contribuera pas à l'allongement de la durée de vie de la panne ou de la douille. Sur le haut de la station de soudage, un compartiment permet de ranger les pointes à souder (chaudes). Son tapis amovible en caoutchouc de silicone noir pourrait être utilisé comme gantelet de protection. Simple suggestion...

À en juger par sa couleur et sa forme, la station de soudage WT 1 appartient à l'évidence à une gamme de produits Weller. Elle se compose d'un boîtier conséquent doté de solides boutons (mise en service/hors service, sortie, défilement bas, défilement haut et menu) et d'un écran LCD de bonnes dimensions avec rétroéclairage, bien lisible, comportant un éventail de paramètres et d'options bien plus étendu qu'on ne l'imaginerait. Les options vont d'ailleurs bien au-delà des besoins d'un utilisateur lambda. Le manuel fourni est volumineux, mais les toutes premières pages constituent un guide illustré qui ressemble à s'y méprendre aux modes d'emplois d'une entreprise suédoise, spécialiste des meubles à assembler soi-même... Pour le reste, il s'agit essentiellement d'instructions de sécurité et de détails relatifs aux garanties, dans différentes langues. Néanmoins, les illustrations donnent des informations suffisantes pour mettre immédiatement en service la station de soudage, même s'il est un petit peu regrettable que le manuel *Supplemental Operating Instructions* [2] ne soit pas fourni avec le produit. Celui-ci contient en effet la description détaillée du fonctionnement et des réglages de la station WT 1. Comme nous l'avons déjà mentionné, la plupart de ces fonctions n'ont aucune importance pour une utilisation normale. Pourtant, il n'y a aucun mal, par exemple, à savoir qu'il est possible de verrouiller l'appareil, et peut-être plus important encore, de savoir comment le déverrouiller.



Outre le connecteur du fer à souder en face avant, l'appareil comporte, à proximité de la base, une prise jack de 3,5 mm pour égaliser les potentiels (fiche de réserve livrée), ainsi qu'une prise RJ11 pour le système d'extraction Zero Smog de Weller. Le cordon d'alimentation secteur se branche à l'arrière.

Après la mise en service, l'un de mes préjugés à propos de Weller a été immédiatement battu en brèche : bien que le produit ne soit pas possédé par le démon de la vitesse, le fer à souder a atteint sa température de fonctionnement dans un temps tout à fait acceptable. Les premiers essais de soudage ont été concluants. J'ai eu la surprise de constater que le fer s'était refroidi le temps que je rédige quelques phrases pour ce banc d'essai. Vous ne remarquez le phénomène que lorsque vous commencez à essayer de souder. En réalité, la station de soudage possède une mise en veille ajustable. Lorsque la durée établie est atteinte, la WT 1 commande le fer à souder en diminuant la température (également réglable) pour prolonger la durée de vie de la panne. Pour quitter le mode veille, vous devez appuyer brièvement sur le bouton de défilement « haut » ou « bas » du fer à souder WSP-80. Si vous laissez s'écouler un temps encore plus long, la station WT 1 éteint complètement le fer et vous devez alors appuyer sur les deux boutons pour pouvoir le remettre en service. Ce délai est également réglable, bien entendu.

Weller dispose de fers dotés d'un détecteur de mouvement qui indique à la station si l'appareil est utilisé. Ainsi, l'intervalle de mise en veille est en permanence réinitialisé ou l'état de veille est automatiquement annulé si quelqu'un actionne le fer à souder. Malheureusement, le WSP-80 ne dispose pas d'un tel capteur et il n'y a donc pas d'alternative au réveil périodique de la station en mode manuel. Vous pouvez bien



entendu augmenter l'intervalle de mise en veille (jusqu'à 99 min max.) pour limiter les effets de cet inconvénient, mais gardez à l'esprit que cette fonction a une raison d'être : prolonger la vie de votre panne. En dépit de ce point mineur, la WT 1014 a parfaitement fonctionné.

### Conclusion

La station de soudage WT 1014 a complètement changé mon point de vue sur Weller. J'ai trouvé le fer à souder agréable à manier et parfaitement opérationnel, même si une panne plus fine pour les opérations les plus délicates (composants CMS) ne serait pas superflue. Il existe de nombreux types de pannes de soudage, dans toutes les dimensions, proposées à des prix raisonnables. De plus, avec un fabricant comme Weller, vous pouvez tabler sur des pièces (de rechange) disponibles pour de nombreuses années. Ensemble, le support, la station et le fer à souder donnent une impression de fiabilité, avec une ergonomie bien pensée. Même s'il faut admettre que le produit n'est pas particulièrement abordable, vous disposerez, en installant la WT 1014 dans votre labo personnel, d'une excellente station de soudage, prête à vous rendre service pour de nombreuses années.

(190074-04)



@ WWW.ELEKTOR.FR

→ Station de soudage Weller WT 1014  
[www.elektor.fr/18649](http://www.elektor.fr/18649)

→ Support pour circuit imprimé Weller ESF 120ESD  
[www.elektor.fr/17481](http://www.elektor.fr/17481)

### Liens

[1] Pannes dans le catalogue Weller :

[https://media-weller.de/weller/data/ebooks/epaper-Weller\\_Catalogue\\_2016-17\\_EN/index.html#108](https://media-weller.de/weller/data/ebooks/epaper-Weller_Catalogue_2016-17_EN/index.html#108)

[2] Manuel 'Supplemental Operating Instructions' : [https://media-weller.de/weller/data/OI/OI/OI\\_WT1\\_WT1H\\_supplementary.pdf](https://media-weller.de/weller/data/OI/OI/OI_WT1_WT1H_supplementary.pdf)

# projet SCCC (2)

## processeur softcore et compilateur C à construire soi-même

Martin Oßmann

Le simulateur du CPU de fabrication maison exécute et teste les programmes en C, même en l'absence de puce FPGA. De plus, si vous souhaitez étendre le processeur, vous pouvez tester le tout avant de l'implémenter dans la FPGA. Dans cette seconde partie, nous verrons en détail comment le compilateur transpose notre code C en un listage en assembleur.

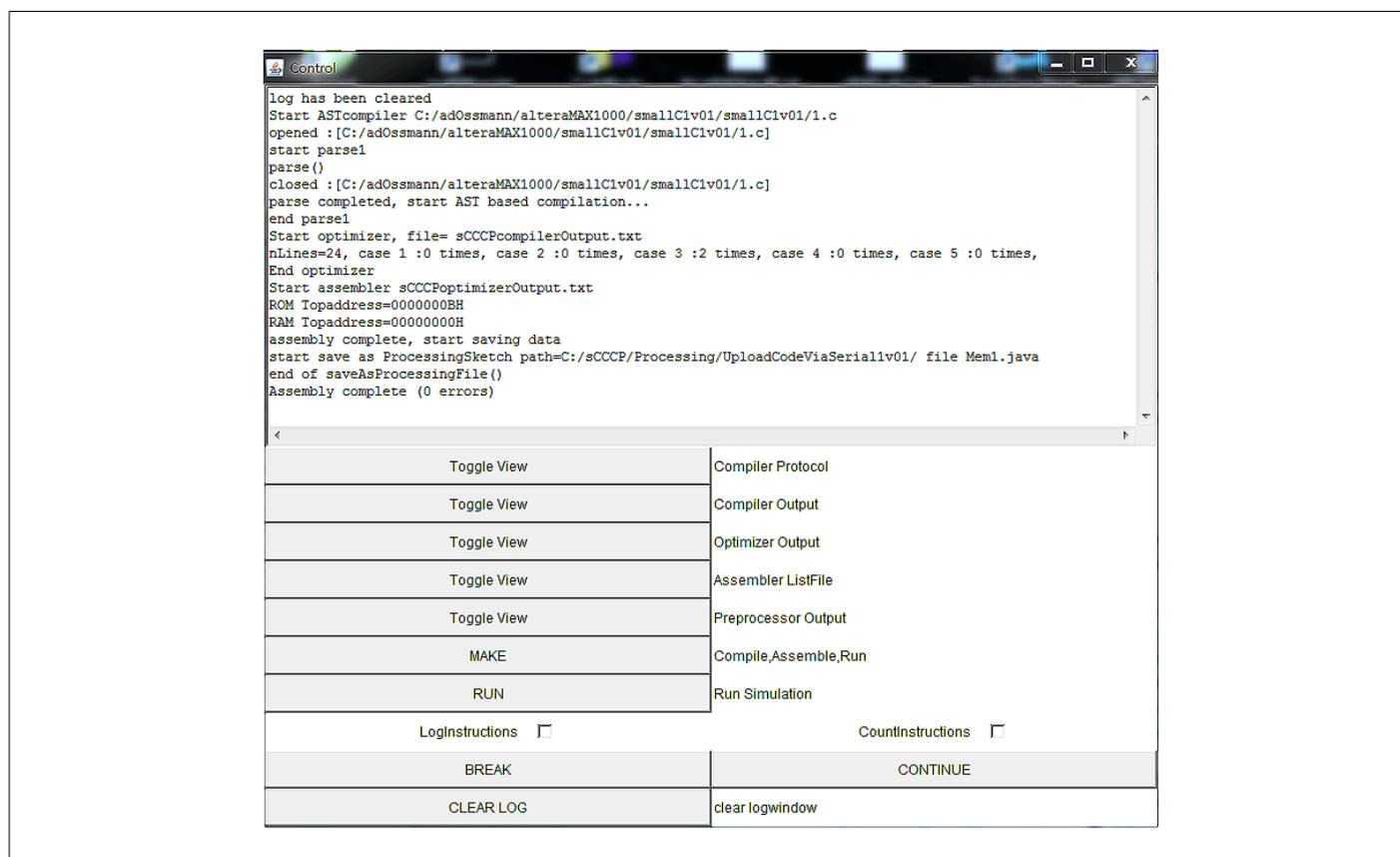
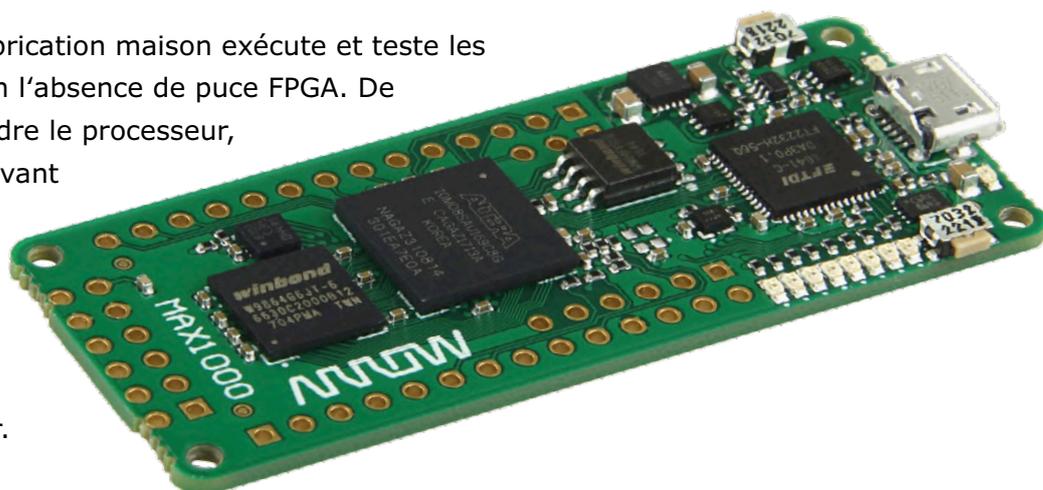


Figure 1. L'interface graphique de l'environnement SCCC.

Dans la première partie de cette série [1], nous avons travaillé dans l'environnement de développement sCCCP capable de compiler les programmes en C du compilateur à construire soi-même (**fig. 1**). L'environnement comprend aussi un simulateur de CPU écrit en langage de programmation Java, exactement comme le compilateur. Avec ce simulateur, il est aisé de découvrir le langage de programmation tinyC (un sous-ensemble du C).

### Expérience 3 : simulation dans l'environnement sCCCP

Nous arrivons à la troisième expérience de cette série ; dans la première partie, nous avons décompacté l'archive logicielle [2] du dossier de projet sCCCP/ et l'avons placé sur un disque local. Ouvrez une fenêtre de commande sur la zone de travail C:/sCCCP/experiment3 et exécutez-y le fichier de commande par lot (fichier *batch*) *c.bat*, qui démarre l'environnement sCCCP. Le fichier *primes.c* fournit l'application du jour, un programme qui détermine les nombres premiers inférieurs à 20. Le bouton **MAKE** lance la compilation du programme. Il ne devrait pas y avoir d'erreur. Cliquez ensuite sur le bouton **RUN** pour exécuter le programme et les nombres premiers commencent à s'afficher dans la fenêtre de sortie. Avant d'appuyer sur **RUN**, si nous cochons la case **LogInstructions**, nous obtenons un journal détaillé de ce que le CPU fait. Une fois le programme terminé (ou après interruption avec le bouton **BREAK**), nous obtenons la liste des instructions utilisées avec mention de leur fréquence d'utilisation. L'auteur a trouvé cela utile pour mettre en évidence si l'optimisation de certaines instructions pouvait se justifier.

### Options

Arrivé à ce point, il est temps d'aborder les différentes options de l'environnement de développement sCCCP. Jusqu'à présent, nous n'avons pratiquement utilisé que le bouton **MAKE** pour compiler notre code source et implanter le code objet à l'endroit voulu. Les fonctions **RUN**, **BREAK** et **CONTINUE** servent à piloter le simulateur de CPU.

Si tout se déroule comme prévu, il n'y a besoin de rien d'autre. Mais, le plus souvent, tout ne se passe pas aussi bien que souhaité. Pour traquer les erreurs, nous pouvons examiner le résultat de chaque étape de travail.

La première est celle du préprocesseur qui parcourt le fichier source et effectue les macroremplacements nécessaires. Si nous souhaitons voir le résultat de cette étape, nous cliquons sur le bouton **Toggle View** voisin de **Preprocessor Output**.

Ensuite, à partir du code source, l'analyseur récursif *Recursive Descent Parser* établit un arbre syntaxique abstrait (ou AST = *Abstract Syntax Tree*). Le bouton **Toggle View** voisin de **Compiler Protocol** donne accès à ce que le *Parser* considère comme important. Pour finir, une représentation de l'AST apparaît.

Une fois l'AST créé, nous passons à la phase de production du code. Le code produit est accessible par le bouton **Toggle View** voisin de **Compiler Output**.

Ensuite, un *Peephole Optimizer* (optimiseur de code local) très simple prend le relais. Ce dernier reconnaît des séquences de code spécifiques composées de plusieurs instructions et remplace ces instructions par une instruction plus efficace : la sortie qui en résulte apparaît en cliquant sur le bouton **Toggle View** voisin de **Optimizer Output**.

Une dernière phase de production de l'assembleur termine le

processus. Là aussi, le résultat apparaît avec le bouton **Toggle View** voisin de **Assembler ListFile**. Le programme assembleur peut effectivement détecter de nombreuses erreurs et les signaler lorsqu'elles se produisent. Il faut examiner le listage de l'assembleur pour remédier à la cause de cette erreur. La déclaration multiple d'une même variable est fréquemment à l'origine de telles erreurs.

Le compilateur crée les fichiers suivants :

- sCCCPAssemblerListing.txt
- sCCCPcompilerOutput.txt
- sCCCPoptimizerOutput.txt

Ces fichiers texte sont le produit des différentes étapes de travail et peuvent donc être examinés après coup.

### Expérience 4 : simulation en Java et Verilog

Si vous souhaitez comparer les différentes simulations, vous devriez tenter l'expérience 4. Pour cet essai, le dossier de tra-

#### Listage 1. Les fonctions du sous-ensemble C.

```
code source                                n° de ligne
int x,y,*xp,xa[10] ;                        00000
                                           00001
                                           00002
setLEDs(int p){                             00003
    p ; //fetch p to R0                      00004
#asm                                         00005
    OUTA 14 ; out R0 to LEDs                00006
#endasm                                     00007
}                                           00008
                                           00009
outString(int *p){                          00010
    while(*p){                              00011
        setLEDs(*p) ;                      00012
        p++ ;                               00013
    }                                       00014
}                                           00015
                                           00016
funA (int px,int py){                      00017
    int localxp ;                          00018
    localxp=&x ;                            00019
    return px<                             00021
}                                           00022
main(){                                     00023
    outString("1234") ;                    00024
    xa[2]=xa[3]+funA(12,34) ;              00025
    y=(x=1) ;                               00026
    if(x>=y) {                              00027
        y=1 ;                               00028
    }                                       00029
    else{                                    00030
        y=2 ;                               00031
    }                                       00032
}                                           00035
```

## Listage 2. Analyse d'une instruction conditionnelle.

```
ASTifNode ifStatement() throws CompilerException
{
    // generate a new node with 2 or 3 subnodes:
    // subnode 1 : condition
    // subnode 2 : if-part statement1
    // subnode 3 : else-part statement optional

    if( ! tccScan.sbTestWordAndSkip("if") )
    {
        // exit if "if" not found
        return null ;
    }

    ASTifNode thisNode=new ASTifNode() ;

    ASTnode cond=condition() ; // get a new node for this statement
    // test for condition
    if(cond==null){Environment.error("condition missing") ; }
    // if not found: error
    thisNode.addSubNode(cond) ; // add condition to the new node
    ASTnode st1=needStatement() ;
    // we need a statement, error if not found
    thisNode.addSubNode(st1) ; // add this statement to the new node
    if( ! tccScan.sbTestWordAndSkip("else") )
    {
        // test for "else"
        return thisNode ; // if no "else" we are done
    }
    ASTnode st2=needStatement() ; // we need the else part statement
    thisNode.addSubNode(st2) ; // add to new node
    return thisNode ; // return this node
}
```

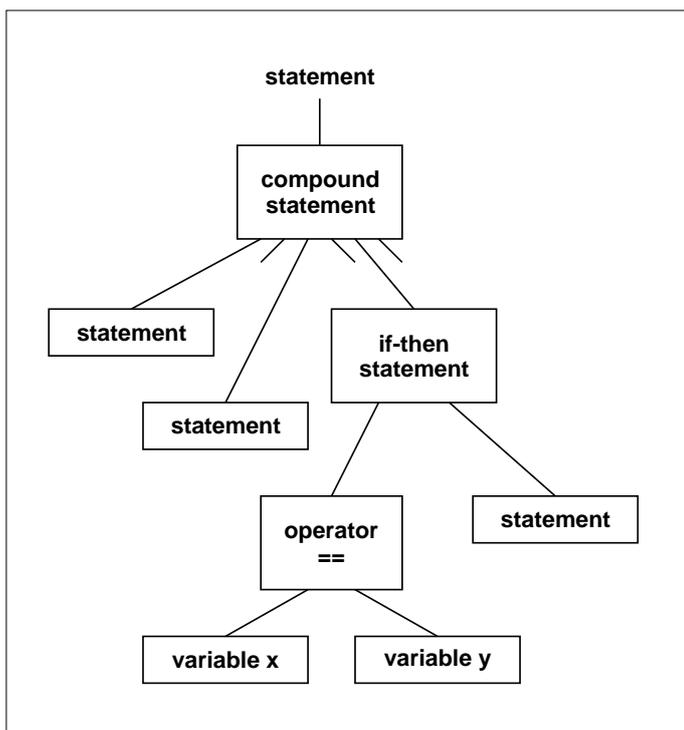


Figure 2. Extrait de l'AST (Abstract Syntax Tree).

vail est `C:/sCCCP/experiment4/experiment4/`.

Nous ouvrons la fenêtre de commande sur ce dossier et lançons l'environnement sCCCP en exécutant `c.bat`. Simultanément, pour effectuer des modifications, nous pouvons ouvrir le fichier `primes1.c` à l'aide d'un éditeur quelconque. Le bouton `MAKE` de notre environnement sCCCP lance la compilation de ce fichier ; le bouton `RUN` lance la simulation dans l'environnement sCCCP. Comme dans cette version du fichier batch `c.bat`, l'option `-i c:\sCCCP\IcarusVerilog` est utilisée, le code objet créé est également copié dans le dossier de travail Icarus/Verilog (voir [1]). Dans ce dernier, le simulateur Icarus peut démarrer le programme en lançant à nouveau les fichiers `batch c.bat` (compilation) et `r.bat` (exécution). Ce banc d'essai est pour l'instant paramétré pour s'arrêter au bout de 80 000 pas dans le simulateur. Si votre propre programme a besoin de plus de pas de programme, il faut augmenter cette limite.

### Le langage de programmation tinyC

Si le cœur vous en dit, vous pouvez d'ores et déjà lancer des simulations de différents programmes applicatifs et examiner les résultats de cette suite d'outils. Pour cela, nous n'avons pas besoin de la carte MAX1000 [3]. Comparée à la MAX10-FPGA, elle possède aussi 8 Mo de mémoire Flash ainsi que 8 Mo de RAM. Elle sera de nouveau utilisée dans la troisième

### Listage 3. Compilation d'une instruction conditionnelle.

```
void compile() throws CompilerException {
    logVisit(this) ;
    label1=Environment.getNewLabel();
    label2=Environment.getNewLabel();
    subNodes.elementAt(0).toHL();
    Environment.assembler.listln("          JFALSE          "+Environment.label(label1)) ;
    subNodes.elementAt(1).compile();
    if(subNodes.size()<3){
        Environment.assembler.listln(Environment.label(label1)+" EQU $" ) ;
    }
    else {
        Environment.assembler.listln("          JMP          "+Environment.label(label2)) ;
        Environment.assembler.listln(Environment.label(label1)+" EQU $" ) ;
        subNodes.elementAt(2).compile();
        Environment.assembler.listln(Environment.label(label2)+" EQU $" ) ;
    }
}
```

partie de cette série.

Essayez simplement de bien assimiler les éléments de notre sous-ensemble C. Le plus simple est de se faire une idée générale en examinant attentivement le contenu du programme exemple du **listage 1**. Il ne fait rien de spécial, mais est correct du point de vue de la syntaxe et montre un aperçu des fonctions réalisées.

Au premier coup d'œil, nous pouvons voir qu'un grand nombre de constructions sont traitées. Rappelons quelques points souvent négligés, mais importants. Le seul type de donnée pris en charge est *signed 32 Bit*. Donc les variables de type *char* sont stockées dans des mots de 32 bits. C'est bien du gaspillage, mais cela simplifie prodigieusement le compilateur et le CPU. Par ailleurs, les tableaux n'ont qu'une dimension et ne peuvent contenir que des entiers.

Pour ce qui concerne les fonctions, il est implicite qu'elles retournent une valeur. Le modèle du paramètre retourné n'est pas spécifié. Les opérateurs +=, -=, etc. ainsi que && et || ne sont pas implémentés.

De plus, #asm et #endasm permettent d'inclure directement du code en assembleur. C'est important si nous voulons connecter le CPU à un autre matériel. Globalement, l'indispensable permettant de programmer des applications embarquées est présent (sans calcul en virgule flottante).

### Structure du compilateur

Si l'étendue du langage est insuffisante, ou si des instructions spéciales sont nécessaires, il est possible d'adapter soi-même le compilateur. C'est l'une des vertus du code ouvert. Voici quelques conseils pour exploiter ce compilateur.

Comme déjà mentionné, le texte source est traité puis transformé en arbre syntaxique ou AST (*Abstract Syntax Tree*) par l'analyseur récursif *Recursive Descent Parser*. Chaque construction du langage est affectée à une routine de l'analyseur, laquelle vérifie si la structure correspondante est présente. La routine `statement()` teste par ex. si une instruction est présente et si oui laquelle. Une instruction (*statement en anglais*) est un bloc de code (en C la plupart du temps, encadré par des accolades) comprenant une ou plusieurs sous-instructions.

La fonction accède ici à la routine `ifStatement()`. En outre, `ifStatement()` utilise elle-même la fonction `statement()` pour analyser la construction *if-then-else*. En cas de réussite, les routines de test syntaxique renvoient un nœud (ASTNode) lequel permet de construire l'arbre.

Le segment de code du **listage 2** montre l'implémentation résultante. S'il n'y a pas de *if* dans la suite du texte, la routine retourne un zéro, cela veut dire que l'analyse de l'instruction *if* a échoué.

La fonction d'analyse `condition()` vérifie si une condition apparaît après dans le texte. Si ce n'est pas le cas, il y a une erreur de syntaxe. Le nœud retourné par `condition()` est intégré dans les nœuds de l'AST. Il doit ensuite y avoir une autre instruction, elle est à son tour intégrée dans les nœuds. Enfin, optionnellement, une instruction *else* avec des sous-instructions peut suivre. Si c'est le cas, ce *else* est également intégré dans les nouveaux nœuds. Ensuite, si l'instruction *if-then-else* est traitée sans erreur, le nœud y afférent est retourné à l'instruction d'appel. L'arbre est ainsi construit pas à pas. La **figure 2** montre à quoi une partie d'un arbre peut ressembler.

### Listage 4. Code source en C et listage en assembleur correspondant.

```
    ;
    ; if(x) { y=1 ; } else { y=2 ; }
    ;
                                LD          @ __x
                                JFALSE     __LBL00000
                                LDI        1
                                ST         @ __y
                                JMP        __LBL00001
__LBL00000 EQU $
                                LDI        2
                                ST         @ __y
__LBL00001 EQU $
```

### Listage 5. Expressions arithmétiques simples et leur transposition en assembleur.

```
-----  
;  
; x=y  
LD @ __y ; R0 = y  
ST @ __x ; x = R0  
-----  
;  
; x=x+y*5  
LDPUSH @ __x ; push x onto stack  
LDPUSH @ __y ; push y onto stack  
LDI 5 ; R0 becomes 5  
POP R1 ; y popped from stack to R1  
IMUL ; R0=R0*R1 = 5*y  
POP R1 ; x popped from stack to R1  
IADD ; R0=R0+R1=y*5+x  
ST @ __x ; store R0 on address x  
-----  
;
```

### Production du code

Une fois l'arbre construit, il est parcouru pour produire le code. Pour cela, chaque nœud de l'AST possède la méthode `compile`. Le listage 3 donne une idée de ce que la méthode `compile` contient pour le nœud `ASTifNode`.

En premier lieu, deux étiquettes (label) sont chargées pour référence ultérieure. `subNodes.elementAt(0).toHL` sert à produire le code correspondant à la condition. Ensuite, un saut conditionnel est créé, il permet de sauter le code correspondant au cas if. `subNodes.elementAt(1).compile` produit le code correspondant au cas if. S'il n'y a pas de else, une étiquette est finalement définie. S'il y a un else, le code correspondant doit être créé.

C'est plus facile à comprendre avec un exemple. Le listage 4 montre quel code a été produit pour l'instruction `if(x) else`. Si vous souhaitez examiner d'autres exemples de production de code : compilez le fichier `C:/sCCC/demo6/demo6/demo6.c` et examinez ensuite le listage en assembleur. Nous voyons ici (lignes de commentaires) le code source en C suivi à chaque fois du code objet.

Le listage 5 est un exemple qui montre comment des expressions arithmétiques simples sont transposées en assembleur. Pour l'instruction simple d'affectation `x=y`, la valeur de `y` est d'abord chargée dans le registre `R0`. Ensuite, la valeur de `R0` est transérée en mémoire à l'adresse de `x`. Au-dessous figure la transposition de l'expression `x=x+y*5`. Les résultats intermédiaires sont chargés dans la pile. Le traitement d'expressions arithmétiques constitue par ailleurs la majeure partie du code

du compilateur.

### Perspectives

Dans cette seconde partie de la série, nous nous sommes limités à de la simulation. Dans le prochain article, nous ferons de nouveau appel à la carte MAX1000. Nous connecterons la petite carte de développement au PC via une ou deux interfaces série pour télécharger le programme et réaliser des entrées/sorties série. ◀

(180394-B-04 – version française : Yves Georges)



### Liens

- [1] « projet SCCC (1) », Elektor 03-04/2019 : [www.elektormagazine.fr/180394-04](http://www.elektormagazine.fr/180394-04)
- [2] Page de l'article : [www.elektormagazine.fr/180394-B-04](http://www.elektormagazine.fr/180394-B-04)
- [3] Carte de développement FPGA MAX1000 : [www.elektor.fr/max1000](http://www.elektor.fr/max1000)



@ WWW.ELEKTOR.FR

→ Carte de développement FPGA MAX1000  
[www.elektor.fr/max1000](http://www.elektor.fr/max1000)

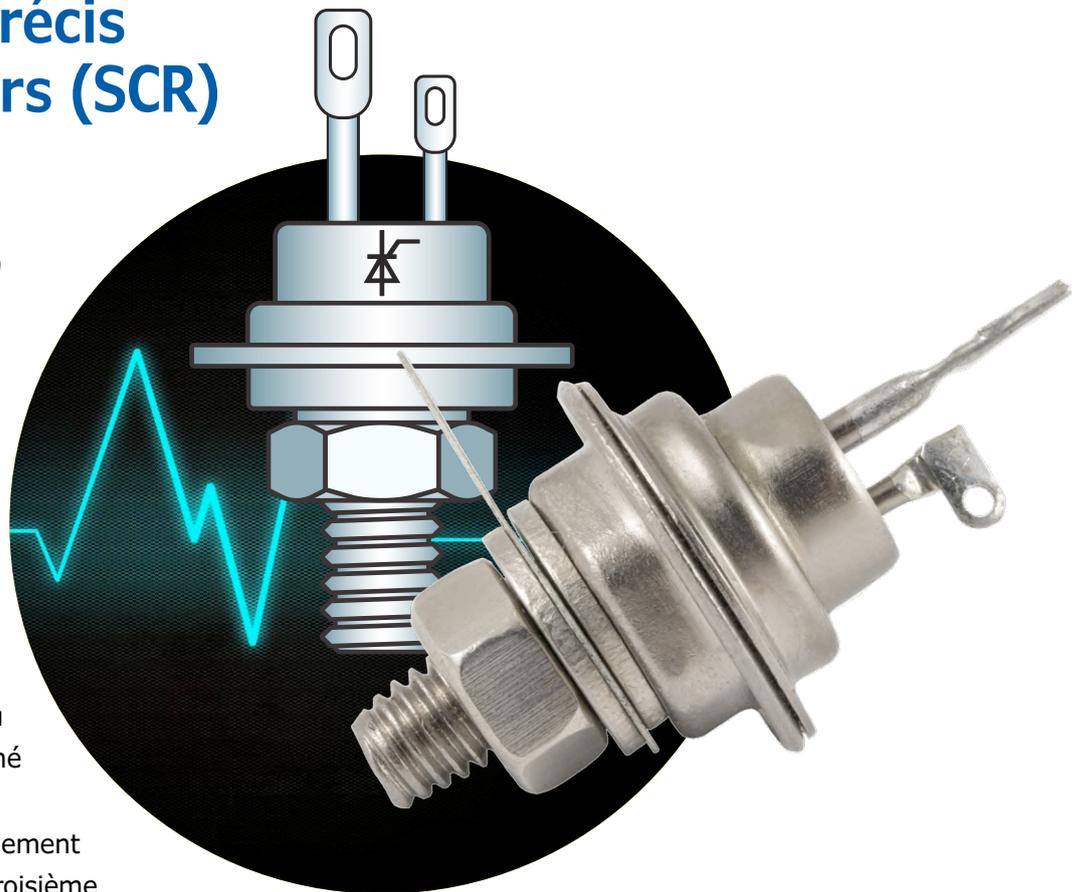
→ Livre en anglais, 'Microprocessor Design Using Verilog HDL'  
[www.elektor.fr/verilog](http://www.elektor.fr/verilog)

# générateur d'impulsions de gâchette

## amorçage précis des thyristors (SCR)

Ton Giesberts (labo d'Elektor)

À la fin des années 50 sont apparus des composants pour applications de puissance, les SCR, de *Silicon Controlled Rectifier*, redresseurs commandés au silicium ou thyristors, mot-valise formé de thyr(atron) et (trans)istor. Il s'agit fondamentalement d'une diode dotée d'une troisième électrode, la gâchette de commande. Pour amorcer un thyristor, rendre la diode passante, il faut une petite impulsion sur sa gâchette, dite aussi grille. Elle peut intervenir à un moment choisi au cours de l'alternance, avec un certain retard de phase, ce qui permet de modifier la durée de passage du courant, donc son intensité moyenne. La conduction ne s'arrête que quand le courant de maintien s'annule, ce qui se fait tout seul sous tension alternative.



Le circuit présenté ici est un prototype de module de commande pour thyristor, à adapter à ses besoins et à protéger efficacement des tensions dangereuses dans un boîtier ad hoc. Il a été conçu pour des expérimentations sur des circuits économes en énergie, à savoir les cartes de l'article « thyristor MOSFET à double anode » également publié dans ce numéro.

### Le circuit en opération

Dans le schéma de la **figure 1**, le photocoupleur IC1, un CNY65, sert de détecteur de passage par zéro de la tension alternative appliquée sur l'entrée, à savoir K1. Deux résistances de 100 k $\Omega$ , R1 et R2, commandent la LED du photocoupleur. Bien entendu, si la tension sur K2 est

plus basse, il faut adapter les valeurs de ces résistances. Par sécurité, le photocoupleur CNY65 est conforme à l'écartement requis de plus de 6 mm entre la partie haute tension (HV) et le côté basse tension (LV) du circuit imprimé. Quand on se sert de la sortie +HV du redresseur pleine onde D1 à D4 (K1), il peut être nécessaire de relier LV à HV en **installant une résistance de 0  $\Omega$  ou un cavalier à l'endroit indiqué R3. Mais n'oubliez pas que dès lors, tout le circuit sera soumis à la tension du secteur, jusqu'au connecteur K4 !** Le signal à la sortie du photocoupleur est inversé, tamponné et amplifié par le transistor T1, un BC547B. Il précède IC2, un comparateur double LM393 qui a une plage d'entrée en mode commun de 0

## INFOS SUR LE PROJET



test et mesures  
SCR thyristor



débutant  
→ **connaisseur**  
expert



env. 1 h



fer à souder



env. 20 €

à 1,5 V (à 25 °C) sous le niveau de la ligne d'alimentation positive. Si l'une des entrées est à l'intérieur de cette plage de mode commun et l'autre au-dessus, il n'y aura pas d'inversion de phase sur la sortie à collecteur ouvert.

Les comparateurs IC1A et IC1B prennent tous deux comme référence une tension de 2,5 V, bâtie par le diviseur R11/R12 et découplée par C3. Avec l'ajustable P1 et le condensateur C1, on règle la constante de temps RC de l'impulsion. On n'aurait besoin de C2 que s'il fallait plus de précision ou pour adapter le circuit à un sec-

teur à 60 Hz. N'oubliez pas que la plupart des potentiomètres sont entachés d'une imprécision de 20 %. Ici, la constante de temps  $\tau$  (tau) vaut :

$$\tau = (C1 + C2) \times P1 \quad [s]$$

La diode D6 décharge le condensateur C1 juste après le passage par zéro de la tension d'entrée. La fréquence de sortie sur +HV du redresseur pleine onde (K1) est de 100 Hz pour le réseau à 50 Hz branché sur K2. Cette tension de sortie sur K1 sert à alimenter en puissance un circuit à thyristor. Pour rendre le circuit d'emploi plus général, le connecteur K4 fournit, en plus des impulsions de commande, l'alimentation en +5 V en provenance d'un bloc extérieur.

**Il faut noter encore que si la résistance R3 de 0  $\Omega$  est installée sur le circuit imprimé, cette alimentation de +5 V sera aussi soumise au potentiel du secteur relié à K2 et qu'il ne faut pas négliger les protections nécessaires pour éviter un contact direct avec n'importe quelle partie du montage. Donc, débranchez-le du secteur avant chaque intervention.**

La tension de référence de 2,5 V est également présente sur un contact de K4, lequel peut servir d'entrée ou de sortie, ce qui peut être utile dans certaines applications. K3 est prévu pour ajouter un potentiomètre extérieur, mais uniquement un modèle en plastique ! Mieux vaut alors ôter l'ajustable qui serait en parallèle. Préférez toutefois cet ajustable seul.

J'ai ajouté l'étage de sortie T2 pour assurer la chute jusqu'à 0 V de la tension de sortie. Prendre une faible valeur pour la résistance de rappel vers le haut à la sortie de IC2B pour commander la grille d'un thyristor laisse inmanquablement un résidu de tension en sortie à cause de la sortie à collecteur ouvert du comparateur. Bien que faible, cette tension peut maintenir le thyristor en conduction. La valeur de 330  $\Omega$  de R9 assurera à la plupart des thyristors un courant de gâchette suffisant avec plus de 10 mA disponibles.

### Construction de l'appareil

À priori, le circuit imprimé à simple face dont on voit les pistes de cuivre inférieures a l'air un peu grand, mais il respecte les règles de sécurité édictées, telles que l'écart minimal de 3 mm là où la tension du secteur est présente. Dans la partie réservée aux composants directement soumis à la haute tension, toutes les pistes sont distantes de 6 mm des bords de la carte, pour raison de sécurité. La rigidité diélectrique du CNY65 est spécifiée à 13,9 kV pendant 1s. ◀

(180611-03

version française : Robert Grignard)

 @ [WWW.ELEKTOR.FR](http://WWW.ELEKTOR.FR)

→ Circuit imprimé nu du générateur d'impulsions de gâchette, réf. 180611-1  
[www.elektor.fr/180611-1](http://www.elektor.fr/180611-1)

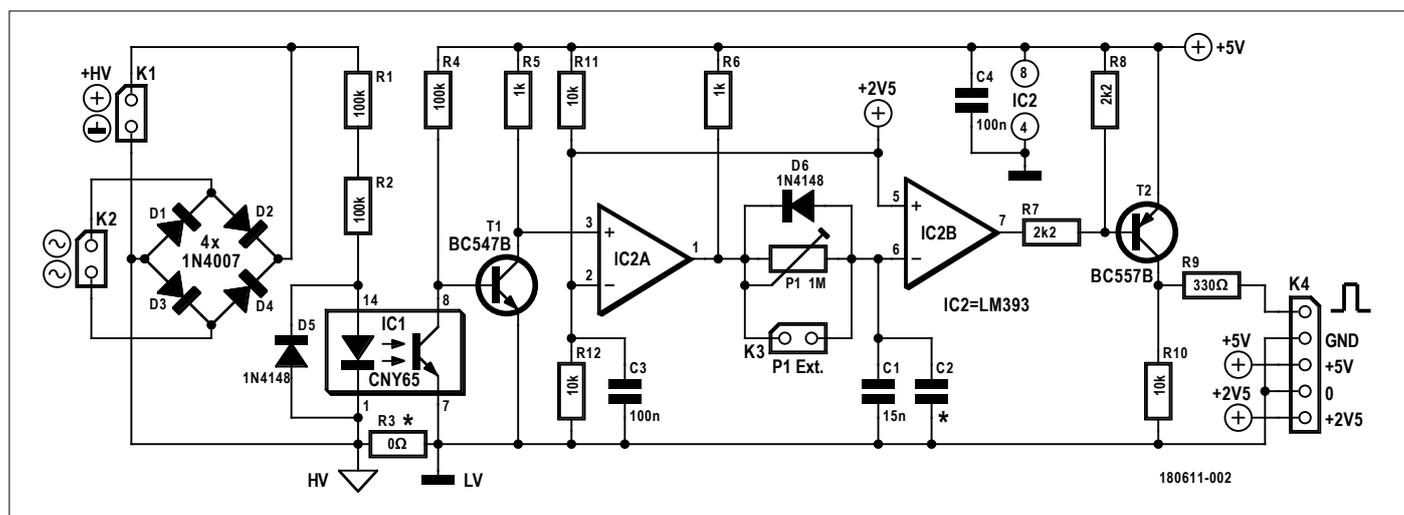


Figure 1. Ce petit circuit fait fonctionner les thyristors en toute sécurité en conditions réelles, pour un test (avec prudence !) ou pour toujours.



## LISTE DES COMPOSANTS

### Résistances

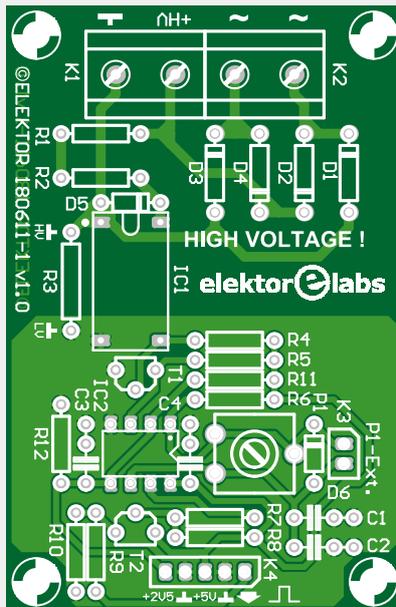
R1, R2, R4 = 100 k $\Omega$ , 5 %, 0,25 W, 250 V  
 R3 = 0  $\Omega$ , 5 %, 0,25 W (ou cavalier) cf. texte  
 pour règles de sécurité importantes  
 R5, R6 = 1 k $\Omega$ , 5 %, 0,25 W  
 R7, R8 = 2,2 k $\Omega$ , 5 %, 0,25 W  
 R9 = 330  $\Omega$ , 5 %, 0,25 W  
 R10, R11, R12 = 10k $\Omega$ , 5 %, 0,25 W  
 P1 = 1 M $\Omega$ , potentiomètre d'ajustage,  
 horizontal

### Condensateurs

C1 = 15 nF, 100 V, 10 %, PET,  
 au pas de 5,0/7,5 mm  
 C2 = normalement non monté, cf. texte  
 C3, C4 = 100 nF, 50 V, 10 %, X7R,  
 au pas de 5,0/7,5 mm

### Semi-conducteurs

D1 à D4 = 1N4007, DO-41  
 D5, D6 = 1N4148, DO-35  
 T1 = BC547B, TO-92  
 T2 = BC557B, TO-92  
 IC1 = CNY65, DIP-4 HV  
 (L x l = 17,8 mm x 9,6 mm)  
 IC2 = LM393P, DIP-8



### Divers

K1, K2 = bornier à 2 vis, au pas de 7,62 mm,  
 tension nominale 630 V  
 K3 = embase à 2 picots, au pas de 2,54 mm,  
 vertical



K4 = embase à 5 picots, au pas de 2,54 mm,  
 vertical  
 support DIL à 8 contacts, au pas de 2,54 mm,  
 17,8 mm de large (en option pour IC2)  
 circuit imprimé réf. 180611-1 (www.elektor.fr)

Publicité

## Vous souhaitez publier votre montage dans le magazine ?

Rendez-vous sur la page du labo d'Elektor : [www.elektormagazine.fr/labs](http://www.elektormagazine.fr/labs) pour y enregistrer votre projet.

Cliquez sur « Créer un projet ».

Connectez-vous (créez un compte gratuit si vous n'en avez pas encore).

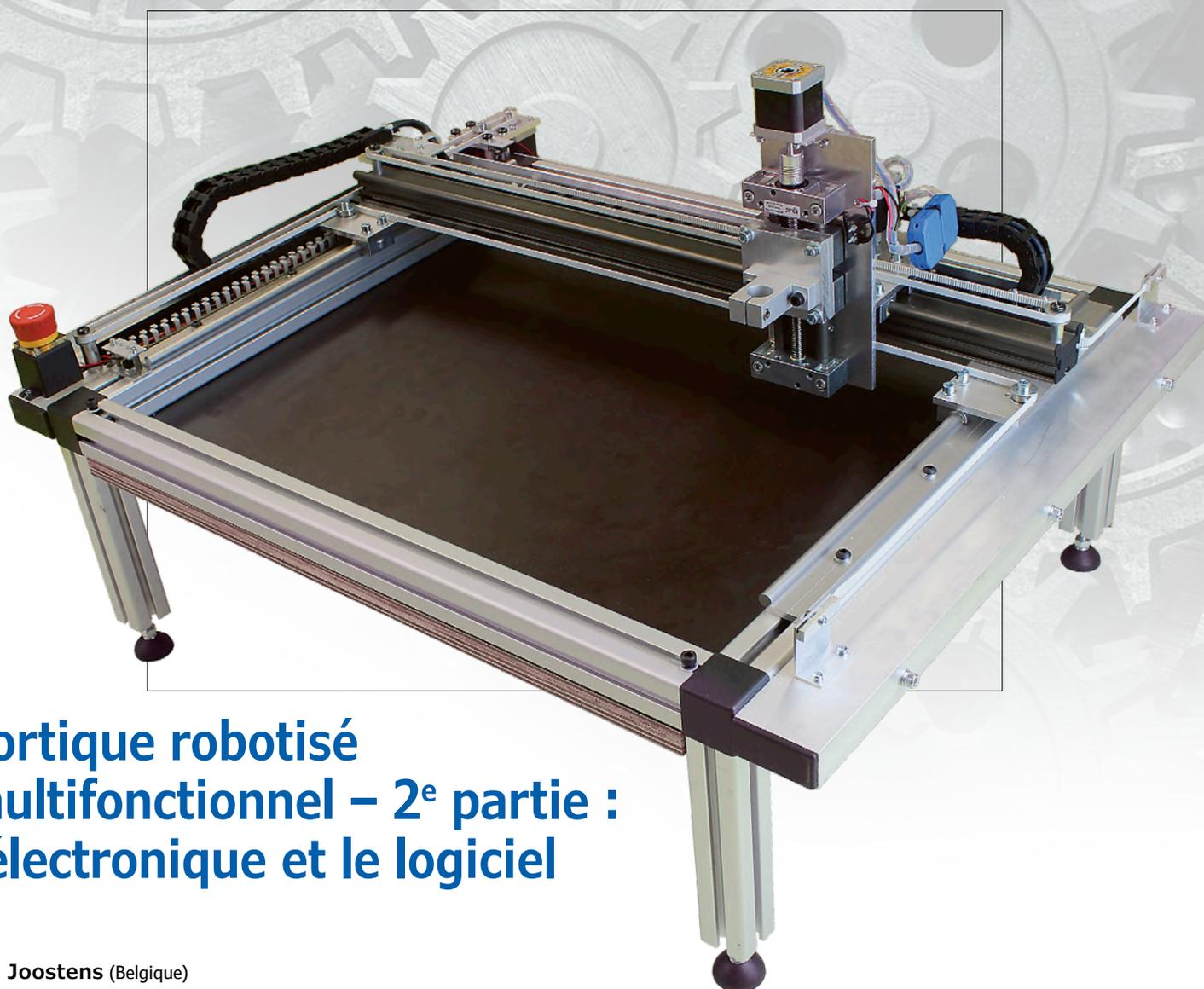
Remplissez les différents champs du formulaire.

Votre proposition de montage sera examinée par l'ensemble des rédacteurs du magazine. Si votre projet est retenu pour sa publication dans le magazine, un rédacteur prendra contact avec vous pour vous accompagner dans la rédaction de l'article.

## Labo d'Elektor : [www.elektor-labs.com](http://www.elektor-labs.com)

Découvrir, créer et partager... l'électronique !

# machine-outil à commande numérique Wolverine



## portique robotisé multifonctionnel – 2<sup>e</sup> partie : l'électronique et le logiciel

**Ilse Joostens** (Belgique)

Dans la 1<sup>ère</sup> partie de cet article (magazine de mars/avril 2019 [1]), nous avons décrit les aspects mécaniques de notre machine-outil à commande numérique par ordinateur (CNC). Il est temps de passer à l'électronique et au logiciel de commande. Quelques projets simples vous permettront de prendre l'appareil en main et de découvrir divers programmes de CAO, pour ensuite laisser libre cours à votre imagination.

Lors de la construction de notre premier prototype, nous avons passé en revue divers circuits électroniques de commande numérique avec leur logiciel associé, et nous en avons testé quelques-uns. Nous préférons en général utiliser des matériel et logiciel libres de droits, mais l'offre en matière de CNC ne nous a pas séduits. Une collection hétéroclite de cartes reliées par un fatras de câbles, ou une tour de *shields* Arduino avec des contrôleurs *StepStick*, ça n'a pas seulement l'air bordélique. Les résultats vont aussi souvent de pair, des mauvais contacts aux interférences électromagnétiques (EMI). N'oublions pas non plus que nous travaillons ici en milieu quasi industriel, avec des vibrations mécaniques et de grandes différences de température.

En ce qui concerne les logiciels, nous avons subi divers plantages et autres bogues dans l'interface utilisateur. Lorsque la machine se met à aller à contresens alors qu'on est en mode manuel, il y a de quoi être surpris ! Les risques ne sont pas élevés – notre machine n'est pas une aléreuse/fraiseuse industrielle..., mais cela suffit pour ruiner un projet ou casser un foret.

### La commande numérique, le cœur de la machine

Suite à nos déboires avec les CNC libres, nous avons préféré une solution professionnelle et nous avons choisi une carte de *Eding CNC*, société spécialisée depuis plus de 25 ans dans la commande de moteurs et de machines [2].

La carte CNC310, introduite en 2018, répond à tous nos besoins ; elle est fiable et pas trop chère. Ci-après ses caractéristiques principales.

- Liaison avec un ordinateur via USB ou Ethernet (en option)
- Alimentation de 5 V externe ou via USB
- Sorties pour 3 axes avec interface standard STEP/DIR
- 4<sup>e</sup> axe en option
- Fréquence de pas maximale de 125 kHz
- Entrées pour interrupteurs de fin de course (en série), sonde et bouton d'arrêt d'urgence
- Sorties pour refroidissement, activation des moteurs et état du système
- Logiciel CNC 2.5D, avec possibilité

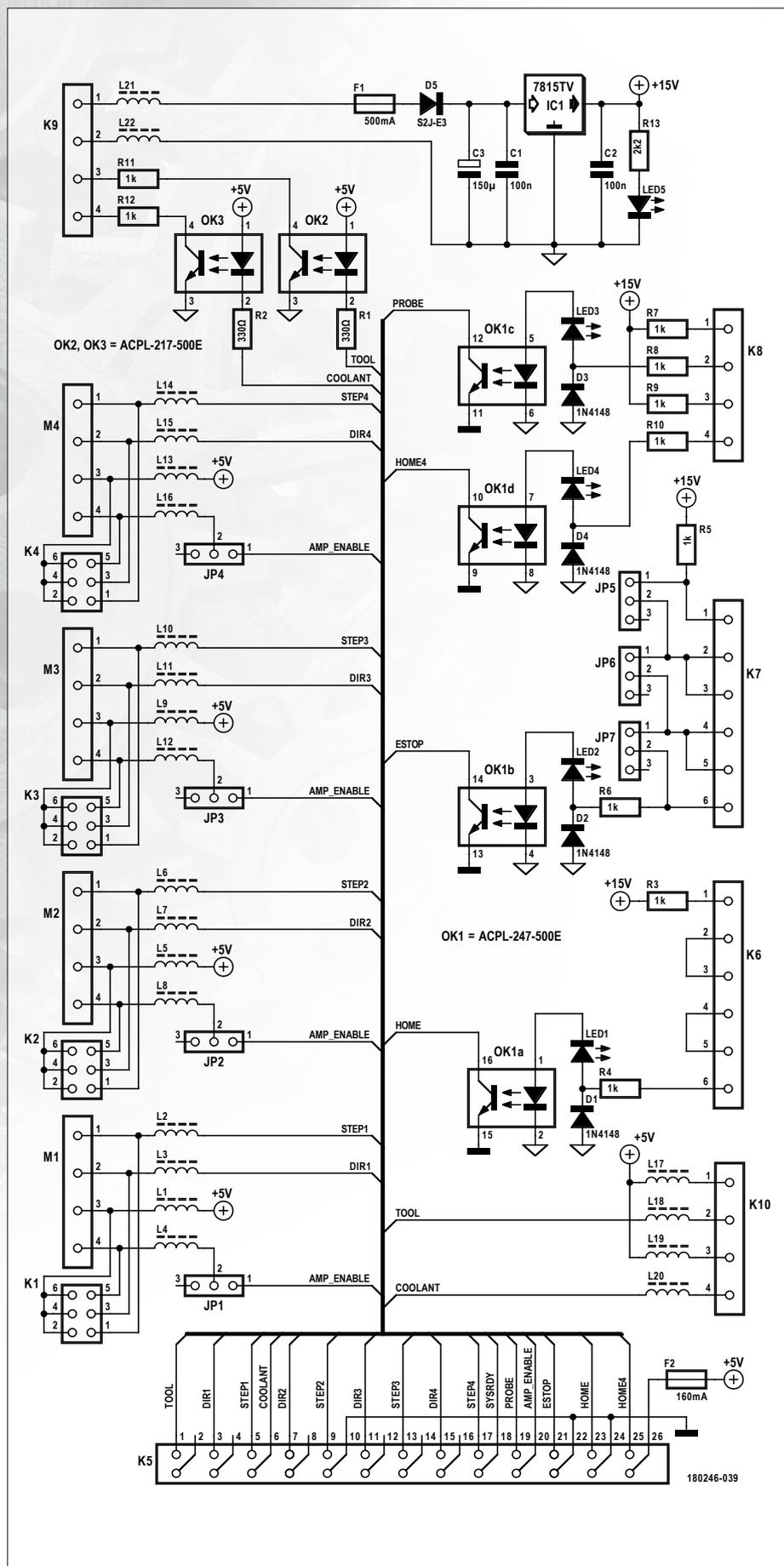


Figure 1. Schéma de la carte de liaison ; on y voit surtout les connecteurs.

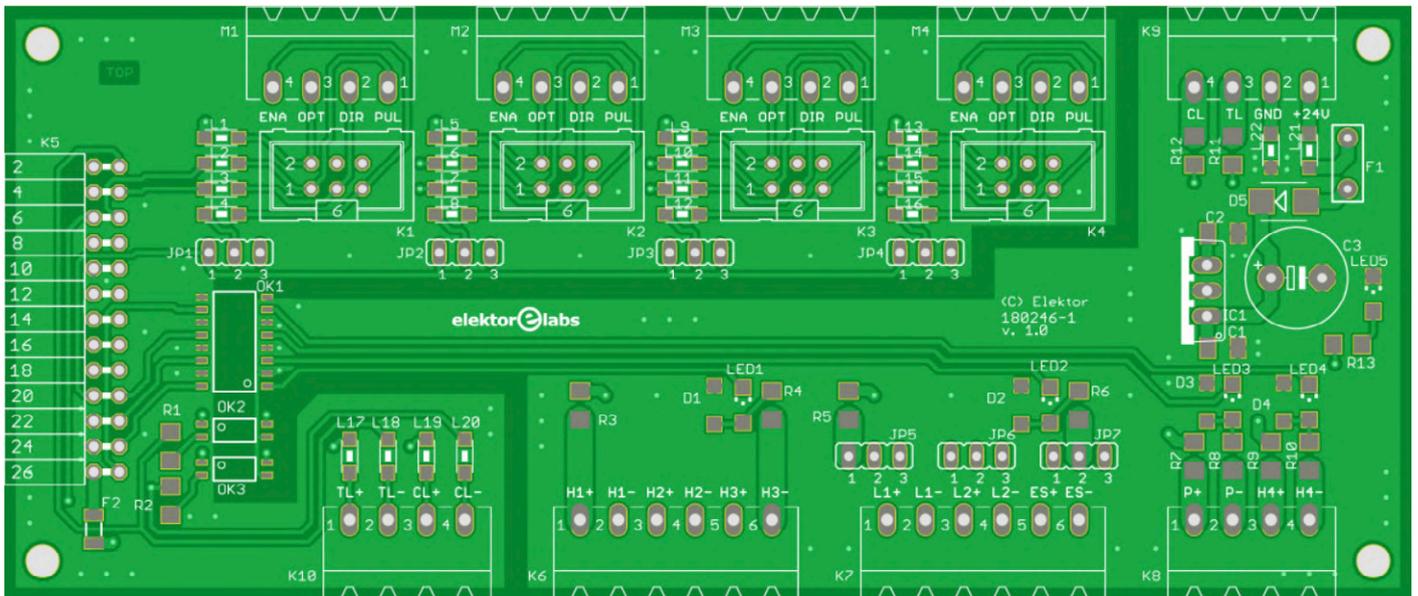


Figure 2. Circuit imprimé de la carte de liaison, côté composants.

- de passer en 3D (en option)
- Possibilité de raccorder un pendant

Deux inconvénients mineurs de la CNC310 : le nombre relativement limité d'E/S et le fait que les circuits de commande des moteurs pas-à-pas ne soient pas sur la carte. D'un autre côté, l'interface STEP/DIR permet de choisir le type de moteur pas-à-pas, petit ou gros, et sa commande adaptée, avec ou sans boucle de contrôle. On peut même y raccorder des servos industriels équipés d'un moteur à courant continu.

Si on a besoin de plus d'E/S, par ex. pour la commande du placement de CMS (*pick-and-place*), on peut choisir une autre carte de commande, plus chère, dans le catalogue d'Eding CNC.

Le raccordement de la carte CNC310 au reste de l'électronique se fait via un connecteur coudé de 26 broches. Nous avons dessiné une carte de liaison (*breakout board*) pour faciliter les liaisons aux autres composants.

### Carte CNC310

La carte CNC310 a en principe tout ce qu'il faut pour commander directement notre machine, et notre carte de liaison n'aurait pu contenir rien d'autre que des connecteurs. Nous avons cependant préféré ajouter un peu d'électronique – des optocoupleurs et des perles de ferrite

(**fig. 1**). Cela prévient les risques d'interférences électromagnétiques et rend le tout plus robuste.

Les optocoupleurs ont un double but : l'isolation galvanique entre la carte de commande et le reste de l'électronique, et la protection des lignes d'E/S grâce aux résistances de 1 kΩ.

Si une des broches d'E/S (connecteurs K6 à K9) venait à entrer accidentellement en contact avec l'alimentation de 24 V, il n'y aurait pas de casse. La puissance à dissiper par les résistances serait dans ce cas de 0,6 W ; nous avons donc prévu des pistes de cuivre assez larges pour faciliter l'évacuation de la chaleur (**fig. 2**).

L'isolation entre la carte CNC310 et le reste de l'électronique fait que les lignes d'E/S de cette carte ne sont pas trop proches des câbles menant aux moteurs. Il n'y a donc pratiquement aucun risque d'interférence pour l'interface USB.

En milieu industriel, Eding CNC recommande d'ailleurs l'utilisation de l'interface Ethernet, qui est de facto isolée galvaniquement. Il faut cependant une extension de la licence d'utilisation du logiciel, et peut-être une carte spéciale supplémentaire sur l'ordinateur hôte.

Le raccordement aux circuits de commande des moteurs pas-à-pas se fait

via les connecteurs M1 à M4 ou K1 à K4. Les cavaliers J1 à J4 permettent d'envoyer ou non le signal *Amp Enable* de la CNC310 vers les circuits de commande des moteurs. Ceci permet le cas échéant de mettre les moteurs en ou hors service par commande logicielle, avec un inconvénient lors de la remise en marche des moteurs : ceux-ci peuvent changer quelque peu de position. Sauf pour une application spécifique qui nécessiterait cette fonction, nous recommandons de laisser les moteurs sous tension en permanence lorsque la machine est en fonctionnement.

Le connecteur K6 permet le raccordement des interrupteurs de point de départ (*homing*) des trois premiers axes, en principe X, Y et Z. Nous rappelons que ces interrupteurs sont du type fermé au repos ; il est plus sûr d'avoir à rompre un contact que d'avoir à l'établir. Comme la CNC310 ne dispose que d'une seule entrée pour ces interrupteurs, les connexions sont en série sur le circuit imprimé. Une entrée non utilisée devra donc être court-circuitée.

Suivant le même principe, on peut raccorder à K7 un bouton d'arrêt d'urgence et/ou des interrupteurs de fin de course, ces derniers ayant le même effet qu'un arrêt d'urgence. Ces fonctions sont en option et devront être activées dans le programme d'Eding CNC. Les entrées

non utilisées peuvent être mises hors service avec les cavaliers JP5 à JP7. Une sonde, ou un système de réglage d'outil, et un interrupteur de départ pour un quatrième axe peuvent être raccordés à K8. Cet axe nécessitera une extension de la licence d'utilisation.

Les signaux de sortie de la CNC310 pour la perceuse/fraiseuse et la pompe de liquide de refroidissement sont disponibles tels quels sur K10. On peut y raccorder un relais électronique, pour par ex. commander par logiciel la mise en marche et l'arrêt d'un outil alimenté par le secteur 230 V. Ces signaux, isolés par des optocoupleurs, se retrouvent également sur K9 pour d'autres applications éventuelles, par ex. la commande d'un laser ou d'une machine de placement de CMS.

L'alimentation de 24 V de la carte de liaison est aussi raccordée à K9. Un régulateur linéaire à trois broches la transforme en une tension stable de 15 V.

### Commande des moteurs pas-à-pas

Nous avons choisi des moteurs NEMA 17, et nous pouvons donc utiliser n'importe quel circuit de commande qui supporte ces moteurs. Pour notre prototype (**fig. 3**), nous avons conçu notre propre circuit de commande sur la base du TMC2160 de *Trinamic* [3], relativement récent. Ce circuit de commande permet des valeurs efficaces de courant jusque 3 A pour l'alimentation des moteurs ; parfait pour des NEMA 17, 33 ou 34, ces derniers avec quelques limitations. La commande des moteurs est silencieuse ; ce n'est sans doute pas important pour une fraiseuse, mais c'est appréciable dans d'autres applications moins bruyantes. Nous décrivons ce circuit de commande en détail dans un prochain numéro d'Elektor.

La commande de moteurs pas-à-pas est moins évidente qu'il n'y paraît, et un circuit de commande de bonne qualité peut faire toute la différence. Nous déconseillons donc les circuits bon marché chinois et les *StepSticks*. Ces derniers ne sont conçus que pour des petits moteurs ; vu leurs dimensions réduites, leur capacité à dissiper la chaleur est limitée et ils peuvent devenir très chauds. Une isolation galvanique entre les signaux de commande et l'alimentation des moteurs

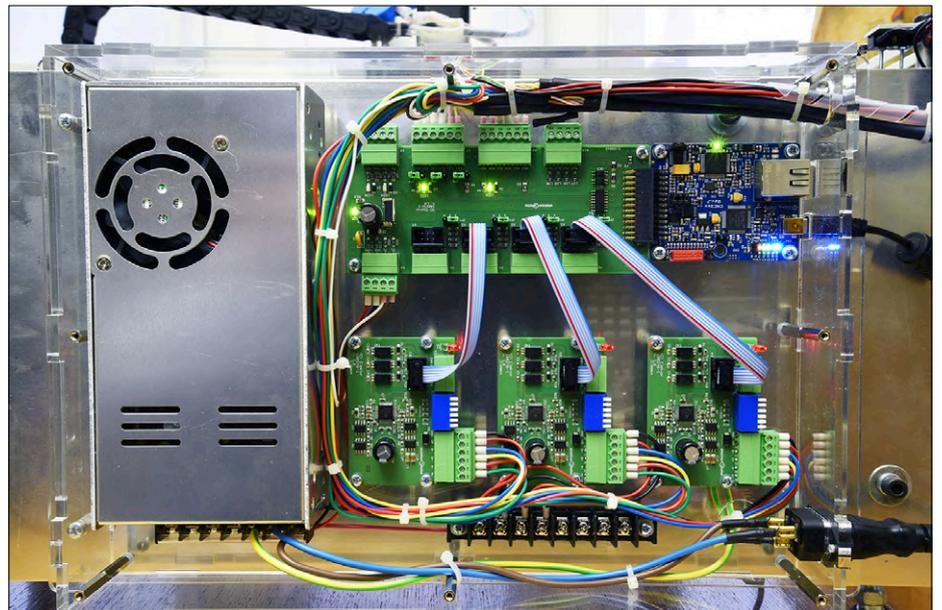


Figure 3. L'électronique se compose de la carte CNC310 d'Eding CNC, de notre carte de liaison, et de nos circuits de commande moteurs pas-à-pas à base de TMC2160.

est aussi importante. Si vous ne voulez pas attendre la description de notre circuit, nous recommandons le DM422C de *Leadshine*, avec lequel nous avons obtenu de bons résultats.

Si vous utilisez plusieurs circuits de commande, ils doivent être alimentés à partir d'un point central, et pas à la chaîne. Les circuits pourraient s'influencer mutuellement, ce qui n'est évidemment pas le but recherché !

### Logiciel de la CNC

Le logiciel pour la CNC310 (*Eding CNC Starter Software*) et les manuels associés peuvent être téléchargés du site du fabricant [2], il ne tourne que sous Windows 7, 8 ou 10. Veillez en tout

cas à bien suivre les instructions pour l'installation.

N'oubliez pas d'installer les pilotes USB, et de désactiver le contrôle du compte de l'utilisateur (UAC) et la mise à jour automatique du logiciel. Un arrêt intempestif lors d'un travail assez long parce qu'une mise à jour est disponible ne serait certes pas apprécié...

Réglez aussi les paramètres d'alimentation et de mise en veille, et donnez au logiciel d'Eding CNC des droits d'administrateur. C'est avec ces droits qu'il faut démarrer le programme.

Le fichier `cnc.ini` peut être téléchargé de la page de l'article [4] et doit être

### Liens

- [1] Elektor, 03-04/2019 : <http://www.elektormagazine.fr/180246-04>
- [2] Eding CNC : <http://www.edingcnc.com>
- [3] Trinamic : <http://www.trinamic.com>
- [4] Page de l'article : <http://www.elektormagazine.fr/180246-B-04>
- [5] DraftSight : <http://www.3ds.com/products-services/draftsight-cad-software/free-download>
- [6] EstlCAM : <http://www.estlcam.de>
- [7] Flatcam : <http://flatcam.org>
- [8] Canal YouTube d'Elektor : <http://www.elektor.tv>

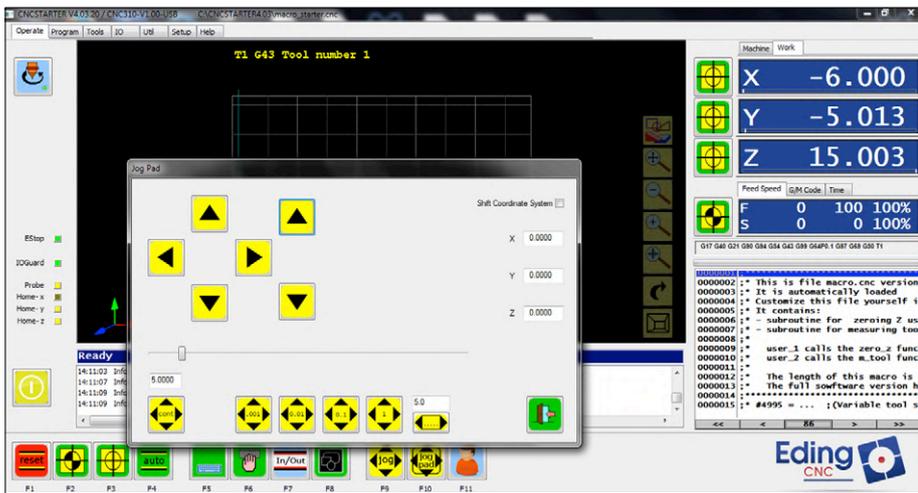


Figure 4. Choisissez un pas de 5 à 10 mm et déplacez les axes manuellement à l'aide du Jog Pad.

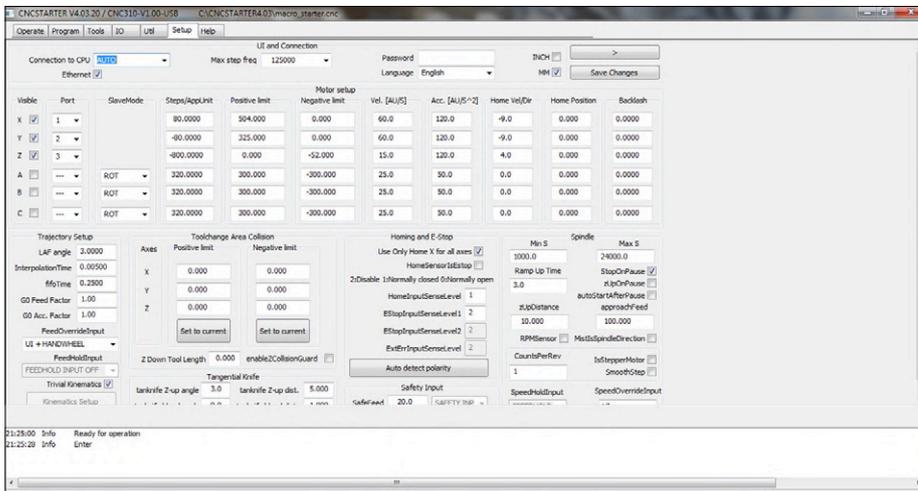


Figure 5. Réglage des paramètres principaux.

placé dans le même dossier que le logiciel d'Eding CNC avant la première utilisation. La plupart des réglages devraient ainsi être à peu près bons.

Placez les axes en position médiane, allumez la machine et lancez le programme. La touche F1 permet de relancer le programme et d'activer les circuits de commande des moteurs.

On ouvre ensuite Jog Pad (manette de commande virtuelle, **figure 4**) et on sélectionne un déplacement manuel des axes de 5 à 10 mm. On peut ensuite déplacer les axes en appuyant sur les boutons. Si un des axes va dans la mauvaise direction, il faut ouvrir le menu Setup et ajouter ou supprimer un signe

moins aux valeurs correspondantes de Steps/AppUnit.

Enregistrez les réglages, retournez à l'écran principal et appuyez sur la touche F1. Vous pouvez maintenant tester le branchement correct des interrupteurs de point de départ.

Si tout fonctionne bien, on peut passer au calibrage de la machine. Les valeurs initiales de Steps/AppUnit du menu Setup (**fig. 5**) peuvent être fixées à ±80 pour les axes X et Y, et ±800 pour l'axe Z (micro-pas de d'1/8<sup>e</sup> de pas). Ces valeurs représentent le nombre de pas d'un moteur nécessaires pour un déplacement de l'axe de 1 mm. Ouvrez Jog Pad et déplacez les axes sur une

distance fixe, par ex. 20 ou 40 mm. Mesurez le déplacement réel à l'aide d'un pied à coulisse et modifiez les valeurs de Steps/AppUnit pour qu'il y ait correspondance entre le déplacement logiciel et le réel. On peut aussi utiliser des instructions en G-code au lieu du Jog Pad. Par ex. G00 X100 déplace l'axe X de 10 cm vers la droite à partir de la position « 0 » ; si on tape ensuite G00 X200, l'axe X se déplacera à nouveau de 10 cm vers la droite à partir de la position précédente, ce que l'on pourra vérifier avec le pied à coulisse.

Lorsque tous les axes ont été calibrés, la machine est prête à l'emploi. Les paragraphes ci-après illustrent la mise en œuvre à l'aide de deux projets simples : une lampe de table et un circuit imprimé pour une baladeuse.

## G-code

Le G-code, normalisé par l'ISO sous la référence RS-274-D, est le langage de programmation développé pour la commande de machines-outils à l'aide d'ordinateurs (CNC). Diverses instructions permettent la commande de moteurs et le déplacement d'outils suivant un schéma défini et à vitesse réglable, le but étant de façonner un matériau par enlèvement de matière. Le G-code était initialement destiné aux machines-outils d'usinage, mais il sert désormais aussi à la commande d'autres processus : découpe au laser, mesure, impression graphique en mode trait, et bien entendu impression 3D.

C'est à la fin des années 50 qu'un langage de programmation de commandes numériques a été mis en œuvre au laboratoire de servomécanismes du MIT. Diverses organisations ont développé par la suite leur propre version du G-code, avec comme conséquence des problèmes de compatibilité. Beaucoup de constructeurs de machines-outils ont alors adopté dans les années 70 à 90 le jeu d'instructions du fabricant japonais FANUC. De nos jours, le problème ne se pose plus, car les logiciels de fabrication assistée par ordinateur (FAO) peuvent produire un code adapté à la machine à commander.

Notre machine est elle aussi commandée en G-code. On peut écrire le programme à la main, mais c'est tout de même beaucoup plus facile d'utiliser un logiciel de FAO...

## Logiciel de FAO

L'offre en matière de logiciels de FAO est variée, du code source ouvert gratuit au programme à quelques milliers d'euro. Il y a des logiciels intégrés de CAO/FAO permettant la conception de A à Z d'un objet, et d'autres de FAO qui transforme les données de CAO en G-code. On peut citer – dans le désordre – Fusion 360, SolidCAM, Vectric, Flatcam, CAMBAM, EstlCAM, DeskProto, SprutCAM, Sheet-CAM, Inventor (avec Inventor HSM), et Dolphin CAD/CAM.

C'est un peu déroutant, si l'on n'a que peu ou pas du tout d'expérience en la matière, et nous avons sélectionné *Draftsight* [5], un programme de CAO 2D, et *EstlCAM* [6]. Le premier peut être utilisé sans licence pour des applications non commerciales, et une version complète du second est à l'essai gratuitement pour une durée déterminée. *EstlCAM* vous coûtera ensuite 49 € si vous en êtes satisfait ; c'est un programme très intuitif et simple d'emploi, mais très complet et puissant.

## Une lampe de table pour un éclairage d'ambiance

Notre lampe de table est un simple bocal à bonbons avec un socle usiné en fibres de bois (MDF). La lumière varie du blanc chaud au blanc froid grâce à des LED illuminant le bocal par le dessous. La lampe est alimentée par un adaptateur secteur de 12 V. Nous avons rempli le bocal de copeaux de Plexiglas (fig. 6), mais on peut aussi utiliser des billes, des petites bouteilles, ou des petits cubes en plastique. Il faut que la lumière des LED soit diffractée et réfléchiée dans toutes les directions, et que la transition entre lumière chaude et froide soit bien perceptible. Trois plaques de MDF d'une épaisseur de 12 mm ont été usinées avec une fraise de 3 mm en métal renforcé *Proxxon* 28761, puis collées et vissées ensemble.

Nous commençons par dessiner les trois parties du socle avec *Draftsight* (fig. 7). C'est relativement facile et rapide, il ne s'agit que de quelques cercles. Le résultat est sauvegardé dans des fichiers au format DXF (*Drawing eXchange Format*). Vous pouvez utiliser tout autre logiciel qui sort des fichiers au format DXF.

Les trois fichiers obtenus sont ensuite traités avec *EstlCAM*, et nous prenons



Figure 6. Notre lampe de table.

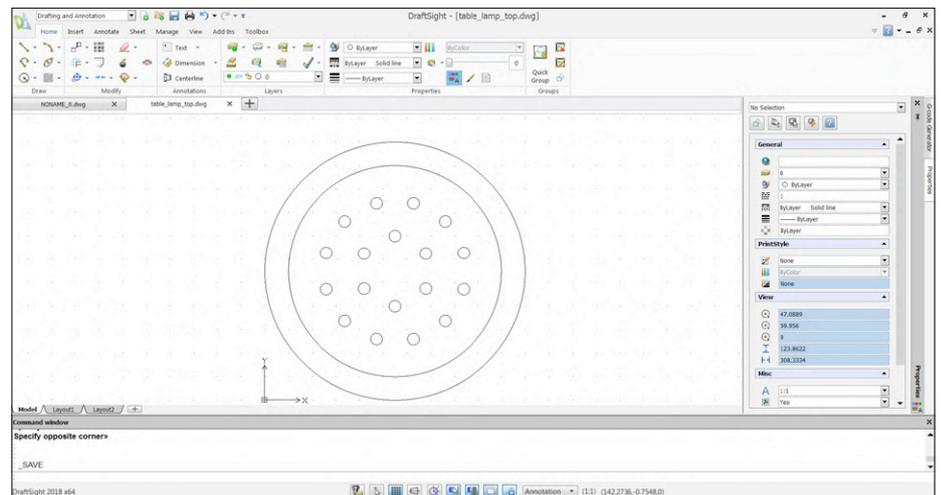


Figure 7. Les dessins sont produits avec *Draftsight* et sauvegardés au format DXF.

@ WWW.ELEKTOR.FR

→ Carte Eding CNC310

[www.elektor.fr/cnc310](http://www.elektor.fr/cnc310)

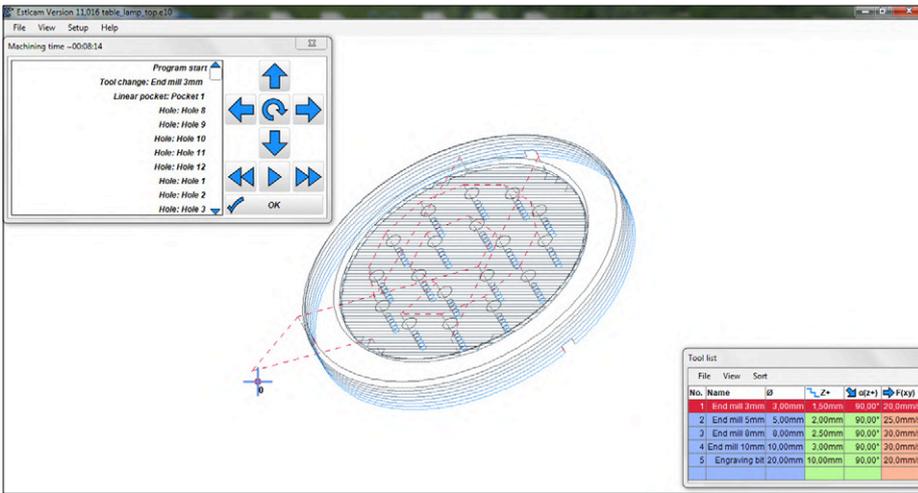


Figure 8. Les fichiers DXF sont importés dans *EstlCAM*, pour la production du G-code.

la plaque supérieure du socle comme exemple (**fig. 8**). On sélectionne d'abord l'outil d'usinage, dans ce cas une fraise de 3 mm (*End Mill 3 mm*), ou on en crée un nouveau si nécessaire. On règle ensuite divers paramètres : *Depth per pass - Step down* = 1,5 mm ; *Plunge angle* = 90° ; *Feedrate* = 1.200 mm/min (20 mm/s) ; *Plunge feedrate* = 600 mm/min (10 mm/s) ; *Stepover* = 90%. La vitesse de rotation (*RPM*) n'a ici pas d'importance, car nous fixons la vitesse de la *Proxxon* manuellement à environ 18.000 tours/min.

Il y a bien entendu des formules et des tableaux pour calculer les déplacements des axes, mais notre machine n'est pas des plus rigides et la puissance de la *Proxxon* est limitée. Une approche plus pragmatique est donc conseillée.

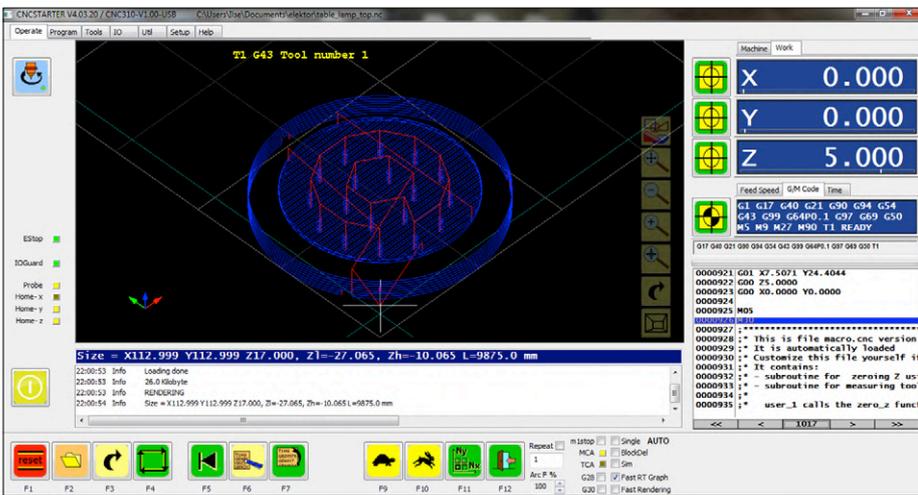


Figure 9. Le G-code peut être ouvert dans le programme d'*Eding CNC* pour l'exécution de l'usinage.

On appuie sur le bouton *Hole* (à gauche) et on sélectionne le cercle du milieu. On fixe *Toolpath depth* = 3 mm, *Start level* = 0 mm et *Machining order* = 10. On appuie sur *Pocket* et on sélectionne *Pocketing strategy* = *Linear*. Le résultat : le cercle médian sera fraisé sur une profondeur de 3 mm en deux étapes, par mouvements unilatéraux de la fraise.

Au tour des petits trous pour les LED de 5 mm : *Toolpath depth* = 9 mm, *Start level* = 3 mm, *Machining order* = 20.

On appuie maintenant sur le bouton *Part*, à gauche, et on clique sur le cercle extérieur. On fixe *Toolpath depth* = 12 mm, *Start level* = 0 mm et *Machining order* = 30. On place ensuite trois attaches

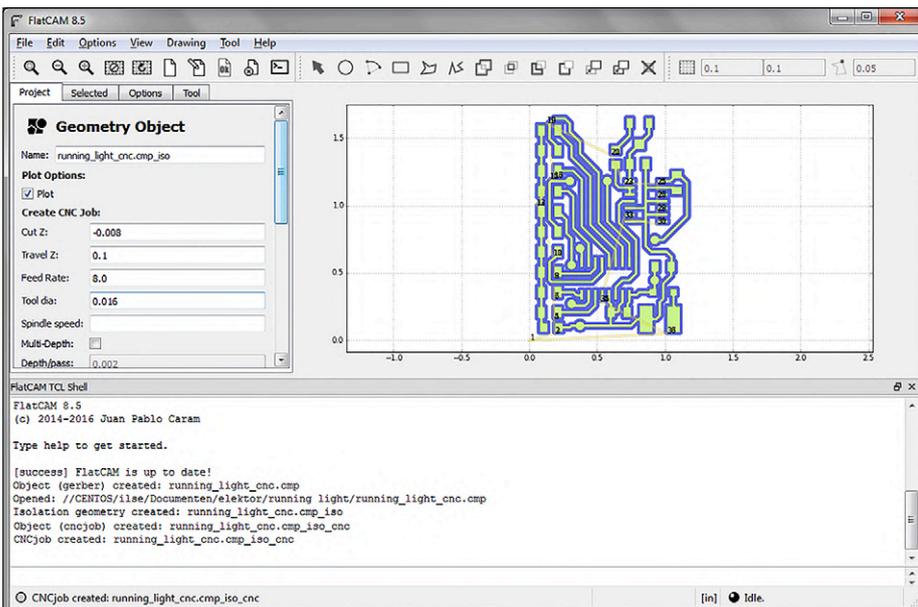
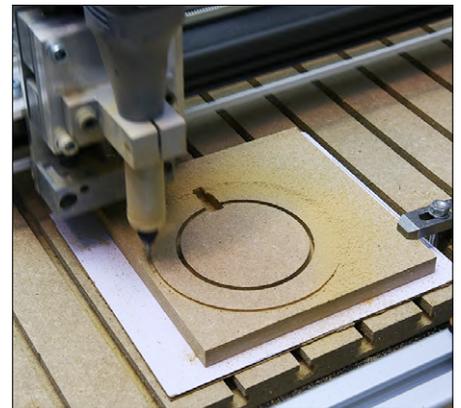


Figure 10. Avec *Flatcam*, on peut obtenir du G-code à partir de fichiers Gerber.





## LISTE DES COMPOSANTS

### Résistances :

R1, R2 = 330 Ω (CMS 1206)  
 R3-R12 = 1 kΩ, 660 mW (CMS 1206), par ex.  
 Panasonic ERJP08F1001V  
 R13 = 2,2 kΩ (CMS 1206)

### Condensateurs :

C1, C2 = 100 nF, 50 V, X7R  
 C3 = 150 μF, 50 V

### Semi-conducteurs :

D1-D4 = 1N4148 (CMS SOD123)  
 D5 = S2J-E3  
 LED1-LED5 = LED verte (CMS 1206)  
 OK1 = ACPL-247-500E  
 OK2, OK3 = ACPL-217-500E  
 IC1 = 7815 (TO-220)

### Divers :

L1-L22 = perles de ferrite, 2512067007Y3 (CMS 1206)  
 F1 = fusible réarmable PTC, 500 mA, 60R050XPR  
 F2 = fusible réarmable PTC, 160 mA (CMS 1206)  
 JP1-JP5 = connecteur mâle à 3 pôles avec cavalier, au pas de 2,54 mm  
 M1 à M4, K8 à K10 = connecteur Phoenix Contact MC1.5/4-G-3.81 + MC1.5/4-ST-3.81  
 K1 à K4 = connecteur mâle engagé à 2×3 contacts, au pas de 2,54 mm  
 K5 = connecteur femelle coudé à 2×13 contacts, au pas de 2,54 mm  
 K6, K7 = connecteur Phoenix Contact MC1.5/6-G-3.81 + MC1.5/6-ST-3.81

On se rend ainsi compte des mouvements de l'outil, sans aucun risque de crash.

Si vous comptez usiner des matériaux plus durs, comme des métaux, recherchez des vidéos sur le fraisage trochoïdal, ça en vaut vraiment la peine !

Enfin, n'oubliez jamais votre propre sécurité : portez toujours des lunettes de protection, et si nécessaire un masque.

### Circuit imprimé pour une baladeuse

Nous voulons maintenant graver un circuit imprimé pour une baladeuse. Le circuit est basé sur un compteur décimal CD4017 et un sextuple inverseur avec bascule de Schmitt CD40106 pour le signal d'horloge ; le montage est à base de CMS. Avec Flatcam [7], nous avons produit un fichier en G-code sur la base des fichiers Gerber (**fig. 10**). L'usinage a été réalisé avec une fraise à graver à pointe en métal renforcé (0,1 mm / 30°).

Tous les fichiers pour les deux projets sont sur la page de l'article [4], ce qui vous permettra de vous lancer directement. Il y a également des photos supplémentaires, pour éclaircir certains points, et des vidéos sur le canal YouTube d'Elektor [8]. ◀

(180246-B-04

version française : Jean-Louis Mehren)

(*tab*) sur le cercle avec le bouton *Holding tab* (*Holding tab length* = 1,5 mm et *Holding tab height* = 3 mm). Ceci permet d'interrompre le fraisage à l'emplacement des attaches, et l'objet reste en place lorsque la fraise le traverse de part en part. Sans cette précaution l'objet pourrait être « happé » par la fraise et endommagé.

Le projet doit être sauvegardé dans un fichier au format *.e10*. Une simulation de l'usinage par la machine-outil est possible avec *Preview*. Si c'est bien ce que l'on veut, on produit le G-code avec *File -> Save CNC program* et on le sauvegarde dans un fichier au format *.nc*. On peut encore sélectionner un post-processeur

avec *Setup -> CNC Program settings* ; l'option *Default* semble bien fonctionner, mais on peut aussi choisir *Eding CNC*.

On peut maintenant lancer le programme d'*Eding CNC*. On utilise le *Jog Pad* pour placer la fraise dans sa position initiale, que l'on enregistre comme position « 0 ». On positionne ensuite l'axe Z de telle manière que la fraise soit environ 5 mm au-dessus de la pièce de MDF et on importe le fichier contenant le G-code (**fig. 9**). On met la *Proxxon* en marche, on règle la vitesse à 18.000 tours/min et on démarre le travail, en veillant à aspirer régulièrement les copeaux. Si on n'est pas encore très sûr de soi, on peut fraiser « en l'air », sans l'objet à usiner.

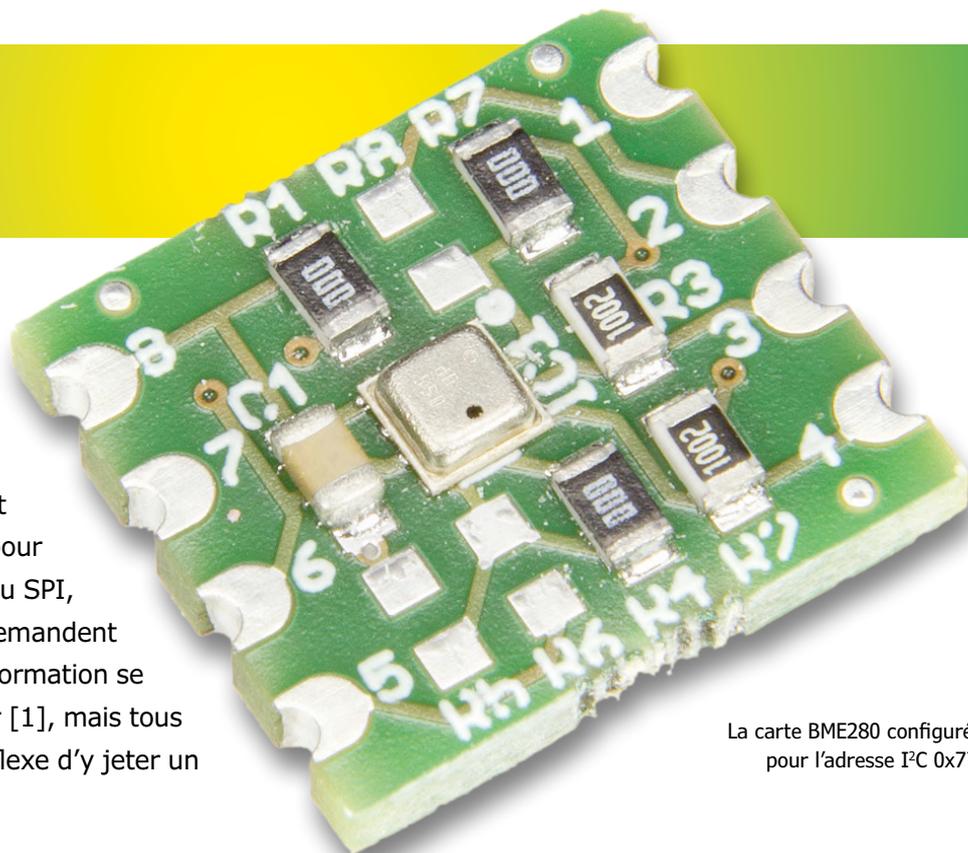


# configuration de la carte de liaison BME280 d'Elektor

Clemens Valens (labo d'Elektor)



Il y a quelques années, Elektor a conçu une petite carte de liaison ou *break-out board* (BoB) pour le capteur environnemental BME280 de Bosch Sensortec. Il est possible de configurer le capteur pour qu'il communique via un bus I<sup>2</sup>C ou SPI, mais nombre de nos lecteurs se demandent comment faire. Bien entendu, l'information se trouve sur le site du labo d'Elektor [1], mais tous les lecteurs n'ont pas encore le réflexe d'y jeter un coup d'œil. Voici la version papier.



La carte BME280 configurée pour l'adresse I<sup>2</sup>C 0x77.

La carte BME280 se configure avec des résistances de 0 Ω. Voici les trois options principales :

- I<sup>2</sup>C avec adresse 0x76
- I<sup>2</sup>C avec adresse 0x77
- SPI

Pour sélectionner le SPI, montez R8, mais pas R7 ; pour l'I<sup>2</sup>C, montez R7, mais pas R8.

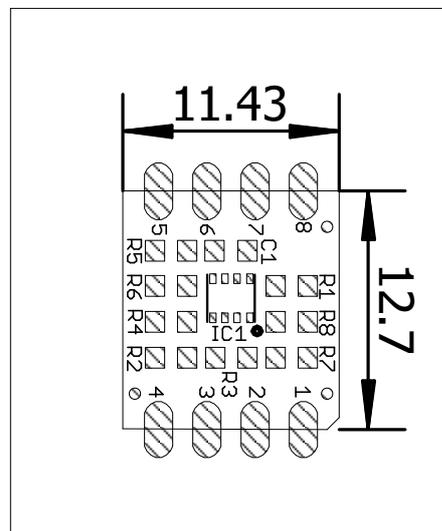
Reliez la broche SDO du BME280 à la masse pour choisir l'adresse esclave I<sup>2</sup>C 0x76 (R4 absente, R5 montée) ; reliez-la à VIO pour sélectionner l'adresse 0x77 (R4 montée, R5 absente).

Le BME280 peut être alimenté avec une tension comprise entre 1,71 V et 3,6 V. Son interface possède sa propre broche d'alimentation, VIO, qui accepte des tensions entre 1,2 V et 3,6 V. La résistance R1 relie VDD à VIO, sans doute le cas d'utilisation le plus courant. Retirez R1 lorsque vous souhaitez utiliser une tension différente de la tension d'alimentation pour l'interface.

Toutes les options possibles se trouvent dans le **tableau 1**.

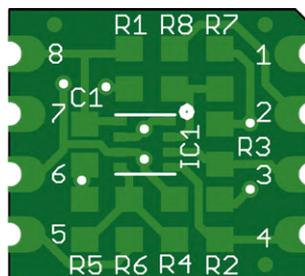
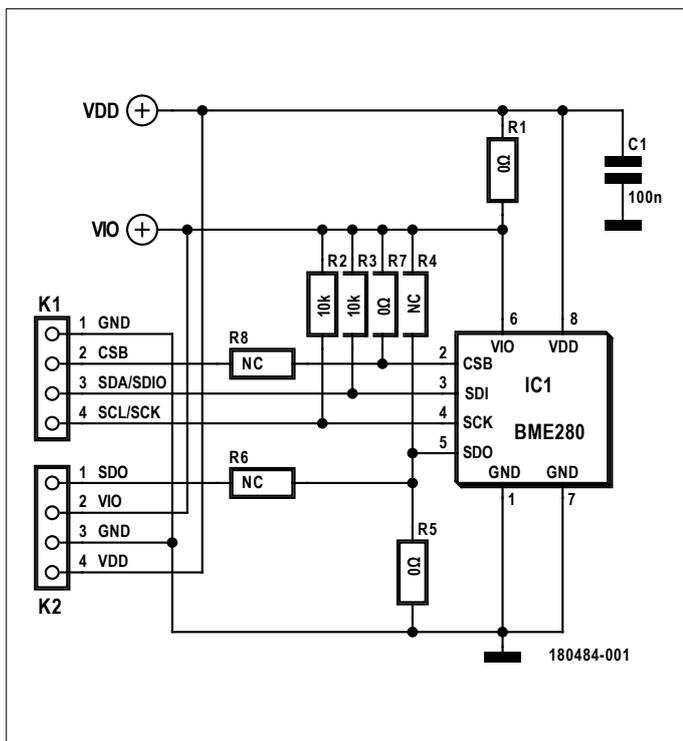
Elektor a également conçu un pilote compatible Arduino et mbed pour le BME280 ; il est disponible sur GitHub [2]. ◀

(180484-02 – version française : Kévin Petit)



I <sup>2</sup> C (0x76)	I <sup>2</sup> C (0x77)	SPI
R1 = 0Ω	R1 = 0Ω	R1 = 0Ω
R2 = 10kΩ	R2 = 10kΩ	R2 = NC
R3 = 10kΩ	R3 = 10kΩ	R3 = NC
R4 = NC	R4 = 0Ω	R4 = NC
R5 = 0Ω	R5 = NC	R5 = NC
R6 = NC	R6 = NC	R6 = 0Ω
R7 = 0Ω	R7 = 0Ω	R7 = NC
R8 = NC	R8 = NC	R8 = 0Ω

Tableau 1. Toutes les combinaisons de résistances utiles.



### Liens

- [1] Page du projet au labo d'Elektor [www.elektor-labs.com/1745](http://www.elektor-labs.com/1745)
- [2] Pilote BME280 pour Arduino & mbed : <https://github.com/ElektorLabs/bme280-driver>

TEXAS INSTRUMENTS | Mouser.com

**Mouser stocke la plus vaste sélection de produits TI**

**Plus de 46 000 produits TI**

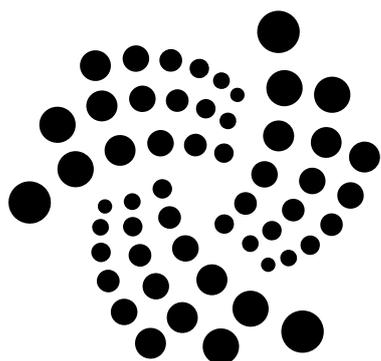
**Plus de 4 000 outils de développement TI**

Mouser Electronics – votre distributeur TI agréé, stockant de nombreux autres produits pour vos prochaines conceptions. [mouser.fr/ti](http://mouser.fr/ti)

**M MOUSER ELECTRONICS**

# cryptomonnaie IOTA (et FPGA pour RPi)

## 1<sup>ère</sup> partie : les machines paient les machines



I O T A

Julia Orlitzky et Thomas Pototschnig

Grâce au progrès technique, la taille des appareils de l'Internet des Objets (IdO) chute, leurs prix dégringolent et leur consommation énergétique s'effondre. Mais cela s'accompagne d'une marée montante de données qu'il faut pouvoir maîtriser. L'une des solutions pourrait consister en une offre de puissance de calcul sur le réseau capable de s'adapter avec souplesse à la demande, ce qui exigerait un mode de paiement que des nœuds IdO soient en mesure de gérer automatiquement. Ici interviennent de nouveaux types de cryptomonnaie comme IOTA. Dans cet article, nous vous en donnerons un aperçu général ; dans le prochain article, nous présenterons une carte d'extension pour le RPi capable de traiter des transactions IOTA avec célérité.

Les possibilités d'utilisation de l'IdO sont multiples. Le réfrigérateur qui gère lui-même son contenu, la voiture qui décide seule de son itinéraire à l'aide d'informations en temps réel sur le trafic, ou la machine à laver qui envoie un SMS en fin de programme de lavage, n'en sont que les exemples les plus connus. Il n'y a pas de limite à l'imagination.

Beaucoup d'appareils de l'IdO s'occupent exclusivement de l'acquisition de données. On a ainsi des capteurs qui collectent des données météorologiques pour analyser le climat au niveau mondial, des sondes qui captent des niveaux de radiations pour établir une carte de la radioactivité sur le territoire japonais, des compteurs électriques intégrés à une maison intelligente qui évaluent la consommation électrique et les habitudes de l'utilisateur. Grâce à ces données, les compagnies d'électricité peuvent mieux prédire les pointes de courant et optimiser leur réseau.

Mais l'IdO va encore plus loin. Il rend possible une économie entre machines, où des machines peuvent rémunérer d'autres machines pour services rendus. La société d'études de marché

Gartner a prédit en 2017 [1] qu'en 2020 environ vingt milliards d'objets seraient connectés à l'internet. Mais avec le nombre d'objets croît également la masse de données produites, rendant prévisible une saturation des technologies actuelles.

La question qu'on doit se poser est : que faire de toutes ces données ? Et quand on aura trouvé un espace où les stocker, comment va-t-on y accéder et comment peut-on les utiliser ? C'est précisément ici qu'intervient IOTA. IOTA s'efforce de résoudre toutes les difficultés techniques que ne cesse d'apporter la popularité toujours croissante de l'IdO.

### Quel est le problème qu'IOTA cherche à résoudre ?

Le développement des cryptomonnaies telles que Bitcoin ou Ethereum s'est accompagné de « livres de comptes » virtuels décentralisés, infalsifiables et sûrs, que tout le monde peut utiliser anonymement et sans contrôle d'accès, au moyen d'une clé cryptographique créée par l'utilisateur. Cette vision était séduisante, car il devenait soudain possible de transférer de

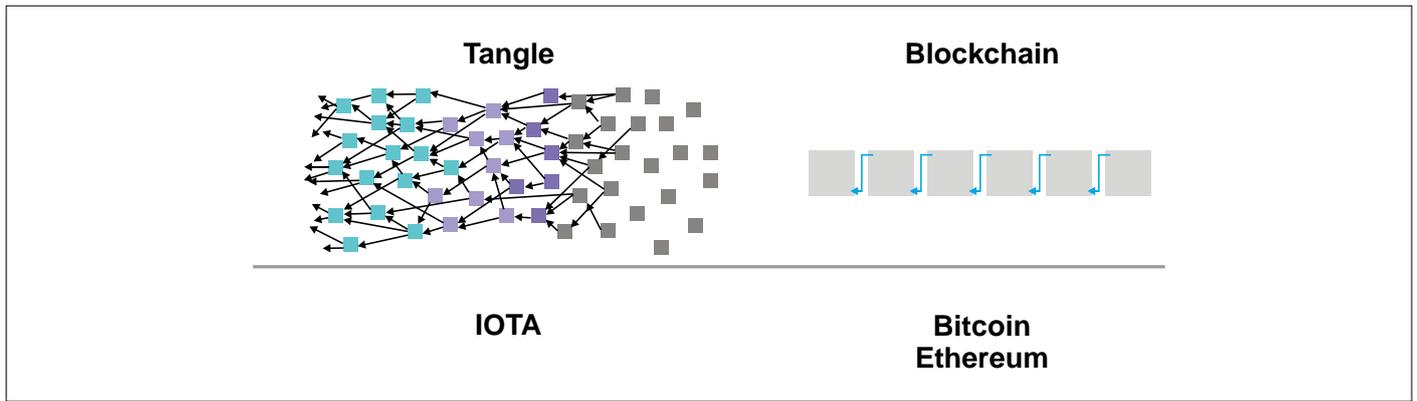


Figure 1. Le graphe orienté acyclique comparé à une Blockchain [8].

l'argent virtuel entre pairs, sans intermédiaires ni recours à un organisme central de confiance.

Malheureusement, des problèmes de conception imprévus sont rapidement apparus. La conséquence est que le débit de transactions des *blockchains* de la première génération (Bitcoin) et de la deuxième génération (Ethereum avec des *smart contracts*) est très limité, ce qui rend ces monnaies incompatibles avec le futur Internet des Objets. Il y a aussi un problème avec le modèle de calcul des frais de traitement qui soumet à des enchères la place d'une transaction dans la queue des chaînes de blocs chiffrés, ce qui, en période de pointe, conduit à des coûts de transaction absurdement élevés ou à des délais de validation prohibitifs.

IOTA peut être qualifiée de cryptomonnaie de troisième génération. Elle s'écarte totalement de la technologie blockchain linéaire et s'appuie sur une technique de livre de comptes distribué (*Distributed Ledger Technology, DLT*), dans laquelle toutes les transactions sont enregistrées dans un graphe orienté acyclique (*directed acyclic graph*) (fig. 1). Ce réseau, avec ses données, s'appelle un tangle (graphe enchevêtré ou entrelacs). L'avantage est évident : alors que les blockchains ont de graves problèmes d'adaptation d'échelle, les tangles d'IOTA n'ont pas ce genre de goulot d'étranglement qui limite le débit maximal de transactions. Le tangle fonctionne même d'autant mieux qu'il y a davantage de transactions à traiter (fig. 2).

Outre le transfert de cryptomonnaie, IOTA offre la possibilité d'effectuer des transactions non monétaires, c'est-à-dire d'utiliser le tangle comme infrastructure et dépôt de paquets de données de toutes sortes, ce qui, par exemple, le rend idéal pour les données de mesure, et sans prélever des frais de transactions. De plus, il existe des extensions officielles du protocole à des niveaux plus élevés comme MAM (*Masked Authenticated Messaging*) qui a la capacité de gérer des flux continus de données chiffrées.

quBIC, un équivalent des *smart contracts* d'Ethereum est en cours de développement, mais n'est pas encore employé. À la différence des *smart contracts*, IOTA va utiliser un langage de programmation fonctionnel qui rendra possible une preuve mathématique de fonctionnement correct.

### Les transactions dans le tangle

Avec la cryptomonnaie bien connue Bitcoin, toutes les transactions doivent être incorporées à des blocs par des agents de traitement, les mineurs. Pour ce travail, les mineurs sont

rémunérés avec de nouveaux Bitcoins, ce qui donne une motivation pécuniaire à travailler pour le réseau. Au cours du minage, il faut résoudre un problème mathématique dit « preuve de travail » (*Proof Of Work, PoW*), ce qui prend dix minutes environ. Si la production de blocs augmente parce qu'on dispose de plus de puissance de calcul, la difficulté du problème est accrue de manière à ne pas dépasser la vitesse de six blocs à l'heure – cela automatise le réseau du Bitcoin.

Mais cette bonne idée apparente fait du minage le plus grand point faible du Bitcoin. Il représente un goulot d'étranglement qui limite fortement le débit des transactions, car la capacité maximale de données d'un bloc est fixe. Actuellement, environ 300 000 transactions sont validées chaque jour, soit à peu près 3,4 transactions par seconde. Ethereum s'en tire un peu mieux : 20 transactions par seconde.

Il y a encore plus de problématique : la priorité avec laquelle une transaction est intégrée à un bloc est déterminée par la rémunération du mineur, qui a intérêt à privilégier les transactions « les plus importantes ». Cela est apparu clairement lors du dernier emballage du Bitcoin, fin 2017, lorsqu'il a fallu payer jusqu'à 50 \$ de frais pour qu'une transaction devienne définitive dans la Blockchain (fig. 3).

Il en va autrement avec IOTA : pour exécuter une transaction dans le tangle, deux transactions précédentes doivent être validées, opération dont la rémunération consiste en la validation de sa propre transaction par une transaction suivante. Avec ce système de validation « Payer au suivant » (*Pay it Forward*), il n'y a pas besoin d'incitations financières.

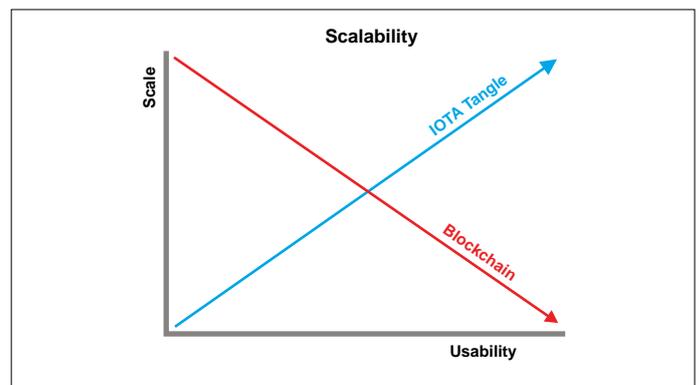


Figure 2. En ce qui concerne l'adaptation d'échelle, le tangle d'IOTA est largement supérieur à la Blockchain [9].

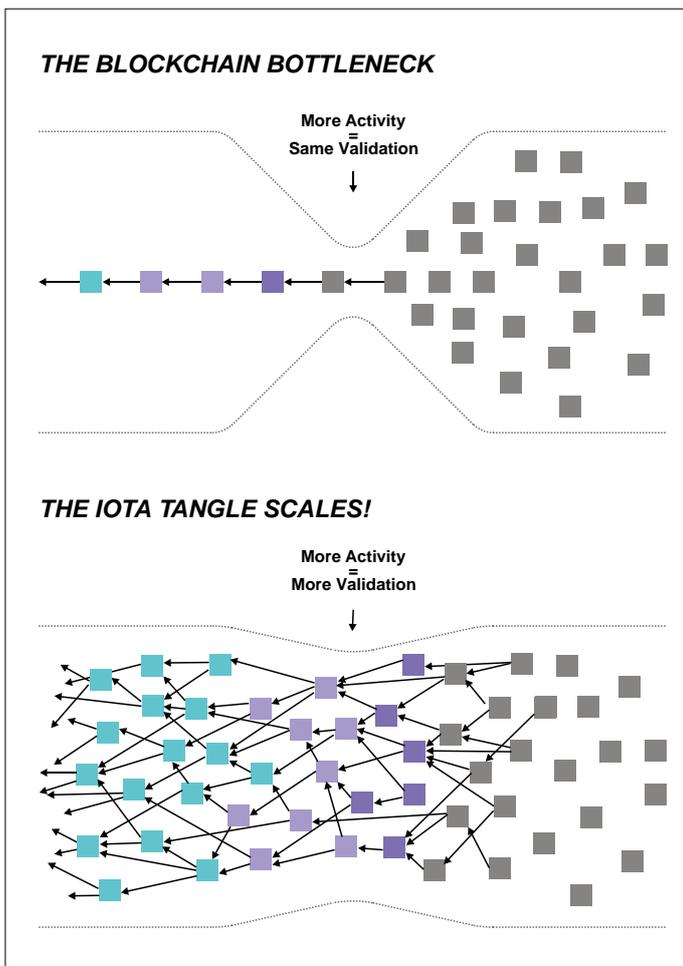


Figure 3. À l'inverse de la Blockchain, l'augmentation d'activité du tangle IOTA entraîne l'accroissement du nombre de transactions valides [10] !

Sans la nécessité de récompenses financières, IOTA ne dépend d'aucun calcul de valeur des transactions. Il est possible d'enregistrer des informations en toute sécurité dans des transactions du tangle ou même de répartir de grosses quantités d'informations sur plusieurs transactions groupées ou liées. Cette structure permet aussi une grande capacité d'adaptation d'échelle. Plus grande est l'activité dans le tangle, plus rapide est la validation des transactions.

Comme il n'y a pas de frais de transaction dans IOTA, on peut l'utiliser sans problème pour les micropaiements. Pour le courant fourni au réseau, l'exploitant d'une centrale solaire peut donc être payé en temps réel par la compagnie distributrice, sans avoir à régler des frais de transaction.

Pour opérer une transaction, il n'y a en principe que deux règles :

- **Une transaction doit valider deux autres transactions.**

Voilà qui a l'air simple... à première vue ! Les extrémités ouvertes du tangle s'appellent des *tips* (pointes). Un algorithme (recommandé par la fondation IOTA) choisit deux *tips* auxquels la nouvelle transaction est reliée.

Le terme « recommandé » mérite une attention spéciale, car l'usage du tangle d'IOTA n'est censé être soumis à aucune contrainte, ce qui signifie qu'en principe chacun peut en faire ce qu'il veut, y compris choisir une paire de *tips* au hasard,

même si la pratique a montré que cela donne des résultats médiocres (mauvais débit de validations). On considère donc que le participant à un tangle accepte de son plein gré d'utiliser un algorithme recommandé parce qu'il donne de bons résultats. C'est dans l'intérêt non seulement de l'utilisateur, mais aussi du réseau dans son ensemble.

- **Une transaction doit être directement ou indirectement référencée par un jalon.**

Un jalon (*milestone*) est une transaction signée d'un organisme central (*coordinator*) qui spécifie la direction de croissance du tangle. Ce coordinateur est à la fois une malédiction et une bénédiction ainsi que la principale critique faite à IOTA. Il faut comprendre quel est le rôle effectif de ce coordinateur.

Sa tâche essentielle est de sécuriser le tangle. La sécurité du tangle augmente avec le nombre de transactions traitées, mais un débit minimal de transactions est nécessaire pour que le tangle puisse se passer du coordinateur. Au début du Bitcoin, il y a eu des « points de contrôle » (*check points*) similaires pour garantir un état sécurisé. De telles précautions sont nécessaires, car une attaque majoritaire (à 51%) ne peut être exclue, tant que le débit du réseau reste faible. Cela a permis dans le passé l'attaque « Double Spend », où des Bitcoins furent dépensés deux fois.

Quoique sa fonction ait un sens, le coordinateur représente aux yeux des partisans des cryptomonnaies non seulement un risque de point unique de défaillance (*Single Point Of Failure, SPOF*), mais aussi un lieu de concentration d'un pouvoir qui pourrait théoriquement être utilisé par la Fondation pour, par exemple, privilégier certaines transactions et en ignorer d'autres.

Un autre inconvénient est que cette instance centrale dégrade notablement les performances de l'ensemble du réseau, car elle agit comme un goulot d'étranglement. C'est à comparer avec un ordinateur multicœurs dont les cœurs seraient tous en attente du même fil.

La Fondation IOTA est consciente de ce problème et a constitué un groupe de projet chargé de travailler à faire disparaître le coordinateur (*Coordicide*) [2].

### La Fondation IOTA

La Fondation IOTA a été créée le 17 novembre 2017. Elle est financée par des dons (environ 5 % de la totalité des jetons IOTA). C'est une fondation de droit civil et d'intérêt public dont le siège est à Berlin (Allemagne). Mais elle est également présente dans d'autres villes comme Tel Aviv, Oslo ou Taipei. Au niveau mondial, la Fondation emploie actuellement (novembre 2018) 80 personnes environ. Il s'y ajoute plus de 70 000 membres actifs de la communauté et plus de 120 000 personnes attentives aux publications de la Fondation.

Dans le cadre d'une promotion active de l'écosystème autour d'IOTA, la Fondation poursuit ses propres recherches afin de développer une base scientifique solide pour la technologie tangle. De plus, elle s'appuie sur ses recherches pour développer des composants de base tels que les « IOTA-Full Nodes » (IRI) et des bibliothèques en divers langages de programmation destinées à simplifier l'intégration de la technologie IOTA dans les projets des partenaires qu'elle assiste activement par des conseils et des interventions. Un autre aspect important est la standardisation de la technologie IOTA, indispensable pour son adoption par le milieu industriel.

## Où trouver davantage d'informations ?

- Site web IOTA : [www.iota.org/](http://www.iota.org/)
- Partenaires : <http://iotaarchive.com/>
- Blog officiel : <https://blog.iota.org/>
- Écosystème : <https://ecosystem.iota.org/>
- Reddit : [www.reddit.com/r/Iota/](http://www.reddit.com/r/Iota/)
- Discord : <https://discordapp.com/invite/fNGZXvh>
- Twitter : <https://twitter.com/iotatoken>

### Marché des données IOTA

Le marché des données IOTA est en développement depuis 2015. Le but est de proposer un marché décentralisé sur lequel peuvent se vendre ou s'acheter des données de mesure. En l'occurrence IOTA se sert du fait qu'il n'y a pas de frais liés à l'utilisation du tangle, ce qui autorise le micro-paiement. Quoique ce marché soit encore dans un état bêta et fermé, il est déjà possible d'acheter des valeurs de mesure. Il y a par exemple à Kaiserslautern (Allemagne) un capteur qui accumule des données sur la qualité de la fourniture de courant électrique et les met à disposition sous la forme de flux de données pour un prix très bas de 1366 i (soit 0,000273 €) (NDT : 'i' est le symbole de l'unité de compte iota. Sur les marchés financiers, on utilise le mégaiota, Mi). Cette année, ce marché devrait ouvrir ses portes à tous les utilisateurs et lancer le commerce des données de mesure.

### Écosystème pour projets

La Fondation IOTA promeut également un écosystème pour projets et code. Sur la plateforme web développée par ses soins, les développeurs peuvent publier leurs projets et recevoir des dons en iotas. Aujourd'hui, il y a sur cette plateforme plus de 2500 utilisateurs enregistrés et près de 200 projets publiés. Par son fonds *Ecosystem Development Fund*, la Fondation soutient elle-même directement des projets sélectionnés. Le pot est basé sur des dons volontaires de l'année 2015 et contient à l'heure actuelle (décembre 2018) six millions de dollars environ. Les conditions essentielles pour un soutien sont que le projet soit à logiciel ouvert et que la communauté IOTA puisse profiter des résultats. Ce fonds peut aussi servir à soutenir des événements.

### Le processeur ternaire JINN

IOTA est ternaire : au lieu de bits, il utilise des trits, au lieu d'octets, des trytes. Au lieu de se contenter de 0 et de 1, on utilise le système équilibré à trois chiffres -1, 0 et +1. Cela semble à première vue inefficace, IOTA ne s'exécutant pour l'instant que sur des machines travaillant en binaire, ce qui impose des conversions répétées des données de binaire en ternaire et inversement, comportant de nombreuses multiplications et divisions, ce qui pose problème avec les petits microcontrôleurs.

Le système ternaire équilibré est toutefois très efficace en soi - plus efficace que le binaire. Les nombres positifs et négatifs y sont unifiés en un seul système de nombres, sans nécessiter de bit de signe. Au lieu de passer par le complément à deux (inversion de tous les bits et addition de 1), le négatif d'un nombre ternaire s'obtient en changeant le signe de chaque chiffre. Deux nombres ternaires peuvent être soustraits en prenant le négatif de l'un et en l'additionnant avec l'autre. Les opérations logiques et les fonctions complexes comme un



## Un événement oublié ?

Vous organisez une conférence, un salon... ou bien vous participez à un séminaire ou tout autre événement qui aurait sa place ici, partagez cette information avec tous les lecteurs.

Envoyez-nous tous les détails à [redaction@elektor.fr](mailto:redaction@elektor.fr).

## mai 2019

### ◇ Iseramat - salon des passionnés de radiocommunication et d'électronique

04/05 - Tullins-Fures  
[iseramat.org/](http://iseramat.org/)

### ◇ 22<sup>e</sup> édition du Forum 5i

15/05 - Grenoble  
[www.forum5i.fr/](http://www.forum5i.fr/)

### ◇ Viva Technology

16 au 18/05 - Paris  
[vivatechnology.com/](http://vivatechnology.com/)

### ◇ Tournoi national de robotique

17 au 19/05 - Nîmes  
[www.robot-sumo.fr/](http://www.robot-sumo.fr/)

### ◇ Ubuntu Party - découverte du logiciel libre et de ses enjeux

18 au 19/05 - Paris  
[ubuntu-paris.org/](http://ubuntu-paris.org/)

### ◇ EVS32 (a world of E MOTION)

Symposium mondial sur le véhicule électrique et hybride  
19/05 au 22/05 - Lyon  
[evs32-france.com/](http://evs32-france.com/)

### ◇ Go numérique !

20/05 - Lyon  
[go-numerique.fr/](http://go-numerique.fr/)

### ◇ Paris Healthcare Week

21 au 23/05 - Paris  
[www.parishealthcareweek.com/](http://www.parishealthcareweek.com/)

### ◇ MiNaPAD Forum 2019

7th Micro/Nano-electronics Packaging and assembly, design and manufacturing forum  
22 au 23/05 - Grenoble  
[france.imapseurope.org/](http://france.imapseurope.org/)

### ◇ Semaine européenne du développement durable

29/05 au 04/06 - partout en France

### ◇ Coupe de France de robotique

30/05 au 01/06 - Laroche-sur-Yon  
[www.coupederobotique.fr](http://www.coupederobotique.fr)



additionneur complet sont aussi faciles à réaliser qu'en binaire. Le système ternaire est aussi très séduisant du point de vue du matériel. Pour stocker ou transmettre un nombre donné en ternaire, il suffit d'un nombre de cellules mémoire ou de lignes bien inférieur au binaire. Par exemple, un nombre binaire de 64 bits ne nécessite en ternaire que 41 cellules mémoire. La conception de la puce s'en trouve simplifiée, la surface de galette de silicium par puce est réduite, le nombre de portes logiques nécessaires diminue, en bref, la conception devient dans l'ensemble plus efficace, plus petite et moins consommatrice d'énergie électrique.

L'idée de l'ordinateur ternaire n'est pas nouvelle. Il en existait déjà dans les années 1960, le Setun soviétique, par exemple. Au total, cinquante de ces ordinateurs ont été construits en 1995 à des fins éducatives et scientifiques (entre autres pour la recherche nucléaire) [3]. Mais jusqu'ici, ils n'ont pas réussi à s'imposer sur le marché.

Cependant, le principe de l'ordinateur ternaire n'a jamais été totalement oublié. En 2014, il est devenu public que le développement d'un processeur ternaire appelé JINN était secrètement en cours depuis des mois [4]. Le financement de ce projet extrêmement ambitieux aurait été assuré grâce à un financement participatif. Au total, 100 000 parts JINN ont été mises aux enchères sur la plateforme de commerce NXT [5]. La société Jinn Labs a été créée par les cofondateurs de l'IOTA, Sergey Ivancheglo et David Sørnstebo. Le premier a également fondé la plateforme de cryptomonnaie NXT.

En 2016, la chaîne Youtube de Hackaday a publié une vidéo intitulée « Construction d'un ordinateur en base 3 » dans laquelle Jessie Tank explique le fonctionnement de la logique ternaire [6]. Cette vidéo présente la première puce expérimentale appelée JINN et mentionne pour la première fois IOTA, censée mettre à disposition l'infrastructure nécessaire au processeur ternaire. Sur la puce JINN elle-même, on dispose de peu d'informations, le développement se déroulant à huis clos. Le processeur doit avoir une architecture située entre processeur central et processeur graphique, mais n'est pas optimisé en performance, car il doit pouvoir évoluer « horizontalement ». Alors que l'évolutivité « verticale » résulte d'une augmentation de la vitesse des cœurs du processeur, l'évolutivité « horizontale » provient d'une multiplication des cœurs. JINN est une petite puce économe en énergie destinée aux appareils de l'IdO, et peut être utilisée pour le traitement massivement distribué dans le tangle.

Déjà en 2014, les jeux en ligne MMOG (*Massively Multiplayer Online Games*) étaient considérés comme une application idéale de JINN. Il n'est donc pas surprenant qu'ait été annoncée pour cette année une plateforme de jeux en réalité virtuelle appelée Paracosm [7].

Une autre application possible de JINN serait l'intelligence artificielle, car la logique ternaire est bien mieux adaptée aux réseaux de neurones que la logique binaire. Dans ce cas, l'infrastructure d'IOTA et la puce JINN pourraient être utilisées pour créer un gigantesque réseau de neurones embrassant la Terre entière. ◀

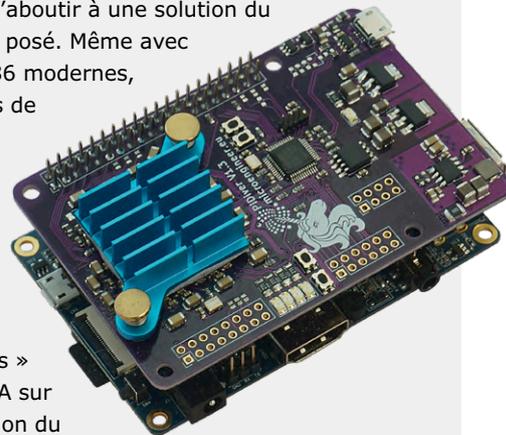
(180361-04 – version française : Helmut Müller)

### Dans le prochain numéro...

Une preuve de travail exige une grande quantité de calculs, ce qui ralentit considérablement l'envoi de transactions par les petits microcontrôleurs comme les ARM. L'une des principales causes est qu'il n'est pas rare de devoir effectuer plusieurs millions de hachages (algorithme CURL-P81) avant d'aboutir à une solution du problème de calcul posé. Même avec des processeurs x86 modernes, on n'y parvient pas de manière efficace.

Dans le prochain article, nous présenterons un portage à code ouvert de l'algorithme « Pêcheur de perles » (*Pearl Diver*) d'IOTA sur une carte d'extension du RPi, qui accélère notablement

le calcul de la preuve de travail. Ce « PiDiver » effectue jusqu'à 15,8 MH/s, soit un gain d'un facteur 300 sur le temps de calcul de la preuve de travail. Non seulement la puce FPGA est plus rapide qu'un processeur haut de gamme i5 à quatre cœurs avec une bibliothèque multifil optimisée SSE, mais elle se contente aussi d'une fraction de sa consommation électrique.



### Liens

- [1] Service de presse de Gartner (connexion exigée) : [www.gartner.com/en/newsroom/press-releases/2017-02-07-gartner-says-8-billion-connected-things-will-be-in-use-in-2017-up-31-percent-from-2016](http://www.gartner.com/en/newsroom/press-releases/2017-02-07-gartner-says-8-billion-connected-things-will-be-in-use-in-2017-up-31-percent-from-2016)
- [2] Blogue 'Coordinator: The Path to Coordicide' : <https://blog.iota.org/coordinator-part-1-the-path-to-coordicide-ee4148a8db08>
- [3] Ordinateur ternaire Setun : <https://en.wikipedia.org/wiki/Setun>
- [4] JINN sur le forum NTX : [https://nxtforum.org/news-and-announcements/\(ann\)-jinn/](https://nxtforum.org/news-and-announcements/(ann)-jinn/)
- [5] Jinnlabs : <https://web.archive.org/web/20141005232852/http://www.jinnlabs.com/>
- [6] Construire un ordinateur en base 3 : [www.youtube.com/watch?v=EbJMtJq20NY](http://www.youtube.com/watch?v=EbJMtJq20NY)
- [7] Présentation Paracosm : [https://paracosm.company/Paracosm\\_Showcase\\_ENG.pdf](https://paracosm.company/Paracosm_Showcase_ENG.pdf)
- [8] Source de la figure 1 : <https://twitter.com/iotatokenews/status/955455377096757249?lang=de>
- [9] Source de la figure 2 : <https://steemit.com/german/@moodledidoodledi/die-internet-der-dinge-iot-revolution-iota>
- [10] Source de la figure 3 : <https://blog.iota.org/on-the-tangle-white-papers-proofs-airplanes-and-local-modifiers-44683aff8fea>

# détecteur d'évènement nucléaire

## drôle de composant n°40

Neil Gruending

(Canada)

J'ai découvert les détecteurs d'évènement nucléaire par le biais d'un tweet en montrant un en photo. Comme je n'en avais jamais vu auparavant, j'ai voulu en savoir plus. Bien m'en prit, ces détecteurs à usages divers sont des composants passionnants.

Un détecteur d'évènement nucléaire décèle la présence d'un rayonnement gamma et active sa sortie lorsque le niveau mesuré excède un certain seuil. Il existe différentes façons de mesurer un tel rayonnement (avec un compteur Geiger p. ex.) mais les détecteurs à jonction PN se révèlent ici les plus pratiques puisqu'ils produisent un photocourant proportionnel au rayonnement incident.

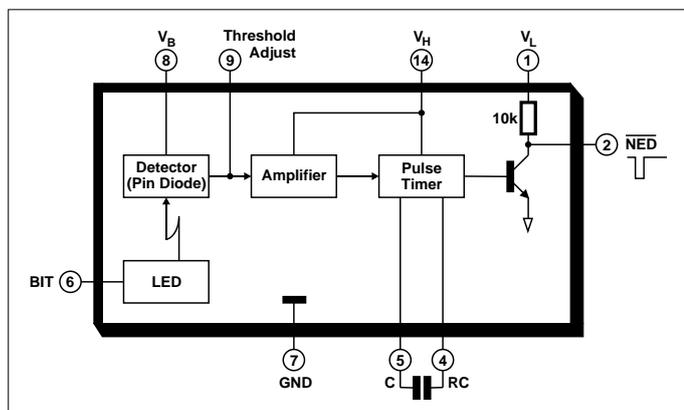
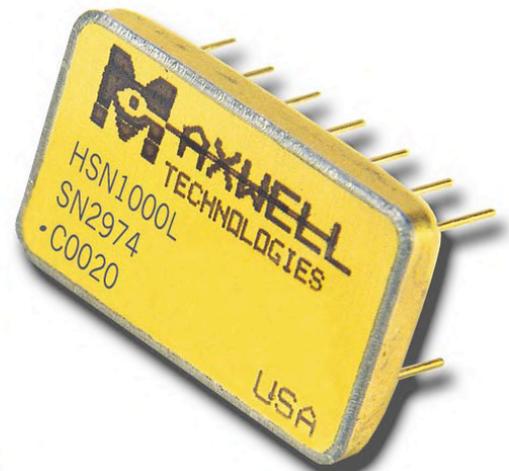


Figure 1. Schéma de principe du HSN1000 [1].

La puce HSN1000 de Maxwell Technologies (**fig. 1**) fait partie de cette famille de détecteurs. Elle utilise une diode PIN semblable à une diode PN ordinaire, si ce n'est qu'elle possède entre ses régions P et N une région supplémentaire non-dopée, dite région intrinsèque, qui accroît sa rapidité et sa sensibilité de détection. Sa sortie est reliée à un amplificateur qui active un temporisateur à impulsion (*Pulse Timer*) produisant un signal de niveau bas (appelé  $\overline{NED}$ ) en cas de détection gamma. Un des intérêts de ce signal est qu'il peut être exploité pour protéger un circuit électronique. Les composants résistants aux effets du rayonnement étant en effet coûteux, une façon de



Source: [www.maxwell.com/images/documents/hsn1000\\_rev3.pdf](http://www.maxwell.com/images/documents/hsn1000_rev3.pdf)

protéger un système est de le mettre dans un état « sûr » ou de le désactiver temporairement lorsqu'un évènement de type ionisant est détecté.

Sans cette mesure, les semi-conducteurs irradiés pourraient entraîner un fonctionnement erratique du processeur et donc des erreurs logicielles, voire subir un verrouillage à l'état passant ou un claquage. Les risques de verrouillage à l'état passant ne doivent pas être ignorés, car dans ce cas la broche d'E/S logique peut rester bloquée dans un état impossible à réinitialiser. Le claquage de la jonction dépend du type du transistor ainsi que de l'intensité du photocourant induit par le rayonnement.

Les détecteurs d'évènement nucléaire sont avant tout utilisés dans des applications militaires, mais leur sensibilité aux rayons gamma est également exploitée dans les astronefs et les satellites. Par le passé, les systèmes critiques des engins spatiaux étaient protégés au moyen de divers blindages, mais ce surpoids augmentait le coût du lancement. De nos jours, l'industrie spatiale privilégie donc les composants résistants aux effets du rayonnement et les détecteurs d'évènement nucléaire, plus légers qu'un blindage.

Ces composants sont malheureusement chers et difficiles à trouver, mais il s'agit sans doute d'un mal pour un bien puisque leur déclenchement nécessite une source de rayons gamma... Au moins nous auront-ils permis de découvrir un usage insolite des diodes PN. ◀

(180575-B-04 - version française : Hervé Moreau)

### Lien

[1] [Fiche technique du HSN1000 de Maxwell Technologies : www.maxwell.com/images/documents/hsn1000\\_rev3.pdf](http://www.maxwell.com/images/documents/hsn1000_rev3.pdf)

# amélioration avec une horloge en temps réel précise

## retouche du réveil à triple affichage

Mathias Claußen (labo d'Elektor) et Thomas Scherer (Allemagne)



Il n'y a rien de plus agaçant qu'une horloge imprécise. Prenez un réveil : aussi belle que soit son esthétique ou génial son mécanisme, s'il vous réveille trop tôt ou trop tard ou s'il faut le remettre à l'heure tous les mois, il n'aura pas rempli sa fonction essentielle. Grâce à la carte de liaison (*Breakout Board*, BoB) décrite dans cet article, voilà qui est facile à améliorer sur les horloges qui utilisent la puce d'horloge en temps réel (RTC) DS 1302 ou un modèle équivalent. Voyons cela de plus près.

Il y a un an est parue dans Elektor la description du « réveil à triple affichage » d'Olivier Croiset [1], un module d'horloge compact équipé d'un afficheur graphique LCD, qui m'a tout de suite séduit. C'était un réveil d'utilisation relativement simple. Voilà qui convenait parfaitement à ma mère qui est sur le pied de guerre avec le mode d'emploi plutôt cryptique des réveils modernes et des horloges d'origine chinoise. C'est pourquoi j'ai assemblé le kit disponible chez Elektor [2] et publié mon expérience dans un banc d'essai [3].

### Imprécisions

Comme décrit dans mon banc d'essai, le montage fut simple et ma mère ravie d'un réveil d'aspect séduisant et qu'elle pouvait régler elle-même (fig. 1). Mais sa félicité fut rapidement troublée par la tendance de ce réveil à vouloir prendre de l'avance sur le mouvement de la Terre. En quatre mois, cette avance se montait déjà à 24 min (j'avais prié ma mère de ne pas toucher au réveil pour me permettre d'évaluer sa dérive avec précision). Cela faisait six intolérables minutes par mois, soit presque douze secondes par jour. Comment cela était-il possible ?

Le réveil lit l'heure sur une puce RTC DS 1302 cadencée (comme beaucoup d'autres horloges) par un quartz classique de 32.768 Hz. Une réserve de quelques heures de fonction-



Figure 1. Mon « réveil à triple affichage » dans un boîtier en aluminium noir.

nement en cas de coupure du secteur etc. est fournie par un super-condensateur. La puce est certainement innocente, le fautif est indubitablement le quartz. Je suis sûrement tombé sur

un mouton noir, car sans commune mesure avec les  $\pm 50$  ppm spécifiés pour ce genre de quartz. Le mien était près de trois fois pire !

## Solutions

Que faire ? Changer le quartz et espérer ? Ces quartz d'horloge sont fabriqués en Extrême-Orient en millions d'exemplaires à quelques centimes. Et ils respecteraient tous la tolérance spécifiée ? J'ai comme un doute... Mais ouvrir le réveil plusieurs fois pour remplacer le quartz jusqu'à ce que ça marche bien, non merci ! Alors, je me suis mis en quête d'oscillateurs plus précis pour une vraie mise à niveau. Et juste à ce moment-là, j'ai appris qu'au labo d'Elektor, Mathias Claußen travaillait sur une option de mise à niveau qui améliorerait la précision de ce réveil. Transmission de pensée ? Magie ?

En tant qu'électronicien, j'admets, bien sûr, les coïncidences, ce qui n'a pas gâché le plaisir que m'a apporté celle-ci. Et il y a eu d'autres points d'accord : comme moi, Mathias était arrivé à la conclusion qu'un oscillateur à quartz super précis et compensé en température pour remplacer le tout-venant reviendrait trop cher ou aurait une consommation pénalisante qui épuiserait trop vite le super-condensateur de secours. Il fallait donc trouver un remplaçant pratique et suffisamment précis. Matthias se décida pour le ASH7KW, un « 32.768kHz Low Power SMD Crystal Oscillator » d'Abrakon, dont la consommation typique est de  $1,8 \mu\text{A}$  seulement et la tension d'alimentation aussi basse que  $1,2 \text{ V}$ . C'est le candidat idéal pour une alimentation par super-condensateur. Sa précision est spécifiée à  $\pm 20$  ppm. Avant que vous fassiez la moue devant ce chiffre pas terriblement meilleur que celui du quartz tout-venant, jetez donc un coup d'œil sur la courbe de tolérance en fonction de la température sur la feuille de caractéristiques [4]. Dans la zone des températures habituelles d'une chambre à coucher ( $15$  à  $30 \text{ }^\circ\text{C}$ ),

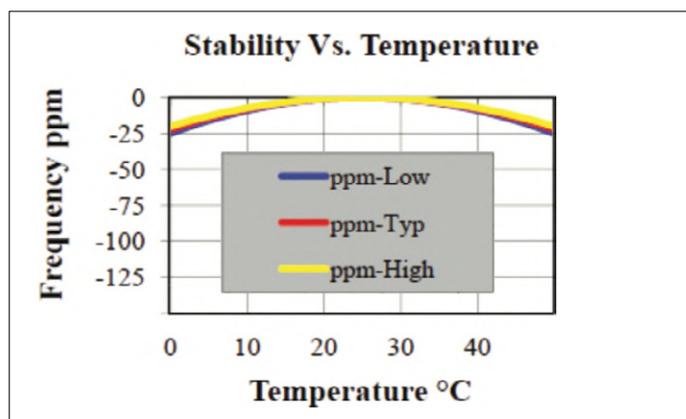


Figure 2. Extrait de la courbe dérive/température de la feuille de caractéristiques de la puce ASH7KW d'Abrakon.

la précision se situe dans un domaine de  $\pm 10$  ppm maximum, et plutôt en-deçà de  $\pm 5$  ppm (fig. 2). On devrait donc pouvoir compter sur une dérive mensuelle inférieure à 13 s, soit de 2,5 mn par an. On est donc dans une tout autre classe, d'autant plus appréciable que l'investissement reste très limité.

## Réalisation

Si l'on considère le circuit de la figure 3, il est évident qu'il n'y a pas grand-chose à voir ! J1 est constitué d'embases mâles ou de broches isolées dans une configuration DIL 8 ; presque toutes les broches sont reliées aux pattes de IC1 (une puce DS 1302 au format CMS). Après tout, la carte de liaison doit remplacer directement la puce DS 1302 au format DIL sur la carte du réveil. Le signal d'horloge désormais bien plus précis fourni par l'oscillateur X1 se trouve sur « X1 », c'est-à-dire la broche 2 de IC1.

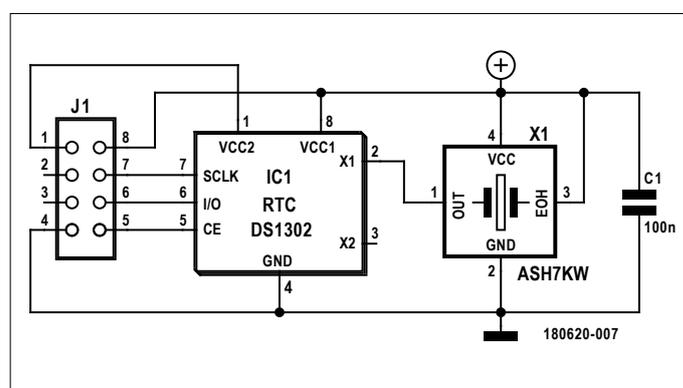


Figure 3. Le circuit de la mise à niveau de la précision est particulièrement simple.

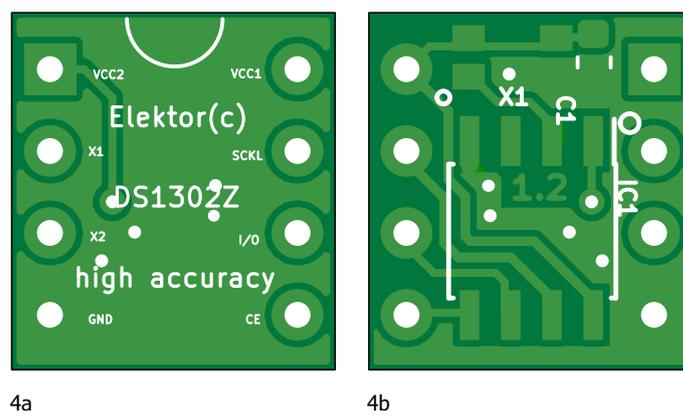


Figure 4. Faces supérieure (4a) et inférieure (4b) du minuscule BoB.

## Liens

- [1] Réveil à triple affichage, Elektor 05-06/2018 : [www.elektormagazine.fr/160590](http://www.elektormagazine.fr/160590)
- [2] Kit du réveil à triple affichage avec afficheur TFT 2,2" : [www.elektor.fr/3-way-display-alarm-kit](http://www.elektor.fr/3-way-display-alarm-kit)
- [3] Banc d'essai du kit du réveil à triple affichage : [www.elektormagazine.fr/news/banc-d-essai-kit-du-reveil-a-triple-affichage-elektor](http://www.elektormagazine.fr/news/banc-d-essai-kit-du-reveil-a-triple-affichage-elektor)
- [4] Oscillateur ASH7KW : <https://abracon.com/Oscillators/ASH7KW.pdf>

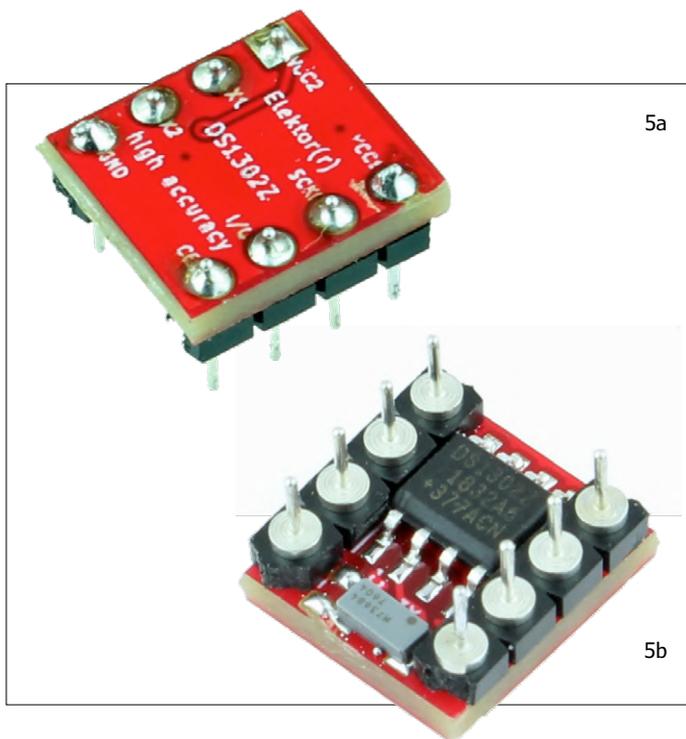


Figure 5. Faces supérieure (5a) et inférieure (5b) du prototype équipé.

Ce circuit électronique minimal est câblé sur la carte de liaison (voir **figure 4**). Les broches, IC1, X1, ainsi que le condensateur de découplage C1 sont tous montés sur la face inférieure de la carte. Pour les broches, on peut en utiliser des cylindriques ou bien, par défaut, huit chutes de fil rigide, par exemple des extrémités de résistances traversantes, qu'on coupe à la bonne longueur après le soudage. L'ordre d'assemblage des composants est le même que d'habitude, du plus petit au plus grand, donc d'abord X1 et C1, puis IC1 et enfin les broches. Le BoB terminé devrait avoir l'aspect de la **figure 5**.

### Mise à niveau

Avec ce BoB comme base de temps précise, on peut en principe mettre à niveau n'importe quelle horloge utilisant une puce du type DS 1302 ou équivalente en boîtier DIL. Dans le cas du « réveil à triple affichage », il faut tenir compte de certaines

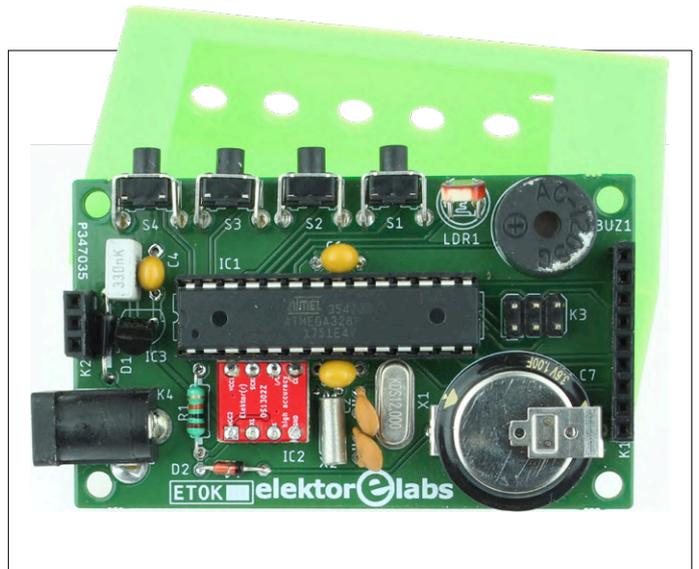


Figure 6. Carte du réveil avec le BoB installé à la place de la puce RTC.



Figure 7. Sandwich composé de la carte du réveil et du module d'affichage débarrassé de son logement pour la carte SD.

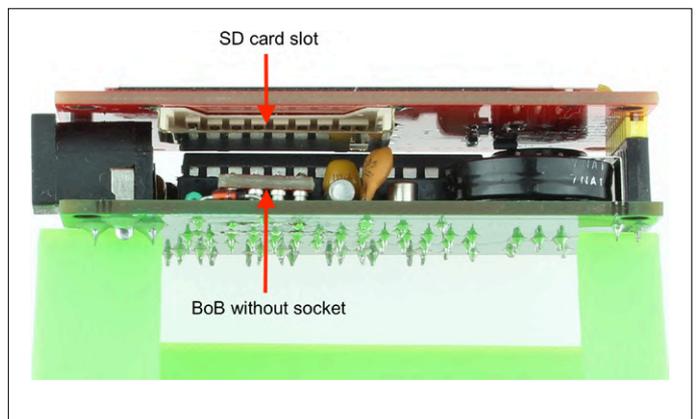


Figure 8. Sandwich composé de la carte du réveil avec le BoB soudé directement (sans embase) et du module d'affichage avec le logement de la carte SD conservé.

## LISTE DES COMPOSANTS

**Condensateur**  
C1 = 100 nF, CMS 0603

**Semi-conducteurs**  
IC1 = DS1302, SOIC8  
X1 = ASH7KW, 32.768 Hz, CMS

**Divers**  
Embase mâle à 2 × 4 contacts, au pas de 2,54 mm, voir le texte  
Circuit imprimé réf. 180620-1

particularités, ce qui justifie une description plus détaillée de l'installation.

Sur ce réveil, la puce RTC DS 1302 est embrochée dans une embase à 8 broches. L'échange est en principe facile : on ouvre le réveil et on déconnecte l'afficheur de la carte du réveil. On déloge ensuite la puce RTC (IC2) de son embase dans laquelle on embroche le BoB dans le bon sens. Pour cela, les broches sont repérées par des inscriptions sur la face supérieure du BoB. Pour l'orientation, on peut aussi s'en remettre à la **figure 6**. Mais ce n'est pas aussi simple ! Le module d'affichage possède au verso un logement pour une carte SD dont la tôle mettrait les broches du BoB en court-circuit si on le remettait tout simplement en place sur la carte du réveil. Pour résoudre ce problème, il y a trois solutions simples :

- On supprime le logement pour carte SD sur le module d'affichage. Si on remet en place le module débarrassé de cette tôle sur la carte du réveil, on obtient ce que l'on voit sur la **figure 7**. Il y a assez de place pour le BoB.
- On dessoude l'embase de la puce sur la carte du réveil. Cette opération est plus facile si l'on commence par enlever les parties en plastique en les morcelant avec une pince coupante diagonale et qu'on dessoude ensuite les broches une à une. Après enlèvement de la soudure au moyen d'une tresse à dessouder, on peut souder directement le BoB et on ne sera pas gêné par le logement de la carte SD. La **figure 8** montre cette solution.
- On installe les embases mâles sur la face supérieure du BoB et on soude les broches au verso de la carte du réveil, à l'endroit même où est soudée l'embase de IC2. Mais comme cela augmente l'épaisseur du réveil et lui donne un aspect « bricolé », nous avons renoncé à illustrer cette solution.

## Conclusion

Cette amélioration de la précision est une opération simple, gratifiante et très bon marché. On peut ainsi facilement faire le « réveil à triple affichage » et quelques autres horloges comportant une puce DS 1302 en boîtier DIL. Pour moi, cette solution s'est présentée juste au bon moment et m'a épargné bien du bricolage et, grâce au BoB, un câblage sauvage.

Merci Mathias ! ◀

(180620-03 – version française : Helmut Müller)



## Un événement oublié ?

Vous organisez une conférence, un salon... ou bien vous participez à un séminaire ou tout autre événement qui aurait sa place ici, partagez cette information avec tous les lecteurs.

Envoyez-nous tous les détails à [redaction@elektor.fr](mailto:redaction@elektor.fr).

## juin 2019

- ◇ **6<sup>èmes</sup> assises nationales des énergies marines renouvelables**  
04/06 - Dunkerque  
[www.assises-nationales-emr.fr](http://www.assises-nationales-emr.fr)
- ◇ **Edge cloud summit 2019**  
04 au 06/06 - Paris  
[www.uppersideconferences.com/edgecloud](http://www.uppersideconferences.com/edgecloud)
- ◇ **MIDEM**  
04 au 07/06 - Cannes  
[www.midem.com/](http://www.midem.com/)
- ◇ **Web2day**  
(festival des professionnels et des passionnés des nouvelles technos)  
05 au 07/06 - Nantes  
[web2day.co/](http://web2day.co/)
- ◇ **Le Temps des Hélices**  
(meeting aérien)  
08 au 09/06 - La Ferte-Alais  
[www.ajbs.fr/presentation-du-meeting-2019/](http://www.ajbs.fr/presentation-du-meeting-2019/)
- ◇ **AI Paris 2019**  
11 au 12/06 - Paris  
[www.aiparis.fr](http://www.aiparis.fr)
- ◇ **Forum Teratec 2019**  
(simulation et big data)  
11 au 12/06 - Palaiseau  
[www.teratec.eu/forum/](http://www.teratec.eu/forum/)
- ◇ **Conférence européenne des essais et télémesure (ETTC)**  
11 au 13/06 - Toulouse  
[www.see.asso.fr/ETTC2019](http://www.see.asso.fr/ETTC2019)
- ◇ **Expobiogaz**  
12 au 13/06 - Lille  
Salon du gaz renouvelable
- ◇ **Global LiFi Congress**  
12 au 13/06 - Paris  
[www.lificongress.com/](http://www.lificongress.com/)
- ◇ **Salon international de l'aéronautique et de l'espace**  
17 au 23/06 - Paris  
[www.siae.fr/](http://www.siae.fr/)
- ◇ **Leti Innovation Days**  
24 au 28/06 - Grenoble  
[www.leti-innovation-days.com/](http://www.leti-innovation-days.com/)
- ◇ **NanoMatEn & Nanometrology & Nanotech**  
26 au 28/06 - Paris  
[www.setcor.org/conferences](http://www.setcor.org/conferences)



@ **WWW.ELEKTOR.FR**

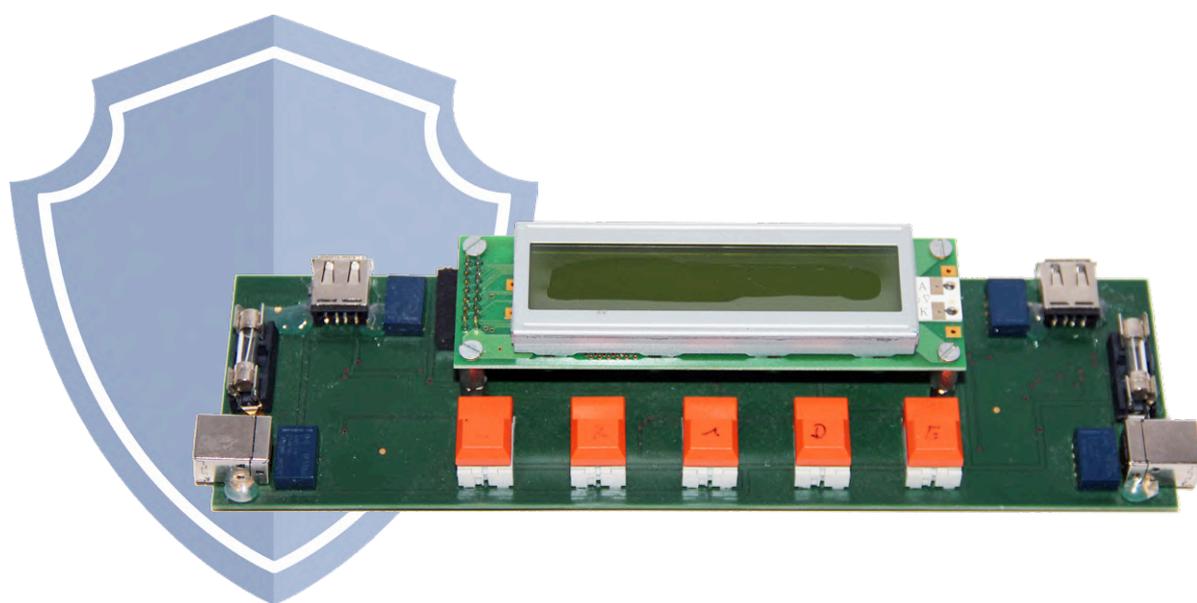
→ Kit du réveil à triple affichage avec afficheur TFT de 2,2"  
[www.elektor.fr/3-way-display-alarm-kit](http://www.elektor.fr/3-way-display-alarm-kit)

→ Circuit imprimé nu de l'horloge en temps réel (RTC) précise (180620-1 v1.2)  
[www.elektor.fr/18831](http://www.elektor.fr/18831)

→ Carte de liaison de l'horloge en temps réel (RTC) partiellement garnie  
[www.elektor.fr/18832](http://www.elektor.fr/18832)

# surveillance de la consommation sur port USB

## protection intelligente pour ports USB



René Gräber

L'idée de ce projet est née quand j'ai expérimenté un circuit maison alimenté (comme bien souvent) avec le 5 V fourni par le port USB de mon PC. Le système a émis le message suivant : « Un périphérique a dépassé la limite de puissance du concentrateur USB ». Malheureusement la communication avec mon circuit passait aussi par cette connexion USB, ce qui a compliqué quelque peu l'analyse du défaut. J'avais besoin d'un outil de mesure de la consommation de mon circuit qui ne désactive pas immédiatement le port en cas de surcharge de courte durée. De plus, le PC devait être protégé et la communication USB maintenue, même lorsque l'alimentation est externe. Voici ma solution.

Grâce à mon expérience (douloureuse) de l'alimentation par l'USB, je savais que les 500 mA spécifiés ne sont pas toujours respectés et que, le plus souvent, les ports USB ne sont pas suffisamment protégés.

Ce projet a été conçu en deux étapes. Le circuit imprimé de la première version était divisé en plusieurs pièces séparées de petite taille, interconnectées à l'aide d'embases mâle et femelle. Cette version a été réalisée avec Eagle et conçue de manière à pouvoir graver les circuits par fraisage. Il y a eu ensuite une deuxième version sous la forme d'un circuit imprimé unique,

pour lequel j'ai utilisé le logiciel DesignSpark. Le schéma de cette version est représenté sur la **figure 2**.

### À propos du matériel

Le circuit comporte deux ports USB pour connecter des appareils : J4 et J5. La connexion au PC s'effectue par le port K2 ou K3. La mesure du courant est effectuée par l'une ou l'autre des deux puces INA219 de Texas Instruments [1]. La puce INA219 est un moniteur de courant/puissance avec interface I2C ; elle est lue cycliquement. On utilise un moniteur par

canal. On mesure la tension aux bornes des résistances R19 et R30 et on en déduit le courant qui les traverse. Les groupes de résistances R22 à R25 et R33 à R36 servent à définir les adresses I2C.

Les relais Re1 et Re3 permettent de mettre l'appareil connecté sous ou hors tension. Les fusibles F1 et F2 protègent contre les surintensités. La commande des relais est confiée au microcontrôleur U1.

Les relais Re2 et Re4 permettent de remplacer les 5 V USB du PC (délivrés à l'objet de la mesure) par une alimentation de 5 V locale. Cette alimentation arrive par J2 et alimente aussi le reste du circuit.

Le cœur du circuit est le microcontrôleur PIC18F6620 [2]. Les LED1 à LED4 permettent d'identifier l'état de l'appareil, les alarmes et défauts éventuels. Sur ses entrées AN1 et AN2, le microcontrôleur mesure les tensions délivrées sur les bornes d'alimentation des appareils à mesurer (J4 et J5). Sur son entrée AN0 (EXT\_PWR), il mesure la tension de l'alimentation locale amenée sur J2. Pour ces trois mesures, les tensions sont réduites d'un facteur 4 par des diviseurs résistifs (R13 à R18). Ce facteur 4 est pratique, car il suffit au microcontrôleur d'appliquer un décalage de 2 bits pour obtenir la valeur exacte de la tension. Une puce MAX6004 [3] fournit la tension de référence de 4,096 V pour le convertisseur A/N à 10 bits embarqué. Les touches S1 à S5 sont connectées aux entrées d'interruption (INT) et clavier (KB) du microcontrôleur ; le micrologiciel s'occupe de leur anti-rebond et de leur analyse.

J1 est le connecteur de programmation in situ et JP1 fournit une interface de type UART 5 V pour communiquer avec un PC. Les valeurs de mesure sont affichées sur un afficheur standard de 2x20 caractères, relié au microcontrôleur par quatre fils pour les données et trois fils pour les signaux de commande RS, R/W et E. Le potentiomètre R10 permet de régler le contraste.

### Listage 1. Calcul des valeurs de configuration de l'INA219

```

/* Calcul de la valeur du registre de
 * configuration selon la feuille de
 * caractéristiques
 * R-Shunt = 0R1
 * V-BusMax 6V
 * I - Max 2A2
 * Min LSB = 2A2/32767 => 0,00006714
 * Max LSB = 2A2/4096 => 0,0005371
 * Choose an LSB in the range: Minimum_LSB
 * Current_LSB = 0,00007
 * Cal = trunc[0,04096/(Current_LSB * R-Shunt)]
 = 0,04096/(0,00007*0r1) => 5851
 * Power_LSB = 20 * Current_LSB = 20 * 0,00007 =
 0,0014
 * Max_Current = Current_LSB * 32767 = 0,00007 *
 32767 => 2,29369A
 */

```

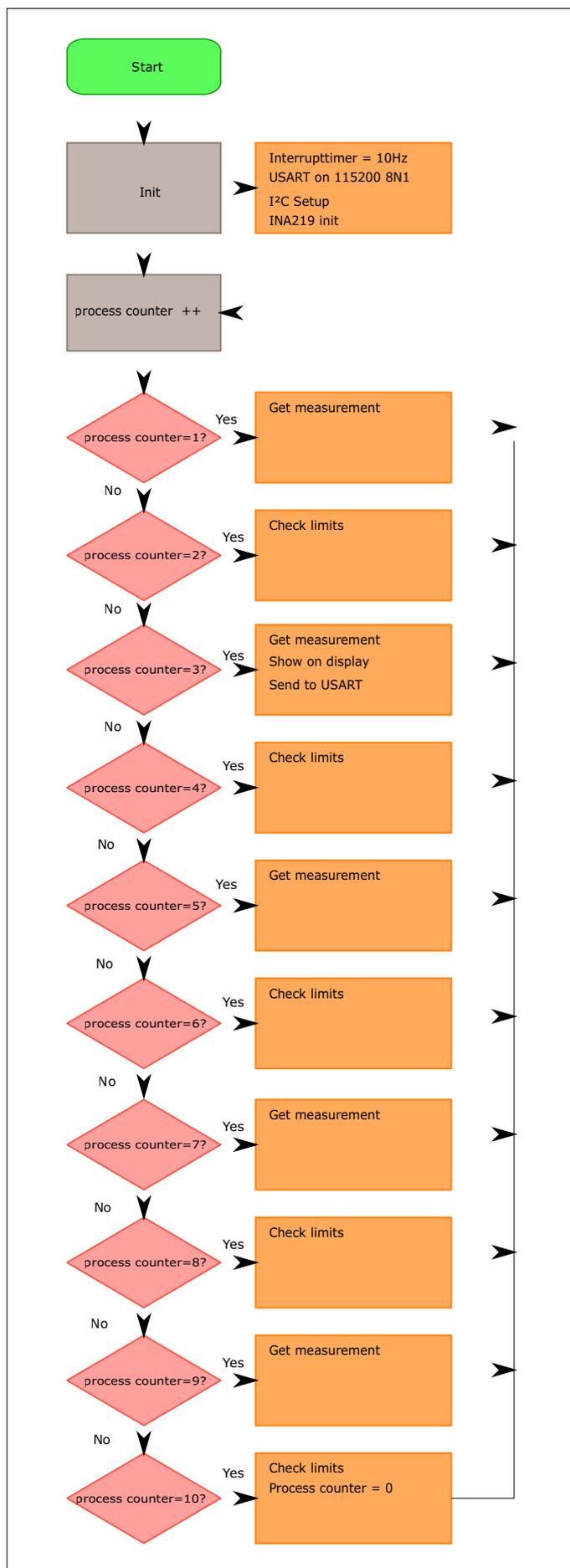


Figure 1. Organigramme de l'exécution du micrologiciel dans le microcontrôleur.

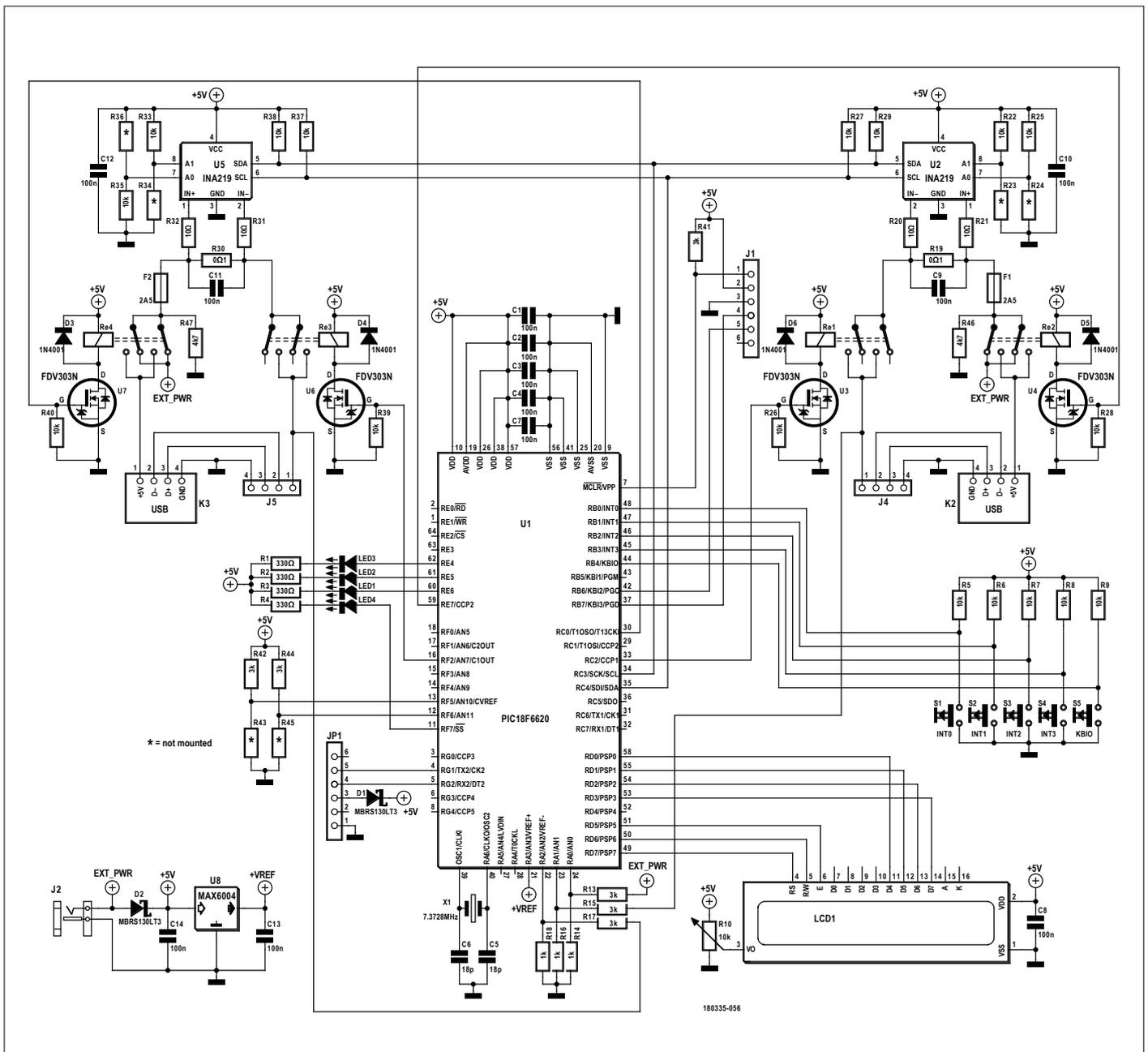


Figure 2. Schéma de la version définitive.

### Rien ne va sans logiciel

Le microgiciel a été écrit en C sous l'EDI MPLAB X. La **figure 1** en présente l'organigramme. Le registre de « calibrage » de l'INA219 doit être initialisé au démarrage. Le capteur utilise les paramètres de calibrage pour calculer les valeurs du cou-

rant et de la puissance. Les paramètres sont évalués à partir des besoins du projet et des résistances de shunt utilisées. Le **listage 1** donne le calcul. Le projet utilise la valeur 5851, qui est écrite dans le registre des deux INA219. Les données des capteurs sont lues périodiquement et traitées

### Liens

- [1] Feuille de caractéristiques de l'INA219 : [www.ti.com/product/INA219](http://www.ti.com/product/INA219)
- [2] PIC18F6620 : [www.microchip.com/wwwproducts/en/PIC18F6620](http://www.microchip.com/wwwproducts/en/PIC18F6620)
- [3] MAX6004 : [www.maximintegrated.com/en/datasheet/index.mvp/id/1916](http://www.maximintegrated.com/en/datasheet/index.mvp/id/1916)
- [4] Page de l'article : [www.elektormagazine.fr/180335-B-04](http://www.elektormagazine.fr/180335-B-04)

## Listage 2. Le gestionnaire de tâches est le cœur du microciel.

```
/******  
/! \fn      void tast_manager(void)  
* \brief    update the tasks  
* \param    none  
* \exception none  
* \return   none  
*****/  
void task_manager(void){  
  
    static uint8_t prozesscounter = 0;  
  
    update_Status_Task();  
    // Switch case pour l'exécution successive  
    // des tâches  
    switch(prozesscounter++){  
        case 1:  
        case 5:  
        case 7:  
        case 9:{  
            // Mise à jour des valeurs de mesure  
            // toutes les 100 ms  
            update_measurements_task();  
            break;  
        }  
        case 2:  
        case 4:  
        case 6:  
        case 8:{  
            //Test des limites  
            check_limits();  
            break;  
        }  
        case 3: {  
            update_displaystring();  
            update_measurements_task();  
            break;  
        }  
        case 10: {  
            prozesscounter = 0;  
            check_limits();  
            break;  
        }  
    }  
    application_run();  
}
```

par le microcontrôleur. Si l'une des valeurs limites réglées est franchie, le microcontrôleur coupe le courant qu'il ne rétablit qu'après que l'utilisateur a réactivé le canal. De plus, le microcontrôleur doit afficher une alarme avant le déclenchement. Le programme d'initialisation écrit la valeur de calibrage dans les deux INA219, règle la vitesse du port série sur 115200 bauds avec 8N1 et fixe la fréquence des interruptions pour le gestionnaire de tâches à 10 Hz. Pour les autres paramètres de calibrage, leur dernière valeur enregistrée est récupérée dans l'EEPROM interne. Pour l'enregistrement des données, il est prévu dix emplacements dans l'EEPROM du contrôleur. Le gestionnaire de tâches démarre toutes les 100 ms un nouveau traitement des tâches (récupérer les valeurs, les vérifier, les afficher). À chaque exécution, le programme incrémente la variable `prozesscounter`. Un aiguillage `switch` contrôle les exécutions des différentes tâches.

### Menu utilisateur

La fonction `application_run()` est exécutée à la fin de chaque passe. Derrière cette fonction se dissimule le menu utilisateur. Ce menu est conçu comme un automate à états finis où chaque partie de menu est gérée par une fonction spécifique. Pour passer d'un menu à un autre, le programme avance simplement un pointeur vers la fonction à exécuter à chaque nouveau passage dans la boucle.

Le menu permet de programmer ses propres valeurs de courant et de tension pour déclencher une alarme ou signaler un défaut. Le franchissement de l'une de ces valeurs est signalé par l'allumage de l'une des LED d'état. En cas de défaut, l'alimentation du canal concerné est coupée.

Les deux lignes d'affichage ont le format suivant :

```
K 0000mV 0000mA on_off_X, in_ext_X
```

où :

- K => numéro de canal (1 ou 2)
- 0000mV => valeur courante de la tension en mV
- 0000mA => valeur courante de l'intensité en mA
- on\_off\_X => canal actif ou non, ou en alarme ou en défaut
- in\_ext\_X => canal alimenté par l'USB ou en local

### Téléchargements

Tous les fichiers du projet, programmes et documents DesignSpark du projet sont disponibles au téléchargement gratuit sur la page web du projet [4]. ◀

(180335-B-04 - version française : Helmut Müller)

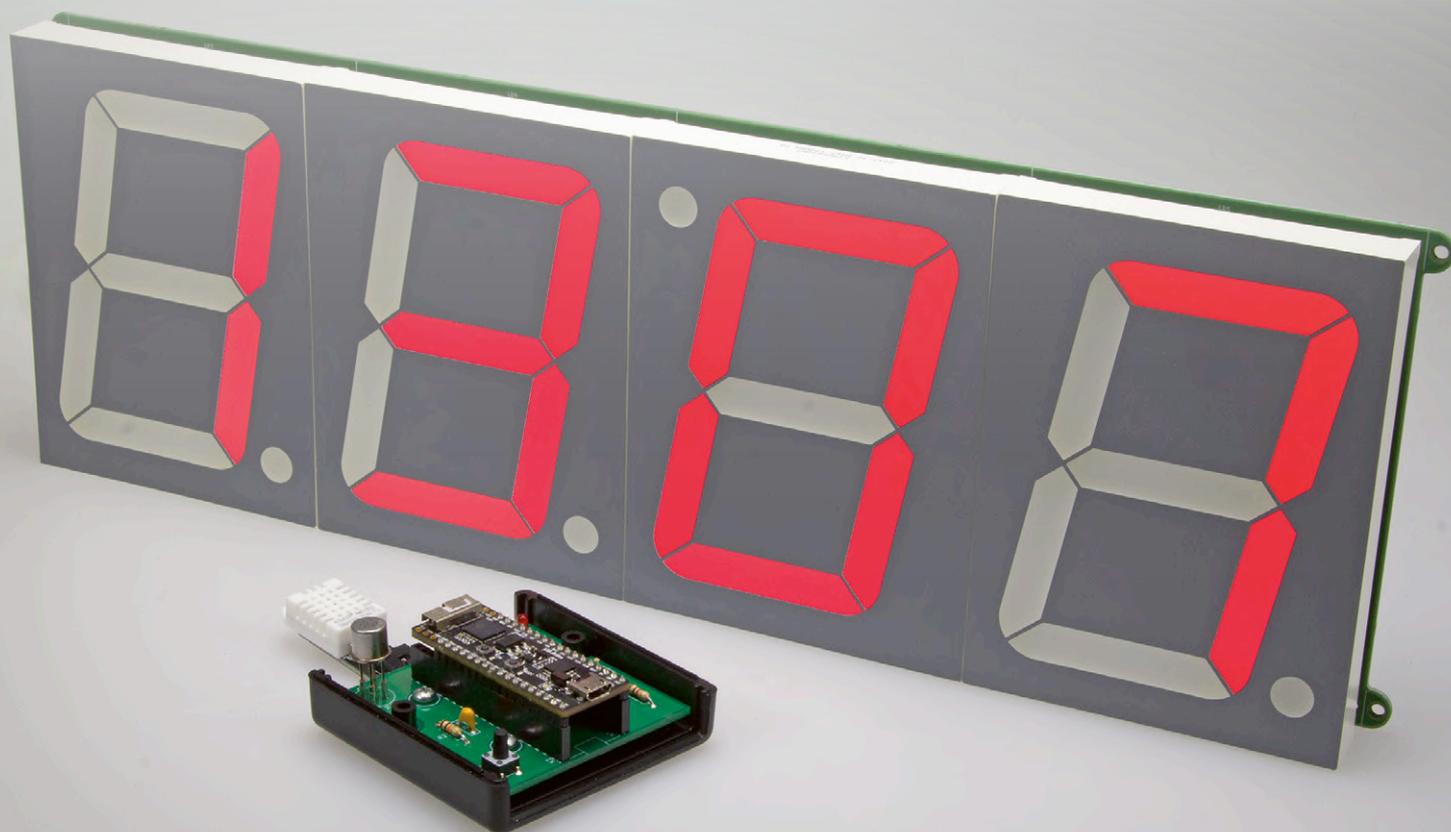


@ WWW.ELEKTOR.FR

→ Carte de liaison USB pour expérimenter avec l'USB  
[www.elektor.fr/usb-breakout](http://www.elektor.fr/usb-breakout)

# horloge à LED géante avec Wi-Fi et mesures météo

des chiffres hauts de 10 cm, le temps d'internet par ESP32 et des capteurs d'ambiance sans fil



**Mathias Claußen** (labo d'Elektor) et **Rainer Krugmann** (Allemagne)

Les petites cartes à microcontrôleur ESP32 qui peuvent se connecter à un réseau local sans fil (Wi-Fi) constituent une formidable base pour la réalisation de projets. Rainer Krugmann était bien de cet avis quand il s'est lancé dans la construction d'un afficheur géant à chiffres à sept segments qu'il s'était procurés quelques années auparavant. Avec beaucoup de soins dans la conception, son idée de départ a évolué vers une véritable horloge qui, grâce à l'interface MQTT, est devenue un excellent châssis pour ses projets et extensions. Le capteur sans fil pour mesurer la qualité de l'air, la température et l'hygrométrie était déjà prévu.

Tout d'abord, nous avons appris avec regret le décès, peu avant Noël 2018, de l'auteur et lecteur de longue date d'Elektor Rainer Krugmann. Ses enfants nous

ont demandé de publier malgré tout son projet. Comme cette création est très intéressante, Mathias Claußen du labo d'Elektor a accepté de reprendre

le circuit et de le terminer. Cette publication est donc à la fois un hommage et un remerciement en souvenir du temps passé ensemble !

## Un colosse ?

Attrayant dès l'abord, cet afficheur au format hors normes l'est certainement. Les quatre chiffres à sept segments du modèle SA40-19SRWA de Kingbright mesurent chacun 122 mm de haut et 90 mm de large, ils donnent une hauteur active de chiffre de 101,2 mm. L'ensemble constitue un afficheur de la largeur impressionnante de 36 cm, l'idéal pour une horloge que l'on peut lire facilement à cinq mètres de distance, c'est assez fantastique.

Bien entendu, on peut construire une horloge plus colossale encore avec des matrices de LED plus grandes, mais l'afficheur à sept segments, avec ses bords bien nets, a plus belle apparence. On n'y voit pas de pixels individuels, ce qui, à l'ère des écrans rétinés à haute résolution, serait assez anachronique. En outre les segments sont extrêmement plus simples à piloter. Et enfin vous n'avez pas besoin de passer du temps à sélectionner des LED d'égale luminosité et à recalibrer les autres. Une matrice réalisée avec des LED de 5 mm sur une hauteur de 10 cm demanderait facilement 1 000 LED et s'il fallait se procurer des matrices toutes faites de ce format-là, on devrait y mettre le prix. En revanche, les chiffres de 10 cm de Kingbright sont disponibles en différentes couleurs à moins de 15 € pièce, voire moins cher encore en cherchant bien sur l'internet.

## À l'heure ?

On comprend aisément pourquoi Rainer Krugmann a choisi le module à processeur en kit ESP32 Pico [1], même si, pour fabriquer une horloge à quatre chiffres, l'ATmega, le PIC ou l'Arduino aurait pu convenir. Avec son prix si petit, l'ESP32 est à peine plus cher qu'un microcontrôleur solitaire et tout de même facile à programmer avec l'EDI d'Arduino. La connectivité au Wi-Fi intégrée garantit une présentation de l'heure avec une précision absolue, exempte de la dérive d'un module d'horloge en temps réel (RTC), vu que le module peut l'obtenir directement de l'internet par l'un des nombreux serveurs NTP (*Network Time Protocol*). Du même coup, il se passe d'une réserve de marche sur batterie de secours pour pallier une panne de courant ou un simple déménagement. Les serveurs de temps sont toujours à l'heure exacte. Même s'il est vrai qu'une horloge doit donner l'heure, l'une ou l'autre fonction supplémentaire ne ferait pas de

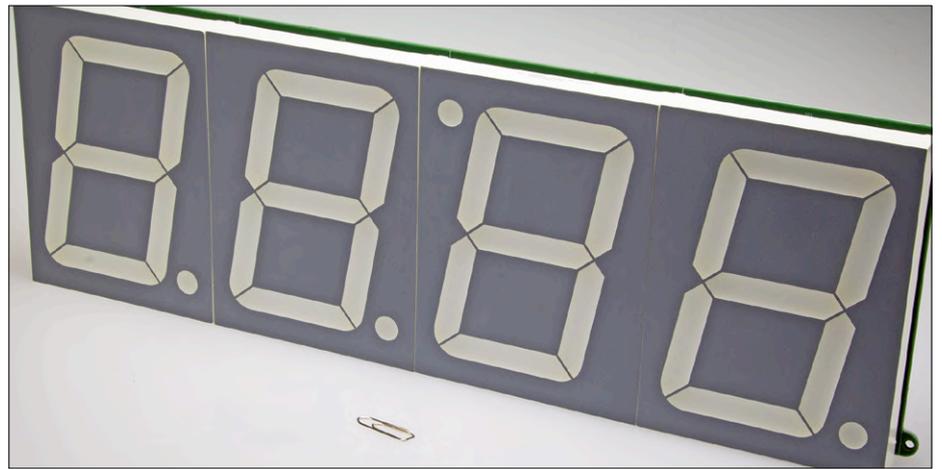


Figure 1. Un afficheur géant. Si vous ne le croyez pas, comparez-le au trombone.

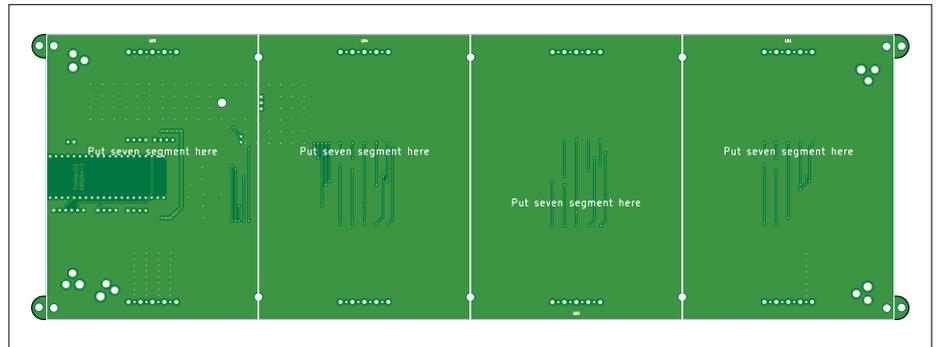


Figure 2. Côté composants ou verso ? Dans un cas comme dans l'autre, infligez astucieusement un demi-tour au chiffre LD2 lors de l'implantation.

mal. Si par exemple, au petit matin, vous avez envie de savoir quelle température il fait dehors, un thermomètre sans fil sera le bienvenu. Pour éviter les tracas d'un protocole propriétaire établi sur des fréquences libres, mais contaminées par les interférences, pourquoi ne pas interroger le capteur par internet ? Le plus simple est alors de recourir à un autre module ESP32 couplé à des capteurs existants. Il pourra tout aussi bien se connecter au Wi-Fi et profiter d'un trafic sécurisé des données dans la zone de couverture. Dans un élan de normalisation, c'est ici le protocole MQTT (Téléométrie Transport) [2] qui a été adopté. L'essentiel du projet est ainsi défini : une horloge visible de loin avec l'heure d'internet, mais qui permet aussi l'affichage d'autres données comme la température en passant par MQTT.

## Affichage, matériel et le reste

La **figure 1** donne à voir les quatre chiffres à sept segments LED de l'affi-

chage monumental de l'horloge produite au labo d'Elektor. À bien y regarder, vous constaterez une particularité : le troisième chiffre depuis la gauche est monté à l'envers. Un petit truc qui permet de rassembler les points décimaux des deuxième et troisième chiffres pour reproduire l'habituel double point clignotant entre les heures et les minutes, coutumier des horloges numériques. Il faudra y penser lors du soudage sur le circuit imprimé de la **figure 2**.

La commande des afficheurs nécessite des puces de pilotage. Par souci de réduction du câblage nécessaire à la commande, celle-ci est multiplexée, ce qui demande un pilote pour les chiffres et un autre pour les segments. On aboutit ainsi au circuit de la **figure 3**, un schéma d'une grande clarté. Pour les anodes communes des quatre chiffres, on a pris quatre des huit lignes du pilote IC3 comme sources de courant fort, commutées sur le +12 V, tandis que les huit drains Darlington de IC2 ramènent à la

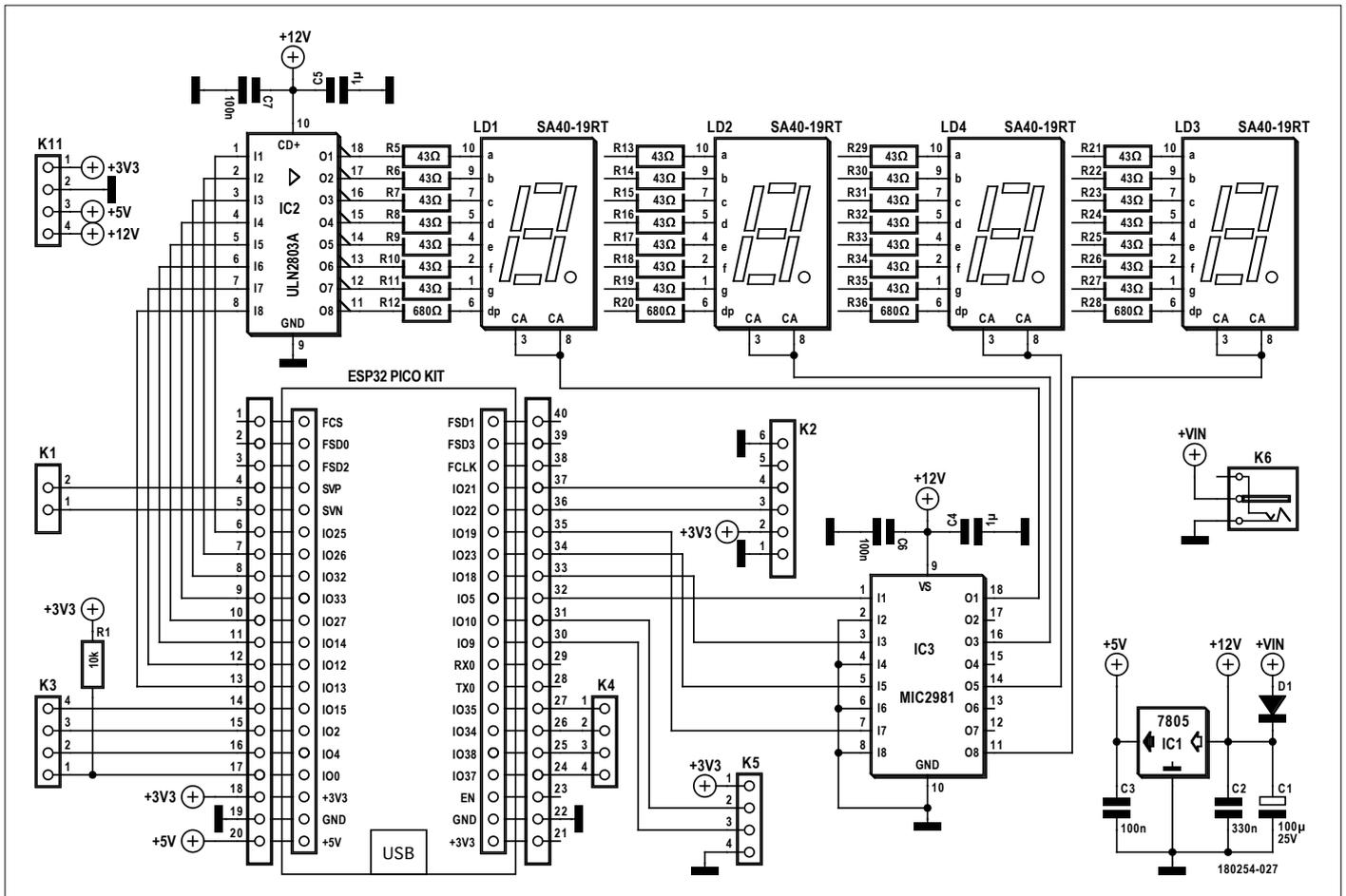


Figure 3. Si le circuit de l'horloge au complet avec affichage reste assez simple, c'est grâce aux performances de l'ESP32.

masse chacun des segments. On notera que la luminosité des afficheurs dépend à la fois de la tension d'alimentation de +12 V et du rapport cyclique de la MLI aux commandes. On utilise pour cela une alimentation secteur

stabilisée de 12 V. Avec une tension plus haute, il faudrait absolument adapter les valeurs de R5 à R28. Il en va de même si vous remplacez, à brochage compatible, les chiffres rouges par des LED d'une autre couleur, puisque les tensions de

seuil des LED en dépendent. Par segment, plusieurs LED sont mises en série, il y en a moins pour les points. On dénombre plusieurs variantes en exécution normale et super brillante en rouge, vert et jaune-orange, parmi les-

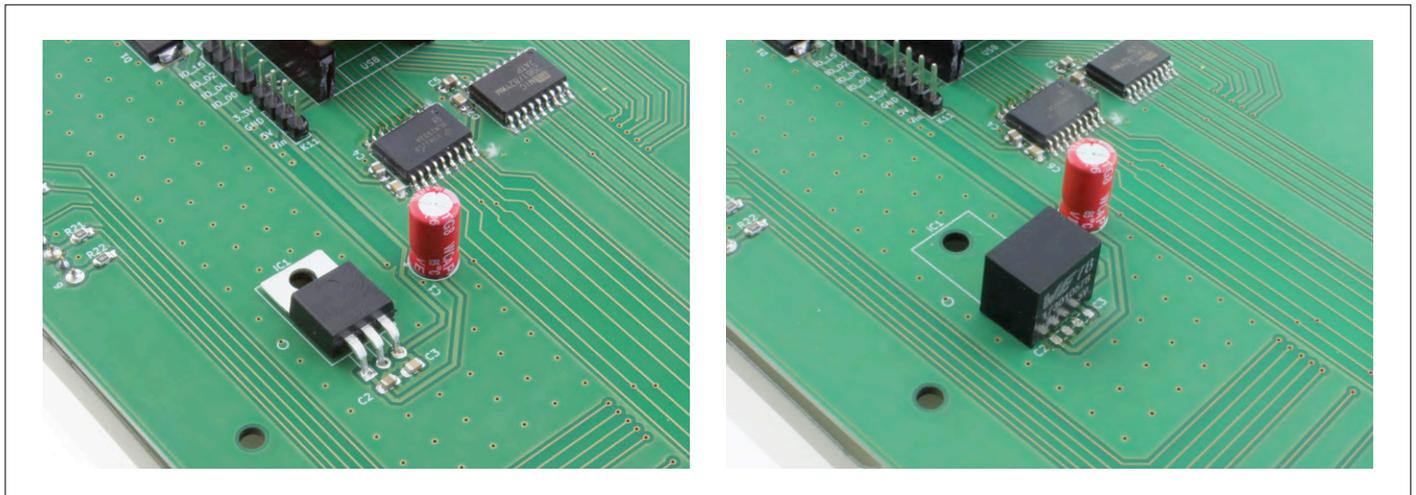


Figure 4. Très aisé d'installer un convertisseur CC/CC de Würth à la place du 7805, il réduit la dissipation.

quelles les afficheurs rouges super brillants SA40-SRWA à 7,4 V et les verts jusqu'à 8,8 V par segment, ce qui ne donne pas encore de grandes différences. Leur choix dépend des préférences de chacun et de l'endroit où on les met. Dans une chambre à coucher, les super brillants ne sont pas le meilleur choix. C'est à partir de l'alimentation de 12 V que le régulateur linéaire IC1 stabilise le +5 V pour la carte ESP32, puis de là un autre régulateur fournit le 3,3 V pour alimenter les extensions. Au labo, nous avons mesuré une consommation de 320 mA avec un affichage brillant, ce qui représente une dissipation d'environ 2,5 W par IC1. Si vous voulez réduire les pertes, remplacez le 7805 par un régulateur à commutation de 5 V de même brochage, comme le type WPMDL1100501S de Würth, dont la dissipation est inférieure à 0,4 W, mais qui fournit une tension un peu plus ondulée (fig. 4). Avec le régulateur de Würth, n'installez pas les condensateurs C2 et C3. L'essai de régulateurs d'autres fabricants a montré que la tension de sortie s'élevait plus lentement, ce qui peut donner des ennuis avec l'ESP32 au moment du démarrage. Pour les éviter, on met un condensateur de 10 µF entre « EN », la broche 23 de l'ESP32, et la masse. L'astuce retarde le lancement jusqu'à la stabilisation de la tension sur 5 V.

Les broches inutilisées de l'ESP32 sont disponibles sur K1 à K5 et les trois tensions d'alimentation sur K11. Voilà des connexions à employer pour vos propres extensions ! Nous avons ainsi fait le tour de tout ce qui concerne le matériel de l'horloge proprement dite.

### Capteurs

Mais il y a encore à découvrir la carte avec les capteurs, basée aussi sur l'ESP32 et connectée par Wi-Fi. On y trouve des capteurs, mais aussi un bouton-poussoir et trois LED. Le schéma complet est à la figure 5. Avec de l'électronique moderne, tout reste très simple.

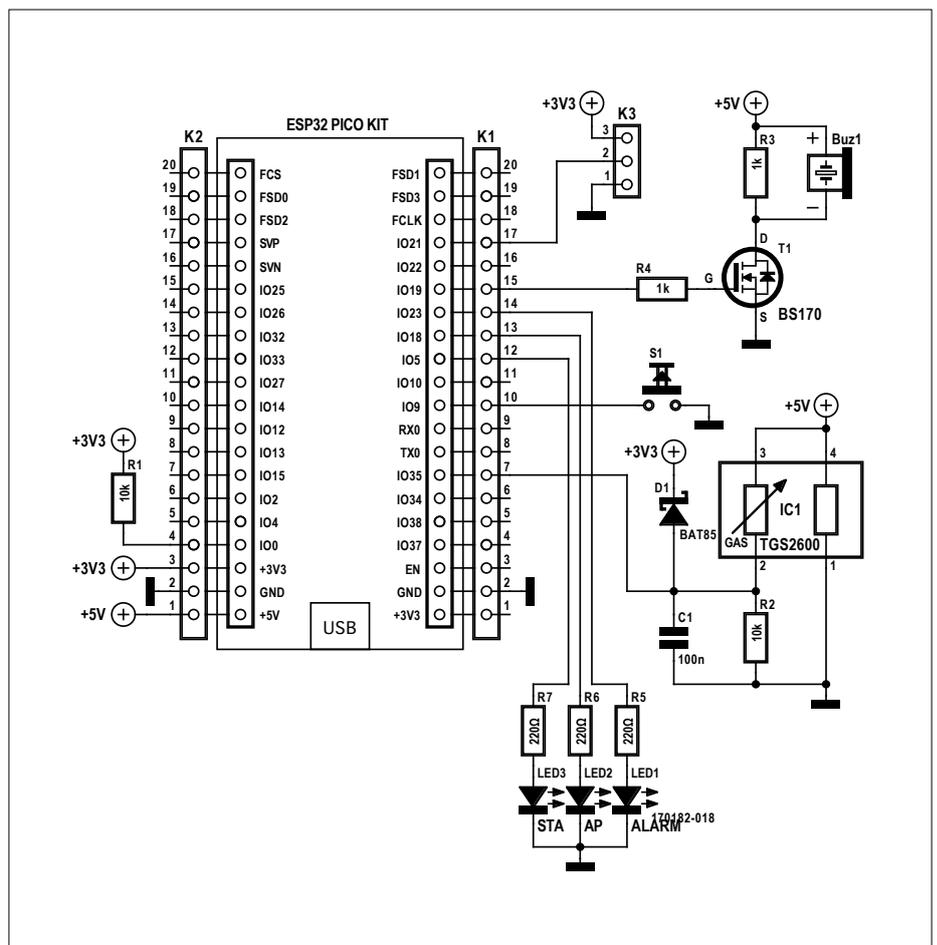


Figure 5. La carte des capteurs porte à la fois l'ESP32, un ronfleur, trois LED, un bouton-poussoir, un capteur de niveau de pollution et un connecteur (K3) pour la thermométrie.

Intéressons-nous au capteur IC1 [3] qui surveille la qualité de l'air. Il est alimenté en 5 V en même temps que son élément de chauffe entre les broches 4 et 1. La diode Schottky D1 sur le 3,3 V protège des surtensions l'entrée analogique IO35 (K1 / broche 7) de l'ESP32. Le ronfleur est actionné par T1. Le connecteur K3 offre encore, entre la masse et le 3,3 V, une entrée numérique par bus 1-wire pour le signal d'un autre capteur. C'est à ce connecteur à 3 broches que se branche le capteur d'humidité et de température DHT22 [4]. En fait, il n'y a pratiquement plus rien à

dire sur ce matériel simple et fiable, si ce n'est peut-être que le boîtier ne doit pas être étanche à l'air, sous peine de réduire à néant l'efficacité de la puce IC1. Il faudra donc que la carte des capteurs soit installée à l'extérieur, mais à l'abri de la pluie sous un petit toit. Le capteur du thermomètre peut quant à lui en rester un peu éloigné, connecté par un câble tri-polaire. Bien entendu, ce boîtier doit rester à portée du réseau sans fil. Comme l'ensemble des capteurs consomme de l'ordre de 100 mA sous 5 V, il lui faut un petit bloc secteur ou une solution plus complexe avec cellules solaires, régula-

Publicité

All inclusive.  
Mag13C Power Modules  
www.we-online.com/powermodules  
Start now with SIP-3!  
WÜRTH ELEKTRONIK

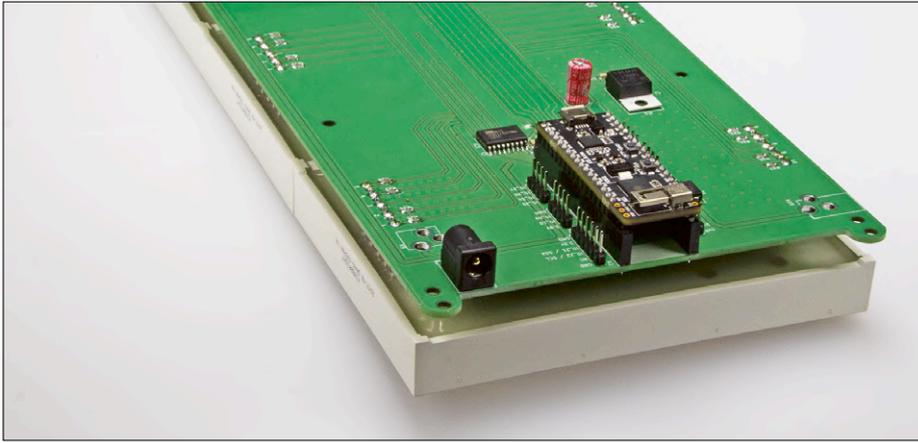


Figure 6. L'électronique de la carte des capteurs sur le prototype terminé. Le connecteur d'alimentation dans le coin inférieur peut être tourné vers le côté ou vers l'arrière de la carte.

teur de charge et accumulateur dont la capacité pour quatre jours de réserve de marche devrait avoisiner 10 Ah. Une dépense sérieuse, mais une liaison par Wi-Fi offre l'avantage d'une exposition nettement moindre aux perturbations radio sur les bandes de 433 ou 866 MHz prisées par des sondes bon marché.

### Construction

Les deux circuits sont d'une telle simplicité que leur fabrication n'offre pas de grande difficulté. Les cartes ESP32 sont enfichables, il y a encore quelques composants à fils, des CMS et des embases à souder. Au dos du grand afficheur, les résistances et condensateurs CMS sont au format 0805 qui reste facile à sou-

der à la main. Les circuits intégrés ont aussi des contacts relativement espacés. En bas de la **figure 6**, à gauche, on peut voir à l'arrière de la carte terminée du prototype une particularité. La prise pour la fiche creuse de l'alimentation 12 V peut pointer vers le côté ou l'arrière de la carte.

La carte des capteurs ne devrait poser aucune difficulté au soudeur peu expérimenté, il n'y a aucun CMS, tous les composants sont traversants, comme les embases mâles et femelles. La **figure 7** montre le prototype du labo d'Elektor sans l'ESP32 de manière à bien reconnaître tout l'agencement du côté composants. Dans la **figure 8**, l'ESP32 et le DHT22 sont à leur place.

### Logiciel et encore plus

Avec tous les projets à microcontrôleur, après avoir installé le matériel, câblé, soudé et mis sous tension, on n'est encore nulle part. L'électronique moderne en veut davantage. Le savoir-faire nécessaire pour les fonctions plus complexes telles que l'intégration au réseau Wi-Fi, l'accès aux serveurs de temps, l'émission et la lecture de données environnementales, c'est dans le logiciel qu'on le trouve : il faut encore le charger dans les processeurs des deux ESP32. Par chance, ils sont compatibles avec l'EDI d'Arduino, ce qui simplifie nettement la programmation. Comment se servir de cet EDI et autres considérations, nous ne les aborderons pas ici, cela irait au-delà du cadre de cet article.

### Affichage

Pour la partie affichage, nous pouvons heureusement recycler un grand morceau de code existant. Il provient de l'article sur les afficheurs de flipper [5] paru dans le numéro précédent du magazine. Il faut d'abord en écarter tout le code relatif à la mécanique du flipper pour ne laisser que celui de l'horloge seule. Puis encore éliminer le code pour l'horloge en temps réel et la connexion par I<sup>2</sup>C dont nous n'avons que faire ici. Mais bien sûr les fonctions MQTT de l'original restent, ainsi que la lecture des données environnementales externes, température et qualité de l'air. On pourra configurer

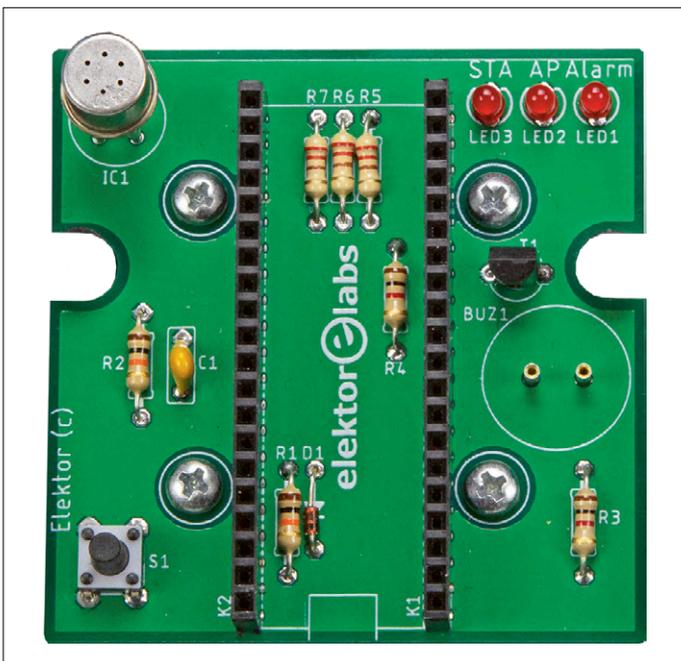


Figure 7. Dépouillée de l'ESP32 et du DHT22, la carte expose les composants sous-jacents.



Figure 8. Le prototype de la carte des capteurs à présent au complet.

l'interface MQTT au moyen de l'interface web de l'horloge.

La configuration standard veut que toutes les dix minutes (par ex. 15h09, 15h19, 15h29, etc.), la mesure de la température actuelle remplace l'affichage horaire, mais seulement dans le cas où la température a varié depuis la dernière fois. Rainer Krugmann utilisait à l'origine la bibliothèque *SevenSeg* pour la commande de l'affichage. Mais ici, la luminosité dépend de la durée de l'itération dans la boucle et donc de ce que le processeur de l'ESP32 doit effectuer. Pour diminuer cet effet, on a confié au module *LEDC* de l'ESP32 la réalisation de la MLI, laquelle est basée sur une horloge à 40 MHz. Rien que cela procure une considérable résolution dans le réglage de la luminosité sur 13 bits avec un taux d'échantillonnage de 5 kHz. On peut sortir cette MLI sur n'importe quelle ligne de GPIO. Chaque segment peut ainsi recevoir son propre signal MLI, tout en changeant de chiffre au rythme de 2,5 ms en multiplexage. Il en résulte une fréquence d'échantillonnage de 100 Hz sans scintillement pour l'affichage complet – avec une MLI à 5 kHz, on ne peut même pas voir un insecte. Une résolution aussi grande en luminosité permet un réglage très fin dans une large plage, il n'y a rien de tel pour une chambre à coucher.

### Luminosité

Vous pouvez facilement régler la luminosité par exemple avec un ordiphone, via l'interface web de l'horloge (fig. 9). Par rapport à l'horloge à flipper, nous avons ici un menu de plus. On pourrait aussi penser à raccorder un capteur de luminosité à l'une des broches disponibles pour effectuer un réglage automatique de l'affichage sur l'ambiance lumineuse environnante, spécialement dans une chambre, sachant que les LED peuvent donner beaucoup de lumière. Avec les 13 bits de résolution, il est encore possible d'ajuster finement le niveau dans un milieu sombre. Voilà de quoi vous donner des idées de logiciel.

### Protocole MQTT

Le reste de l'interface web offre des réglages pour le Wi-Fi et l'heure, tout comme dans l'horloge à flipper. En revanche, les paramètres en MQTT (fig. 10) sont différents. L'horloge attend des messages MQTT formatés en JSON [6] pour acquérir les valeurs de température. Si la nouvelle valeur

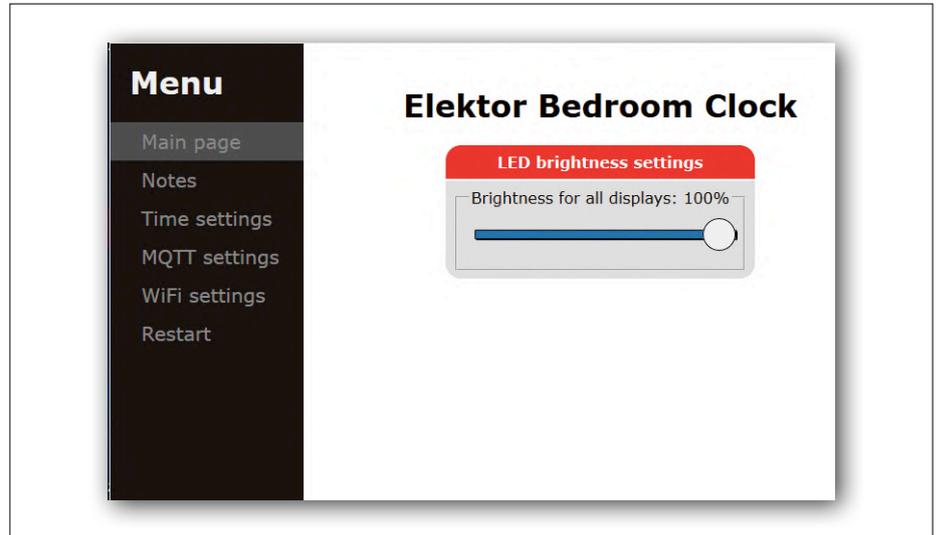


Figure 9. Facile de régler la luminosité de l'horloge avec un ordiphone.

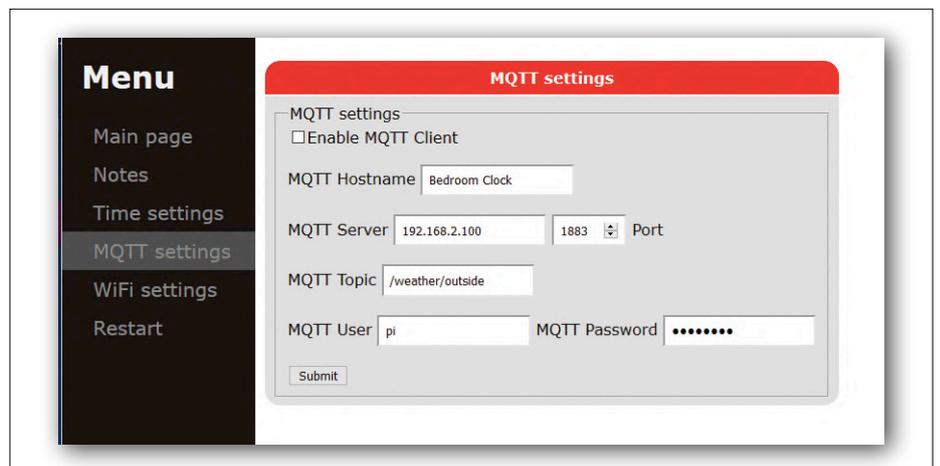


Figure 10. Les paramètres pour le protocole MQTT dans les réglages internet.

de température date de plus de neuf minutes, on l'ignore et on attend un nouveau changement pour l'afficher. Il faut pour cela modifier quelques paramètres : le nom de l'hôte, ici *Bedroom Clock*, l'adresse IP du « broker » (courtier) MQTT ainsi que son numéro de port (1883 par défaut), sans oublier le sujet attendu, ici « /weather/outside » pour la température du capteur extérieur. Enfin, il y a encore le nom d'utilisateur (*User*) et le mot de passe du serveur utilisé. Si vous utilisez un Raspberry Pi (RPI) comme serveur MQTT, c'est alors là que s'effectueront les entrées. Pour tout changement de paramètre, il faut de nouveau saisir en entier le nom d'utilisateur et le mot de passe.

### Démarrage

Une fois le micrologiciel nécessaire com-

pilé, transféré dans la carte ESP32 et les paramètres voulus réglés, on peut brancher l'adaptateur secteur sur l'horloge qui va ainsi lancer le programme et directement afficher *STA* ou *AP* selon qu'elle est configurée comme station Wi-Fi ou comme point d'accès. Tout de suite après, l'adresse IP de l'horloge apparaît de manière à pouvoir la configurer par PC ou ordiphone.

### Source des données MQTT

Une horloge capable de recevoir des données météo et de les afficher, c'est bien, encore faut-il les produire, ces données. La carte à capteurs qui communique par MQTT est destinée à cela. Le labo possédait déjà le logiciel pour la « surveillance du niveau de pollution de l'air », projet paru dans le numéro de mars/avril 2019 [5]. Il a été adapté

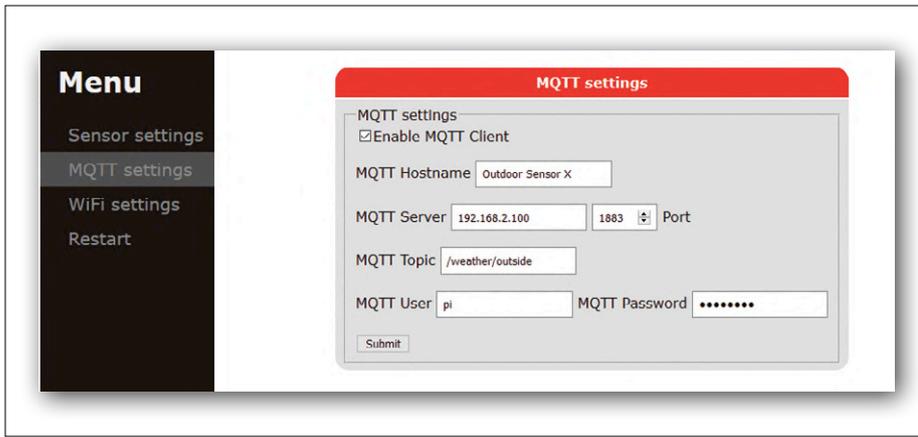


Figure 11. Réglages MQTT dans l'interface de la carte des capteurs.

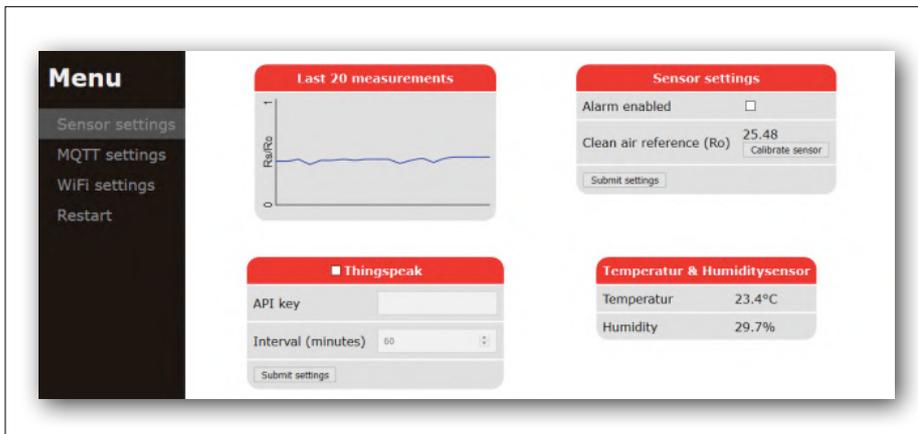


Figure 12. Paramétrage dans l'interface internet de la carte des capteurs.

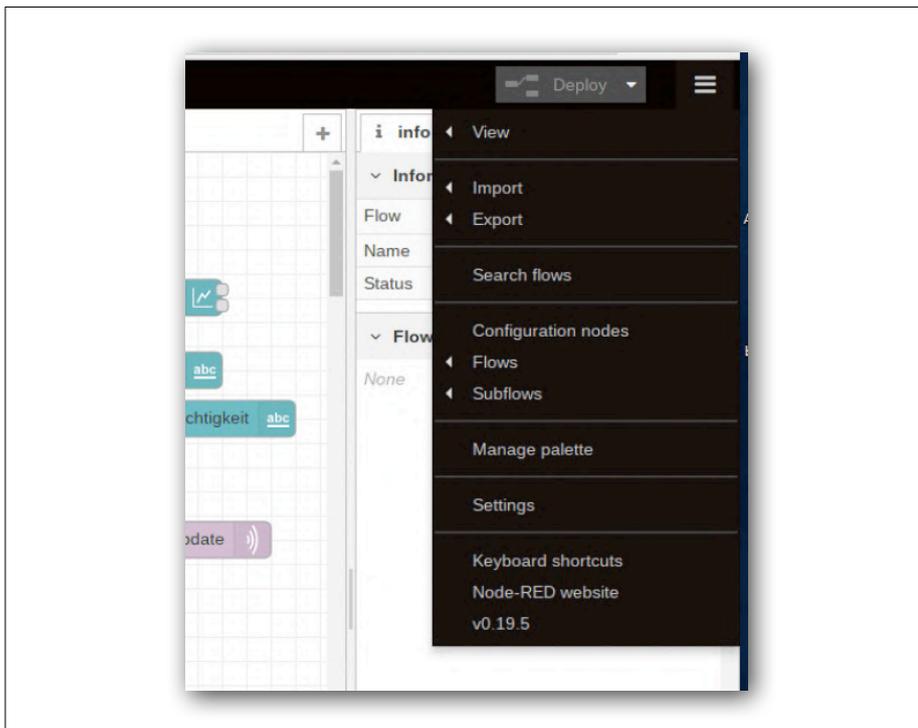


Figure 13. La page web de Node-RED sur le Raspberry Pi.

pour envoyer les données mesurées à un courtier MQTT et fournir en permanence les paramètres web de la carte des capteurs. Pour la communication MQTT, il faut réserver son propre *thread* (fil) qui fonctionne sur l'un des noyaux du processeur de l'ESP32, tout comme on l'a fait pour la carte d'affichage. De cette façon, les autres composants du logiciel s'exécutent isolément, sans gêne mutuelle. Dans l'interface web de la **figure 11**, on peut bien sûr modifier les paramètres MQTT, mais il faut se rappeler que dans une telle interface web, tous les échanges de paramètres entre le site et l'appareil sont transmis en clair. Le nom d'hôte est ici « Outdoor Sensor X », le reste est pareil à ce qu'il y a sur l'horloge.

Si vous n'avez pas installé le capteur DHT22, vous n'aurez rien d'autre que les données de qualité de l'air de IC1. Dans le menu des capteurs de la **figure 12**, on voit les valeurs mesurées actuellement. L'ensemble des mesures est envoyé toutes les 30 s au courtier MQTT. Il va de soi que vous pouvez ici aussi intervenir dans le logiciel pour y implémenter vos propres idées. À l'avenir, vous pourrez trouver sur le site d'Elektor d'autres extensions pour acquérir des valeurs de mesure compatibles MQTT.

### Raspberry Pi comme courtier MQTT

Dans cette formule-ci, les données des capteurs ne sont pas transmises par radio pour être vues sur l'affichage de l'horloge, mais bien stockées de manière standardisée sur un serveur d'où l'horloge les prélèvera au fur et à mesure. Il y faut donc un serveur. Si vous êtes familiarisé à cette technique de serveur, c'est un avantage. Mais on peut faire plus simple avec un petit serveur installé sur un Raspberry Pi bon marché qui, pour la cause, ne sera pas débordé de travail. Pas besoin du dernier modèle le plus puissant.

On fait tourner sur ce RPi un courtier MQTT avec Node-RED pour le traitement des données. Pour ce faire, mieux vaut commencer avec une installation native de Raspbian sur une carte SD et y ajouter les composants nécessaires.

### Mosquitto

C'est Mosquitto qui a été choisi comme courtier MQTT. Avec un RPi complètement configuré pour votre réseau local, vous aurez besoin d'un terminal pour introduire d'autres commandes. Effectuez

d'abord une mise à jour vers les nouvelles versions du logiciel. On l'exécute dans un terminal avec les commandes :

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get dist-upgrade
```

Pour l'installation de Mosquitto, on saisit dans le terminal :

```
sudo apt-get install mosquitto
mosquitto-clients
```

Finalement, on configure Mosquitto comme application à démarrage automatique :

```
sudo systemctl enable mosquitto
```

### Node-RED

Continuons maintenant avec le langage de programmation graphique Node-RED. Pour cela, suivez les instructions sur [8]. Il faut encore installer le tableau de bord (*dashboard*) de Node-RED. Dans la ligne d'adresse d'un navigateur internet, on saisit le texte « `http://[adresse IP du RPi]:1880` » et la page Node-RED s'affiche (**fig. 13**). Cliquez sur l'entrée du menu *Manage palette*, où vous pouvez voir un aperçu des « add-ins » déjà installés et de ceux à installer (**fig. 14**). On choisit alors `node-red-dashboard` et on l'installe aussi. On peut finalement importer le code du traitement des données en choisissant dans le menu (**fig. 13**) l'entrée *Import* puis le presse-papier comme source. Pour cela, vous devez bien sûr mettre du code dans le presse-papier, donc ouvrir le fichier ASCII `Listing.txt` dans un éditeur de texte à votre goût, en copier le contenu et l'envoyer dans

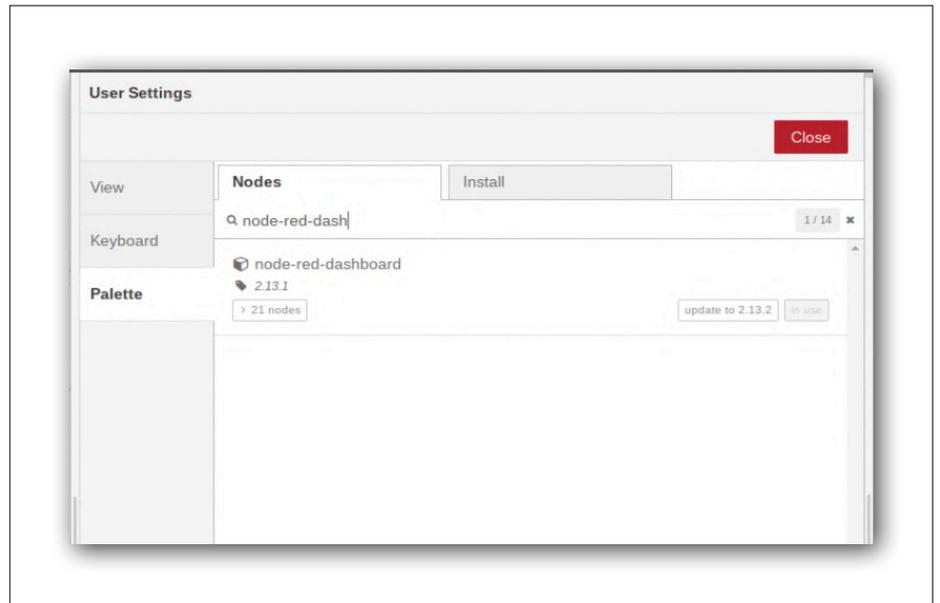


Figure 14. Importation du tableau de bord de Node-RED.

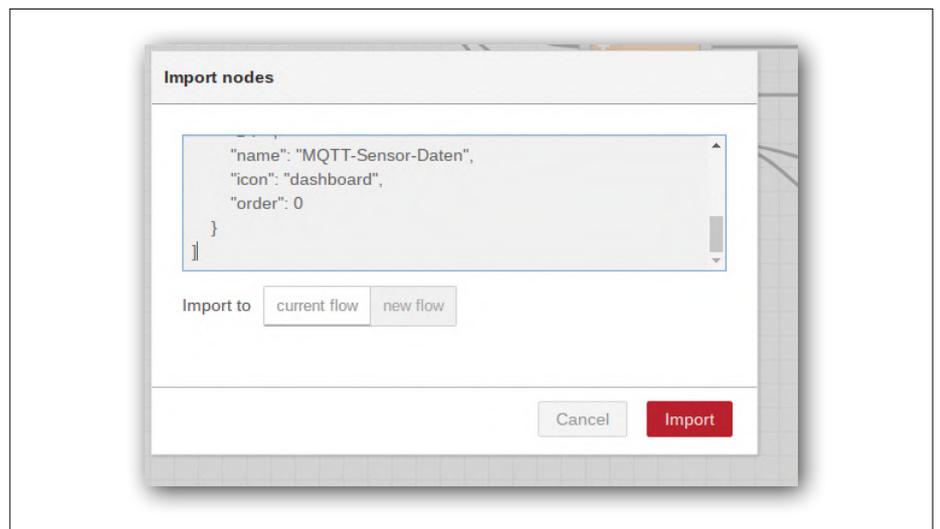


Figure 15. La fenêtre d'importation après copie du code du fichier `Listing.txt`.

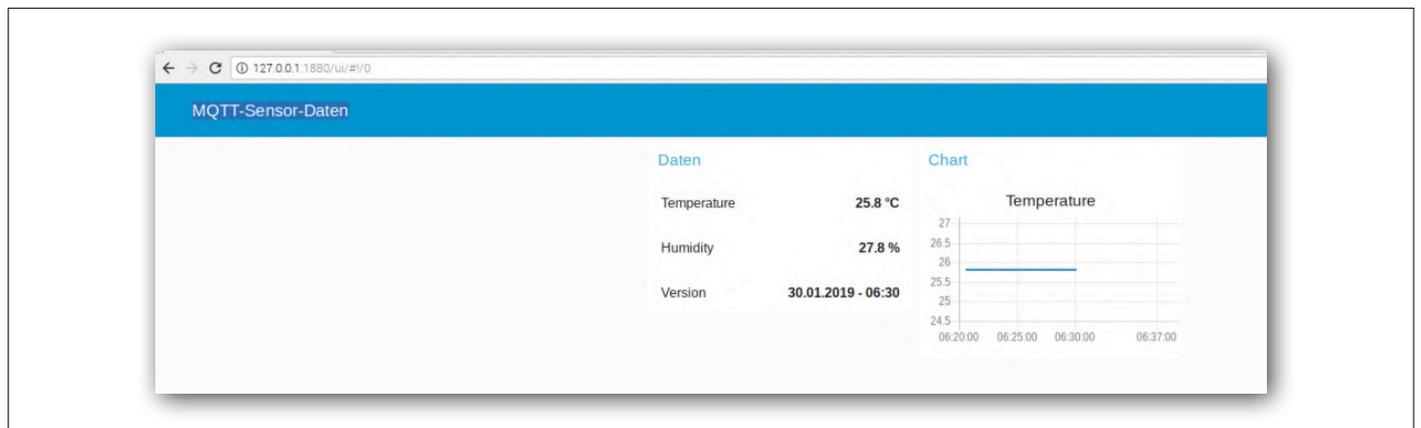


Figure 16. Interface utilisateur du RPi avec les données déjà reçues.



la fenêtre de la **figure 15**. Le fichier lui-même est disponible dans le téléchargement pour ce projet sur la page correspondante [9] du site d'Elektor, tout

Figure 17. Prototype préliminaire d'une variante plus conventionnelle de l'horloge à LED avec des afficheurs à sept segments plus petits.

comme les fichiers du code source pour l'horloge et la carte des capteurs. Après le clic sur **Import**, si tout s'est bien déroulé, le RPi sera disponible comme serveur MQTT. L'URL qui suit : « `http://[adresse IP du RPi]:1880/ui/` » vous donne accès à l'interface utilisateur et vous pourriez déjà voir des valeurs comme à la **figure 16**. Maintenant, le



## LISTE DES COMPOSANTS DE L'HORLOGE

### Résistances :

(CMS 0805, ¼ W)

R1 = 10 kΩ

R5 à R11, R13 à R19, R21 à R27, R29 à R35 = 43 Ω \*

R12, R20, R28, R36 = 680 Ω \*

### Condensateurs :

(sauf indication contraire CMS 0805)

C1 = 100 µF / 25 V, électrolytique vertical, au pas de 2,5 mm

C2 = 330 nF / 50 V, X7R \*

C3, C6, C7 = 100 nF / 50 V, X7R \*

C4, C5 = 1 µF / 50 V, X7R

### Semi-conducteurs :

D1 = MBR540

LD1 à LD4 = SA40-19SRWA, Kingbright, p.ex.

Reichelt SA40-19RT \*

IC1 = 7805, régulateur 5 V \*

IC2 = ULN2803A, SOIC18-W

IC3 = MIC2981, SOIC18-W

ESP32 Pico kit, Espressif, cf. [1]

### Divers :

Prise encartable pour fiche creuse, verticale

ou horizontale (Würth 694108402002 ou 694108301002) \*

2× embase SIL à 20 contacts pour ESP32, au pas de 2,54 mm

K1 = embase à 2 picots au pas de 2,54 mm

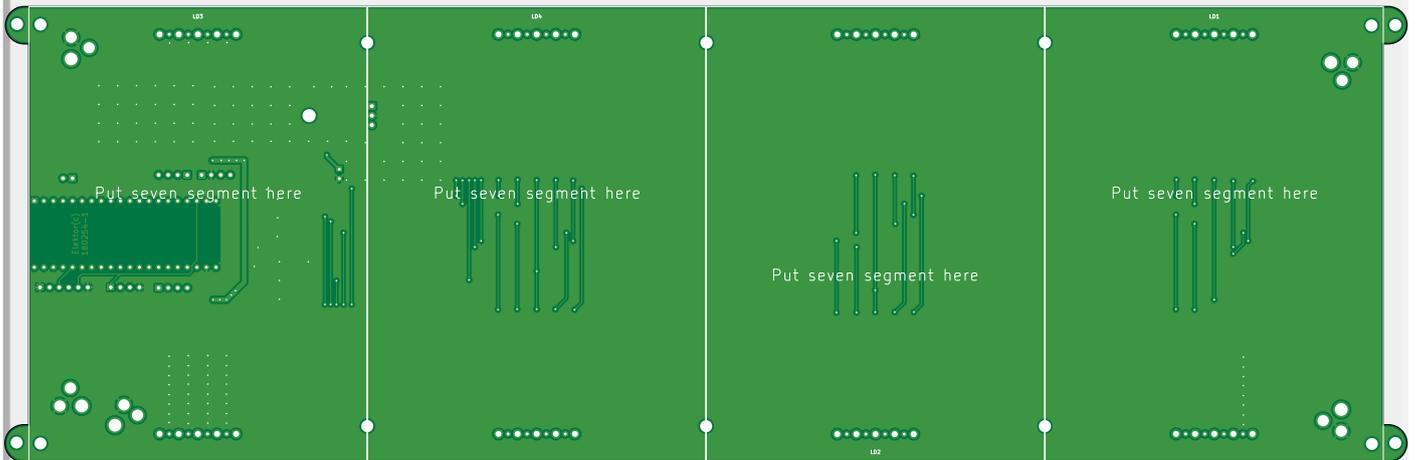
K2 = embase SIL à 6 picots au pas de 2,54 mm

K3 à K5, K11 = embase SIL à 4 picots

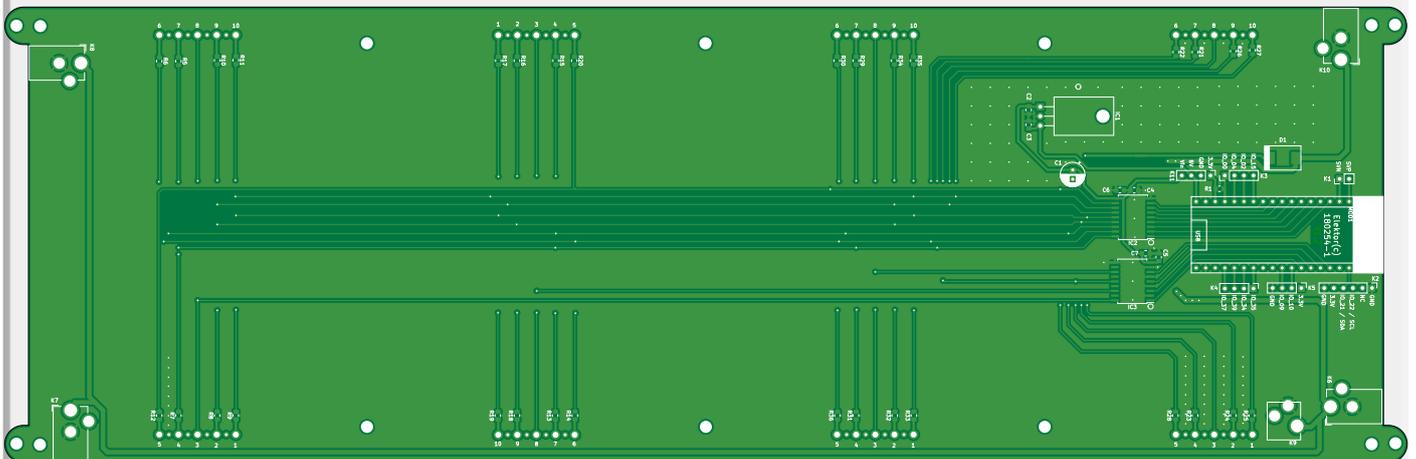
au pas de 2,54 mm

circuit imprimé réf. 180254-1 v1.2

\* = cf. texte



50% de la taille réelle





## LISTE DES COMPOSANTS DE LA CARTE DES CAPTEURS

### Résistances :

(à fils, ¼ W)  
 R1, R2 = 10 kΩ  
 R3, R4 = 1 kΩ  
 R5 à R7 = 220 Ω

### Condensateurs :

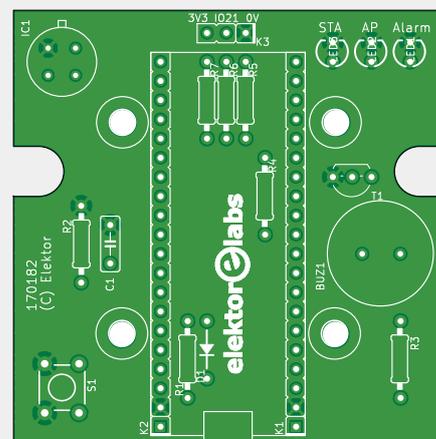
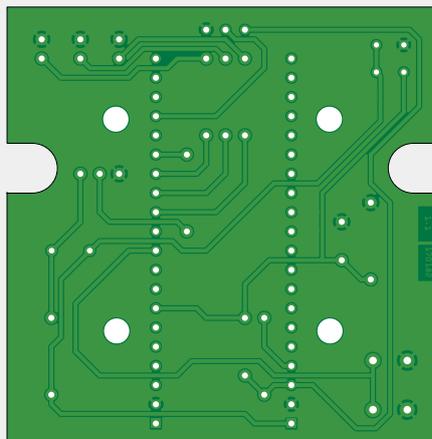
C1 = 100 nF / 50 V, céramique, vertical,  
 au pas de 5 mm

### Semi-conducteurs :

LED1 à LED3 = LED 3 mm, rouge  
 D1 = BAT85  
 T1 = BS170  
 IC1 = TGS2600, Figaro  
 ESP32 Pico Kit, Espressif, cf. [1]

### Divers :

Module de capteurs de température  
 et d'humidité DHT22 \*  
 Buz1 = ronfleur piézo p.ex. PS1420P02CT, TDK  
 K1, K2 = embase SIL à 20 contacts pour  
 ESP32, au pas de 2,54 mm



K3 = embase à 3 picots soudés  
 au pas de 2,54 mm

S1 = bouton-poussoir CMS, p.ex. S-TACTILE-  
 6X6-PTH, Alps

Circuit imprimé réf. 170182-1 v1.1

\* = cf. texte

trio composé de l'horloge, de la carte des capteurs et d'un RPi comme serveur MQTT collabore.

### Le mot de la fin

Nous pouvons conclure que cette réalisation constitue une bonne base de départ pour d'autres projets. Elle démontre la possibilité de mettre en œuvre, à peu de frais en matériel et avec un minimum de logiciel, une transmission de données stable dans un réseau Wi-Fi en passant par un serveur. L'utilisation de l'afficheur à quatre chiffres ne se limite certainement pas à donner l'heure. Elektor proposera d'autres capteurs qui envoient leurs données avec le protocole MQTT. Si l'idée vous a plu, mais que vous n'aimez pas les afficheurs géants, nous avons encore autre chose en préparation. La **figure 17** montre un prototype à chiffres à sept segments aussi, qui jouit des mêmes fonctions, mais se contente d'un affichage de grandeur plus normale. Le circuit imprimé adéquat ne sera pas encore disponible lors de la parution de cet article du fait que, généralement, il faut compter au moins trois révisions avant de sortir une carte. Elle vous sera livrée plus tard, c'est promis ! ◀

(180254-04 - version française : Robert Grignard)

### Liens

- [1] Pico-Kit ESP32 : [www.elektor.fr/esp32-pico-kit-v4](http://www.elektor.fr/esp32-pico-kit-v4)
- [2] Protocole MQTT : <https://fr.wikipedia.org/wiki/MQTT>
- [3] TGS2600 : [www.figarosensor.com/products/entry/tgs2600.html](http://www.figarosensor.com/products/entry/tgs2600.html)
- [4] DHT22 : <https://bit.ly/1TZDFPz>
- [5] « horloge rétro à afficheurs de flipper », Elektor 03-04/2019 : [www.elektormagazine.fr/180307-04](http://www.elektormagazine.fr/180307-04)
- [6] JSON : [https://fr.wikipedia.org/wiki/JavaScript\\_Object\\_Notation](https://fr.wikipedia.org/wiki/JavaScript_Object_Notation)
- [7] Eclipse Mosquitto: <https://mosquitto.org>
- [8] NODE-RED: <https://nodered.org/docs/hardware/raspberrypi>
- [9] Page de l'article : [www.elektormagazine.fr/180254-04](http://www.elektormagazine.fr/180254-04)



@ WWW.ELEKTOR.FR

→ Circuit imprimé de l'horloge (180254-1 v1.2)  
[www.elektor.fr/18851](http://www.elektor.fr/18851)

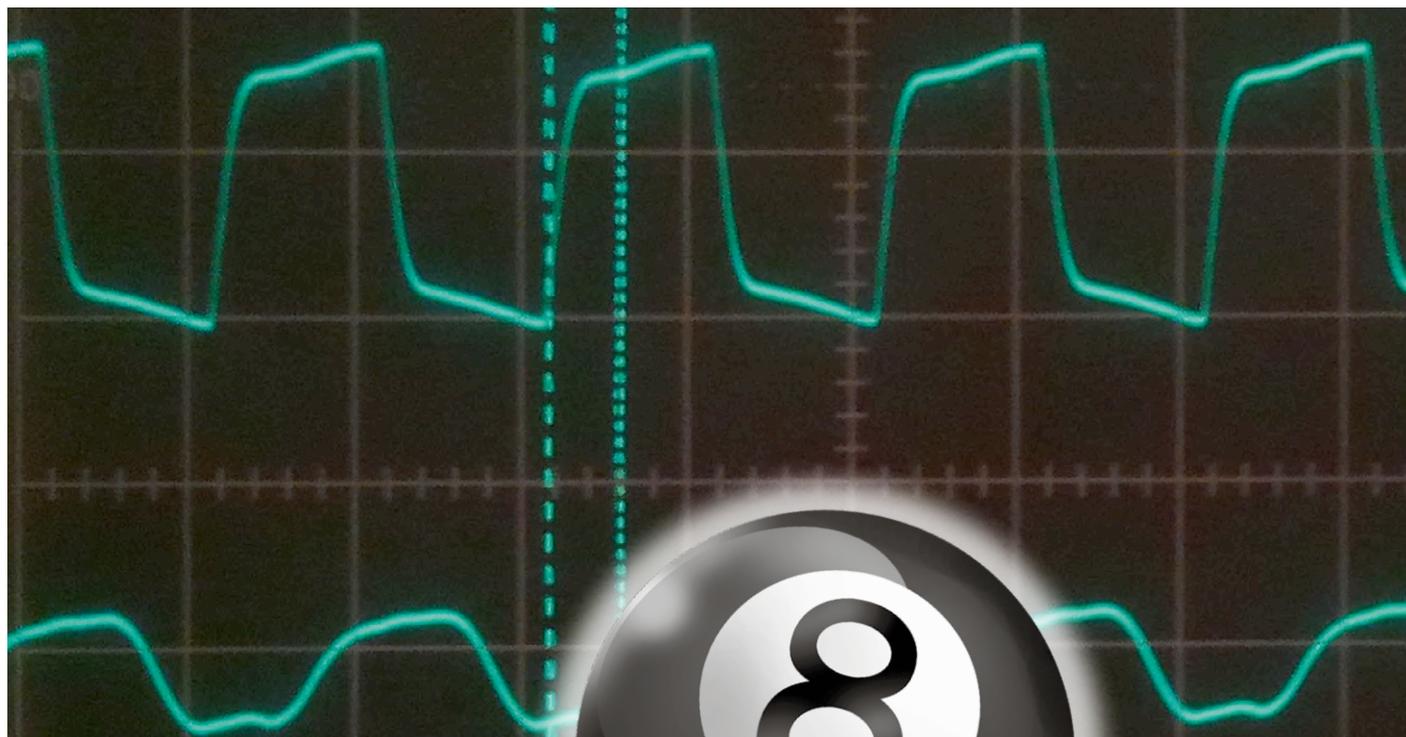
→ Circuit imprimé des capteurs (170182-1 v1.1)  
[www.elektor.fr/18695](http://www.elektor.fr/18695)

→ ESP32 Pico-Kit V4  
[www.elektor.fr/esp32-pico-kit-v4](http://www.elektor.fr/esp32-pico-kit-v4)

→ Raspberry Pi 3 (modèle A+) avec compilation 2 RPi offerte  
[www.elektor.fr/raspberry-pi-3-model-a-plus](http://www.elektor.fr/raspberry-pi-3-model-a-plus)

# du nouveau dans la famille des processeurs à 8 bits

flash ultra-endurance  
et cellules logiques dans les contrôleurs PIC



Tam Hanna

Microchip a doté ses derniers « enfants » de caractéristiques intéressantes, inconnues des développeurs habitués à d'autres PIC. Cet article se propose d'en présenter brièvement deux : la mémoire flash ultra-endurance qui est une partie robuste de la mémoire flash à laquelle le contrôleur a accès en lecture et en écriture ; les cellules logiques flexibles qui permettent de configurer une partie matérielle du contrôleur, ce qui peut économiser beaucoup de code et de temps de calcul.

En ces temps de Raspberry Pi et de microcontrôleurs bon marché, on pourrait se demander ce qui justifie encore l'existence de processeurs à 8 bits classiques.

Il y a quelques semaines, l'auteur était face à la situation suivante : « mettre en sommeil » pour une certaine durée un ordinateur basé sur une carte Orange Pi, temps pendant lequel celui-ci devait consommer le moins d'énergie possible. L'expérience a montré qu'arrêter le cœur des processeurs ARM n'est pas la panacée, l'auteur a donc décidé de confier la gestion de la consommation à un autre microcontrôleur.

Sa tâche consistait à surveiller en permanence l'état du système entier et à le réveiller à la demande. Pour cela, il a choisi une

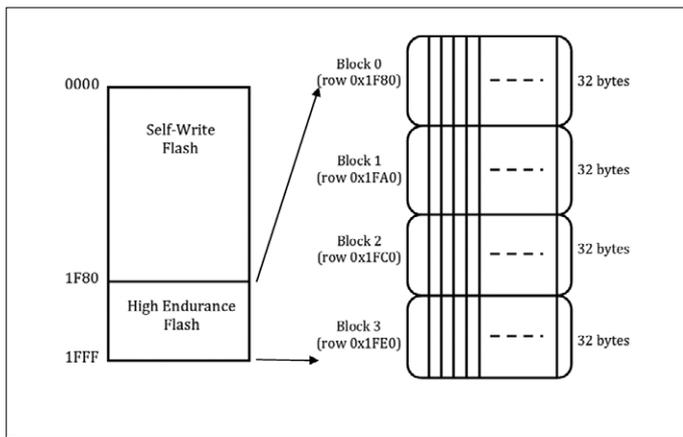


Figure 1. Le domaine de mémoire HEF se situe à l'extrémité supérieure de la mémoire du programme (source : Microchip).

**PIC12(L)F1501/PIC16(F)L150x Family Types**

Device	Data Sheet Index	Program Memory Flash (words)	Data SRAM (bytes)	IOs <sup>(1)</sup>	10-bit ADC (ch)	Comparators	DAC	Timers (8/16-bit)	PWM	EUSART	MSSP	CMOS	CLC	NCO	ICD	XLP
PIC12(L)F1501	(1)	1024	64	6	4	1	1	2/1	4	—	—	1	2	1	—	—
PIC16(L)F1503	(2)	2048	128	12	8	2	1	2/1	4	—	1	1	2	1	—	—
PIC16(L)F1507	(3)	2048	128	18	12	—	—	2/1	4	—	—	1	2	1	—	—
PIC16(L)F1508	(4)	4096	256	18	12	2	1	2/1	4	1	1	1	4	1	Y	Y
PIC16(L)F1509	(4)	8192	512	18	12	2	1	2/1	4	1	1	1	4	1	Y	Y

Figure 2. Grâce à sa logique interne de débogage (ICD), l'acheteur d'un 16F1509 économise les 50 € environ du prix d'achat d'une carte d'extension (source : Microchip).

puce de la société Microchip. La série PIC16F15xx est idéale pour cela dans la mesure où son prix reste attractif même en petites quantités.

**Mémoire flash intelligente**

Les microcontrôleurs PIC classiques sont en principe équipés d'une mémoire flash réinscriptible 10.000 fois environ. Elle sert à l'enregistrement des instructions à 14 bits du programme. Les modèles classiques comme le 16F84 possèdent en outre une petite mémoire EEPROM où sont enregistrées des informations de configuration produites au cours de l'exécution du programme.

Pour la nouvelle famille PIC16F15xx, le fabricant a choisi une approche différente, sans doute pour des raisons de coût, en

remplaçant la mémoire non volatile par une sorte de contrôleur « automodifiable ». Les PIC ont la capacité de réécrire, en cours de fonctionnement, la totalité de leur mémoire de programme. Comme la durée de vie de cette mémoire ne dépasse pas 10.000 cycles d'écriture environ, on a ajouté en haut de la mémoire, une cellule supplémentaire, qualifiée de flash ultra-endurante (*High Endurance Flash*, HEF). Lorsque vous examinerez la représentation schématique de la **figure 1**, gardez à l'esprit que seuls huit des quatorze bits de chaque mot de programme sont implémentés en utilisant des cellules HEF. Les essais auront lieu dans l'environnement MPLAB X. L'environnement de développement de l'auteur est basé sur Ubuntu Linux, mais le logiciel se comporte de manière analogue sous Windows et Mac. Comme le projet de l'auteur utilise un 16F1503, le débogage s'effectue au moyen d'une carte d'extension de type AC244051/AC244052, connectée à un appareil de commande de type ICD3. Remarquez que la logique de débogage ne fait pas défaut sur tous les membres de la famille. Les deux dernières colonnes du tableau de la **figure 2** indiquent les puces qui n'ont pas besoin d'un « adjuvant ».

**Pratique de la HEF**

On peut de servir de la mémoire flash HEF de deux manières différentes : on peut utiliser le configurateur de code (MCC) intégré à MPLAB, qui produit à la demande du code inscriptible aussi bien dans la mémoire flash normale que dans la mémoire HEF (voir l'**encadré**). Mais surtout, pour les petits contrôleurs, il y a une bibliothèque conçue par Lucio di Jasio, publiée sous forme de note d'application, à laquelle nous aurons recours par la suite.

Pour commencer, créez comme d'habitude un nouveau projet sous MPLAB X ; dans la suite, nous admettrons, déjà pour des raisons de place, que cet environnement vous est familier. Comme compilateur, nous utiliserons la version gratuite de XC8 ; les exemples sont tellement simples que les options d'optimisation avancées de la version professionnelle ne sont pas nécessaires.

Ensuite, ouvrez le MCC et définissez un projet sans options particulières. Prenez garde à NE PAS inclure le périphérique MEMORY de la **figure 3** : le code produit n'est que partiellement compatible avec la HEF.

Suivez le lien [1] afin de télécharger le code source associé à la note d'application AN1673. Extrayez l'archive dans un répertoire facilement accessible sur votre station de travail. Ne soyez pas surpris de voir apparaître deux groupes de fichiers. *Flash.\** sert à l'accès à la mémoire, tandis que *HEFlash.\** fournit une logique de niveau supérieur. Le « L » majuscule dans les noms de fichier n'est pas une faute de frappe d'Elektor !

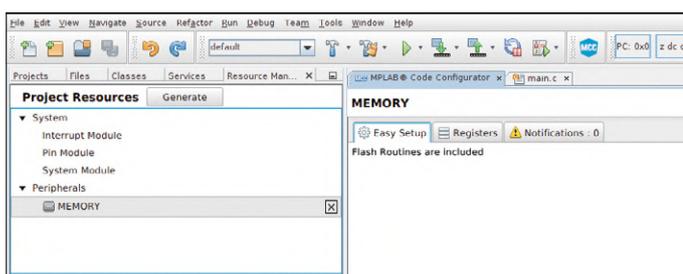


Figure 3. Le périphérique mémoire n'a pas sa place dans notre solution.

Nous allons maintenant extraire du code téléchargé les parties dont nous avons besoin pour notre projet. Ce qui nous manque plus particulièrement, ce sont les constantes qui définissent le début et la fin de l'espace mémoire représenté sur la figure 1. On les trouve dans le fichier *Flash.h*. Il faut récupérer le bloc de déclarations en entier. Il a l'aspect suivant (fortement abrégé ici) :

```
#if defined( __16F1501)
//1K
#define FLASH_ROW_SIZE 16 // size of a row
#define HEFLASH_START 0x0380
```

```

// first address in HE Flash memory
#define HEFLASH_END 0x03FF
// last address in HE Flash memory
#elif defined( __16F1503) || defined( __16F1507) ||
    defined( __16F1512) || \
defined( __16F1703) || defined( __16F1707)
//2K
. . .
#endif

```

La façon dont vous incorporez les constantes dans votre projet est à votre convenance. L'auteur a coutume de créer un fichier appelé *Globals.h* qui lui sert à abriter ce genre de paramètres. Les deux fichiers *HEFlash.c* et *HEFlash.h* sont importés tels quels dans la solution. Créez simplement de nouveaux fichiers avec ces noms et remplissez-les avec les contenus des anciens fichiers par copier-coller. Au lieu de *Flash.h*, incluez votre fichier d'en-tête global.

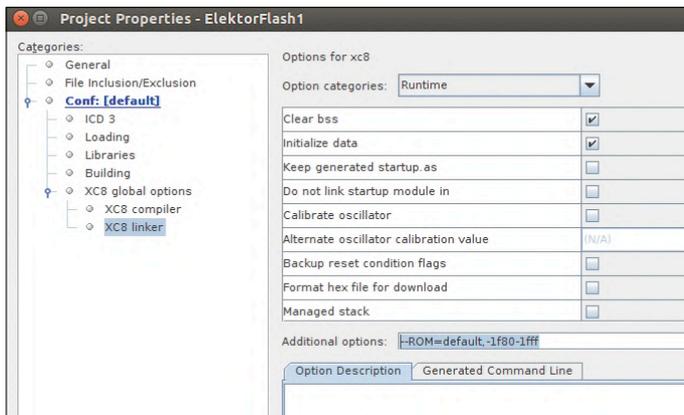


Figure 4. Attention : les valeurs hexadécimales diffèrent d'une puce à l'autre.

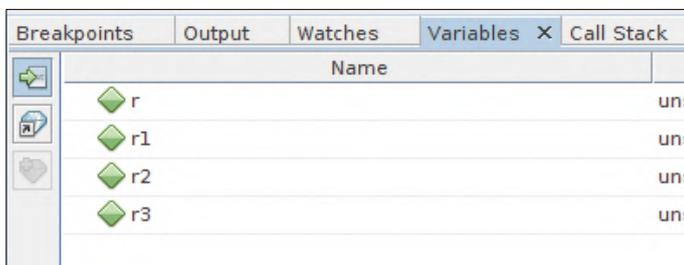


Figure 5. Cette option entraînera le chaos.

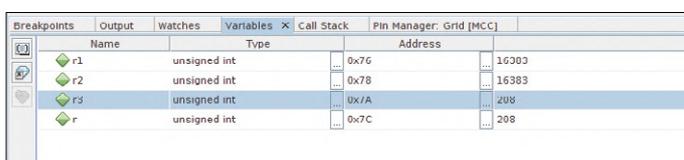


Figure 6. Il y a quelque chose de pourri, là !

En ce point, il convient de se ranger à la procédure recommandée par Diomidis Spinellis dans son livre « Code Reading » : un programme porté partiellement est supposé fonctionner, le compilateur menant progressivement au but par ses messages d'erreur. Dans le cas de notre exemple, les méthodes *FLASH\_erase*, *FLASH\_read* et *FLASH\_write* sont sollicitées, le reste du fichier *Flash.c* n'est pas indispensable et n'a nul besoin d'être inclus dans notre projet.

Pour finir, il nous faut une fonction qui exécute les opérations nécessaires au déverrouillage de la mémoire par une séquence unique d'instructions en assembleur, que nous reproduisons ici, essentiellement à cause de l'usage de la macro *#asm* :

```

void _unlock( void)
{
#asm
BANKSEL PMCON2
MOVLW 0x55
MOVWF PMCON2 & 0x7F
MOVLW 0xAA
MOVWF PMCON2 & 0x7F
BSF PMCON1 & 0x7F,1 ; set WR bit
NOP
NOP
#endasm
} // unlock

```

Avant de pouvoir nous attaquer à la programmation de la mémoire flash, il y a encore une opération préparatoire. En raison de la structure présentée sur la figure 1, il est possible que le code produit par MPLAB vienne empiéter sur l'espace de notre programme. Pour résoudre ce problème, il faut indiquer à MPLAB que l'accès à une partie de la mémoire est verrouillé. La **figure 4** illustre la procédure.

Notre tâche suivante est de commencer à remplir la mémoire avec des données. Pour cela, il faut le code suivant, qui devrait théoriquement écrire une séquence de nombres dans la mémoire et la relire :

```

void main(void)
{
// initialisation du système
SYSTEM_Initialize();
unsigned r, r1, r2, r3;
FLASH_erase(HEFLASH_START);
// Mettre un point d'arrêt ici
FLASH_write(HEFLASH_START, 64, 0);
r = FLASH_read( HEFLASH_START);
FLASH_erase(HEFLASH_START+1);
FLASH_write(HEFLASH_START+1, 128, 0);
r = FLASH_read( HEFLASH_START+1);
FLASH_erase(HEFLASH_START+2);
FLASH_write(HEFLASH_START+2, 208, 0);
r = FLASH_read( HEFLASH_START+2);
r1 = FLASH_read( HEFLASH_START);
r2 = FLASH_read( HEFLASH_START+1);

```

```

r3 = FLASH_read( HEFLASH_START+2);

while (1)
{
// Insérer le code de votre application ici
}
}

```

D'un point de vue technique, ce code ne devrait guère poser de problème. Veillez simplement à ce que le dernier argument de `FLASH_write` soit toujours 0 pour que le microcontrôleur procède effectivement à une opération d'écriture. Comme toujours, le projet est disponible sur la page web de cet article [2].

### Les aléas du débogage

Le moteur de débogage du 16F1509 ne peut traiter qu'un seul point d'arrêt à la fois. Après que MPLAB a arrêté le programme après le premier cycle d'effacement, on fait un clic droit sur le point d'arrêt et on le désactive avec *Breakpoint Enabled*. Ouvrez ensuite l'onglet *Variables* au bas de l'EDI et utilisez les trois boutons apparus en colonne pour désactiver les fonctions de confort inutiles (**fig. 5**).

Vous pouvez ensuite parcourir ce code pas à pas en appuyant plusieurs fois sur *Step Over*. Avant l'exécution de la deuxième instruction de lecture dans r3, les variables numériques individuelles se présentent comme illustrées sur la **figure 6**. On remarquera que les instructions de lecture effectuées précédemment ont toujours affiché la valeur correcte dans r.

Lors du test des programmes enregistrés dans la mémoire Flash, il est particulièrement important d'attendre que toutes les opérations d'écriture soient terminées avant d'analyser les résultats. En pratique, le cas revient toujours où une fonction d'écriture finit par annuler l'opération qu'elle avait commencée. Dans notre cas, c'est exactement ce qui se produit, car l'effacement des cellules flash affecte également les éléments voisins.

### Un petit piège

La situation se complique dans la mesure où le PIC16F1503 et le PIC16F1509 diffèrent non seulement par la taille de la mémoire de programme mais aussi par celle des cellules. On en a un exemple avec le code suivant, qui détruit la valeur de r1 :

```

FLASH_erase(HEFLASH_START);
FLASH_write(HEFLASH_START, 64, 0);
FLASH_erase(HEFLASH_START+1*FLASH_ROW_SIZE );
FLASH_write(HEFLASH_START+1*FLASH_ROW_SIZE , 128, 0);
FLASH_erase(HEFLASH_START+2*FLASH_ROW_SIZE );
FLASH_write(HEFLASH_START+2*FLASH_ROW_SIZE , 208, 0);
r1 = FLASH_read( HEFLASH_START);
r2 = FLASH_read( HEFLASH_START+1*FLASH_ROW_SIZE );
r3 = FLASH_read( HEFLASH_START+2*FLASH_ROW_SIZE );

```

Si l'on recherche manuellement dans *Flash.h* la valeur correcte de `FLASH_ROW_SIZE` et qu'on l'applique, on obtient le résultat de la **figure 7**. Voici une façon de le réaliser par programme :

```

FLASH_erase(HEFLASH_START); // Point d'arrêt ici
FLASH_write(HEFLASH_START, 64, 0);
FLASH_erase(HEFLASH_START+1*32 );

```

Name	Type	Address	Value
r1	unsigned int	0x76	64
r2	unsigned int	0x78	128
r3	unsigned int	0x7A	208
r	unsigned int	0x7C	208

Figure 7. Tout est maintenant bien en ordre.

```

FLASH_write(HEFLASH_START+1*32, 128, 0);
FLASH_erase(HEFLASH_START+2*32 );
FLASH_write(HEFLASH_START+2*32, 208, 0);
r1 = FLASH_read( HEFLASH_START);
r2 = FLASH_read( HEFLASH_START+1*32);
r3 = FLASH_read( HEFLASH_START+2*32);

```

En pratique, cela signifie deux choses pour les développeurs : premièrement, lors d'un changement de modèle de microcontrôleur, il faut être particulièrement soigneux dans la modification de la configuration dans le fichier d'en-tête. Deuxièmement, il est recommandé de toujours écrire dans la mémoire flash cellule par cellule ; si nécessaire, les valeurs existantes doivent d'abord être sauvegardées dans un tampon.

Les trois routines HEFlash contournent partiellement ce problème en obligeant l'utilisateur à écrire et à lire des blocs entiers. Sachez cependant que ce problème d'incompatibilité de la taille de bloc entre différents contrôleurs, dont il vient d'être question, se produit également dans votre cas.

### FPGA ou MCU ?

Depuis quelque temps – la tendance a commencé vers 2014, il y a de plus en plus de microcontrôleurs qui embarquent un petit FPGA ou un petit CPLD. L'idée sous-jacente est très simple : le développeur peut déplacer de la circuiterie fréquente dans une unité matérielle dédiée (*Configurable Logic Cell, CLC*). Tout d'abord, cela simplifie le programme d'application, car il n'a plus besoin de passer du temps à la scrutation, etc. Ensuite, les CLC peuvent substantiellement réduire la taille de l'ensemble : avec une conception intelligente, la CLC de certains contrôleurs peut éliminer un ou plusieurs groupes de composants externes, réduisant ainsi les coûts globaux.

Microchip propose depuis quelque temps plusieurs versions de CLC et, par son rachat d'Atmel, se retrouve en possession d'une version supplémentaire. Quoiqu'il en soit, la CLC du PIC utilisé ici se compose de quatre segments, comme le montre clairement la **figure 8**.

Commençons par les entrées : selon le PIC, jusqu'à 16 sources différentes sont disponibles. La première étape vers une CLC « fonctionnelle » consiste à en sélectionner quatre, la logique interne ne pouvant traiter que quatre signaux à la fois.

À l'étape suivante, les signaux sont traités par une structure appelée *Data Gating*. Ce sont des portes « configurables » qui soumettent les signaux à différentes opérations logiques décidées par le développeur. Le plus important est la configuration des entrées : un signal peut être connecté à la porte correspondante tel quel, ou après inversion, ou pas du tout. De plus, le signal de sortie peut être transmis directement ou inversé. La troisième partie représente la véritable « substance » de la

## Accès à la mémoire moderne

Les routines d'accès à la mémoire flash développées par Lucio di Jasio ne sont guère utilisées en dehors des tout petits PIC, on leur préfère le code créé par le MCC. Pour ce faire, ouvrez le MCC et ajoutez un périphérique flash. La première méthode significative dans le fichier nouvellement créé, *memory.c*, sert à la lecture des données ; elle commence par désactiver les interruptions :

```
uint16_t FLASH_ReadWord(uint16_t flashAddr)
{
    uint8_t GIEBitValue = INTCONbits.GIE;
    // Sauvegarder l'état des interruptions
    INTCONbits.GIE = 0;
    // Désactiver les interruptions
```

Ensuite, le code renseigne les registres de contrôle avec l'adresse de destination et désactive l'accès à l'espace de configuration. La mise à un du drapeau RD démarre la lecture :

```
NVMADRL = (flashAddr & 0x00FF);
NVMADRH = ((flashAddr & 0xFF00) >> 8);
NVMCON1bits.NVMREGS = 0;
    // Désactiver l'espace de configuration
NVMCON1bits.RD = 1;    // Démarrer la lecture
```

Il faut deux NOP pour donner à la puce le temps de lire. Cela fait, on rétablit le bit de validation des interruptions dans son état précédent et on retourne la valeur acquise à l'appelant :

```
NOP();
NOP();
INTCONbits.GIE = GIEBitValue;
    // Restaurer l'état des interruptions
return ((uint16_t)((NVMDATH << 8) | NVMDATL));
}
```

Pour l'écriture des données, on dispose de deux méthodes, *WriteWord* et *WriteBlock*. *WriteWord* écrit un mot unique dans la mémoire, alors que *WriteBlock* suppose que les données arrivent sous le format d'un bloc flash. Comme *WriteWord* est le plus compliqué, nous commençons par son examen :

```
void FLASH_WriteWord(uint16_t flashAddr, uint16_t
    *ramBuf, uint16_t word)
{
    uint16_t blockStartAddr = (uint16_t)(flashAddr &
        ((END_FLASH-1) ^
        (ERASE_FLASH_BLOCKSIZE-1)));
    uint8_t offset = (uint8_t)(flashAddr &
        (ERASE_FLASH_BLOCKSIZE-1));
    uint8_t i;
```

Lors de l'écriture d'un seul mot, l'architecture interne de la

mémoire flash nous joue un tour puisque nous devons effacer toute la cellule en bloc. La méthode règle ce problème grâce au paramètre `uint16_t *ramBuf` qui est l'adresse d'une zone de mémoire tampon dans laquelle elle peut sauvegarder temporairement les données du bloc.

Ensuite, la méthode *FLASH\_ReadWord* copie les données contenues dans la mémoire flash dans cette zone tampon :

```
// Tout le bloc va être effacé.
// On fait une copie de sauvegarde des données
for (i=0; i
{
    ramBuf[i] = FLASH_ReadWord((blockStartAddr+i));
}

// Ecriture dans le tampon
ramBuf[offset] = word;
```

La routine utilise la variable d'offset comme index d'écriture de la valeur lue dans le tampon *ramBuf*, qui sera ensuite écrite en bloc dans la mémoire flash. Pour cela, la fonction appelle sa collègue *WriteBlock* :

```
// Ecriture de ramBuf dans bloc courant
FLASH_WriteBlock(blockStartAddr, ramBuf);
}
```

Le code de l'écriture d'un bloc entier n'a besoin que de l'adresse de destination et de celle du tampon qui contient le bloc à écrire :

```
int8_t FLASH_WriteBlock(uint16_t writeAddr, uint16_t
    *flashWordArray)
{
    uint16_t blockStartAddr = (uint16_t )
        (writeAddr &
            ((END_FLASH-1) ^
            (ERASE_FLASH_BLOCKSIZE-1)));
    uint8_t GIEBitValue = INTCONbits.GIE;
    // Sauvegarde de l'état des interruptions
    uint8_t i;
```

Comme les opérations d'écriture doivent toujours être alignées sur une frontière de bloc, la fonction commence par vérifier la validité de *writeAddr*. Si elle est invalide, la valeur -1 est retournée à l'appelant.

```
// Flash write doit commencer
// à une frontière de bloc
if( writeAddr != blockStartAddr )
{
    return -1;
}
```

Ensuite, on désactive les interruptions et on déverrouille la mémoire flash par *FLASH\_EraseBlock* :

```

INTCONbits.GIE = 0;
// Désactiver les interruptions

// Séquence d'effacement de bloc
FLASH_EraseBlock(writeAddr);

```

L'écriture proprement dite exige une configuration de bits appropriée dans le registre *NVMCON1*, qui est réalisée par trois accès à son champ de bits :

```

// Séquence d'écriture de bloc
NVMCON1bits.NVMREGS = 0;
// Désélectionner l'espace de configuration
NVMCON1bits.WREN = 1; // Autoriser l'écriture
NVMCON1bits.LWLO = 1;
// Charger uniquement les tampons d'écriture

```

Après le paramétrage de la mémoire flash, l'écriture proprement dite commence. Le PIC ne traite qu'un seul mot à la fois, même dans le cas d'une cellule complètement effacée. C'est pourquoi une boucle `for` est nécessaire pour traiter les octets un par un.

```

for (i=0; i<WRITE_FLASH_BLOCKSIZE; i++)
{
// Charger les 8 bits de poids faible de
// l'adresse d'écriture
NVMADRL = (writeAddr & 0xFF);
// Charger les 8 bits de poids fort de
// l'adresse d'écriture
NVMADRH = ((writeAddr & 0xFF00) >> 8);
// Copier les données vers l'adresse courante
NVMDATL = flashWordArray[i];
NVMDATH = ((flashWordArray[i] & 0xFF00) >> 8);
}

```

La première opération administrative consiste à charger les registres d'adresse et de données du moteur d'écriture. Comme la mémoire de programme de notre PIC est d'une largeur de 14 bits, nous devons paramétrer l'espace inférieur aussi bien que l'espace supérieur.

Une fois tous les registres de contrôle initialisés, on donne l'ordre d'écriture :

```

if(i == (WRITE_FLASH_BLOCKSIZE-1))
{
// Démarrage de l'écriture de la mémoire flash
NVMCON1bits.LWLO = 0;
}
NVMCON2 = 0x55;
NVMCON2 = 0xAA;
NVMCON1bits.WR = 1;
NOP();
NOP();
writeAddr++;
}

```

Après l'écriture des octets, la fonction désactive l'autorisation d'écriture et restaure le bit de validation des interruptions :

```

NVMCON1bits.WREN = 0;
// Interdire l'écriture
INTCONbits.GIE = GIEBitValue;
// Restaurer l'état des interruptions
return 0;
}

```

Pour finir, l'API contient encore une commande d'effacement d'un bloc. Son code est constitué, grosso modo, d'éléments déjà passés en revue dans le processus d'écriture.

```

void FLASH_EraseBlock(uint16_t startAddr)
{
uint8_t GIEBitValue = INTCONbits.GIE;
// Sauvegarder l'état courant

INTCONbits.GIE = 0;
// Désactiver les interruptions
// Charger les 8 bits de poids faible de
// l'adresse d'effacement
NVMADRL = (startAddr & 0xFF);
// Charger les 8 bits de poids fort de
// l'adresse d'effacement
NVMADRH = ((startAddr & 0xFF00) >> 8);

// Séquence d'effacement du bloc
NVMCON1bits.NVMREGS = 0;
// Désélectionner l'espace de configuration
NVMCON1bits.FREE = 1;
// Spécifier une opération d'effacement
NVMCON1bits.WREN = 1;
// Autoriser les cycles d'effacement
// Début de la séquence de validation de
// l'effacement
NVMCON2 = 0x55;
NVMCON2 = 0xAA;
NVMCON1bits.WR = 1; // Lancer l'effacement
NOP();
NOP();

NVMCON1bits.WREN = 0; // Désactiver l'écriture
INTCONbits.GIE = GIEBitValue;
// Restaurer l'état des interruptions
}

```

`FLASH_EraseBlock` se distingue d'une fonction d'écriture par le fait qu'elle manipule le bit d'effacement dans le registre *NVMCON1*. Sinon, on y retrouve la routine de déverrouillage vue précédemment et la mise à 1 du bit *WR* pour lancer l'écriture.

Dans le fichier d'en-tête, il y a encore trois constantes qui fixent la taille d'un bloc et l'adresse de fin de la mémoire flash.

```

#define WRITE_FLASH_BLOCKSIZE 32
#define ERASE_FLASH_BLOCKSIZE 32
#define END_FLASH 0x8000

```

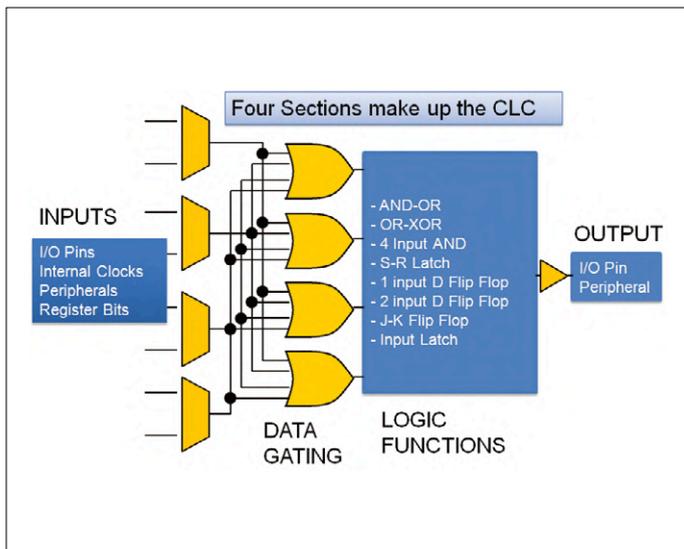


Figure 8. La CLC est constituée de quatre éléments connectés en série (source : Microchip).

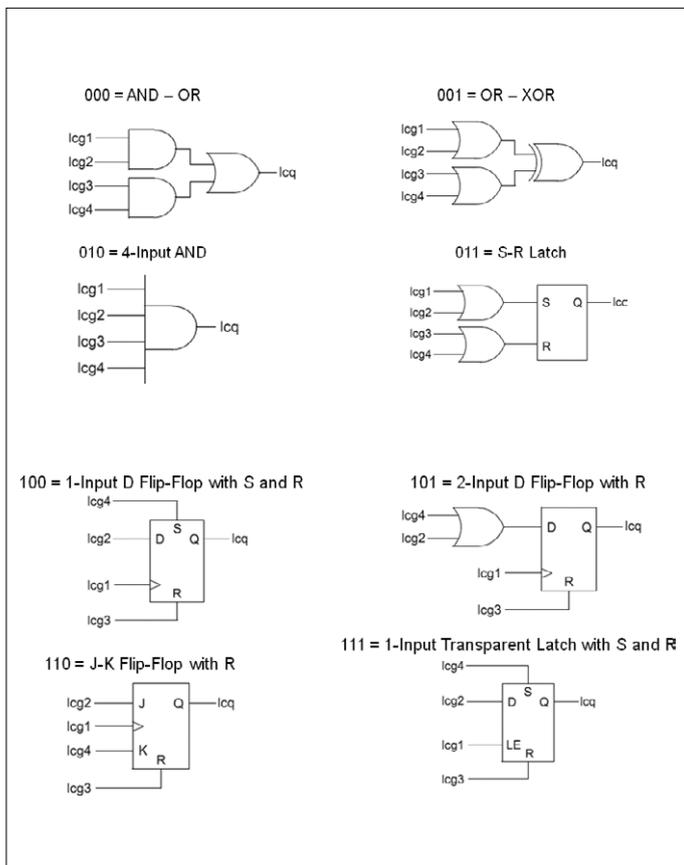


Figure 9. La cellule logique comprend huit modes de fonctionnement différents (source : <http://microchipdeveloper.com/8bit:clc>).

CLC, une des huit fonctions logiques différentes – là encore, la meilleure façon de présenter les huit variantes est un tableau de symboles (fig. 9).

Enfin, la question se pose de savoir que faire des résultats. C'est de la responsabilité du bloc de sortie : outre le déclenchement d'interruptions, il est également possible de router les résultats directement vers une broche de sortie dédiée (fig. 10).

### Et ça marche à quelle vitesse ?

La première réaction de « vieux de la vieille » de l'auteur est de mettre la CLC sur la sellette en ce qui concerne le temps de latence. Pour cela, livrons-nous à une expérience simple : mettons à l'entrée d'une porte ET un signal carré fourni par un générateur de fonctions.

On peut alors mesurer le temps de latence avec un oscilloscope. Pour éviter l'ennui d'un tâtonnement manuel dans les registres de configuration, Microchip a ajouté au générateur de code MCC un module de configuration graphique.

Par souci de simplicité, créez un nouveau projet MPLAB et ouvrez le MCC. Puis ajoutez un élément de type CLC à la configuration pour afficher à l'écran le diagramme de configuration de la figure 11. Le PIC16F1503 comporte deux CLC ; pour l'instant, nous allons utiliser CLC2. Le cycle de travail (rubrique « Internal Clock ») est de 500KHz<sub>MF</sub>, ce qui représente un bon compromis entre la mesurabilité et la charge du générateur de formes d'onde arbitraires.

Prenez soin de ne pas utiliser CLC1 par erreur, car il y a là un petit hic. Cela vous obligerait à rediriger MCLR, ce qui, à cause de la structure interne du contrôleur, empêche le débogage, assez curieusement même si l'on utilise une extension de débogage. Comme l'assignation entre CLC et entrée est imposée par Microchip, la solution du problème passe par l'utilisation de CLC2. Elle utilise RC0 comme sortie et RC3 ou RC4 comme entrée, ce qui exclut une collision entre broches.

Commencez par cocher l'option *Enable CLC Output*, pour établir la liaison entre la sortie de la CLC et la broche de sortie. MPLAB effectue automatiquement cette connexion, il n'y a rien de plus à faire.

Remarquez que la CLC ne se contente pas du rôle de « pilote » direct d'une broche de sortie. La valeur calculée par le circuit interne peut également être lue via un registre – pour cela, MCC produit un code basé sur le schéma suivant :

```
bool CLC1_OutputStatusGet(void) {
return(CLC1CONbits.LC1OUT);
}
```

### Liens

- [1] Note d'application avec code de démo : [www.microchip.com/wwwAppNotes/AppNotes.aspx?appnote=en567894](http://www.microchip.com/wwwAppNotes/AppNotes.aspx?appnote=en567894)
- [2] Page de l'article : [www.elektormagazine.fr/180734-04](http://www.elektormagazine.fr/180734-04)
- [3] Document 'Configurable Logic Cell Tips 'n Tricks' : [ww1.microchip.com/downloads/en/DeviceDoc/41631B.pdf](http://ww1.microchip.com/downloads/en/DeviceDoc/41631B.pdf)

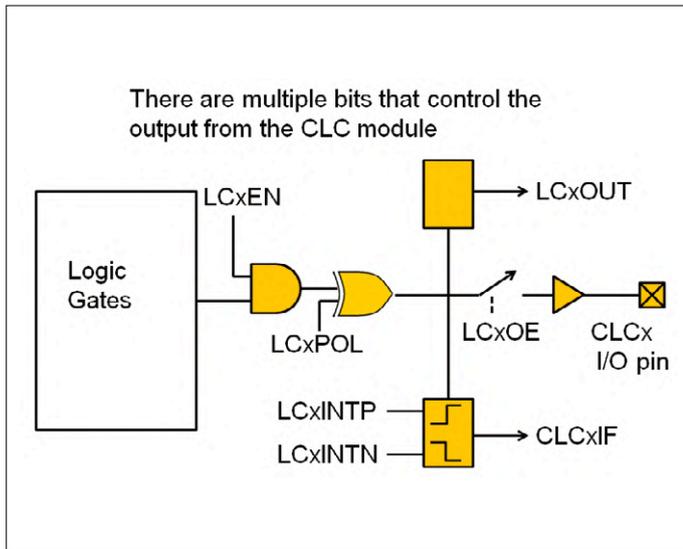


Figure 10. La souplesse de Microchip s'exprime même du côté des sorties (source : Microchip).

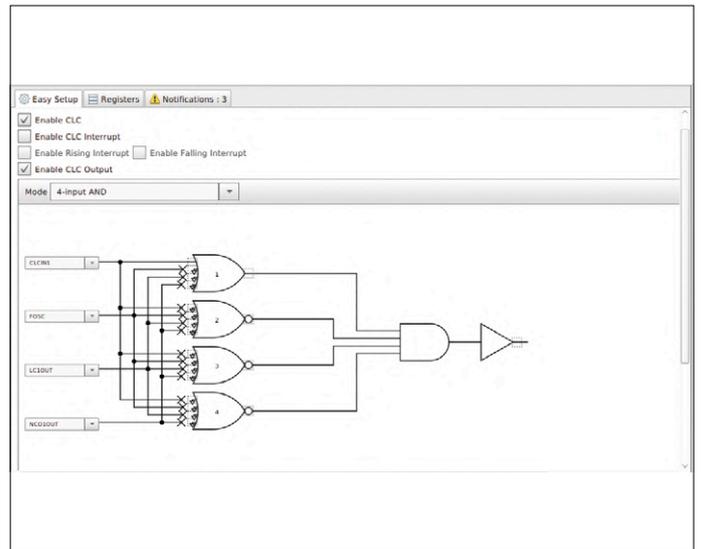


Figure 11. Configuration des cellules logiques.

De plus, il est possible d'utiliser les fronts montants ou descendants pour provoquer des interruptions. Mais nous ne pouvons en dire plus ici.

Notre tâche suivante consiste à sélectionner la topologie interne de la CLC. Le mode de fonctionnement *4AND*, qui, comme indiqué ci-dessus, fournit une porte ET à quatre entrées, nous suffira pour les étapes suivantes. Le tampon, ou l'inverseur situé devant le « puits de données » est obligatoire : s'il est en mode *tampon*, il n'influencera toutefois pas le signal à traiter. Nous pouvons donc alimenter le PIC avec le signal carré et obtenir la mesure du temps de latence de la **figure 12**.

Microchip propose sous [3] une série d'exemples de circuits décrivant des cas d'utilisation avancée de la CLC. Outre la possibilité de connecter des CLC « en série », notons que la CLC continuera normalement de fonctionner même avec le cœur du processeur en sommeil, manière très efficace d'éviter les cycles de réveil inutiles. ◀

(180734-04 – version française : Helmut Müller)

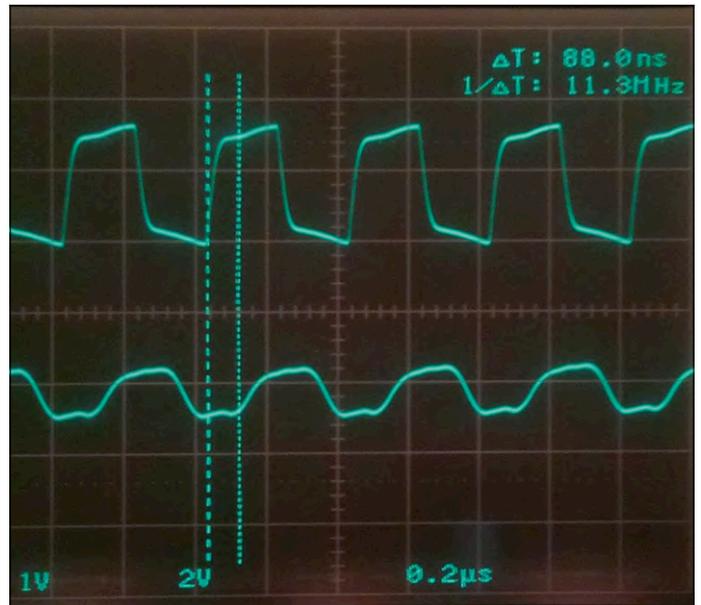
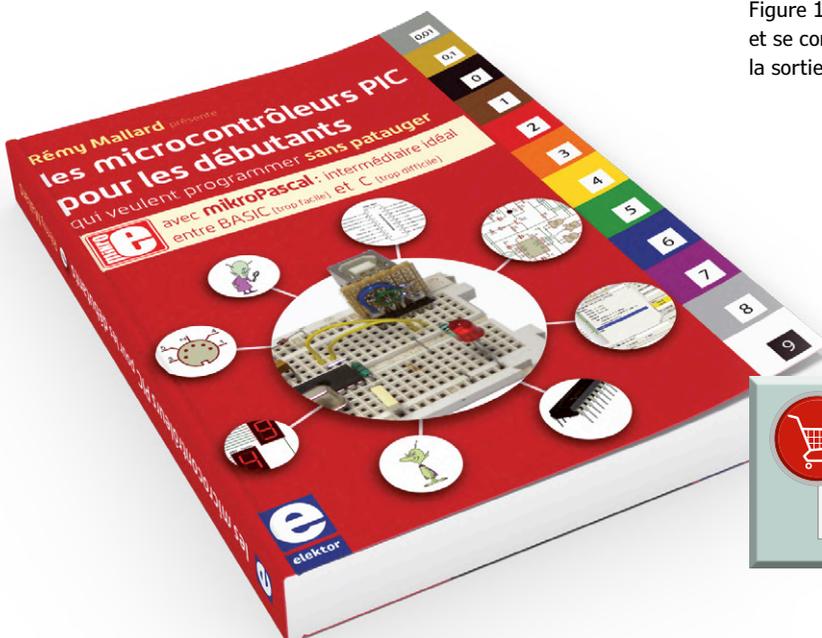


Figure 12. Mesure du temps de latence. La CLC est configurée en tampon et se contente de transférer directement le signal d'entrée (en haut) vers la sortie (en bas).



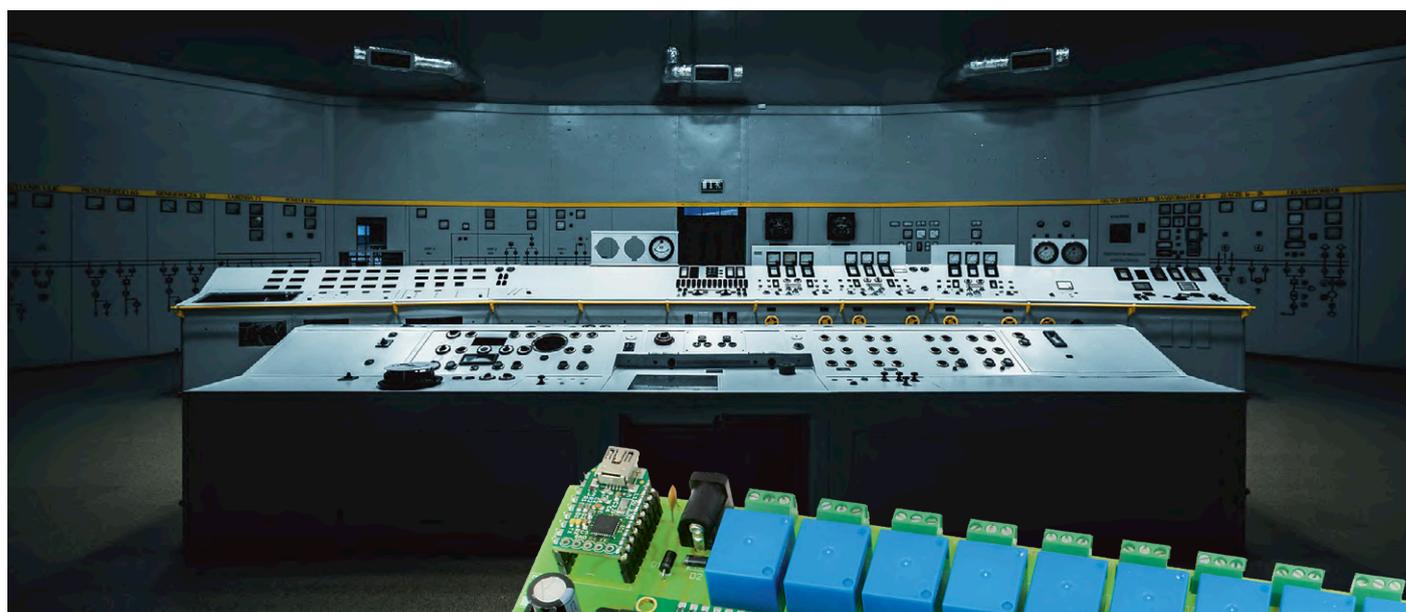
@ [WWW.ELEKTOR.FR](http://WWW.ELEKTOR.FR)

→ Livre « Les microcontrôleurs PIC pour les débutants », Rémy Mallard, Elektor

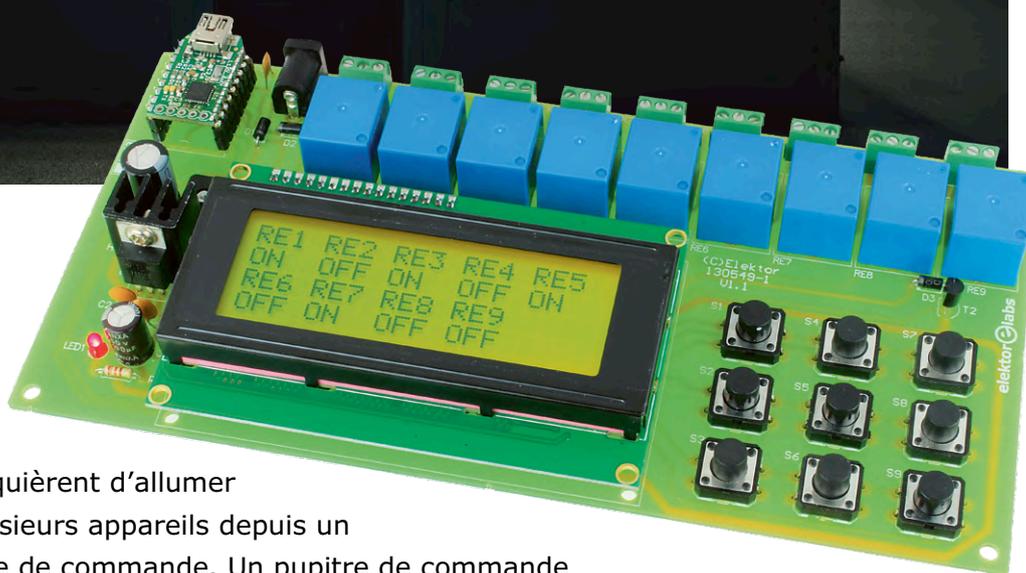
[www.elektor.fr/les-microcontroleurs-pic-pour-les-debutants](http://www.elektor.fr/les-microcontroleurs-pic-pour-les-debutants)

# carte de commutation à relais à 9 canaux

## avec commande locale ainsi qu'à distance



Sunil Malekar (Inde) et  
Clemens Valens (labo d'Elektor)



De nombreuses situations requièrent d'allumer ou d'éteindre séparément plusieurs appareils depuis un point central comme une salle de commande. Un pupitre de commande avec un interrupteur pour chaque appareil semble une solution simple, mais elle ne permet pas la commande à distance, ni par ordinateur. Le circuit présenté ici apporte ces fonctions.

Les relais sont d'excellents dispositifs pour l'automatisation des commutations. Commandés par une tension externe, ils peuvent commuter toutes sortes de charges presque comme un interrupteur mécanique. Les relais sont faciles à piloter avec un microcontrôleur épaulé par des transistors de commande. Les

microcontrôleurs sont programmables et embarquent tout ce qu'il faut pour communiquer avec d'autres appareils comme les ordinateurs et les réseaux. C'est à peu près tout ce qu'il faut pour démarrer notre projet : ajoutez quelques extras comme un clavier et un afficheur pour en profiter.

### Un schéma des plus simples

La **figure 1** montre le schéma du projet. Vous aurez sans doute deviné que le microcontrôleur est le grand rectangle marqué IC2. Il est du type PIC18F45K22, choisi principalement pour son nombre impressionnant de lignes de ports d'E/S (un énorme 36), sans oublier son petit



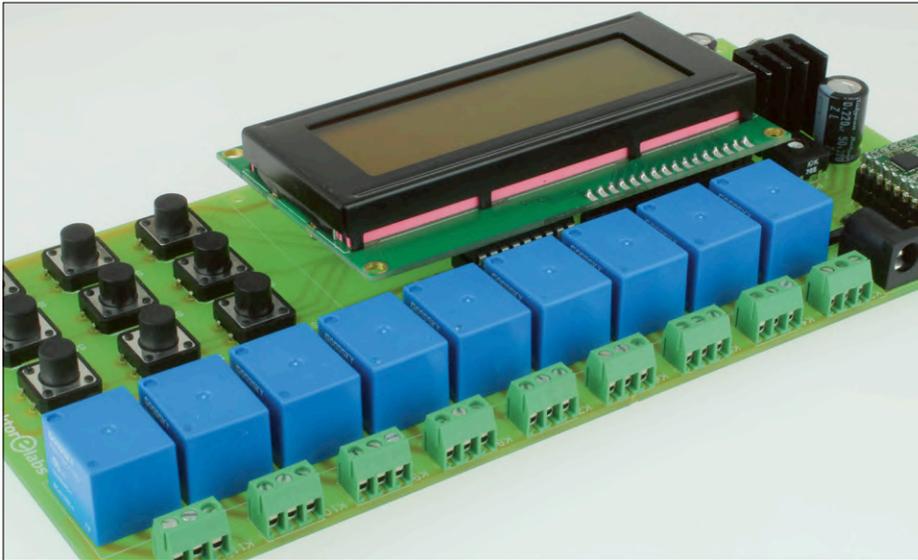


Figure 2. Chacun des neuf relais a un contact normalement ouvert (NO) et un contact normalement fermé (NC), accessibles sur un bornier à 3 broches.

type « 4 × 20 »). Il intègre une LED de rétroéclairage, pilotée ici par T1 car l'UC ne peut pas fournir la quantité de courant nécessaire.

Le mode de pilotage de l'afficheur est simplifié en écriture seule sur 4 bits, ce qui économise des broches et des pistes du CI. En contrepartie, l'UC n'a alors aucun moyen de savoir si l'afficheur fait ce qu'il est censé faire. Cependant, tant que l'afficheur dispose d'assez de temps pour traiter les commandes qu'on lui envoie, cela ne pose pas de problème. À la première mise sous tension de la carte, n'oubliez pas de régler le contraste de l'afficheur LCD avec le potentiomètre P1. À défaut, le texte sur l'afficheur risque de ne pas être visible.

### Clavier 3 × 3

Des boutons-poussoirs permettent de commander les relais séparément et de manière pratique. Les boutons étant disposés en matrice de 3 × 3, six liaisons suffisent pour les relier à l'UC. Le réseau de résistances RN1 fournit les résistances de tirage vers le haut pour que les niveaux au repos soient bien définis. Notez la robustesse des boutons-poussoirs utilisés, avec une empreinte

de CI de 12 × 12 mm qui rend leur manœuvre simple et confortable.

### Port série-USB

Comme le circuit n'utilise que des composants traversants, nous avons choisi un module de conversion USB vers série comme interface de communication. Ceci évite d'avoir à souder de minuscules composants CMS. Le module USB est un vétéran [1] qui a maintes fois fait ses preuves. N'oubliez pas que ce module a un cavalier (JP1) pour régler le niveau de signal (5 V ou 3,3 V). Veillez à le bloquer dans la position « 5V » avec une goutte de soudure.

La diode D2 empêche la connexion de la ligne 5 V du port USB à l'alimentation 5 V du circuit. Dans l'autre sens, cela fonctionne, donc on peut si besoin laisser la broche d'alimentation USB déconnectée. Dans ce cas, le circuit ne chargera pas le port USB de l'hôte.

### Alimentation

Comme les relais sont de type 12 V, la tension d'entrée – à appliquer sur K1 – est forcément de 12 VCC. Le courant maximal consommé (tous relais activés, rétroéclairage allumé, UC à plein

régime) est d'environ 500 mA, ce qui fait qu'un bloc d'alimentation de 12 V, 6 VA CA-vers-CC devrait suffire pour alimenter la carte. Comme les relais disposent de contacts normalement ouverts (NO) et normalement fermés (NC) (fig. 2), les charges qui sont supposées être actives plus de 50% du temps doivent être connectées aux contacts NC pour économiser un peu d'énergie.

Le circuit IC1 abaisse la tension d'entrée de 12 V à 5 V pour le reste du circuit. IC1 doit supporter un écart de tension de presque 7 V et de ce fait il peut chauffer beaucoup. C'est pourquoi un radiateur est recommandé.

La LED1 est un voyant d'alimentation.

### Programmation in situ

Enfin et surtout vient K2, nécessaire pour programmer le micrologiciel dans l'UC. Ce connecteur respecte le standard de brochage ICSP de Microchip, donc la connexion d'un programmeur compatible PIC (PICKit/ICD) devrait fonctionner.

### Logiciel

Un dispositif aussi simple que cette carte à relais requiert un logiciel rudimentaire pour fonctionner. Le logiciel est écrit en pur C et peut être compilé gratuitement avec MPLAB X de Microchip. Le portage du code vers d'autres compilateurs ne devrait pas être trop difficile.

L'UC fonctionne à 16 MHz grâce à son oscillateur interne, ce qui explique l'absence d'un quartz dans le schéma.

Après initialisation des relais (tous désactivés) et de l'afficheur, un minuteur est lancé et le port série est activé. Puis le programme attend que quelque chose se produise – soit la pression d'une touche, soit l'arrivée d'une commande sur le port série. Dès qu'une commande est reçue sur le port série ou suite à la pression d'une touche, l'UC procède à l'exécution de la commande.

Voici comment fonctionne le balayage du clavier. Une seule rangée (comme dessiné sur la figure 1) est à l'état bas à la fois. L'UC lit le niveau des colonnes et si l'une d'elles est à l'état bas, l'action correspondante est exécutée. Ainsi

### Liens

[1] Passerelle USB/série BOB-FT232R : [www.elektormagazine.fr/110553](http://www.elektormagazine.fr/110553)

[2] Page du projet au labo d'Elektor : [www.elektor-labs.fr/1778](http://www.elektor-labs.fr/1778)

lorsque la rangée 1 est mise à la masse, seules les touches S1, S4 et S7 seront lues. La rangée 2 concerne les touches S2, S5 et S8, et la rangée 3 les touches S3, S6 et S9.

### \$RLYxO[N|F]...&

... est la syntaxe de la commande des relais prise en charge par l'UC sur le port série. Ce code énigmatique signifie que le message vers la carte doit commencer par le caractère « \$ » et se terminer par un « & ». Le bloc « RLYxO » dans la syntaxe ci-dessus indique le relais à commuter, où « x » est le numéro de 1 à 9. Nous découvrons encore ici un autre avantage de ne pas avoir plus de neuf relais : la longueur de la partie utile du message est toujours la même. L'état du relais est codé par « N » (actif) ou par « F » (inactif). Exemples :

- \$RLY3ON& — active le relais n°3
- \$RLY7OF& — désactive le relais n°7
- \$RLY4OFF& — désactive le relais n°4

Notez le « F » superflu dans le dernier exemple. C'est uniquement une option qui facilite la compréhension des commandes par l'être humain.

La réception d'une commande de manœuvre d'un relais est confirmée par un message d'état sur l'état actuel du relais (**RELAYx SWITCHED [ON|OFF]**). Lorsque l'état actuel retourné ne correspond pas à l'état demandé, il y a peut-être un problème à résoudre...

Pour demander l'état de tous les relais d'un coup, envoyez la commande **\$STATE&**.

### À l'aide !

Lorsque l'UC reçoit la commande \$H&, elle va renvoyer l'écran d'aide avec la liste de toutes les commandes disponibles.

### Construction

La construction de la carte à relais est plutôt simple, car il n'y a pas beaucoup de composants. Elle ne requiert pas un haut niveau de compétences en soudage. Commencez par installer les composants à deux pattes comme les résistances et les diodes, puis continuez avec les autres composants par taille croissante. Attention à la polarité des composants polarisés comme les diodes et les condensateurs électrolytiques.

Avant d'enficher les circuits intégrés, connectez la tension d'entrée de 12 V et vérifiez avec un voltmètre que l'alimentation de 5 V est bien du 5 V.

## LISTE DES COMPOSANTS

### Résistances

Par défaut : 5%, 0,25 W  
 R1 = 330 Ω  
 R2, R7 = 4,7 kΩ  
 R3, R4, R6 = 0 Ω  
 R5 = 4,7 Ω  
 RN1 = 10 kΩ, réseau SIL à 8 éléments  
 P1 = potentiomètre de réglage 10 kΩ

### Condensateurs

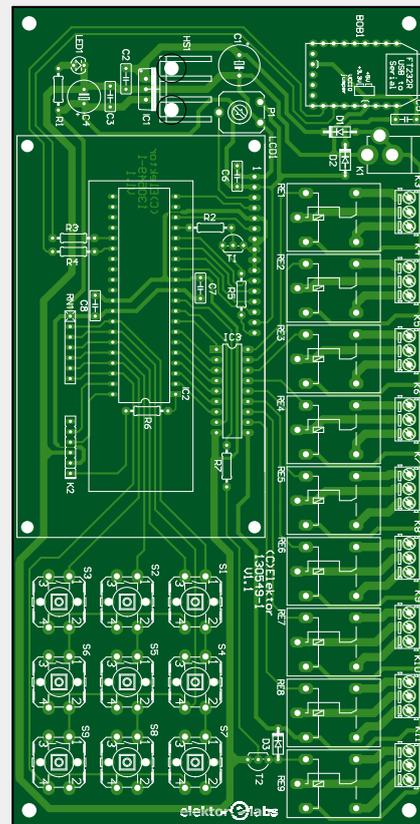
C1 = 220 µF, 50 V  
 C2, C3, C5, C6, C7, C8 = 100 nF  
 C4 = 100 µF, 50 V

### Semi-conducteurs

D1, D3 = 1N4007  
 D2 = BAT43  
 IC1 = MC7805  
 IC2 = PIC18F45K22-E/P, programmé  
 IC3 = ULN2803A  
 T1, T2 = BC547  
 LED1 = rouge, 3 mm

### Divers

K1 = fiche jack, broche centrale 1,95 mm  
 K2 = barrette mâle à 6 broches, au pas de 2,54 mm  
 K3-K11 = bornier à 3 broches, au pas de 3,5 mm  
 LCD1 = LCD, 4×20, 5 V, avec rétroéclairage, barrette mâle et support à 14 broches  
 RE1-RE9 = relais G5LE-1, bobine 12 VDC ou équivalent  
 S1-S9 = bouton-poussoir, 12 mm, B3F-4000 ou équivalent  
 BoB1 = passerelle/BoB USB/série FT232R, www.elektor.fr réf. 15435



Support DIP, 40 contacts  
 Support DIP, 18 contacts  
 HS1 = radiateur de style TO220

mentation de 5 V est bien du 5 V. Débranchez l'alimentation de la carte et installez les circuits intégrés restants (attention à l'orientation). Connectez un programmeur compatible ICSP et alimentez la carte. Vous devriez maintenant pouvoir charger le fichier HEX (à télécharger depuis [1]) dans l'UC. Ensuite débranchez l'alimentation et installez l'afficheur LCD. Rebrancher la carte devrait produire l'affichage d'un message de bienvenue sur l'afficheur. Vous pourriez devoir agir sur P1 pour le voir. Appuyer sur les boutons-poussoirs devrait faire cliquer les relais correspondants.

### BoB1 au contrôle sol

Enfichez BoB1, le module USB-série, sur la carte et connectez-le à un port USB libre d'un PC. Ce dernier devrait le détec-

ter en tant que port série. Lancer un émulateur de terminal et configurez-le sur 9600 bauds, huit bits de données, sans parité et un bit d'arrêt (9600n81). Lorsque la carte à relais est alimentée, elle devrait maintenant répondre aux commandes que vous tapez dans le terminal. Si c'est le cas, c'est terminé ; sinon, vous avez un peu de débogage à faire.

Joyeuses commutations ! ◀

(190027-03

version française : Denis Lafourcade)

# kits à microcontrôleurs pour utilisateurs chevronnés pour les projets exigeants

Il arrive un moment où l'Arduino Uno et les cartes à base de microcontrôleurs à 8 bits ne suffisent plus. Les projets deviennent plus ambitieux, alors le besoin de davantage de puissance de calcul rend inévitable le passage à des processeurs à 32 bits et à des environnements de développement intégrés professionnels. Mais ce n'est pas sorcier, comme le montrent dans cet article nos auteurs Viacheslav Gromov et Tam Hanna.

Tam Hanna, l'un des journalistes de langue allemande les plus connus dans le domaine du logiciel, vit d'ordinaire à Bratislava (Slovaquie), mais il voyage souvent d'un congrès d'électronique à une session d'informatique. Il s'est spécialisé dans la programmation pour PC, Android, les ordinateurs monocartes et l'Internet des Objets. Tam a testé pour nous la carte « Wemos Lolin ESP32 OLED » qui peut communiquer avec le monde de la toile par Wi-Fi ou Bluetooth.

Viacheslav Gromov ([readers@gromov.de](mailto:readers@gromov.de)) est un journaliste

indépendant et un développeur en électronique. Depuis sa scolarité, il écrit des articles et des livres techniques pour des publications diverses. Il s'est spécialisé dans l'intelligence artificielle, les microcontrôleurs ARM et les FPGA, sur lesquels il ne se contente pas d'écrire des manuels pour les écoles supérieures et les fabricants de semi-conducteurs ; il participe aussi à des programmes internationaux de développement de produits. Il a récemment créé une startup de conception de solutions d'intelligence artificielle pour la voiture autonome.

## Wemos Lolin ESP32 OLED

Tam Hanna

La carte Lolin ESP32 OLED du fabricant chinois Wemos combine un contrôleur ESP32 et son antenne avec un petit afficheur OLED piloté par un contrôleur SSD1306. Les deux contrôleurs sont interconnectés par un bus I2C. Le module peut être programmé au moyen de l'EDI Arduino ; une bibliothèque pour le contrôleur OLED est disponible.

À titre d'exemple, nous allons réaliser un système d'acquisition de données muni d'une fonction de sélection automatique de gammes de mesure. Comme d'habitude, mon ordinateur de développement tourne sous Linux.

### Aspect technique

C'est le moment de faire le point sur l'architecture finale. Dans notre cas, elle se présente comme sur la **figure 1**. Si l'on étudie avec attention le diagramme fonctionnel, on constate qu'à ce moment, je n'ai pas encore décidé du type de matériel pour la réalisation de la commutation automatique de gamme.

Si, comme moi, vous êtes pressé par les délais, il est d'autant plus important de s'en tenir à du code « tout fait ». Dans les collections de bibliothèques, il existe un pilote d'afficheur. Comme le montre la **figure 2**, il est possible de télécharger les fichiers avec l'EDI Arduino.

Un vieux dicton prétend qu'il y a une quantité de façons de dépouiller un chat. Cela s'applique aussi aux afficheurs I<sup>2</sup>C : le fournisseur de la carte a le libre choix des broches par lesquelles

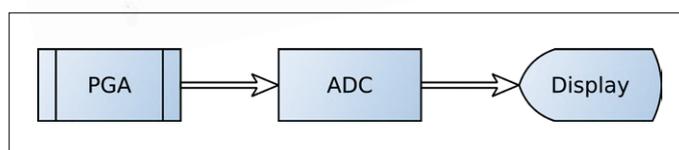
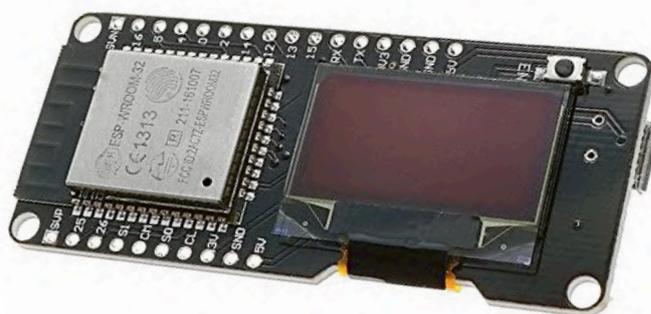


Figure 1. La fonction de sélection automatique de la gamme de mesure améliore l'usage du CA/N.

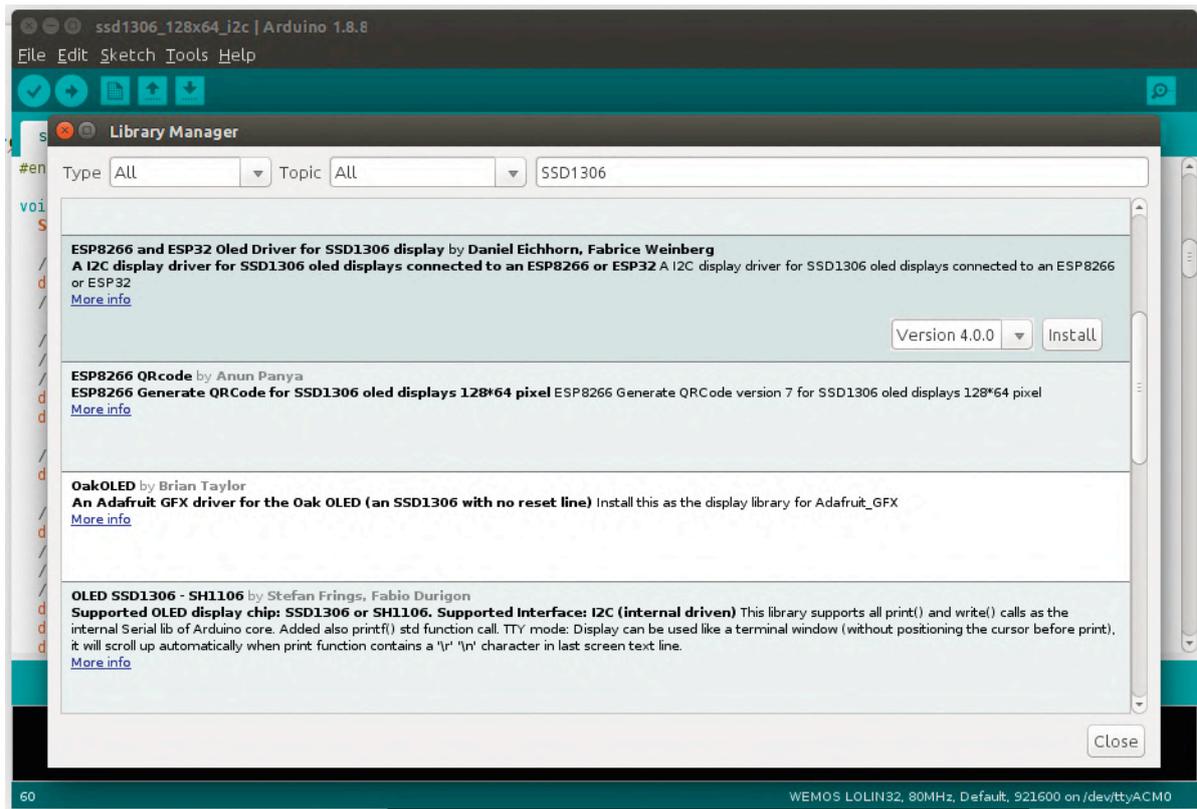


Figure 2. La bibliothèque gère le pilotage de l'afficheur.

il connecte l'afficheur avec l'ESP32. Le corpus ne contient donc que la déclaration suivante qui n'est pas compilable :

```
// Initialize the OLED display using Wire library
SSD1306Wire display(0x3c, D3, D5);
```

Reportons dans notre programme les valeurs valides pour l'I<sup>2</sup>C, fournies par la documentation de la carte Lolin. Il vient :

```
// Initialize the OLED display using Wire library
SSD1306Wire display(0x3c, 5, 4);
```

Pour commencer, utilisons la commande `dmesg` pour localiser l'ESP32 dans l'arborescence `/dev` du système :

```
tamhan@tamhan-thinkpad:~$ dmesg
...
[29329.470219] cp210x 3-2:1.0: cp210x converter
detected
[29329.470635] usb 3-2: cp210x converter now attached
to ttyUSB0
```

Ensuite, il est judicieux d'exécuter l'exemple de code d'affichage téléchargé. Le but de cette approche est de vérifier la chaîne d'outils de compilation.

### Un dégraissage ciblé

Nous utiliserons le programme de démonstration comme point de départ de notre propre projet. Diomidis Spinellis déclare

dans son livre « Code Reading » [1] qu'il faut supposer, lors du portage d'un programme, que celui-ci « fonctionne ». La base de code est compilée et retravaillée jusqu'à ce qu'il n'y ait plus d'erreur.

Avec les pilotes matériels, on procède de manière analogue. Comme solution minimale, le programme suivant a fonctionné :

```
#include <Wire.h>
#include "SSD1306Wire.h"

SSD1306Wire display(0x3c, 5, 4);

void setup() {
  display.init();
  display.flipScreenVertically();
  display.setFont(ArialMT_Plain_10);
}

void loop() {
  display.setColor(WHITE);
  display.drawCircle(32, 32, 2*3);
  display.display();
  delay(1000);
}
```

Il est important que le pilote alloue un tampon de trame « interne ». Ce dernier est vidé par l'appel de la méthode `init()` et son contenu est transféré à l'afficheur à l'appel de

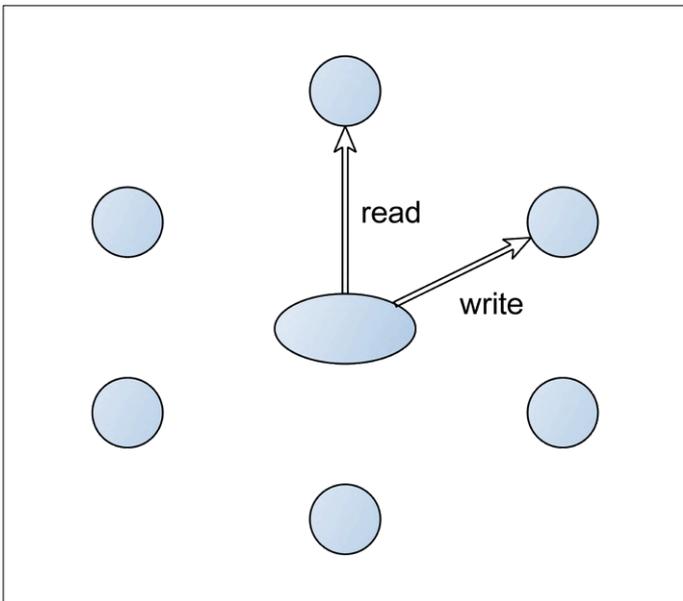


Figure 3. Six cercles aident à la compréhension du système.

la méthode `display()`. Si l'une de ces méthodes manque, l'affichage reste noir (voir l'**encadré** « L'OLED zombie »).

### Du rendu

Une fois qu'un cercle apparaît sur l'écran, la tâche suivante consiste à fournir une infrastructure d'acquisition. Notre enregistreur fonctionnera relativement lentement - il suffit d'enregistrer un signal toutes les 250 ms.

Cette « lenteur » est utile, entre autres, parce qu'elle nous épargne d'avoir à travailler avec des interruptions, etc. Les enregistreurs de données ne diffèrent des oscilloscopes que par le fait qu'ils acceptent continuellement des informations et les affichent en permanence comme un traceur de courbes caractéristiques, produisant ainsi un diagramme de données en perpétuelle évolution.

La réalisation d'un enregistreur de données exige l'usage d'un procédé d'enregistrement *round-robin* (tourniquet). Derrière ce terme anglo-saxon se cache une idée simple, représentée schématiquement sur la **figure 3**. Pendant le fonctionnement, les deux index tournent en rond. L'index de lecture « précède » l'index d'écriture d'un emplacement mémoire. Il en résulte au cours du temps un diagramme « continu » de données. Même si notre ESP32 est généreux en mémoire, nous allons,

## L'OLED zombie

Les afficheurs organiques ont la désagréable particularité d'avoir l'air morts lorsqu'ils présentent un écran complètement noir. Cela a été une cause fréquente de problèmes lors de l'introduction des écrans organiques dans le domaine des ordinateurs de poche. Les moteurs de jeux affichaient souvent un écran noir au cours de leur fonctionnement, ce qui était interprété par les développeurs comme un dysfonctionnement du matériel ou du système d'exploitation.

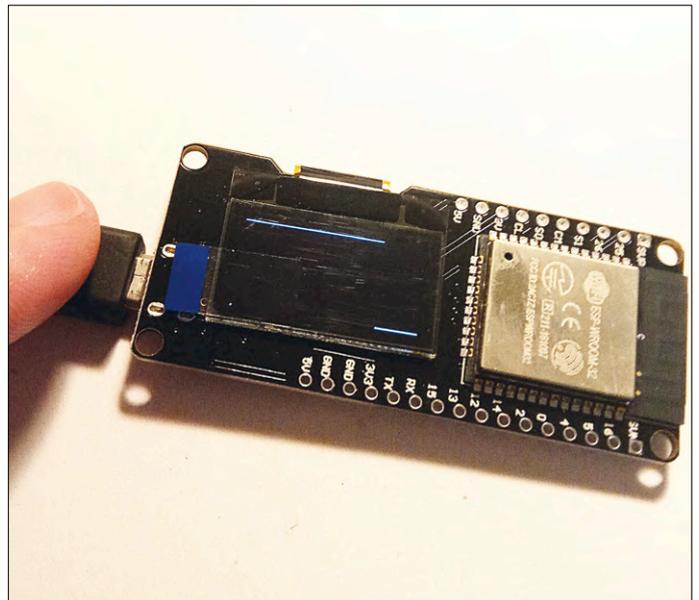


Figure 4. Après le démarrage, les valeurs élevées remplacent les zéros sur l'afficheur.

pour commencer, limiter la quantité de données en fonction de la capacité d'affichage de l'afficheur. Le code suivant est alors nécessaire pour déclarer le tampon circulaire :

```
SSD1306Wire display(0x3c, 5, 4);
```

```
//CAVE - 12bit ADC
int myAcquisitionField[128];
int writingNow=0;
int readingNow=1;
```

Dans le commentaire, l'auteur indique que les variables `int` utilisées pour l'enregistrement comportent un peu de « mou » – un indice pour les développeurs tiers qui, à l'occasion, voudraient analyser ce code à la recherche d'optimisations.

Aucune modification de la fonction `setup()` n'étant nécessaire, nous pouvons faire l'économie de l'impression de son code. La méthode `loop()` présente davantage d'intérêt :

```
void loop() {
    // put your main code here, to run repeatedly:
    myAcquisitionField[writingNow] = adcRead();
    writingNow = (writingNow + 1) % 128;
    readingNow = (readingNow + 1) % 128;
    drawChart();
    delay(100);
}
```

Pour commencer, on va lire la valeur courante du convertisseur A/N et on l'écrit dans l'emplacement mémoire sur lequel pointe l'index d'écriture, puis les deux index sont incrémentés. Cette opération est suivie de l'application de l'opérateur modulo du langage C, qui retourne le reste de la division par 128. Les index évoluent donc entre 0 et 127.

Enfin, et nous en aurons terminé, il faut encore tracer le diagramme et insérer un délai.

## Valeurs de mesure simulées

Pour les tests, nous simulons les valeurs de mesure. Si notre code est bien modulaire, nous pouvons aisément simuler le convertisseur A/N (CA/N) en remplaçant l'acquisition des données par une méthode qui fournit des valeurs précalculées. Pour commencer, envoyons la valeur 4000 :

```
int adcRead()
{
    return 4000;
}
```

Ce choix n'est pas fortuit : en choisissant des valeurs de mesure, il faut s'efforcer de prendre des valeurs qui sont en bordure du domaine de validité. Les instabilités algorithmiques et autres phénomènes sympathiques se produisent le plus souvent quand les valeurs tutoient la frontière.

Nous pouvons alors réaliser la fonction d'affichage :

```
void drawChart()
{
    display.clear();
    display.setColor(WHITE);
    for(int i=0;i<128;i++){
        int pointVal = myAcquisitionField[(i+readingNow) % 128];
        pointVal = pointVal >> 6;
        display.setPixel (i, pointVal);
    }
    display.display();
}
```

Les enregistreurs de type « bande de papier avec une pointe-feutre » ne sont jamais au repos puisqu'ils doivent rafraîchir le contenu de l'afficheur à chaque nouvelle mesure. Cela signifie qu'il ne reste généralement plus de temps pour d'autres processus complexes – la possibilité de déplacer et de « recycler » par DMA (*Direct Memory Access*) des parties du tampon est ici exclue.

Les divisions ont toujours causé pas mal de souci aux ordinateurs. Notre programme les évite par une astuce. Le convertisseur A/N de l'ESP32 fournit des valeurs dans le domaine de 0 à 4095. Notre afficheur a une résolution de 64 sur l'axe des ordonnées, ce qui est une valeur sympathique pour un diviseur, car elle transforme la division en une opération de décalage bien moins consommatrice en ressources. Le reste du programme est du code standard.

Nous pouvons maintenant procéder à un premier test. La variable `myAcquisitionField` ayant été déclarée globale, le compilateur l'initialise à zéro. Pendant l'exécution du programme, on constate qu'après le démarrage, les valeurs élevées se substituent aux valeurs basses sur l'afficheur (**fig. 4**).

## Commutation de gamme automatique

Nous devons ensuite nous préoccuper de la commutation de la gamme de mesure. Comme précédemment, la meilleure idée est celle qu'on emprunte à un collègue. Les multimètres sont des appareils qui embarquent depuis toujours des amplificateurs de mesure.

Quoique les manuels d'entretien des multimètres classiques soient habituellement une merveilleuse source d'inspiration

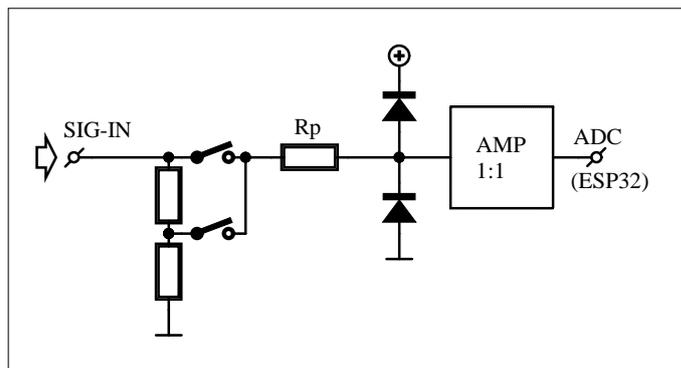


Figure 5. Ce circuit munit l'ESP32 de deux gammes de mesure.

## Holzleitner vous en dit plus !

Les courants de fuite des diodes sont toujours une source de problèmes, en particulier avec les convertisseurs A/N de haute précision - dans la note d'application figurait, par exemple, l'ID101 d'Intersil, qui n'est plus disponible. Peter Holzleitner, bien connu dans le monde de la réparation des appareils de mesure Solartron, propose sous [3] une excursion intéressante dans le domaine des « diodes alternatives ».

pour les concepteurs en herbe, ces circuits sont exagérément surdimensionnés pour notre projet, ne serait-ce qu'à cause de la linéarité plutôt minimale de notre convertisseur A/N.

On pourra donc plutôt s'intéresser à l'offre bas de gamme des concepteurs de puces de multimètres. Dans leurs notes d'application, on devrait pourtant trouver quelque chose d'utilisable ! La lecture du document [2] de Renesas est particulièrement instructive, mais une bonne partie en est malheureusement très insuffisante, en particulier ce qui concerne la description de la commutation automatique de gamme. C'est pourquoi nous utiliserons le circuit à gamme automatique, représenté sur la **figure 5**. Les relais sont absolument indispensables – les diodes sont normalement intégrées au convertisseur, elles sont représentées ici à titre d'éclaircissement.

Pour la « conception » du circuit, un porte-mine équipé d'une gomme est bien pratique pour corriger au vol les suppositions fausses. Le circuit est très simple : les deux relais sélectionnent laquelle des deux tensions arrive au convertisseur. La résistance série  $R_p$  limite le courant circulant dans les deux diodes en cas de surtension. Aussi simpliste qu'il en a l'air, ce circuit fonctionne et est utilisé dans presque tous les multimètres bas de gamme.

Comme pour tous les diviseurs de tension, le circuit fonctionne d'autant mieux que le courant d'entrée du convertisseur est plus faible. Pour cela, j'ai prévu, dans l'état actuel du projet, un amplificateur de gain unitaire.

## On se retrouve au labo !

À mon retour au labo, ma première tâche fut d'évaluer le courant d'entrée.

Pour cela, il faut équiper la carte avec une barrette de broches. Ces composants sont difficiles à manipuler avec des machines,

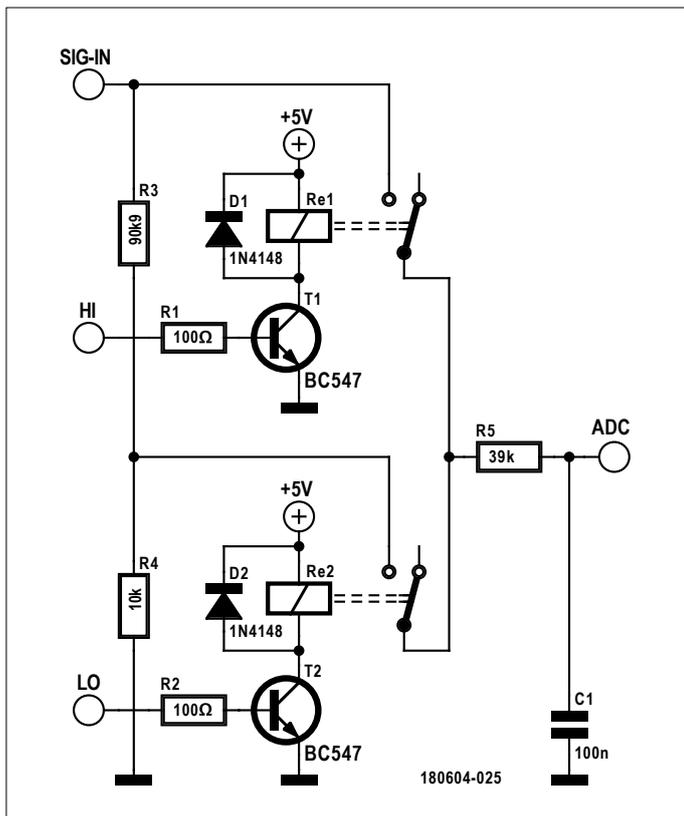


Figure 6. Avec ce circuit, l'enregistrement de données peut commencer.

logiciel, la lecture des données est simple, il suffit d'adapter `adcRead()` :

```
int adcRead()
{
    return adc1_get_raw(ADC1_CHANNEL_0);
}
```

Pour le test, nous utilisons une résistance de 50 kΩ branchée entre le 3,3 V et le CA/N. C'est là que se vérifie le vieil adage qu'un Solartron n'est pas un bon appareil de mesure – la mesure est effectuée avec un Keithley 177.

À cause de la lenteur de notre boucle principale, il faut évidemment prendre nos mesures avec précaution – l'affichage de 0,010 sur la gamme 20 μA montre toutefois qu'il ne circule aucun courant significatif. La stabilisation requiert d'installer un condensateur entre ADC et la masse, et de neutraliser les pics apparaissant au cours de la mesure. Sur l'écran, on voit en tout cas que l'enregistreur réagit au changement d'état de l'entrée.

### On rassemble tout !

La dernière tâche concerne l'atténuateur. Pour cela, nous commençons par vérifier que l'ESP32 est capable d'alimenter les relais utilisés. Connectons simplement la bobine entre le 5 V et la masse – pour les relais utilisés par l'auteur, des SRD-05VDC classiques de Songle, cela fonctionne sans problème. Armés de cette certitude, nous sommes prêts pour la réalisation du dernier circuit. La **figure 6** montre le circuit « final » de l'auteur. Techniquement, c'est loin du niveau aérospatial ; toutefois il ne faut pas omettre les diodes de protection devant les transistors à signaux faibles qui, sinon, lâcheraient au bout de quelque temps.

Il est particulièrement important de respecter la règle « ouverture avant fermeture ». Les relais sont des organes de commutation à forte inertie, qui demandent du temps pour s'ouvrir.

```
char currentState=232;
void setAttenuators(char _high)
{
    if(currentState==_high)return;
    digitalWrite(16, 0);
    digitalWrite(5, 0);
    delay(25);
```

La configuration des nouveaux réglages de l'atténuateur n'a lieu qu'ensuite. Grâce à sa mémorisation de l'état en cours, la fonction élimine des commutations inutiles.

```
if(_high==1){
    digitalWrite(5, 1);
}
else{
    digitalWrite(16, 1);
}
currentState=_high;
}
```

Pour assurer une initialisation correcte, la variable globale `currentState` contient une valeur initiale invalide qui force un cycle de commutation. Dans `setup()`, la configuration s'effectue selon le modèle suivant :

**Les concentrateurs protègent l'ordinateur !**

Il est fortement recommandé de ne pas connecter les cartes de test directement à une station de travail. Passer par un concentrateur (*hub*) à alimentation propre de bonne qualité prolonge substantiellement la vie des cartes mères.

c'est pourquoi cette tâche incombe à l'utilisateur sur pratiquement toutes les cartes d'évaluation.

Pour la mise en service, il faut commencer par se décider pour un convertisseur A/N. L'API CAN du noyau Arduino pour l'ESP32 est un peu sommaire, c'est pourquoi nous allons inclure la bibliothèque complète :

```
#include <esp_adc_cal.h>
...

```

Ensuite, il faut un peu de configuration pour un réglage correct de l'entrée :

```
void setup() {
    ...
    adc1_config_width(ADC_WIDTH_BIT_12);
    adc1_config_channel_atten(ADC1_CHANNEL_0,
    ADC_ATTEN_DB_11); //ADC_ATTEN_DB_11 = 0-3,6V
}
```

Ici, je me décide pour l'entrée ADC0, qui est reliée à la broche SVP sur notre carte. Grâce à la conception modulaire du

```
void setup() {
  . . .

  pinMode(16, OUTPUT);
  pinMode(5, OUTPUT);
  setAttenuators(1);
}
```

Notre programme n'active plus l'afficheur lorsque le relais est branché, même si l'on confie l'alimentation de 5 V à une alimentation externe de laboratoire. La raison en est que l'écran communique avec le contrôleur via un groupe de lignes de données également accessibles sur le connecteur. Dans les expérimentations de l'auteur, le système fonctionne mieux en utilisant les broches 16 et 2 (au lieu de la broche 5 comme dans le premier essai ci-dessus).

Ensuite, nous devons réaliser la commutation entre les deux sources. Voici une version naïve de la méthode :

```
int adcRead()
{
  int val;
  val = adc1_get_raw(ADC1_CHANNEL_0);
  if(currentState==0 && val > 3600)
  {
    setAttenuators(1);
  }
  else if(currentState==1 && val < 200)
  {
    setAttenuators(0);
  }

  return val;
}
```

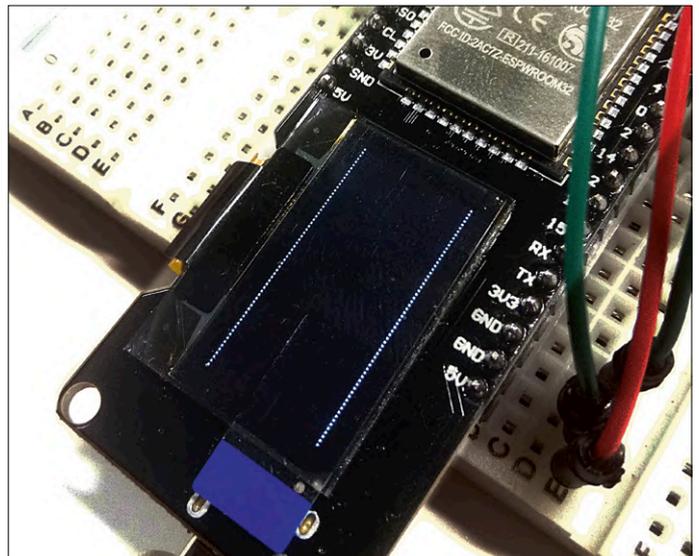
L'exécution de ce programme provoque un battement assourdissant des relais. Sur l'afficheur de la carte apparaît le résultat de la **figure 7**.

Pour résoudre ce problème, il faut commencer par régler les seuils de commutation :

```
int adcRead()
{
  int val;
  val = adc1_get_raw(ADC1_CHANNEL_0);
  if(currentState==0 && val > 3600)
  {
    setAttenuators(1);
  }
  else if(currentState==1 && val < 5)
  {
    setAttenuators(0);
  }

  return val;
}
```

À cause de l'inertie de commutation des relais, il est recommandé d'insérer un délai supplémentaire à la fin du processus de commutation :



Ici, il manque un peu d'hystérésis !

```
char currentState=232;
void setAttenuators(char _high)
{
  if(currentState==_high) return;
  digitalWrite(16, 0);
  digitalWrite(0, 0);
  delay(25);
  if(_high==1){
    digitalWrite(0, 1);
  }
  else{
    digitalWrite(16, 1);
  }
  delay(300);
  currentState=_high;
}
```

Nous voici avec une version du programme prête à l'emploi. Essayez de connecter l'entrée au 3,3 V pour provoquer un cycle de commutation des relais.

### Possibilités d'extension

Il existe une foule de possibilités d'amélioration de notre petit enregistreur. On pourrait par exemple modifier les valeurs des résistances pour éviter les domaines de non-linéarité ou de bruit intense du CA/N. Même dans le domaine du logiciel, il y aurait à faire – peut-être y aura-t-il bientôt un nouvel article pour apprendre d'autres trucs à notre petit enregistreur.

Vous trouverez un banc d'essai de cette carte en ligne : [www.elektormagazine.fr/news/mon-voyage-dans-le-nuage-27-wemos-lolin-carte-esp32-oled](http://www.elektormagazine.fr/news/mon-voyage-dans-le-nuage-27-wemos-lolin-carte-esp32-oled)

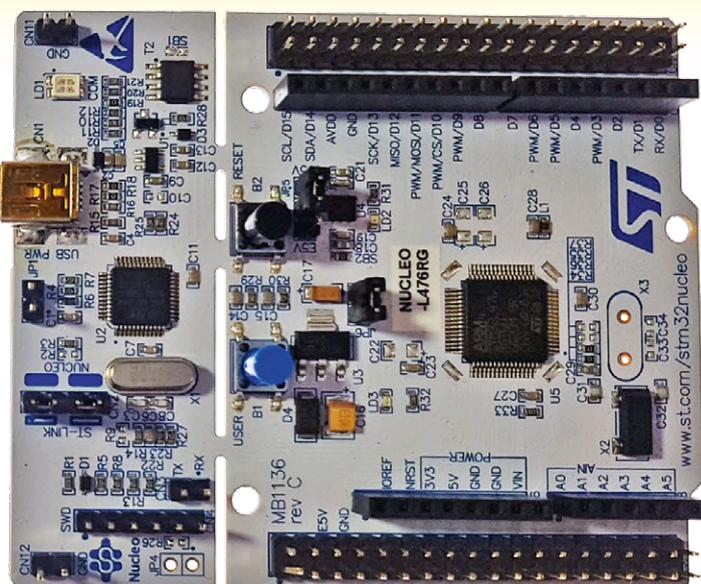

@ WWW.ELEKTOR.FR  
→ Module d'affichage OLED à EPS32 Lolin de Wemos  
[www.elektor.fr/wemos-lolin-esp32-oled-module-with-wifi](http://www.elektor.fr/wemos-lolin-esp32-oled-module-with-wifi)

# ST-Nucleo-L476RG avec STM32CubeMX et Keil MDK 5

Viacheslav Gromov

Commençons par un coup d'œil au microcontrôleur ARM Cortex-M4 de la série à ultra-basse consommation STM32L476xx à 64 pattes de ST implanté sur la carte Nucleo. Avec une fréquence d'horloge pouvant atteindre 80 MHz, une unité d'arithmétique flottante (FPU), un canal d'accès direct à la mémoire (DMA) standard ARM-Cortex-M, 1 Mo de mémoire flash et 128 Ko de mémoire SRAM, il y a suffisamment de puissance et de capacité, même pour de gros projets. En plus des fonctions d'économie d'énergie, les quatre filtres numériques intégrés, les 24 canaux tactiles et le pilote LCD parlent d'eux-mêmes. Il convient également de mentionner, outre l'interface USB, les 19 interfaces supplémentaires, les 16 minuteries et une grosse unité analogique (**fig. 1**). La compatibilité avec le 5 V de certaines broches d'entrées/sorties s'avère souvent être un avantage lors de la migration depuis des contrôleurs à 8 bits standard.

L'avantage de la famille très diversifiée des cartes Nucleo à 64 broches est le débogueur *ST-Link / V2-1* entièrement



(Photo : Viacheslav Gromov)

intégré, y compris une interface série virtuelle. Le débogueur peut également être connecté à d'autres microcontrôleurs (MCU) via l'interface SWD s'il est séparé de la MCU de la carte (le cas échéant, en cassant la partie de carte correspondante). Sans oublier, bien sûr, un périphérique matériel de base pour l'utilisateur, composé d'une LED et d'un bouton. À cela s'ajoute la conception avec une connectique à la Arduino Uno et des connecteurs Morpho (toutes les broches du microcontrôleur sont routées). Au besoin, des *shields* ST peuvent également être enfilés sur ces derniers, ce qui peut s'avérer utile, en particulier pour des prototypes à bas prix dotés de capteurs sophistiqués. Et pour l'affectation des broches et la circuiterie interne, cela vaut la peine de se plonger dans le manuel utilisateur UM1724 [4].

Si vous souhaitez en savoir plus sur la carte, nous vous recommandons la plate-forme en ligne conviviale *Mbed*, pour laquelle la carte se présente comme un simple périphérique de mémoire de masse USB. Comme nous voulons exploiter au mieux les capacités du microcontrôleur et utiliser les outils C professionnels mis à disposition par ST, nous allons directement commencer avec CubeMX et Keil. C'est pourquoi la description suivante présuppose quelques connaissances en microcontrôleurs et en logiciel.

Puisque, pour la démonstration du développement d'un petit projet pour utilisateurs avancés, il nous faut quelques outils logiciels, le mieux est que dès maintenant vous téléchargez et installez les dernières versions gratuites de *STM32CubeMX* [6] et de *Keil MDK-5-Lite* [5]. Pour les deux outils, assurez-vous que les paquets logiciels nécessaires pour la série STM32L4 sont bien présents.

## Configuration de base avec CubeMX

Nous pouvons dès lors commencer le projet. Le but de l'exer-

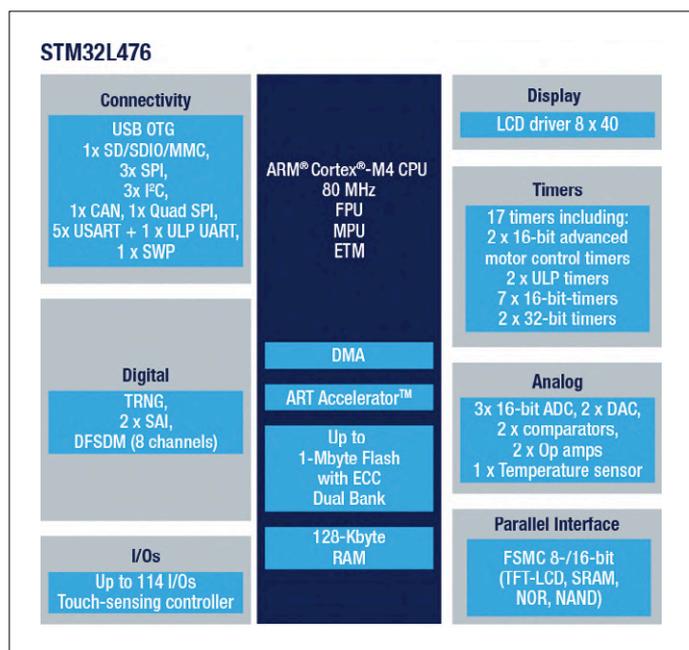


Figure 1. La structure générale du STM32L476-MCU sous forme de diagramme fonctionnel (source : ST).

cice est de transmettre au PC l'heure courante donnée par l'horloge en temps réel du microcontrôleur à l'aide de la passerelle ST-Link via l'interface UART intégrée lorsqu'on appuie sur le bouton B1 de la carte. Cette fonction doit ensuite être désactivée pendant une seconde pour que l'affichage soit rafraîchi toutes les secondes quand on maintient le bouton enfoncé. La LED LD2 doit être allumée pendant cette période de blocage.

Comme nous n'utiliserons que le matériel disponible sur la carte, commençons tout de suite avec le logiciel. STMicroelectronics met à disposition le STM32CubeMX, un outil graphique multiplateforme de création de projet, comprenant toutes les bibliothèques de base et les configurations de périphériques, c'est notre premier choix.

Pour démarrer, il faut créer dans CubeMX un nouveau projet sous la rubrique **File**. Dans la fenêtre suivante, vous devez choisir en haut l'onglet **Board Selector** pour faire défiler les cartes jusqu'à celle avec la référence **NUCLEO-L476RG**, la sélectionner et confirmer ce choix en haut à droite avec **Start Project**. Si nécessaire, vous pouvez accéder à la documentation PDF associée à la carte choisie sous **Docs & Resources** dans le bandeau supérieur.

Cette opération est suivie d'un message demandant si les modules de base de la carte choisie doivent d'ores et déjà être chargés, question à laquelle vous devez répondre **Yes** dans tous les cas. Si vous passez par le **Board Selector**, c'est le plus souvent le bon choix pour faciliter les étapes suivantes. Parce que, contrairement au **MCU Selector**, cela configure l'ensemble des périphériques présents sur la carte, tels que, par exemple, l'UART, le bouton et la LED. Après ce message, l'interface utilisateur habituelle de CubeMX s'affiche, vous permettant, selon l'onglet, de consulter à droite l'affectation des broches ou bien les modules sélectionnés. À gauche se trouvent tous les périphériques et autres modules à microcontrôleurs disponibles qu'on peut sélectionner et configurer. Comme – vous pouvez le constater – la LED, le bouton et l'UART sont déjà préconfigurés, seule l'horloge en temps réel (RTC) manque encore pour notre projet. On la trouve à gauche sous l'onglet **Timers**. Une fois la RTC choisie, apparaissent deux fenêtres pour **Mode** et **Configuration**. Si sous **Mode** vous cochez la case **Activate Clock Source**, cela active la RTC dans le logiciel, et les bibliothèques correspondantes sont importées dans le projet au moment de la génération. Sous **Configuration**, il n'y a dans ce cas (rare) rien à changer si c'est le format « 24 heures » qui est choisi.

Nous en avons déjà terminé avec les modules. Avant d'ouvrir l'onglet **Project Manager**, consultez une fois encore **Clock Configuration** et **Tools**. Dans **Project Manager**, vous devez

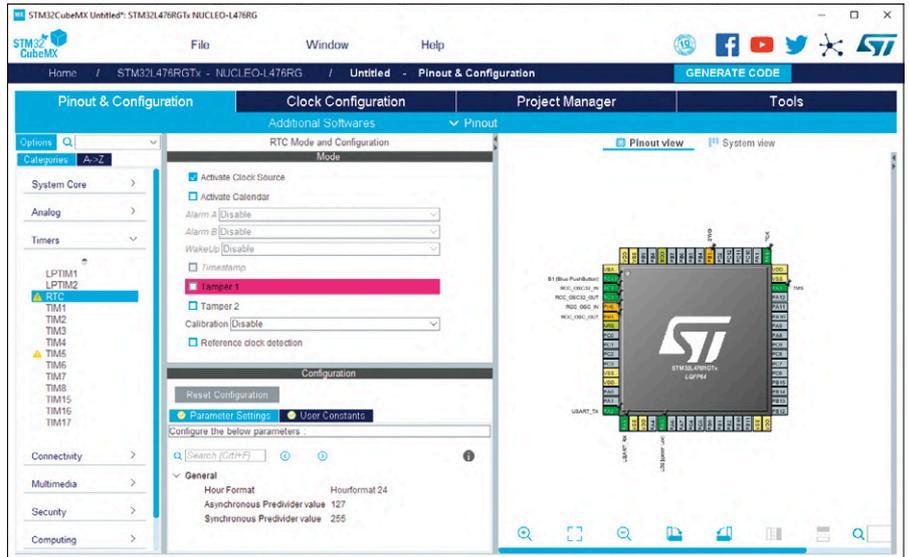


Figure 2. L'aspect habituel de l'interface utilisateur STM32CubeMX est représenté ici avec le paramétrage pour l'horloge en temps réel.

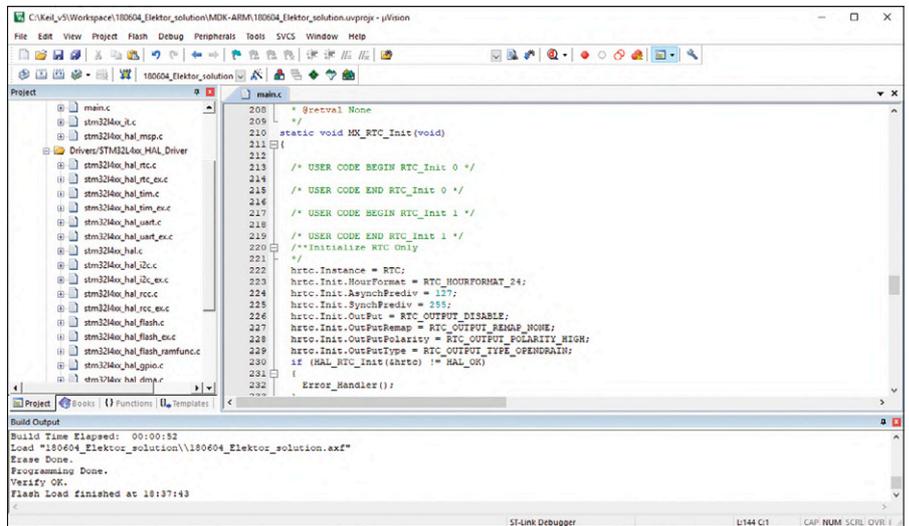


Figure 3. L'environnement de développement Keil affiche aussi la partie des bibliothèques de la couche d'abstraction du matériel (HAL) utilisée dans le projet.

donner au projet un nom approprié, indiquer le chemin de l'espace de stockage (de préférence l'espace de travail de votre EDI Keil), et choisir la chaîne d'outils MDK-ARM V5 (fig. 2). La structure de projet **Basic** est suffisante pour nos besoins. Vérifiez encore une fois les réglages sous **Code Generator** et **Advanced Settings** à gauche avant de lancer la création du projet avec les paramètres souhaités avec **GENERATE CODE** en haut à droite et de quitter CubeMX.

### Code avec Keil MDK

Après la production du code, vous recevrez immédiatement un message qui vous donne la possibilité d'ouvrir directement le projet créé dans le MDK (kit de développement pour microcontrôleurs) Keil avec l'option **Open Project**. Sinon, vous devrez ouvrir manuellement le projet dans le dossier précédemment choisi.

Une fois que vous avez ouvert Keil, vous pouvez vous familiariser avec la structure de base du projet créé. Avant tout,

### Listage 1.

```
/* USER CODE BEGIN 2 */
time.Hours = 15; //15:30:00
time.Minutes = 30;
time.Seconds = 0;
HAL_RTC_SetTime(&hrtc, &time, RTC_FORMAT_BIN); //mettre la RTC à l'heure "time"
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
if(!HAL_GPIO_ReadPin(B1_GPIO_Port, B1_Pin)) continue; //seulement si le bouton S1 est enfoncé
HAL_GPIO_WritePin(LD2_GPIO_Port, GPIO_PIN_5, GPIO_PIN_SET); //allumer la LED LD2
HAL_RTC_GetTime(&hrtc, &time, RTC_FORMAT_BIN); //lire heure dans RTC
HAL_RTC_GetDate(&hrtc, &date, RTC_FORMAT_BIN); //lire date dans RTC (non utilisée ici)
HAL_UART_Transmit(&huart2, time_word, 8, HAL_UART_TIMEOUT_VALUE); //transmettre "\r\nTime: " via UART2
Transmit_dec(time.Hours); //convertir et transmettre heures via UART2
HAL_UART_Transmit(&huart2, &time_word[6], 1, HAL_UART_TIMEOUT_VALUE); //transmettre ":" via UART2
Transmit_dec(time.Minutes); //convertir et transmettre minutes via UART2
HAL_UART_Transmit(&huart2, &time_word[6], 1, HAL_UART_TIMEOUT_VALUE); //transmettre ":" via UART2
Transmit_dec(time.Seconds); //convertir et transmettre secondes via UART2
HAL_Delay(1000); //délai de 1 s
HAL_GPIO_WritePin(LD2_GPIO_Port, GPIO_PIN_5, GPIO_PIN_RESET); //éteindre la LED LD2
}
/* USER CODE END WHILE */
```

examinez les bibliothèques de manière détaillée (**fig. 3**). Les nombreux commentaires contribuent à la bonne compréhension. Encore plus utile, le document de quelque deux mille pages (!) UM1884 [7], qui décrit de manière complète et compréhensible chaque fonction, chaque structure et bien d'autres choses. Pour notre projet, nous n'avons qu'à éditer le fichier principal. Un rapide coup d'œil révèle que toutes les fonctions de configuration ou d'initialisation des composants sélectionnés du microcontrôleur sont déjà incluses.

Avant d'inclure notre projet dans la fonction principale, nous devons déclarer ou initialiser un tableau contenant les caractères ASCII pour la chaîne « Time: », précédés d'un saut de ligne, ainsi que les deux structures pour l'heure et la date de l'horloge en temps réel. Veuillez insérer ces trois lignes immédiatement après le commentaire `USER CODE BEGIN 0` :

```
uint8_t time_word[] = ;
//ASCII for "\r\nTime: "
RTC_TimeTypeDef time;
//declare structures for time and date
RTC_DateTypeDef date;
```

À l'avenir, veuillez prendre bien garde à ce genre de commentaire, qui protège votre code de l'écrasement lors de la modification du projet dans CubeMX.

Mais passons maintenant à la fonction principale, qui devrait ressembler à celle du **listage 1**. Avant la boucle infinie, les seules opérations sont de renseigner les structures déclarées précédemment avec l'heure courante (que vous pouvez adapter en conséquence) et de les passer à l'horloge en temps réel. Dans la boucle sans fin, une pression sur le bouton provoque l'allumage de la LED et la lecture de l'heure et de la date.

### Listage 2.

```
/* USER CODE BEGIN 4 */
static void Transmit_dec(uint8_t i)
{
uint8_t buff[] = ; //allouer et initialiser un tampon
buff[0] += i / 10; //convertir et stocker le 1er chiffre
buff[1] += i % 10; //convertir et stocker le 2e chiffre
HAL_UART_Transmit(&huart2, buff, 2, HAL_UART_TIMEOUT_VALUE); //transmettre le tampon via l'UART
}
/* USER CODE END 4 */
```

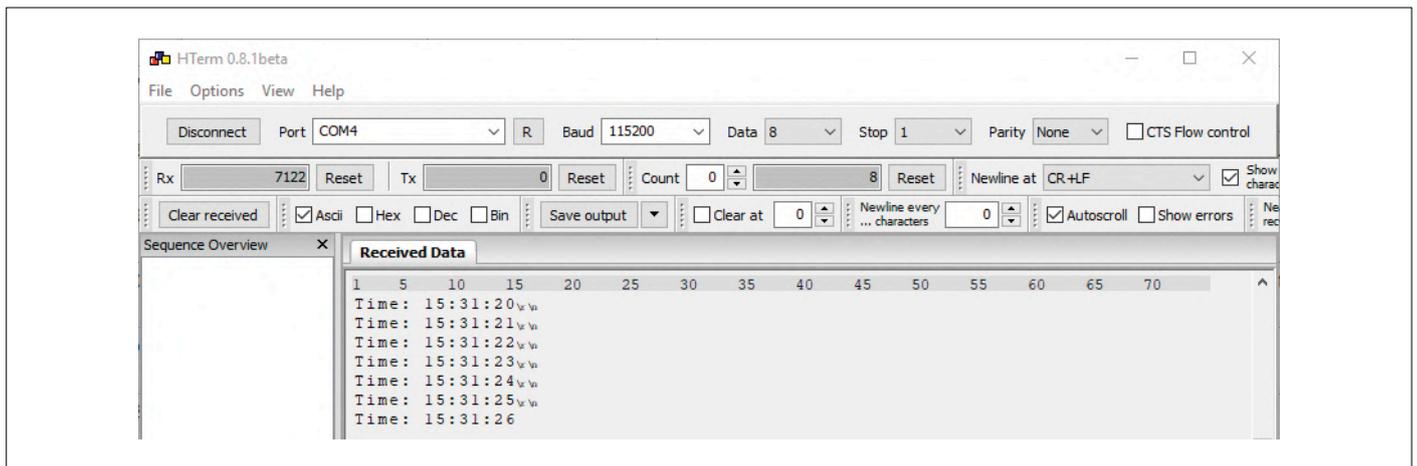


Figure 4. Le défilé de l'heure quand on appuie sur le bouton. On reconnaît aussi le paramétrage de l'émulateur de terminal.

Bien que seule l'heure soit utilisée dans notre projet, sa lecture doit être immédiatement suivie de celle de la date, car la fonction s'attend à ces deux opérations. Autrement, l'horloge s'arrêterait littéralement.

Dans la suite du programme, l'heure lue est toujours transmise par une fonction séparée via l'UART sous la forme de nombres à deux chiffres. Le double point nécessaire entre les nombres est simplement prélevé dans la chaîne « Time: ». L'armement du délai d'attente d'une seconde (fonction du *timer Systick*) et l'extinction de la LED embarquée terminent notre fonction principale.

`Transmit_dec()` est une fonction séparée afin de montrer la conversion binaire-ASCII d'un nombre à deux chiffres, qu'il faut insérer au même endroit que dans le **listage 2**. Ici, on exploite les effets de la division sans conversion de type, avant d'envoyer les deux caractères via l'UART après la conversion. Il est préférable d'insérer le prototype de cette fonction immédiatement après `USER CODE BEGIN PFP` :

```
static void Transmit_dec(uint8_t i);
```

Le projet est maintenant prêt. Pour le tester, vous pouvez le compiler avec les fonctions Keil appropriées (*Build*) et le charger sur le microcontrôleur (et le réinitialiser manuellement !) ou procéder à son débogage. Si aucun des deux ne fonctionne malgré des paramètres corrects, cela est généralement dû à l'absence du pilote ST-Link, que vous pouvez télécharger depuis ST. Si vous rencontrez des problèmes avec CubeMX ou

le code, le projet terminé est également disponible gratuitement sous [8]. Si tout fonctionne, vous pouvez utiliser n'importe quel émulateur de terminal (paramétré comme sur la **figure 4**) pour lire l'heure lorsque vous appuyez sur B1. Félicitations, vous venez de finir le premier projet avec succès ! Afin d'approfondir le sujet, nous vous recommandons de modifier le projet pour que la date soit également affichée. En outre, vous pouvez également utiliser les interruptions de l'horloge, car l'exécution du code prend en réalité un peu plus longtemps qu'une seconde. N'oubliez pas que pour les programmeurs avancés, l'apprentissage par la pratique est aussi important que pour les débutants, mais cela ne vous épargne pas l'étude de la documentation. Nous vous souhaitons beaucoup de succès dans le monde du 32 bits professionnel ! ◀

(180604-B-04 - version française : Helmut Müller)

**@ WWW.ELEKTOR.FR**

→ Carte STM32 Nucleo L476RG  
[www.elektor.fr/stm32-nucleo-l476rg-board](http://www.elektor.fr/stm32-nucleo-l476rg-board)

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

## Liens

- [1] 'Code Reading' : <https://www.amazon.com/Code-Reading-Open-Source-Perspective/dp/0201799405>
- [2] Amplificateurs de mesure : <https://www.renesas.com/eu/en/www/doc/application-note/an028.pdf>
- [3] La vie secrète des diodes (en allemand) : <https://holzleitner.com/el/picoampere/index-de.html>
- [4] Manuel de l'utilisateur de l'UM1724 : <http://bit.ly/2iAbnxf>
- [5] Générateur de code d'initialisation STM32Cube : <https://www.st.com/en/development-tools/stm32cubemx.html>
- [6] Kit de développement pour microcontrôleurs de Keil : <http://www2.keil.com/mdk5/>
- [7] Manuel de l'utilisateur de l'UM1884 : <http://bit.ly/2X3CNzF>
- [8] Page de l'article : <http://www.elektormagazine.fr/180604-B-04>



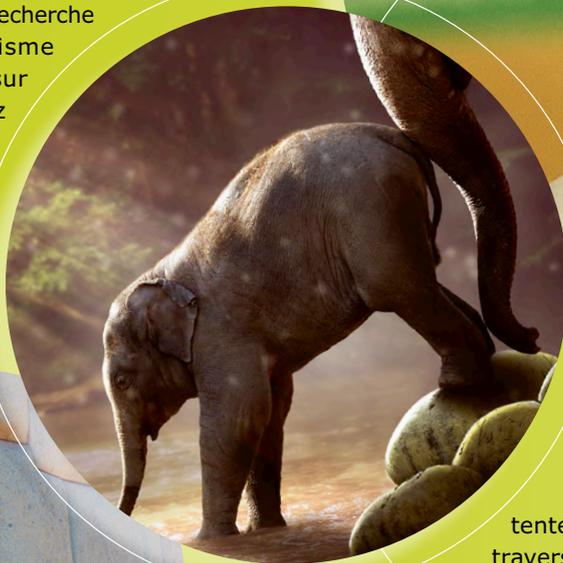
# vol tous azimuts

**l'électronique par monts, maux et merveilles**

Clemens Valens (labo d'Elektor)

## Qui a inventé le bouton-poussoir ?

Placez n'importe qui devant une machine, et il ou elle se mettra aussitôt à en presser les boutons. Mettez cette même personne devant une table à induction, elle se mettra rapidement à en tapoter la surface, quand bien même celle-ci ne comporte aucun bouton. Lorsque quelque chose ne s'ouvre pas ou ne fonctionne pas immédiatement, nous appuyons ici ou là machinalement à la recherche de quelque mécanisme inconnu. Appuyer sur les choses est chez nous un réflexe. Mes deux chats savent apparemment pousser une porte entrebâillée, mais en réalité ils



Outre les arbres, les éléphants savent aussi pousser leurs petits dans la bonne direction.

tentent simplement de l'ouvrir en remuant leur museau à travers son ouverture. Les chats ne savent pas pousser, une thèse audacieuse que j'ose extrapoler aux chiens et aux autres animaux (à l'exception peut-être des éléphants, car il leur arrive de pousser des arbres. J'ai vu ça sur YouTube.)

L'homme est quant à lui capable de pousser les choses depuis moult générations. On peut le déduire de ces films à la Indiana Jones où le héros (rarement une héroïne) doit déplacer les pierres d'un temple inca pour accéder à un trésor (de l'or ou une gentille demoiselle en détresse). Ce simple fait prouve que les Incas avaient déjà poussé des choses et, par extension, que leurs ancêtres aussi avaient dû pousser des choses. D'ail-

Quelle pierre pousser pour ouvrir la porte du temple ?

leurs, s'ils ne l'avaient pas fait, pourquoi auraient-ils inventé la poussée ? Il me semble que les mathématiciens appellent preuve par induction ce genre de raisonnement imparable. Possédons-nous un gène de la poussée, ou s'agit-il d'un comportement acquis ? La question est difficile et je n'ai pas (encore) de réponse. Avons-nous inventé le bouton parce que nous sommes des pousseurs-nés, ou avons-nous appris à pousser parce que nous avons inventé le bouton ? Et au fait, qui a inventé le bouton ? Difficile à dire, la page que lui consacre Wikipédia n'a pas de section Histoire. Doit-on en déduire qu'aucun historien ne s'est penché sur son origine ? Que son invention restera à jamais un paragraphe vide dans l'histoire de l'humanité ? « Faire glisser » est par contre un comportement acquis très récent. Cela ne fait aucun doute. S'il avait été plus ancien, Indiana Jones aurait tenté d'accéder au trésor caché en faisant

glisser des pierres plutôt qu'en les poussant. (En passant, les égyptologues ont-ils essayé ce geste dans les pyramides pour en percer les secrets ?) La pression impérieuse sur le bouton sera-t-elle un jour réduite à un timide tapotement, et finalement remplacée par ce geste de balayage plutôt arrogant qu'est le « faire glisser » ?

P.S.

Avez-vous remarqué combien sont différents les signaux envoyés par ces deux gestes ? Appuyer sur quelque chose revient à dire « Je veux ça », tandis que faire glisser signifie l'inverse : « Pas ça ! ».

Pour pousser plus loin : Rachel Plotnick, *Power Button - A History of Pleasure, Panic, and the Politics of Pushing*, MIT Press, 2018.

## Identification biométrique avec le 555

Qui n'a jamais utilisé ce bon vieux et phénoménal « 555 », le célèbre circuit intégré polyvalent à 8 broches ? Le NE555 a été conçu en 1971 par Hans R. Camenzind et a été commercialisé en 1972 par la société Signetics (aujourd'hui filiale de NXP). Il a depuis été produit à des milliards d'exemplaires. Sa fiche technique originelle suggérait de l'utiliser comme temporisateur, ce qui semble logique pour une puce présentée comme *timer IC*.

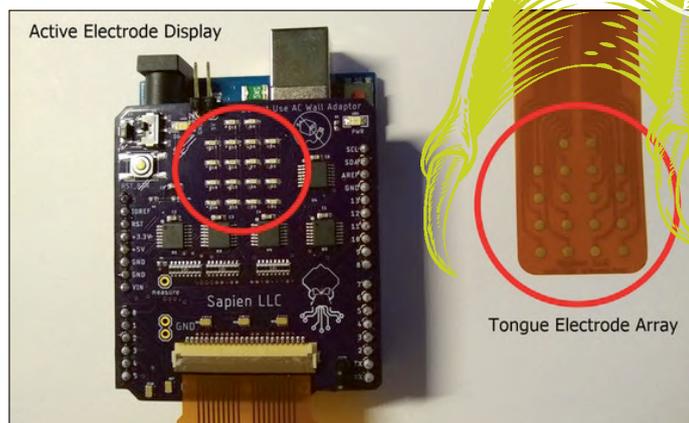
Texas Instruments a également fabriqué le 555 et l'a accompagné d'une fiche technique en 1973. Avec la mise à jour de 2014, les champs d'application ont eux aussi été enrichis : reconnaissance de l'iris et des empreintes digitales, ainsi que lecteurs RFID. Qui aurait cru que ce vénérable circuit intégré de presque 50 ans aurait un jour sa place dans des applications de pointe ?

Je prédis que le 555 sera toujours d'actualité en 2219 et recommandé pour de nombreuses applications, dont la téléportation.

<http://www.ti.com/lit/ds/symlink/ne555.pdf>



# L'appel de Cthulhu : goûtez les couleurs !



Un prototype du shield Cthulhu. (Crédit : Sapien LLC)

Cthulhu (prononcez ktleul-lou sur un ton guttural) est une entité cosmique monstrueuse imaginée en 1926 par l'écrivain H. P. Lovecraft pour sa nouvelle *L'appel de Cthulhu*. N'étant décrite qu'avec des mots, son apparence dépend de votre imaginaire. Plusieurs artistes en ont donné des représentations saisissantes, mais jusqu'ici aucun n'avait attribué à Cthulhu la forme d'une carte d'extension Arduino.

Le shield Cthulhu paraît à première vue aussi étrange que l'entité imaginée par Lovecraft puisqu'il se relie à... la langue. Voilà une interface pour le moins inhabituelle, mais après tout est-elle si improbable que ça ? Oui ? Non ? Oui. Ou peut-être que non. Ou plus sûrement : je vous laisse juger.

C'est par une nappe flexible dotée de 18 électrodes que l'utilisateur relie Cthulhu à sa langue. Le shield peut soit stimuler la langue au moyen d'impulsions électriques, soit mesurer les signaux électriques qu'elle produit.

La mesure de l'activité électrique de la langue permet à l'utilisateur de commander toutes sortes de dispositifs. Une personne handicapée pourra ainsi interagir avec son environnement, mais le shield Cthulhu peut aussi servir de troisième main là où deux ne suffisent pas.

Outre ce rôle d'interface matérielle, le shield Cthulhu est aussi un dispositif de substitution et d'augmentation sensorielle. La stimulation de la langue permet par exemple de goûter des couleurs ou de percevoir la direction donnée par un système de navigation (en mode mains libres). La transformation d'une information visuelle ou sonore en retour tactile peut de même aider un non-voyant ou un malentendant. C'est

aussi une façon d'ajouter de nouveaux sens à l'humain et d'augmenter la sensibilité de ceux dont il est déjà pourvu.

Destiné à la création de dispositif de substitution ou d'augmentation sensorielle (des interfaces langue-machine), le shield Cthulhu est à code source ouvert. Son code est sur GitHub.

<https://github.com/SapienLLCdev/Cthulhu/>

Voici à quoi vous ressemblez une fois le shield relié à votre langue. (Crédit : Sapien LLC)



## Outil de labo indispensable

Placer une pointe de touche sur la broche d'un circuit intégré, puis tourner la tête pour lire l'écran d'un instrument de mesure tout en veillant à ne pas mettre en court-circuit la broche et sa voisine relève de l'art du Mikado. Surtout quand on s'aperçoit que l'on n'a pas sélectionné le bon calibre de ce maudit instrument placé hors de portée. La mission s'avère quasiment impossible avec deux pointes, et que dire de trois ou quatre. Alors on soude des fils sur la puce, on y relie les pointes de touche, et une fois le travail terminé on dessoude le tout en priant qu'aucun court-circuit ne se soit produit. Mais ça, c'était avant PCBite 2.0, l'outil de mesure mains libres. Ses bras aimantés flexibles peuvent immobiliser jusqu'à quatre sondes de mesure. Fermement et sans arrimage mécanique.

<https://www.elektor.fr/pcbite-kit-incl-4-pcbite-probes-and-test-wires>



## Chéri(e), j'ai rétréci la maison

La taille des objets ne cesse de diminuer. Songez par exemple à ce qu'étaient les téléphones mobiles des années 1980 par rapport à nos ordiphones modernes (ce n'est à vrai dire pas le meilleur exemple car les ordiphones actuels tendent à grossir). Encore plus frappant à cet égard est la micro-maison assemblée par le laboratoire de recherche français Femto-ST (en 2016, le temps passe...)

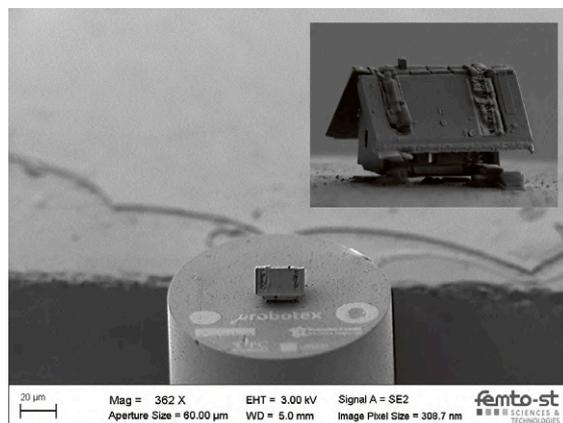


Photo courtesy Femto-ST

Selon le lien ci-dessous, la micro-maison ne mesure que  $10 \times 20 \mu\text{m}$  (ce qui est curieux pour une maison, qui possède habituellement une troisième dimension) et a été réalisée, je cite, par pliage et assemblage d'éléments découpés dans une micro-membrane de silice de  $0,7 \mu\text{m}$  d'épaisseur. Lesdits éléments ont été collés entre eux avec du naphthalène, la cheminée étant une goutte de cette même naphthalène. La micro-maison, qui repose sur l'extrémité d'une fibre optique de  $80 \mu\text{m}$  de diamètre (elle a la fibre !) a été construite pour démontrer les possibilités de Robotex, une plateforme dédiée à la caractérisation, la manipulation et l'assemblage de micro et nano-systèmes ( $< 10 \mu\text{m}$ ).

Si seulement ce labo parvenait à rétrécir aussi les humains. La surpopulation ne serait plus un problème.

<http://projects.femto-st.fr/microrobotex/fr/realisations>



productronica  
fast forward  
the start-up platform  
powered by Elektor

## Placez votre startup en pole position pour le concours Fast Forward productronica 2019

équipe Elektor Fast Forward



Crédit des photos : Jens Nickel et Clemens Valens  
@ electronica Fast Forward 2018.

Il est plutôt amusant de se remémorer que les débuts du concours Fast Forward, la rampe de lancement de startups animée par Elektor, ont été très similaires à ceux d'une jeune pousse des plus discrètes (en réalité, à peine quelques idées notées sur un coin de table). Après deux éditions pleinement réussies en 2016 et 2018 sur le salon *electronica*, la plateforme Fast Forward n'a plus à démontrer sa dynamique résolument novatrice, de nouveau mise en place sous l'égide de Messe München au cœur du salon *productronica 2019* du 12 au 15 novembre. Fière des réalisations accomplies depuis 2016, l'équipe organisatrice du concours a la sincère conviction que son expérience sans équivalent peut motiver votre startup, porteuse d'une idée remarquable et prête à concourir avec d'autres, à élargir son public et à lancer pour de bon sa dynamique marketing et commerciale.

Les lecteurs du magazine *Elektor* et du magazine *Elektor Industry* (disponible en anglais et allemand uniquement) gravitent dans le monde de l'électronique que ce soit en tant qu'amateurs éclairés ou professionnels. Ces deux magazines partagent un intérêt commun : aider les startups et autres entreprises en pleine expansion à atteindre un niveau professionnel ainsi qu'à trouver des partenaires pour financer, produire et vendre leur idée.

« C'est génial / super / top / prometteur ! Est-ce qu'Elektor peut faire ça pour moi aussi ? »

Bien sûr ! Le bolide d'Elektor qui rend cela possible à une vitesse vertigineuse et avec un impact extraordinaire dans le monde de l'électronique s'appelle **Fast Forward**. Il est sur le circuit depuis 2016, date à laquelle l'équipe FFWD a fait forte impression lors du prestigieux salon *electronica* de Munich.

### Déroulement du concours

- Les candidats remplissent un formulaire d'inscription en ligne avant le 1<sup>er</sup> octobre 2019.
- Le jury examine toutes les candidatures et invite les candidats les plus prometteurs à fournir des informations plus détaillées.
- Le jury sélectionne un groupe de candidats qui auront la chance de défendre et de présenter leur start-up lors du salon *productronica 2019*. Ces finalistes disposent d'un espace de présentation sur la plateforme dédiée aux startups sur le salon *productronica 2019*.
- En novembre 2019, sur le salon *productronica*, les finalistes présentent leur startup sur la plateforme dédiée aux startups. Le jury sélectionne les trois gagnants pendant le salon.



- Présenter votre produit ou votre service novateur sur le stand Fast Forward lors du salon *productronica 2019*.
- Concourir aux côtés d'autres finalistes pour gagner une campagne marketing d'une valeur de 25.000 €, réalisée conjointement avec Elektor.
- Attirer l'attention de partenaires potentiels, d'investisseurs et clients venus du monde entier.
- Faire connaître votre startup aux quelque 250.000 lecteurs d'Elektor.

### Des équipes audacieuses

Les startups du monde de l'électronique devraient être aussi audacieuses que l'équipe d'Elektor qui organise le concours FFWD. Ne perdez pas votre temps à écouter des histoires sur des produits existants... Si vous êtes une startup avec :

- un financement existant, même s'il est faible ou virtuel ;
- des plans d'affaires et de marketing bien définis ;
- une approche professionnelle ;
- une excellente présentation ;
- de l'endurance ;
- de l'énergie à revendre ;
- une ambition sérieuse de devenir célèbre avec votre équipe et vos projets ;
- et que vous ne craignez pas le jury Fast Forward et son équipe de tournage.

alors appuyez sur le bouton *Fast Forward* (avance rapide) comme l'ont fait les gagnants des FFWD 2016 et 2018 ! Il suffit de vous inscrire à la manifestation p-FFWD à laquelle Elektor participera du 12 au 15 novembre 2019 au salon *productronica* de Munich.

### Les prix

Toutes les candidatures seront examinées et évaluées par les membres du jury. Les meilleurs candidats seront invités à défendre leur startup lors du salon *productronica 2019* et tenteront de remporter des ensembles d'outils et d'actions de promotion fournis par Elektor, d'une valeur de plusieurs milliers d'euros, ainsi qu'un stand sur le salon *productronica 2021*.

- **1<sup>er</sup> prix** : un ensemble d'outils et d'actions de promotion d'une valeur de 25.000 € fourni, par Elektor International Media, comprenant notamment un stand d'exposition sur le salon *productronica 2021*.
- **2<sup>e</sup> prix** : un ensemble d'outils et d'actions de promotion d'une valeur de 15.000 €, fourni par Elektor International Media.
- **3<sup>e</sup> prix** : un ensemble d'outils et d'actions de promotion d'une valeur de 5.000 €, fourni par Elektor International Media.

### Devenez un sponsor

*productronica* Fast Forward, la rampe de lancement de startups animée par Elektor, offre aussi une belle occasion pour les entreprises déjà établies dans le secteur de l'électronique de promouvoir leurs produits ou leurs services en s'engageant dans le sponsoring, ou en proposant une démonstration ou une présentation dans le cadre du forum Fast Forward 2019. Dans votre rôle de sponsor de l'événement, vous bénéficierez d'une campagne médiatique complète englobant entre autres promotions de l'événement sur Internet, publicité dans les magazines et bannières de pages web. Différentes offres de sponsoring vous seront proposées en fonction de vos besoins. Pour en savoir plus, contactez : [margriet.debeij@lelektor.com](mailto:margriet.debeij@lelektor.com).

L'événement a été renouvelé avec succès en 2018. Le bolide FFWD sera de nouveau sur la piste lors du salon *productronica 2019* en novembre prochain.

Vous inscrire au concours p-FFWD 2019, c'est vous donner une chance de présenter votre innovation à l'ensemble de la communauté mondiale de l'électronique. Si vous êtes sélectionné, vous aurez la chance de :

N'attendez plus, rendez-vous sur :

[www.elektormagazine.fr/p-ffwd](http://www.elektormagazine.fr/p-ffwd)

L'équipe est impatiente de vous accueillir ! ◀

190147-03

# commande pour la découpe de polystyrène

## alimentation à faible perte du fil chauffant grâce à un variateur à basse tension

Peter Grundmann

En modélisme comme en bricolage domestique, un couteau bien aiguisé fait souvent l'affaire pour la découpe de blocs de polystyrène. Mais les blocs s'effritent, et la précision qui laisse à désirer ne permet pas la découpe de formes complexes. C'est pourquoi il vaut mieux découper le polystyrène à l'aide d'un fil chauffant. Le courant nécessaire sera fourni par une commande réglable à découpage de phase sans perte.

Si vous n'appréciez guère les petites billes de polystyrène qui collent partout pendant la découpe à cause des charges électrostatiques, et que vous y êtes confronté plus d'une fois par an, alors vous devriez construire votre propre appareil de découpe. Le principe du façonnage manuel de la mousse de polystyrène sans miettes est simple : il faut de la chaleur. Plus précisément : on utilise

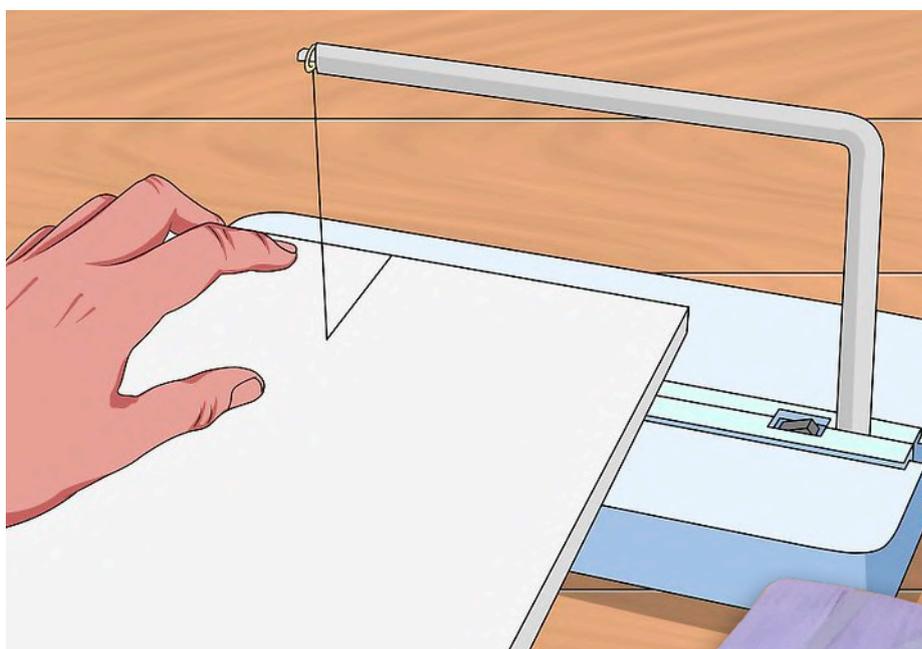
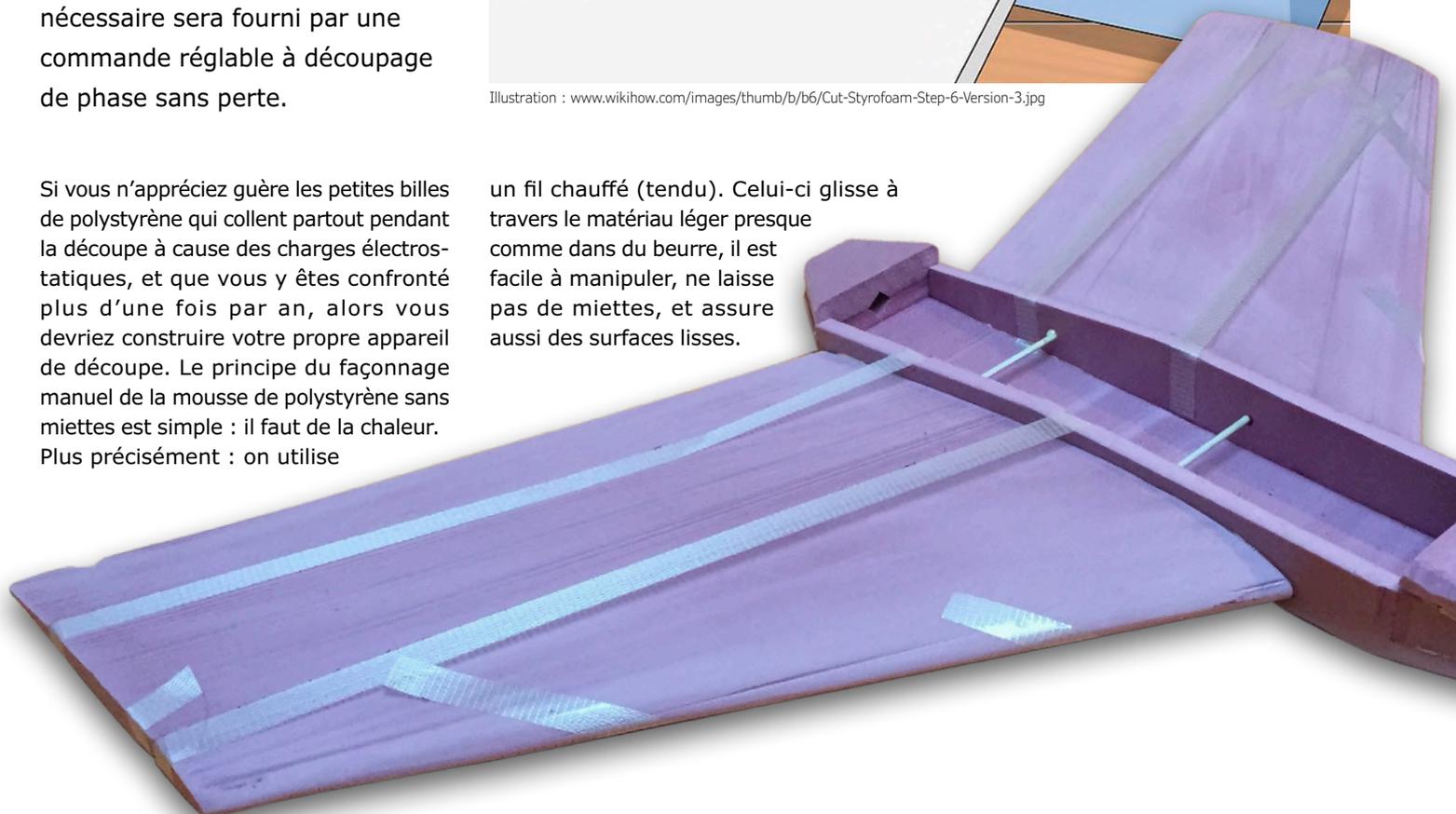


Illustration : [www.wikihow.com/images/thumb/b/b6/Cut-Styrofoam-Step-6-Version-3.jpg](http://www.wikihow.com/images/thumb/b/b6/Cut-Styrofoam-Step-6-Version-3.jpg)

un fil chauffé (tendu). Celui-ci glisse à travers le matériau léger presque comme dans du beurre, il est facile à manipuler, ne laisse pas de miettes, et assure aussi des surfaces lisses.



La circulation d'un courant dans le fil permet d'obtenir la chaleur requise. Ce courant doit être réglable afin d'ajuster l'énergie libérée dans le fil en fonction des conditions de la découpe. L'alimentation devra être à faible perte, car la puissance pourra facilement atteindre plusieurs dizaines de watts. À cet effet, un type de variateur à basse tension convient parfaitement, car il « commute » et ne présente donc qu'une faible chute de tension en mode réglage. Un tel variateur est précisément l'objet de cet article.

### Commande par découpage de phase

Il ne fait aucun doute qu'un tel fil chauffant pour la découpe du polystyrène ne peut être simplement relié à une résistance élevée alimentée directement par le secteur via un gradateur standard. Sans isolation galvanique du secteur 230 V, ce type de loisir s'apparenterait tout bonnement à la roulette russe (« La phase est-elle maintenant reliée au fil chauffant ? »). Par conséquent, un transformateur s'avère obligatoire, ce qui présente également l'avantage de pouvoir mieux ajuster la tension d'alimentation du fil dont la résistance est généralement faible. Assez discuté : la **figure 1** montre le circuit complet conçu et testé par l'auteur.

Les amateurs de circuits analogiques y trouveront leur compte : pas l'ombre d'un microcontrôleur à l'horizon ! La commutation est prise en charge par le temporisateur universel NE555, IC2, qui

est directement commandé par la fréquence du secteur. Le délai entre le passage par zéro de la fréquence du secteur et l'allumage du triac est ajusté par un potentiomètre. Le gradateur a son propre petit transformateur d'alimentation, et est ainsi complètement indépendant du transformateur qui alimente le fil chauffant. La simplicité du circuit est telle que vous pouvez, avec un minimum de soin, le réaliser sur une carte à trous.

### Le circuit

De gauche à droite : le circuit électronique est alimenté par Tr1. Un petit transformateur de 1,5 VA fait l'affaire, le circuit ne consomme que quelques milliampères. Les diodes D1 à D4 forment un pont redresseur. Comme la diode D5 isole le condensateur de filtrage C1, on observe sur R2 une demi-onde sinusoïdale à 100 Hz d'amplitude 20 V environ (car Tr1 est à peine chargé). Le transistor T1 est ainsi commandé à cette fréquence et fournit sur son collecteur un signal rectangulaire dont les flancs montent peu de temps avant le passage par zéro de la tension secteur, et descendent peu de temps après. La tension d'alimentation continue de 12 V est fournie par le régulateur IC1 qu'il est inutile de refroidir, celui-ci étant à peine chargé.

Le temporisateur IC2 est configuré en multivibrateur monostable dont la constante de temps est déterminée par C4 et la combinaison de P1, R6 et R7. Celle-ci peut être ajustée dans la gamme de 0,3 à 8 ms selon la formule  $\tau = 1,1 \times R \times C$ . C'est suffisant pour cou-

vrir à l'aide de P1 une plage de puissance allant de pratiquement 100% à <5%. IC2 est déclenché par le flanc descendant du signal provenant de T1. Immédiatement après, sa sortie sur la broche 3 devient haute. Une fois la constante de temps écoulée, ce signal passe à l'état bas. Il commande le transistor T2 via R10 et C6, et donc le triac T3 via R11, qui est ensuite débloqué lors du passage par zéro suivant. Les diodes D8 et D9 assurent une décharge rapide de C5 et C6.

Remarque à propos de la résistance R6 en parallèle sur le potentiomètre : les tolérances des potentiomètres sont souvent élevées, donc si la résistance parallèle équivalente est trop élevée, la durée d'impulsion maximale du temporisateur risque de dépasser les 10 ms d'une demi-période des 50 Hz, et donc le triac ne se déclenchera que pour une demi-période sur deux. Par conséquent, le transformateur serait chargé de manière asymétrique et la tension minimale ne se laisserait pas ajuster correctement. Donc, soit vous mesurez la largeur d'impulsion maximale à l'oscilloscope (elle devrait être  $\leq 9$  ms), soit vous déterminez la résistance parallèle équivalente (elle devrait être  $\leq 80$  k $\Omega$ ), et vous ajustez R6 si nécessaire.

### Par ailleurs...

Peter Grundmann a réalisé ses premiers prototypes à l'aide de cartes à trous. Il a en plus conçu un circuit imprimé (**fig. 2**) disponible sur la page de l'article [1] sous la forme d'un fichier pour le logiciel Tar-

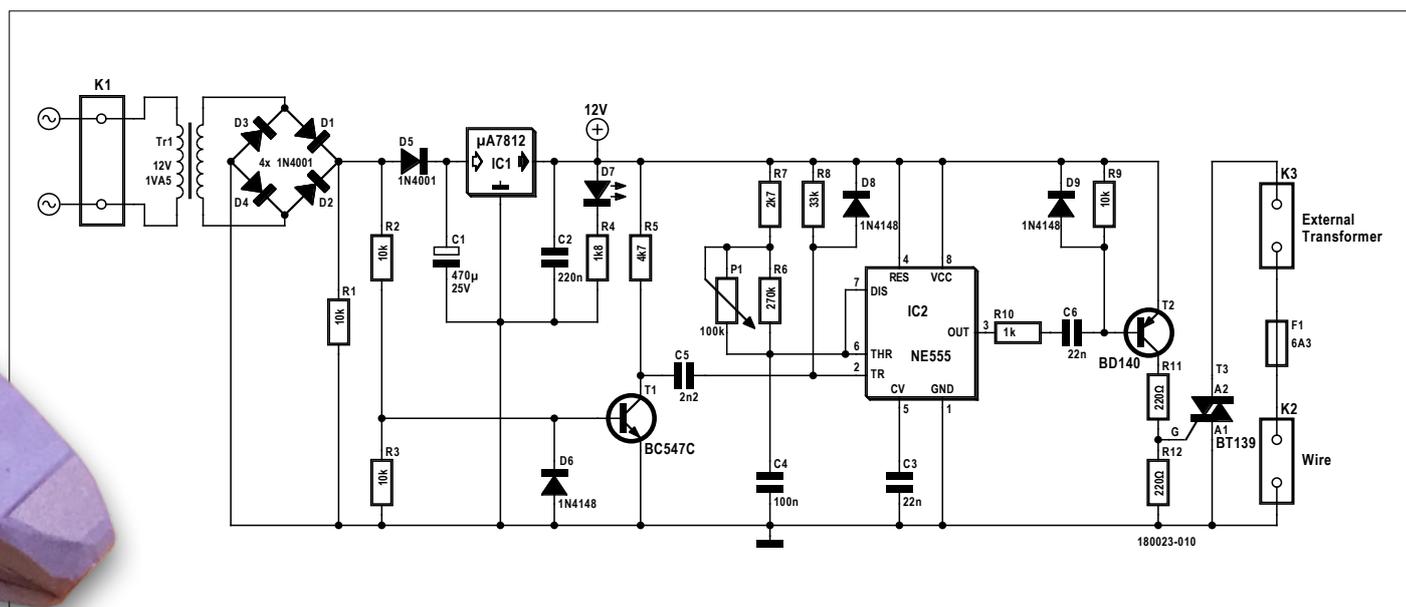


Figure 1. Le schéma du circuit à commande de phase pour la découpe de polystyrène ne nécessite pas de microcontrôleur.

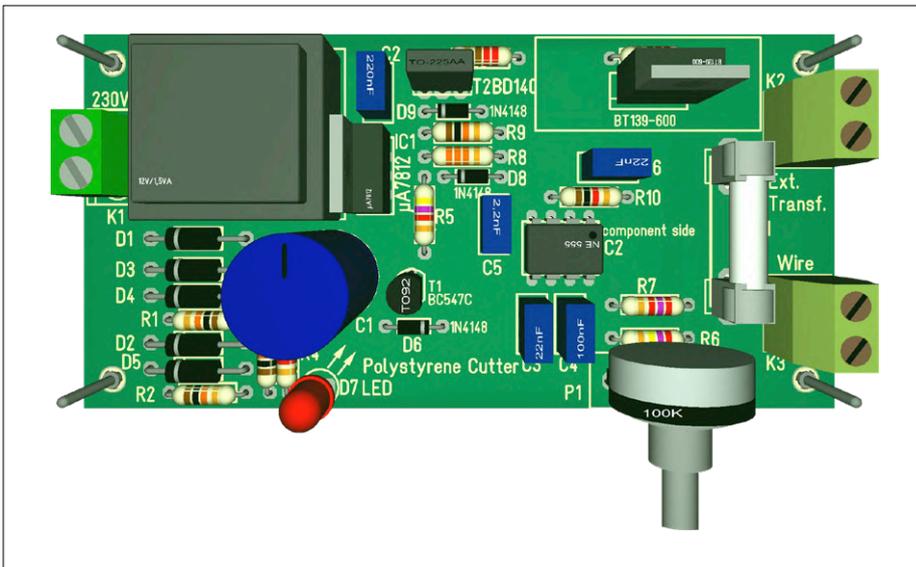


Figure 2. Le circuit imprimé équipé devrait ressembler à ceci en simulation 3D.

@ [WWW.ELEKTOR.FR](http://WWW.ELEKTOR.FR)

→ 3D Printing and CNC Fabrication with SketchUp (livre en anglais)  
[www.elektor.fr/3d-printing-and-cnc-fabrication-with-sketchup](http://www.elektor.fr/3d-printing-and-cnc-fabrication-with-sketchup)

**Lien**  
 [1] Page de l'article : [www.elektormagazine.fr/180023-04](http://www.elektormagazine.fr/180023-04)

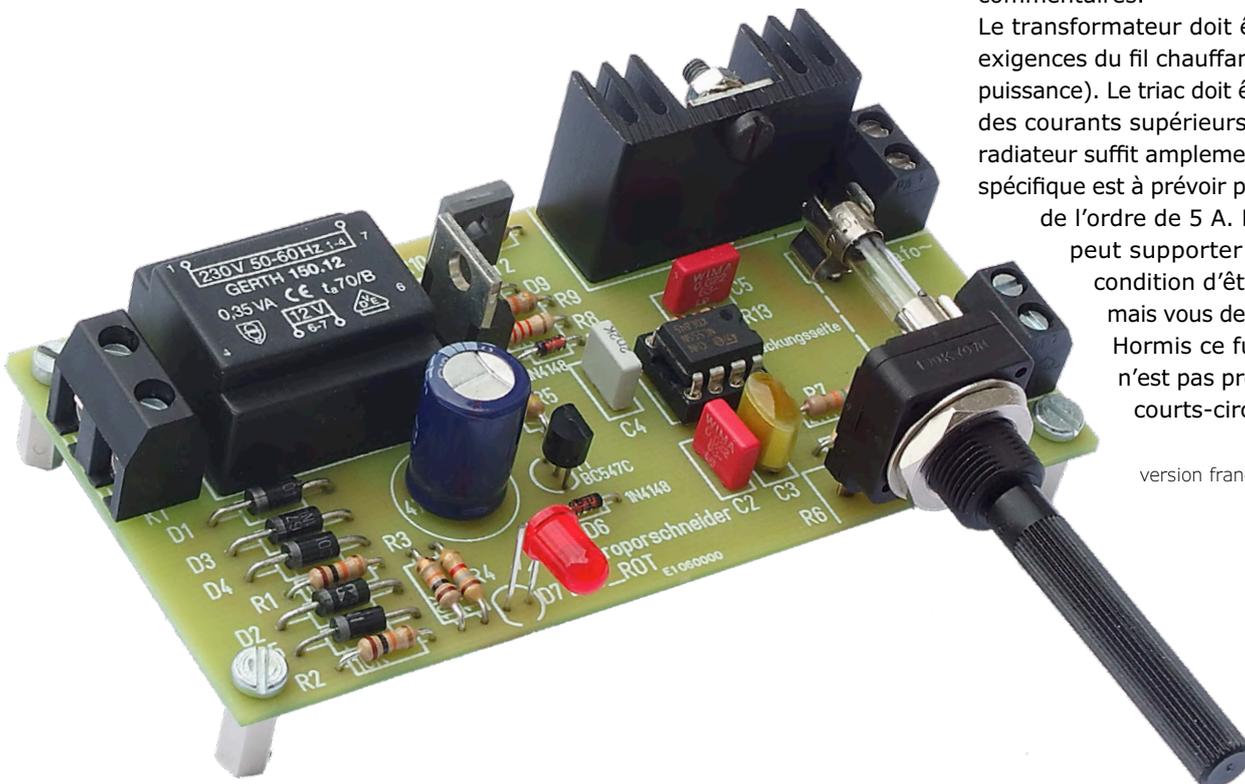


Figure 3. Le dernier prototype (sur circuit imprimé) a conduit à une modification possible de R7 (voir texte).

get3001. Vous n'avez pas besoin de la version complète : la version de démonstration de ces outils de CAO suffit amplement à cet effet. Le circuit imprimé a ensuite été garni par l'auteur (fig. 3). Il s'est avéré que le circuit fonctionne mieux avec R7 égale à 15 kΩ au lieu de 2,7 kΩ : vous devrez donc essayer. De même le type de fil utilisé pour la découpe du polystyrène a toute son importance. Les fils résistifs usuels sont en principe inappropriés en raison de leur faible résistance spécifique, car ils nécessitent un courant élevé aux basses tensions. L'auteur récupère du fil dans de vieux sèche-cheveux, grille-pains, fours ou appareils chauffants de ce type. Pour obtenir de bons résultats, vous devrez le dérouler et le lisser soigneusement sans faire de plis. Le fil d'acier fin utilisé en modélisme (ou encore des cordes de guitare) convient moins en raison du coefficient de température défavorable. Alors qu'une température élevée est nécessaire, celui-ci a tendance à refroidir, et sa résistance et donc sa puissance diminuent, ce qui en fin de compte s'avère contreproductif. En termes de résistance à la traction, un compromis est à trouver entre le risque de rupture du fil et la précision du travail. Un fil trop lâche donnera une précision de coupe médiocre. Vous trouverez de nombreuses vidéos Youtube sur ce sujet, nous arrêtons donc là les commentaires.

Le transformateur doit être adapté aux exigences du fil chauffant (résistance et puissance). Le triac doit être refroidi pour des courants supérieurs à 1 A. Un petit radiateur suffit amplement. Un radiateur spécifique est à prévoir pour les courants de l'ordre de 5 A. Le transistor T3 peut supporter plus de 10 A à condition d'être bien refroidi, mais vous devrez adapter F1. Hormis ce fusible, le circuit n'est pas protégé contre les courts-circuits. ◀

(180023-04

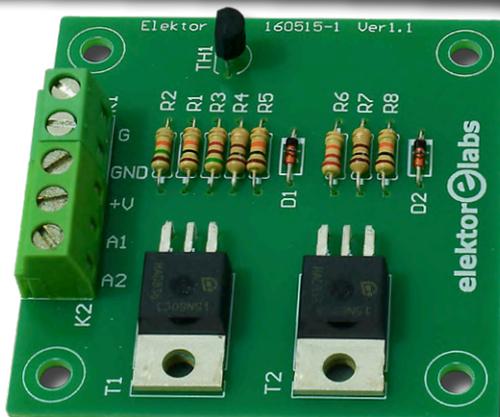
version française : Xavier Pfaff)

# thyristor MOSFET à double anode

plus de vitesse et moins de gaspillage  
qu'avec les vieux thyristors

Michael A. Shustov (Allemagne) et Ton Giesberts (labo d'Elektor)

Du fait de leur conception élémentaire, les thyristors et triacs qui servent à commuter des charges se caractérisent à l'état « ouvert » par une forte tension entre cathode et anode. En outre leurs caractéristiques de commutation laissent à désirer. En conséquence, les redresseurs commandés au silicium (SCR) sont sujets à une forte dissipation de puissance, provoquant quelquefois une surchauffe et à terme la défaillance de l'ensemble du montage électronique. De plus, l'échauffement lui-même d'un thyristor traditionnel constitue un gaspillage d'énergie électrique, qui entraîne des pertes considérables lorsqu'on passe à l'échelle de l'utilisation planétaire dans des applications industrielles ou domestiques. Il est grand temps de disposer d'un dispositif plus « vert », le voici.





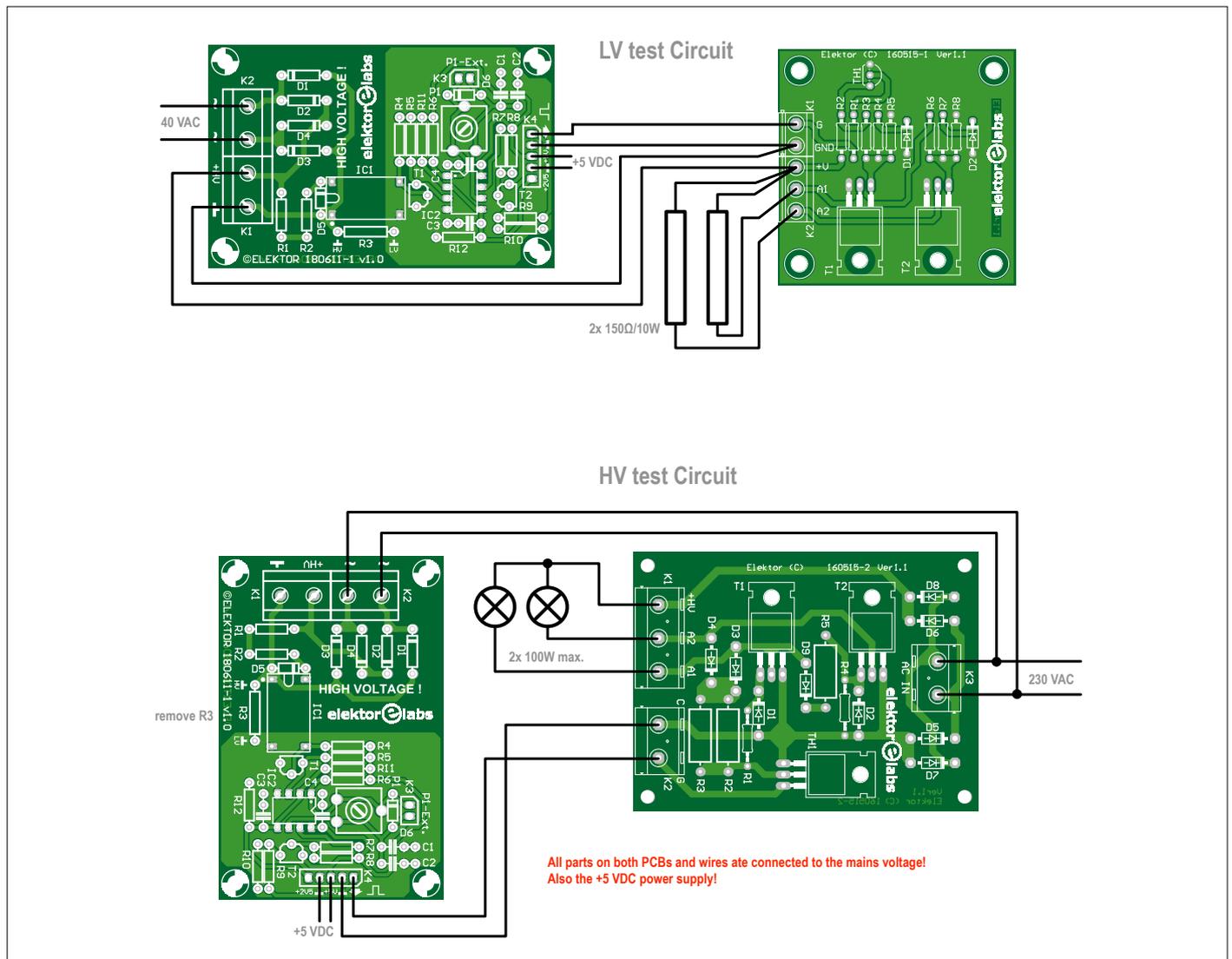


Figure 2. Diagramme de câblage des deux versions du « thyristor amélioré » à base de MOSFET.

Le circuit imprimé simple face (55 × 55 mm) conçu pour ce projet a ses composants bien espacés et ceux-ci peuvent être placés et soudés en quelques minutes. Le tracé a un plan cuivré connecté à la masse et de ce fait est inadapté à des tensions supérieures à environ 40 VCA !

### La version HT CA

Ce circuit est destiné à des charges résistives comme des lampes incandescentes ou d'autres charges aptes à être commandées par un variateur pour environ 100 W maximum. Le circuit est composé essentiellement du pont redresseur D5 à D8, du thyristor Th1 et des MOSFET T1, T2. Le cœur du circuit est néanmoins Th1, car il commande les deux « anodes » formées par les MOSFET à canal N T1 et T2. On peut connecter deux charges entre +HV et le drain de chaque

MOSFET (c.-à-d. les bornes A1 et A2). La tension CA du secteur est appliquée sur le connecteur K3. Comme le pont redresseur est embarqué sur cette version du circuit, sa sortie +HV est directement connectée au reste du circuit, via D3 et D4. On peut relier les deux charges au connecteur K2 – se reporter au diagramme de câblage. Comme pour le circuit BT, l'impulsion de gâchette pour Th1 est appliquée sur le connecteur K1, et on peut utiliser le « générateur d'impulsions de gâchette » pour fournir des impulsions de gâchette (G) synchronisées avec le secteur. Les résistances R4, R5 et la diode D9 permettent une commutation en douceur du MOSFET T2. R2, R3 et R5 sont des résistances à tension nominale de 500 V, car en fonction de la phase de l'impulsion, la tension maximale à leurs bornes est presque égale à celle du secteur. D9 et

R4 assurent respectivement la charge et la décharge de la capacité de grille de T2. Les diodes Zener D1 et D2 protègent les grilles des MOSFET T1 et T2, et limitent les tensions de grille à environ 12 V. Le courant de déclenchement du thyristor de type BT151-800R est typiquement de 2 mA et maximum de 15 mA. Improbable, mais pas impossible : le courant de déclenchement fourni par le « générateur d'impulsions de gâchette » pourrait être un peu trop faible. On peut sans problème diminuer à 150 Ω la résistance R9 de la carte du « générateur d'impulsions de gâchette ».

Le circuit a été testé avec deux charges de 100 W, avec un courant d'entrée de 0,43 A sous 230 VCA et un facteur de puissance de 1,0 (c.-à-d. des charges résistives). Lors de la conception du circuit imprimé simple face, nous avons porté

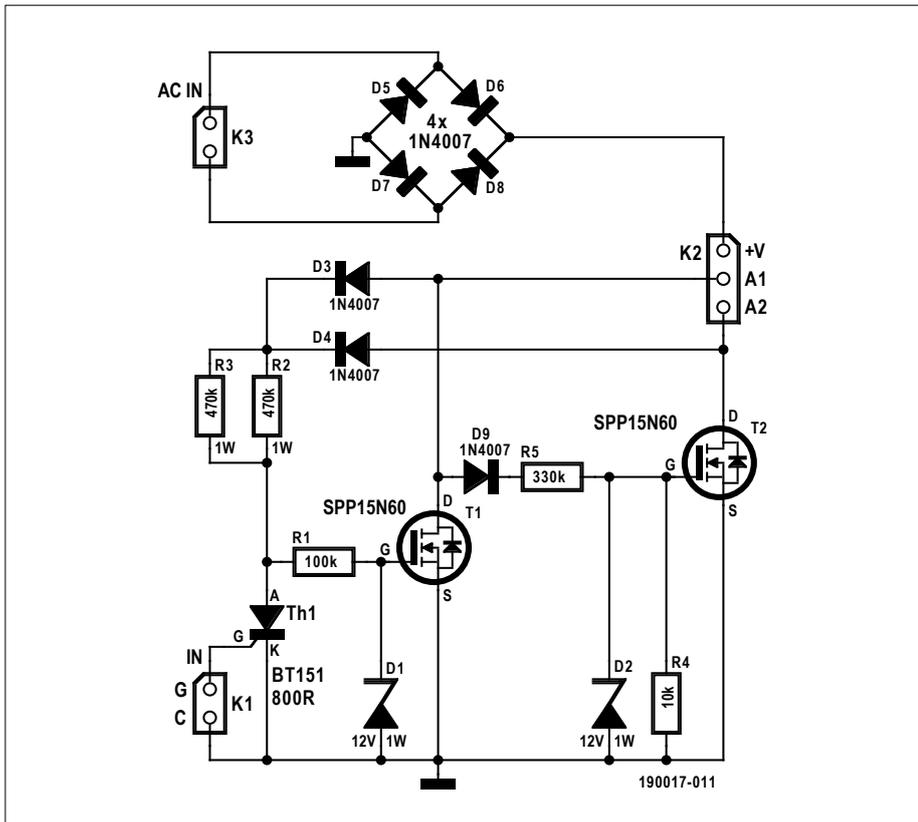


Figure 3. Cette version du thyristor MOSFET à double anode est conçue pour commander deux charges alimentées par le secteur d'environ 100 W chacune.

une attention toute particulière à la sécurité électrique. Les broches des blocs de borniers à vis sont espacées de 7,5 mm pour dépasser largement les 3 mm requis pour la sécurité électrique.

**Attention, une fois la tension du secteur présente sur K3, tous les com-**

**posants sont en principe au potentiel du secteur CA, y compris le générateur d'impulsions connecté à K1 et son alimentation de +5 V.**

#### Procédures d'essais

Reportez-vous au diagramme de câblage de la figure 2.

### 1. Thyristor MOSFET à double anode, version BT CA

Il n'y a pas de pont redresseur sur le circuit imprimé, il en faut donc un externe (comme sur le « générateur d'impulsions de gâchette ») sans condensateur de filtrage pour produire la tension +V (c.-à-d. CC pulsé). Ce +V (+HV sur le « générateur d'impulsions de gâchette ») doit être connecté à K2. Il faut également connecter les deux lignes de terre bien qu'elles le soient déjà si R3 (0 Ω ou cavalier) est installée sur la carte du « générateur d'impulsions de gâchette ». Supprimez R3 si vous devez aussi tester le circuit HT.

- Connectez la charge 1 et la charge 2 à K2 avec une extrémité reliée à +V, l'autre respectivement à A1 et A2. Les charge 1 et charge 2 sont des résistances de 150 Ω / 10 W par anode (A). La charge 1 sera pour l'état actif, la charge 2 pour l'état inactif.
- Connectez le « générateur d'impulsions de gâchette » (n°180611) et branchez le 5 V sur la carte de commande.
- La terre du circuit de puissance (K1) et celle du circuit de commande (K4 sur 180611) doivent être reliées l'une à l'autre.
- Tournez le potentiomètre P1 qui commande la puissance instantanée dans les charges 1 et 2.
- À cause de la dissipation de puissance dans les résistances de charge 1 et 2, il peut y avoir un fort



## LISTE DES COMPOSANTS

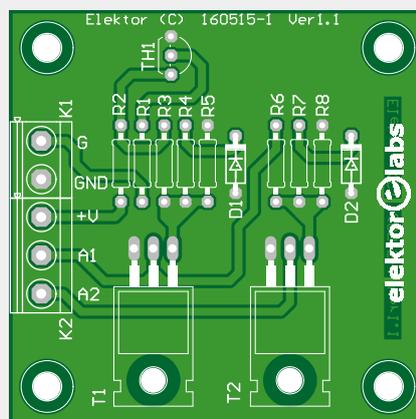
Circuit imprimé du thyristor MOSFET à double anode version BT, réf. 160515-1

#### Résistances

- R1, R5, R8 = 10 kΩ, 5%, 0,25 W, 250 V
- R2 = 15 kΩ, 5%, 0,25 W, 250 V
- R3 = 5,6 kΩ, 5%, 0,25 W, 250 V
- R4, R7 = 100 Ω, 5%, 0,25 W, 250 V
- R6 = 22 kΩ, 5%, 0,25 W, 250 V

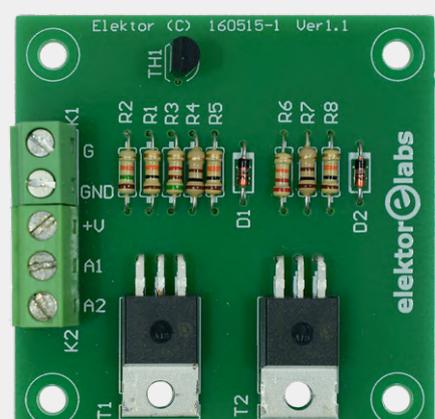
#### Semi-conducteurs

- D1, D2 = 1N5242BTR, diode Zener, 12 V, 5%, 0,5 W, DO-35
- Th1 = P0102DA 1AA3, thyristor, 800 mA, 400 V, TO-92
- T1, T2 = SPP15N60C3XKSA1, MOSFET à canal N, 600 V, 15 A, TO-220



#### Divers

- K1 = bornier à vis à 2 bornes, au pas de 5 mm, 630 V



- K2 = bornier à vis à 3 bornes, au pas de 5 mm, 630 V

Circuit imprimé, réf. 160515



## LISTE DES COMPOSANTS

Circuit imprimé du thyristor MOSFET à double anode version HT, réf. 160515-2

### Résistances

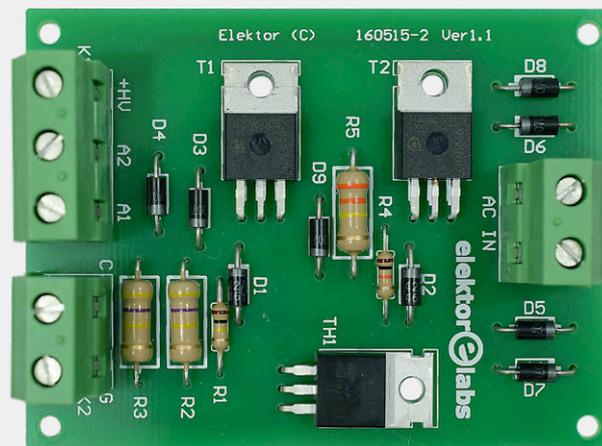
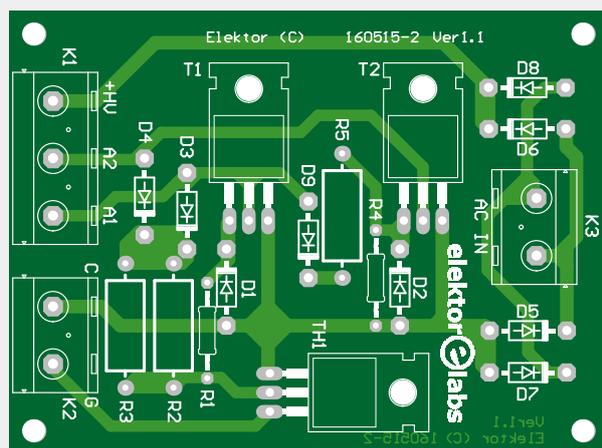
R1 = 100 k $\Omega$ , 5%, 0,25 W, 250 V  
R2, R3 = 470 k $\Omega$ , 5%, 1 W, 500 V  
R4 = 10 k $\Omega$ , 5%, 0,25 W, 250 V  
R5 = 330 k $\Omega$ , 5%, 1 W, 500 V

### Semi-conducteurs

D1, D2 = 1N4742A, diode Zener, 12 V, 5%, 1 W, DO-41  
D3-D9 = 1N4007, DO-41  
Th1 = BT151-800R, thyristor, 800 V, 7,5 A,  $I_{gt}$  max. 15 mA, TO-220  
T1, T2 = SPP15N60C3XKSA1, MOSFET à canal N, TO-220

### Divers

K1, K3 = bornier à vis à 2 bornes, au pas de 7,5 mm, 630 V  
K2 = bornier à vis à 3 bornes, au pas de 7,5 mm, 630 V  
Circuit imprimé, réf. 160515-2 (www.elektor.fr)



dégagement de chaleur (environ 10 W max. par charge, à 40 VCA), ce qui nécessite des précautions lors des manipulations.

## 2. Thyristor MOSFET à double anode, version HT CA

- Connectez la charge 1 et la charge 2 à K2 avec une extrémité reliée à +HV, l'autre respectivement à A1 et A2. Les charge 1 et charge 2 sont des ampoules de 100 W, mais on peut avoir aussi une charge de 25 W sur chaque anode.
- Reliez le secteur CA au bornier à vis K3 du circuit imprimé.
- La charge 1 sera pour l'état actif, la charge 2 pour l'état inactif.
- **Déconnectez complètement le secteur, c.-à-d. les deux bornes.**
- Enlevez R3 du « générateur d'impulsions de gâchette » (n°180611).
- Connectez le circuit de commande

au circuit de puissance comme indiqué dans le diagramme de câblage et appliquez 5 V à la carte de commande.

- La terre du circuit de puissance (K1) et celle du circuit de commande (K4) doivent être reliées l'une à l'autre.
- Reconnectez le secteur.
- Tournez le potentiomètre P1 sur le « générateur d'impulsions de gâchette » avec un outil de réglage en plastique (par ex. H-90/H-91/H-92-1 de Bourns). P1 commande la sortie de puissance vers les charges 1 et 2, et donc l'intensité de la lampe. Attention : les ampoules peuvent aussi devenir brûlantes !

Utilisez si possible un autotransformateur variable (variac, les plus petits font en général 500 VA) et un transformateur de sécurité 1:1 230 VA pour alimenter le

circuit. Ce dernier est nécessaire, car un autotransformateur a une sortie directement connectée à l'entrée ! Avec un transformateur de sécurité en plus, vous pouvez augmenter progressivement et en toute sécurité la tension du secteur. Mais n'allez jamais jusqu'à 100%. La tension de sortie avec l'indication « 100% » est souvent 15% supérieure à la tension d'entrée, voire plus avec de faibles charges. C'est bien pour tester les situations de surtension (sans le transformateur de sécurité bien sûr), mais **fortement déconseillé** ici ! ◀

190017-03

## Littérature

*Analogue of thyristors with field effect transistors*, Shustov, M.A., Radio (Russia), 2016, No. 12. p. 27.



@ WWW.ELEKTOR.FR

→ Circuit imprimé du thyristor MOSFET à double anode, version BT, 160515-1  
[www.elektor.fr/160515-1](http://www.elektor.fr/160515-1)

→ Circuit imprimé du thyristor MOSFET à double anode, version HT, 160515-2  
[www.elektor.fr/160515-2](http://www.elektor.fr/160515-2)

# introduction aux codes correcteurs d'erreurs

## syndrome de Huffman ? Pas de panique !

Robert Lacoste

Depuis un an, je vous propose dans cette rubrique des articles autour des techniques de communication : modulations numériques, codage en ligne, filtrage de canal, mélangeurs standard et I/Q, etc. Afin de terminer cette petite série, il est temps de vous présenter les bases des codes correcteurs d'erreurs, sans lesquels nos communications seraient... bien moins performantes que ce qu'elles sont !

### Une introduction

Toute transmission de données implique un risque d'erreur. Le nombre de bits faux sera plus ou moins élevé selon le rapport signal/bruit, mais il ne sera malheureusement jamais strictement nul. Il est donc plus que nécessaire de prévoir et de gérer ce risque d'erreur pour obtenir des communications fiables.

Deux méthodes sont envisageables. La première est la plus évidente : la retransmission en cas d'erreur. Pour cela, on ajoute un moyen pour détecter les erreurs, typiquement une somme de contrôle en fin de message (« checksum » pour les anglophiles). Le récepteur peut alors recalculer cette somme de contrôle et vérifier si le message est correct. Dans le cas

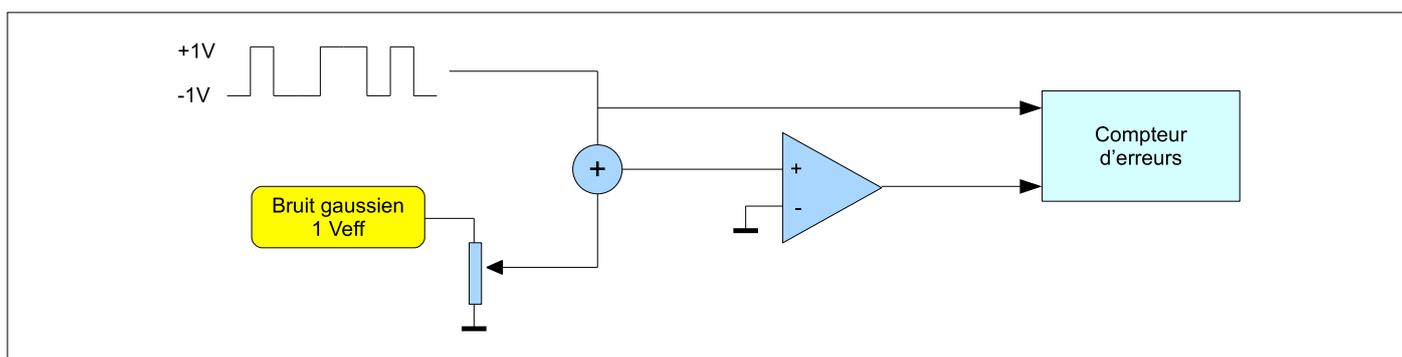


Figure 1. Un exemple de récepteur rudimentaire : un signal NRZ et du bruit gaussien sont ajoutés, le récepteur mesure le signe de la tension de la ligne.

contraire, il peut demander à l'expéditeur une retransmission. C'est simple, mais ce n'est pas utilisable s'il y a trop d'erreurs. Pourquoi ? Imaginez que les messages ont une longueur de 50 octets (400 bits), et qu'un bit sur 100 est faux. Faites le calcul, vous verrez que la probabilité de réception d'un message sans erreur est quasi nulle, même avec des milliers de retransmissions, c'est comme au loto...

La seconde approche est plus subtile : on ajoute des données redondantes au message transmis, de manière à ce que le destinataire puisse identifier, mais aussi corriger les erreurs de transmission tout seul, sans demander de retransmission. Cette méthode est appelée correction d'erreur directe (en anglais : *Forward Error Correction*, ou **FEC**). Évidemment cela ne marche que tant que le taux d'erreur reste raisonnablement faible, mais c'est beaucoup plus efficace que la méthode précédente. Bien entendu, les deux méthodes, la retransmission et la FEC, peuvent être utilisées conjointement pour des performances encore meilleures.

Vous ne serez pas surpris d'apprendre que les solutions de type FEC sont utilisées partout, du codage des CD-ROM aux puces RAM dites « ECC », mais aussi des réseaux de téléphonie cellulaires à la TNT ou aux satellites. Dans cet article, je vais vous présenter en détail l'algorithme FEC le plus rudimentaire, baptisé **code de Huffman**, et ses caractéristiques. Vous verrez que vous pouvez facilement le mettre en œuvre dans vos projets, que ce soit dans du logiciel ou du matériel. Je vais aussi vous donner quelques notions pour comprendre le fonctionnement de codes plus avancés.

### Taux d'erreur et bruit

Avant de parler de codes correcteurs, un petit rappel de quelques bases sur les erreurs sera le bienvenu. Prenons un exemple très simple. Supposons que vous envoyez un message entre deux dispositifs via un simple UART. C'est un exemple de flux codé en mode NRZ (non-retour à zéro), ce qui veut simplement dire que les bits successifs sont transmis les uns après les autres avec deux niveaux de tension (cf. [1]). Supposons que les tensions utilisées sont  $-1$  V pour les zéros et  $+1$  V pour les uns. Supposons aussi qu'il y a du bruit sur la ligne : ajoutons donc à ce signal une tension de bruit (**fig. 1**), typiquement ce qu'on appelle un bruit gaussien. Comment fonctionne le récepteur ? Il va essayer de récupérer les bits de données en mesurant la tension de la ligne, par exemple avec un comparateur de tension réglé à  $0$  V : si la tension est supérieure à  $0$  V, il dira que c'est un 1, et inversement.

Maintenant que se passe-t-il lorsqu'il y a du bruit ? Tant que le niveau de bruit reste faible, la probabilité qu'un 1 soit pris pour un 0, et vice-versa, est très faible : très peu de bits seront erronés (**fig. 2**). Cette probabilité devient de plus en plus élevée quand le niveau de bruit augmente, c'est-à-dire lorsque le rapport signal/bruit se dégrade. Dans mon exemple, avec un bruit gaussien, un calcul montre que si la tension de bruit est de  $1$  V<sub>eff</sub> alors le taux d'erreur binaire est de 7,8%. Attention quand même, ce taux d'erreur n'est jamais nul : une gaussienne peut avoir une valeur très petite, mais n'est jamais nulle. Cela veut dire que même si le bruit est faible alors la probabilité qu'un jour il y ait un bit faux n'est jamais exclue...

Pour un système donné, le taux d'erreur (en anglais **BER**, pour *Bit Error Rate*) dépend donc du rapport signal/bruit. Les caractéristiques du récepteur sont donc représentées sous forme d'un graphe qui donne le BER en fonction du **rapport signal/**

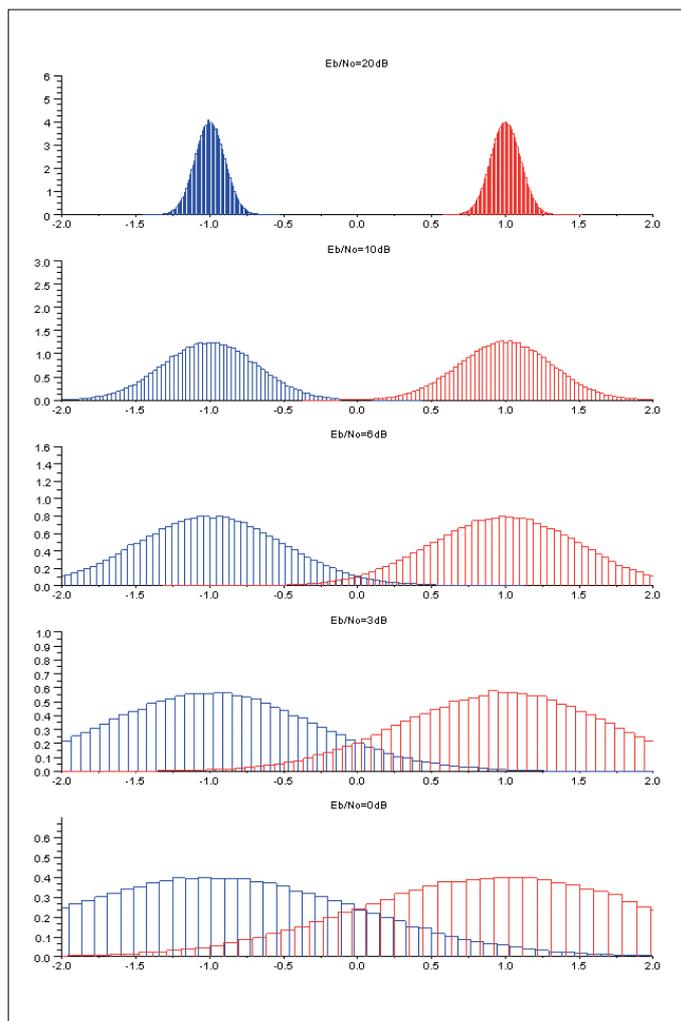


Figure 2. Ces histogrammes montrent l'effet du bruit sur la tension reçue avec un niveau de bruit gaussien faible (haut) à très élevé (bas). Les erreurs de bits apparaissent lorsque les deux courbes se chevauchent.

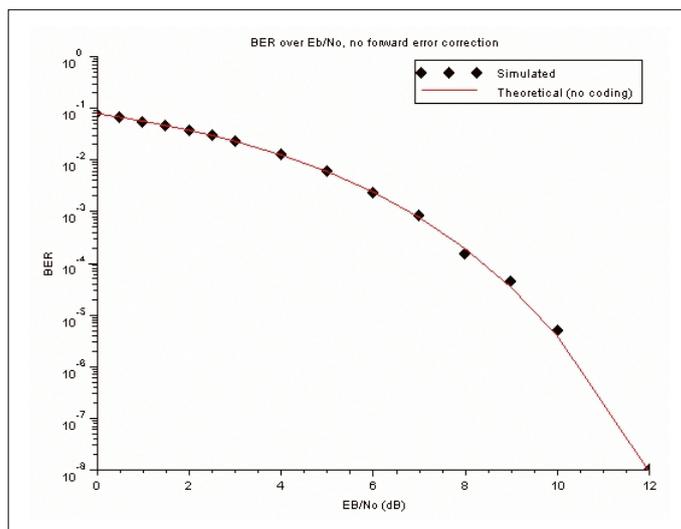


Figure 3. Ce graphe donne le taux d'erreur binaire (BER) en fonction du rapport signal/bruit ( $E_b/N_0$ ) dans le cas d'une transmission NRZ sans code correcteur.

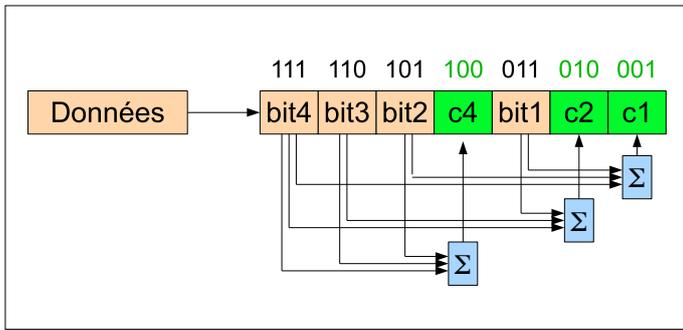


Figure 4. Un code de Huffman (7,4) utilise des blocs de sept bits, dont trois sont réservés aux bits de correction d'erreurs. Ce sont ceux aux positions 001, 010 et 100. Chacun contient la parité calculée pour certains bits de données.

**bruit (SNR).** Dans le cas du NRZ, le résultat est la courbe de la **figure 3**. L'axe vertical est le BER, l'axe horizontal le SNR, ou plutôt une grandeur sœur baptisée dans la littérature  $E_b/N_0$ . Normalement c'est un peu plus compliqué, mais dans cet exemple c'est la même chose,  $E_b/N_0 = SNR$ , donc ne vous en faites pas trop. Par exemple  $E_b/N_0 = 6$  dB signifie une puissance de bruit inférieure de 6 dB au signal (c'est-à-dire quatre fois inférieure, car  $10^{6/10} = 4$ ), donc une tension efficace (RMS) deux fois inférieure, car la tension est proportionnelle à la racine carrée de la puissance (vous vous rappelez,  $U = \sqrt{P \times R}$  ?). Regardez le graphe, vous verrez qu'alors le BER est de l'ordre de 0,24 %.

*Nota : tous les graphes présentés dans cet article ont été créés avec de petits scripts en SciLab, un outil gratuit de simulation numérique [2]. Si ça vous intéresse, téléchargez-les donc depuis le site d'Elektor [3] !*

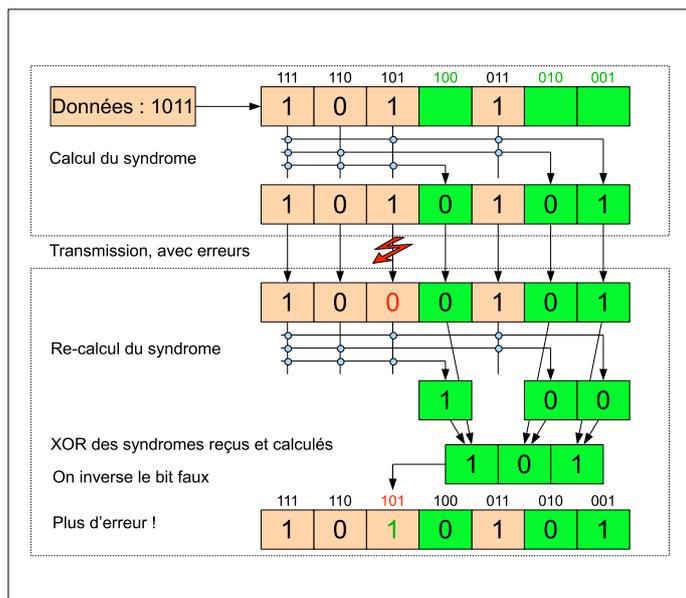


Figure 5. Illustration du codage et du décodage d'un code de Huffman. Ici, les bits d'entrée sont '1011' et sont transmis avec trois bits FEC supplémentaires. Même si un bit est inversé pendant la transmission, le décodeur est capable de reconstruire le message.

## FEC ?

Revenons donc à nos moutons, et aux codes correcteurs d'erreurs de type FEC. Il y a, comme vous pouvez l'imaginer, beaucoup d'algorithmes. Ils sont divisés en deux catégories : les codes qui fonctionnent sur des blocs de données de longueur fixe, et les codes convolutifs qui gèrent un flux continu de bits. Commençons par les premiers. L'approche la plus élémentaire de FEC est simplement... la répétition. Supposons que vous ayez un message « Coucou » à envoyer. Vous pourriez simplement l'envoyer dix fois en ajoutant une somme de contrôle à chaque message. Même si quelques bits sont faux, le destinataire pourra récupérer le message sans erreur tant qu'au moins un des messages est reçu sans erreur. Mieux, il pourrait comparer tous les messages reçus bit à bit et prendre un vote majoritaire. Bien sûr cette méthode fonctionne, mais elle n'est pas très efficace.

Il existe une multitude d'algorithmes de codes correcteurs de blocs : Golay, BCH, Reed-Solomon... Toutefois je me concentrerai sur le plus simple d'entre eux pour vous montrer comment fonctionne un tel code FEC : le code de Huffman. Pourquoi ? Parce qu'il est à la fois très facile à réaliser, et assez magique à comprendre. Les codes de Huffman sont utilisés par exemple dans les mémoires à correction d'erreurs (ECC) ou dans des protocoles radio comme LoRa (cf [4]).

## Le code de Huffman

Allons y. La **figure 4** illustre comment un code FEC de Huffman est créé côté émetteur. C'est très simple : le message et le code correcteur associé vont être envoyés en blocs, et la taille des blocs en bits doit être impérativement une puissance de deux moins un, c'est-à-dire 7 bits, 15 bits, 31 bits, etc. On va regarder comment fonctionne le code de Huffman dans le cas de mots de 7 bits.

Nous numérotions chaque bit de 1 à 7, ce qui donne en binaire : 001, 010, 011, 100, 101, 110, 111. Regardez de nouveau la figure 4 : trois de ces bits ont été colorés en vert. Ce sont les positions dont le numéro de bit, représenté en binaire, ne contient qu'un seul '1' : il s'agit donc des bits 100, 010 et 001. Ces trois bits sont réservés aux bits de correction d'erreur (code FEC). Il reste donc  $7 - 3 = 4$  bits, qui vont être utilisés pour transférer un bloc de 4 bits du message utile. Ce code de Huffman particulier est appelé « Huffman (7,4) », car 4 bits de données sont transmis dans un bloc de 7 bits.

Que doit faire l'émetteur ? Simplement décomposer le message en blocs de 4 bits, et recopier ces 4 bits dans les positions orange de la figure 4. Il doit ensuite calculer la valeur des bits des positions vertes, c'est-à-dire du code FEC. Comment ? C'est toujours simple. Nous avons vu que chaque bit de correction (case verte) porte un numéro de position avec un seul bit à '1'. Il faut additionner les bits du message utile qui se trouvent dans les cases dont le numéro de position comporte ce même bit à '1'. Si le résultat de l'addition est pair, le bit de correction considéré prend la valeur '0' ; pour un résultat impair, la valeur '1'. C'est compliqué à expliquer, mais simple à appliquer ! Regardez encore la figure 4. Par exemple la case verte de numéro 010 reçoit la parité des bits du message qui sont dans des cases dont le numéro est de type X1X, c'est-à-dire ceux des cases 111, 110 et 011, mais pas 101. Les trois bits de correction d'erreur (cases vertes) sont appelés le **syndrome de Huffman**. Ensuite, il suffit d'envoyer ces 7 bits.

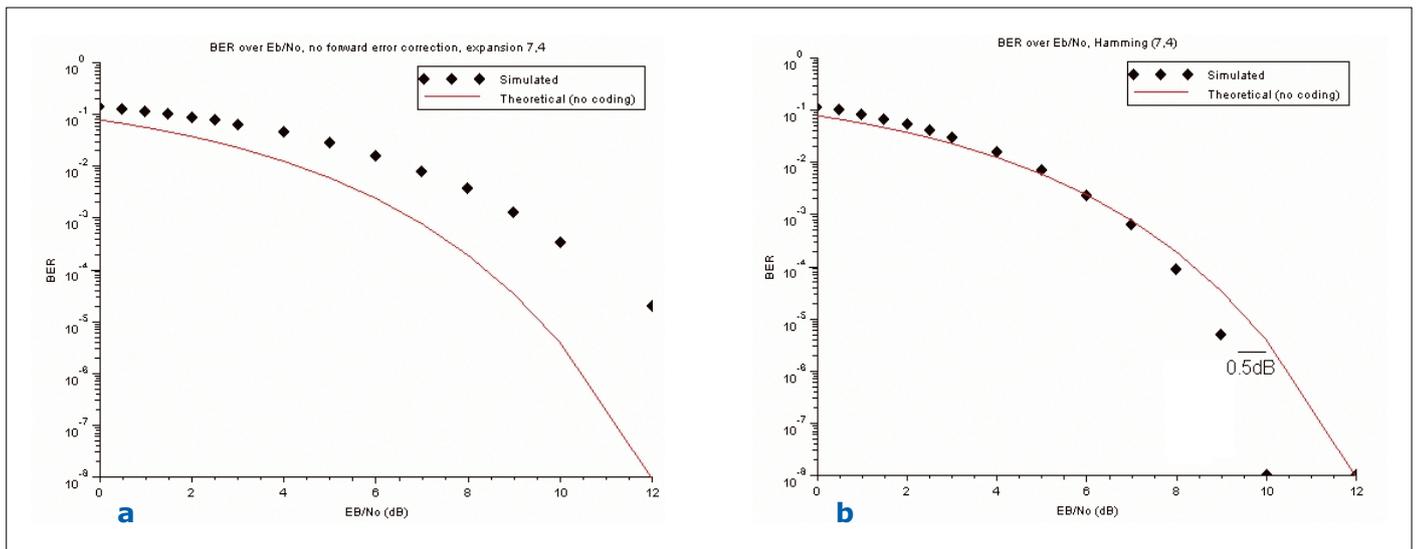


Figure 6. a) L'utilisation d'un code de Huffman (7,4) dégrade le BER lorsque le décodeur FEC n'est pas activé, car le débit est plus rapide. b) Avec le FEC en action, le BER est considérablement réduit, mais apporte un gain uniquement si le taux d'erreur n'est pas trop grand.

Prenons donc un exemple (**fig. 5**), et supposons que les 4 bits du message à transmettre sont '1011'. Recopions ces 4 bits dans les cases orange. Que vaut le syndrome ? Son premier bit (case 100) sera la parité des bits 111, 110 et 101, qui valent respectivement 1, 0 et 1. Le total est pair, le bit 100 du syndrome est donc égal à '0', qu'on recopie dans la case verte correspondante. Vérifiez pour les autres bits, vous verrez que le syndrome calculé est '001', donc le message transmis est '1010101'.

### Et le récepteur ?

La beauté de ce mécanisme se trouve du côté de la réception, car la correction d'erreurs est aussi simple à faire que le calcul du code côté émetteur. Regardez donc encore la figure 5. L'émetteur a transmis le bloc '1010101', mais supposons que le troisième bit a été reçu avec un niveau faux. Le récepteur a donc reçu '10(0)0101'. Comment va-t-il faire pour déterminer s'il y a une erreur et la corriger ?

Ouvrez vos yeux et soyez émerveillés : le récepteur calcule le syndrome du bloc reçu, avec le même algorithme que côté émetteur. Pour cela, il n'a qu'à regarder la valeur des 4 bits orange et effectuer trois calculs de parité. Ici ça donne '100', alors que le syndrome reçu est '001', il doit donc y avoir une erreur quelque part. Mais quel bit est faux ? Pour le savoir, il suffit d'appliquer un OU exclusif de ces deux syndromes :  $100 \oplus 001 = 101$ . Et 101 est... la position binaire du bit qui est faux ! Bingo ! Il suffit d'inverser ce bit et on retrouve le message sans erreur.

Je vous encourage à vérifier cet algorithme avec d'autres bits incorrects, que ce soit un bit orange ou un bit vert, vous découvrirez qu'un tel code FEC peut corriger un bit faux dans un paquet de 7 bits, mais pas plus. S'il y a deux bits incorrects, l'algorithme ne s'en sortira pas et fera empirer les choses. Un bit de contrôle supplémentaire, comme une parité globale, permettrait de détecter une telle double erreur, mais pas de la corriger.

Évidemment, vous aurez noté que toutes ces opérations sont

des opérations binaires élémentaires. Elles peuvent donc être réalisées soit avec quelques portes logiques soit avec quelques lignes de code. Un tel code est donc très simple et surtout très rapide.

### Performances ?

Quel est l'effet de l'ajout d'un tel code FEC de Huffman (7,4) sur le taux d'erreur d'une transmission ? Tout d'abord un point important pour éviter de dire des bêtises : un code correcteur corrige des erreurs, mais rallonge le message. Par exemple, ce code FEC ajoute 3 bits à chaque groupe de 4 bits de données, donc le nombre de bits transmis est augmenté d'un rapport  $7/4 = 1,75$ . On pourrait garder le même débit et passer plus de temps à envoyer le message, mais c'est... tricher. En effet on pourrait aussi ne pas utiliser de FEC et envoyer les

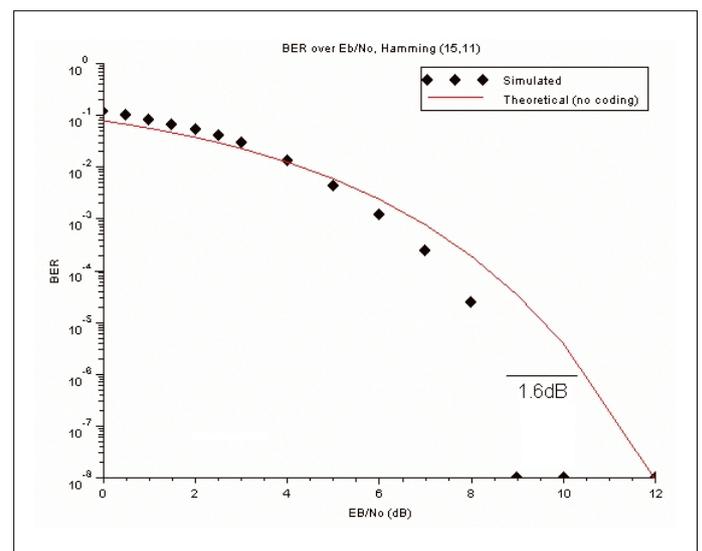


Figure 7. Le passage à des codes plus longs, ici un code de Huffman (15,11), apporte une grande amélioration, jusqu'à 1,6 dB.

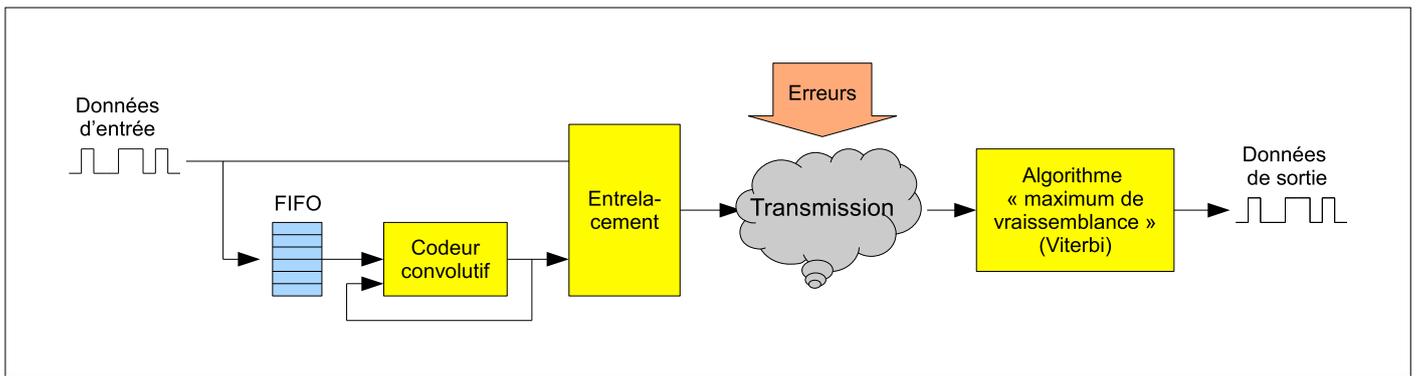


Figure 8. Ce diagramme montre le principe d'un codeur et d'un décodeur convolutifs. L'idée est d'envoyer le flux de données d'entrée, mais aussi un flux dérivé qui prend en compte l'« historique » des données.

bits plus lentement, la résistance au bruit serait meilleure. Pour voir si utiliser un code correcteur est mieux que de ne rien faire, il faut donc considérer que la durée d'émission doit rester la même.

Avant de prendre en compte les effets positifs du code FEC sur le taux d'erreur, il faut donc ici multiplier par 1,75, ce qui équivaut à décaler la courbe de  $10 \log(1,75) = 2,4$  dB. Regardez la figure 6a, c'est malheureusement notre point de départ avec ces bits ajoutés. Ensuite, l'effet positif du code FEC doit être pris en compte et il améliore le BER, du moins tant que le nombre d'erreurs est raisonnablement faible. Le résultat est fourni sur la figure 6b. **Que constate-t-on ?** Pour des taux d'erreur très élevés (en haut à gauche), le code FEC n'arrive pas à améliorer la situation. Le BER est donc encore plus élevé que sans code FEC, en raison de l'augmentation du débit. À partir d'un certain seuil, les avantages apportés par le code correcteur commencent à être supérieurs aux inconvénients, et le BER avec code correcteur devient meilleur que sans code.

Même si ce n'est pas intuitif au premier abord, un code correcteur d'erreurs est donc efficace surtout s'il y a relativement peu d'erreurs, car l'augmentation du débit n'est pas trop gênante et le code parvient à corriger les erreurs restantes. Avec des taux d'erreur suffisamment faibles, on peut montrer que le gain devient constant en dB, égal à  $10 \log(R \times (t+1))$ . R est le taux d'expansion du code, ici 4/7, et t le nombre d'erreurs binaires qui peuvent être corrigés, ici 1. L'amélioration théorique sera donc  $10 \log(4/7 \times (1+1))$ , soit 0,58 dB. Comment obtenir de meilleures performances ? En utilisant un code FEC plus sophistiqué. Par exemple, si nous remplaçons le code de Huffman (7,4) par un code (15,11), c'est-à-dire 11 bits de données et 4 bits de code d'erreur, le gain théorique passe à 1,6 dB. Regardez la figure 7 pour la simulation correspondante.

### Codes convolutifs ?

Les codes de Huffman sont les plus simples des codes correcteurs. De bien meilleures performances peuvent être obtenues soit avec des codages par blocs plus sophistiqués, soit avec des codes convolutifs dont je vous parlerai dans une minute, soit avec une combinaison des deux. L'inconvénient est que ces techniques sont nettement plus complexes. Je vais donc juste tenter de vous expliquer le principe général.

Jetez donc un œil à la figure 8, qui illustre le fonctionnement d'un codeur convolutif. De tels systèmes fonctionnent avec un flux continu de bits à transmettre. Du côté de l'émetteur, l'idée est de calculer un deuxième flux de bits, le code FEC, dérivé des bits des données d'entrée. Pour déterminer un bit donné de ce flux, seuls les k derniers bits de l'entrée sont utilisés. Cette valeur k est appelée longueur de **contrainte du code**, et donne en quelque sorte la « mémoire » du code. Les bits des deux flux sont alors entrelacés et envoyés par la liaison. La solution la plus simple est d'avoir un bit FEC pour chaque bit d'entrée, ce qui donne un codeur convolutionnel de débit 1/2. Par exemple, la mission spatiale Voyager a utilisé un code convolutionnel avec  $k = 7$  et un taux de 1/2, mais d'autres taux sont également possibles. On peut avoir un débit 2/3 (deux bits de données, un bit de FEC) pour réduire le coût du codage : il suffit de jeter un bit de FEC sur deux. Inversement, pour les canaux très bruités, un nombre beaucoup plus élevé de bits FEC est possible : le code FEC utilisé pour la sonde Cassini vers Saturne avait un taux de 1/6 (un

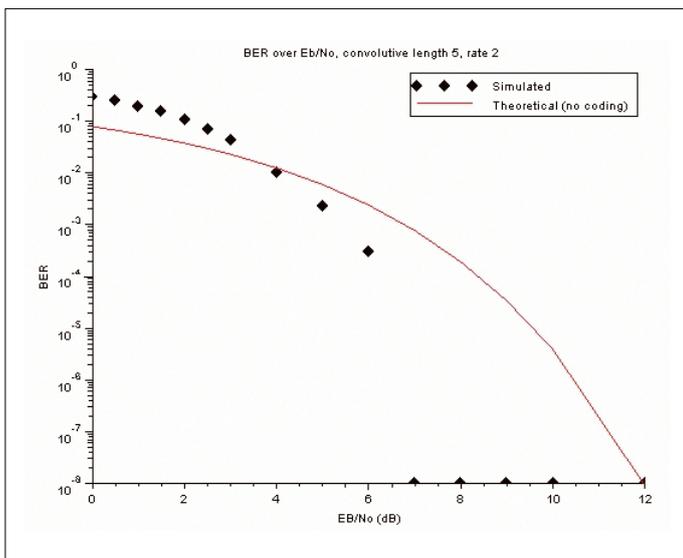


Figure 9. Voici la simulation d'un code convolutif avec une longueur  $k = 5$  et un rapport de 1/2. Impressionnant, non ?

bit de données et cinq bits de code FEC...), suffisant pour augmenter drastiquement la redondance du signal et faire face à des budgets de liaison très défavorables.

La partie compliquée pour les codes convolutifs est du côté récepteur. L'idée ici aussi est de recalculer le flux FEC en fonction des bits de données, et de le comparer avec les bits FEC reçus. En cas de désaccord, on peut alors retrouver le bit incorrect le plus probable en utilisant un algorithme dit de Viterbi (ou encore algorithme « maximum de vraisemblance »). Comme son nom l'indique, cet algorithme a été inventé par Andrew Viterbi en 1967. L'idée est de chercher intelligemment à travers tous les chemins possibles des bits d'entrée pour trouver celui qui est le plus susceptible de fournir le flux de données reçu, même en présence d'erreurs aléatoires. Si vous voulez plus d'informations sur ce sujet, consultez les articles cités dans les références, et en particulier le très bon article de Charan Langton [10]. Regardez le résultat de la simulation d'un tel code en figure 9, **vous** verrez que c'est bien meilleur qu'un code de Huffman, mais cela se paye en complexité.

Un point important à noter ici : quelle que soit la méthode, on peut en général réaliser une correction d'erreurs bien plus efficace si l'on connaît, outre les bits reçus, un indicateur sur la « qualité » de cette réception. Pourquoi ? Parce que si certains bits d'un message ont été moins bien reçus que d'autres, la probabilité qu'il y ait une erreur sur ces bits-là est bien plus grande... Un algorithme comme celui de Viterbi peut utiliser très efficacement une telle information « analogique ». De tels récepteurs nécessitent évidemment plus d'intelligence qu'un simple UART, cette technique est appelée « soft decoding ».

### Pour conclure

Le domaine des codes correcteurs est un terrain constant d'innovation depuis les premiers travaux de Shannon et Huffman en 1948. On peut par exemple citer les turbo-codes inventés au début des années 90 par des chercheurs français de l'ENST

Bretagne, Claude Berrou et Alain Glavieux. Cette invention leur a valu la médaille Richard W. Hamming de l'IEEE en 2003. Ces turbo-codes sont utilisés massivement depuis, en particulier pour les réseaux 3G et 4G. Si vous trouvez mieux, vous pouvez devenir riche et célèbre !

Vous avez compris que certains codes peuvent être très complexes à mettre en œuvre et très gourmands en ressources de calcul, mais d'autres, comme les codes de Huffman, peuvent être très simples et pourraient améliorer considérablement les performances ou la fiabilité de vos projets. Expérimentez donc un code FEC dans votre prochain projet ! ◀

(190138-01)

Cet article a été publié dans la revue Circuit Cellar (n°235, février 2010).

### Liens

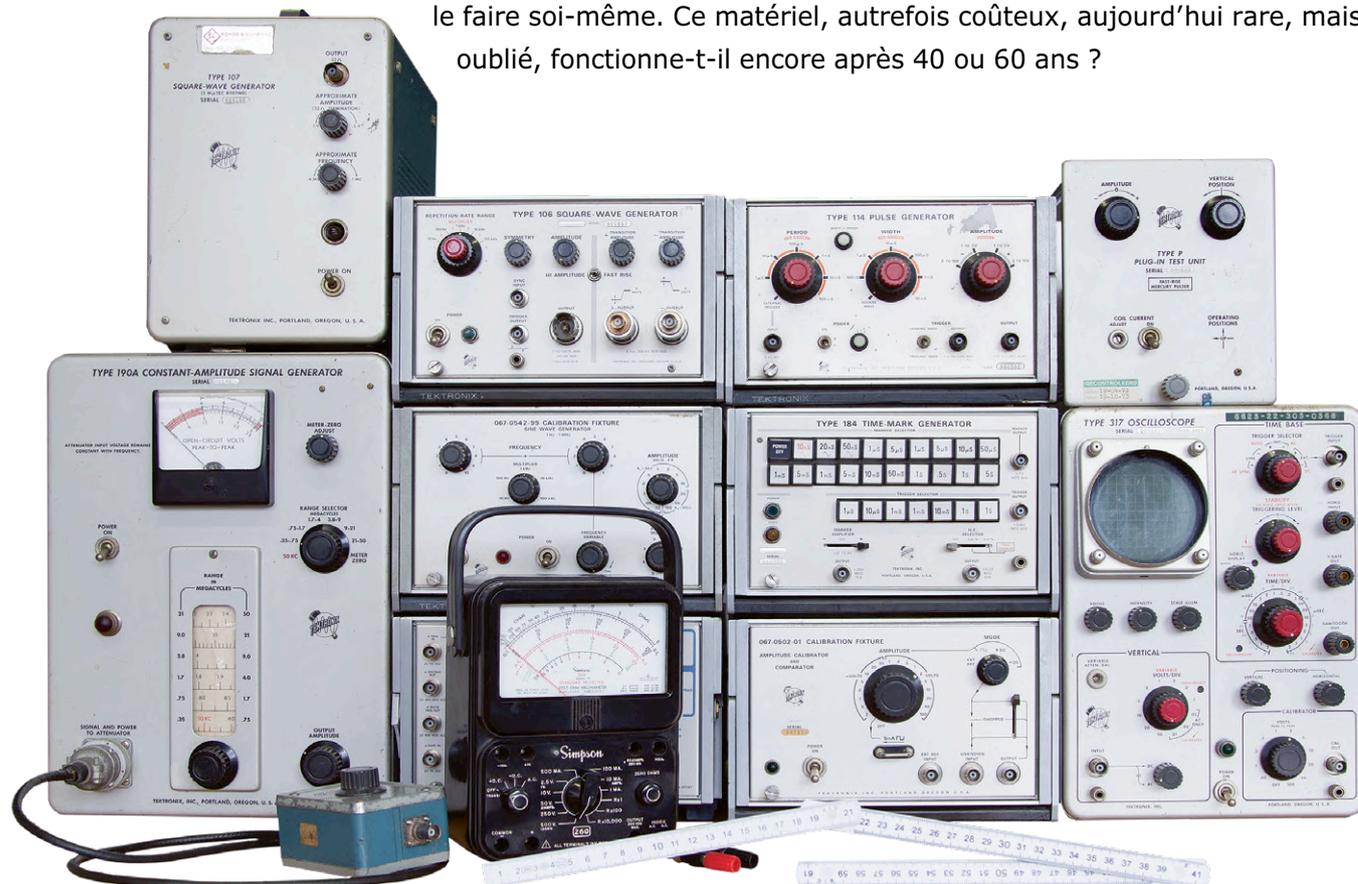
- [1] « Hors Circuits - Modulations numériques : codage des données | NRZ, Manchester & Co », Elektor 09-10/2018 : [www.elektormagazine.fr/180370-01](http://www.elektormagazine.fr/180370-01)
- [2] Outil de simulation Scilab : [www.scilab.org](http://www.scilab.org)
- [3] Page de l'article : [www.elektormagazine.fr/190138-01](http://www.elektormagazine.fr/190138-01)
- [4] « Hors Circuits - LoRa, échange débit contre portée », Elektor 01-02/2017 : [www.elektormagazine.fr/160236](http://www.elektormagazine.fr/160236)
- [5] 'Forward Error Correction', Wikipedia : [http://en.wikipedia.org/wiki/Forward\\_error\\_correction](http://en.wikipedia.org/wiki/Forward_error_correction)
- [6] 'Communications in the presence of noise', Claude E. Shannon : <http://nms.csail.mit.edu/spinal/shannonpaper.pdf>
- [7] 'A coding theory tutorial', Randy Yates, Digital Signal Labs : <https://web.archive.org/web/20110710143034/http://www.digitalsignallabs.com/tutorial.pdf>
- [8] 'What's All This Eb/No Stuff, Anyway?', Jim Pearce, Spread Spectrum Scene : [www.sss-mag.com/ebn0.html](http://www.sss-mag.com/ebn0.html)
- [9] Algorithme de Viterbi, Wikipedia : [https://fr.wikipedia.org/wiki/Algorithme\\_de\\_Viterbi](https://fr.wikipedia.org/wiki/Algorithme_de_Viterbi)
- [10] 'Coding and decoding with convolutional codes', Charan Langton : <http://complextoreal.com/wp-content/uploads/2013/01/convo.pdf>
- [11] 'In search of a 2-dB coding gain', J.H. Yuen & Q.D.Vo, NASA : [http://tmo.jpl.nasa.gov/progress\\_report/42-83/83C.PDF](http://tmo.jpl.nasa.gov/progress_report/42-83/83C.PDF)
- [12] 'Near Shannon limit error-correcting coding and decoding: turbo-codes', Claude Berrou, Alain Glavieux & Punya Thitimajshima, ENST Bretagne, 1993 : <https://ieeexplore.ieee.org/document/397441>

# instruments d'étalonnage des oscilloscopes Tektronix

## le réglage fin le plus précis

Jan Buiting, rédacteur de la rubrique Rétronique

C'est un grand amusement et même un voyage exploratoire complet que de réparer et remettre aux normes des appareils d'électronique vénérables avec les instruments, les étalons et les accessoires recommandés et fournis par le fabricant. Avec cette aventure dans la Rétronique, vous ferez un tour parmi les instruments vintage de Tektronix. Bien que la firme ait offert un service d'étalonnage et d'entretien pour ses oscilloscopes, elle a depuis fourni l'information et l'appareillage nécessaires pour le faire soi-même. Ce matériel, autrefois coûteux, aujourd'hui rare, mais pas oublié, fonctionne-t-il encore après 40 ou 60 ans ?



Dans le temps, quand vous achetez un lot de bancs de mesure électronique de haut niveau chez un fabricant coté, un plan de séances d'étalonnage faisait partie de la transaction. Vous promettiez de soumettre annuellement l'appareillage au concessionnaire agréé, payé

d'avance, et d'attendre des semaines, voire des mois, le retour du matériel. Dans certains rares (et chers) cas, un technicien autorisé venait dans votre labo passer quelques jours à étalonner vos appareils à l'aide d'instruments de précision dont vous pouviez

à peine rêver d'en posséder un. Au labo d'Elektor, certains instruments HP des années 80 ont survécu à une série de déménagements et de liquidations, ils portent encore des étiquettes pâlies d'étalonnage datant du milieu des années 90.

## Étalonnage d'oscilloscope

Tandis que je collectais, réparais et documentais des oscilloscopes Tektronix de la période 1955 à 1980, j'ai souvent trouvé des références à des appareils conçus spécialement pour l'étalonnage de ces instruments de valeur. De fait, plusieurs manuels de ces célèbres oscilloscopes Tek dédient un vaste chapitre à leur étalonnage par l'utilisateur. Évidemment, on n'a pas besoin de disposer de l'appareillage prescrit par Tek pour effectuer le travail, mais rechercher ces instruments est vraiment digne d'intérêt. J'ai pu constater que certains d'entre eux sont aussi rares que des dents de poule. Dans certains cas, les instruments conçus d'abord pour les procédures d'étalonnage à l'usine ont été développés pour être

**EST<sup>®</sup> 2004**

www.elektor.tv



Rétronique est une rubrique mensuelle sur les pages glorieuses et jaunies de l'électronique, avec occasionnellement des montages de légende décrits dans Elektor. Si vous avez des suggestions de sujets à traiter, merci de les télégraphier à [redaction@elektor.fr](mailto:redaction@elektor.fr)

intégrés dans les catalogues de vente. Ils portent généralement un numéro « 067- » plutôt que l'indicatif « Type xxx ». Tous les 067 que j'ai vus jusqu'à présent ont le même boîtier, ce qui simplifie l'empilement et le transport. Réaliser un étalonnage complet de niveau utilisateur d'un oscillo type 535 de 1959, par exemple, demande dix heures, même si vous disposez des appareils auxiliaires indiqués par Tek. Ce qui prend le plus de temps, et de loin, c'est l'ajustage de la

ligne à retard à l'aide d'un générateur d'impulsions rapides comme le 106. Procéder à une séance complète d'étalonnage révèle à coup sûr plusieurs pannes et des composants vieillis qu'il faut remplacer. En récompense, vous avez un instrument qui fonctionne impeccablement et qui dépasse de loin les spécifications du constructeur indiquées « conservatrices » dans le manuel. ◀

## 067-0502-01 : appareil d'étalonnage du calibre et du comparateur d'amplitude



**Description :** générateur d'onde carrée à 1 kHz avec sortie précise en fréquence et amplitude. Plage de sortie de  $200\text{ mV}_{cc}$  à  $100\text{ V}_{cc}$  sur charge de  $1\text{ M}\Omega$  en 18 bandes. Boîtier Tektronix standard de  $35 \times 23 \times 14\text{ cm}$ .

Cet appareil se distingue par une alimentation stabilisée tout transistors à plusieurs tensions et un oscillateur à tube 6DJ8 à haute stabilité. L'amplificateur de sortie sur tube 7734 est à refroidissement passif suivi d'un atténuateur à 18 pas.

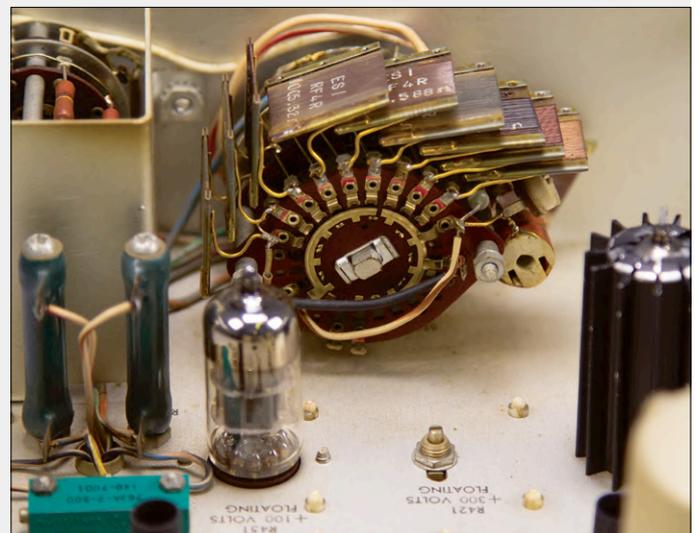
**Application :** étalonnage d'entrée verticale et atténuateurs d'oscilloscope, comparaison et ajustage précis de tensions continues.

**Historique :** acheté au propriétaire du matériel d'un ancien service d'étalonnage.

**Détails :** le numéro de type est « Tektronix internal ». L'instrument est équipé d'un « découpeur » pour une comparaison précise d'une tension continue externe avec un niveau réglé par la commande d'amplitude du 67-0502-01, commutateur de mode mis sur +DC. En raison du découpage (à 60 Hz), cela revient à comparer un oscilloscope sensible ou un millivoltmètre connecté à la sortie d'un instrument capable de

montrer, jusqu'aux millivolts, la différence entre un niveau continu « inconnu » et le niveau du calibre.

**Réparations :** j'ai trouvé un signal de sortie très faible et instable. Le rail d'alimentation à  $-100\text{ V}$  était trop bas, à  $-36\text{ V}$ . J'ai découvert une des six pattes d'un double transistor 2N2913 complètement détachée, à l'horizontale entre le boîtier du transistor TO-5 et son support. Après réparation de cette panne évidente, on repasse à  $-100\text{ V}$  et l'instrument donne un signal de sortie. Mais il est asymétrique à 70/30 du rapport cyclique, donc ce n'est plus un signal carré. Ce n'était pas la double triode 6DJ8/CV2493 dans l'oscillateur, comme on aurait pu le penser, la coupable, mais l'un des deux condensateurs au mica argenté de  $200\text{ pF}$  à 1 %. J'ai remplacé les deux par des polystyrènes de  $220\text{ pF}$  à 2,5 % de mon stock et l'oscillateur a retrouvé la symétrie, mais sur  $1\,286\text{ Hz}$  au lieu des  $1\,000\text{ Hz}$  voulus. Toutes autres tensions,  $+300\text{ V}$ ,  $+125\text{ V}$  et  $+100\text{ V}$  dans les limites d'amplitude ( $\pm 3\%$ ) et d'ondulation.



**À faire :** finir de nettoyer, calibrer l'alimentation de  $-100\text{ V}$ , trouver deux condensateurs de  $200\text{ pF}$  à  $1\%$  (polystyrène ou mica argenté) pour remplacer. Vérifier la précision réelle en amplitude.

**Utilisation actuelle :** contrôle rapide de sensibilité verticale de tous les oscilloscopes ancestraux en réparation ou étalonnage.

**Précision :** amplitude à  $0,25\%$  près dans les 18 gammes, au départ des résistances de très haute précision montées directement sur le commutateur à galettes de la **photo**.

**Notes :** rien pour l'instant.

## 067-0542-99 : appareil de calibrage de générateurs d'ondes sinusoïdales



**Description :** générateur sinus de  $10\text{ Hz}$  à  $1\text{ MHz}$  par paliers, précis en fréquence et en tension. Sortie entre  $20\text{ mV}_{cc}$  et  $100\text{ V}_{cc}$  sur charge de  $50\ \Omega$  ( $20\text{ V}$ ,  $50\text{ V}$ ,  $100\text{ V}$  sur  $450\ \Omega$ ), en 12 gammes. Fréquence ajustable par pas et à variation continue. Boîtier standard Tek :  $35 \times 23 \times 14\text{ cm}$ .

L'oscillateur classique est basé sur un tube avec régulateur d'amplitude (6DJ8 et 6A6), mais la sortie est à transistors avec un 2N4063 RCA en commande de l'atténuateur de sortie. L'amplitude de l'onde sinus est stabilisée par boucle de rétroaction (qui n'est pas trop rapide).

**Application :** étalonnage d'amplis d'entrée verticaux et d'atténuateurs d'oscilloscopes ; test de circuits de déviation, de déclencheurs, de réponse en BF jusqu'à  $1\text{ MHz}$ .

**Historique :** acheté sur eBay à un collectionneur espagnol, négociant en antiquailles.

**Divers :** le numéro de type est « Tektronics internal ». La fonction de balayage manque. Le seul document joint est une notice d'étalonnage d'usine gribouillée pour la firme Tektronics, à usage interne seul. Elle contient un schéma qui diffère sensiblement de la matérialité de mon appareil.

**Réparations :** niveau de sortie insuffisant, réponse erratique du réglage d'amplitude. Potentiomètre « amplitude multiplier » incomplet, l'une des broches détachée de la piste de carbone. J'ai reporté le fil sur l'autre extrémité (inutilisée) pour rétablir le fonctionnement, il s'agit d'un simple rhéostat.

**À faire :** continuer de nettoyer (boîtier fort corrodé) ; curer le contact rongé du  $1\text{ kHz}$  sur l'interrupteur du multiplicateur ; trouver la documentation complète ; mesurer la distorsion.

**Utilisation actuelle :** rare ; dépannage d'appareils audio ; réponse BF (sous  $100\text{ Hz}$ ) d'oscilloscopes.

**Précision :** l'amplitude sinusoïdale reste dans les  $2\%$  sur toutes les gammes, la fréquence dans les  $3\%$ . J'ai vérifié l'instrument au fréquencemètre HP 5131A emprunté au labo d'Elektor. Dans les gammes extrêmes de  $100\text{ Hz}$ , puis de  $1\text{ MHz}$ , la déviation est de  $0,7\%$  et  $0,9\%$ , les autres, entre  $0,01$  et  $0,1\%$  ce qui est tout bonnement étonnant pour un vieil instrument et supérieur à ce que prétendait Tektronix. L'amplitude de sortie dans les positions  $1\text{ V}$  et  $3\text{ V}$  est décalée de millièmes. Tous les tests ont eu lieu après  $25\text{ min}$  de chauffe.

**Notes :** excellent appareil, principalement pour sa plage de fréquence étendue et le niveau de sortie fiable.

## Type 106 : générateur d'ondes carrées

**Description :** générateur d'impulsions récurrentes entre  $10\text{ Hz}$  et  $1\text{ MHz}$  à temps de montée rapide ( $<1\text{ ns}$ ). Boîtier Tek standard :  $35 \times 23 \times 14\text{ cm}$ .

Il a un tube oscillateur/multivibrateur HF suivi d'un tube amplificateur fort puissant qui attaque deux paires de diodes à commutation très rapide. Deux sorties : l'une de  $500\text{ mV}$  sur  $50\ \Omega$  à temps de montée de moins de  $1\text{ ns}$  et l'autre de

max.  $120\text{ V}$  et  $240\text{ mA}$  et montée en  $12\text{ ns}$ . Les connexions de déclenchement et de synchro sont prévues.

**Application :** analyse du temps de montée et de la réponse en impulsion de l'oscilloscope ; mesure des défauts dans les conducteurs par TDR (*time domain reflectometer*) au réflectomètre ; mesure du dépassement de la réponse et du rétablissement.



**Historique :** acheté sur eBay à un radioamateur allemand.  
**Divers :** l'instrument est doté de trois connecteurs de sortie GR874. C'est le seul connecteur RF hermaphrodite jamais apparu sur le marché, devenu une rareté. Malgré un étiquetage clair sur le colis, l'appareil a été livré à une mauvaise adresse par DHL Allemagne et n'a refait surface qu'avec l'aide de l'expéditeur et malgré le silence de celui qui l'avait reçu.

**Réparations :** pas de  $-150\text{ V}$  à l'intérieur ! Le pont redresseur et la résistance série de protection de  $5,1\ \Omega$  étaient morts. Le remplacement par quatre diodes BY127 a restitué le  $-150\text{ V}$ . La résistance de protection du pont de l'alimentation de  $20\ \text{V}$  était aussi brûlée. Après remplacement, plus de panne du tout.

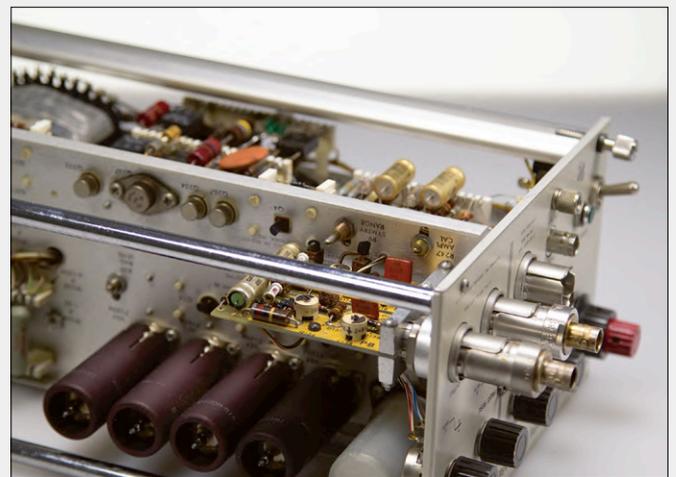
**À faire :** calibrer aux normes Tek ; vérifier une montée de

moins de  $1\ \text{ns}$  sur un oscilloscope rapide sans bousiller son entrée.

**Utilisation actuelle :** vérification du temps de montée de tous les oscilloscopes après chaque réparation ou réétalonnage. Démontir les mythes sur la vitesse et la « capacité de dépistage des transitoires rapides » attribuée aux oscilloscopes numériques modernes.

**Précision :** les aberrations sur le flanc montant d'impulsion sont  $< 2\%$  dans les premières  $5\ \text{ns}$  sur  $50\ \Omega$ .

**Notes :** le plaisir de mesurer les temps de montée est addictif. En dessous de  $1\ \text{ns}$ , il vous apprend à pressentir les capacités et inductances parasites que peuvent présenter certains microcontrôleurs. Le signal de sortie sur  $50\ \Omega$  est incroyablement rectiligne. Le connecteur GR874 est impressionnant, un pur délice pour les amateurs de RF. L'impulsion de  $120\ \text{V} / 240\ \text{mA}$  issue du 106 est **mortelle** pour les oscilloscopes de l'ère *Plastic Fantastic*.



## Tiroir de mesure type P

**Description :** un tiroir pour la série d'oscilloscopes Tektronix 500 jusqu'à  $30\ \text{MHz}$ , c'était surprenant en 1959 ! Il fournit un signal BF ( $100$  ou  $120\ \text{Hz}$ ) à montée en  $4\ \text{ns}$ . L'impulsion



vient d'un interrupteur à mercure ouvert et fermé par un transformateur. Le type P ne désigne pas un instrument à part entière, mais un module additionnel pour les séries Tek 53x/54x/55x uniquement.

**Application :** test du temps de montée et de la réponse en impulsion des oscilloscopes Tek 53x/54x/55x ; ajustage de la ligne à retard.

**Historique :** reçu gratuitement d'une faculté universitaire, trop heureuse de se débarrasser du mercure.

**Détails :** le matériel est installé sur un panneau fixé à  $45^\circ$  sur l'horizontale, ce qui permet de placer l'oscilloscope testé sur le côté droit pour ajuster la ligne à retard interne. La face avant porte une indication sommaire à ce sujet. La durée de vie de l'interrupteur à mercure n'est que de quelques dizaines d'heures. Le schéma du « P » est imprimé sur le panneau incliné. Ceci est, à ma connaissance, l'unique appareil de test Tektronix qui porte une description incorrecte sur le panneau avant ! Le type P ne teste pas un tiroir ; c'est un tiroir... pour tester l'oscilloscope, il faut... le brancher dessus :-). On a dû choisir « P » pour *pulse*.



**Réparation :** aucune, il a fonctionné à la sortie de sa boîte après 60 ans.

**À faire :** limiter à un minimum la durée de travail du P. Peut-être prévoir un interrupteur au mercure de rechange.

**Utilisation actuelle :** test jusqu'à 30 MHz du temps de montée sur les oscilloscopes Tek 53x/54x/55x avec baie pour tiroir.

**Précision :** de peu d'importance, pourvu que l'impulsion (4 ns) issue du P soit plus rapide que la montée de l'oscilloscope.

**Notes :** dérangent, le ronflement que produit le P. La récurrence (100 Hz) de l'impulsion et la vitesse de montée (environ 4 ns), une combinaison qui entrave la synchronisation à l'écran.

## Type 190A : générateur de signal d'amplitude constante



**Description :** générateur sinusoïdal LC à rétroaction pour produire un signal d'amplitude virtuellement constante à l'entrée de l'appareil à tester, malgré le changement de fréquence dans la plage de 350 kHz à 50 MHz environ. Si l'amplitude est constante, à des poussières de dB près, c'est grâce à un détecteur de crête à crête monté à l'intérieur d'une tête d'atténuateurs pour fournir le signal de rétroaction au régulateur d'amplitude du 190 A. La boucle de régulation corrige l'ondulation à la sortie de l'oscillateur du 190A ainsi que l'atténuation (fonction de la fréquence) du coaxial entre la tête de l'atténuateur et l'instrument lui-même.



Le 190A n'était sans doute pas destiné à calibrer sur le terrain. Son poids et sa dimension ont dû le confiner au laboratoire, comme au labo d'Elektor à Aix-La-Chapelle où mon 190A est exposé. Il est pourtant idéal pour le calibrage.

**Application :** vérifier la réponse en fréquence (planéité) d'oscilloscopes spécifiés jusqu'à 50 MHz environ.

**Historique :** acheté à un ingénieur retraité d'un labo de radio et téléphonie.

**Détails :** un 190A n'a que peu d'utilité sans la tête de l'atténuateur adapté. Le mien est du type 190B. L'intérieur du 190A est un chef-d'œuvre, comme la plupart des appareils à tubes de Tektronix des années 50 et 60, voyez dans la **photo** le condensateur d'accord à 3 sections et le tambour de fréquence. L'échelle de réglage ressemble au cadran d'un ancien poste à OC, on tourne facilement avec une roue en caoutchouc le tambour de l'échelle de fréquences. De cou-

leur crème derrière le plastique, elle s'éclaire doucement et vous donne envie de réécouter Route 66 en AM. Comme le 190A n'a pas de balayage en fréquence, à vous de tourner à la main le bouton dans toutes les gammes pour apercevoir les creux d'amplitude sur l'écran de votre oscillo.

**Réparations :** remplacement d'une résistance de puissance cassée (pas brûlée) de 100  $\Omega$  et d'un tube très fatigué pour la remise en état du 190A.

**À faire :** redresser l'entrée de câble du connecteur Cannon, calibrer l'appareil selon les spécifications d'usine.

**Notes :** instrument au look impressionnant doublé d'une source de signal RF avec un bel indicateur de niveau. Une stabilisation d'amplitude un peu lente jusqu'au niveau demandé. L'accord est agréable et l'échelle éclairée attire l'attention.

## Type 184 : générateur de repères de temps



**Description :** offre un choix de 16 repères temporels et 5 intervalles de marques de sinus, pour une plage totale allant de 2 ns à 5 s. Impulsions de déclenchement pour l'appareil à tester entre 1  $\mu$ s et 5 s. La précision du 184 est due à un quartz à 10 MHz confiné dans un four à température contrôlée. Boîtier Tek standard de 35 x 23 x 14 cm.

**Application :** calibrage et ajustement de circuits de base de temps et d'ampli horizontal d'oscilloscopes.

**Historique :** acheté auprès d'un particulier, considéré comme double emploi.

**Divers :** toutes les faces du boîtier percées de trous pour le refroidissement de l'électronique. Le four du quartz est maintenu à 70 °C environ et chauffe encore appareil éteint. Un témoin jaune s'allume et on entend un relais s'activer quand il réchauffe.

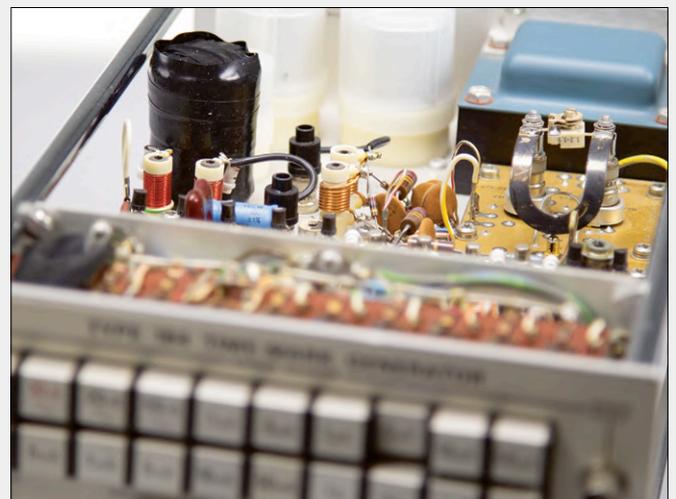
**Réparations :** aucune.

**À faire :** calibrer l'oscillateur interne à quartz sur le 10 MHz du standard de fréquence GPS.

**Utilisation actuelle :** fréquente, instrument recommandé pour l'ajustage et l'étalonnage de pratiquement tous les circuits « horizontaux » des vieux oscilloscopes, y compris base de temps et déflexion.

**Précision :** fréquence de 10 MHz à 0,001 % ; dérive <3 ppm en 24 h.

**Notes :** extrêmement utile, facile d'emploi. Long temps de chauffe pour arriver dans les normes. Belle construction, surtout l'oscillateur central.





# qu'attendons-nous de l'intelligence artificielle ?

Tessel Renzenbrink (Pays-Bas)

« Les robots nous volent notre travail ! », « Les téléphones portables détournent notre attention ! ». Ce sont deux phrases parmi tant d'autres, censées illustrer les effets de la technologie sur notre vie quotidienne. Leo Marx, ancien professeur au MIT, a dit en 2010 [1] que cela n'avait aucun sens d'accuser les machines de quoi que ce soit ; ce sont bien des êtres humains qui décident de remplacer des travailleurs par des robots. Marx constate également que l'on a tendance à attribuer une volonté propre à la technologie, et qu'on pourrait oublier que les hommes sont responsables de l'impact de la technologie sur la société.

La tendance à prêter à la technologie une capacité à agir de manière autonome est certainement valable pour l'intelligence artificielle (IA) aussi. Le succès des scénarios d'apocalypse mettant en œuvre une IA en est bien la preuve : l'humanité est d'abord asservie puis exterminée par ses propres créations. Il est clair que le choix du mot « intelligence » joue ici un rôle important. On ne peut s'empêcher d'imaginer une entité autonome, capable de choix rationnels et d'actions volontaires. Et par suite le développement, dans un avenir proche, d'ordinateurs capables de penser comme nous. L'IA ne fonctionne cependant pas comme notre cerveau, même si elle accomplit des tâches semblables : jouer aux échecs ou au go, ou encore reconnaître des chats sur des photos. La manière dont ces tâches sont menées à bien est tout à fait différente. Il n'y a pas de définition uniformisée de l'IA, mais on retrouve toujours trois ingrédients principaux : données, algorithmes et puissance de calcul. On inonde un ordinateur de données d'un côté, on les travaille à l'aide d'algorithmes, et on obtient un résultat de l'autre côté.

## Les ordinateurs n'ont pas de sens historique

Ce fonctionnement différent entraîne forcément des résultats différents pour un même problème. Yarden Katz, qui a



obtenu son doctorat en *Brain and Cognitive Sciences* au MIT et qui travaille à la Harvard Medical School et au Berkman Klein Center, nous rappelait dans un article écrit en 2017 [2] que les ordinateurs n'ont pas la moindre idée d'un contexte historique, de choix politiques ou d'émotions. Il donne comme exemple la photo d'un groupe de personnes courant pour sauver leur vie alors qu'une maison est emportée par les flots ; un système de reconnaissance de formes n'y voit qu'un groupe de personnes sur une plage... N'importe quel être humain verrait directement la menace pesant sur les personnes, un ordinateur ne la perçoit pas. Comme dit Katz : « La connaissance des objets et de leurs relations, ou des intentions et des émotions humaines – qui sont bien visibles sur les visages des personnes photographiées, manque ».

Pour démontrer cette différence dans l'interprétation, il a aussi introduit des photos dans *Show and Tell*, un logiciel d'apprentissage profond de Google qui ajoute automatiquement des commentaires à des photographies. Une de ces photos montre un soldat israélien maintenant un jeune Palestinien au sol. La famille du garçon tente de le délivrer : deux femmes tirent le soldat vers l'arrière et une fille le mord à la main. Sous-titre de l'IA : personnes assises ensemble sur un banc. « Les relations complexes entre les êtres humains sur la

photo ne sont pas vues par l'IA ; les motivations, les intentions et la finalité passent totalement inaperçues », nous rappelle Katz.

Il affirme encore : « Notre réaction à ces exemples est qu'un apprentissage adéquat permettrait l'interprétation correcte de ces photos ». D'aucuns supposent qu'une légende suffit à interpréter le contenu d'une photo : toute l'information est présente dans l'image, il suffit donc de trouver le modèle adéquat pour la déchiffrer. On peut néanmoins douter qu'un tel apprentissage permette à un ordinateur d'interpréter correctement la célèbre photo de Ruby Bridges en 1960.

### Compter les étoiles

La manière dont les systèmes d'IA résolvent les problèmes peut s'avérer utile, et il y a de nombreuses applications où les ordinateurs viennent bien à point pour soutenir la recherche scientifique. Citons à titre d'exemple SKI-CAT (*Sky Image Cataloging and Analysis Tool*), un système d'apprentissage automatique dont le but est d'identifier quelque 500 millions d'étoiles, planètes et autres objets célestes sur la base de photos, et de les cataloguer. Ce serait pour les astronomes et astrophysiciens une tâche sans fin et monotone. Une IA n'est en fait pas intelligente ou futée, ce n'est qu'un programme qui transforme des données en un résultat. Il faut toujours avoir cela à l'esprit si cette IA doit intervenir directement dans notre vie. Les exemples de ce qui peut mal tourner sont légion : par ex. un algorithme qui ne choisit que des hommes blancs pour un poste, parce que ce groupe était majoritairement représenté dans la phase d'apprentissage. Nous réagirions immédiatement si toutes les femmes et tous les gens de couleur étaient automatiquement rejetés, mais un ordinateur reste de marbre.

D'autres exemples viennent de l'introduction de l'IA dans le système judiciaire américain. Un logiciel calcule la probabilité de récidive d'un accusé, et le juge peut en tenir compte pour déterminer la peine : une amende ou un temps d'emprisonnement seront par ex. proportionnels à cette probabilité.

Une enquête de ProPublica, une plateforme indépendante de journalisme, a montré en 2016 que les noirs obtenaient régulièrement – et souvent à tort, un plus mauvais score [3]. Le récidivisme des blancs était injustement sous-estimé. Pour en revenir aux réflexions de Leo Marx, nous pouvons affirmer que la discrimination et le racisme exposés ci-dessus ne sont pas attribuables à l'IA. C'est bien le résultat d'une accumulation de manipulations humaines : le choix de laisser un logiciel prendre des décisions dans des affaires judiciaires, et l'introduction de données d'apprentissage qui contiennent des préjugés par la suite reproduits par les algorithmes. L'avènement de l'IA ne doit pas être vu comme un changement radical de notre mode de vie où nous devons lutter pour survivre aux machines. Comme le dit si bien Marx : « En autorisant la technologie à agir avec une certaine autonomie, nous nous détournons des relations humaines – entre autres socio-économiques et politiques, qui doivent rester le principal outil d'évolution de notre société ». L'intelligence artificielle est un outil très puissant qui peut favoriser le progrès ; à nous de savoir l'utiliser judicieusement

(180572-C-04

version française : Jean-Louis Mehren)



Ruby Bridges, 6 ans, lors de son premier jour dans une école jusqu'alors réservée aux enfants blancs (1960) ; elle est escortée par trois agents fédéraux américains chargés de la protéger. Google AI interprète cette photo comme « un groupe de personnes côte à côte ».

(Photo : domaine public. Source : Wikipédia – <https://is.gd/BDAGux>)

### Liens

- [1] 'Technology, The Emergence of a Hazardous Concept', Leo Marx : <https://is.gd/0jnMem>
- [2] 'Manufacturing an Artificial Intelligence Revolution', Yarden Katz : <https://is.gd/3bvOPe>
- [3] 'Machine Bias', ProPublica : <https://is.gd/YHVDXX>



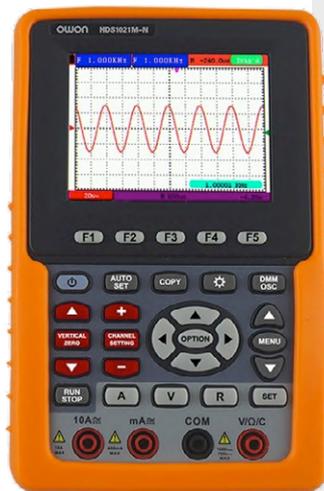
# bienvenue dans votre e-choppe

la rédaction recommande



## Oscilloscope/multimètre portable (20 MHz) OWON HDS1021M-N

Pour moins de 200 € (un peu plus de 200 \$), l'Owon HDS1021M-N vous permet de disposer de deux instruments dans un même boîtier : un oscilloscope à mémoire numérique et un multimètre, c'est une association très pratique. En plus, ce double instrument peut vous accompagner partout puisqu'il est portable et pas plus encombrant qu'un multimètre classique. L'oscilloscope présente une bande passante d'entrée de 20 MHz et une fréquence d'échantillonnage maximale de 500 Méch./s. En outre un grand nombre de paramètres sont réglables (déclenchement, curseurs, option FFT...). Le multimètre est un outil complémentaire très utile. Il affiche de bonnes caractéristiques : résolution de 3 3/4 chiffres pour la mesure, précision égale ou inférieure à 1%... À ce prix-là, l'Owon HDS1021M-N devrait déjà être dans votre mallette de dépannage !



Harry Baggen (labo d'Elektor)

[www.elektor.fr/owon-hds1021](http://www.elektor.fr/owon-hds1021)

## Vos favoris :

1. The Complete ESP32 Projects Guide (livre en anglais)  
[www.elektor.fr/esp32-projects-guide](http://www.elektor.fr/esp32-projects-guide)



2. Fer à souder Miniware TS100  
[www.elektor.fr/miniware-ts100](http://www.elektor.fr/miniware-ts100)

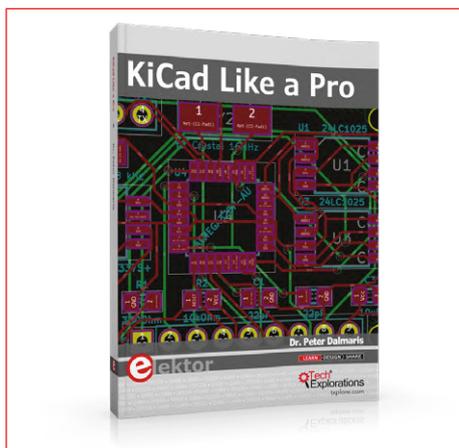
3. Kit PCBite  
[www.elektor.fr/pcbite-kit](http://www.elektor.fr/pcbite-kit)

4. Raspberry Pi 3A+  
[www.elektor.fr/rpi-3a-plus](http://www.elektor.fr/rpi-3a-plus)

5. Testeur USB Bakeey UM25C  
[www.elektor.fr/bakeey-um25c](http://www.elektor.fr/bakeey-um25c)

6. Embedded in Embedded (livre en anglais)  
[www.elektor.fr/embedded-in-embedded](http://www.elektor.fr/embedded-in-embedded)

## KiCad Like a Pro

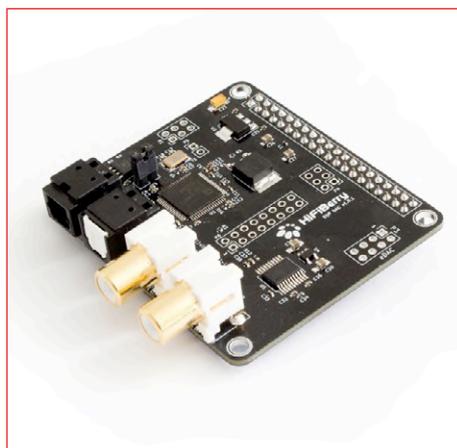


Ce livre (en anglais) vous apprendra à utiliser KiCad. Que vous soyez amateur ou ingénieur en électronique, grâce à ce livre, vous serez rapidement en mesure de dessiner vos schémas électroniques et de concevoir vos circuits imprimés. L'approche pratique de cet apprentissage repose sur quatre projets de difficulté progressive.

Prix (membres) : 35,96 €

[www.elektor.fr/kicad-like-a-pro](http://www.elektor.fr/kicad-like-a-pro)

## HiFiBerry DAC+ DSP



Le HiFiBerry DAC+ DSP est un convertisseur numérique-analogique à haute résolution pour le Raspberry Pi. Il combine un convertisseur N/A Burr-Brown à entrée et sortie numériques avec un puissant processeur de signal numérique.

Prix (membres) : 71,96 €

[www.elektor.fr/hifiberry-dac-dsp](http://www.elektor.fr/hifiberry-dac-dsp)

## JoyPi – kit d'expérimentation pour Raspberry Pi



Le JoyPi est un boîtier d'expérimentation animé par un Raspberry Pi. Il est idéal pour s'initier à l'électronique et la programmation. Le système de boîtier offre un environnement tout-en-un parfait, qui permet d'éviter la pagaille des composants et des câbles éparpillés sur la pailasse. Le JoyPi dispose d'un grand nombre de capteurs et de modules accessibles aux débutants.

Prix (membres) : 224,10 €

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



# Embedded in Embedded

(livre en anglais)

Ce livre est une étude de cas dans le domaine de la conception embarquée. Il aborde le matériel, l'initialisation du processeur, le développement de pilotes de bas niveau et la conception de l'interface d'application pour un produit. Les explications de l'auteur reposent sur une application spécifique d'une carte de développement Cortex-M3. Toutefois le lecteur acquerra des compétences essentielles pour devenir un excellent développeur de produits. La carte de développement présentée complète parfaitement le livre. La plateforme de travail ainsi créée pourra resservir par la suite.

Le programme universitaire Embedded in Embedded a pour but d'enseigner des compétences fondamentales utilisables dans n'importe quel environnement de conception.



Membre : 44,96 €

[www.elektor.fr/embedded-in-embedded](http://www.elektor.fr/embedded-in-embedded)

## Outil de finition chauffant pour impression 3D

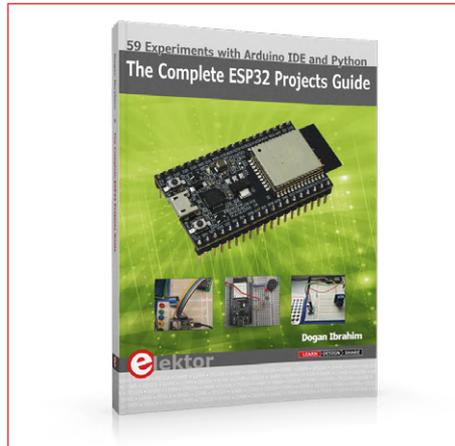


Cet outil chauffant de Modifi3D est muni d'embouts interchangeables avec un revêtement spécial. Il est conçu pour la finition, la réparation et la modification de pièces imprimées en 3D. Il permet de poncer, gratter... Il se présente sous la forme d'un stylo, ce qui le rend confortable à utiliser et très facile à manipuler pour atteindre une grande précision. Il permet de travailler sur du PLA, de l'ABS et d'autres matériaux d'impression en 3D.

 Prix (membres) : 31,46 €

[www.elektor.fr/modifi3d](http://www.elektor.fr/modifi3d)

## The Complete ESP32 Projects Guide



L'objectif de ce livre en anglais est d'enseigner la programmation avec l'EDI Arduino et le langage MicroPython à l'aide de projets basés sur l'ESP32, en utilisant la très populaire carte de développement DevKitC ESP32. Les nombreux projets sont de niveau de difficulté croissant. Ils ont tous été testés et fonctionnent. Les explications sont complétées par des schémas de principe, les schémas des montages et les programmes.

 Prix (membres) : 35,96 €

[www.elektor.fr/esp32-projects-guide](http://www.elektor.fr/esp32-projects-guide)

## Outil de mesure « mains libres » PCBite



PCBite est l'outil parfait pour manipuler un circuit imprimé pendant sa conception. Quatre puissants aimants posés sur un support en acier inoxydable permettent d'immobiliser des circuits imprimés de formes et de tailles différentes. Quatre bras également aimantés et flexibles sont terminés par une pointe de touche. Il suffit de les relier à un multimètre ou tout autre instrument de mesure pour garder les mains libres.

 Prix (membres) : 121,46 €

[www.elektor.fr/pcbite-kit](http://www.elektor.fr/pcbite-kit)

# Hexadoku casse-tête pour elektorniciens

Votre magazine se termine toujours et encore par une grille de sudoku particulière. Éteignez le fer à souder, oubliez les écrans de PC et d'oscilloscope, rangez les pinces et les fils. Faites travailler votre matière grise d'une autre façon : attrapez un crayon pour remplir la grille d'hexadoku. N'oubliez pas d'envoyer votre réponse pour tenter de gagner un bon d'achat.

Une grille hexadoku est composée de chiffres du système hexadécimal, de 0 à F. Remplissez le diagramme de 16 x 16 cases de telle façon que **tous** les chiffres hexadécimaux de 0 à F (0 à 9 et A à F) n'apparaissent **qu'une seule et unique fois** dans chaque rangée, colonne et carré de 4 x 4 cases (délimités par un filet gras).

Certains chiffres, déjà placés dans la grille, en définissent la situation de départ.

Pour participer, inutile de nous envoyer toute la grille, il suffit de nous donner **la série de chiffres** sur fond grisé.



### Participez et gagnez !

Nous tirons au sort **cinq** des réponses internationales correctes reçues dans les délais ; leurs auteurs recevront chacun un bon d'achat Elektor d'une valeur de **50 €**. À vos crayons !

### Où envoyer votre réponse ?

Envoyez votre réponse (les chiffres sur fond grisé) avec vos coordonnées par courriel, avant le **20 mai 2019** à l'adresse **hexadoku@elektor.fr**

### Les gagnants

La solution de la grille du numéro de mars/avril 2019 est **3BEF0**.

Les cinq bons Elektor d'une valeur de **50 €** vont à :

David **Creusot** (Australie), Lisa **de Blok** (Pays-bas), Gilbert **Luyckx** (Belgique), Tobias **Raulf** (Allemagne), Reuel **Victa** (Canada)

Bravo à tous les participants et félicitations aux gagnants !

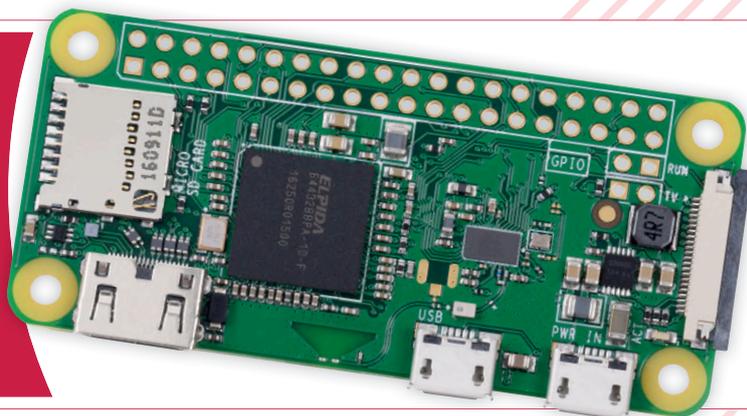
	A			5					F	7	D	9	2	6
D		1		6	9			E	5		3			C
	2	7		D		C					E			A
			8	E			4	D			3	5		7
A	7	4	B			D			2				5	1
	1						3	E	7					
		8	7			1			D	C	4		6	E
			F		6	A								7
	B								0	9		F		
3	D		E	A	B			1			6		7	
					E	1		3						9
8	9				4			C			1	D	E	3
F		2	D			3		8			0	6		
6			9					3		A		1	4	
5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

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

# ABONNEZ-VOUS ET RECEVEZ

## RPI ZERO W GRATUIT



Souscrivez dès maintenant un abonnement d'un an au magazine MagPi, nous vous offrons :

- Six numéros du magazine MagPi
- Une carte Raspberry Pi Zero W
- Un boîtier avec trois couvercles différents
- Un connecteur pour module de caméra
- Un câble HDMI/mini-HDMI et un câble micro-USB/USB OTG

SEULEMENT  
**54,95 €**  
PAR AN  
(6 NUMÉROS)

TOUS LES 2 MOIS, LES DERNIÈRES NOUVELLES DU RASPBERRY PI ET LES MEILLEURS PROJETS !

Vos avantages :

- Prix au numéro réduit
- Chaque numéro directement dans votre boîte aux lettres
- Tous les numéros disponibles sous forme numérique (PDF)
- Cadeau de bienvenue d'une valeur de 22,95 €
- Découverte de chaque nouveau numéro avant sa sortie en kiosque



# ABONNEZ-VOUS : [WWW.MAGPI.FR](http://WWW.MAGPI.FR)



**productronica**  
**fast forward**  
the start-up platform  
powered by Elektor

**PARTICIPEZ AU CONCOURS POUR  
LANCER VOTRE STARTUP**

**SUR LE SALON**

**PRODUCTRONICA 2019**

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

informations détaillées :  
[www.elektormagazine.fr/p-ffwd](http://www.elektormagazine.fr/p-ffwd)



Platinum Sponsor:  
**DISTRELEC**  
Distribution with a difference



**productronica**

**elektor**  
INNOVATION STARTUP TRADE

Productronica Fast Forward is brought to you by