

nouvelle formule !

mars / avril 2017 | n° 464

www.elektormagazine.fr

# elektor

DÉCOUVRIR

CRÉER

PARTAGER

## horloge RGBDigit



afficheur  
à 7 segments  
en couleur

Dans cette édition :  
8 projets du labo  
7 projets de lecteur  
Elektor Uno R4,  
Gnublin 2,  
tests NIST ...

passerelle IoT et nœuds sans fil •

kits d'initiation & cartes de développement

• signaux de télécommande (format NEC) •

R. Lacoste – LoRaWAN • Swiss Pi - extensions

matérielles • banc d'essai : STEMLab 125-10 • thérémin avec des JFET •

Android sur Raspberry Pi • générateur de nombres vraiment aléatoires

• RFID pour Arduino Uno • commande de vitesse par MLI • MicroPython

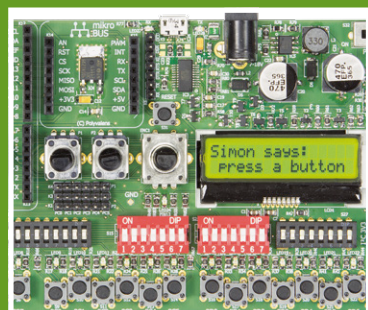
• sonde différentielle active V2 • carte Linux à monter soi-même •

régénérateur de batteries au plomb



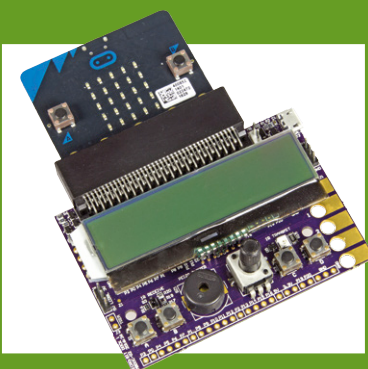
testeur portatif OBD2

avec Raspberry Pi  
nouveau logiciel de diagnostic



AVR Playground

carte de développement  
« façon Arduino »



BBC micro:bit

station d'accueil  
espace de prototypage

ISSN 0181-7450

L 19624 - 464 - F: 15,50 € - RD



(B) 16,50 € • (CH) 29,00 FS • (CAN) 20,99 \$ca • (And) 15,50 €  
DOM surface 16,50 € • DOM avion 16,90 € • N Cal/S 2000 cfp

« Votre achat est amorti  
très rapidement »



**FESTIVAL  
DES PRIX**

VT-7126

**0,58**



**A+**

### AMPOULE LED, GU 10

- Puissance : 3 watts
- Puissance lumineuse : 210 lm
- Couleur de la lumière : blanc chaud
- Durée de vie : 20 000 h
- Angle de rayonnement : 110°
- CRI (Ra) : 80

**RENOUVELER L'ÉCLAIRAGE ET ÉCONOMISER MAINTENANT**

### RUBAN LED

- Puissance : 18 watts
- Angle de rayonnement : 120°
- Longueur : 5 m ■ Largeur : 8 mm
- Installation facile et modulable
- CRI (Ra) : 80



**A**

l'unité **4,50**

VT-2016	4,50	blanc chaud
VT-2041	4,50	blanc neutre
VT-2005	4,50	blanc froid

### AMPOULE LED, E 27

Ces ampoules LED peuvent remplacer pratiquement toutes les lampes à économie d'énergie et lampes classiques sur un culot E27. Vous pouvez ainsi tout de suite commencer à économiser l'énergie.

- Puissance : 10 watts
- Puissance lumineuse : 806 lm
- Couleur de la lumière : blanc chaud
- Durée de vie : 20 000 h
- CRI (Ra) : 80



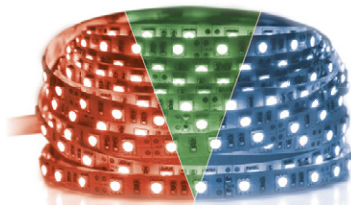
**A+**

VT-4209

**1,42**

### RUBAN LED

- Puissance : 18 watts
- Angle de rayonnement : 120°
- Longueur : 5 m ■ Largeur : 8 mm
- Installation facile et modulable



**A**

à partir de **4,50**

VT-2015	4,50	rouge
VT-2011	4,50	vert
VT-2013	4,50	bleu
VT-2124	6,68	RGB

**PLUS DE TECHNIQUE LUMINEUSE LED**

Vous trouverez les articles de cette annonce et bien d'autres encore sur notre boutique en ligne: <http://rch.it/zw>





40<sup>ème</sup> année,  
n° 464 – mars/avril 2017

ISSN 0181-7450  
Dépôt légal : février 2017  
CPPAP 1113 U 83713  
Directeur de la publication : Donatus Akkermans

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

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

Tél. : (+33) 01.49.19.26.19  
du lundi au vendredi de 10h à 13h

Fax : (+33) 01.49.19.22.37

[www.elektor.fr](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 :

Fabio Romagnoli +32 485 65 40 90  
[fabio.romagnoli@eimworld.com](mailto:fabio.romagnoli@eimworld.com)

#### DROITS D'AUTEUR :

© 2017 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.



## Attention : peinture fraîche !

Cet hiver, votre magazine s'est refait une beauté. Il y a du nouveau tant dans la forme que dans le fond.

La formule révisée d'Elektor ne se résume pas à un changement de format, maintenant A4 ; la mise en page des articles a également évolué. Vous constaterez surtout que le contenu est recentré sur l'essentiel : **les projets** ! Que ce soient ceux du labo d'Elektor ou ceux de nos lecteurs, ils occupent de nouveau la plus grosse part du magazine. Chaque description de projet du labo est enrichie d'un pavé qui donne sur les montages des informations synthétiques importantes : niveau de connaissance souhaitable, coût, matériel recommandé, durée de réalisation.

Certaines rubriques ont laissé la place à de nouvelles comme le « vol tous azimuts » piloté par Clemens Valens, chef du labo. D'autres fleuriront dans les prochains numéros. Le printemps arrive !

Comme après toute période de travaux, il reste les finitions, des retouches ici et là, des réglages à affiner... N'hésitez pas à nous faire part de vos impressions et suggestions ([redaction@elektor.fr](mailto:redaction@elektor.fr)).

Si vous êtes adepte du papier imprimé et familier des marchands de journaux, vous trouverez désormais sur [www.elektormagazine.fr](http://www.elektormagazine.fr) un outil pour trouver près de chez vous les kiosques approvisionnés en magazines Elektor.

*Elektor, à votre service.*

### Mariline Thiebaut-Brodier

#### 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 :	Ton Giesberts, Luc Lemmens, Clemens Valens (responsable), Jan Visser
Coordination :	Hedwig Hennekens
Ont coopéré à ce numéro :	Patrick Bechler, Pascal Duchesnes, Yves Georges, Robert Grignard, Denis Lafourcade, Guy Matt, Jean-Louis Mehren, Denis Meyer, Helmut Müller, Kévin Petit, Xavier Pfaff, Guy Raedersdorf, Alexandre Roy
Service de la clientèle :	Cindy Tijssen
Graphistes :	Giel Dols, Mart Schroijsen, Patrick Wielders
Elektor en ligne :	Daniëlle Mertens

## 31 bruits de labo...

## 47 agenda

mars-avril 2017

## 64 vol tous azimuts

piloté par Clemens Valens

## 76 projet 2.0

corrections, mises à jour et courrier des lecteurs

## 96 le Fetron, un tube à transistors

drôle de composant n°29

## 112 sur la toile : conception d'enceintes

programmes gratuits pour le calcul d'enceintes

## 128 l'e-choppe d'Elektor

## 130 Hexadoku

casse-tête pour elektorniciens

## en coulisse

### 12 kits d'initiation et cartes de développement

petit tour d'horizon

### 42 hors-circuits avec R. Lacoste

LoRaWAN : architecture, protocole, sécurité et opérateurs

### 60 débogage et décodage de communications numériques

à l'aide du SmartScope

### 66 banc d'essai : STEMLab 125-10

la petite sœur de la Red Pitaya originale

### 78 logiciel - Android sur Raspberry Pi (1)

mesurer, commander, réguler : les broches GPIO

### 84 ForumLED Europe

de l'Internet des Objets à l'Internet de la Lumière

### 108 kit d'initiation RFID pour Arduino Uno

thermomètre et contrôle d'accès

### 118 MicroPython

le Python des petits systèmes

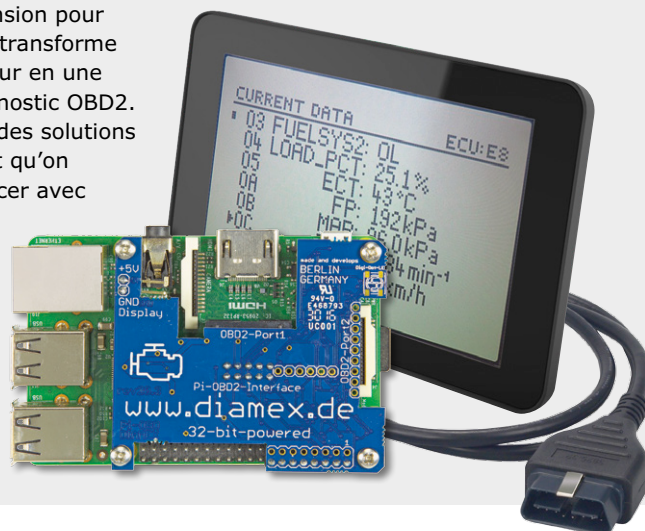
# 6



## testeur portatif OBD2 à base de Raspberry Pi avec un nouveau logiciel de diagnostic

# 26

Le module Pi-OBd-HAT est une carte d'extension pour Raspberry Pi qui transforme ce nano-ordinateur en une interface de diagnostic OBD2. Un inconvénient des solutions précédentes était qu'on devait les interfacer avec un émulateur de terminal. Le logiciel de diagnostic HHGui présenté ici est d'un emploi bien plus simple et commode.



## sur la scène : les projets

### 6 horloge RGBDigit

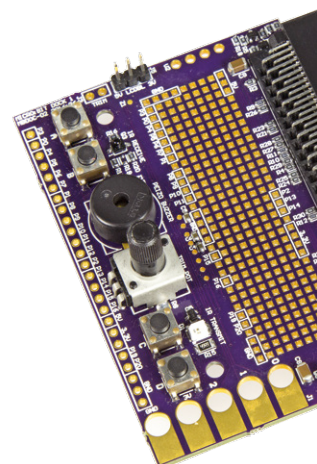
afficheur à 7 segments en couleur

### 18 passerelle IoT et nœuds sans fil

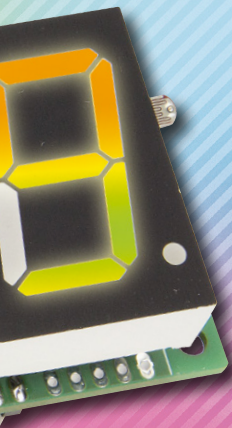
1<sup>ère</sup> partie : le matériel

### 23 signaux de télécommande au format NEC

exclusivement pour Elektor Uno R4

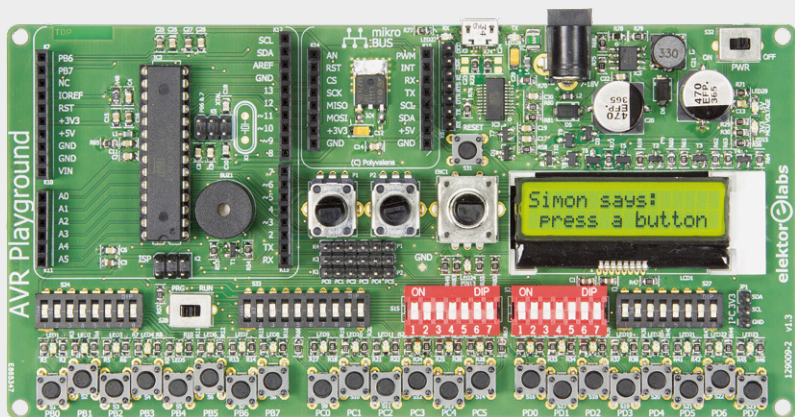






# horloge RGBDigit

## afficheur à 7 segments en couleur

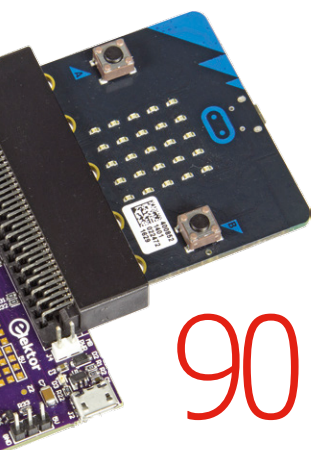


## AVR Playground

ou comment faire mieux « façon Arduino »

Cet article présente l'AVR Playground, une carte hybride entre l'Arduino Uno et une carte classique de développement pour microcontrôleur. L'objectif est de « faire de l'Arduino » sans mauvaises connexions ni fils volants (même si la carte ne l'empêche pas).

52



90

**26** testeur portatif OBD2  
à base de Raspberry Pi  
avec un nouveau logiciel de diagnostic

**32** Swiss Pi toujours plus polyvalent  
grâce aux extensions matérielles

**48** fabriquer son thérémin  
des JFET à la place des tubes

**52** AVR Playground  
ou comment faire mieux « façon Arduino »

**68** générateur de  
nombres réellement aléatoires  
tests NIST & composants abordables

**90** station d'accueil pour BBC micro:bit  
« base à micro:bit :  
vous êtes autorisé à vous poser »

**97** commande de vitesse par MLI  
avec régulation du rapport cyclique

**100** capteurs (3)  
pour Arduino et Cie

**106** sonde différentielle active V2  
alimentée maintenant par USB

**114** carte Linux  
alternative à Raspberry Pi  
à monter soi-même

**120** régénérateur  
de batteries au plomb 0-30 V  
avec contrôle de  
la qualité de l'accumulateur



bientôt sur ces pages

### Extrait du sommaire du prochain numéro :

- Analyseur de signaux universel par Ethernet, la suite
- Chargeur Li-Ion
- Détecteur de caméra-espionne
- Éclairage extérieur
- L-Board, T-Board réinventée
- nWatch, électronique vestimentaire
- Préamplificateur pour micro de studio
- Supervision en 3D avec des moteurs pas-à-pas
- Supra 2.0, la suite
- Télécommande XBee
- Etc.

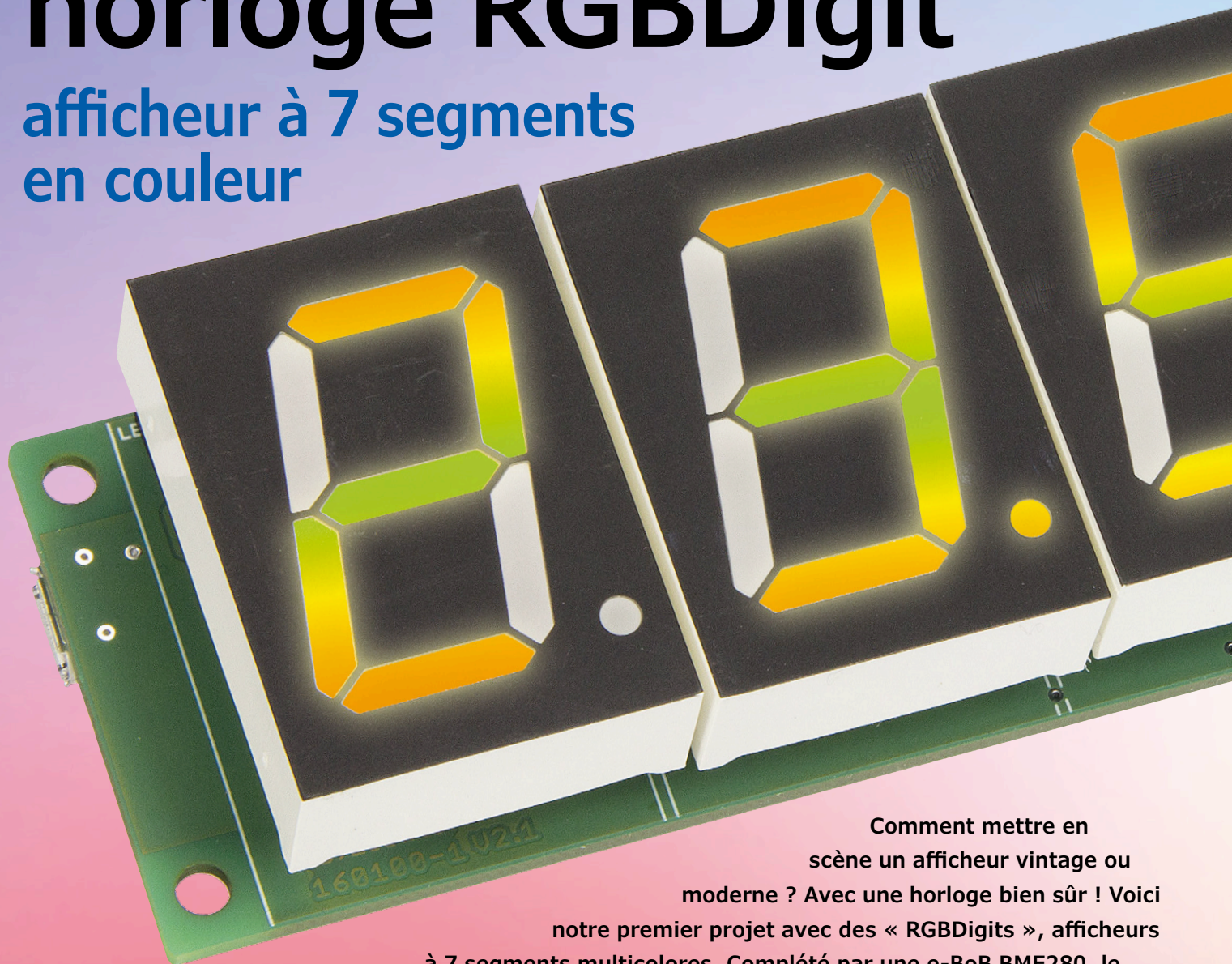
Sous réserve de modification.

Le numéro de mai-juin paraîtra le 19 avril 2017.



# horloge RGBDigit

afficheur à 7 segments  
en couleur



Comment mettre en scène un afficheur vintage ou moderne ? Avec une horloge bien sûr ! Voici notre premier projet avec des « RGBDigits », afficheurs à 7 segments multicolores. Complété par une e-BoB BME280, le montage affichera température, humidité et pression atmosphérique.

**Coen de Bruijn** (Pays-Bas)

L'horloge est pilotée par un module de transmission ESP12, ce qui permet la synchronisation avec un serveur de temps de l'internet, les réglages depuis un terminal mobile ou un ordinateur sur le réseau et la transmission par Wi-Fi des données du capteur.

## Qu'est-ce que c'est ?

Les RGBDigits sont fabriqués par une société hollandaise éponyme. Chaque

afficheur comprend huit LED RVB NeoPixel 5050 (sept segments plus le point décimal) avec pilotes intégrés, l'utilisateur commande la couleur et la luminosité de chaque segment individuellement via un bus à 3 fils (VCC, GND et DATA). Il est possible de réunir jusqu'à dix afficheurs, via leurs broches DATA IN et DATA OUT.

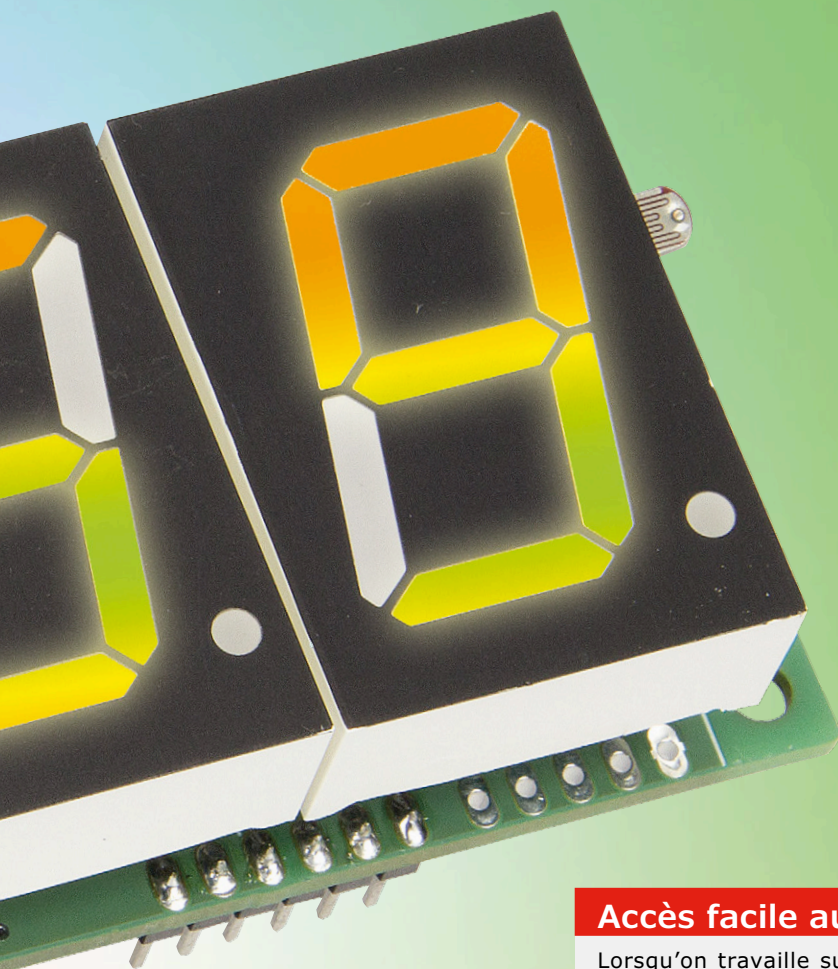
Chaque LED de couleur primaire d'un NeoPixel est réglable sur 256 niveaux de luminosité, d'où une énorme palette de

$256 \times 256 \times 256 = 16.777.216$  couleurs par segment.

## Fonctionnement du circuit

Le schéma (**fig. 1**) montre que l'horloge est alimentée en 5 V par le connecteur micro-USB K1. L'alimentation est protégée par le fusible réarmable CTP 2 A F1, avec la diode Schottky D1 en protection contre l'inversion de polarité. IC1 est le régulateur de tension principal de 3,3 V et IC6 alimente en 3,3 V les capteurs Qtouch IC4 et IC5 (la séparation des





alimentations évite les interférences).

Les deux boutons tactiles Button0 et Button1 servent à commander les modes d'affichage de l'horloge. S1 et S2 ne servent qu'à la RàZ de l'horloge et la programmation du micrologiciel.

IC2 est un convertisseur de niveau unidirectionnel : 3,3 V côté ESP12E → alim. V+ des afficheurs (env. 4,5 V). L'EEPROM IC3 conserve les réglages de l'horloge après extinction ; la photorésistance R1 baisse la luminosité des afficheurs dans la pénombre.

Il est possible de connecter un cordon de type FTDI 3,3 V (ou tout autre UART USB 3,3 V) à K2 pour programmer l'ESP12E ou déboguer le logiciel.

Le connecteur K3 donne accès à l'alimentation 3,3 V et à trois broches d'E/S d'usage général de l'ESP12E pour vos propres projets et/ou futures extensions. Une carte de liaison BME280 (e-choppe, réf. 160109-91, **fig. 2**) peut être connectée à K4. L'horloge fonctionnera sans, mais les données du capteur BME280 ne seront pas disponibles. Il était impossible d'embarquer ce capteur – la chaleur des afficheurs aurait faussé les mesures.

## À PROPOS DU PROJET



### Test & Mesure

RVB LED  
horloge capteurs



débutant

→ **connaissseur**

expert



environ 3 h



Ordinateur avec  
EDI Arduino



environ 100 €

## Accès facile aux données du capteur

Lorsqu'on travaille sur un projet comme celui-ci, et d'ailleurs sur tout projet de l'IdO, il est parfois utile d'avoir accès à un serveur de stockage. Les techniciens de Sparkfun l'ont compris et ont mis en place un service gratuit et ouvert à tous. Un serveur accessible par [data.sparkfun.com](https://data.sparkfun.com) stockera et (partagera !) vos données de manière pratique. Il fonctionne comme une sorte de base de données et vous permet de stocker et examiner vos données (et celles d'autres) dans un tableau (voir la **copie d'écran**).

Commencez par créer un flux de données en cliquant sur *create*. Complétez les champs, vous recevrez une clé publique, une clé privée et une clé d'effacement. Ces clés servent à envoyer des données dans le flux ou à les effacer. Notre horloge aura besoin des clés publique et privée pour envoyer des données au serveur de données de Sparkfun.

De là vous pouvez d'un clic (en haut à droite) exporter vos données vers *analog.io*, un service en ligne d'affichage de données (IdO). Vous obtiendrez avec *analog.io* un joli graphique de vos données et/ou mesures.

RGB clock - Sensor RGB clock which keeps track of time, temperature (°C), humidity (%) and airpressure (hPa).

data.sparkfun.com

time	time	time	time
2017.27	68.07	16.26	2019-12-07T19:23:23Z
1017.29	68.12	16.25	2019-12-07T19:24:14Z
1017.33	68.16	16.25	2019-12-07T19:25:06Z
1017.26	68.11	16.26	2019-12-07T19:26:17Z
1017.31	68.13	16.26	2019-12-07T19:27:08Z
1017.29	68.09	16.26	2019-12-07T19:28:00Z
1017.26	68.13	16.26	2019-12-07T19:28:52Z
1017.21	68.21	16.25	2019-12-07T19:29:44Z
1017.29	68.29	16.23	2019-12-07T19:30:36Z
1017.17	68.10	16.21	2019-12-07T19:31:28Z
1017.16	68.37	16.20	2019-12-07T19:32:20Z
1017.18	68.37	16.19	2019-12-07T19:33:12Z
1017.04	68.45	16.18	2019-12-07T19:34:04Z



## Programmer le croquis

Le module Wi-Fi ESP12E qui contient un ESP8266 est pris en charge par l'EDI Arduino. Pour charger le logiciel dans le module, il faut sélectionner l'ESP8266 (NodeMCU) dans l'EDI Arduino. Avant de continuer, suivez attentivement les

instructions pour ajouter l'ESP8266 (NodeMCU) qui se trouvent en [1].

Téléchargez en [2] l'archive de ce projet (160100-11.zip). Décompressez le fichier dans un dossier de croquis de votre disque dur. Attention : le sous-dossier

DATA (qui contient la page web de l'ESP12) doit rester dans ce dossier.

Maintenant suivez ces étapes :

1. reliez votre ordinateur à K2 sur l'horloge RVB avec un cordon FTDI 3,3 V (!) ;

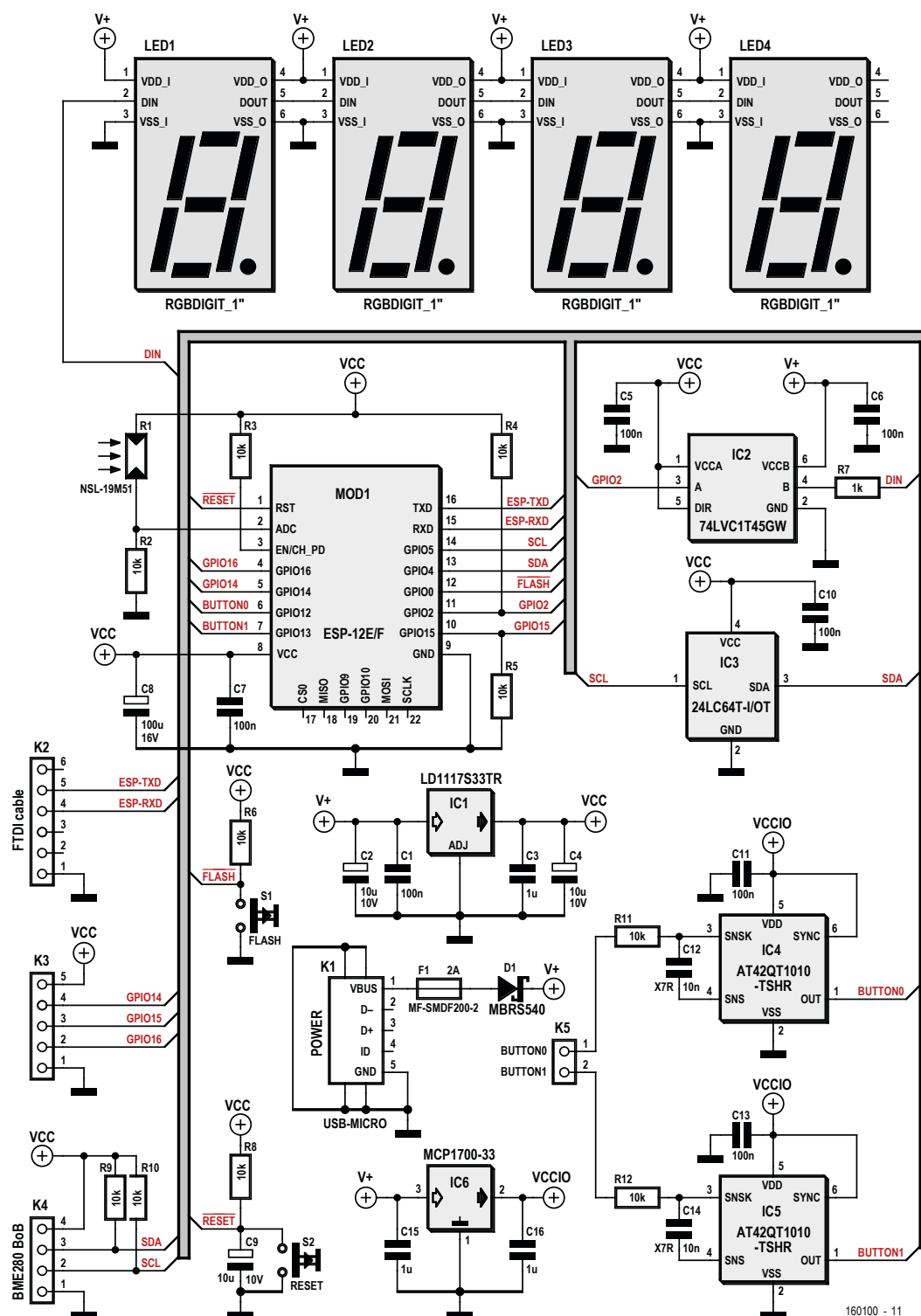


Figure 1. Que de connexions ! Pour garder le schéma lisible, beaucoup d'entre elles sont regroupées dans un « bus »



- connectez une alimentation de 5 V au connecteur micro-USB K1 ;
- ouvrez l'EDI Arduino ;
- dans le menu **Outils → Type de carte**, sélectionnez la carte **NodeMCU 1.0 (ESP-12E Module)** ;
- dans le menu **Outils → Port**, sélectionnez le port COM associé au cordon FTDI ;
- dans le menu **Fichier → Ouvrir**, sélectionnez le croquis `esp12_rgb_clock.INO` inclus dans notre archive ;
- sur la carte de l'horloge RVB, pressez et maintenez enfoncé le bouton de RàZ S2, idem avec le bouton de programmation S1, relâchez S2 puis S1 pour autoriser la programmation du micrologiciel ;
- compilez et téléchargez le croquis (CTRL+U).

Une sortie du compilateur avec une erreur sera probablement due à une ou plusieurs bibliothèques Arduino manquantes. Le compilateur indiquera le nom de la première bibliothèque manquante, mais vérifiez aussi que toutes les bibliothèques utilisées dans les directives `INCLUDE` du croquis `esp8266_rgb_clock` sont installées dans votre EDI Arduino avant de lancer le compilateur.

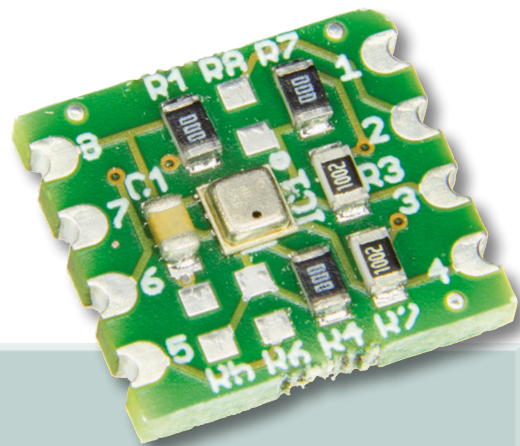
## Charger les pages web de l'horloge

On peut effectuer tous les réglages de l'horloge par Wi-Fi. La connexion initiale doit être faite via le Point d'Accès (PA) de l'ESP8266. Connectez votre ordiphone/tablette/portable (ou PC avec connexion Wi-Fi directe) au réseau ouvert nommé « RGB Clock » qui devrait apparaître dans la liste des réseaux Wi-Fi disponibles après avoir programmé le micrologiciel de l'horloge. Toutefois, il faut d'abord charger les pages web de l'horloge dans l'ESP8266 pour accéder facilement aux réglages de l'horloge.

Vous trouverez comment faire en [3] sous « *Uploading files to the file system* » :

- téléchargez en [4] l'outil « Arduino ESP8266 filesystem uploader » ;
- décompressez l'outil dans le répertoire `tools` (le chemin devrait ressembler à `<home_dir>/Arduino/tools/ESP8266FS/tool/esp8266fs.jar`) ;
- redémarrez l'EDI Arduino ;
- ouvrez le croquis `esp12_rgb_clock.INO` (s'il ne s'ouvre pas

Figure 2. Notre e-BoB BME280 complète l'horloge RGBDigit avec température, pression et humidité.



## LISTE DES COMPOSANTS

### Résistances

R1 = photorésistance NSL-19M51\*  
R2,R3,R4,R5,R6,R8,R9,R10,R11,R12 = 10 kΩ  
R7 = 1 kΩ

### Condensateurs

C1,C5,C6,C7,C10,C11,C13 = 100 nF, 50 V, X7R, 0805  
C2,C4,C9 = 10 μF, 10 V, tantale, 1206  
C3,C15,C16 = 1 μF, 50 V, X5R, 0805  
C8 = 100 μF, 16 V, 2312  
C12,C14 = 10 nF, 50 V, X7R, 0805

### Semi-conducteurs

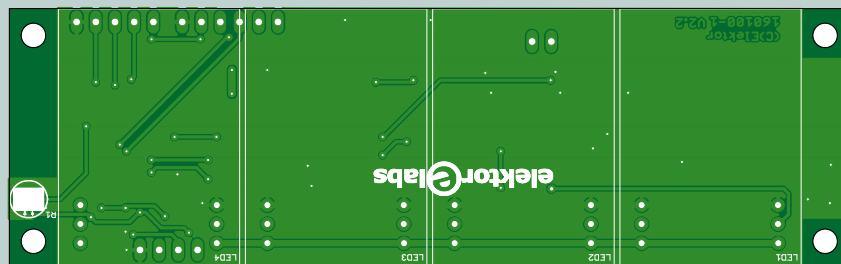
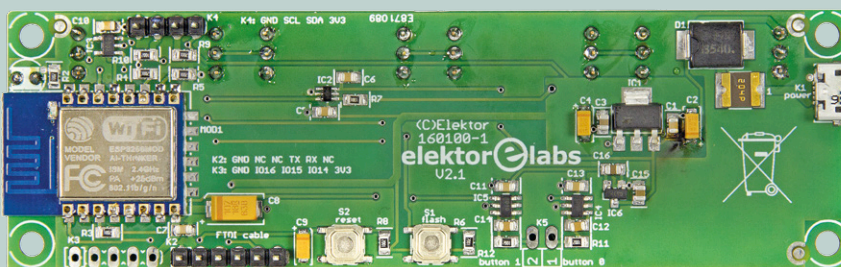
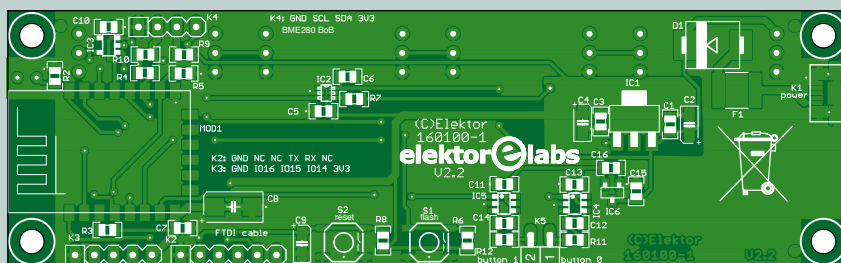
D1 = MBR540  
LED1,LED2,LED3,LED4 = afficheurs à 7 segments RVB\*  
IC1 = LD1117S33TR  
IC2 = 74LVC1T45GW  
IC3 = I2C EEPROM 8K × 8 bits, type 24LC64T-I/OT  
IC4,IC5 = bouton tactile Qtouch, type AT42QT1010-TSHR

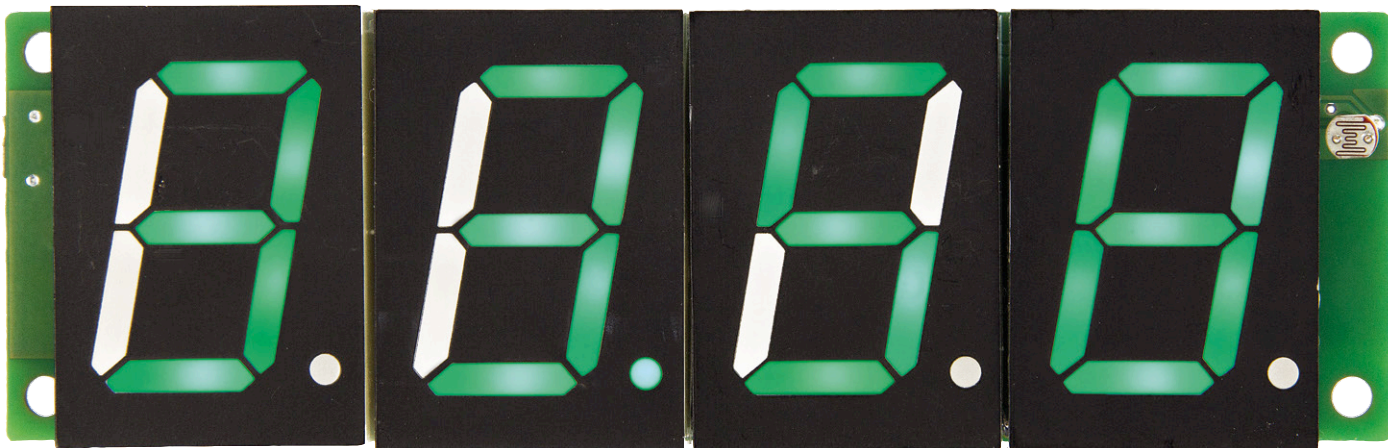
IC6 = MCP1700T-3302E/TT

### Divers

F1 = fusible réarmable CTP 2 A, type MF-SMDF200-2 (Bourns)  
K1 = embase micro-USB, type B, montage au-dessous  
K2 = barrette mâle, 1 × 6 broches, au pas de 2,54 mm, coudée  
K3 = barrette mâle, 1 × 5 broches, au pas de 2,54 mm, coudée  
K4 = barrette mâle, 1 × 4 broches, au pas de 2,54 mm, droite  
K5 = barrette mâle, 1 × 2 broches, au pas de 2,54 mm, coudée  
MOD1 = ESP8266-12F  
S1 = interrupteur tactile  
Circuit imprimé, réf 160100-1

\*montage sur la face inférieure du circuit imprimé





- automatiquement) ;
5. assurez-vous d'avoir sélectionné la bonne carte et le bon port COM, et fermé le moniteur série ;
  6. pressez et maintenez enfoncé S2, puis idem avec S1, relâchez S2 puis S1 pour autoriser le chargement des pages web ;
  7. sélectionnez **Outils → ESP8266 Sketch Data Upload**. Le transfert dans le système de fichiers de l'ESP8266 des fichiers issus notre dossier *data* devrait démarrer. Une fois terminé, la barre d'état de l'EDI affichera « SPIFFS Image Uploaded ».

Si vous avez envie de modifier la page web de l'horloge, c'est possible. Voir l'encadré **Comment créer/modifier une page web pour l'ESP12**.

### Maintenant amusons-nous !

Sur votre PC/tablette/ordiphone, rejoignez le réseau Wi-Fi « RGB Clock », connectez-vous à 192.168.4.1:81 depuis

un navigateur. Dans le menu (en haut en gauche), sélectionnez **Wi-Fi settings** et entrez l'identifiant et le mot de passe de votre réseau sans fil. L'horloge utilise ce réseau pour se connecter au serveur de temps (NTP) et se mettre à l'heure. Le serveur [time.nist.gov](http://time.nist.gov) est défini dans le croquis comme serveur NTP pour votre horloge. L'horloge enverra aussi les données de son capteur à un serveur chez Sparkfun (voir encadré **Accès facile aux données du capteur**).

### Quelques remarques :

- Comme le point d'accès de l'ESP12 reste actif après avoir changé les réglages Wi-Fi, vous pouvez toujours les modifier (ainsi que les autres réglages) avec votre terminal mobile connecté au réseau « RGB Clock ».
- Les pare-feu – comme celui des bureaux d'Elektor – peuvent bloquer le protocole NTP d'un serveur de temps sur l'internet, ce qui rend impossible la synchronisation

automatique de l'horloge RVB. Pour contourner ce problème, utilisez un ordiphone configuré en borne Wi-Fi pour récupérer les données de temps pour l'horloge. Une fois synchronisée, elle fonctionnera sans données NTP. Autre solution : réglez l'heure manuellement sur la page web de l'horloge.

- Pour connecter l'horloge à un réseau non sécurisé (c.-à-d. sans mot de passe), cochez la case « Unprotected Wi-Fi Network » dans les réglages Wi-Fi de l'horloge.

Tout terminal avec un accès à votre routeur Wi-Fi peut maintenant accéder à l'horloge pour modifier ses réglages. Il suffit de taper « clock.local:81 » dans la barre d'adresse d'un navigateur pour arriver sur les pages web de l'horloge. ◀

(160100 – version française : Denis Lafourcade)

## Comment créer/modifier une page web pour l'ESP12 ?

Nous avons utilisé HTML-Kit 292 pour construire le site web interne de l'ESP12 ([www.htmlkit.com/download](http://www.htmlkit.com/download)). Ce programme libre permet de créer et concevoir son propre site. Notre page web comporte trois fichiers : index.html, style.css et script.js. Index.html est le fichier principal, programmé en HTML. Il constitue la structure du site. Dans style.css, nous définissons l'aspect de notre page. Script.js détermine ce que fait la page.

La page de l'ESP12 est constituée de plusieurs sections <div>. La mise en page et le fonctionnement sont définis dans *style.css* et *script.js* à partir de ces éléments.

### Liens

- [1] [www.arduinesp.com/getting-started](http://www.arduinesp.com/getting-started)
- [2] [www.elektormagazine.fr/160100](http://www.elektormagazine.fr/160100)
- [3] <https://github.com/esp8266/Arduino/blob/master/doc/filesystem.md>
- [4] <https://github.com/esp8266/arduino-esp8266fs-plugin/releases/download/0.2.0/ESP8266FS-0.2.0.zip>



### DANS L'E-CHOPPE

- 160100-1 Circuit imprimé
- 160100-91 Module assemblé
- 160109-91 Carte e-BoB BME280 assemblée (en option)
- 17789 Afficheur RGBDigit à 7 segments



# la communauté d'Elektor

DÉCOUVRIR CRÉER PARTAGER

82

pays

248153

membres actifs

1040

experts & auteurs

489

publications

235332

visiteurs (par mois)

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



**Boutique en ligne d'Elektor :** la caverne d'Ali Baba des électroniciens ! 10 % de remise permanente pour les membres GREEN et GOLD.  
[www.elektor.fr](http://www.elektor.fr)



**Magazine Elektor :** six numéros par an d'électronique inédite (projets, bancs d'essais, cours, trucs et astuces, actualité...)  
[www.elektormagazine.fr](http://www.elektormagazine.fr)



**Elektor PCB Service :** gravure à la demande de vos circuits imprimés, en un ou plusieurs exemplaires. [www.elektorpcbservice.com](http://www.elektorpcbservice.com)



**Elektor GRATUIT, version numérique :** lettre d'information à l'affût des nouveautés. C'est gratuit : [www.elektor.fr/inscription](http://www.elektor.fr/inscription)



**Elektor Academy :** l'apprentissage par la pratique au travers de webinaires, séminaires, ateliers, DVD... [www.elektor.fr](http://www.elektor.fr)



**Livres & DVD Elektor :** Arduino, Raspberry Pi, microcontrôleurs, Linux et bien d'autres. 10 % de remise pour les membres !  
[www.elektor.fr](http://www.elektor.fr)



**Elektor TV :** présentations, comptes-rendus, déballage et évaluation de produits. Pour y voir plus clair.  
[www.youtube.com/user/ElektorIM](http://www.youtube.com/user/ElektorIM)



**Elektor Labs :** l'incubateur où partager ses projets et participer à ceux des autres. Nous faisons grandir vos idées !  
[www.elektormagazine.fr/labs](http://www.elektormagazine.fr/labs)

## connectez-vous à notre communauté

### formule

#### GREEN 92,50 € par an

- ✓ accès à l'archive d'Elektor
- ✓ 10% de remise dans l'e-shoppe
- ✗ 6x magazine imprimé
- ✓ 6x magazine numérique
- ✓ des offres exclusives
- ✓ accès à plus de 1 000 fichiers Gerber
- ✗ le DVD annuel d'Elektor

connectez-vous

[www.elektor.fr/formule-greencard](http://www.elektor.fr/formule-greencard)

### formule

#### GOLD 127,50 € par an

- ✓ accès à l'archive d'Elektor
- ✓ 10% de remise dans l'e-shoppe
- ✓ 6x magazine imprimé
- ✓ 6x magazine numérique
- ✓ des offres exclusives
- ✓ accès à plus de 1 000 fichiers Gerber
- ✓ le DVD annuel d'Elektor

connectez-vous

[www.elektor.fr/formule-goldcard](http://www.elektor.fr/formule-goldcard)

### formule

#### GRATUITE

- ✗ accès à l'archive d'Elektor
- ✗ 10% de remise dans l'e-shoppe
- ✗ 6x magazine imprimé
- ✗ 6x magazine numérique
- ✓ des offres exclusives
- ✗ accès à plus de 1 000 fichiers Gerber
- ✗ le DVD annuel d'Elektor

connectez-vous

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



[facebook.com/ElektorFR](https://facebook.com/ElektorFR)



[twitter.com/Elektor](https://twitter.com/Elektor)

# kits d'initiation et cartes de développement

## petit tour d'horizon

Il y a juste un an, nous faisons ici même un tour du monde des nouvelles cartes à microcontrôleurs. Nous sommes repartis en voyage et nous avons rencontré de nouvelles cartes d'initiation et plateformes de développement. Attention, cette liste est loin d'être exhaustive.

**Viacheslav Gromov** (Allemagne),  
readers@gromov.de

Les cartes d'initiation (ou d'embarquement) intéressantes et autres kits de démarrage découverts l'an dernier [1] ne se distinguaient pas tellement par leurs avancées technologiques, parce qu'ils restent très universels et produits en grande série. Mais cela ne signifie pas que les grandes plateformes ne conviendraient pas pour une initiation, bien au contraire. Mieux vaut souvent commencer sur des cartes à microcontrôleur plus spécifiques, plus complexes et dès lors peut-être aussi plus professionnelles. Ou bien dénicher un environnement de développement taillé sur mesure pour de futures applications. Plus le démarrage est difficile, mieux on en apprend. D'autre part, il sera ensuite plus facile de passer à des microcontrôleurs plus petits et moins puissants que de faire le contraire.

C'est pourquoi ce tour d'horizon dans le monde des microcontrôleurs nous conduira vers des cartes plus exotiques ou particulièrement récentes qui sont toutes très différentes et souvent vouées à des domaines d'application spécialisés. Du coup, leurs prix fluctuent d'une dizaine d'euros à plusieurs centaines d'euros. Les paragraphes qui suivent sont ordonnés en fonction du prix. Ce n'est pas une revue des environnements de développement intégré (EDI) standard de chaque famille de microcontrôleurs, mais seulement les nouveautés en matière de logiciel ou des outils (supplémentaires) tout neufs. Certainement de quoi trouver quelque chose pour votre prochain projet !

Avant d'y aller, je voudrais vous prévenir qu'il ne sera pas question ici du ras de

marée des cartes IdO à périphériques sans fil. Nous leur consacrerons prochainement tout un article.

### Mini-boîtes d'allumettes

C'est une évolution de la carte XMC 2Go présentée l'an dernier, basée sur le bien équipé XMC1100 (M0+, 48 MHz, 64 Ko). À laquelle s'ajoutent trois versions avec capteur de courant SPI TLI4970 (**Current Sensor 2Go**, environ 28 €, [2]), capteur de champ magnétique 3D I<sup>2</sup>C TLV493D-A1B6 (**3D Magnetic Sensor 2Go**, environ 27 €, [2]) et puces de pont en H SPI IFX9201SG (**H-Bridge Kit 2Go**, environ 13 €, [3]) (**figure 1**). Chaque capteur a ses caractéristiques propres. Le TLI4970 mesure le courant dans un conducteur au moyen de deux capteurs à effet Hall. Du fait de la mesure différentielle employée, le résultat est insensible aux parasites extérieurs. Le signal (jusqu'à  $\pm 50$  A, 12 bits, CA/CC) est numérisé en interne et nettoyé par un filtre réglable avec un DSP et traité de manière à correspondre le plus linéairement possible au courant mesuré. Le TLV493D-A1B6 mesure le champ magnétique environnant jusqu'à  $\pm 130$  mT (milli tesla) sur trois axes

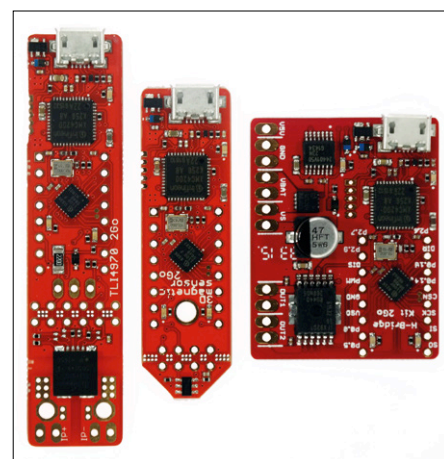


Figure 1. Des capteurs 2Go : de courant, de champ magnétique 3D et le kit de pont en H 2Go.

avec une résolution de 12 bits chacun. En outre, il contient un thermomètre pour compléter la mesure. Son interface vers un microcontrôleur est du type I<sup>2</sup>C, ce qui permet de le loger dans un très petit boîtier à 6 contacts TSOP.

La puce IFX9201SG est celle des trois qui n'a pas de capteur, en revanche, elle constitue un pont en H typique pour les petits moteurs à CC ou d'autres charges inductives jusqu'à 6 A. La commande

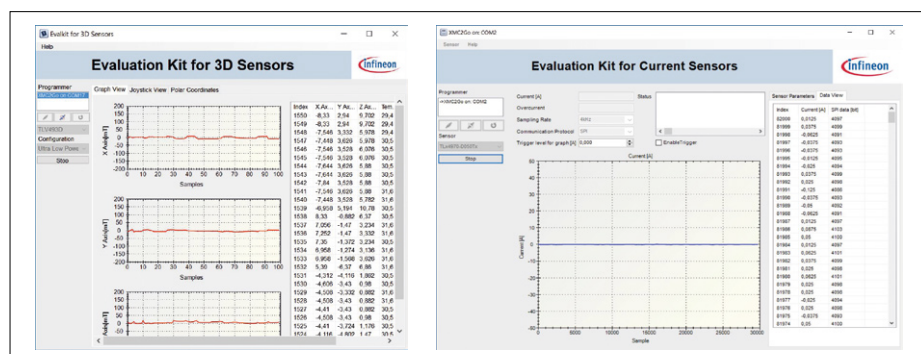


Figure 2. Pour les deux capteurs 2Go, on dispose d'outils logiciels compacts pour Windows, ils permettent d'essayer déjà ses propres programmes.



se fait principalement par la broche de direction et l'entrée MLI dont la fréquence peut monter à 20 kHz. On peut la surveiller par son interface SPI.

Les cartes ont en commun, outre la base XMC, de posséder les sorties importantes vers le microcontrôleur et toujours deux LED disponibles. Une particularité des deux cartes à capteur, c'est de pouvoir le séparer de la carte. Ce n'est pas possible avec le kit de pont en H. Aussi, la carte dispose d'un connecteur d'alimentation pour simplifier la construction de l'application. Au capteur de champ magnétique, on peut ajouter une manette de jeu ou un codeur rotatif qui fait alors du capteur un organe de commande variable en continu. Il existe pour chaque 2Go quelques bibliothèques de base d'emploi facile et des projets pour le célèbre environnement DAVE, en plus des outils logiciels accessoires pour les capteurs (figure 2).

### Avec écran tactile pour 25 €

STMicroelectronics a toujours été connu pour ses cartes de développement très bon marché, mais pourtant bien équipées. Mais ST s'est encore dépassé. Le kit **STM32F412G Discovery** (figure 3) [4], fondé sur un nouveau représentant de la famille STM32F4 à 32 bits ARM-Cortex-M4, le STM32F412ZG (100 MHz, 1 Mo), offre de sublimes atouts : écran tactile TFT, une sortie audio avec puce de codec et amplificateur en classe D, deux microphones numériques MEMS, une mémoire additionnelle de quatre fois 128 Mbits, un socle micro-SD et une interface micro-USB. Bien sûr, il y a aussi de nombreuses E/S GPIO (*general purpose input/output*), même sous forme d'embase à picots à l'empreinte Arduino, du moins actuellement sur les cartes Nucleo. L'équipement de base typique des cartes Discovery n'a pas été oublié : quatre LED libres, une manette de jeu (à la place du clavier) et le brave débogueur ST-Link/V2-1. Dès qu'on alimente la carte par le port USB, rien que par le programme de démonstration, on ressent la puissance incroyable de ce minuscule microcontrôleur, en fait, grâce à la mise en œuvre de la carte mémoire sur un lecteur vidéo et audio avec écran. Et tout cela est obtenu pour une bonne vingtaine d'euros, autant que ce qu'il faut déboursier pour une Arduino Uno.

Programmer et déboguer la carte se fait de préférence avec Keil (figure 4), IAR

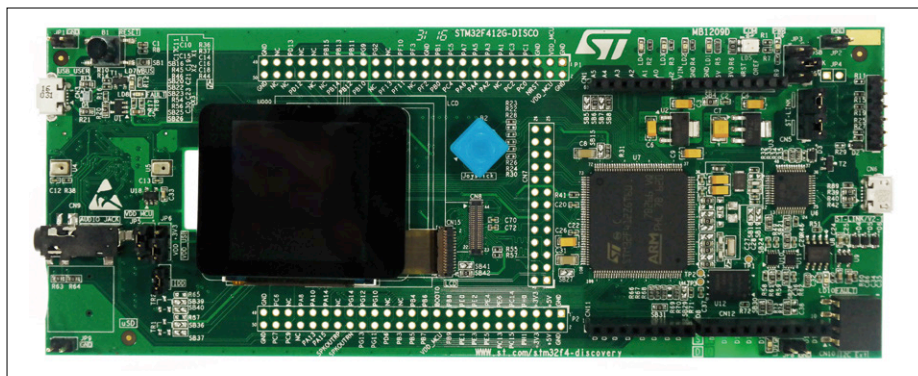


Figure 3. Tout en longueur, les kits STM32F412G-DISCO ne se mettent pas en boîtier avec Arduino et les autres connecteurs quand ils supportent un *shield*.

ou openSTM32. Le déjà célèbre configurateur STM32CubeMX ainsi que les bibliothèques et exemples de programme STM32Cube y jouent un rôle central.

### Attiny rattrape son retard

Depuis quelques années, on considérait les microcontrôleurs ATtiny comme nettement plus faibles et moins bien équipés que leurs grands frères Mega. Mais au salon *electronica 2016*, la famille ATtiny s'est exposée avec les types 8xx à 20 MHz, de 14 à 24 broches ainsi que 4 Ko ou 8 Ko de mémoire flash étendue et par-dessus le marché, la mini-carte Xplained.

À l'heure actuelle, ni les Mega ni les ARM ne peuvent encore considérer avec dédain le tout petit Tiny alors qu'il dispose largement des mêmes structures internes et éléments périphériques que les grands successeurs des AVR et même ARM-SAM. Une manifestation claire en est le système *Event* ainsi que les *Core Independent Peripherals* (CIP) et peut-être le *Peripheral Touch Controller* (PTC) qui travaillent de manière indépendante pour ne pas peser sur l'unité centrale. Depuis peu, on trouve aussi une carte **ATTiny817 Xplained Pro** (pour environ 36 €) avec le même contrôleur (figure 5) [5]. C'est bien le signe d'un agrandissement de la famille des Tiny, aussi parce que la carte comme le microcontrôleur sont pris en charge par l'environnement START en ligne [6] dont on peut déjà intégrer les grosses bibliothèques et les périphériques dans le navigateur, avant de passer à l'EDI Studio d'Atmel.

Comme presque toutes les Xplained Pro, la carte est équipée de deux boutons tactiles, d'embases à picots, d'un *Embedded Debugger* (EDBG) et même d'un circuit de mesure ajustable pour la consomma-

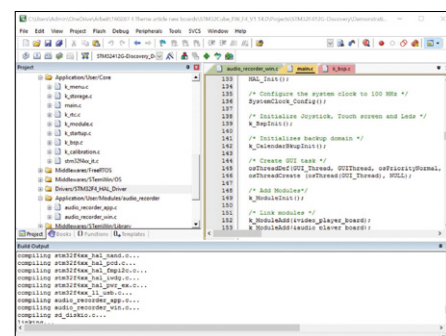


Figure 4. Si vous utilisez la version gratuite de l'EDI  $\mu$ Vision de Keil avec les bibliothèques et les programmes de démonstration du STM32-Cube-F4, faites attention à ne pas dépasser la limite permise pour le code.

tion de courant. Mais son gros avantage par rapport à la carte Mini, c'est de pouvoir utiliser les extensions de Xplained Pro d'Atmel, depuis le capteur de température jusqu'aux afficheurs OLED, pratiquement tout y est.

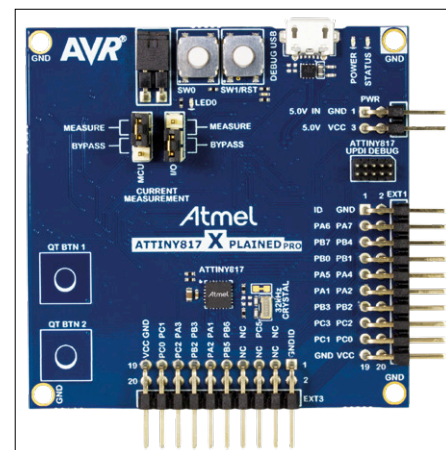


Figure 5. La carte Attiny-817-Xplained-Pro est à peu près aussi grande qu'une Xplained-Mini, mais elle offre plus de possibilités (source : Microchip).



Figure 6. Une copie d'écran de l'environnement de START sur internet. On peut encore configurer plus en détail chacun des éléments sélectionnés ci-dessus, avant d'importer le projet en cours dans le Studio 7 d'Atmel.

## Contacteur le monde analogique

Voici un tout nouveau membre de la famille PSOC à 32 bits, ARM-Cortex-M0+ : le coprocesseur analogique (48 MHz, 32 Ko de flash, avec DMA) spécialement orienté vers le monde analogique. Il constitue ainsi l'interface idéale (pour de grands projets, sans doute) entre le domaine des capteurs analogiques et le processeur hôte, comme on dit. Le coprocesseur dispose d'une périphérie analogique immense : deux CA/N (à 12 et 10 bits), deux IDAC à 7 bits (pour fournir le courant, surtout à la technologie tactile), quatre amplificateurs opérationnels, deux comparateurs, un *CapSense* tactile et un bloc analogique universel (UAB) qui peut servir de filtre, de CA/N à 14 bits ou de CN/A à 13 bits. De concert avec quelques mul-

tiplexeurs, on a de nombreuses possibilités, et toute la toute liberté, de relier un périphérique analogique à une borne de connexion. Malgré l'importance de l'analogique, ce sont quand même les périphériques numériques qui dominent avec par exemple les temporisateurs ou les interfaces. Ce qui mérite d'être souligné, c'est la présence d'un pilote de LCD intégré avec de nombreuses possibilités (mode de veille à 2,5  $\mu$ A) ainsi qu'une tension d'alimentation minimale de 1,71 V [7].

Pour faire connaître toutes les possibilités et la souplesse de la technologie PSOC, un kit a été développé, le **CY8CKIT-048 PSoC Analog Coprocessor Pioneer** [8] à la **figure 7**. Cette petite carte est dotée de cinq capteurs analogiques : un détecteur de mouvement PIR, des capteurs de température, d'humidité et de lumière ainsi qu'une bobine de réception imprimée pour faire un capteur de proximité inductif. La bobine a le vent en poupe parce qu'elle présente beaucoup d'avantages par rapport à la technique capacitive. Le capteur est disposé de manière à détecter la distance par rapport à une pièce de monnaie métallique livrée aussi, mais sur le même principe, on peut aussi bien construire un capteur tactile qui perçoit, au contact du doigt, la flexion, même minime, d'une plaque métallique.

Pour les différents capteurs présents sur la carte, il y a des possibilités de commutation par interrupteurs ou cavaliers. Beaucoup de liaisons sont accessibles sur des embases, mâles en dessous, femelles au-dessus, pour que ces cartes puissent servir de *shield* (Arduino) ou leur permettre de s'empiler dessus. Pour la partie numérique, il y a encore sur la carte une LED RVB et deux boutons-poussoirs utilisables. Comme passerelle entre la prise mini-USB et le contrôleur hôte, on se sert du programmeur et débogueur KitProg2 qui permet de programmer et déboguer la carte directement par le Creator PSoC (**figure 8**).

## Encore le grand style ?

L'an dernier, Renesas a présenté une petite carte en promotion basée sur la

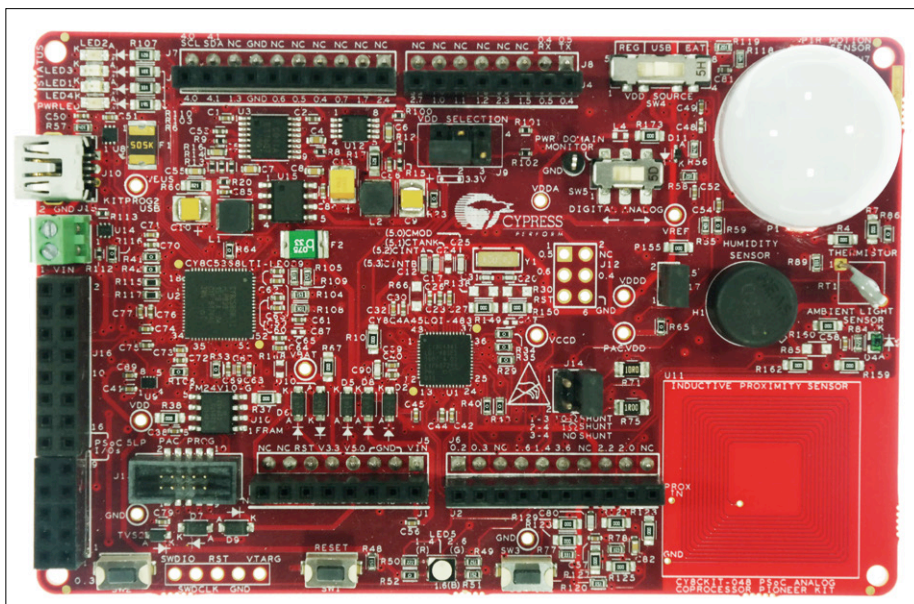


Figure 7. Le contenu du CY8CKIT-048 : la carte, le câble, une embase pour extension personnelle, mais aussi une pièce de monnaie métallique pour tester le capteur de proximité à induction.

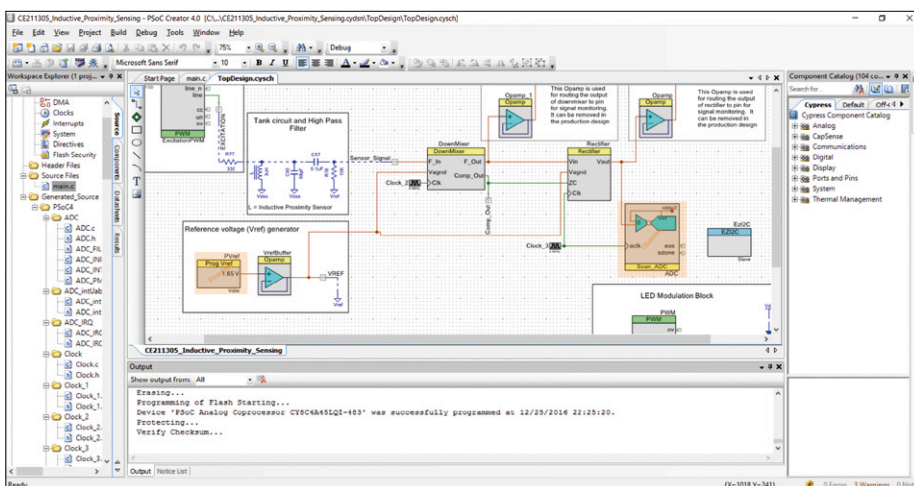


Figure 8. Voici le circuit des périphériques internes qui sont utilisés pour l'évaluation du capteur inductif dans le PSoC Creator. Tout est fait pour épargner au développeur une grande part de l'écriture du code.



famille en 16 bits RL78. Maintenant, il préconise à plus long terme la famille de microcontrôleurs à 32 bits ARM-Cortex-M-Synergy. En règle générale, pour un kit de découverte de Renesas, il faut puiser sérieusement dans ses finances, mais avec le **Starter Kit SK-S7G2** avec microcontrôleur Synergy-S7 (240 MHz, 4 Mo de code flash, avec DMA) il en va tout autrement (**figure 9**) [9].

Le fabricant a utilisé ce kit comme plateforme d'initiation pour de nombreux projets, il le vend au prix relativement modeste de 80 € environ et le distribue gratuitement sur les salons. Les points forts du S7 (Cortex M4) résident dans l'interface homme/machine (HMI) et le domaine de la connectivité. Il montre ses compétences avec le contrôleur graphique LCD, la communication Ethernet (avec deux contrôleurs MAC intégrés) ou encore le codec JPEG. Il devrait trouver sa place dans l'industrie 4.0, l'IdO et la domotique. Cela se répercute évidemment sur la construction des cartes. En plus des innombrables embases à picots avec les lignes du microcontrôleur et les embases femelles au format Arduino, il y a un écran tactile TFT à bord, plusieurs éléments tactiles, des interfaces USB, audio et Ethernet. Il y a encore des possibilités d'extension, comme le raccordement d'un module BLE (*Bluetooth Low Energy*). N'oublions pas les trois LED libres et les deux boutons, ainsi que forcément le débogueur J-Link intégré à la carte. Avec ce kit, vous pouvez franchement oublier le coûteux émulateur E1. Incidemment, sachez qu'il existe pour ce kit et la totalité de la famille Synergy un nouveau livre très recommandé, disponible au téléchargement [10].

L'EDI standard pour la totalité de la plateforme Synergy depuis le RL78 est bien sûr l'habituel E<sup>2</sup>studio de Renesas, mais il s'y ajoute la Gallery (**figure 10**), une plateforme rien que pour Synergy [11]. On y trouve toujours les dernières bibliothèques, les outils accessoires, par exemple l'affichage graphique ou la configuration de sécurité, et bien entendu beaucoup de documentation et d'aide.

### Duo de cartes sans fil

Récemment, NXP a sorti le KW41Z (48 MHz, 512 Ko flash, DMA) [12], un microcontrôleur à 32 bits ARM-Cortex-M0+, totalement dédié à la veine du sans-fil. Grosso modo, il combine les compétences du KW21Z et du KW31Z dans la bande des 2,4 GHz, ce qui

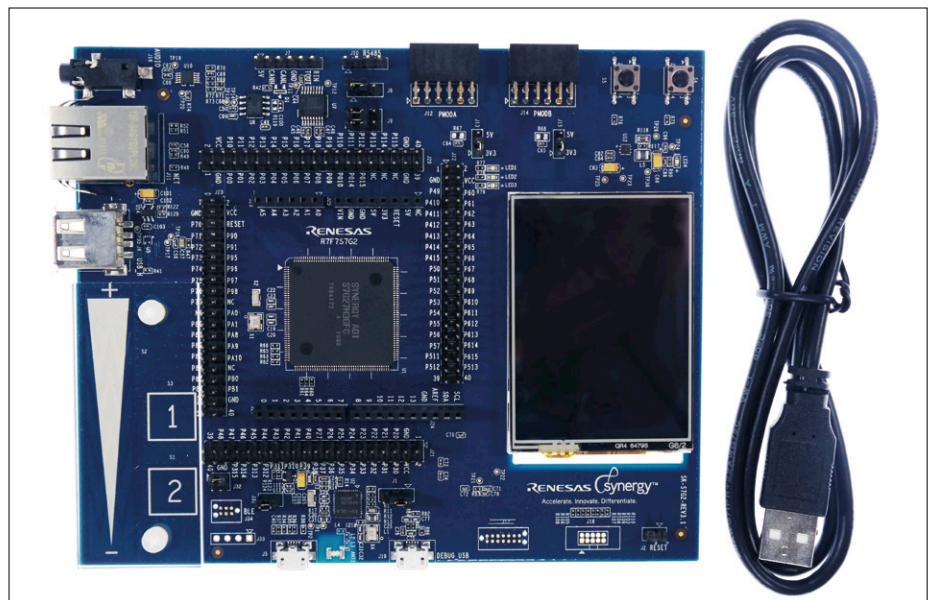


Figure 9. Le SK-S7G2 se compose d'un circuit imprimé et d'un câble USB, il est très bon marché, mais il n'a besoin de rien d'autre pour déboguer.

lui permet de communiquer aussi bien en mode BLE (*Bluetooth Low Power 4.2*) que selon le standard IEEE 802.15.4 (auquel appartiennent encore d'autres protocoles, comme *Thread*). Raison pour laquelle on appelle aussi le KW41Z un microcontrôleur *Dual Mode*. Comme toujours chez NXP (anciennement Freescale), on lui a concocté un kit *Freedom* immatriculé **FRDM-KW41Z** (**figure 11**), qui se présente sous la forme de deux cartes identiques, de manière à pouvoir essayer tout de suite les communications (environ 150 €).

Ces cartes possèdent, outre le débogueur OpenSDA du Studio Kinetis, une antenne tracée sur la carte et aussi une prise pour

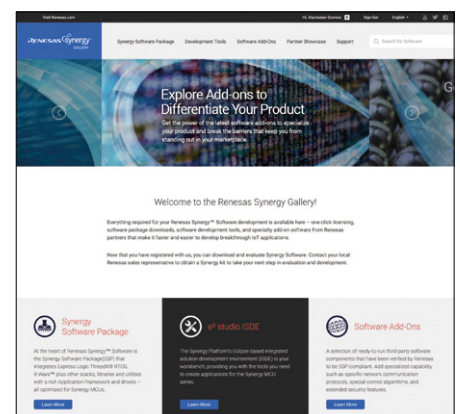


Figure 10. Sous la copie d'écran de Synergy Gallery, on voit les plus importantes pièces qui le composent.



Figure 11. Le duo de cartes et les deux câbles USB font partie du kit.



Figure 12. Quand tout est installé et aménagé, vous pouvez faire tourner le programme d'exemple avec la vidéo *Getting Started*.

antenne extérieure. On y voit aussi un coupleur de pile bouton, une LED RVB et une autre libre, deux boutons libres, un capteur de température analogique et le capteur FXOS8700CQ-I<sup>2</sup>C qui donne en 3D l'accélération et la mesure du champ magnétique.

Comme du temps de Freescale, NXP n'est pas avare de documentation. La description interactive *Getting Started* sur la page du produit [13] est aussi convain-

cante (cf. **figure 12**). Vous y trouverez des explications pas à pas de l'installation de l'EDI et du logiciel pour KW41Z, sur l'essai du programme d'exemple jusqu'à la construction de projets personnels à l'aide de nombreuses vidéos.

## L'expert de l'analogique en numérique

À côté des fabricants de contrôleurs très connus que sont Microchip, ST, NXP & Co, on oublie trop souvent celui qui s'est d'abord fait une renommée dans le monde de l'analogique, Analog Devices qui produit aussi des microcontrôleurs et de très beaux kits accompagnés d'exemples d'applications. Parmi les plus récents et plus connus microcontrôleurs de la série ARM-Cortex-M3, il y a l'ADuCM3029 (26 MHz, 256 Ko flash). Comme tout ce qui se termine par M3, il est solidement armé dans tous les domaines [14]. Il resplendit spécialement par ses quatre modes d'économie d'énergie, dont l'un peut faire chuter la consommation à moins de 60 nA.

Pour seulement 200 €, vous aurez un kit  **EVAL-ADuCM3029 EZ (figure 13)**, constitué essentiellement d'une carte à microcontrôleur avec deux boutons et trois LED libres, un vibreur, des capteurs de température (ADXL363) et d'accélération (ADT7420), un connecteur mini-USB (série), deux coupleurs de piles, des embases à picots au format Arduino et un débogueur externe J-Link-LITE. On peut le programmer avec les habi-

tuels EDI de Keil et IAR, riches en code et documentation, ou bien leur propre *CrossCore-Embedded-Studio*, mais qui est payant [15]. Un aspect à ne pas sous-estimer quand on veut réaliser son projet avec cette famille de microcontrôleurs, c'est l'expérience du fondateur de silicium dans le domaine analogique. Analog Devices propose des capteurs et d'autres composants pour microcontrôleurs pour lesquels il fournit les bibliothèques correspondantes et toute forme d'aide. Pour les premières expériences, il est recommandé d'aller voir les nombreux *shields* (à capteurs) disponibles. Ce fabricant s'est bien introduit dans la sphère numérique avec des cartes comme l'**ADZS-BF707-BLIP2** [16] et toute la technologie qui s'y rapporte. Cette carte vaut à peu près 200 €, dispose de nombreuses interfaces et d'un microcontrôleur ADZS-BF707 à 32 bits cadencé à 400 MHz. Avec caméra et capteurs embarqués, on est capable de reconnaître fidèlement des objets, des personnes ou même des visages. Pour des développements avec cette carte, on doit disposer de l'émulateur ICE-1000 correspondant au kit et aussi du *CrossCore Embedded Studio* (une licence pour 120 jours est jointe à la livraison). Il existe aussi un bel outil, juste pour s'en faire une idée.

## Comment glaner de l'énergie

Le fabricant de semi-conducteurs Linear Technology a récemment développé avec Würth Elektronik, l'expert en composants passifs et connecteurs, le **Gleanergy Demonstration Kit** qui coûte environ 400 € (**figure 14**). *Gleanergy* est un mot-valise composé de l'anglais *to glean*, glaner et *energy*, de l'énergie. Le kit contient principalement deux circuits imprimés et un émetteur-récepteur USB DC2274A-A [17].

Le premier circuit imprimé DC2344A [18] est responsable de la partie analogique du glanage d'énergie. On y trouve une pile (*primary battery*), un accumulateur (*secondary battery*), un condensateur Supercap, un élément Peltier, deux cellules solaires et le circuit tampon correspondant avec de nombreux réglages (par cavaliers), embranchements de mesure et sorties. Au milieu, quatre puces de régulation de Linear Technology. Deux d'entre elles sont des modèles universels pour cellules solaires, une autre pour un élément piézoélectrique ou un généra-

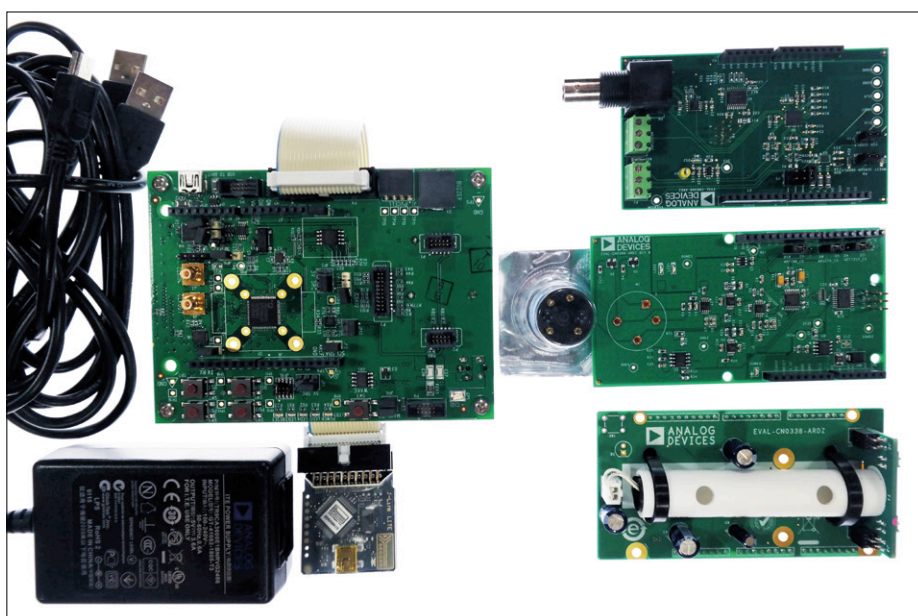


Figure 13. Les principaux composants du kit sont le débogueur, la carte, le câble USB ainsi que l'alimentation secteur. À droite, on peut voir une petite sélection de *shield* A/N, ici avec les capteurs de pH, de température, d'humidité ainsi que de gaz.



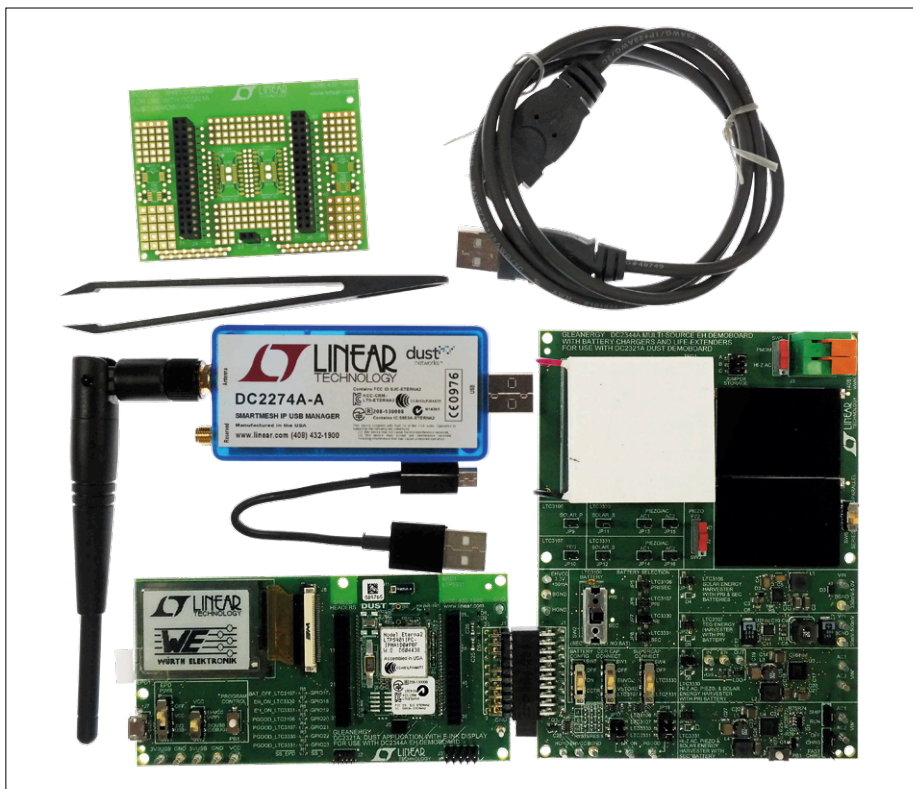


Figure 14. On n'en a jamais que pour son argent : les deux circuits imprimés, le câble de qualité supérieure, le récepteur USB avec antenne, une notice pour des extensions et même une petite pince.

teur. La grande différence provient de la cellule à charger. La LTC3330 sert à la pile, la LTC3331 fonctionne avec l'accumulateur. Il y a encore la LTC3107 pour l'élément Peltier et la pile, ainsi que la LTC3106 uniquement pour la cellule solaire, en fait toutes deux pour entreposer l'énergie (pile/accu). Entre les cavaliers et interrupteurs de l'avant du circuit imprimé et les résistances en pont sur l'arrière, il y a une ribambelle de commutations possibles. On peut encore enclencher, déclencher ou mettre en parallèle chaque régulateur séparément. Mais la documentation relative à la carte est très pénible à ce sujet.

Le deuxième circuit imprimé peut se brancher directement sur le côté du DC2344A, c'est le DC2321A [19] qui regroupe le module Dust-Mote LTP5901 et un écran à encre électronique. Un *mote* n'est rien d'autre qu'un nœud de réseau sans fil, *Dust* est son nom commercial. Le LTP5901 est un représentant de la famille *Dust SmartMesh IP Wireless* de LT, il travaille à la mode du Zigbee bien connu dans la bande de 2,4 GHz selon la norme IEEE 802.15.4e et possède un noyau ARM-Cortex-M3. Avec la technologie Dust, on peut échafauder des réseaux

sans fil à structure compliquée (pensez IdO). Dans Gleanergy, le module LTP5901 a pour mission la surveillance du glanage d'énergie. Inutile de préciser qu'il est lui-même animé de la philosophie de la moindre consommation. Il envoie les données en série à l'interface micro-USB, destination l'écran encre. Là-dessus, il y a encore un outil de démonstration et d'analyse de toutes les configurations importantes des régulateurs pour, au bout du compte, en arriver à la phase du prototype en vue de sa propre récolte d'énergie (**figure 15**). Avec le *shield* vide livré DC2510A, on peut encore étendre le circuit. La formule complète peut bien sûr fonctionner sans fil avec la technologie SmartMesh et la clé USB DC2274A-A. Et puis il existe un **software development kit** (SDK) correspondant. En conclusion, si vous cherchez des solutions sans fil ou du glanage d'énergie, l'achat en vaut la peine pour réaliser vos projets et raccourcir la phase de prototypage. ■

(160287 – version française : Robert Grignard)

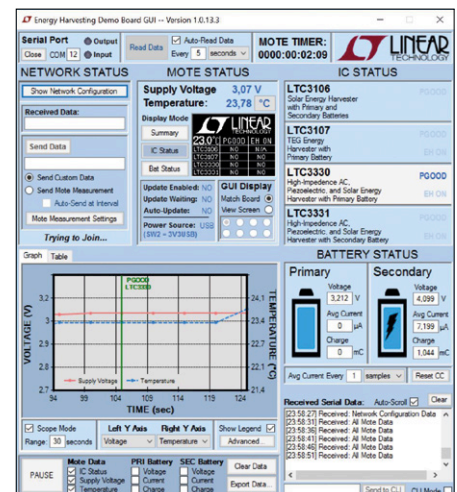


Figure 15. L'outil QuikEval est entièrement dédié au développement d'applications de glanage d'énergie.

## Liens

- [1] [www.elektormagazine.fr/150687](http://www.elektormagazine.fr/150687)
- [2] [www.infineon.com/cms/en/product/promopages/sensors-2go/](http://www.infineon.com/cms/en/product/promopages/sensors-2go/)
- [3] [goo.gl/S17DDt](http://goo.gl/S17DDt)
- [4] [www.st.com/en/evaluation-tools/32f412gdiscovery.html](http://www.st.com/en/evaluation-tools/32f412gdiscovery.html)
- [5] [www.atmel.com/tools/attiny817-xpro.aspx](http://www.atmel.com/tools/attiny817-xpro.aspx)
- [6] [start.atmel.com](http://start.atmel.com)
- [7] [www.cypress.com/products/psoc-analog-coprocessor](http://www.cypress.com/products/psoc-analog-coprocessor)
- [8] [goo.gl/E0Q92g](http://goo.gl/E0Q92g)
- [9] [goo.gl/7jsViQ](http://goo.gl/7jsViQ)
- [10] [www.renesas.com/en-us/products/synergy/book.html](http://www.renesas.com/en-us/products/synergy/book.html)
- [11] [synergycastle.renesas.com/](http://synergycastle.renesas.com/)
- [12] [goo.gl/d9OSpQ](http://goo.gl/d9OSpQ)
- [13] [goo.gl/RhxxvQH](http://goo.gl/RhxxvQH)
- [14] [goo.gl/RhZ0Dd](http://goo.gl/RhZ0Dd)
- [15] [goo.gl/au8na5](http://goo.gl/au8na5)
- [16] [goo.gl/lw4kaA](http://goo.gl/lw4kaA)
- [17] [www.linear.com/solutions/5744](http://www.linear.com/solutions/5744)
- [18] [www.linear.com/solutions/7717](http://www.linear.com/solutions/7717)
- [19] [www.linear.com/solutions/7662](http://www.linear.com/solutions/7662)

# passerelle IoT et nœuds sans fil

## 1<sup>ère</sup> partie : le matériel



Il manque toujours une fonction spécifique à un système de domotique du commerce. C'est pourquoi l'auteur a conçu son propre système où les différents nœuds (*nodes*) communiquent sans fil avec la passerelle (*gateway*) centrale. Cette dernière utilise le protocole MQTT pour envoyer des données de mesure à un serveur OpenHAB, qui les traite et les supervise. Ensuite, plusieurs autres utilisateurs ont repris son projet et l'ont étoffé. En deux articles, l'auteur décrit les éléments les plus importants de son système – des informations plus détaillées et le logiciel associé sont disponibles gratuitement sur l'internet.

**Hennie Spaninks** (Pays-Bas)

Les systèmes domotiques font rêver. C'est magique de voir les lampes s'allumer et s'éteindre toutes seules, de constater que le chauffage « sait » quand vous êtes chez vous ou de recevoir un message qui signale un quidam à la porte. On trouve dans le commerce de nombreux systèmes, mais ils « souffrent » en général des défauts suivants :

- Ils sont souvent unidirectionnels ; on peut envoyer une instruction, sans avoir la certitude qu'elle est arrivée. Si on lit sous une lampe, pas de problème, mais lorsqu'on est en vacances à l'étranger, savoir que tout va bien rassure.
- Souvent, ils ne sont pilotables qu'avec une appli spécifique : chaque fabricant a la sienne et les protocoles utilisés sont loin d'être compatibles. D'où un ordinophone encombré d'une pléthore d'applis.
- L'attrait de la domotique est qu'elle permet d'automatiser les choses : envoyer des instructions en réponse à la

valeur de mesure d'un capteur. Comme la plupart des systèmes disponibles dans le commerce ne sont pas compatibles, il faut l'intervention d'un tiers (*IfThisThenThat* par ex.) pour qu'il se passe quelque chose.

- Les systèmes sans fil font souvent appel au Wi-Fi. Dans les maisons en béton armé, la portée est limitée.
- La sécurité est loin d'être optimale : risque de voir le voisin allumer vos lampes.

Voilà suffisamment de motifs pour retrousser ses manches et construire son propre système domotique !

### Panorama

Pour les capteurs et actionneurs (« nœuds terminaux », *end nodes*), nous optons de préférence pour des versions abordables et frugales en énergie. La commande centrale doit bien entendu reconnaître les protocoles réseau standard tels que TCP/IP.

C'est pourquoi nous avons opté pour un système à deux niveaux : les nœuds terminaux, une passerelle et une unité



de commande centrale. On en retrouve la structure dans le synoptique de la **figure 1**.

- Pour la communication avec les nœuds terminaux, nous utilisons une connexion sans fil en duplex à fréquence relativement basse. En Europe, 433 MHz et 868 MHz sont les fréquences éligibles. Choix du 868 MHz, la bande de 433 MHz est assez encombrée. La société HopeRF fournit des émetteurs-récepteurs (*transceiver*) pilotables via un bus SPI qui travaillent sur cette fréquence ; nous avons choisi la version à la puissance d'émission la plus élevée, le RFM69HW. Ce module embarque un cryptage matériel, la sécurité des données est donc garantie.
- Le RFM69HW n'a pas de pile TCP/IP, d'où l'échange, via la liaison radio, d'un bloc de données fixe de 66 octets. Il nous faut une passerelle qui traduise les données du réseau sans fil vers TCP/IP et vice versa.
- Pour la distribution des données du « producteur » (nœuds de capteurs) au « consommateur » (ordiphone par ex.), nous optons pour un protocole standard différent qui tourne sous TCP/IP : MQTT, un protocole de messagerie conçu pour envoyer de façon simple des messages courts à de multiples clients. Comme MQTT est un standard établi, les flux de données de notre système de domotique sont utilisables avec d'autres systèmes MQTT. MQTT requiert un serveur qui sert de courtier (*broker*, centrale de messages). Un client (ordiphone par ex.) peut s'abonner à un flux de données pour recevoir les messages véhiculés par ce flux. Pour réaliser le courtier MQTT, nous utilisons Mosquitto [1], une implémentation *open source* d'un courtier MQTT, disponible pour différentes plateformes (pour Raspberry Pi entre autres).
- Notre centrale domotique devrait pouvoir être utilisée avec les équipements de tiers. Elle doit donc en plus être capable de communiquer par le biais d'autres protocoles. Nous avons opté pour un système basé sur OpenHAB [2]. Ce logiciel *open source* peut convertir des messages de protocoles différents vers un flux de messages universel. On peut ainsi également piloter des lampes Hue.
- OpenHAB dispose d'une appli standard (Android et iOS) qui, via l'internet, communique de manière sécurisée avec votre propre système OpenHAB. Aucun souci donc de pare-feu ou de cryptage au niveau du lien reliant notre téléphone à notre système domotique. OpenHAB est accessible sur le réseau local via une interface web.
- OpenHAB comporte une base de données sous-jacente ; ceci permet de convertir des valeurs de mesure en graphiques simples. Il est facile, sous OpenHAB, de créer des lignes (d'instruction) et des scripts, pour automatiser des processus. OpenHAB dispose aussi d'interfaces externes pour les messages d'alarme par courriel ou via l'appli OpenHAB.
- Nous utilisons des composants disponibles et bon marché. Arduino nous servira de contrôleur des nœuds sans fil et un Raspberry Pi de plateforme pour Mosquitto et OpenHAB.

### Protocole pour les nœuds terminaux

Plusieurs fonctions sont définies dans les nœuds terminaux : lire des capteurs de température et d'humidité, des interrupteurs, des détecteurs PIR, voire piloter des relais et des écrans LCD.

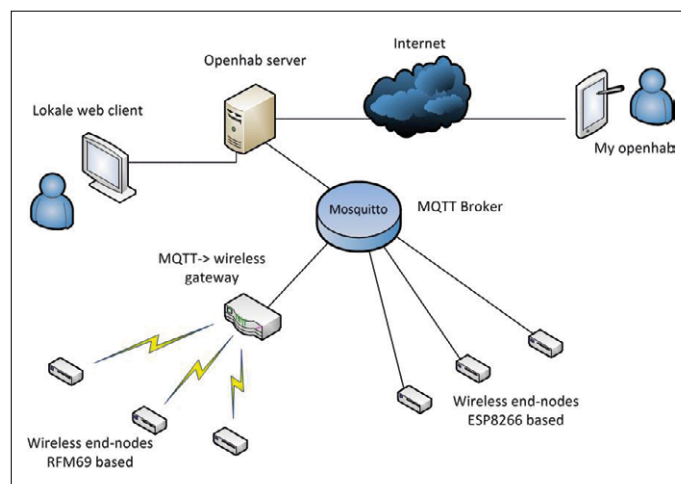


Figure 1. Synoptique du système de domotique.

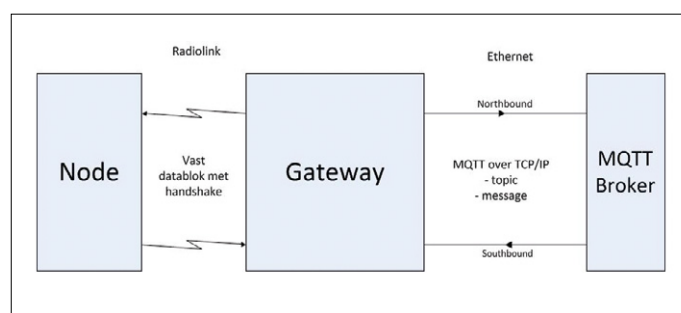


Figure 2. Schématisation des flux de données véhiculés par la passerelle.

**Tableau 1. Identificateurs d'appareil (Device identifiers)**

Plage des DevID	Fonction
0-15	Appareils système
16-31	Sortie binaire (relais, lampe)
32-39	Appareils à sortie de type entier (gradateur, MLI)
40-47	Entrée binaire (interrupteur, module PIR)
48-63	Entrée décimale (température, humidité)
64-71	Entrée entière (clavier, interrupteur)
72	Transfert de chaîne (affichage LCD, lecteur RFID)
73-90	Pour usage ultérieur
90-99	Messages d'erreur

Les nœuds terminaux sont en mesure de fournir des valeurs de mesure à intervalle régulier. Il est possible aussi, sur demande, d'envoyer ces valeurs (après une instruction de lecture).

La **figure 2** montre le flux de données par la passerelle. Un nœud terminal peut être programmé pour différentes fonctions. Pour veiller à ce que la passerelle gère de façon correcte le trafic de chaque nœud, on a effectué une classification fixe des fonctions (*device* = appareil). En fonction du code d'appareil (DevID), la passerelle sait comment traiter les données entrantes. Le **tableau 1** donne un aperçu des DevID reconnus par la passerelle. Les *devices* 0 à 15 sont des fonctions système. Le **tableau 2** en donne la description. Pour finir, le **tableau 3** indique les messages d'erreur que produit la passerelle.

**Tableau 2. Fonctions système**

DevID	Nom	RW	Fonction
00	Uptime	R	Minutes depuis démarrage du nœud
01	TxInterval	RW	Intervalle d'envoi en secondes (0 = pas de transmission périodique)
02	RSSI	R	Intensité du signal radio
03	Version	R	Version du logiciel du nœud terminal
04	Voltage	R	Tension de la pile
05	ACK	RW	Drapeau de confirmation des instructions émises
06	Toggle	RW	Drapeau de fonction bascule du bouton-poussoir du nœud terminal
07	TImmer	RW	Drapeau de fonction <i>timer</i> du bouton-poussoir du nœud terminal
08	Btnpress	RW	Drapeau d'envoi du message « bouton pressé »
09	TXreply	R	Nombre de répétitions requises sur la liaison radio

**Tableau 3. Messages d'erreur**

Erreur ID	Nom	Description
90	Link error	La liaison radio est interrompue.
91	Syntax error	Le message MQTT comporte une erreur de syntaxe.
92	Invalid device	Le DevID adressé n'existe pas dans le nœud terminal.
99	Wakeup	Message envoyé lors du démarrage du nœud.

**Tableau 4. Exemples MQTT**

Sujet	Message	Description
home/rfm_gw/sb/node02/dev16	ON	Allume la LED du nœud 02.
home/rfm_gw/sb/node02/dev16	READ	Demande l'état de la LED du nœud 02.
home/rfm_gw/sb/node03/dev01	300	Change l'intervalle TX du nœud de 3 à 5 min.
home/rfm_gw/sb/node03/dev01	0	Désactive les transmissions périodiques du nœud 3.
home/rfm_gw/sb/node18/dev48	READ	Lit la température du nœud 18.
home/rfm_gw/sb/node05/dev02	READ	Lit l'intensité du signal radio du nœud 5.
home/rfm_gw/sb/node05/dev03	READ	Lit la version du logiciel du nœud 5.

## Messages MQTT

Le protocole MQTT travaille avec des abonnements, appelés *topics* (sujets) (cf. [3]). Si un client est abonné à un sujet, le courtier veillera à ce que tous les messages présents dans ce sujet lui soient livrés.

Le sujet MQTT utilisé a le format :

`home/rfm_gw/direction/nodeID/deviceID`

- **direction** indique ici le sens du flux de données. **nb** (*Northbound*) signifie du nœud vers serveur OpenHAB (valeurs de capteur), **sb** (*Southbound*) de OpenHAB vers le nœud (instructions).

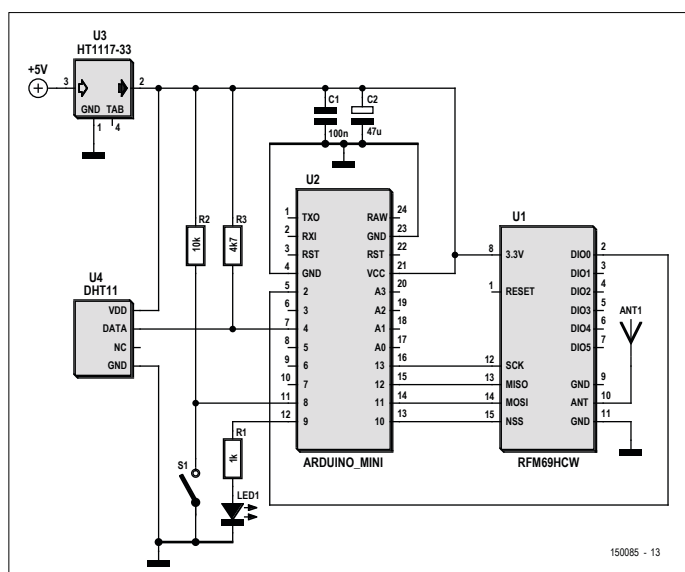


Figure 3. Schéma d'un nœud terminal.

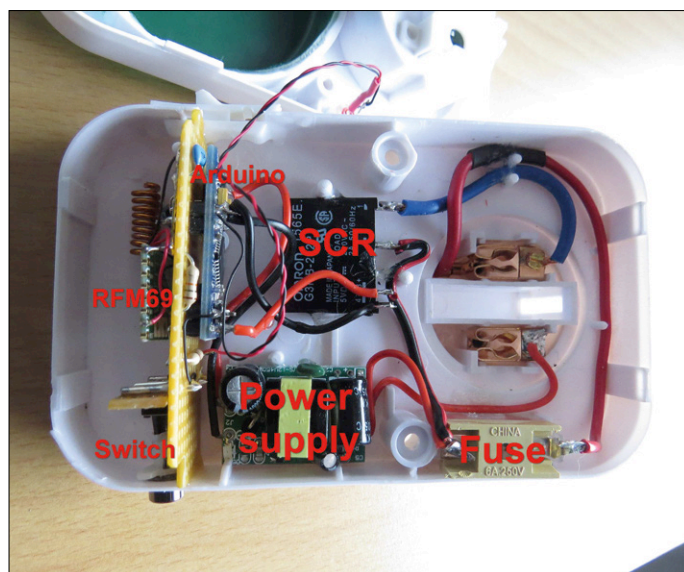


Figure 4. Nœud DIG dans un boîtier enfichable.



- **nodeID** est l'**ID**entifiant du nœud. Chaque nœud a un identifiant unique (adresse) déterminé au cours de la compilation du code. La passerelle a toujours le **nodeID** 01.
- **deviceID** est l'identifiant de la fonction, comme décrit plus haut.

Le message MQTT (*payload* = charge utile) dépend de l'appareil et de la direction :

- **Southbound** : le message contient des instructions pour le nœud (ON, OFF, READ) ou des valeurs à remettre au nœud (le nombre de secondes de l'intervalle de transmission par ex.).
- **Northbound** : le message contient la valeur du paramètre associé au DevID.

La passerelle s'abonne à tous les messages *Southbound* pour tous les nœuds du réseau avec le sujet suivant avec un caractère générique (*wildcard*) :

`home/rfm_gw/sb/#`

OpenHAB tient bien sûr à recevoir tous les messages venant des nœuds et s'abonne donc au sujet :

`home/rfm_gw/nb/#`

Le **tableau 4** donne quelques exemples de messages MQTT.

## Nœud terminal

Le cœur du nœud terminal est un Arduino. La conception tient compte des conditions connexes suivantes :

- Le module RFM69HW travaille sous 3,3 V. Les entrées ne supportent pas plus de 3,9 V. Pour ne pas compliquer les choses, la carte Arduino Pro Mini est utilisée sous 3,3 V et peut être reliée au RFM69HW sans adaptation de niveau. Alternative : la carte Arduino Buono R3 qui peut être commutée à 3,3 V.
- À l'émission, le module RFM consomme brièvement un courant intense (130 mA) ; il faut donc une alimentation de capacité suffisante.
- La communication entre l'Arduino et le RFM69HW se fait via le bus SPI ; on utilise pour cela les broches standard de l'Arduino.
- Le RFM69HW ne doit pas être utilisé sans antenne ; dans la plupart des cas, un fil de 8,6 cm fera l'affaire.

La **figure 3** donne le schéma d'un nœud terminal qui comporte un capteur de température/humidité DHT11 relié à la broche 4 de l'Arduino. Un bouton-poussoir attaque la broche numérique 8 et une LED est reliée à la broche 9.

Le module RFM est couplé à l'Arduino via le bus SPI (SCK, MOSI, MISO et NSS). La tension d'alimentation de 3,3 V fait appel à un régulateur AMS1117 épaulé par les condensateurs de découplage requis pour encaisser les crêtes de courant. Le logiciel chargé dans l'Arduino détermine la fonction du nœud terminal. À l'heure actuelle, les nœuds terminaux suivants sont fonctionnels :

- DHT est un nœud terminal avec des capteurs de température et d'humidité. Le nœud comporte une sortie numé-

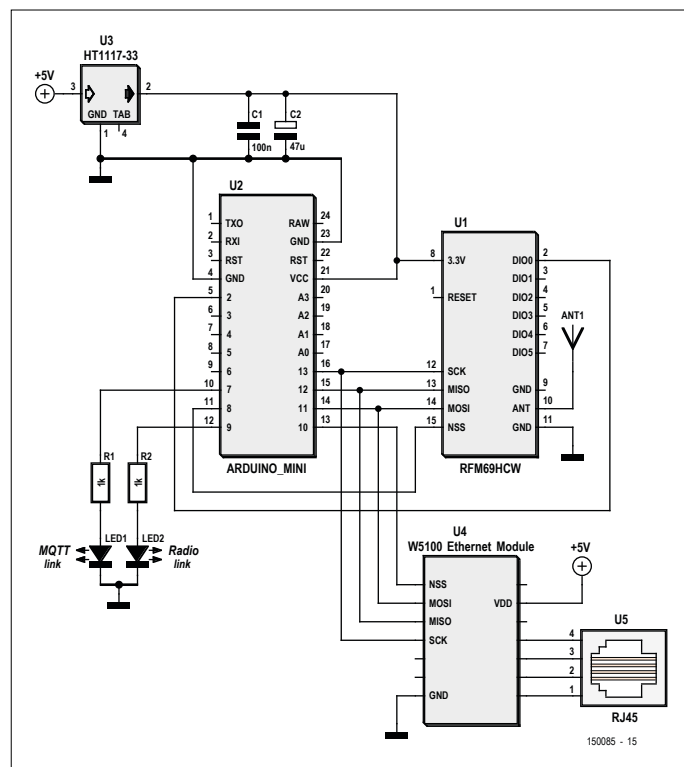


Figure 5. Schéma de la passerelle.

rique et un bouton-poussoir.

- DIG est une version simple du nœud DHT. Il se résume à un bouton-poussoir et une sortie numérique.
- RFID est un nœud doté d'un lecteur RFID. En cas de détection d'un système RFID, le nœud envoie l'ID(enti-fiant) RFID au serveur OpenHAB.
- LCD est un nœud avec un écran LCD. Il est possible depuis OpenHAB d'envoyer des lignes de texte à l'écran.
- RC est un nœud équipé d'un émetteur de 433 MHz. Il est possible, à partir de ce nœud, de commander les interrupteurs du système de domotique *KlikAanKlikUit* [4].

Le logiciel est modifiable aisément pour y ajouter des fonctions ou en modifier. La structure du nœud terminal dépend beaucoup de la fonction. On pourra par ex. utiliser un programmeur mécanique bon marché pour y caser un nœud DIG, solution qu'illustre la **figure 4**.

## La passerelle (gateway)

La passerelle est constituée d'un Arduino, d'un RFM69HW et d'un module Ethernet W5100. On en retrouve le schéma en **figure 5**. La passerelle aussi utilise un Arduino qui travaille sous 3,3 V.

On voit sur le schéma que les RFM69 et W5100 se partagent le bus SPI dans la passerelle. Les signaux SCK, MOSI et MISO sont connectés en parallèle. On fait appel, pour la sélection du bon appareil sur le bus, à différents signaux *Slave Select* (SS) : la broche 8 pour le RFM69 et la broche 10 pour le module Ethernet.

À nouveau, c'est un AMS1117 qui se charge de la tension d'alimentation de 3,3 V. Les deux LED indiquent l'état de la passerelle.

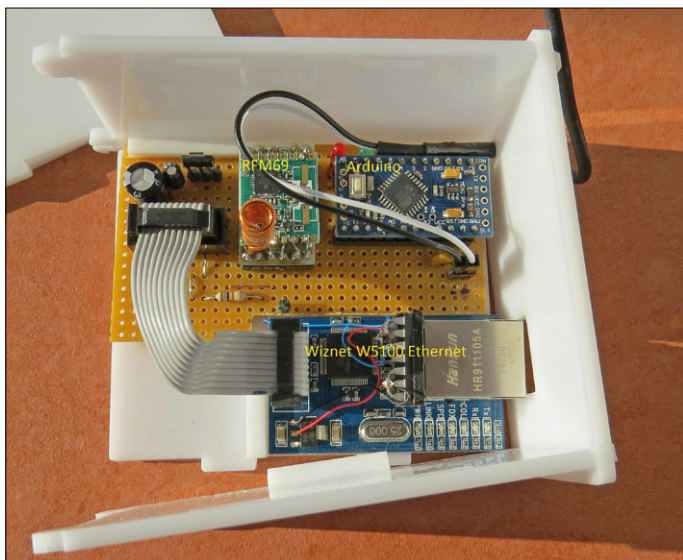


Figure 6. Voilà la passerelle telle que réalisée par l'auteur.

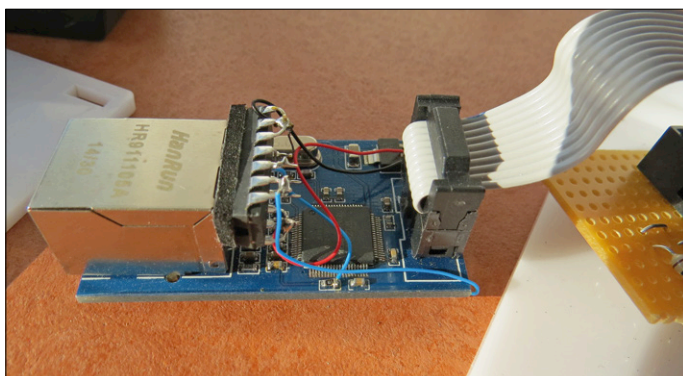


Figure 7. Le 4011 est fixé au boîtier du connecteur réseau et produit le signal SEN pour le W5100 à partir du signal CS.

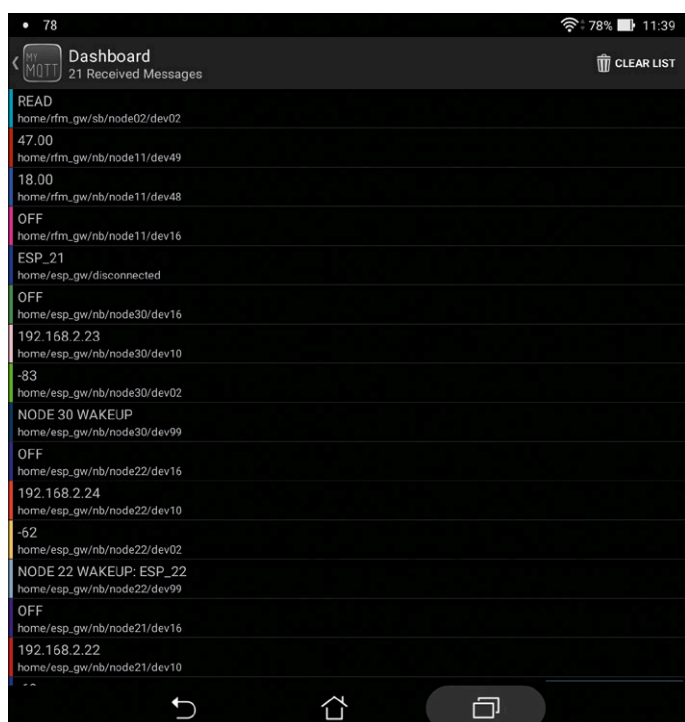


Figure 8. Cette capture d'écran montre un exemple du trafic MQTT.

La connexion Ethernet peut se faire à l'aide d'un module distinct. Si on a choisi une carte Arduino Buono, on pourra utiliser un *shield* Ethernet Wiznet. Une passerelle par système de domotique suffit ; c'est pourquoi l'auteur a réalisé sa passerelle sur une carte de prototypage (**fig. 6**).

Il y a deux aspects auxquels il faut veiller lors de la réalisation de la passerelle :

### Le bogue W5100 (W5100 bug)

Le matériel du W5100 ne convient pas directement à une utilisation de plusieurs appareils sur le bus SPI. Lorsque le W5100 n'utilise pas le bus, les signaux de bus doivent être « libérés » pour qu'un autre appareil puisse utiliser le bus. En standard, le W5100 ne le fait pas. On résout ce problème en alimentant le signal SEN par le biais du signal CS après passage par un inverseur (cf. [5] pour plus de détails), ledit inverseur a été intégré dans les versions ultérieures du *shield* W5100. Ce n'était pas le cas du module acheté par l'auteur sur Ebay, il l'a donc rajouté sous la forme d'un circuit intégré (4011) (cf. **fig. 7**). Cette figure montre également qu'il est facile de connecter le signal SEN au travers d'une résistance de forçage au niveau haut (*pull-up*) sur le circuit imprimé. Ne pas oublier de mettre à la masse les entrées inutilisées du 4011 !

### Court-circuit par le biais de l'embase ICSP

En cas d'utilisation d'un Arduino de format standard (tel que le Buono) en combinaison avec un *shield* W5100, l'Arduino se voit alimenté en 5 V par le *shield* Ethernet au travers de l'embase ICSP. La tension d'alimentation de 3,3 V de l'Arduino est ainsi rehaussée avec le risque d'endommager le RFM69HW. Pour éviter cela, supprimez la broche VCC de l'embase ICSP ou repliez-la vers l'extérieur, ce qui empêche ainsi tout risque de contact.

### À suivre...

Dans le prochain article, nous nous intéresserons au logiciel de la passerelle et des nœuds ainsi qu'à la configuration de Mosquitto et d'OpenHAB. Les impatients trouveront le logiciel en [6]. Il faut disposer de Mosquitto pour tester la passerelle et un client MQTT s'avérera utile lors des tests. Sur Android, l'appli *MyMQTT* fait parfaitement l'affaire. Sur Windows, on utilisera *Chrome Lens* ou *MQTT.FX*. La **figure 8** montre à quoi ressemblent les messages.

Ce projet a déjà été réalisé en plusieurs exemplaires. Le forum [7] sert de plateforme de discussion pour des améliorations, en cas de problèmes et autres expériences. ◀

(150085 - version française : Guy Raedersdorf)

### Liens

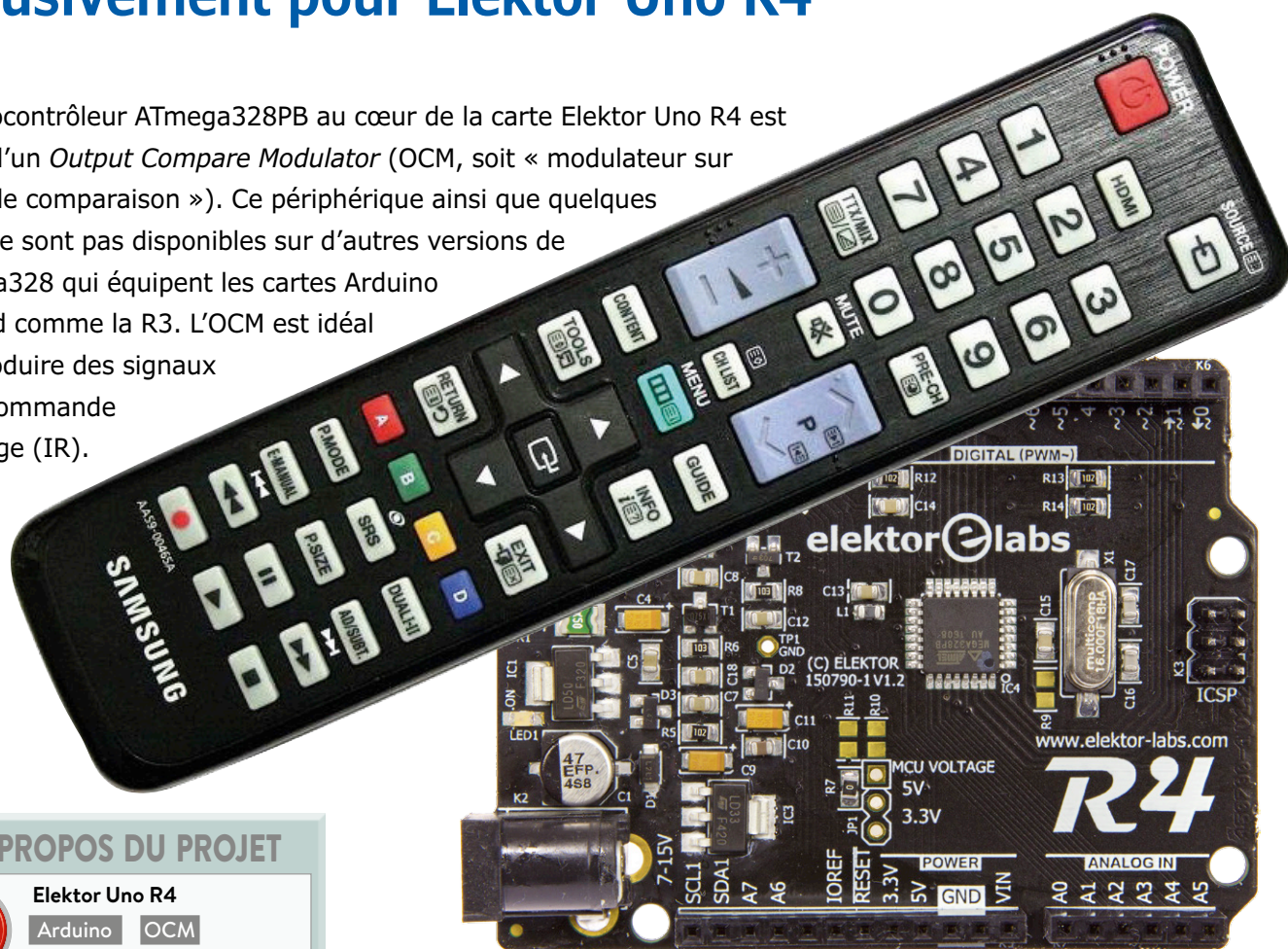
- [1] <https://mosquitto.org>
- [2] [www.openhab.org](http://www.openhab.org)
- [3] [www.elektormagazine.fr/news/mon-voyage-dans-le-nuage-iot-13-un-petit-capteur](http://www.elektormagazine.fr/news/mon-voyage-dans-le-nuage-iot-13-un-petit-capteur)
- [4] [www.klikaanklikuit.nl](http://www.klikaanklikuit.nl)
- [5] <http://john.crouchley.com/blog/archives/662>
- [6] <https://github.com/computourist/RFM69-MQTT-client>
- [7] <http://homeautomation.proboards.com/board/2/openhab-rfm69-based-arduino>



# signaux de télécommande au format NEC

## exclusivement pour Elektor Uno R4

Le microcontrôleur ATmega328PB au cœur de la carte Elektor Uno R4 est équipé d'un *Output Compare Modulator* (OCM, soit « modulateur sur sorties de comparaison »). Ce périphérique ainsi que quelques autres ne sont pas disponibles sur d'autres versions de l'ATmega328 qui équipent les cartes Arduino standard comme la R3. L'OCM est idéal pour produire des signaux de télécommande infrarouge (IR).



### À PROPOS DU PROJET

#### Elektor Uno R4

Arduino OCM

Télécommande RC-5

débutant

→ connaisseur

expert

env. 1 h

Elektor Uno R4,  
ordinateur avec  
EDI d'Arduino

env. 25 €

Clemens Valens (labo d'Elektor)

L'OCM combine les sorties des *Timer3* et *Timer4* (compteurs/minuteurs) – une autre exclusivité de l'Elektor Uno R4 – et sort le résultat sur PD2, broche 2 de l'Arduino et 32 du microcontrôleur. Combiner consiste ici en un ET ou un OU logique des deux signaux (voir **fig. 1**). La valeur du registre PORTD2 détermine l'opération effectuée : si PORTD2 est à 1, c'est un OU ; s'il est à 0, c'est un ET. On trouve fréquemment des signaux comme ceux de la figure 1 dans les télé-

commandes à infrarouges, pour moduler le courant qui alimente la LED IR. Mon domicile et le vôtre, je pense, sont parsemés de télécommandes. Il y a quelque temps j'avais besoin d'une télécommande compatible Philips RC-5, je les ai toutes testées, mais aucune d'entre elles n'utilisait le protocole RC-5, pas même mon lecteur de DVD Philips (RC-6). La plupart de mes télécommandes employaient une variante du protocole NEC. C'est logique, puisque la plupart de mes appareils télécommandés, et les vôtres aussi je suppose, sont fabriqués en Asie. Comme nous avons notre péri-

phérique OCM, utilisons-le pour produire des signaux compatibles avec le protocole NEC.

Un code de type NEC commence par un préambule, une salve de 9 ms suivie d'un silence (pause) de 4,5 ms. Ensuite, les bits, 32 en tout, sont transmis l'un après l'autre. Un « 0 » se traduit par une salve puis une pause, de 562,5 µs chacun (au total 1.125 µs), et un « 1 » par une salve de 562,5 µs puis une pause de 1.687,5 µs (au total 2.250 µs). La séquence doit se terminer par une salve de fin pour « englober » le dernier bit. Toutes les durées indiquées ici sont des multiples de 562,5 µs. Par la suite, nous appellerons « impulsion » la durée d'une salve, et « période » la durée d'une salve + pause.

Une salve n'est rien de plus qu'un signal à haute fréquence, une porteuse modulée en tout ou rien. *Timer3* peut délivrer la porteuse, et *Timer4* la moduler ; ou inversement, mais j'ai dû choisir. Tous les compteurs de l'ATmega328, y compris *Timer3* et *Timer4*, peuvent produire des interruptions lors d'un débordement ou du dépassement d'une certaine valeur. Ceci offre un sympathique mécanisme pour moduler la porteuse avec les données à transmettre : à chaque déclenchement de l'interruption, le prochain bit à transmettre est lu dans le tampon d'émission puis utilisé pour déterminer l'impulsion et la période nécessaires pour émettre un 0 ou un 1. Le croquis suivant montre comment faire.

```
/*
 * Emission infrarouge
 * Utilise le module OCM, ATmega328PB uniquement.
 */

#define CARRIER 38000
#define RC_CODE 0x6170807f

const int ir_led = 2;
const uint16_t pulse = 1125; // 2*562.5
uint32_t tx_buffer = 0;
uint32_t tx_index = 0;

#define TIMER3_ON TCCR3B |= (1<<CS30)
/* prédiviseur à 1 */
#define TIMER3_OFF TCCR3B &= ~(1<<CS30)
#define TIMER4_ON TCCR4B |= (1<<CS41)
/* prédiviseur à 8 */
#define TIMER4_OFF TCCR4B &= ~(1<<CS41)

ISR(TIMER4_COMPB_vect)
{
    if (tx_index!=0)
        // Reste-t-il des bits à envoyer ?
        {
            OCR4B = pulse;
            // On fixe la durée de l'impulsion.
            if ((tx_buffer&tx_index)!=0)
            {
                OCR4A = OCR4B<<2; // Période '1'
            }
            else
            {
                OCR4A = OCR4B<<1; // Période '0'
            }
            tx_index >>= 1; // Bit suivant
```

```

        }
    }
    else
    {
        if (OCR4A>=2*OCR4B)
        {
            OCR4A = OCR4B + 10; // Envoi salve de fin.
        }
        else
        {
            // Terminé, on range.
            TIMSK4 &= ~(1<<OCIE4B);
            // Désactive les interruptions.
            TIMER3_OFF;
            TIMER4_OFF;
        }
    }
}

void ir_send_code(uint32_t code)
{
    // Copie code dans tampon d'émission.
    tx_buffer = code;
    tx_index = 0x80000000; // 32 bits
    // Préparer Timer4 pour le préambule
    OCR4A = 2*12*pulse;
    OCR4B = 2*8*pulse;
    TIFR4 = (1<<OCF4B);
    // Masquer interruptions en attente
    TIMSK4 = (1<<OCIE4B);
    // Autoriser interruptions de comparaison
    // Go!
    TIMER3_ON;
    TIMER4_ON;
}

void setup(void)
{
    pinMode(ir_led,OUTPUT); // Configure en sortie.
    digitalWrite(ir_led,LOW); // OCM en mode ET.
    // Timer3 mode 4 (CTC),
    // inversion sortie si égalité.
    TCCR3A = (1<<COM3B0);
    TCCR3B = (1<<WGM32);
    OCR3A = 16000000/CARRIER/2;
    // Passage 76 kHz -> 38 kHz.
    OCR3B = OCR3A>>1; // Rapport cyclique 50%.
    // Timer4 mode 15 (MLI rapide),
    // sortie à 0 si égalité.
    TCCR4A = (1<<WGM41) | (1<<WGM40) | (1<<COM4B1);
    TCCR4B = (1<<WGM43) | (1<<WGM42);
    ir_send_code(RC_CODE);
}

void loop(void)
{
}
}
```

Tout commence dans la fonction `setup` avec la déclaration en sortie de la broche 2 (PD2, port auquel est connectée la LED IR). Sautez cette étape et rien ne sortira jamais de cette broche. Mettre cette sortie à l'état bas éteint la LED, mais fait



également passer l'OCM en mode ET. On configure ensuite *Timer3* en mode CTC (comparaison, mode 4) pour une fréquence de 38 kHz, une porteuse fréquente des télécommandes. Nota : le compteur semble programmé pour le double de cette fréquence, mais ce facteur de deux est nécessaire afin de compenser le basculement de la broche de sortie à chaque égalité, ce qui revient effectivement à diviser la fréquence par deux, pour obtenir du 38 kHz en sortie. La broche 2 est la sortie d'OC3B contrôlée par OCR3B, on configure donc ce registre à la moitié de la valeur d'OCR3A pour obtenir un rapport cyclique de 50%. *Timer4* est configuré en mode 14 (MLI rapide) où OCR4A détermine la fréquence et OCR4B le rapport cyclique. Le reste de sa configuration attendra la fonction `send_code`. Cette fonction commence par copier le code à émettre dans le tampon d'émission de 32 bits. Comme un code commence par un préambule, les durées correspondantes d'impulsion (9 ms) et de période (13,5 ms) sont chargées dans les registres *impulsion* (OCR4B) et *période* (OCR4A). Une multiplication par deux est nécessaire ici, car les impulsions et périodes sont spécifiées en microsecondes, mais l'horloge du  $\mu C$  est de 16 MHz et le prédiviseur ne peut diviser que par 1, 8, 64, 256 ou 1 024, mais pas 16. Comme on divise par 8 au lieu de 16, les durées sont doublées. L'étape suivante sera la préparation des interruptions et le démarrage des compteurs.

La routine d'interruption COMPB est appelée à la fin d'une impulsion. Quand cela se produit, et que tous les bits n'ont pas encore été envoyés, le compteur doit être reconfiguré pour une période de soit 1 125  $\mu s$  (bit = 0), soit 2 250  $\mu s$  (bit = 1), et réglé sur une largeur d'impulsion de 562,5  $\mu s$  (identique pour 0 et 1). Comme précédemment, les valeurs sont toutes multipliées par deux pour compenser la prédivision par 8 au lieu de 16. Les 32 bits du code doivent s'achever par une salve de fin, sinon le récepteur ne saurait pas si le dernier bit était un 1 ou un 0. Quand `tx_index` est à 0, tous les bits ont été envoyés. On peut alors utiliser le registre OCR4A par ex. avec une valeur particulière pour la salve de fin, ce qui implique un nouvel appel de la routine d'interruption. Si nous choisissons cette valeur particulière inférieure à 1 125  $\mu s$  (mais supérieure à OCR4B), cet état sera facilement détecté par la routine d'interruption lors de son prochain appel. Naturellement, il serait possible d'utiliser un autre mécanisme avec une variable statique ou quelque chose de similaire, mais le registre étant là, pourquoi ne pas s'en servir ? La dernière tâche qu'effectue la routine d'interruption après le préambule, les 32 bits de code et la salve de fin, est d'éteindre la lumière, euh, les compteurs, car on n'en a plus besoin.

La fonction `loop` est restée vide, car on n'émet le code qu'une fois après la pression sur le bouton *Reset*.

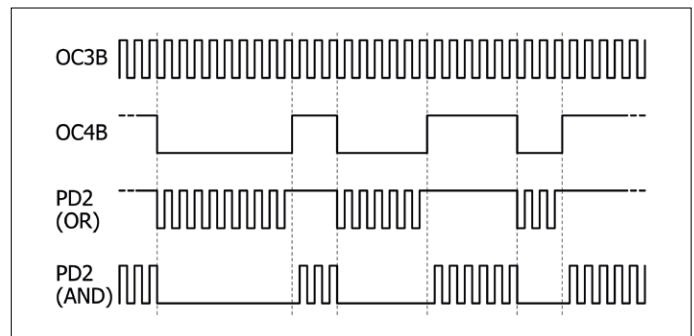


Figure 1. En mode OCM, PD2 est le résultat de OC3B OU OC4B (PORTD2 = 1) ou bien OC3B ET OC4B (PORTD2 = 0).

L'émission d'un code est réalisée uniquement avec des interruptions, le reste du programme peut faire autre chose, tant que cela reste en tâche de fond. Le programme peut scruter le bit OCIE4B du registre TIMSK4 pour vérifier si l'émission est toujours en cours. De plus, comme vous pouvez le voir, il n'y a aucune interaction entre *Timer3* et *Timer4* sauf quand la routine d'interruption les arrête tous les deux (ce qui reste optionnel), et il n'y a aucune broche à mettre à 0 ou 1. Tout est piloté directement par le matériel. La contrepartie de cette simplicité est que tout se passe sur la broche 2 uniquement. ◀

(160306 – version française : Guy Matt)

## Lien

[1] [www.elektormagazine.fr/160306](http://www.elektormagazine.fr/160306)

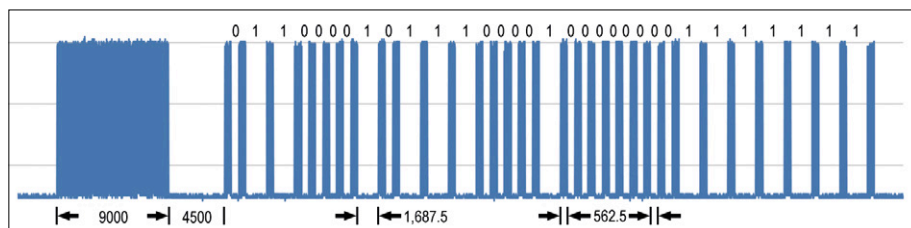
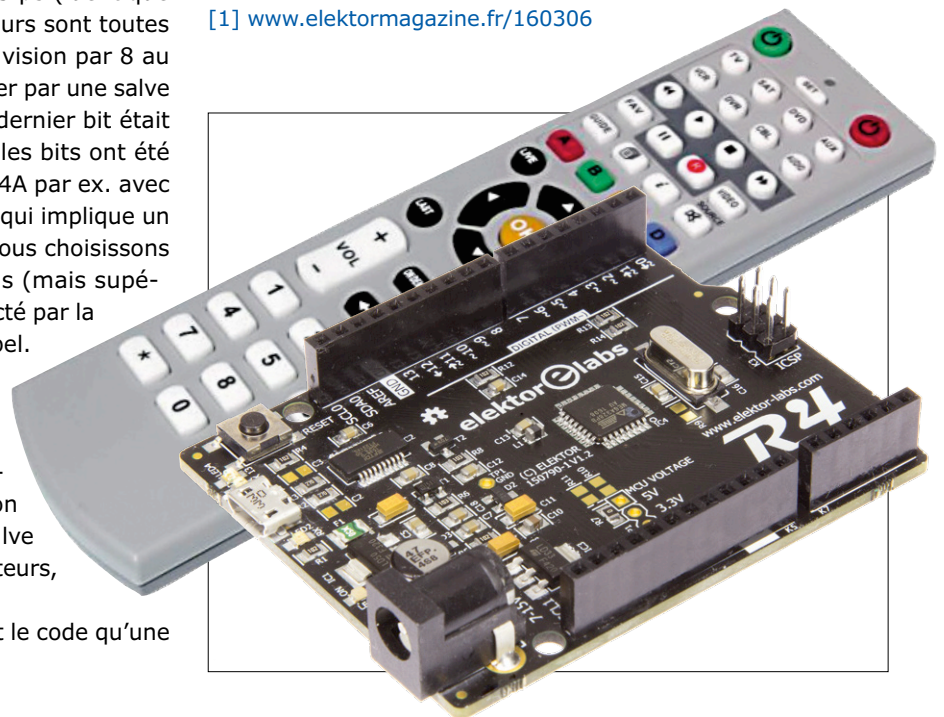


Figure 2. Un oscillogramme du signal produit sur PD2 par le croquis.



### DANS L'E-CHOPPE

→ 150790-91  
Elektor Uno R4

# testeur portatif OBD2 à base de Raspberry Pi avec un nouveau logiciel de diagnostic

Thomas Beck



Le module Pi-OB2-HAT est une carte d'extension pour Raspberry Pi qui transforme ce nano-ordinateur en une interface de diagnostic OBD2. Un inconvénient des solutions précédentes était qu'on devait les interfacer avec un émulateur de terminal. Le logiciel de diagnostic HHGui présenté ici est d'un emploi bien plus simple et commode.

Le logiciel HHGui (interface graphique pour Pi-OB2 portatif) est une variante de mon émulateur portatif OBD2, HHEmu [1], une espèce de couteau suisse pour les développements autour de l'« analyseur OBD2 NG/DIAMEX Handheld Open » (HHOpen). Ce testeur

OBD2 portatif a déjà été présenté dans Elektor (09/2009, voir [2]). L'analyseur OBD2 NG (*Nouvelle Génération*) est basé sur le module DXM-OB2 [3], un prédécesseur de l'interface AGV4000C du module Pi-OB2-HAT (*OnBoard Diagnostic - Hardware Attached on Top*).

Aujourd'hui HHGui est adapté à ce module Pi-OB2-HAT.

## Historique du développement

L'itinéraire inhabituel du développement du logiciel montre qu'il est possible de porter un micrologiciel écrit à l'origine



pour un microcontrôleur vers un Raspberry Pi ou un PC – dans l'idéal sans avoir à en modifier le code source.

À l'origine, HHEmu est un outil de développement pour mettre à jour à partir d'un PC le micrologiciel du µC AVR de l'analyseur OBD2, et sans véhicule pour pouvoir le tester. C'est pourquoi HHEmu contient également un simulateur OBD2 qui fournit des données simulées au micrologiciel via un module DXM-OB2 également simulé.

Dans la phase suivante du développement, HHEmu est doté d'un accès aux ports série d'un PC, ce qui, en combinaison avec l'extension Bluetooth (04/2010, voir [4]), offre à l'analyseur OBD2 NG des possibilités remarquables. Le micrologiciel exécuté sur un PC dans HHEmu peut alors établir, via le port série d'un adaptateur Bluetooth, une connexion avec l'extension Bluetooth d'un véritable analyseur NG et donc commander le module DXM. Il est possible de tester un nouveau micrologiciel sur un véhicule sans avoir à le charger au préalable dans l'analyseur OBD2. Une

étape importante est franchie, HHEmu devient alors LE logiciel de diagnostic des interfaces de diagnostic OBD2 basées sur DXM. De plus, le simulateur OBD2 de HHEmu peut servir de générateur de données pour le logiciel de diagnostic d'autres fabricants ou pour le développement de nouveaux logiciels. Outre la plus grande facilité de test du micrologiciel, le portage vers le PC montre un autre avantage : les fonctions de chaque nouvelle mise à jour du micrologiciel se retrouvent automatiquement à l'identique dans le logiciel de diagnostic.

Nous voilà au portage de HHEmu vers Raspberry Pi : après toutes les adaptations au module Pi-OB2 et la suppression de tous les menus spécifiques à l'analyseur OBD2 NG, la première version de HHGui est prête pour la publication.

### Modes de fonctionnement

Du fait de son histoire passée, HHGui est en premier lieu un logiciel de diagnostic OBD2 qui imite l'interface utilisateur de l'analyseur OBD2 NG et utilise en interne le micrologiciel de celui-ci, adapté au module Pi-OB2. L'étendue des

### Étendue des fonctions OBD2 du logiciel de diagnostic HHGui

HHGui supporte les services listés ci-dessous, d'après ISO 15031-5. On trouvera une description plus précise de ces services sur la page du projet [5]. Des informations sur les sous-fonctions sont disponibles sur [6], [7] et [9]. Les services et sous-fonctions effectivement disponibles dépendent de chaque véhicule et, plus précisément, de ses contrôleurs OBD2. HHGui interroge chaque contrôleur sur ses sous-fonctions disponibles et n'affiche que celles-ci.

- Supporte au maximum huit contrôleurs, selon ISO 15765-4 [10]
- Mode OBD2 0x01 : lecture des valeurs instantanées (*Current Data*) ; sous-fonctions supportées (*Parameter Identifier, PID*) : 0x00 à 0x60, 0x70, 0x80, 0x8D
- Mode OBD2 0x02 : lecture des données d'environnement des défauts (*Freeze Frame Data*), PID supportés : voir mode 0x01
- Mode OBD2 0x03 : lecture des défauts confirmés (*Confirmed Diagnostic Trouble Codes, DTC*)
- Mode OBD2 0x04 : effacement des défauts, des données d'environnement des défauts, du statut des voyants indicateurs de défaut (*Malfunction Indicator Lamps, MIL*), des données du moniteur OBD2 et autres
- Mode OBD2 0x07 : lecture des défauts en attente (*Pending DTC*)
- Mode OBD2 0x09 : lecture des informations du véhicule (*Vehicle Information*) ; sous-fonctions supportées : 0x00, 0x02, 0x04, 0x06, 0x08, 0x0A, 0x0B
- Mode OBD2 0x0A : lecture des défauts permanents (*Permanent DTC*)

fonctions actuelles du micrologiciel est détaillée dans l'**encadré**.

HHGui affiche ses menus utilisateur et les données OBD2 comme sur un afficheur à cristaux liquides, lequel est simulé et adapté sur l'écran graphique, plus grand, du RPi. On a là un compromis entre l'utilisation de l'écran tactile officiel de 7 pouces recommandé pour le RPi et d'écrans plus petits de 320×240 pixels. Toutefois dans ce cas, une modification du module Pi-OB2 est nécessaire, voir la description en ligne dans la première mise à jour du projet [5].

La deuxième fonction implémentée dans le HHGui est un simulateur OBD2, qui simule le module Pi-OB2 ainsi que jusqu'à huit contrôleurs (*Electronic Control Units*), ce qui offre la possibilité de tester tout le fonctionnement du logiciel en l'absence du module Pi-OB2 ou de véhicule. Comme un véhicule réel donné ne présente qu'un sous-ensemble de toutes les fonctions OBD2 (dont une grande partie se répartit entre modèles à essence et modèles diesel), on dispose d'un moyen simple de tester l'ensemble des fonctions. De plus, cette variante est totalement sans danger, l'effacement involontaire de données OBD2 dans le véhicule étant impossible. La gamme de fonctions OBD2 du simulateur est identique à celle du micrologiciel. La description des commandes à touches du simu-

lateur OBD2 est disponible en ligne, voir le projet dans le labo d'Elektor en [1].

### Possibilités du diagnostic OBD2

Revenons à la fonction principale : le logiciel de diagnostic OBD2. Quels services peut-on ou non en attendre pour un véhicule ?

Quand on lit la spécification la plus importante sur le sujet ISO 15031-5 [6] (ou SAE J1979 [7], avec le même contenu), on se rend compte qu'avec les services OBD2, on ne peut lire et effacer partiellement que des données et des défauts relatifs aux émissions. Ces données sont pour la plupart fournies par le contrôleur du moteur. Pour les véhicules à boîte de vitesses automatique, le contrôleur de celle-ci fournit aussi des données OBD2. Plus rarement, d'autres contrôleurs ou des contrôleurs de plusieurs moteurs (par ex. sur les véhicules hybrides) peuvent fournir des données. Les configurations telles que le verrouillage automatique des portières ou la suppression du signal sonore de la ceinture de sécurité ne sont pas du domaine des services OBD2. À l'exception du voyant de contrôle du moteur, on ne peut pas non plus éteindre les voyants d'alarme du tableau de bord, ni réinitialiser les indicateurs ou les intervalles de maintenance. Tout cela n'est possible qu'avec un moyen de diagnostic spécifique du constructeur, qui varie

d'un constructeur à l'autre.

Vous êtes déçu !? Toutefois il y a un avantage à cette prééminence des constructeurs dans la standardisation du diagnostic OBD2 : aujourd'hui, 20 ans après son introduction aux États-Unis (en l'an 2000 en Europe pour les véhicules à essence), cela fonctionne pour pratiquement n'importe quel véhicule. Depuis 2008, le nombre de protocoles d'échange de données OBD2 s'est fortement réduit, bientôt à un seul, les véhicules neufs ne peuvent plus utiliser que le protocole CAN.

Si l'on examine plus attentivement les sous-fonctions OBD2, on découvre un nombre ahurissant de données de capteurs, de compteurs et autres (en tout, plus de cent), propres à éveiller la curiosité de l'explorateur. De plus, contre toute attente, on n'y trouve pas que des données du contrôleur du moteur. Certaines d'entre elles, étiquetées par le diagnostic comme provenant du moteur, sont en fait issues d'autres contrôleurs, comme celui des freins.

Parmi les données OBD2 faciles à interpréter, on compte celles qui sont déjà affichées par le véhicule, mais après une transformation :

- par filtrage passe-bas (température extérieure, niveau de carburant et d'une manière générale tous les afficheurs à aiguille réalisés avec des moteurs pas-à-pas)
- avec une fonction plateau (température du liquide de refroidissement) [8]
- avec des décalages ajoutés pour respecter des contraintes réglementaires d'affichage minimum, des tolérances, ou simplement des désirs du constructeur (vitesse, tachymètre)

Dans ces cas, le diagnostic OBD2 fournit des données « brutes », non transformées, par ex. la vitesse réelle et non celle affichée.

Ensuite, il y a les données qu'on aimerait bien voir, mais que le constructeur n'a pas prévu d'afficher, par ex. la température de l'huile ou la pression du

turbocompresseur. Même la puissance fournie par le moteur est une information utile.

Pour l'utilisateur lambda, il est intéressant de connaître les données OBD2 qui signalent que le véhicule est en état de passer le prochain contrôle technique. Pour cela il faut que tous les programmes de contrôle OBD2 (les moniteurs OBD2) aient été exécutés avec succès au moins une fois depuis le dernier effacement des défauts. Cet état est atteint quand HHGui affiche dans le menu *Inspection/Maintenance-Readiness* l'état « ...monitoring ready: YES » pour tous les moniteurs OBD2.

À côté de ces données OBD2 faciles à interpréter, il y en a beaucoup d'autres dans le domaine du réglage *lambda* (rapport air/carburant), plutôt réservées aux spécialistes. Les lister toutes ici serait beaucoup trop long. Comme les versions les plus récentes des normes mentionnées ci-dessus ne sont disponibles qu'à titre (très) onéreux, il est recommandé de consulter Wikipedia [9].

## PI-OBD : le matériel

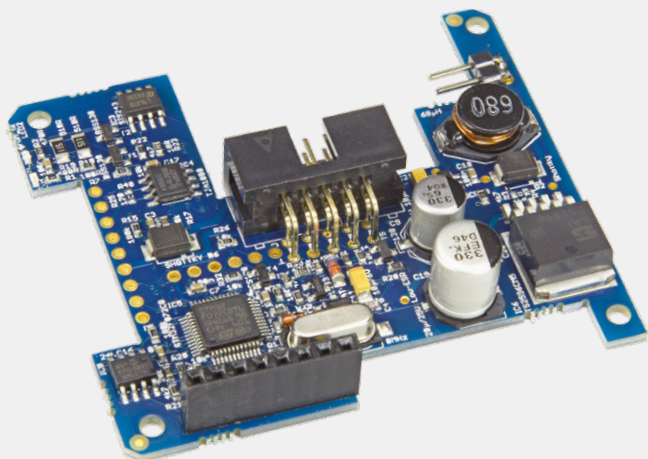
Le module Pi-OBD-HAT (disponible dans l'e-choppe) de Diamex transforme le Raspberry Pi en un gestionnaire de diagnostic OBD2. Si l'on équipe le RPi avec un écran tactile de 7 pouces, on a en main un système de diagnostic autonome pour véhicule, avec une interface OBD2.

Le Raspberry Pi ainsi que l'écran sont alimentés par

du démarrage du Pi-OBD. La communication s'effectue au travers du port série configuré 8N1 (8 bits/sans parité/1 bit de stop) à la vitesse de 115.200 baud.

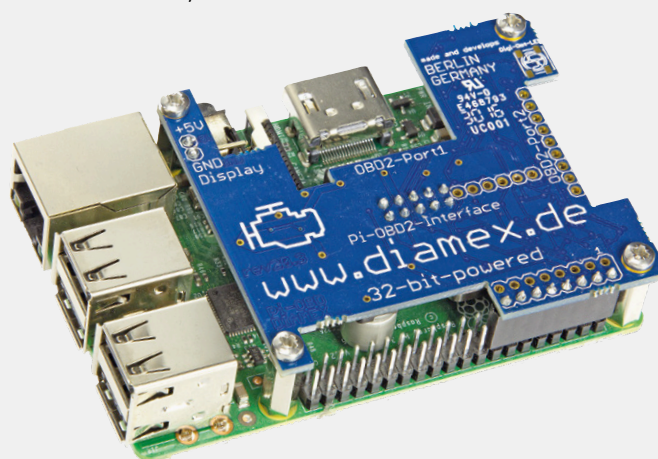
### Montage

Pour commencer, on enfiche le module Pi-OBD sur le RPi.



l'alimentation de bord du véhicule.

Pour cet appareil, toutes les versions du RPi conviennent en principe, toutefois, compte tenu de la vitesse plus élevée de leur processeur, il est indiqué de donner la préférence au RPi 2 modèle B ou RPi 3 modèle B. Le module Pi-OBD n'occupe que huit broches de l'interface GPIO du RPi, pour l'alimentation, le port série, une réinitialisation et l'activation



Le connecteur femelle à 8 broches doit être enfoncé sur la rangée extérieure du connecteur GPIO du RPi. Ensuite le module est fixé solidement au RPi au moyen des entretoises fournies. Le RPi est alimenté par les broches du connecteur GPIO dès que le module Pi-OBD est connecté par câble à la prise OBD2 du véhicule. Excepté pendant le test et la configuration, aucune alimentation par la prise micro-USB du RPi n'est nécessaire. Pour alimenter l'écran de 7 pouces



Passons maintenant à la partie intéressante pour le développeur.

## HHGui : structure et fonctionnement

### Principe de fonctionnement

Le micrologiciel est exécuté dans un fil (*thread*), toutes les entrées et sorties sont traitées dans d'autres fils. Pour cela, il faut simuler des registres et des interruptions du  $\mu$ C.

### Réalisation effective

HHGui se compose de six fils. La **figure 1** en montre un synoptique simplifié avec les interactions entre les fils. Les noms des fils dans les fichiers sources sont différents du fait de la longue histoire de ce logiciel. Le fil **Pi-OBD-Module** y porte le nom de **DxmThread** et le fil **OBD2-Analyser NG** s'appelle **HhopenThread**. Dans la suite, nous allons utiliser ces noms plus courts.

Le **HhopenThread** est le fil dans lequel est exécuté le micrologiciel de l'analyseur OBD2. Ce fil communique de manière

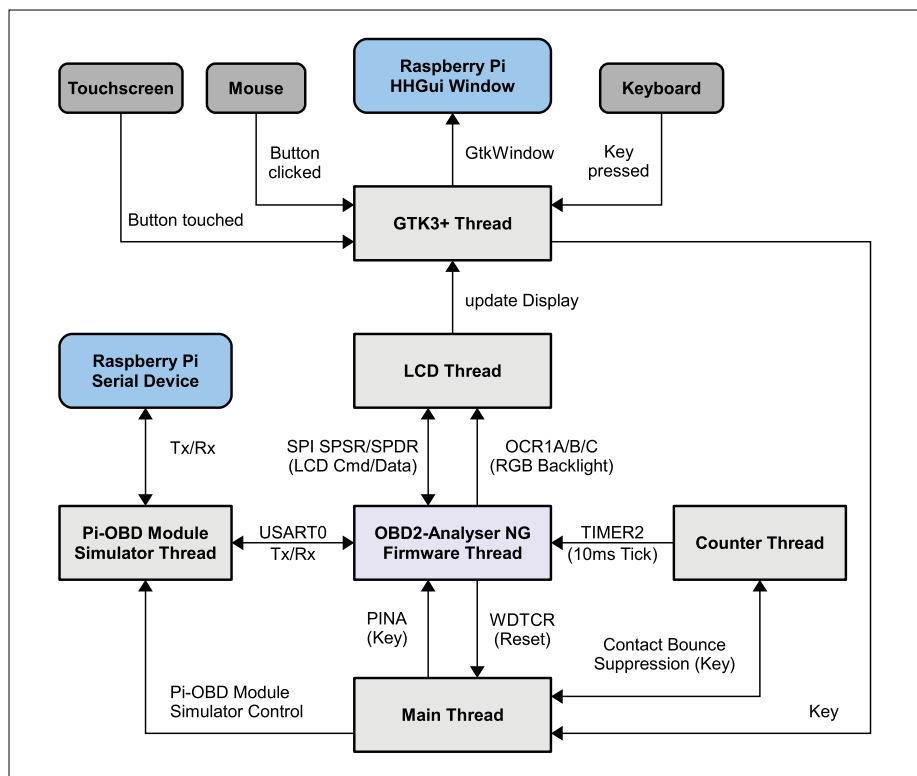
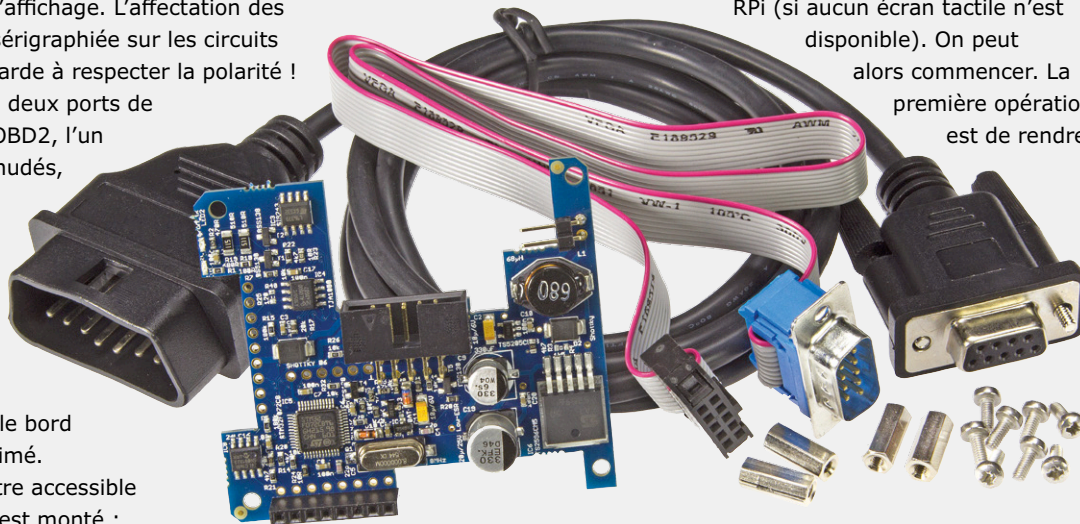


Figure 1. HHGui : diagramme fonctionnel.

du RPi, il faut connecter les deux broches d'alimentation 5 V de son module d'affichage aux broches correspondantes du module Pi-OBD. La **photo** montre cette connexion avec, en bas, le RPi équipé du module OBD et, en haut, l'arrière de l'écran avec le module d'affichage. L'affectation des différentes broches est sérigraphiée sur les circuits imprimés. Prenez bien garde à respecter la polarité ! Le module est équipé de deux ports de raccordement du câble OBD2, l'un (port 2) pour des fils dénudés, l'autre (port 1) pour un câble plat terminé à l'autre bout par un connecteur Sub-D. On peut monter un connecteur plat (mâle ou femelle) sur le port 2 qui se trouve sur le bord extérieur du circuit imprimé. Ce port a l'avantage d'être accessible même quand le module est monté ; attention toutefois à une possible erreur de polarité. Pour le port 1, équipé d'une prise carénée avec détrompeur, une telle erreur est impossible, la présence d'un connecteur Sub-D interdit l'inversion. Il faut toutefois connecter le câble plat avant l'assemblage du RPi et du module, cette opération est impossible par la suite.

### Mise en service

La mise en service du Pi-OBD-HAT nécessite des connaissances de base du Raspberry Pi, de l'utilisation de Linux et de l'installation de logiciels. Pour la configuration et l'utilisation, il faut raccorder un clavier et une souris au RPi (si aucun écran tactile n'est disponible). On peut alors commencer. La première opération est de rendre



disponible le port série du RPi, opération dont la description sortirait du cadre de cet article, d'autant plus qu'elle présente des différences entre les différents modèles du RPi. Cette procédure est toutefois décrite en détail dans le mode d'emploi [11]. Après un test de fonctionnement au moyen d'un émulateur de terminal, l'appareil est prêt à l'emploi.

variée avec les autres fils. Le registre SPDR sert à transférer les données à afficher au fil `LcdThread`, lequel reçoit également les données RVB pour le réglage du rétroéclairage par les registres OCR1A/B/C. Les commandes AT ainsi que les requêtes OBD sont transmises au fil `DxmThread` au travers du registre UDR0 par lequel passent aussi les réponses. Le fil `MainThread` demande un redémarrage du micrologiciel ou de HHGui par l'intermédiaire du registre WTCR et reçoit les actions sur les touches par le registre PINA. Le fil `HhopenThread` est piloté par les fils `CounterThread` et `DxmThread` par appel de ses fonctions de traitement des interruptions. D'autres détails suivent dans les descriptions des cinq autres fils. Le fil `MainThread` est responsable de l'exploitation des paramètres des lignes de commande, de l'initialisation du port série, de la création des autres fils et, lors de la fin d'exécution, de la restitution des ressources. Entretemps, ce fil exploite dans une boucle sans fin les états des touches qui lui sont fournis par le fil `GtkThread`. Parmi ces touches, il faut distinguer les touches de commande du simulateur OBD2 des touches Up/Down/ESC/OK pour le fil `HhopenThread`. Pour ce dernier, une fonction de suppression du rebond des touches incluse dans le micrologiciel doit être exécutée : les touches doivent être présentées au fil `HhopenThread` à travers le registre PINA, dans l'état « pressé » pendant 40 ms, pour que le micrologiciel identifie une pression sur une touche. Le délai de 40 ms est évalué en collaboration avec le fil `CounterThread`. Les touches qui ne sont utilisées que pour le simulateur OBD2 provoquent des modifications des réglages dans le fil `DxmThread`.

La boucle sans fin de gestion des pres-

sions des touches est quittée sur une réinitialisation du chien de garde provoquée par le fil `HhopenThread` (suite à une pression prolongée de la touche ESC) ou bien par la fin de HHGui sur pression de la touche q (*quit*).

Le fil `LcdThread` analyse les commandes et les données à afficher que lui transmet le fil `HhopenThread` à travers l'interface SPI simulée et les convertit en directives pour la bibliothèque graphique utilisée GTK3+. Les valeurs RVB lues au travers des registres OCR1A/B/C pour le rétroéclairage, combinées avec la valeur du contraste (lue par l'interface SPI), déterminent les couleurs affichées. Comme l'afficheur utilisé par l'analyseur OBD2 NG utilise un contrôleur ST7565R, le fil `LcdThread` contient un simulateur de ce contrôleur pour les commandes ST7565R utilisées par le micrologiciel.

Le fil `GtkThread` est nécessaire, car la boucle principale de GTK3+ (fonction `gtk_main()` de la bibliothèque GTK3+) est bloquante. Les fonctions de dessin de GTK3+ ne peuvent être appelées que de l'intérieur du fil `GtkThread` (ou dans le contexte de la boucle principale de GTK3+). Quand le fil `LcdThread` demande un rafraîchissement de l'affichage, il installe pour cela une fonction *callback* qui est appelée ultérieurement par la boucle principale de GTK3+. Le fil `GtkThread` exploite les événements d'entrée de toutes sortes et les transforme soit en pressions sur les touches Up/Down/ESC/OK, soit en événements de touches qui pilotent le simulateur OBD2.

Le fil `CounterThread` sert en premier lieu à émuler les interruptions cadencées à 10 ms du `Timer2`. Pour cela, il appelle toutes les 10 ms la fonction de traitement des interruptions (*Interrupt Service Routine, ISR*) `TIMER2_COMP_vect()`. D'autre

part, il collabore avec le fil `MainThread` pour la fonction anti-rebond des touches, comme déjà mentionné.

En mode *logiciel OBD2*, le fil `DxmThread` fonctionne comme adaptateur d'interface et transmet les données de manière transparente. Les données reçues par le registre USART de données d'entrée-sortie UDR0 (commandes AT pour le module Pi-OBD ou requêtes OBD2) sont envoyées par le port série configuré au véritable module Pi-OBD. Après chaque octet lu, le micrologiciel est averti par l'appel de l'*ISR USART\_UDRE\_vect()* que cet octet a été envoyé et que l'octet suivant peut être écrit dans le registre. Les réponses du module Pi-OBD sont reçues par le port série et envoyées au micrologiciel via le registre UDR0. La réponse (OK, caractère > d'invite du module Pi-OBD, erreur ou données OBD2) est également écrite octet par octet dans le registre UDR0 et signifiée au micrologiciel par appel de l'*ISR USART0\_RX\_vect()*, lequel vient lire les données dans ce registre.

En mode *simulateur OBD2*, le fil `DxmThread` simule le module Pi-OBD2 et un ou plusieurs contrôleurs. Il n'y a alors pas de communication à travers le port série physique. ◀

(160204 – version française : Helmut Müller)



### DANS L'E-CHOPPE

- 17944  
module Pi-OBD-HAT  
(monté et testé)
- 17415  
Écran tactile de 7 pouces pour  
Raspberry Pi, modèle officiel
- 17631  
Raspberry Pi 3

## Liens

- [1] HHEmu : [www.elektormagazine.fr/labs/firmware-update-and-emulator-for-obd2-analyser-ng-wireless-obd2](http://www.elektormagazine.fr/labs/firmware-update-and-emulator-for-obd2-analyser-ng-wireless-obd2)
- [2] *Analyseur OBD2 (Nouvelle Génération)*, Elektor 09/2009 : [www.elektormagazine.fr/magazine/elektor-200909/11390](http://www.elektormagazine.fr/magazine/elektor-200909/11390)
- [3] Module DXM : [www.diamex.de/dxshop/DIAMEX-DXM-OBd2-Modul](http://www.diamex.de/dxshop/DIAMEX-DXM-OBd2-Modul)
- [4] *OBd-2 sans fil, module d'extension Bluetooth pour l'analyseur OBD2 NG*, Elektor 04/2010 : [www.elektormagazine.fr/090918](http://www.elektormagazine.fr/090918)
- [5] HHGui : [www.elektormagazine.fr/labs/obd2-for-raspberry-pi](http://www.elektormagazine.fr/labs/obd2-for-raspberry-pi)
- [6] [www.iso.org/iso/home/store/catalogue\\_ics/catalogue\\_detail\\_ics.htm?csnumber=66368](http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=66368)
- [7] [http://standards.sae.org/j1979\\_201408/](http://standards.sae.org/j1979_201408/)
- [8] Fonction plateau : <http://www.t4-wiki.de/wiki/Plateau-Funktion>
- [9] [https://en.wikipedia.org/wiki/OBD-II\\_PIDs](https://en.wikipedia.org/wiki/OBD-II_PIDs)
- [10] [www.iso.org/iso/home/store/catalogue\\_ics/catalogue\\_detail\\_ics.htm?csnumber=67245](http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=67245)
- [11] Pi-OBD-HAT : [www.elektor.fr/pi-obd-hat-obd2-module-for-raspberry-pi](http://www.elektor.fr/pi-obd-hat-obd2-module-for-raspberry-pi)



# bruits de labo



Divers projets sont postés en permanence sur le site du labo d'Elektor. Certains sont plus élaborés que d'autres ; cependant, tous sont intéressants et basés sur le bien le plus précieux dont nous disposons sur terre : l'amour le silicium.

## Blindez vos communications cryptées P2P

Qu'est-ce qui vous permet de croire que vos communications sont sûres ? Vous avez un de ces « cryptophones » prétendument inviolables ? Vous faites confiance à un logiciel de cryptage ? Avez-vous entendu parler de versions vérolées de programmes pourtant sérieux comme PGP ? Que savez-vous des mouchards intercepteurs qui espionnent tout ce que vous tapez sur votre clavier ? D'ici 2020, tout ordinateur connecté à l'internet devra être considéré comme contaminé. Si vous cherchez vraiment à vous protéger, voici le système de communication cryptée de pair à pair qu'il vous faut !



<https://goo.gl/OhXNEu>

## Construisez un dé électronique (sans connexion au nuage)

Parfois elle vous démange cette envie de construire un petit rien, juste pour le plaisir ou pour retrouver les délicieux effluves de soudure et de décapant. Ce dé électronique est fait pour vous : facile à construire, à faire marcher et à utiliser. Vous pouvez même en monter plusieurs. Aucun microcontrôleur, aucune alimentation à découpage, aucune connexion dans le nuage : seulement un bon vieux 555 et un compteur Johnson 4017. C'est idéal pour oublier les soucis quotidiens ou vous perfectionner dans l'art du soudage.



<https://goo.gl/4tiAu1>

## Smartwatch ou comment ne pas devenir horloger

nWatch est le nom d'une plateforme de développement si portable qu'elle se porte au poignet ! Son  $\mu C$  est un STM32, son écran graphique polychrome est tactile, il y a un décodeur MP3, des capteurs de position, une interface Bluetooth 4.0, etc. Un modèle de boîtier imprimable en 3D est disponible. Le fait d'être à code source ouvert rend ce projet vraiment attrayant : on le modifie comme on veut... ou comme on peut. La mission initiale est essentiellement didactique. Ce projet est l'un des gagnants du concours *Elektor Fast Forward Award*.

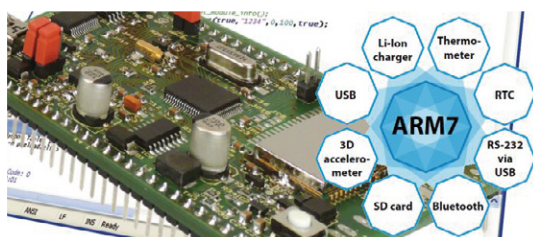


<https://goo.gl/V1EUOC>

## Le sceptre — l'IdO n'est pas une nouveauté

Le Sceptre d'Elektor est une petite carte ARM7 à base de LPC2148. Elle embarque le Bluetooth, un chargeur de batterie, un lecteur de carte SD, un accéléromètre, un thermomètre et l'USB. Déjà publiés dans Elektor en 2010 comme projet à « code et matériel ouverts », les fichiers ont été dispersés par les diverses refontes du site et leur lot de renvois et autres changements d'hôtes. Désormais ces sept articles et les projets associés ont été regroupés pour vous sur le site du labo d'Elektor. ◀

(160293 – version française : Yves Georges)



<https://goo.gl/kuCiyR>



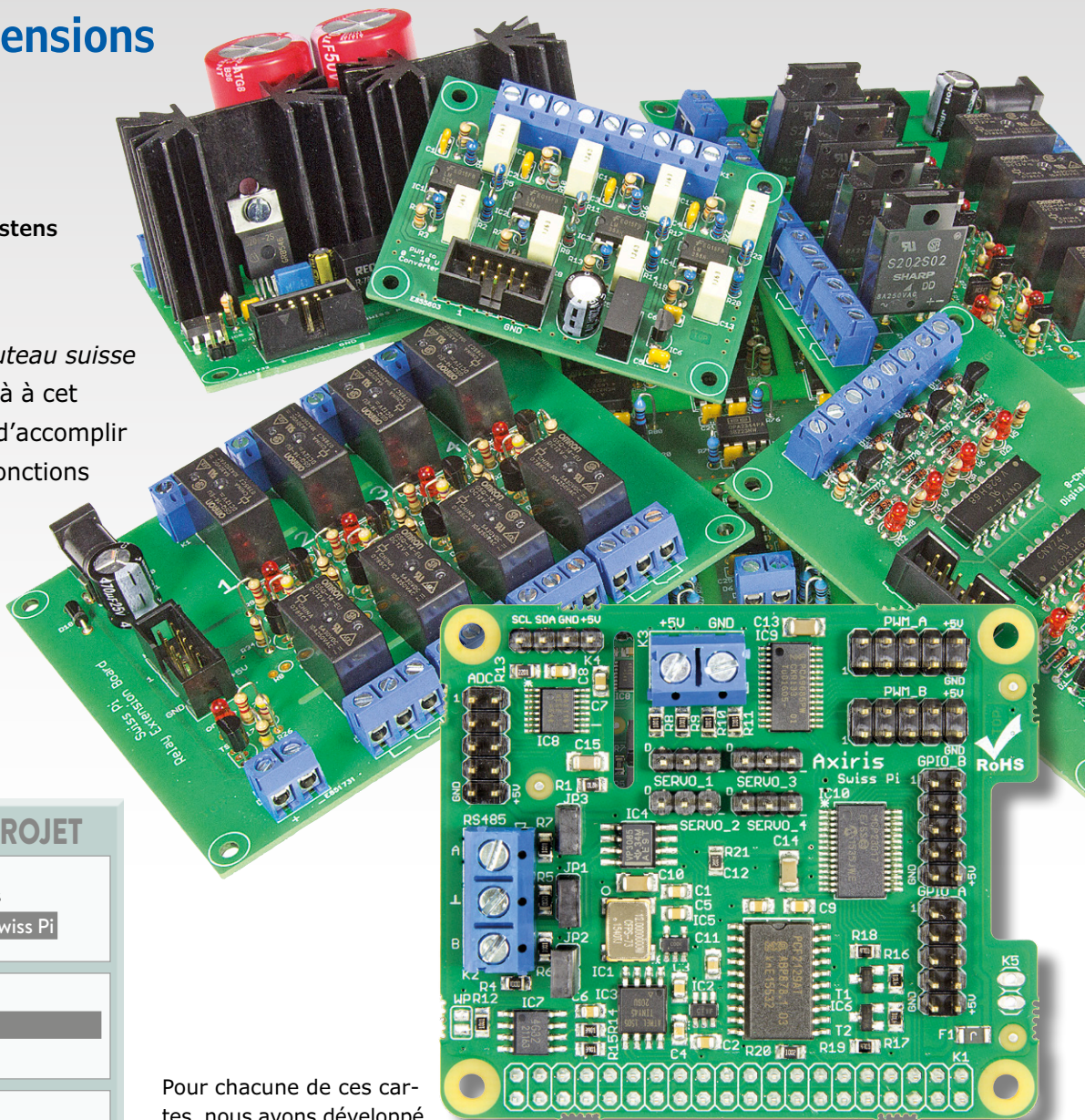
# Swiss Pi

## toujours plus polyvalent

grâce aux extensions matérielles

Peter S'heeren & Ilse Joostens  
(Belgique)

La carte d'extension *couteau suisse* pour RPi [1] permet déjà à cet ordinateur mono-carte d'accomplir toute une panoplie de fonctions utiles. Continuons sur notre lancée avec une série de cartes d'extension matérielle pour différents domaines courants.



### À PROPOS DU PROJET



Microcontrôleurs

Raspberry Pi Swiss Pi



débutant

→ connaisseur

expert



de 30 mn à 1 h30 par carte



poste de soudage avec  
panne adéquate



entre 25 € et 50 € par carte

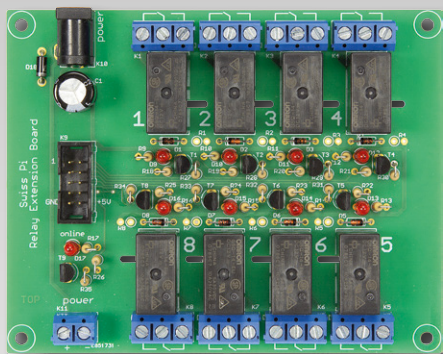
Pour chacune de ces cartes, nous avons développé un circuit imprimé dont vous pouvez télécharger les fichiers Eagle et Gerber [2]. Ils vous permettront de faire réaliser ces circuits imprimés par le fabricant de votre choix. Autant que possible, nous avons privilégié les composants traversants aux CMS, question de vous simplifier la construction. Et si vous souhaitez créer vos propres cartes d'extension, vous trouverez ici l'inspiration nécessaire. Sans oublier que ces

circuits ne sont pas forcément tributaires du Swiss Pi, ils accompagneront sans difficulté d'autres plateformes, un circuit à microcontrôleur de votre cru ou une carte Arduino.

### Carte à relais à 8 canaux

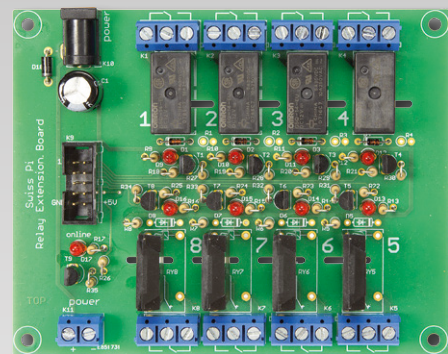
La carte à huit relais (**fig. 1**) utilise des relais électromagnétiques de la série G5Q d'Omron. En lieu et place de ces





relais ordinaires, vous pouvez installer des modèles à semi-conducteurs S202S02 de Sharp. Ceux-ci ne sont plus fabriqués, mais ils sont encore large-

ment disponibles sur eBay. Un autre type donnerait des complications, parce que celui de Sharp est de loin le seul relais statique qui possède la même empreinte que ceux de la série G5Q. L'entrée du S202S02 se fait sur une seule LED, elle a donc besoin d'une résistance série de limitation de courant. Malgré ses dimensions modestes, le relais G5Q répond à toutes les exigences en matière de rigidité diélectrique et de distances de sécurité pour supporter la tension du secteur. Les bornes du relais à semi-conducteur sont un peu plus serrées, aussi avons-nous pratiqué une saignée dans le circuit imprimé pour éviter les courants super-



ficiels de fuite. Même si le relais G5Q est capable de commuter jusqu'à 10 A, la largeur des pistes n'a été dimensionnée que pour 5 A.

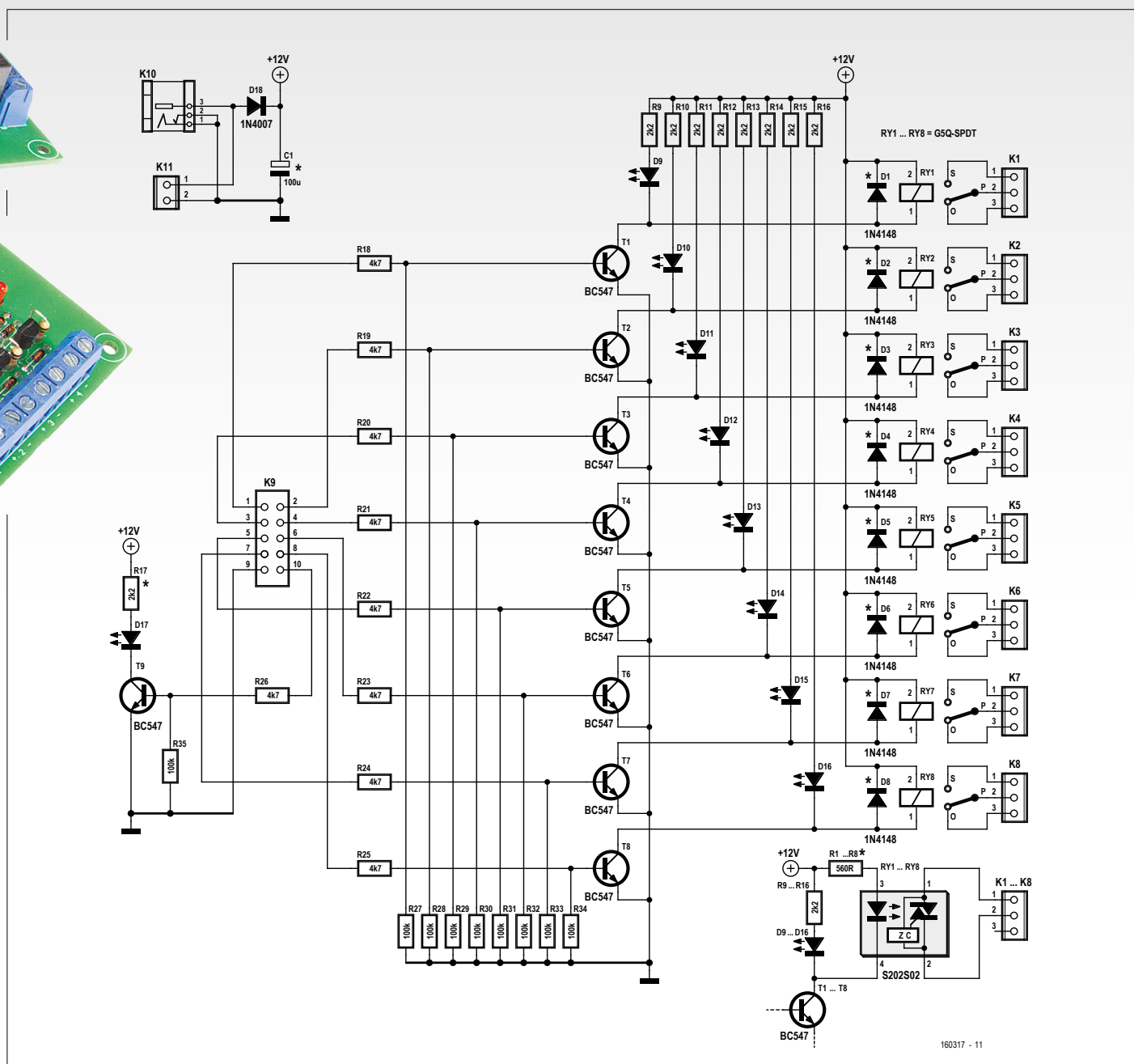


Figure 1. La carte à huit relais convient aussi bien pour les modèles électromagnétiques G5Q SPDT que pour les statiques S202S02.



## LISTE DES COMPOSANTS : CARTE À RELAIS

### Résistances :

R1 à R8 = 560  $\Omega$ \*  
 R9 à R17 = 2,2 k $\Omega$   
 R18 à R26 = 4,7 k $\Omega$   
 R27 à R35 = 100 k $\Omega$

### Condensateurs :

C1 = 100  $\mu$ F/16 V

### Semi-conducteurs :

D1 à D8 = 1N4148\*\*  
 D9 à D17 = LED, 3 mm, rouge  
 D18 = 1N4007  
 T1 à T9 = BC547B

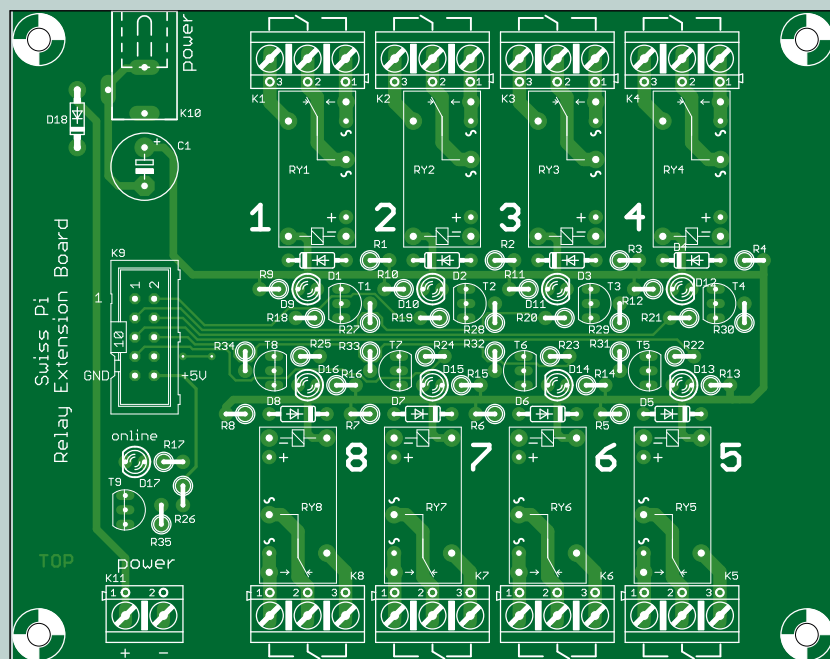
### Divers :

RY1 à RY8 = G5Q-14-EU DC12V ou S202S02\*\*\*  
 K1 à K8 = bornier à vis, à 3 pôles, au pas de 5 mm  
 K9 = embase à collerette à 10 broches  
 K10 = connecteur d'alimentation CC coaxial  
 K11 = bornier à vis, à 2 pôles, au pas de 5 mm

\*à installer avec relais statique S202S02

\*\*à installer avec relais G5Q

\*\*\*à installer par canal au choix ; pour le relais G5Q, on peut choisir une autre tension si nécessaire, mais alors il faut adapter les valeurs des résistances R1 à R17 et du condensateur C1.



Ce sont des transistors NPN standard qui assurent la commande des relais. Un témoin à LED est aussi prévu. Une autre LED indique quand la carte et le Swiss Pi sont sous tension.

Si nécessaire, vous pouvez aussi installer des relais G5Q d'une autre tension, 24 V par exemple, auquel cas il faudra recalculer un certain nombre de résistances.

### Huit entrées numériques protégées

Cette carte d'extension offre huit entrées à isolation galvanique par photocoupleur qui accepte toute tension continue dans

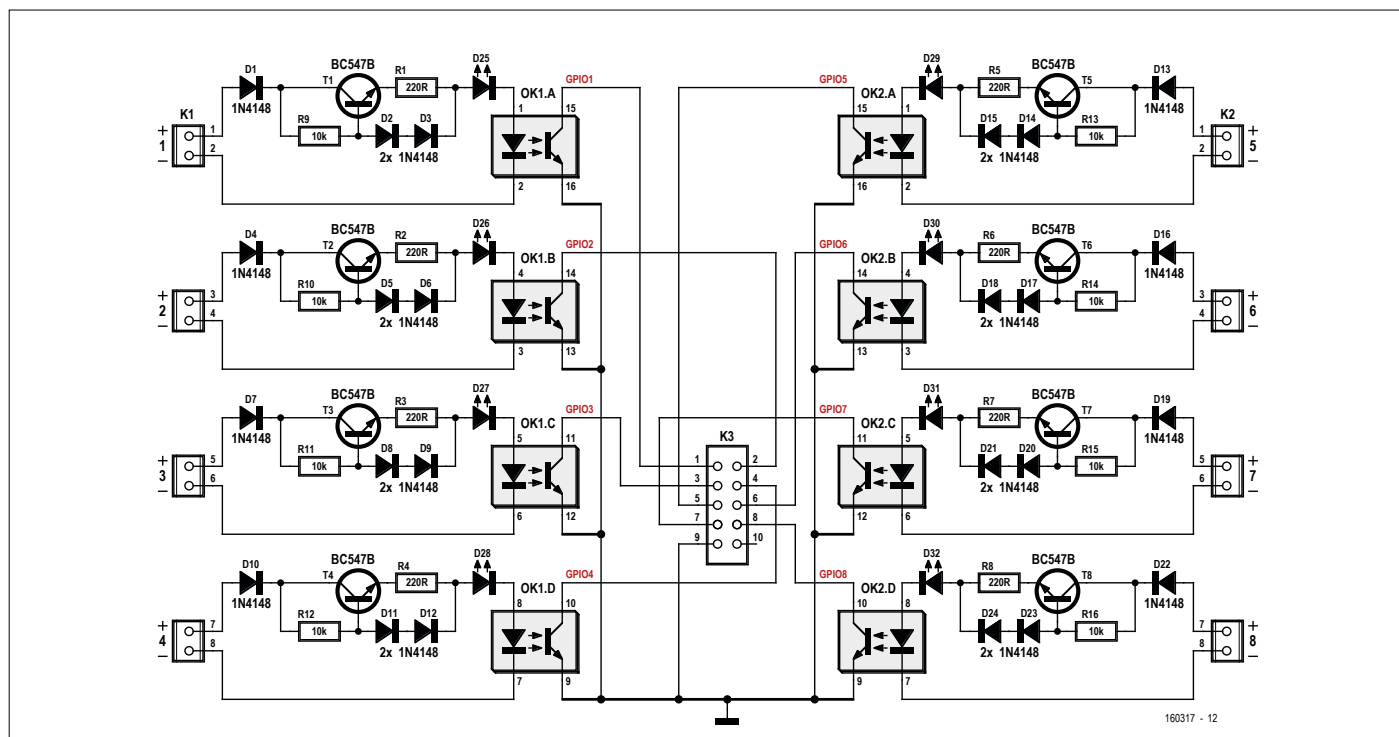


Figure 2. Les huit photocoupleurs assurent l'isolation galvanique des huit entrées numériques.



la plage de 5 à 45 V (cf. **fig. 2**). Des sources de courant par transistors BC547 alimentent les LED des photocoupleurs et les témoins à LED à débit relativement constant par rapport à la tension d'entrée. Le courant y est égal aux deux tensions de seuil des diodes 1N4148 moins la tension base/émetteur du transistor, le tout divisé par 220, la valeur des résistances R1 à R8. Dans la fourchette de tensions d'entrée de 5 à 45 V, il circule dans les diodes un courant entre quelques dizaines de microampères et quelques milliampères. La caractéristique de la 1N4148 montre que c'est dans ce domaine que la tension de seuil varie le plus en fonction du courant. En pratique, le débit produit par les sources de courant n'est pas constant sur toute la plage des tensions d'entrée : il peut aller de 3 à 6 mA.

Si l'on inversait la polarité sur la source de courant, la tension entre collecteur et émetteur du transistor serait négative. Le transistor continuerait à fonctionner comme un (piètre) NPN avec un gain très faible et une basse tension de claquage. Tant que la tension entre collecteur et émetteur reste inférieure à environ -6 V, le transistor reste en conduction et le courant n'est plus limité que par la résistance de 220  $\Omega$  et le courant de fuite

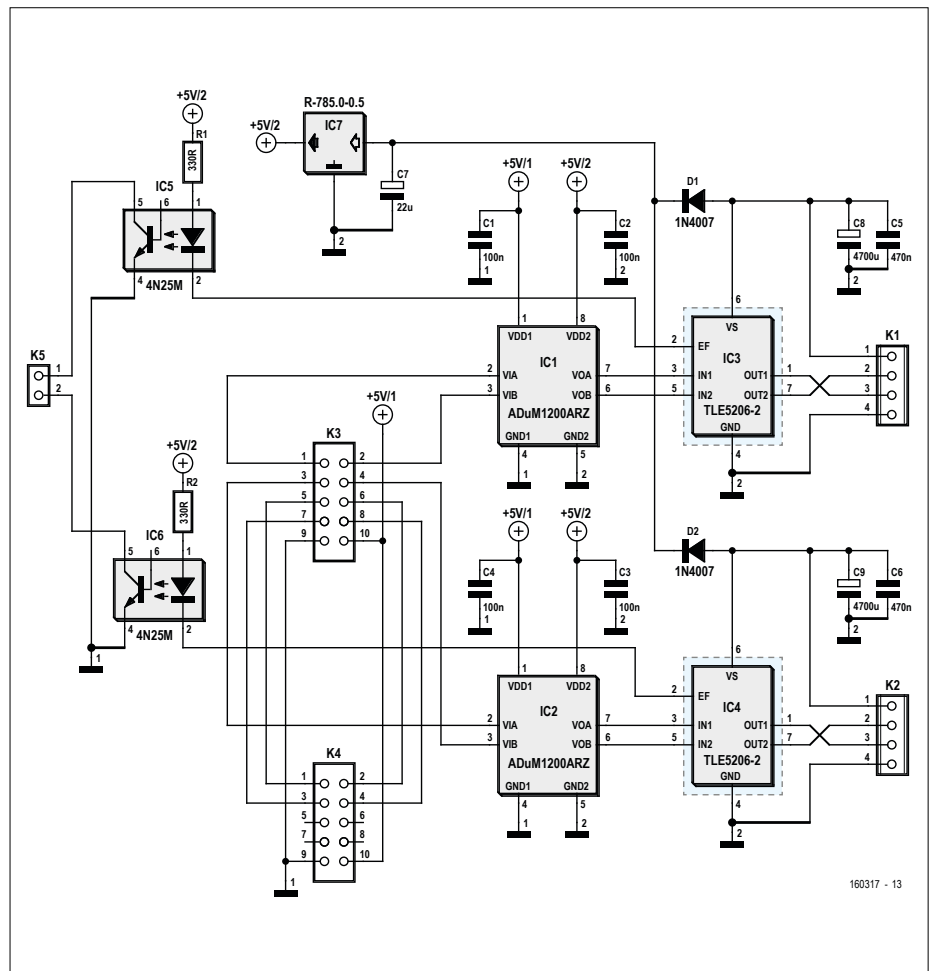


Figure 3. Les commandes de moteur à CC avec séparation galvanique peuvent se coupler par deux pour la commande de huit moteurs au maximum.



## LISTE DES COMPOSANTS : 8 ENTRÉES NUMÉRIQUES

### Résistances :

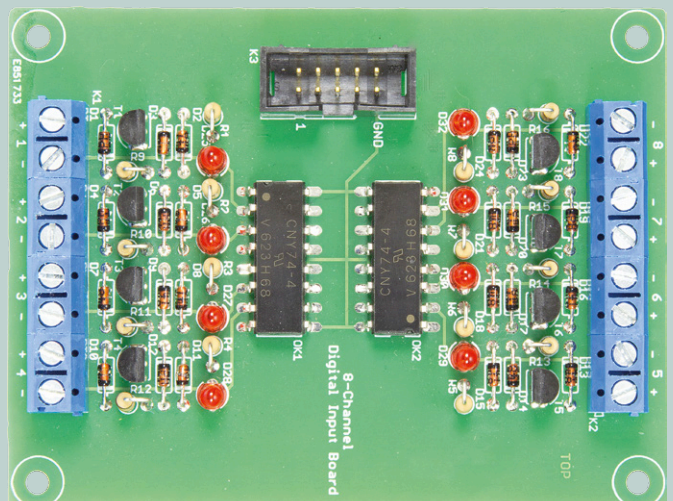
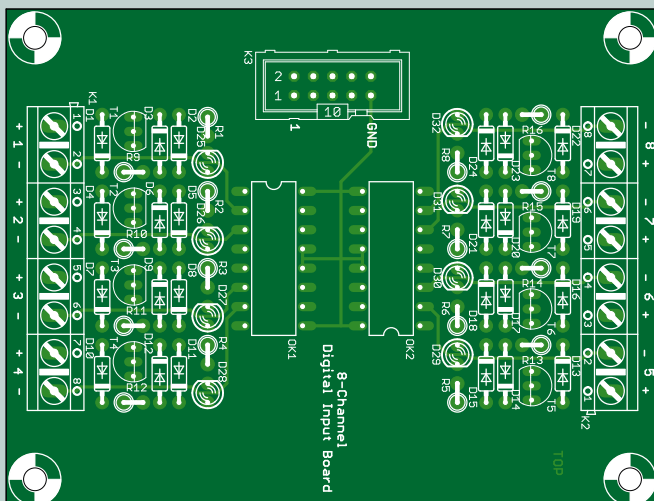
R1 à R8 = 220  $\Omega$   
R9 à R16 = 10 k $\Omega$

### Semi-conducteurs :

D1 à D24 = 1N4148  
D25 à D32 = LED, 3 mm, rouge  
T1 à T8 = BC547B  
OK1, OK2 = CNY74-4

### Divers :

K1, K2 = bornier à vis, à 8 pôles, au pas de 5 mm  
K3 = embase à collerette à 10 broches



des LED du photocoupleur et du témoin. Pour éviter d'éventuels dégâts sur les LED, principalement à cause de tensions d'entrée plus élevées, il y a une diode 1N4148 supplémentaire contre l'inversion de polarité.

Comme LED témoins, vous pouvez choisir d'autres couleurs que le rouge du prototype. Il faut alors malgré tout tenir compte de variations dans la plage de tensions d'entrée, la tension minimale en particulier.

Côté Swiss Pi, les photocoupleurs présentent des sorties à collecteur ouvert et les E/S doivent donc être configurées en entrée avec résistance de polarisation. Une tension sur une entrée fait passer la ligne GPIO correspondante à 0 (logique inverse).

## Commande de moteur à courant continu

Cette carte (**fig. 3**) peut piloter indépendamment deux moteurs à CC sous une tension de travail de 40 V maximum et jusqu'à 5 A. La cheville ouvrière de ce circuit est constituée de deux ponts en H de MOSFET TLE5206-2. Ce circuit intégré est particulièrement facile à commander, il est aussi protégé contre la surintensité et le court-circuit. Avec une  $RDS_{(on)}$  de  $0,2 \Omega$ , la dissipation de puissance reste dans les normes. Il n'est pas très rapide, son temps de mise en conduction de la sortie avoisine  $15 \mu s$ . Aucun souci à se faire, puisque le contrôleur MLI du Swiss Pi ne peut pas monter à plus de 1,5 kHz. Aussi, retenez que la fréquence de commande sera audible dans le moteur, surtout à des valeurs faibles de MLI.

Pour empêcher les parasites du moteur de polluer la tension d'alimentation du Swiss Pi et du RPi, il y a encore une isolation galvanique entre les ponts en H et le Swiss Pi. Pour les signaux MLI, il s'agit d'un isolateur numérique du type ADUM1200ARZ qui est à deux voies et jouit d'une bande passante supérieure à celle des photocoupleurs. Il n'est malheureusement pas disponible comme composant traversant, mais un boîtier SOIC à huit contacts se soude encore facilement à la main. Pour le retour optionnel des signaux d'erreur des ponts en H, ce sont deux photocoupleurs ordinaires qui sont utilisés.

Les commandes de moteurs peuvent être couplées par deux, ce qui donne un total de huit moteurs à CC que le Swiss Pi peut commander.



## LISTE DES COMPOSANTS : COMMANDE DE MOTEUR

### Résistances :

R1, R2 = 330  $\Omega$

### Condensateurs :

C1 à C4 = 100 nF

C5 à C6 = 470 nF/63 V\*

C7 = 22  $\mu F$ /50 V\*

C8, C9 = 4 700  $\mu F$ /50 V\*

### Semi-conducteurs :

D1 à D2 = 1N4007

IC1, IC2 = ADUM1200ARZ

IC3, IC4 = TLE5206-2

IC5, IC6 = 4N25

IC7 = convertisseur CC/CC SIP3, 5 V/1 A vers 40 V (p.ex. Würth 173010542)\*

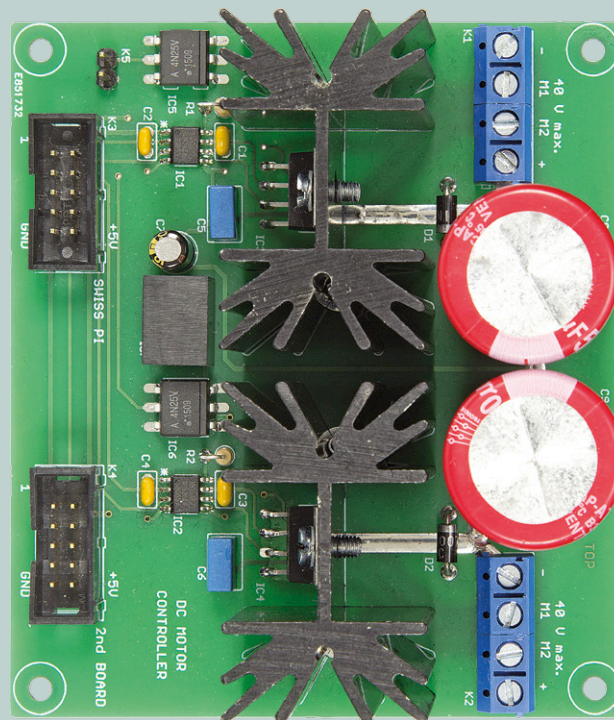
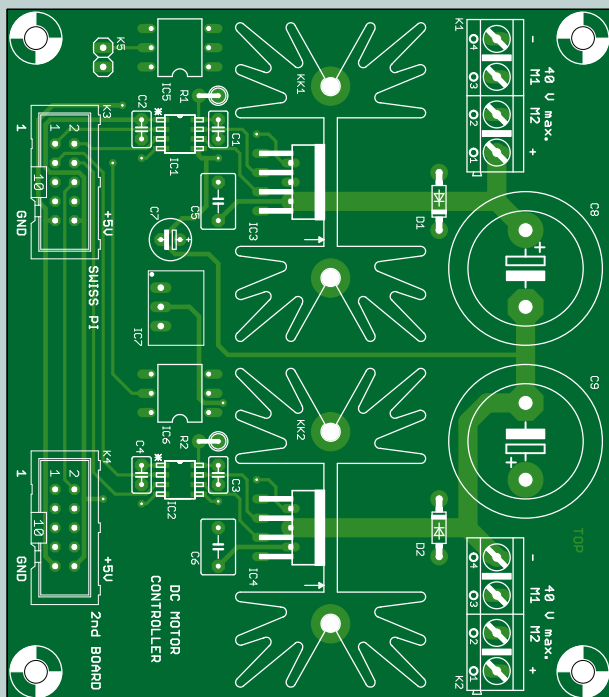
### Divers :

K1, K2 = bornier à vis, à 4 pôles, au pas de 5 mm

K3, K4 = embase à collerette à 10 broches

KK1, KK2 = radiateur, p.ex. SK 129 38,1 STS de Fisher Elektronik ( $38,1 \times 42 \times 25$  mm)

\*si la commande de moteur sert pour des tensions plus faibles, on peut adapter ces composants. IC7 ne doit pas pouvoir fournir 1 A, en fait les convertisseurs CC/CC qui donnent plus de 28 V ne sont disponibles que pour 1 A.





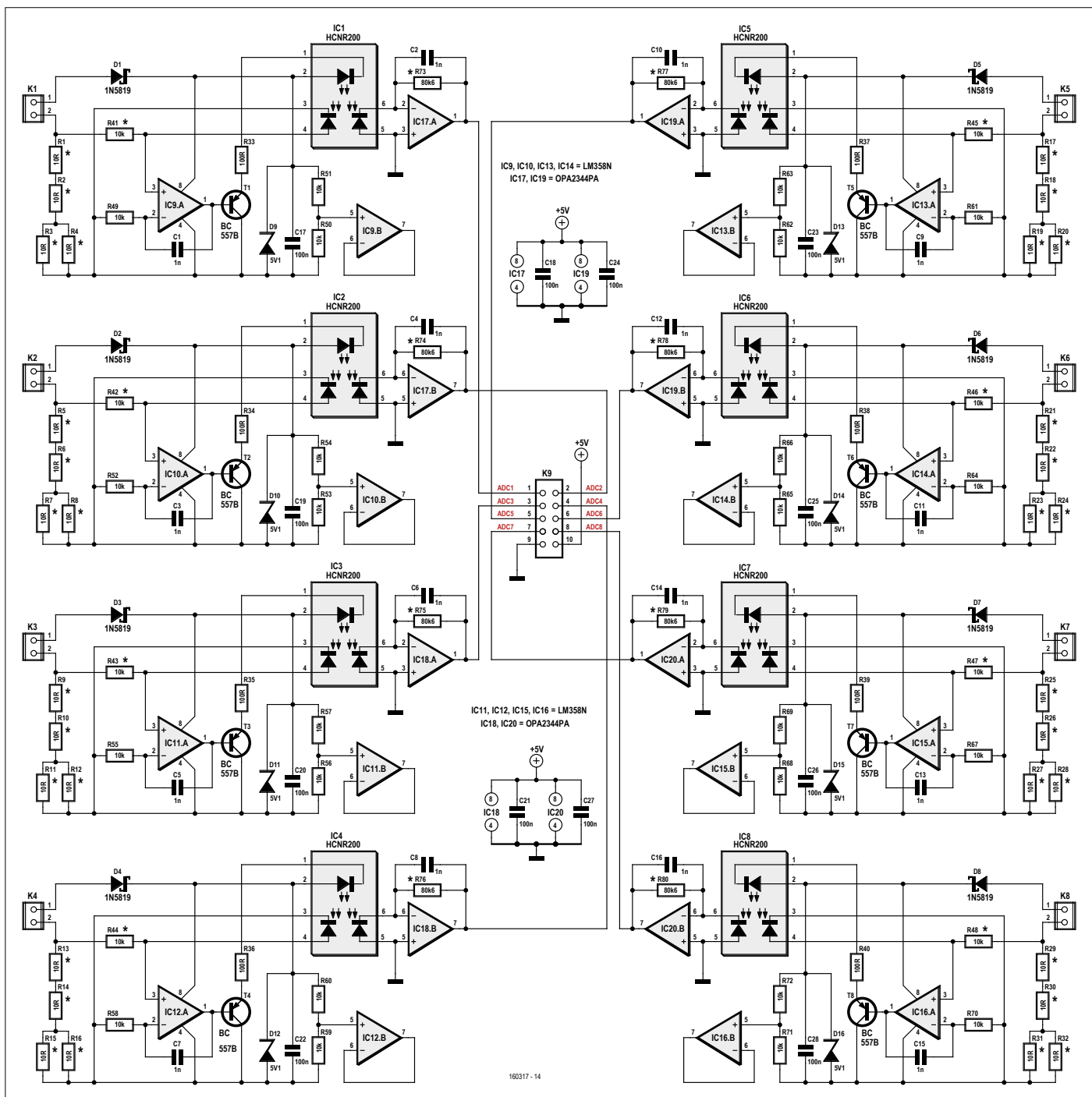


Figure 4. Le schéma complet de la carte d'interface pour boucles de courant semble énorme, mais en fait, c'est le même circuit en huit exemplaires.

Enfin, un convertisseur CC/CC fournit l'alimentation de 5 V pour la logique du côté des ponts en H. Si vous voulez utiliser une tension d'alimentation plus élevée, il importe de choisir un convertisseur CC/CC capable de supporter une tension d'entrée de 40 V.

### Boucles de courant

Dans l'industrie, malgré l'arrivée de solutions purement numériques, on se sert encore régulièrement de boucles de courant pour la transmission des mesures de

capteurs. La boucle de courant est simple et robuste et peut couvrir de grands rayons d'action. L'offre de ce genre de capteurs est donc aussi vaste.

L'interface proposée ici est construite à partir de photocoupleurs analogiques HCNR200 d'Avago (**fig. 4**). Ils sont composés de deux photodiodes appariées et d'une LED. À l'émission, l'une des photodiodes est utilisée en rétroaction de la puissance émise par la LED. Comme le courant dans la photodiode

côté récepteur est pratiquement pareil à celui dans la photodiode de l'émetteur, la caractéristique de transfert est d'une bonne linéarité.

Le circuit reproduit presque à la lettre la note d'application d'Avago, nous n'y avons apporté que quelques minimes retouches, parmi lesquelles le remplacement des amplis op de l'émetteur par des LM358 au lieu des LM158 d'origine. Non seulement beaucoup moins chers, ils sont aussi plus faciles à se procurer. La diffé-



## LISTE DES COMPOSANTS : BOUCLES DE COURANT

### Résistances :

R1 à R32 = 10  $\Omega$ \*  
 R33 à R40 = 100  $\Omega$   
 R41 à R48 = 10 k $\Omega$ \*  
 R49 à R72 = 10 k $\Omega$   
 R73 à R80 = 80,6 k $\Omega$ \*

### Condensateurs :

C1 à C16 = 1 nF  
 C17 à C28 = 100 nF

### Semi-conducteurs :

D1 à D8 = 1N5819  
 D9 à D16 = diode Zener 5V1, 400 mW  
 IC1 à IC8 = HCNR200  
 IC9 à IC16 = LM358N  
 IC17 à IC20 = OPA2344PA  
 (LM6142BIN/NOPB)

### Divers :

K1 à K8 = bornier à vis, à 2 pôles,  
 au pas de 5 mm  
 K9 = embase à collerette à 10 broches

\*1% à film métallique ou mieux

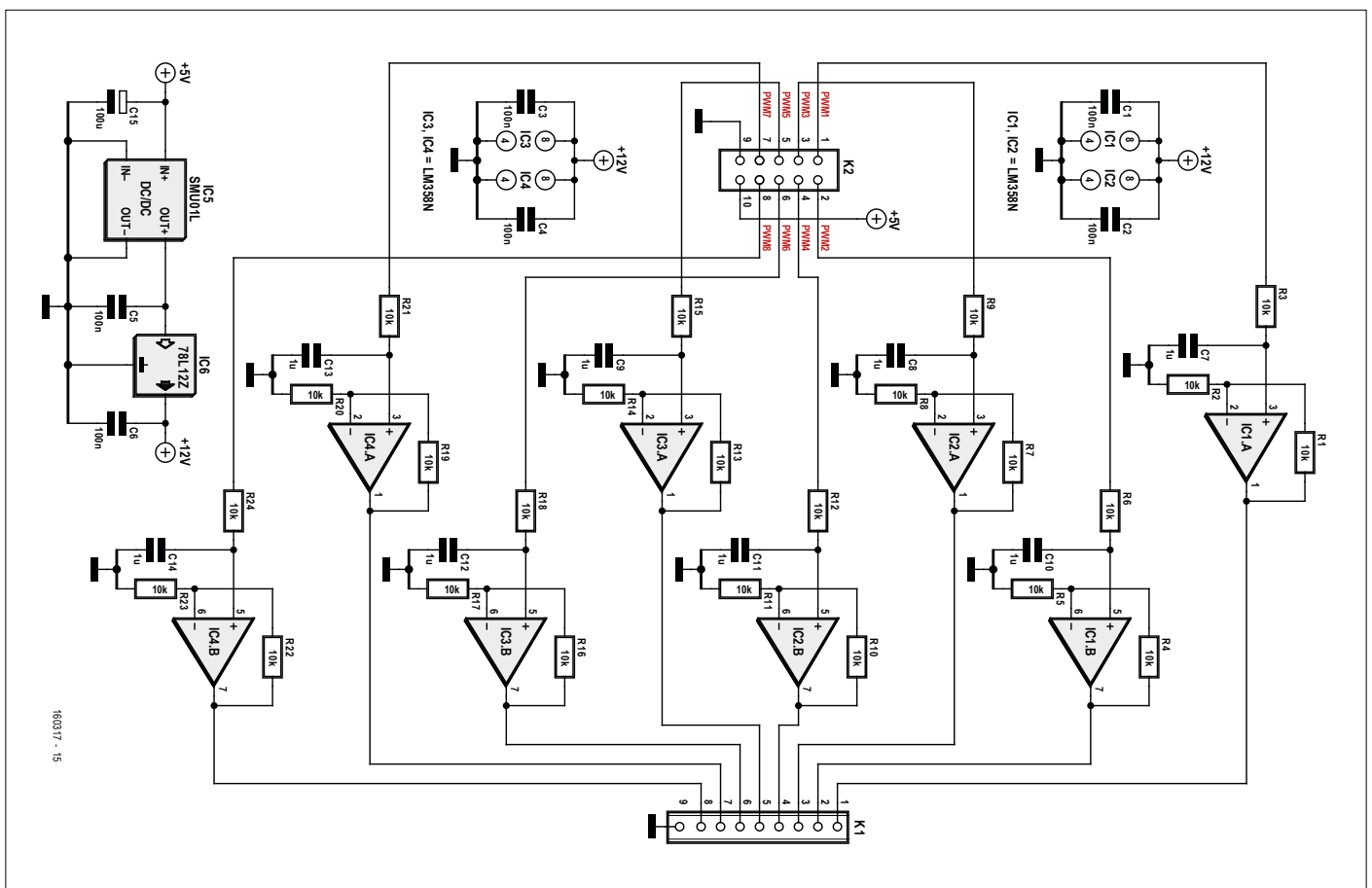
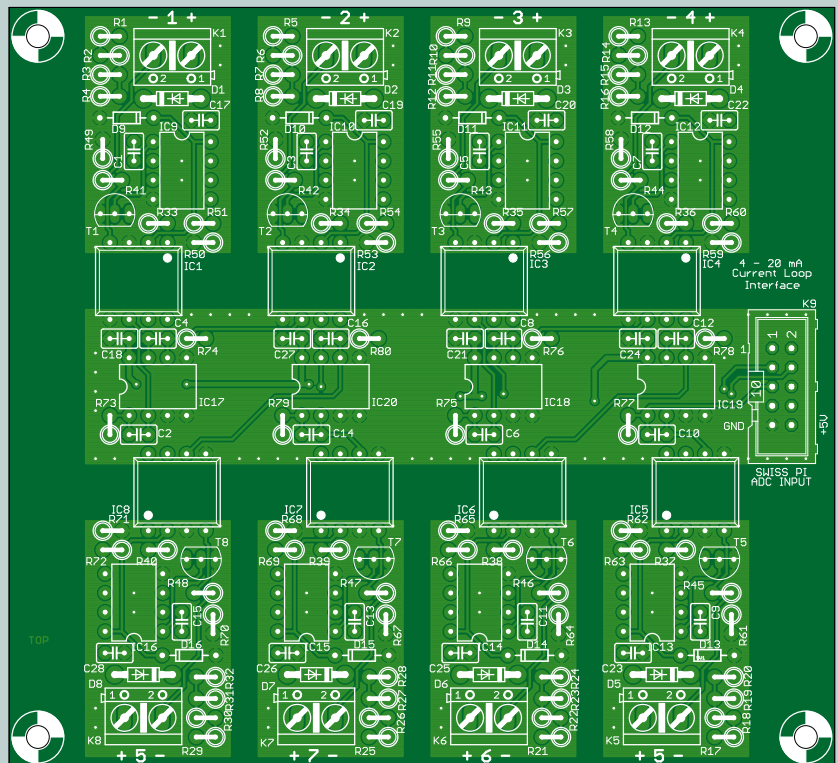
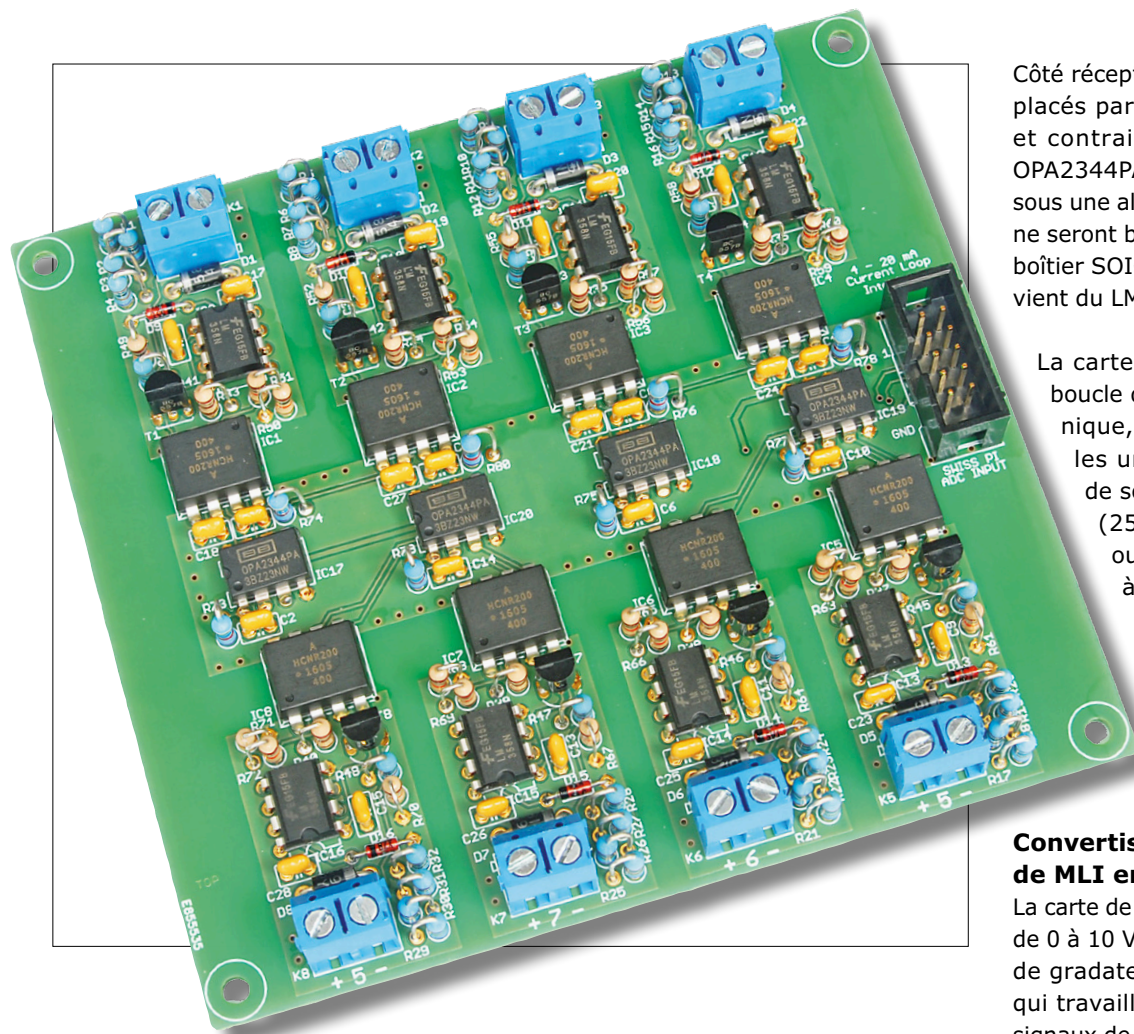


Figure 5. Le convertisseur redresse le signal MLI pour en donner une tension continue proportionnelle entre 0 et 10 V.





Côté récepteur, les LM158 ont été remplacés par des amplis op OPA2344PA et contrairement aux premiers, les OPA2344PA atteignent facilement 4 V sous une alimentation de 5 V. Hélas ! Ils ne seront bientôt plus disponibles qu'en boîtier SOIC-8, mais une autre solution vient du LM6142BIN/NOPB.

La carte porte huit interfaces pour boucle de courant à isolation galvanique, totalement indépendantes les unes des autres. La tension de sortie de l'interface vaut  $I_{in} \times (25 \times 80.600)/(10.000 + 25)$  ou  $I_{in} \times 201$ . Le domaine de 4 à 20 mA correspond donc à des tensions de sortie comprises entre 0,804 et 4,02 V, ce qui reste dans la plage acceptable du convertisseur A/N du Swiss Pi.

### Convertisseur de MLI en tension continue

La carte de conversion de MLI en tension de 0 à 10 V est destinée à la commande de gradateurs, pour LED notamment, qui travaillent généralement avec des signaux de commande dans cette plage.

Le signal MLI issu du Swiss Pi est d'abord converti en tension continue dans un filtre passe-bas, puis est tamponné et doublé par un amplificateur opérationnel de la **figure 5**.

rence principale réside dans le domaine de température spécifié. Le LM158 tolère de  $-55^{\circ}\text{C}$  à  $125^{\circ}\text{C}$ . Comme le RPi ainsi que le Swiss Pi sont réalisés avec des

composants pour applications commerciales pour une plage de température de  $0^{\circ}\text{C}$  à  $70^{\circ}\text{C}$ , il n'y a aucune raison de prendre des LM158.



## LISTE DES COMPOSANTS : MLI VERS TENSION

### Résistances :

R1 à R24 = 10 k $\Omega$

### Condensateurs :

C1 à C6 = 100 nF

C7 à C14 = 1  $\mu\text{F}$

C15 = 100  $\mu\text{F}$

### Semi-conducteurs :

IC1 à IC4 = LM358N

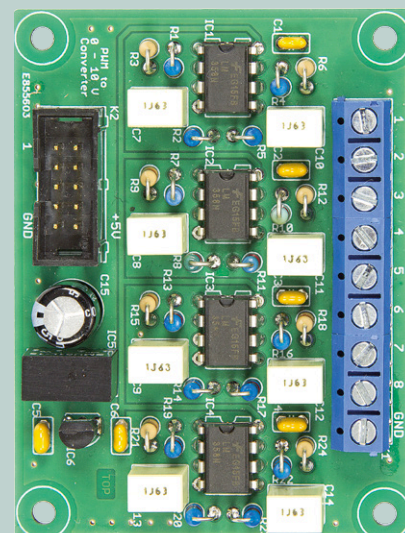
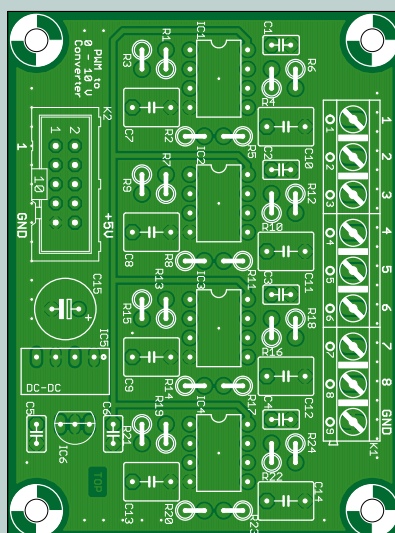
IC5 = SMU01L-15

IC6 = 78L12Z

### Divers :

K1 = bornier à vis, à 9 pôles, au pas de 5 mm

K2 = embase à collerette à 10 broches



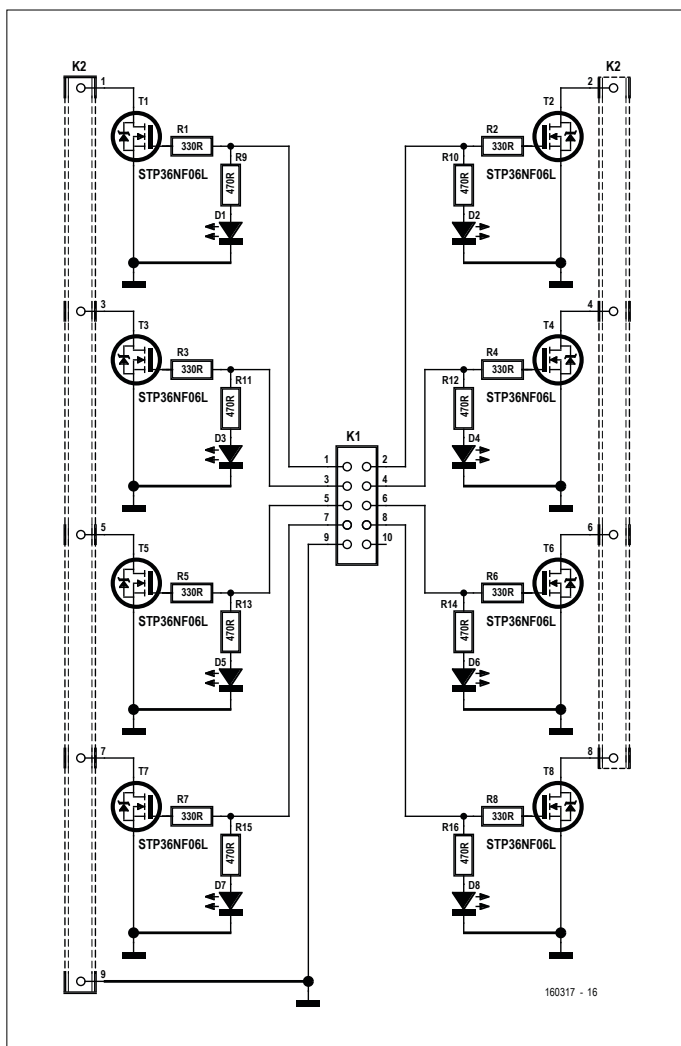


Figure 6. Pour la carte à MOSFET, le choix s'est porté sur le STP36NF06L qui supporte jusqu'à 30 A et 60 V.

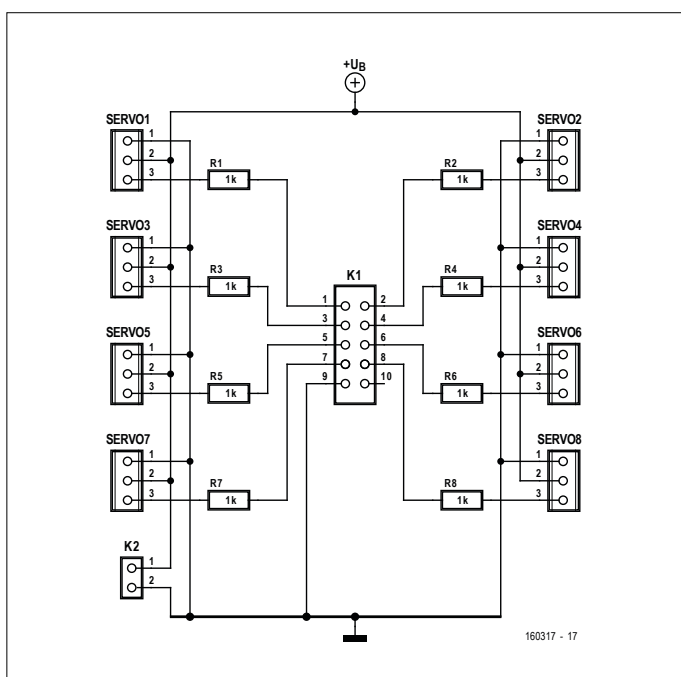


Figure 7. Avec cette simple carte, le Swiss Pi pourra piloter jusqu'à huit servomoteurs.



## LISTE DES COMPOSANTS : CARTE À MOSFET

### Résistances :

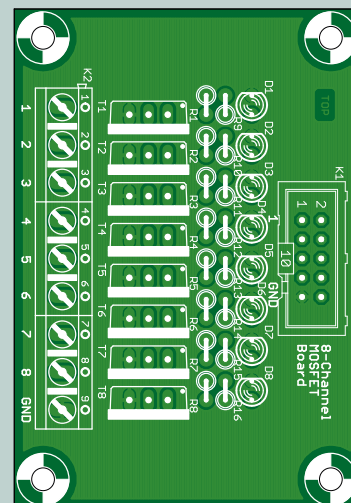
R1 à R8 = 330 Ω  
R9 à R16 = 470 Ω

### Semi-conducteurs :

D1 à D8 = LED 3 mm,  
rouge  
T1 à T8 = STP36NF06L

### Divers :

K1 = embase à collerette  
à 10 broches  
K2 = bornier à  
vis, à 9 pôles,  
au pas de 5 mm



Comme la consommation des amplis op est faible et que leurs sorties 0 à 10 V sont en principe peu chargées, c'est un simple convertisseur CC/CC qui élève la tension pour les amplis. Il transforme le 5 V du Swiss Pi en 15 V, puis un 78L12 en fait du 12 V irréprochable. On n'a donc pas besoin d'alimentation extérieure.

**Attention :** l'excursion de la tension en sortie ne va pas nécessairement de 0 à 10 V, en pratique c'est de 0 V à deux fois la tension d'alimentation de la puce MLI sur le Swiss Pi.

## Carte à MOSFET

La carte de la **figure 6** peut servir à commuter des charges en courant continu. La commande provient soit des lignes ordinaires de GPIO du Swiss Pi, soit des canaux MLI. Le pilotage de rubans de LED RVB est un bel exemple d'application.



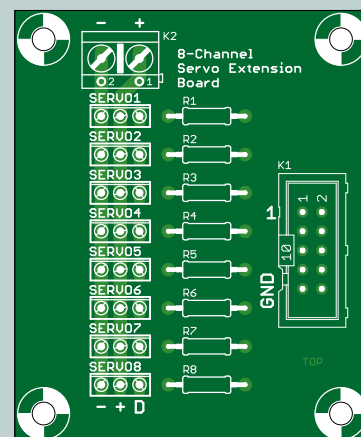
## LISTE DES COMPOSANTS : SERVOMOTEURS

### Résistances :

R1 à R8 = 1 kΩ

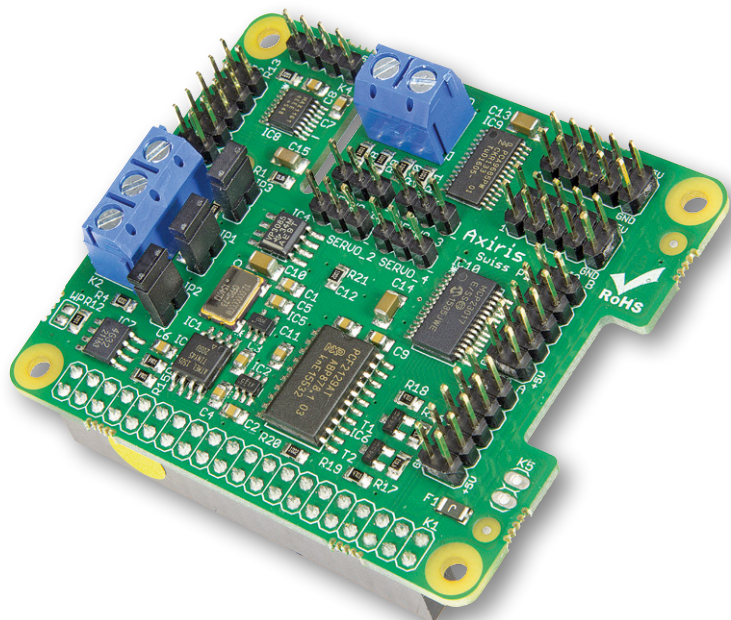
### Divers :

K1 = embase à collerette  
à 10 broches  
K2 = bornier à  
vis, à 2 pôles,  
au pas de 5 mm  
K3 à K10 =  
embase à 3 broches,  
au pas de 2,54 mm





Comme MOSFET, nous avons choisi le STP36NF06L à canal N qui convient pour des tensions jusqu'à 60 V sous 30 A ( $R_{DS(on)} = 0,032 \Omega$  pour  $V_{GS} 10 \text{ V}/15 \text{ A}$ ). C'est un MOSFET commandé par niveau logique, ce qui signifie qu'il est déjà en conduction totale avec une tension de grille de 5 V. Avec une tension  $V_{GS}$  de 5 V, sa  $R_{DS(on)}$  est de  $0,045 \Omega$ , d'où en pratique, il peut commuter jusqu'à environ 5 A sans nécessiter de refroidissement. Chaque canal est en outre doté d'un témoin à LED.



#### Carte pour d'autres servomoteurs

Cette carte-ci permet de commander plus de servomoteurs que les quatre prévus sur le Swiss Pi (**fig. 7**). Avec deux de ces cartes, c'est une ribambelle de seize servos qui s'y ajoutent. Les résistances protègent le Swiss Pi au cas où la fiche tripolaire de l'un des servos serait branchée à l'envers.

Avec les différents logiciels d'exemple parus dans le précédent article [3], vous n'avez plus que l'embarras du choix de projets qui combinent Swiss Pi et Raspberry Pi. ◀

(160317 – version française : Robert Grignard)

#### Liens

- [1] [www.elektormagazine.fr/150584](http://www.elektormagazine.fr/150584)
- [2] [www.elektormagazine.fr/160317](http://www.elektormagazine.fr/160317)
- [3] [www.elektormagazine.fr/160237](http://www.elektormagazine.fr/160237)

### DANS L'E-CHOPPE

→ 150584-91  
Carte Swiss Pi

→ 17631  
Carte Raspberry Pi 3

**mouser.fr**

Les dernières nouveautés pour  
vos conceptions les plus récentes™

La plus **vaste**  
**sélection**  
de produits les  
**plus récents.**

Plus de **4 millions**  
de produits de plus de  
**600 fabricants.**



Distributeur agréé de semi-conducteurs  
et de composants électroniques

# LoRaWAN

## architecture, protocole, sécurité et opérateurs



Dans mon précédent article [1], je vous ai présenté LoRa, une technologie radio développée par Semtech, qui repose sur la modulation CSS (*chirp spread spectrum*). LoRa permet des communications à longue portée (quelques dizaines de kilomètres en champ libre) avec des débits binaires lents. Pour utiliser LoRa, il suffit d'ajouter un circuit intégré dédié sur sa carte ou d'avoir recours à un module prêt à l'emploi qui contient l'un de ces circuits spécialisés. Aujourd'hui, je vous présente LoRaWAN : un protocole ouvert basé sur LoRa et standardisé par une association internationale, la *LoRa Alliance*. Nous verrons aussi à cette occasion les solutions mises en œuvre pour lutter contre les pirates...

**Robert Lacoste** (Chaville)

### Protocole ?

Commençons par un rappel rapide sur les protocoles de communication. Dès le début des réseaux informatiques, on s'est aperçu qu'il était indispensable de segmenter ce type de protocoles en couches indépendantes. Il y a deux raisons à ceci. Tout d'abord cela facilite la spécification, le codage et le test. Mais une telle décomposition apporte aussi une très grande souplesse : dès qu'un protocole est correctement divisé en couches, une couche peut être remplacée par une autre sans impact sur les autres. Par exemple, si vous pouvez utiliser HTTP ou FTP de façon transparente avec une connexion Ethernet ou Wi-Fi, c'est grâce à cette indépendance entre les couches.

Le modèle OSI à sept couches (**fig. 1**) a été adopté en 1984 [2] et il est toujours utilisé partout malgré son grand

âge. De manière un peu schématique, la couche la plus basse est la couche physique, elle traite de la transmission des bits bruts. Au-dessus on trouve la couche « liaison de données » qui gère la transmission point à point, et on empile les couches jusqu'à la couche « application » qui définit la sémantique des données échangées.

Avec ce modèle, LoRa n'est rien de plus qu'une couche physique (niveau 1). LoRa définit comment coder les bits en utilisant la modulation CSS et comment envoyer ces bits en utilisant l'une des bandes de fréquences disponibles. Rien ne vous empêche d'utiliser uniquement LoRa et de construire votre propre protocole, mais vous devrez alors vous occuper vous-même de pas mal de choses : comment sélectionner le bon destinataire, comment gérer les erreurs et les collisions, comment sécuriser les échanges, etc. L'autre solution est d'utiliser LoRaWAN.

C'est un protocole ouvert et standardisé, qui prend place au-dessus de LoRa. Par rapport au modèle OSI, LoRaWAN est assez rudimentaire : il se contente de gérer des liens point à point à travers une couche 2 bien définie, et inclut ce qu'il faut pour sécuriser raisonnablement les échanges. Par contre LoRaWAN ne spécifie pas la couche applicative. Cela veut dire que LoRaWAN transmet vos données d'un bout à l'autre, mais ne se préoccupe pas de leur signification.

### Architecture d'un système LoRaWAN

LoRaWAN est un protocole, mais aussi un modèle de réseau. La **figure 2** montre l'architecture globale d'un réseau LoRaWAN, qui comprend trois sous-systèmes : les objets, les passerelles et les serveurs. Un objet LoRaWAN peut être n'importe quel bidule connecté, imaginez un détecteur d'incendie par exemple. Les objets LoRaWAN envoient



toujours des messages en mode diffusé, c'est-à-dire sans préciser qui est le destinataire.

Comment cela fonctionne-t-il ? Ces messages sont reçus par une ou plusieurs passerelles LoRaWAN dans les environs, qui vont les retransmettre au serveur central du réseau. À la différence des objets, les passerelles utilisent une architecture électronique assez complexe, autour d'un circuit intégré dédié de Semtech : le SX1301. Pourquoi ? Simplement parce qu'une passerelle écoute simultanément plusieurs canaux RF et plusieurs modes de modulation, alors qu'un périphérique n'a besoin de n'en gérer qu'un. Heureusement, vous n'aurez pas besoin de concevoir vous-même une passerelle, car de nombreux fournisseurs en proposent : regardez les produits de Multitech, Kerlink ou IMST entre autres. Pour les plus aventureux, vous pouvez aussi construire une passerelle à l'aide d'un module RF SX1301 fourni par IMST connecté à un nano-ordinateur Raspberry Pi.

Enfin, LoRaWAN nécessite des serveurs centraux. Plus précisément un serveur de réseau et un ou plusieurs serveurs applicatifs (qui peuvent être hébergés, ou pas, sur la même machine). Les passerelles sont essentiellement des boîtes aux lettres, elles reçoivent des trames et les transmettent au serveur de réseau via un réseau 3G ou autre. Les serveurs font tout le reste du travail. De manière plus précise, le serveur de réseau vérifie le contenu de la trame, supprime les doublons, détermine quel serveur applicatif est intéressé par ce message et le lui transmet. Il vérifie aussi si une réponse est nécessaire et, si oui, demande à l'une des passerelles de l'envoyer à l'objet. Il gère aussi plein d'autres choses que nous verrons un peu plus bas comme l'appariement des objets et du réseau, la sécurité, la configuration, etc. Bref, vous l'aurez compris, le serveur est un gros morceau de LoRaWAN. Pour de petits réseaux, vous pourrez trouver des serveurs « légers » physiquement intégrés dans une passerelle (par ex. chez Multitech ou Kerlink), mais tout réseau un peu imposant nécessitera un vrai serveur. Des entreprises comme Actility, Orbiwise ou Lorient vendent ce type de serveur, mais vous trouverez aussi une initiative de serveur LoRaWAN *open source* sur github (voir [3]). Le serveur applicatif lui gère les données elles-mêmes. Par exemple pour un système de détection d'incendie,

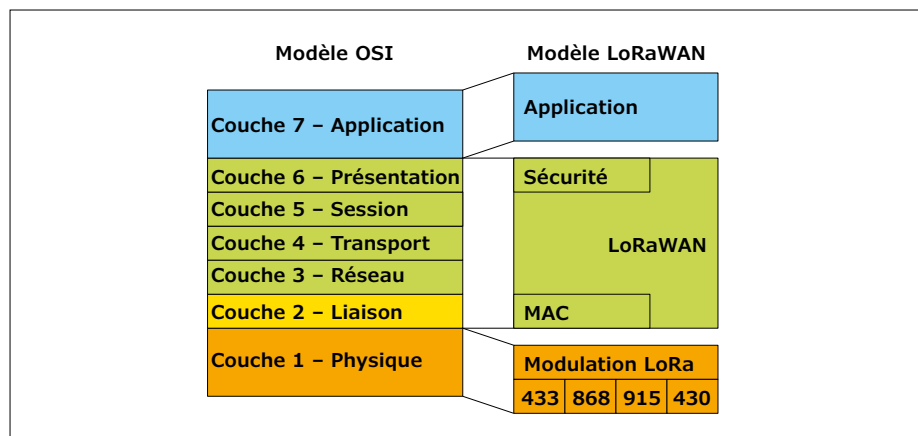


Figure 1. LoRa est un protocole de couche physique, tandis que LoRaWAN est un protocole de communication construit sur LoRa. L'application elle-même ne fait pas partie de LoRaWAN.

c'est lui qui vous enverra un SMS d'alerte et affichera un joli synoptique de votre installation sur votre tablette.

### Alliance LoRa

LoRaWAN est un standard porté par une organisation indépendante de Semtech : la *LoRa Alliance*. Jetez un coup d'œil à leur site web (voir [4]). Vous verrez qu'elle compte déjà plus de 400 membres, dont Semtech bien sûr, mais aussi des sociétés comme IBM, ARM, Orange, Bouygues Telecom, KPN, ST Microelectronics, Microchip, Cisco ou Keysight. Pas trop mal pour une organisation qui a moins de deux ans, non ? Elle gère la spécification LoRaWAN à proprement parler (voir [5]), mais fournit également des codes source pour déve-

lopper un produit LoRaWAN (voir [6]). Évitez toute confusion : LoRaWAN est un protocole ouvert, dont les spécifications comme le code source sont publics, par contre il est basé sur la modulation LoRa qui est propriété de Semtech et dont, à ce jour, les détails ne sont pas divulgués, comme expliqué dans mon précédent article. La *LoRa Alliance* a enfin mis sur pied un processus de certification pour s'assurer que les produits sont conformes à la spécification LoRaWAN et qu'ils seront doncinteropérables. Évidemment rien n'est gratuit, et adhérer à l'Alliance puis certifier son produit coûte un petit tas de milliers d'euros. Par contre, si vous utilisez un module certifié, vous n'aurez pas besoin de passer par cette étape, c'est déjà fait !

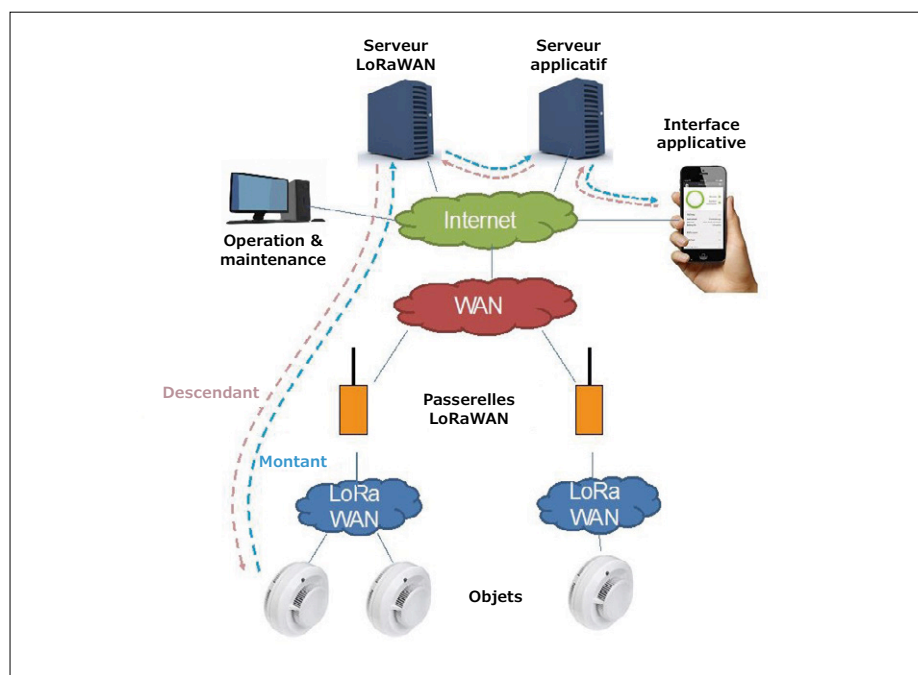


Figure 2. Une illustration de l'architecture globale d'un réseau LoRaWAN.

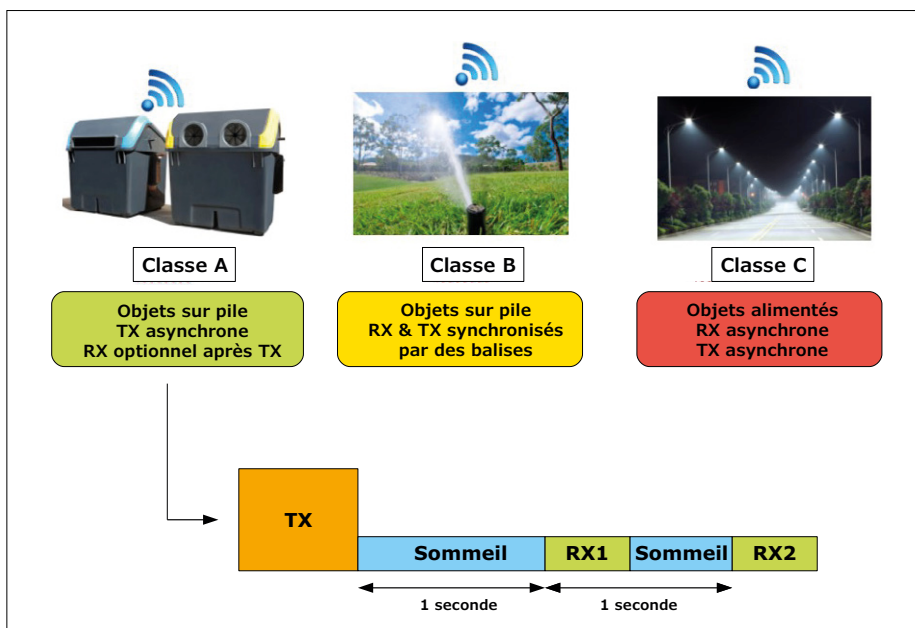


Figure 3. Trois classes de périphériques sont définies dans LoRaWAN. La majorité des applications tombent dans la classe A, où un périphérique transmet quand il veut, puis dort.

### Classes de périphériques

Comment LoRaWAN gère-t-il les échanges de messages entre un objet connecté et le réseau ? Trois classes de produits sont prévues (fig. 3). La plus simple est la classe A : le dispositif est quasiment tout le temps en veille profonde, il se réveille quand il veut et envoie un message LoRaWAN sur n'importe quel canal radio. Il se remet alors en veille et se réveille précisément une seconde plus tard pour écouter le même canal et vérifier si un message descendant lui a été envoyé. Sinon il recommence à écouter une seconde fois une seconde plus tard sur un autre canal « de secours » et se rendort. La

consommation en classe A est donc très faible, puisque la radio est quasiment tout le temps hors tension. Revers de la médaille : on ne peut causer à un dispositif de classe A que lorsqu'il a lui-même transmis quelque chose. A contrario les dispositifs de classe C sont supposés être alimentés par le secteur, ils peuvent donc recevoir en permanence des messages descendants. Enfin la classe B, qui pour l'instant n'est pas complètement supportée, permettra à un objet de se réveiller à intervalles périodiques juste pour écouter s'il y a un message pour lui. Ce mode est beaucoup plus complexe, car il nécessite une synchronisation entre le réseau et les objets pour que le réseau

puisse envoyer les messages aux bons moments. Pas simple.

### Débit adaptatif

Comme expliqué dans mon article précédent, la modulation LoRa permet une grande souplesse dans les débits de données : on peut utiliser des débits très lents pour avoir une grande portée, ou au contraire des débits rapides pour minimiser la consommation énergétique et l'encombrement des bandes radio (fig. 4). Mais comment optimiser tout ça ? LoRaWAN propose un mécanisme baptisé ADR (*Adaptive Data Rate*). Coté objet, la mise en œuvre est très simple, l'objet se contente de décoder et d'interpréter des messages spécifiques transmis par le réseau qui demandent d'utiliser tel ou tel débit. Tout se joue côté serveur de réseau. Celui-ci exécute un algorithme qui détermine si un objet donné peut être basculé, ou pas, sur un débit plus rapide (selon les niveaux de puissance et de qualité des messages reçus). Il envoie alors à l'objet les ordres correspondants. Cet algorithme en lui-même n'est pas spécifié par le standard, chaque serveur de réseau peut choisir celui qu'il veut, mais vous avez compris l'idée.

Si vous avez suivi, vous aurez peut-être deviné qu'il peut y avoir quelques soucis avec l'ADR : imaginez que la réception des messages d'un objet donné est bonne et que le serveur bascule cet objet sur un débit très rapide, ensuite un bâtiment est érigé entre l'objet et la passerelle. Le serveur pourrait ne plus recevoir les messages de cet objet, et ne pourrait donc plus lui demander de revenir à un débit plus lent pour améliorer la réception, puisqu'un objet, du moins en classe A, ne peut rien recevoir sans être lui-même écouté. Il faut donc impérativement que tout objet utilisant l'ADR demande de temps en temps un acquittement pour vérifier que tout se passe bien, et repasse de sa propre initiative sur un débit très lent.

### Et la sécurité dans tout ça ?

Les communications sans fil sont par nature faciles à pirater, mieux vaut donc prévoir un peu de sécurité. Évidemment le niveau de sécurité requis dépend de l'application ; une transaction bancaire n'a pas exactement les mêmes exigences qu'un pisteur d'animal domestique... LoRaWAN prévoit par défaut un niveau a priori adapté à 99% des applications. Entrons un peu dans les détails. Tout

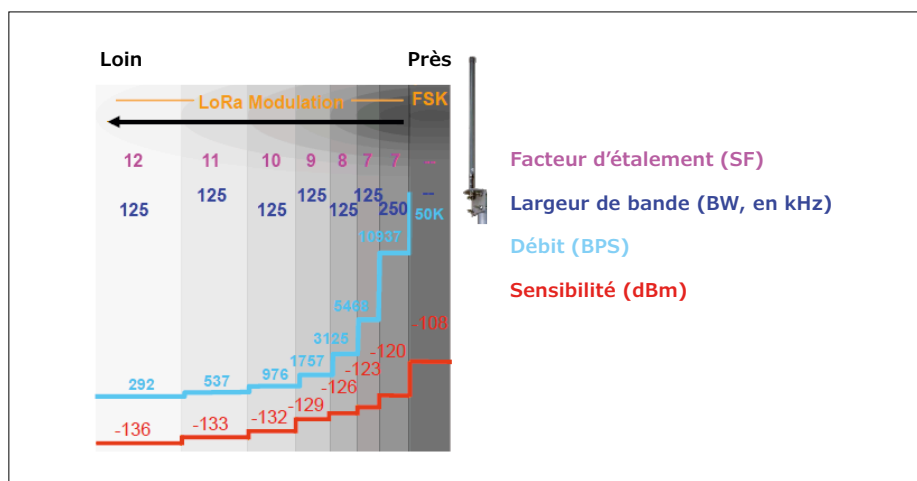


Figure 4. Avec l'ADR (*Adaptive Data Rate*), le débit est optimisé de manière dynamique par le serveur du réseau : débits plus rapides lorsqu'un objet est bien reçu par le réseau.



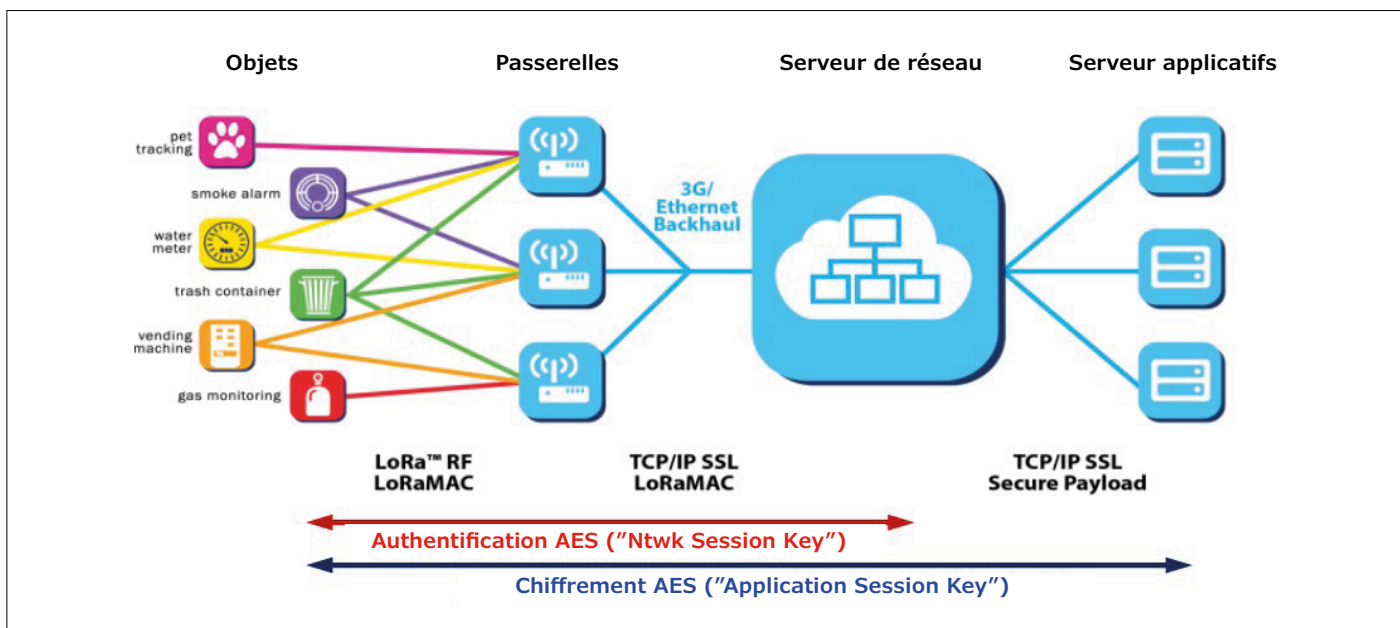


Figure 5. Deux niveaux de sécurité sont prévus dans LoRaWAN : authentification entre objets et serveur de réseau, et chiffrement de bout en bout entre objet et serveur d'applications.

d'abord, LoRaWAN a été conçu pour permettre de supporter simultanément plusieurs applications indépendantes sur le même réseau. Deux niveaux de sécurité sont donc prévus (fig. 5). Tout d'abord, le contenu du message est chiffré par son émetteur, via une clé secrète baptisée « **Application Session Key** », et connue uniquement de l'objet et du serveur applicatif avec lequel il communique. Il s'agit de ce qu'on appelle un chiffrement de bout en bout, c'est-à-dire que les dispositifs intermédiaires du réseau LoRaWAN ne peuvent déchiffrer son contenu.

Ensuite, une empreinte de sécurité est ajoutée au message, calculée grâce à une seconde clé secrète (« **Network Session Key** »), connue cette fois de l'objet et du serveur de réseau. Cette empreinte de sécurité permet au destinataire de s'assurer que le message n'a pas été modifié par un tiers (intégrité) et qu'il provient de celui qui prétend être l'expéditeur (authentification). Le processus est illustré sur la figure 6.

Pourquoi ces deux niveaux sont-ils nécessaires ? En fait la principale utilité de la chose apparaît lorsque plusieurs clients utilisent la même infrastructure LoRaWAN, par exemple celle d'un opérateur. Chaque client ne souhaite pas que ses messages puissent être lus par l'opérateur, mais ce dernier a quand même besoin de s'assurer qu'il ne fait transiter que des messages valides.

Sur le plan technique, ces deux niveaux de sécurité sont construits en utilisant des variantes du même algorithme de chiffrement, le sempiternel AES. C'est ce qu'on appelle un système de chiffrement symétrique, c'est-à-dire que les clés doivent être connues et partagées dès le départ entre les nœuds. LoRaWAN offre deux façons de réaliser cela. La pre-

mière solution est de simplement préprogrammer les deux clés et un identifiant sur chaque objet au moment de la fabrication, et de les stocker sur les serveurs en même temps.

La seconde solution, baptisée OTAA (*Over The Air Activation*) est nettement plus souple, car elle permet à un objet de changer de réseau si besoin... Ici, les

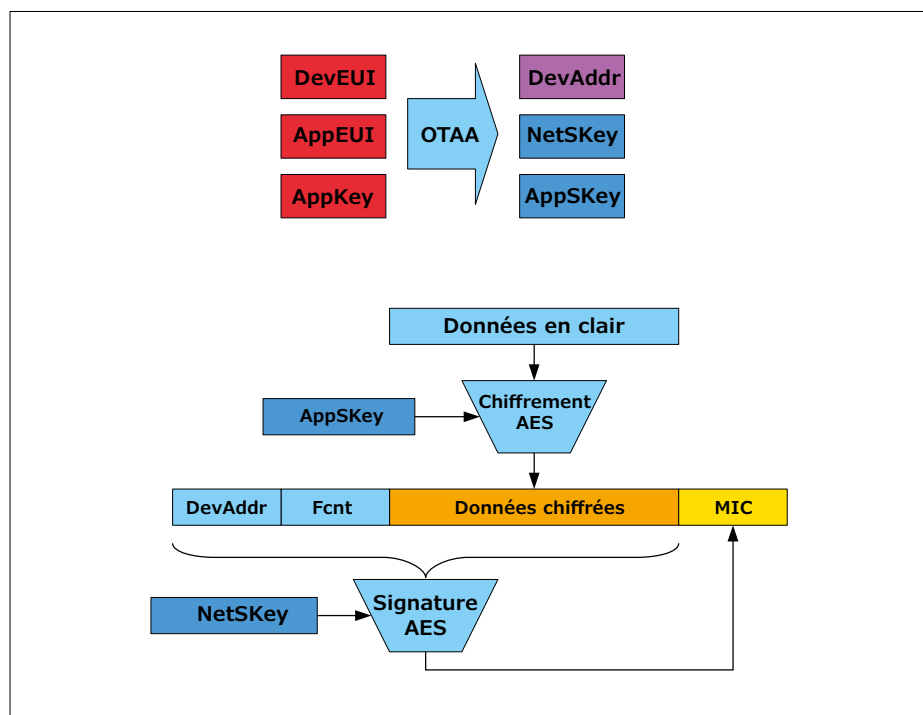


Figure 6. La sécurité nécessite deux clés de chiffrement et une adresse réseau. Ces paramètres peuvent être automatiquement transférés vers l'objet par OTAA.

objets sont préprogrammés avec un identifiant unique et une autre clé secrète qui n'est utilisée que pour l'appariement initial. Cette clé a malencontreusement été baptisée « Application Key », à ne pas confondre avec l'« Application Session Key » décrite précédemment. Lorsque l'objet est mis sous tension, il envoie un message demandant à un réseau de l'accueillir. Si un réseau est disponible et autorise l'accès à l'objet, il produit les deux autres clés dynamiquement et les lui transmet. Ces échanges sont bien sûr chiffrés et authentifiés pour que les clés ne soient jamais transmises en clair. C'est un peu plus complexe, mais ça marche !

### Opérateurs LoRaWAN

Comment utiliser LoRa et LoRaWAN dans vos projets ? Vous avez le choix entre trois solutions. Tout d'abord, vous pouvez acheter des circuits intégrés ou des modules compatibles LoRa et échanger des messages entre eux directement. Pas besoin de LoRaWAN ici, vous créez votre mini-protocole vous-même. C'est la bonne solution si vous voulez réaliser une télécommande pour la porte de votre garage avec une portée de quelques kilomètres (si tant est que cela serve à quelque chose...).

La seconde approche est de construire votre propre réseau LoRaWAN privé. Vous devrez pour cela acheter une ou plusieurs passerelles, un serveur de réseau (intégré ou non dans une passerelle), et développer ou acheter des objets qui contiennent toute la pile de protocole LoRaWAN. Ces derniers bénéficieront alors de toutes les fonctions de LoRaWAN et pourront échanger des données avec le serveur d'applications de votre choix. Plus complexe, mais nettement plus souple, cette solution est adaptée par ex. à un système de communication privé dans une usine ou un hôpital.

La troisième solution, qui a vraiment le

### Liens et références

- [1] *Hors Circuits : LoRa, échange débit contre portée*, Elektor, 01/2017 : [www.elektormagazine.fr/160236](http://www.elektormagazine.fr/160236)
- [2] Modèle OSI : [fr.wikipedia.org/wiki/Mod%C3%A8le\\_OSI](http://fr.wikipedia.org/wiki/Mod%C3%A8le_OSI)
- [3] Logiciel de serveur LoRaWAN *open source* : [github.com/brocaar/loraserver](https://github.com/brocaar/loraserver)
- [4] LoRa Alliance : [www.lora-alliance.org](http://www.lora-alliance.org)
- [5] Spécification LoRaWAN version 1.02 : [www.lora-alliance.org/Contact/Request-Specification-Form](http://www.lora-alliance.org/Contact/Request-Specification-Form)
- [6] Code source LoRaWAN de référence : [github.com/Lora-net/LoRaMac-node/wiki/LoRaMAC-node-Wiki](https://github.com/Lora-net/LoRaMac-node/wiki/LoRaMAC-node-Wiki)

vent en poupe, consiste à s'appuyer sur un opérateur LoRaWAN. C'est une société, en général un opérateur de télécom traditionnel, qui a déployé un réseau LoRaWAN de grande ampleur et vous propose de transmettre vos messages. Il suffit alors de développer les objets compatibles LoRaWAN et si besoin un serveur applicatif associé, et bien sûr de payer un abonnement à l'opérateur. La bonne nouvelle est que des dizaines d'opérateurs déploient actuellement des réseaux LoRaWAN dans le monde : Bouygues Telecom via sa filiale Objenious et Orange en France, Proximus en Belgique, KPN en Hollande, Senet aux USA, Tata en Inde et plein d'autres ! Chacun d'eux installe assez de passerelles pour couvrir l'ensemble du territoire même à l'intérieur des bâtiments.

Très bien, mais combien ça coûte ? Les tarifs ne sont malheureusement pas publics, mais restent modiques (quelques euros par objet et par an au maximum). Par contre il reste difficile à un amateur d'acheter un abonnement, car les opérateurs privilégient à ce stade les clients avec des projets de plusieurs milliers d'objets. Toutefois si vous avez un projet à but commercial, ils seront très heureux de vous offrir quelques abonnements de test, et même les indispensables kits d'évaluation !

### Conclusion

Est-ce que LoRaWAN sera la technologie phare de l'Internet des Objets pour les vingt prochaines années ? Personne ne peut l'affirmer, car tout bouge vite dans ce secteur. D'autres technologies comme NB-IoT pointent leur nez et des concurrents comme Sigfox défendent fermement leurs solutions. Les bandes 868 MHz pourraient aussi être rapidement surchargées s'il y a autant d'objets connectés qu'on nous le prédit.

Ceci dit, il est indéniable que LoRaWAN présente des avantages technologiques et surtout est très dynamique en ce moment. Des réseaux nationaux de plusieurs milliers de passerelles sont installés par de grands opérateurs, beaucoup de jeunes pousses développent des objets compatibles, et les réseaux privés LoRaWAN fleurissent pour des applications industrielles. En outre des évolutions plus qu'intéressantes de LoRaWAN sont promises pour 2017, dont la géolocalisation des objets par triangulation et l'itinérance internationale.

Alors restez à l'écoute, et n'hésitez pas à expérimenter ! ◀

(160308)

Cet article a été publié dans la revue  
Circuit Cellar (n°315, octobre 2016).







## 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).

## mars 2017

### ◆ Conférence Automobile Connectée

02/03/2017 – Paris  
[cac.journalauto.com](http://cac.journalauto.com)

### ◆ Payforum

07 au 08/03 – Paris  
[www.payforum.fr](http://www.payforum.fr)

### ◆ Paris Space Week 2017

08 au 09/03 – Paris  
[www.paris-space-week.com/FR](http://www.paris-space-week.com/FR)

### ◆ Salon du livre

14 au 27/03 – Paris  
[www.salondulivreparis.com](http://www.salondulivreparis.com)

### ◆ Rencontres de l'électronique imprimée

15/03 – Paris  
[www.rencontreselectroniqueimprimee.com](http://www.rencontreselectroniqueimprimee.com)

### ◆ Salon de l'Analyse Industrielle

15 au 16/03 – Paris  
[www.analyse-industrielle.fr](http://www.analyse-industrielle.fr)

### ◆ ENOVA Strasbourg

Le magazine sera disponible  
gratuitement sur ce salon.

15 au 16/03 – Strasbourg  
[www.enova-event.com](http://www.enova-event.com)



### ◆ Semaine de la presse et des médias dans l'école

Elektor participe  
à cet événement.

20 au 25/03 –  
partout en France  
[www.clemi.fr](http://www.clemi.fr)



### ◆ Semaine de l'Industrie

20 au 26/03 – dans toute  
la France  
[www.entreprises.gouv.fr/semaine-industrie](http://www.entreprises.gouv.fr/semaine-industrie)

### ◆ E-LEARNING EXPO (enseignement sur l'internet)

21 au 23/03 – Paris  
[www.e-learning-expo.com](http://www.e-learning-expo.com)

### ◆ I-EXPO (information numérique)

21 au 23/03 – Paris  
[www.i-expo.net](http://www.i-expo.net)

### ◆ Rencontres Interrégionales de Sous-Traitance du Sud-Est

21 au 23/03 – Valence  
[www.rist.org](http://www.rist.org)

### ◆ Industry Business Meetings

22/03 – Roanne  
[industrybusinessmeetings.com](http://industrybusinessmeetings.com)

### ◆ Cloud Computing World

22 au 23/03 – Paris  
[www.cloudcomputing-world.com](http://www.cloudcomputing-world.com)

### ◆ IoT World

22 au 23/03 – Paris  
[www.iot-world.fr](http://www.iot-world.fr)

### ◆ MtoM & objets connectés - embedded systems

22 au 23/03 – Paris  
[www.salons-solutions-electroniques.com](http://www.salons-solutions-electroniques.com)

### ◆ Solutions Datacenter Management

22 au 23/03 – Paris  
[www.datacenter-expo.com](http://www.datacenter-expo.com)

### ◆ FORUMESURE

(instrumentation, accréditation, mesure et contrôle qualité)  
27 au 29/03 – Nantes  
[www.forumasure.com](http://www.forumasure.com)

### ◆ Documation & Data Intelligence Forum

29 au 30/03 – Paris  
[www.documation.fr](http://www.documation.fr)

## avril 2017

### ◆ RICV (rencontres internationales de cerfs-volants)

01 au 09/04 – Berck-sur-Mer  
[www.cerf-volant-berck.com](http://www.cerf-volant-berck.com)

### ◆ Industrie Lyon 2017

04 au 07/04 – Lyon  
[www.industrie-expo.com](http://www.industrie-expo.com)

### ◆ SIdO (Internet des Objets)

05 au 06/04 – Lyon  
[www.sido-event.com](http://www.sido-event.com)

### ◆ SEPÉM INDUSTRIES SUD-EST

25 au 27/04 – Avignon  
[avignon.sepem-industries.com](http://avignon.sepem-industries.com)

### ◆ EUROPEAN GRAPHENE FORUM

26 au 28/04 – Paris  
[www.setcor.org/conferences/EGF-2017](http://www.setcor.org/conferences/EGF-2017)

### ◆ Concours Lépine

27/04 au 08/05 – Paris  
[www.concours-lepine.com](http://www.concours-lepine.com)

# fabriquer son thérémin

## des JFET à la place des tubes

Trouvaille presque centenaire comme les ondes Martenot, le thérémin reste source d'inspiration. C'est un instrument de musique sans contact avec l'instrumentiste, inventé en 1920 par Lev Termen, plus connu sous le nom de Léon Theremin. Vous l'avez sûrement déjà entendu. C'était la belle époque des tubes, aujourd'hui, nous avons les JFET, l'occasion de repenser la conception d'un thérémin à réaliser soi-même.

Burkhard Kainka (Allemagne)



Le principe est simple, il provient de la superposition de deux signaux HF qui engendrent une tonalité de battement. L'un des oscillateurs HF est accordé par l'effet capacitif de la main et modifie la hauteur du son (*pitch*), l'autre oscillateur est à fréquence fixe. Les deux aboutissent à un démodulateur, similaire à celui de la détection sur une radio AM. C'est pareil à la réception d'un signal morse sans modulation. On peut déjà faire un premier essai à peu de frais avec un récepteur existant.

### Première expérience avec le *shield* SDR d'Elektor

Pour un premier essai, il suffit de construire un oscillateur simple que l'on peut influencer suffisamment à la main par couplage capacitif. J'ai utilisé le *shield* SDR d'Elektor sur une carte Arduino Uno avec le logiciel pour le VFO en solo et l'afficheur LCD du *shield* d'extension Elektor. Le même montage que celui proposé pour la réception des ondes courtes [1]. Le *shield* est utilisé ici en mélangeur direct et branché sur un haut-parleur actif. On remplace ainsi le second oscillateur et le mélangeur du

thérémin d'origine. On peut facilement régler avec précision la fréquence superposée (**fig. 1**) et n'utiliser au début que l'oscillateur *pitch*. Sous cette forme, on a déjà un premier thérémin, mais sans réglage de volume. Une application intéressante pour le *shield* SDR.

L'oscillateur à FET de la **figure 2** tra-

vaille à 420 kHz et varie l'accord jusqu'à 5 kHz quand on approche la main. L'antenne SDR est un morceau de fil posé près de l'oscillateur. Le signal est tellement intense qu'il ne faut qu'un faible couplage. Cette fois-ci, sur la sortie stéréo, on ne branche pas un PC, mais directement un haut-parleur actif. Le *shield* SDR, donc sans logiciel, comme simple

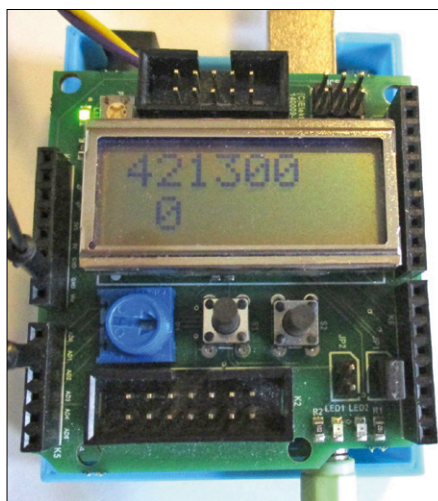


Figure 1. Le récepteur accordé sur 421,3 kHz.

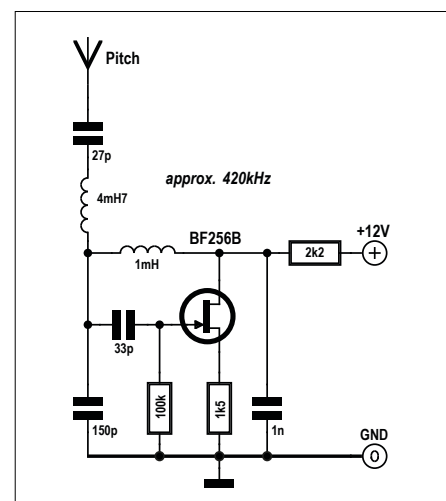


Figure 2. Un oscillateur simple pour la hauteur de note (*pitch*)



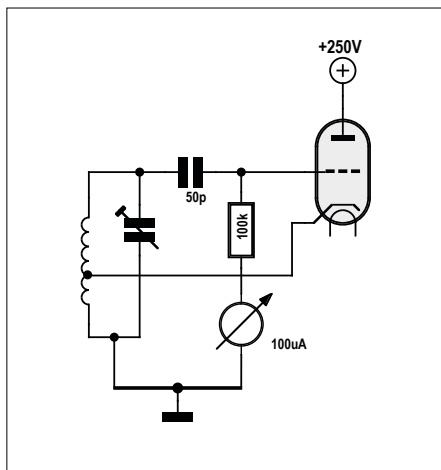


Figure 3. Un ondemètre (*grid-dip-meter*) à triode.

mélangeur direct.

Avec le logiciel indépendant pour le *shield* SDR et celui d'extension, on obtient un accord juste et facile. On cherche d'abord avec les boutons du *shield* d'extension le signal de l'oscillateur pour atteindre le battement nul. En approchant la main, la fréquence de l'oscillateur de *pitch* diminue et la note du battement monte. Alors, ce n'est plus qu'une question de maestria pour arriver à jouer sa première mélodie.

Dans le conducteur d'antenne, il y a une bobine d'allongement de 4,7  $\mu$ H. Elle sert à augmenter suffisamment la variation de fréquence induite par la très faible capacité de la main, environ un picofarad. Pour atteindre la résonance à 420 kHz, il faut ajouter à la bobine de 1 mH une capacité de 143 pF. Avec 144 pF, on arrive 1,5 kHz plus bas. La bobine d'allongement fait bien davantage de ce maigre picofarad. On peut s'imaginer qu'une antenne beaucoup plus longue se trouve près d'une main nettement plus grande. Le truc de la bobine d'allongement, Theremin l'avait déjà utilisé et il a été repris maintes fois depuis lors.

Sur un plan plus scientifique, un circuit résonant série, à proximité de sa résonance, est une petite réactance qui correspond à une grosse capacité en dessous de sa résonance. Avec la bobine employée et une antenne fouet ordinaire, la résonance propre se situe près de 600 kHz. C'est assez proche de la fréquence de travail de 420 kHz et conduit à une augmentation conséquente de la capacité de la main. La sensibilité aussi augmente bien avec la capacité de la

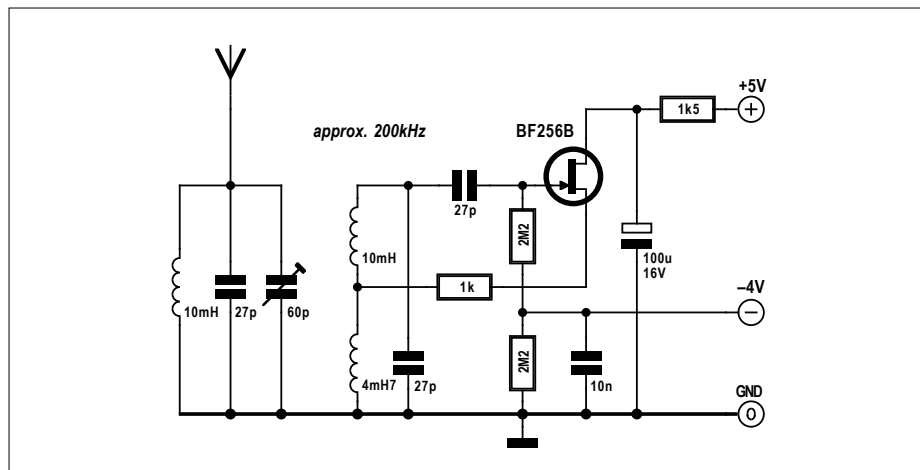


Figure 4. L'ondemètre en version à JFET.

main, produisant une plage de jeu intéressante de 30 cm environ. Grâce à la faible fréquence de l'oscillateur, la stabilité de fréquence est largement suffisante et avec le *shield* SDR, on a un couplage plus faible et un son plus agréable.

### Et le réglage de volume ?

À ce stade, le thérémin manque encore d'un réglage de l'intensité du son. L'original avait pour cela une seconde antenne, que l'on influençait de la main gauche. Quand on s'en approche, le son s'affaiblit et peut même complètement disparaître. C'est indispensable pour bien jouer de l'instrument.

Ma première idée a été d'appliquer l'un des deux signaux d'oscillateur à un circuit résonant à accorder avec l'antenne de volume. Raté ! On influence l'oscillateur de *pitch* en même temps. On en arrive à un effet mutuel entre volume et hauteur de note.

Retour au vieux thérémin. Il se sert d'un troisième oscillateur qu'il utilise à la manière d'un dipmètre (*grid-dip*) ou ondemètre (**fig. 3**) pour mesurer les fréquences de résonance. Un circuit résonant à proximité de la bobine de l'oscillateur en prélève de l'énergie, de sorte que l'amplitude d'oscillation diminue. On s'en aperçoit par la variation du courant de grille (*grid*). À l'endroit de la résonance, on observe une forte baisse (*dip*) du courant de grille.

Le thérémin utilisait la tension négative de grille de l'oscillateur pour commander le gain d'un tube régulateur dans le circuit BF. On élimine ainsi les soucis de

stabilité, puisque la fréquence de *dip* et la BF sont très éloignées de celle de l'oscillateur de superposition. Dans un circuit à tubes, le circuit en oscillation libre pouvait atteindre une amplitude de plus de 10 V. Sur une résistance de retour de grille d'environ 1 M $\Omega$ , on peut s'attendre à environ -10 V. Quand on place un deuxième circuit résonant en couplage lâche et assez proche de la résonance, l'amplitude, et par conséquent la tension négative de grille, peuvent diminuer d'un volt. Ce serait la condition de base pour le plein volume. Si maintenant on désaccorde le circuit passif ou on l'atténue, la tension négative de grille remonte fort. Avec -10 V, un tube BF est à saturation.

La première tentative avec un ondemètre à FET a eu lieu avec de relativement grandes inductances (**fig. 4**). En réalité, le circuit atteint une forte amplitude d'oscillation et une tension de sortie de -4 V. Le creux de résonance est bien là, mais trop faible. En tout, la tension ne bouge que de 0,5 V. De plus, on observe un effet d'entraînement, qui se produit aussi avec un dipmètre normal. L'amplitude change brusquement sur l'un des flancs du *dip*.

### Couplage de Tesla

Nous avons vu dans le schéma du vieux thérémin un genre de bobine d'allongement dans le conducteur d'antenne. À y regarder de plus près, elle fonctionne comme sur un transformateur de Tesla. La bobine d'allongement a une fréquence propre de résonance et possède un rapport L/C extrêmement grand et en outre, un fort couplage avec l'oscillateur. Alors, j'ai accordé l'oscillateur (**fig. 5**) sur la

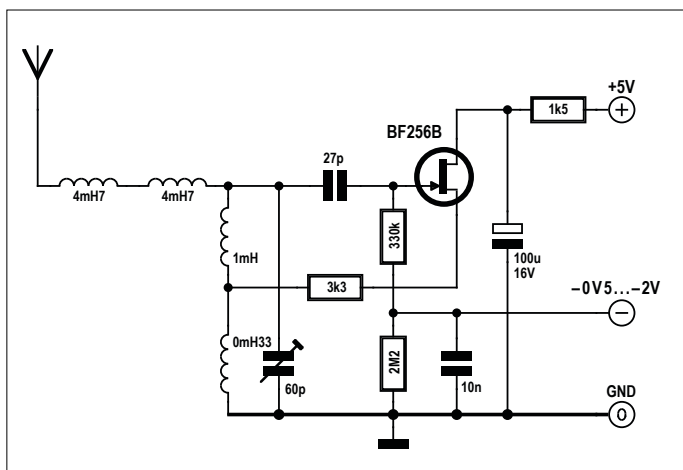


Figure 5. L'oscillateur avec bobine d'antenne.

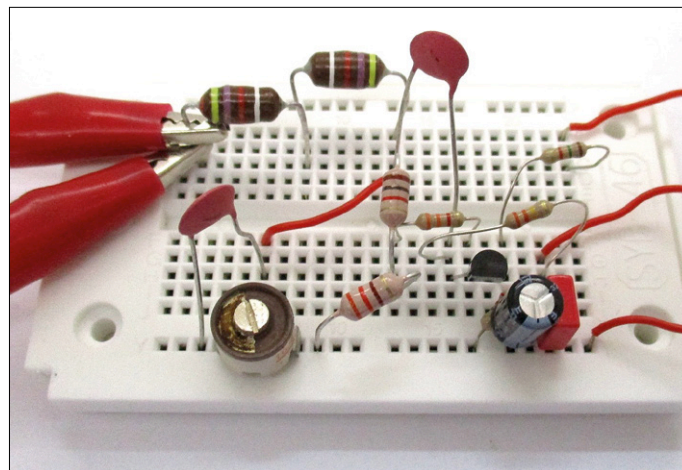


Figure 6. Construction expérimentale de l'oscillateur pour régler le volume.

bobine de Tesla, dont la fréquence de résonance varie fort avec les plus petits changements de capacité.

En cas de résonance, on obtient à l'antenne un surplus multiple de tension HF. Le circuit secondaire pompe alors pas mal d'énergie aux dépens de l'oscillateur, ce qui fait nettement baisser son amplitude. Le creux de tension de grille atteint à peu près  $-0,5\text{ V}$ . Avec un plus grand désaccord ou de l'amortissement, lors d'un contact direct avec l'antenne, cet effet est éliminé et la tension de grille du JFET baisse jusqu'à  $2\text{ V}$ . La résonance se règle bien, il n'y a plus d'effet d'en-

traînement et la tension de réglage est suffisante. Pareil oscillateur s'implante bien sur une plaque d'expérimentation (**fig. 6**). Comme antenne, une boucle faite d'un fil d'essai à pinces crocodile suffit.

### Circuit simplifié d'oscillateur

Avec le circuit d'oscillateur modifié de la **figure 7**, pas besoin d'une inductance compliquée pour le circuit oscillant. Au contraire, le circuit oscillant opère ici avec un diviseur de tension capacitif. On peut adapter le rapport d'impédance des condensateurs du côté grille et drain. Cette fois-ci, l'oscillateur travaille autour de  $1\text{ MHz}$ . On a ici la résonance propre du couple de bobines d'allongement. Comme dans l'expérience précédente, l'amplitude de l'oscillateur s'élève quand on modifie ou affaiblit le circuit du haut. La tension de grille tombe jusqu'à  $-2\text{ V}$ .

### Le thérémín complet

Le circuit complet de la **figure 8** regroupe maintenant le deuxième oscillateur, le mélangeur et un réglage de volume par antenne. On a stabilisé le deuxième oscillateur par résonateur céramique sur  $470\text{ kHz}$ . Il faut pouvoir régler l'oscillateur variable sur la même fréquence, d'où la présence d'un condensateur variable.

Les deux oscillateurs non asservis sont maintenant dotés de bobines de  $1\text{ mH}$ . Pour atteindre  $470\text{ kHz}$ , on y ajoute  $115\text{ pF}$ . La capacité mesurée de l'antenne est de  $5\text{ pF}$ . En approchant la main à quelques centimètres, la capacité monte à  $6\text{ pF}$ . Un simple calcul montre que la variation d'un seul picofarad cause une

variation de fréquence de  $2\text{ kHz}$ . On n'aurait alors qu'une plage de jeu très étroite avec l'antenne. La bobine d'allongement de  $4,7\text{ mH}$  fait en sorte que le changement de capacité est fortement augmenté à l'enroulement inférieur.

L'oscillateur *dip* travaille aussi avec  $1\text{ mH}$  à présent. Pour la bobine d'allongement de l'antenne de réglage du volume, on a trouvé une résonance propre à  $600\text{ kHz}$ . On doit donc pouvoir accorder l'oscillateur non asservi aussi sur cette fréquence, ce qui demande en tout  $70\text{ pF}$ . À  $600\text{ kHz}$ , le circuit d'antenne prend tellement d'énergie de l'oscillateur que la tension de commande se réduit à  $-2,5\text{ V}$ . Influencer le circuit d'antenne en approchant la main fait monter l'amplitude de l'oscillateur au point d'amener la tension de commande à  $-3,5\text{ V}$ .

Avec une bobine d'antenne de  $4,7\text{ mH}$ , il faut  $26\text{ pF}$  pour les mettre en résonance sur  $600\text{ kHz}$ . La plus grande part de cette capacité retourne à la capacité propre du bobinage, seule une petite quotité va à l'antenne. Une modification d'un picofarad en approchant la main déplace la fréquence de résonance de  $12\text{ kHz}$ . Cela suffit aussi pour le *dip* souhaité.

L'amplificateur de sortie qui règle le gain par la tension de grille a été conçu en dernier lieu. Avec  $-3,5\text{ V}$ , le FET est totalement bloqué, le signal de sortie est ramené à zéro. Pour  $-2,5\text{ V}$  sur la grille, on atteint déjà une grande amplification. L'ampli commandé est en montage à grille commune : le signal lui est appliqué sur la source et récupéré sur le drain. Il faut que l'oscillateur *dip* soit

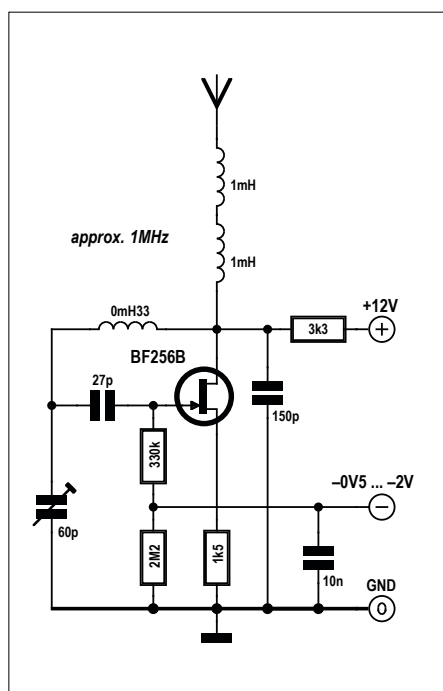


Figure 7. Un autre circuit d'oscillateur.



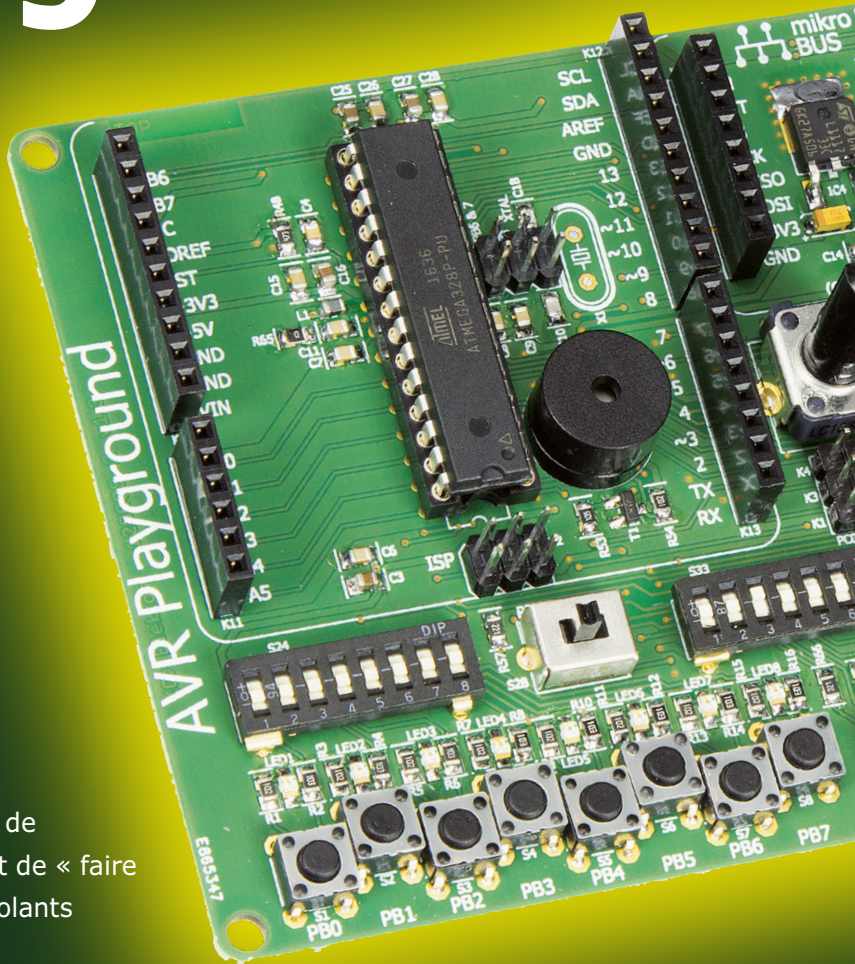


# AVR Playground

ou comment  
faire mieux  
« façon Arduino »

Clemens Valens (labo d'Elektor)

Cet article présente l'*AVR Playground*, une carte hybride entre l'Arduino Uno et une carte classique de développement pour microcontrôleur. L'objectif est de « faire de l'Arduino » sans mauvaises connexions ni fils volants (même si la carte ne l'empêche pas).



Avant l'émergence d'Arduino, les cartes de développement pour microcontrôleur embarquaient des périphériques (boutons-poussoirs, LED, afficheur), un ou plusieurs potentiomètres pour les signaux analogiques, des connecteurs d'extension, etc. et, bien sûr, une alimentation convenable. Le but de ces cartes était de faciliter la prise en main d'un  $\mu C$  sans avoir à souder ni ajouter quoi que ce soit. La carte AVR Playground – une référence explicite au forum Arduino baptisé Arduino Playground – a été conçue dans cette optique et étendue avec des caractéristiques propres à Arduino que nous apprécions.

En bref, la carte peut être divisée en quatre zones horizontales (**fig. 1**), avec du haut en bas et de gauche à droite les fonctions suivantes :

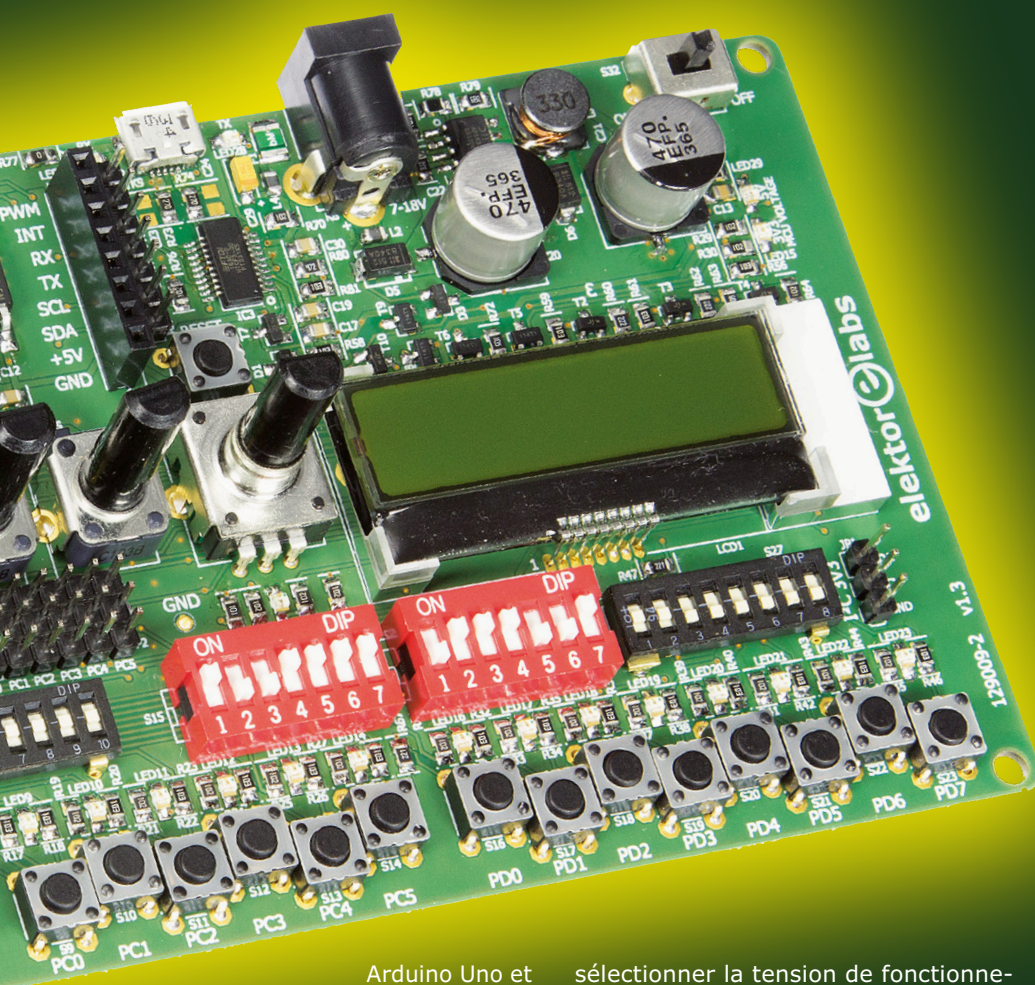
1. connecteurs d'extension, convertisseurs USB-série et alimentation
2. périphériques d'interfaçage avec l'utilisateur (IHM) : buzzer, boutons rotatifs et afficheur
3. commutateurs de configuration
4. boutons-poussoirs et LED

Pour voir le schéma, le circuit imprimé ou la liste des composants de l'AVR Playground, allez-en [1].

## Connecteurs d'extension

Le coin supérieur gauche de la carte accueille le  $\mu C$  et les connecteurs d'extension compatibles Arduino Uno. Cette partie de la carte se comporte exactement comme une Arduino Uno. Juste à côté, il y a un connecteur d'extension mikroBus. Cette extension standard développée par la société serbe MikroElektronika, connaît un franc succès grâce à la disponibilité de centaines de cartes d'extension bon marché : du récepteur GPS au capteur d'humidité en passant par des matrices de LED et des modems téléphoniques. Avec ces connecteurs d'extension





Arduino Uno et mikroBus, l'utilisateur de l'AVR Playground a accès à un immense catalogue de cartes d'extension.

### Boutons-poussoirs, LED & commutateurs DIP

La partie inférieure de la carte, la zone 4, est occupée par des boutons-poussoirs et des LED reliés à chaque port utilisable du  $\mu C$ . Les boutons-poussoirs permettent d'appliquer un niveau logique sur un port, tandis que les LED fournissent un retour visuel. Les commutateurs DIP de configuration de la zone 3 de la carte déterminent :

- si un port GPIO est rappelé à la masse ou au plus, ou ni l'un ni l'autre ;
- lorsqu'on appuie sur un bouton-poussoir, on obtient un état logique bas ou haut ;
- si les LED sont connectées ou non, et si les périphériques de l'IHM de la zone 2 sont connectés ou non.

Il y a également un commutateur pour

sélectionner la tension de fonctionnement du  $\mu C$ , 3,3 ou 5 V. Deux LED, l'une bleue (3,3 V, froid) et l'autre rouge (5 V, chaud), indiquent la tension d'alimentation de la carte. Pour plus de détail, consulter les **tableaux 1 à 5**.

### Alimentation

La qualité et la robustesse de l'alimentation ont présidé à sa conception. Elle est protégée contre les courts-circuits et les températures élevées, et capable de débiter 1 A sous 5 V sans broncher. Toutefois ce n'est possible que si l'on utilise une source externe comme un adaptateur secteur CC qui délivre au moins 7 V (et, cela va de soi, un courant suffisant). Le port USB peut également être utilisé comme source, mais l'intensité disponible est limitée à 500 mA afin de ne pas surcharger votre ordinateur. Le 5 V alimente un robuste régulateur de tension de 3,3 V à faible tension de déchet de sorte qu'en mode 3,3 V, la puissance suffira à vos expérimentations. Un interrupteur M/A peut couper l'alimentation sur le reste de la carte, ce qui permet une reconfiguration du matériel en toute sécurité sans

## À PROPOS DU PROJET

**Microcontrôleurs**

Arduino ATmega328  
programmation

débutant  
→ **connaissseur**  
expert

environ 4 h

appareil de soudage de composants CMS

environ 75 €

avoir à débrancher l'ordinateur, et donc, sans perdre la connexion du port série.

### Convertisseur USB-série

Le convertisseur USB-série alimente la carte en 5 V, mais il sert aussi d'interface de programmation des croquis Arduino et, naturellement c'est un port série compatible USB pour les applications de l'utilisateur. On peut le déconnecter totalement du  $\mu C$ , par ex. pour libérer des broches GPIO ou l'utiliser sur d'autres ports.

### Dispositifs associés à l'IHM

La zone 2, au milieu de la carte, accueille des dispositifs d'IHM classiques : un buzzer, deux commandes analogiques (potentiomètres), un codeur rotatif et un afficheur alphanumérique. En y associant les boutons-poussoirs et les LED, il est facile de créer des applications conviviales sans rien souder. Avec l'Arduino Uno, quelques cartes de prototypage et un embrouillamini de fils de connexion, on peut arriver au même résultat. Il est quand même plus rassurant et plus fiable d'utiliser un outil comme l'AVR Playground.

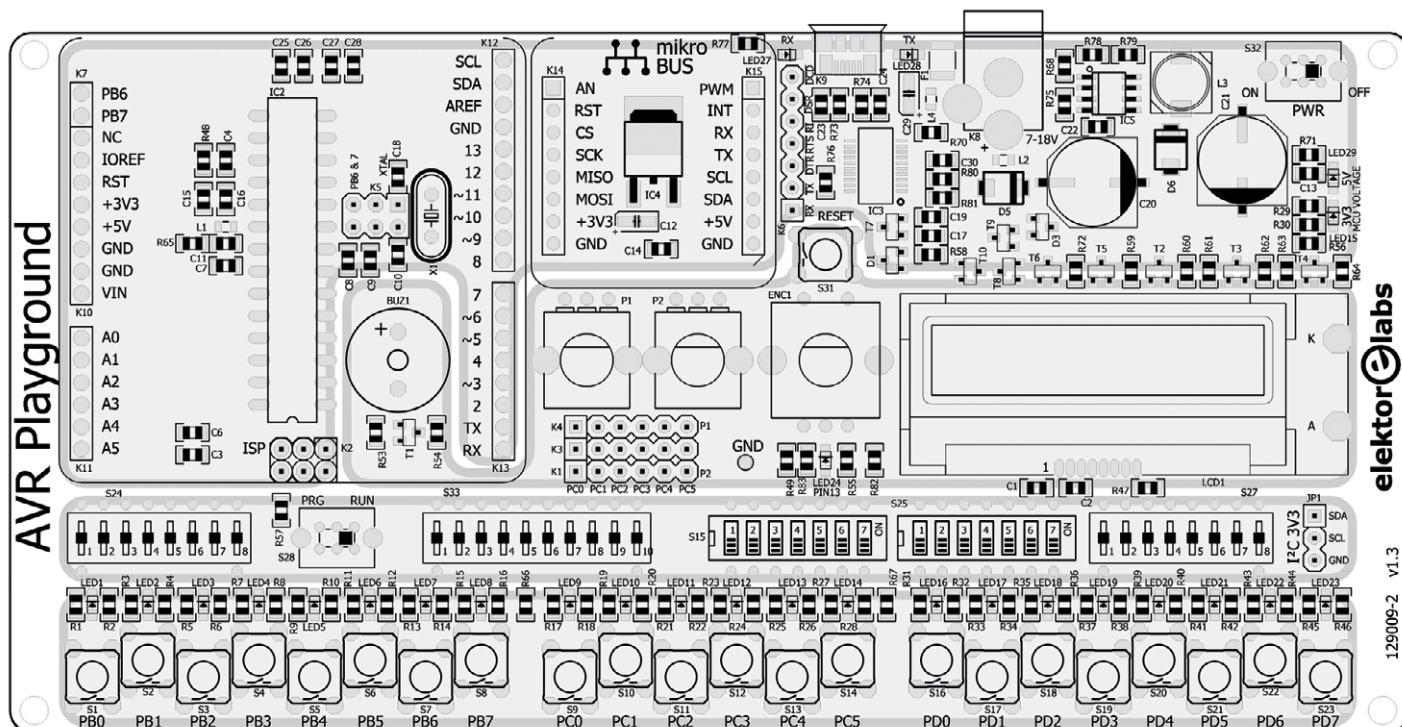


Figure 1. La carte AVR Playground laisse apparaître quatre zones fonctionnelles.

## Afficheur LCD

Le petit afficheur à cristaux liquides avec rétroéclairage possède une interface I<sup>2</sup>C, au lieu des habituelles interfaces

semi-parallèle à 4 bits et parallèle à 8 bits. Deux fils suffisent pour le connecter au µC, c'est son avantage le plus évident. L'inconvénient c'est le pilote spécifique

dont il a besoin, mais nous le fournissons avec nos kits (voir ci-dessous).

## Boutons rotatifs

Deux potentiomètres délivrent des signaux analogiques au µC. En positionnant un cavalier, on les connecte à l'une des six entrées analogiques du µC sans qu'il soit possible de connecter les deux sur la même entrée.

Le codeur rotatif est en réalité équivalent à (max.) trois boutons-poussoirs. Il est en fait relié en parallèle aux boutons-poussoirs sur PD3, PD4 et PD5.

## LED « Arduino »

La LED reliée à PB5 (broche 13) de la carte Arduino Uno est disponible également sur l'AVR Playground, au-dessous du codeur rotatif. Cette LED est présente, car elle est utilisée dans de nombreux croquis.

## Remarques sur la fréquence d'horloge et la réinitialisation

Pour l'AVR Playground, mieux vaut utiliser l'oscillateur RC à 8 MHz interne du µC. Cela garantit que le µC alimenté en 5 ou 3,3 V fonctionnera toujours selon ses spécifications. Comme le quartz est déconnecté par défaut, les ports PB6 et PB7 sont disponibles pour les applications de l'utilisateur. Si vous avez besoin d'un quartz, soudez-en un sur la carte et connectez-le au µC en déplaçant deux cavaliers. En fonctionnement normal, le port PC6

Tableau 1. Fonctions des commutateurs DIP S15.

S15	Fonction	Ouvert	Fermé
1	Buzzer	déconnecté	relié à PB1
2	LED du port B	déconnectée	reliée à GND
3	LED du port C	déconnectée	reliée à GND
4	LED du port D	déconnectée	reliée à GND
5	RXD conv. USB-série	déconnecté	relié à PD1
6	TXD conv. USB-série	déconnecté	relié à PD0
7	DTR conv. USB-série	déconnecté	relié à RàZ

Tableau 2. Fonctions des commutateurs DIP S25 à 7 voies.

S25	Fonction	Ouvert	Fermé
1	Tension du µC	5 V	3,3 V
2	LED « Arduino »	déconnectée	reliée à PB5
3	SDA du LCD	déconnectée	relié à PC4
4	SCL du LCD	déconnectée	relié à PC5
5	Non utilisé		
6	Rétroéclairage LCD	allumé (si S25-7 est fermé)	relié à PD7
7	Rétroéclairage LCD	éteint	commutable

Tableau 3. Fonctions des commutateurs DIP S24.

S24	Port	Bas	Milieu	Haut
1	PB0	rappel bas	sans effet	rappel haut
2	PB1	rappel bas	sans effet	rappel haut
3	PB2	rappel bas	sans effet	rappel haut
4	PB3	rappel bas	sans effet	rappel haut
5	PB4	rappel bas	sans effet	rappel haut
6	PB5	rappel bas	sans effet	rappel haut
7	PB6	rappel bas	sans effet	rappel haut
8	PB7	rappel bas	sans effet	rappel haut



est l'entrée de réinitialisation du µC. Il est possible de désactiver cette fonction en programmant le fusible RSTDSBL du µC. Toutefois, c'est déconseillé, car cela empêche la programmation du µC via le port série **et** le connecteur ISP. Le seul moyen de reprogrammer le µC est alors de le retirer de la carte et de le placer sur un programmeur dit parallèle. Comme la désactivation de l'entrée de RàZ est incompatible avec les objectifs de l'AVR Playground, PC6 n'est pas traité comme les autres ports ; il n'est pas relié à une LED, mais au bouton-poussoir *Reset* pour la réinitialisation.

### Installation de l'AVR Playground

Il est possible d'utiliser l'AVR Playground sans installer le moindre logiciel, mais ce n'est pas souhaitable. Comme cette carte est dotée de nombreuses fonctions non prises en charge par l'EDI standard d'Arduino, nous avons concocté quelques bibliothèques pour faciliter l'utilisation des périphériques embarqués. Si pour une raison quelconque vous ne souhaitez pas ajouter une carte dans l'EDI d'Arduino, utilisez à la place les cartes standard Arduino Pro ou Pro Mini avec *ATmega328* (3,3 V, 8 MHz) (menu « Outils → Processeur »). La tension n'est pas importante, ce qui compte c'est la fréquence qui doit être identique à celle de l'oscillateur de l'horloge du µC.

Par ailleurs, ajouter une nouvelle carte à l'EDI est facile et il y a même un outil spécial pour le faire : le gestionnaire de carte, accessible en haut du menu « Outils → Type de carte ».

Le gestionnaire de carte permet l'installation, la mise à jour et la suppression de cartes tierces. Pour que cela fonctionne, le fabricant de la carte doit fournir un fichier indiquant à l'EDI ce qu'il doit télécharger et utiliser pour cette carte.

La procédure d'installation du fichier de la carte AVR Playground est relativement simple, mais nécessite une connexion à l'internet. Cliquez sur le menu « Fichier » puis ouvrez la fenêtre des « Préférences » de l'EDI Arduino (**fig. 2**, Arduino version 1.6.13 ou plus récente d'arduino.cc ; n'utilisez pas la version 1.7.x d'arduino.org). Copiez l'URL ci-dessous ou, mieux lisez le code QR (**fig. 3**) :

[https://raw.githubusercontent.com/ElektorLabs/arduino/master/package\\_elektor\\_boards\\_index.json](https://raw.githubusercontent.com/ElektorLabs/arduino/master/package_elektor_boards_index.json)

**Tableau 4. Fonctions des commutateurs DIP S33.**

S33	Port	Bas	Milieu	Haut
1	Niveau bouton-poussoir – port B	bas	déconnecté	haut
2	Niveau bouton-poussoir – port C	bas	déconnecté	haut
3	Niveau bouton-poussoir – port D	bas	déconnecté	haut
4	PC0	rappel bas	sans effet	rappel haut
5	PC1	rappel bas	sans effet	rappel haut
6	PC2	rappel bas	sans effet	rappel haut
7	PC3	rappel bas	sans effet	rappel haut
8	PC4	rappel bas	sans effet	rappel haut
9	PC5	rappel bas	sans effet	rappel haut
10	Non utilisé			

**Tableau 5. Fonctions des commutateurs DIP S27.**

S27	Port	Bas	Milieu	Haut
1	PD0	rappel bas	sans effet	rappel haut
2	PD1	rappel bas	sans effet	rappel haut
3	PD2	rappel bas	sans effet	rappel haut
4	PD3	rappel bas	sans effet	rappel haut
5	PD4	rappel bas	sans effet	rappel haut
6	PD5	rappel bas	sans effet	rappel haut
7	PD6	rappel bas	sans effet	rappel haut
8	PD7	rappel bas	sans effet	rappel haut

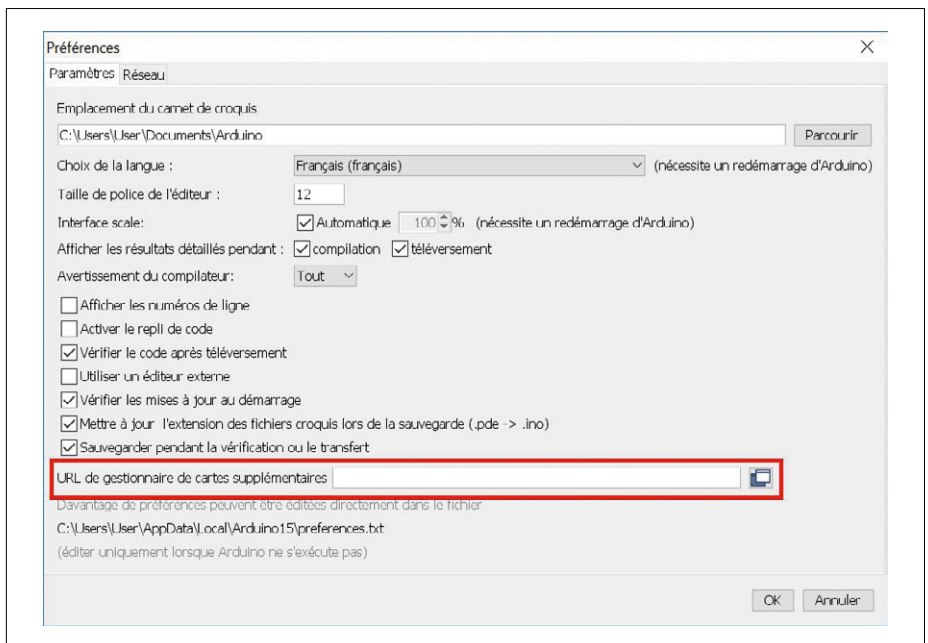
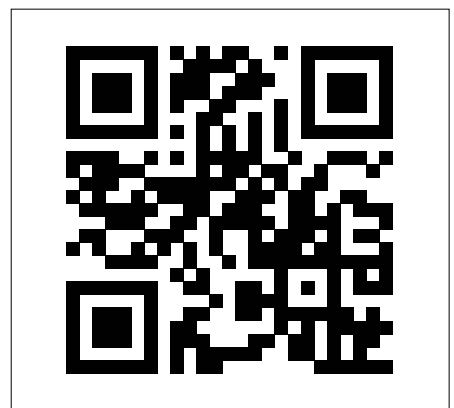


Figure 2. C'est ici qu'il faut saisir l'URL d'accès au logiciel de la carte AVR Playground.

(une seule ligne, aucun espace, attention aux erreurs de frappe) dans la boîte intitulée « URL de gestionnaire de cartes supplémentaires » de la fenêtre des préférences. Quand c'est fait, refermez la fenêtre.

Figure 3. Vous éviterez toute erreur de saisie en lisant ce code QR à l'aide de votre webcam. Ensuite, copiez-collez l'URL dans le champ « URL de gestionnaire de cartes supplémentaires » de la fenêtre des préférences.



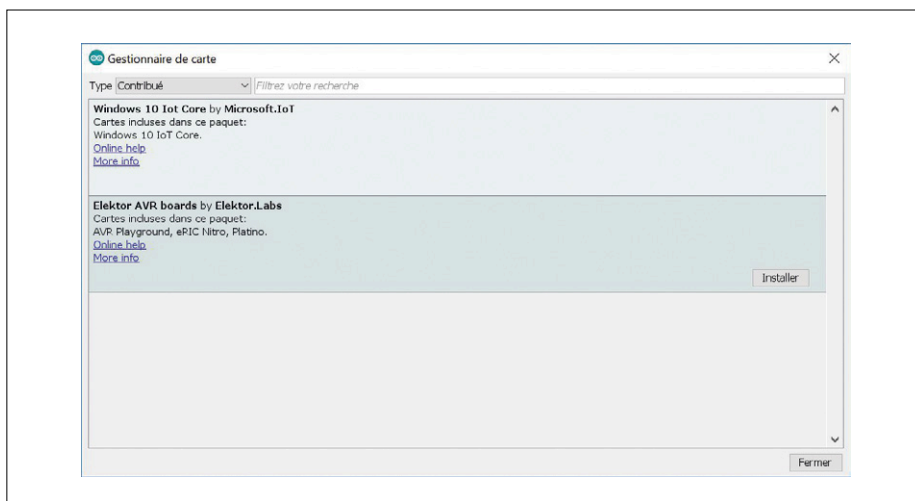


Figure 4. Une fois que l'EDI a localisé le logiciel des cartes AVR Playground, vous pouvez l'installer.

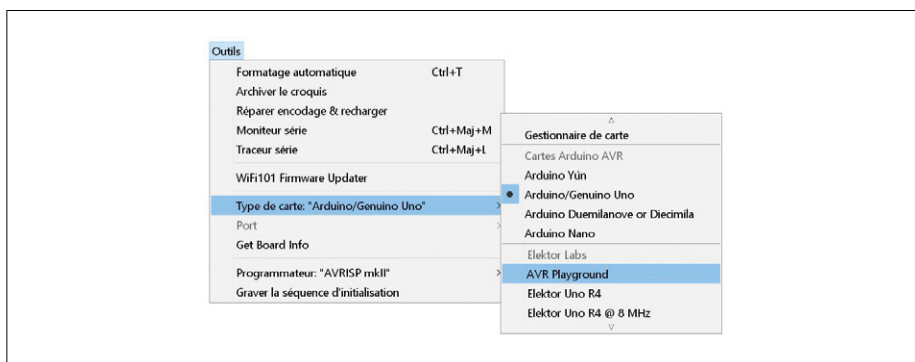


Figure 5. Après installation du logiciel de la carte, déroulez la liste des cartes et sélectionnez AVR Playground.

Ouvrez le gestionnaire de carte (Outils → Type de carte). Dans le coin supérieur gauche de la fenêtre qui s'ouvre, sélectionnez « Contributeur », recherchez AVR Playground sur la liste qui s'ouvre, cliquez dans le champ correspondant, puis cliquez sur le bouton « Installer » qui apparaît alors (fig. 4). L'EDI télécharge les fichiers nécessaires et les enregistre à l'emplacement approprié. Quand c'est fait, refermez la fenêtre. Désormais l'AVR Playground figure sur la liste du menu

« Type de carte » à la rubrique « Elektor Labs » (fig. 5). Naturellement, vous devez connecter l'AVR Playground à l'ordinateur avant de sélectionner son « Port ».

### Devenez développeur de jeu

Nous avons créé une application ludique qui utilise de nombreuses options de la carte : un jeu simple baptisé « Simon Says » (*Jacques a dit*) inspiré d'un célèbre jeu du début de l'ère des micro-contrôleurs. Le jeu d'origine était livré

dans une boîte ronde en plastique noir avec quatre grands boutons colorés et rétroéclairés : rouge, bleu, vert et jaune. L'ordinateur joue une séquence lumineuse aléatoire, chaque couleur est accompagnée d'une note de musique. Ensuite, l'utilisateur doit rejouer la même séquence. Si le joueur se trompe, le jeu recommence. Si le joueur réussit, la séquence est rallongée d'une couleur/note aléatoire. Ce jeu est toujours pratiqué par les programmeurs et vous pouvez facilement en trouver une version pour votre ordiphone ou y jouer en ligne. La carte AVR Playground dispose de tout ce qui est nécessaire pour créer un tel jeu et y jouer : des LED, des boutons-poussoirs, un buzzer et, bien entendu, un µC. En outre, le jeu peut être étendu en mettant l'afficheur LCD à contribution pour afficher les instructions, le score le plus élevé, etc. Il n'y a pas de voyants colorés, mais de petits morceaux de film plastique colorés peuvent faire l'affaire.

Dans la suite, vous trouverez des fragments d'un programme de démonstration, le programme source complet peut être téléchargé gratuitement sur [1].

### Affectation des ports

La première étape consiste à définir quels ports seront utilisés et pour quelles fonctions (tableau 6). Cela détermine aussi en grande partie la configuration des commutateurs DIP (voir tableaux 7 à 11). La configuration choisie allume le rétroéclairage du LCD, et, comme le niveau actif des boutons-poussoirs du port D est réglé sur haut, une LED s'allume automatiquement quand on appuie sur le bouton correspondant. Pour que ce retour visuel fonctionne correctement, les résistances de rappel à la masse de ces boutons-poussoirs doivent être activées.

### Commande des LED

C'est un peu plus compliqué que d'habitude, car les boutons-poussoirs sont reliés aux mêmes broches. Pour faciliter les choses, une table de correspondance est ajoutée. Il suffit de la modifier pour utiliser d'autres LED (listage 1).

### Lecture des boutons-poussoirs

Nous utilisons la même technique que pour allumer les LED, sauf qu'il s'agit de lire une entrée au lieu d'écrire sur une sortie. Avec la configuration choisie pour les commutateurs DIP, une pression

**Tableau 6. Ports, broches et leurs fonctions dans le jeu. Dix broches GPIO sont libres, il est donc possible d'utiliser un µC avec moins de broches, et de réduire le coût de revient.**

Port B	Fonction	Port C	Fonction	Port D	Fonction
PB0	libre	PC0	libre	PD0	RXD
PB1	Buzzer	PC1	libre	PD1	TXD
PB2	libre	PC2	libre	PD2	LED0
PB3	libre	PC3	libre	PD3	LED1
PB4	libre	PC4	SDA du LCD	PD4	LED2
PB5	libre	PC5	SDA du LCD	PD5	LED3
PB6	libre			PD6	LED4
PB7	libre			PD7	LED5



**Tableau 7. Réglages de S15 en fonction de nos plans.**

S15	Fonction	Position
1	Buzzer	fermé
2	LED du port B	ouvert
3	LED du port C	ouvert
4	LED du port D	fermé
5	RXD conv. USB-série	fermé
6	TXD conv. USB-série	fermé
7	DTR conv. USB-série	fermé

**Tableau 9. S24 contrôle le port B, il n'est pas utilisé par notre jeu.**

S24	Port	Position
1	PB0	milieu
2	PB1	milieu
3	PB2	milieu
4	PB3	milieu
5	PB4	milieu
6	PB5	milieu
7	PB6	milieu
8	PB7	milieu

**Tableau 11. S27 contrôle le port D.**

S27	Port	Position
1	PD0	milieu
2	PD1	milieu
3	PD2	bas
4	PD3	bas
5	PD4	bas
6	PD5	bas
7	PD6	bas
8	PD7	bas

**Tableau 8. Réglages de S25 pour notre jeu.**

S25	Fonction	Position
1	Tension du µC	ouvert
2	LED « Arduino »	fermé
3	SDA du LCD (I²C)	fermé
4	SCL du LCD (I²C)	fermé
5	Non utilisé	
6	Rétroéclairage LCD	ouvert
7	Rétroéclairage LCD	fermé

**Tableau 10. S33 contrôle le port C et les niveaux actifs des boutons.**

S33	Fonction	Position
1	niveau bouton-poussoir – port B	milieu
2	niveau bouton-poussoir – port C	milieu
3	niveau bouton-poussoir – port D	haut
4	PC0	milieu
5	PC1	milieu
6	PC2	milieu
7	PC3	milieu
8	PC4	milieu
9	PC5	milieu
10	Non utilisé	

produit un niveau logique haut.

Un bouton-poussoir est un dispositif mécanique et le contact rebondit pendant plusieurs millisecondes avant de se stabiliser. Le logiciel doit éviter cet écueil. En l'absence de contraintes temporelles particulières comme ici, une façon commode de le faire consiste à relire le bouton une seconde fois après un court laps de temps. La pression sur le bouton sera validée uniquement si le niveau haut apparaît deux fois. Avec une scrutation périodique assez rapide (env. 10 Hz), les chances de manquer une pression sont minimales. Un ET logique (« & ») de deux scrutations successives élimine alors les lectures douteuses (**listage 2**).

### Nombres aléatoires

Le jeu a besoin d'une séquence de nombres aléatoires (correspondant au numéro des LED) assez longue pour que la plupart des joueurs se trompent avant la fin. La séquence peut être enregistrée dans un tableau. La taille de ce tableau est importante. S'il est trop petit, il est trop facile pour le joueur de battre la machine. S'il est trop grand, le joueur

est découragé et abandonne. Le tableau peut être rempli au début du jeu.

La fonction `rand` est disponible pour produire des valeurs aléatoires com-

prises entre 0 et `RAND_MAX` (32.767 avec Arduino). Une manière rapide, mais peu élégante de confiner les nombres dans une plage donnée consiste à effectuer

#### Listage 1.

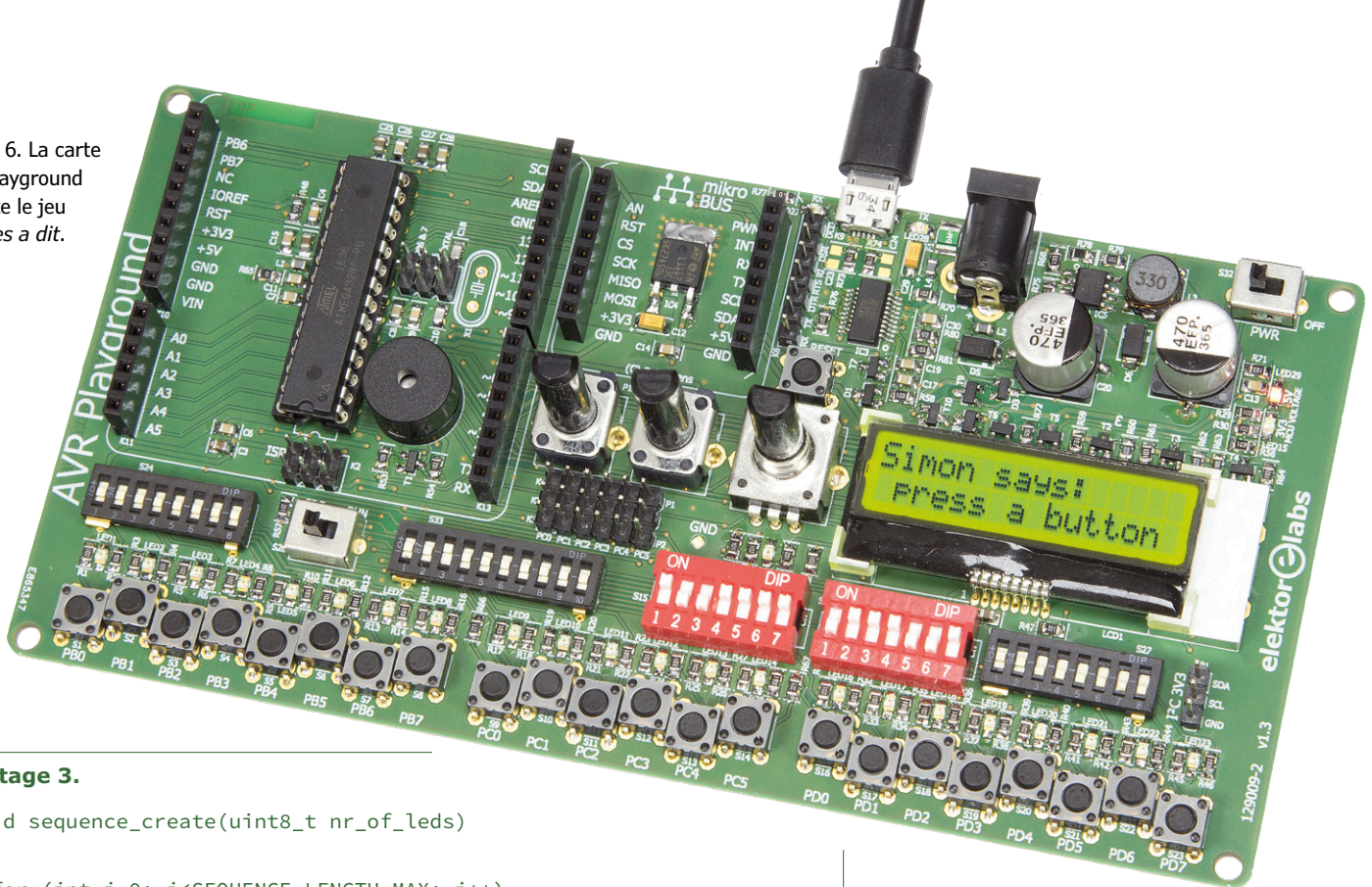
```
#define LED_BUTTON_TABLE_SIZE (4)
uint8_t led_button_table[LED_BUTTON_TABLE_SIZE] = { 4, 5, 6, 7 };

void led_set(uint8_t nr, uint8_t value)
{
    pinMode(led_button_table[nr], OUTPUT);
    digitalWrite(led_button_table[nr], value);
}
```

#### Listage 2.

```
uint8_t button_read_all_debounced(void)
{
    uint8_t buttons = button_read_all();
    delay(10); // Wait for any bouncing to stop.
    buttons &= button_read_all(); // faire un ET avec 2e lecture
    return buttons;
}
```

Figure 6. La carte AVR Playground exécute le jeu Jacques a dit.



### Listage 3.

```
void sequence_create(uint8_t nr_of_leds)
{
    for (int i=0; i<SEQUENCE_LENGTH_MAX; i++)
    {
        sequence[i] = rand()/(RAND_MAX/nr_of_leds + 1);
    }
}
```

### Listage 4.

```
void setup(void)
{
    while (button_pressed()==false);
    srand(millis());
    sequence_create(LED_BUTTON_TABLE_SIZE);
    game_round = 1;
}
```

### Listage 5.

```
const uint8_t buzzer = 9;
const uint16_t button_sound[LED_BUTTON_TABLE_SIZE] =
{ 554, 659, 880, 1319 }; // C#5, E5, A5, E6 in Hz

#define SEQUENCE_SPEED (400) /* ms */

void sequence_play(uint8_t len)
{
    for (int i=0; i<len; i++)
    {
        tone(buzzer,button_sound[sequence[i]],SEQUENCE_SPEED/2);
        led_set(sequence[i],true);
        delay(SEQUENCE_SPEED);
        led_set(sequence[i],false);
        delay(SEQUENCE_SPEED/4);
    }
}
```

une division modulo (« % »). Toutefois, d'un point de vue mathématique, cette approche n'est pas très académique, car la distribution des nombres obtenus ne peut plus être uniforme et indépendante en raison de la manière dont le générateur de nombres aléatoires fonctionne. Une meilleure technique consiste à restreindre la plage de sortie (**listage 4**). Les nombres aléatoires sont problématiques en programmation, car par définition la programmation est déterministe. Pour que la fonction `rand` fonctionne correctement, il faut l'initialiser avec un nombre aléatoire (la graine), un bel exemple du problème de l'œuf et de la poule. Une astuce souvent utilisée consiste à prendre le temps comme graine, car, par définition, il ne s'arrête jamais... sauf sur les systèmes à  $\mu\text{C}$  où il est remis à zéro à chaque redémarrage. Pour notre jeu, notre graine sera tout de même le temps, mais nous utiliserons celui que l'utilisateur met pour appuyer sur le premier bouton. En mesurant ce temps en millisecondes, le même résultat a très peu de chances de se reproduire.

### Lecture d'une séquence, ajout d'un son

À chaque tour de jeu, la séquence comporte 1 bit de plus et doit être jouée à l'utilisateur en repartant du début. Si la séquence est trop rapide, le joueur a moins de chances de pouvoir la mémoriser ; si elle est trop lente, il peut rapi-



dement s'ennuyer. Pour améliorer l'intérêt pour le joueur, nous ajoutons une note musicale à chacune des LED, elle est reproduite par le buzzer (relié à la broche 9, PB1) (**listage 5**).

Le tableau `button_sound` indique la fréquence en hertz de chaque note à jouer. La constante `SEQUENCE_SPEED` (en millisecondes) détermine la vitesse à laquelle la séquence doit être lue. L'argument de la fonction `sequence_play` est la longueur de la séquence à jouer.

### Traitement des actions de l'utilisateur

Une fois que la séquence a été jouée, le joueur doit tenter de la reproduire. Dès que le joueur fait une erreur, le programme peut s'arrêter de lire les boutons-poussoirs. Il y a une seule petite difficulté dont le programme doit tenir compte : il faut éviter qu'une pression très longue soit interprétée comme deux ou plusieurs pressions successives (**listage 6**).

### Jouer le jeu

Il ne reste plus qu'à ajouter une fonction de bouclage `loop` pour exécuter dans le bon ordre les fonctions `sequence_play` et `sequence_read_buttons` et mémoriser la longueur de la séquence et l'état du jeu (**listage 7**).

Le corps des instructions `if-else` est ici minimal ; vous pouvez donner libre cours à votre imagination pour le compléter. Envisagez de jouer un air en fonction de l'état du jeu : un air pour la réussite, un air pour la victoire et un air pour la fin de la partie, le tout judicieusement accompagné de clignotements de LED. Laissez libre cours à votre créativité et amusez-vous à inventer des effets spéciaux. ◀

(160316 – version française : Yves Georges)

### Lien

[1] [www.elektormagazine.fr/160316](http://www.elektormagazine.fr/160316)

#### Listage 6.

```
bool sequence_read_buttons(uint8_t len)
{
    for (int i=0; i<len; i++)
    {
        uint8_t nr, buttons;

        // attendre pression du bouton
        do
        {
            buttons = button_read_all_debounced();
        }
        while (buttons==0);

        // vérifie que la pression est valide
        nr = button_as_number(buttons);
        if (nr!=BUTTON_PRESS_INVALID && nr<LED_BUTTON_TABLE_SIZE)
        {
            // émet un son & attend qu'il se termine
            tone(buzzer,button_sound[nr], SEQUENCE_SPEED/2);
            delay(SEQUENCE_SPEED/2);
            // attend que le joueur relâche le bouton
            while (button_pressed()==true);
            // erreur, ce n'est pas le bon bouton
            if (sequence[i]!=nr) return false;
        }
        else return false;
    }
    return true;
}
```

#### Listage 7.

```
void loop(void)
{
    sequence_play(game_round);
    if (sequence_read_buttons(game_round)==true)
    {
        if (game_round<SEQUENCE_LENGTH_MAX)
        {
            // bien joué, tour suivant
            game_round++;
        }
        else
        {
            // le joueur a gagné, air 'gagné'
            game_start();
        }
    }
    else
    {
        // le joueur a perdu, air 'partie terminée'
        game_start();
    }
}
```





# débogage et décodage de communications numériques

## à l'aide du *SmartScope*

Riemer Grootjans (LabNation)

Le *SmartScope* de LabNation est surtout connu comme oscilloscope complet, mais son connecteur arrière permet de raccorder huit sondes pour mesurer des signaux logiques. Découvrez comment utiliser le *SmartScope* en analyseur logique pour vérifier vos circuits logiques et décoder des protocoles numériques.

La plupart des circuits modernes embarquent de la logique : soit un circuit 100 % logique, soit un cœur logique pour commander un circuit analogique ou acquérir ses signaux de sortie. Si un circuit logique ne fonctionne pas correctement, il est souhaitable d'observer l'état (haut/bas) des lignes logiques et de savoir à quels instants elles changent d'état.

L'utilisation triviale de l'analyseur logique est l'observation d'un événement et de ses effets : vous branchez un fil sur le signal à observer et un autre sur un signal de déclenchement. À chaque déclenchement, l'analyseur logique capture le comportement des deux signaux pendant un intervalle de temps déterminé pour que vous les observiez à loisir.

Dans la plupart des cas, les communications numériques sont des signaux multiples qui doivent respecter une chronologie mutuelle déterminée. Le *SmartScope* offre huit entrées logiques et il peut mesurer la chronologie exacte des changements d'état des signaux (huit max.) dans vos flux de communication. Les analyseurs logiques servent aussi à analyser les communications série entre différentes puces. Des décodeurs vous aident à convertir ces flux de transitions en commandes et valeurs intelligibles.

### Principales caractéristiques

Certaines caractéristiques du *SmartScope* sont importantes pour la fonction analyseur logique.

- **Nombre d'entrées** : bien que deux entrées soient un minimum absolu (signal de déclenchement + signal à observer), il est souhaitable d'en avoir plus, car cela permet de combiner les bits de plusieurs entrées qui représentent des nombres. Il est parfois utile de déclencher sur un nombre donné plutôt que sur un unique signal. Le *SmartScope* dispose de huit lignes logiques sur son connecteur arrière auquel vous pouvez relier soit les fils de 0,1" livrés avec l'appareil, soit le câble spécial analyseur logique de LabNation.
- **Gamme de tension d'entrée et protection** : l'étage analyseur logique du *SmartScope* peut capturer des niveaux logiques hauts entre 1,8 V et 5,0 V. Les nouveaux circuits intégrés utilisent généralement des tensions plus basses pour réduire la consommation et les émissions électromagnétiques produites par le va-et-vient des charges électriques dans les pistes et les broches.



- Chaque entrée du *SmartScope* est protégée par des diodes dans les deux sens pour garantir l'innocuité de signaux fortuits, des pointes de tension et des décharges électrostatiques (ESD). Toutefois n'appliquez jamais de tension **continue** supérieure à 5,5 V.
- **Vitesse d'échantillonnage** : le *SmartScope* échantillonne vos signaux à 100 Méc/s, ce qui permet de visualiser des impulsions brèves (10 ns), ou bien de mesurer le rapport cyclique de signaux à 500 kHz avec une résolution de 1 %.
- **Mémoire d'échantillon/RAM** : une petite mémoire d'échantillons (<10 K) suffit pour observer des phénomènes « cause → effet » simples. Cependant, pour capturer une communication entre des circuits logiques, vous aurez besoin de beaucoup plus de mémoire d'échantillons. C'est pourquoi le *SmartScope* embarque une RAM capable de stocker 4 Méc sur chaque voie. Cela permet de capturer une transmission complète et de passer le temps nécessaire pour observer chaque partie à la loupe, la décoder et contrôler les fronts. Ce qui nous amène au point suivant...
- **Décodeurs de protocole** : vérifier l'ensemble des fronts d'une communication numérique peut être extrêmement fastidieux et frustrant. On peut traduire manuellement de petites séquences, mais le comptage des fronts vous donnera vite des maux de tête.  
Pour faciliter cette tâche, le logiciel du *SmartScope* comprend les routines de décodage des protocoles courants comme 1-wire, UART/RS232, I<sup>2</sup>C, I<sup>2</sup>S et SPI. Le système de décodage du *SmartScope* est très souple de sorte que les décodeurs peuvent fournir des informations complémentaires comme le nombre total de fronts ou l'intervalle de temps entre ceux-ci. Vous pouvez même créer votre propre décodeur [1].  
Pour être complet, ajoutons que sur la plupart des analyseurs logiques USB, le décodage des protocoles s'effectue par logiciel. En conséquence, les données sont capturées sur un intervalle de temps déterminé et transférées au PC qui les décode par logiciel. Bien qu'il soit possible de parcourir les données décodées, il n'y a pas de moyen fiable de déclencher sur celles-ci. Pour cela, il faudrait que les données décodées soient disponibles dans le matériel, et les analyseurs logiques capables de le faire valent plusieurs fois le prix d'un *SmartScope* complet.

## Familiarisation avec l'analyseur logique

Cette partie explique comment bien démarrer avec l'utilisation de l'analyseur logique du *SmartScope* et comment obtenir les meilleurs résultats.

- **Raccordez-vous correctement au matériel** : tout comme un oscilloscope, un analyseur logique a besoin d'être référencé au niveau 0 V de l'objet des mesures. C'est pourquoi vous devez effectuer au moins une connexion entre la masse du *SmartScope* et la masse de l'appareil sur lequel vous faites des mesures. Ensuite, connectez les entrées du *SmartScope* sur les signaux du circuit dont vous voulez capturer les données.
- **Configurez le logiciel** : après avoir raccordé le *SmartScope* à votre PC/tablette/ordiphone, lancez l'appli *SmartScope* et sélectionnez Menu → Digital mode. Par défaut, les huit voies logiques s'affichent. Si vous n'avez

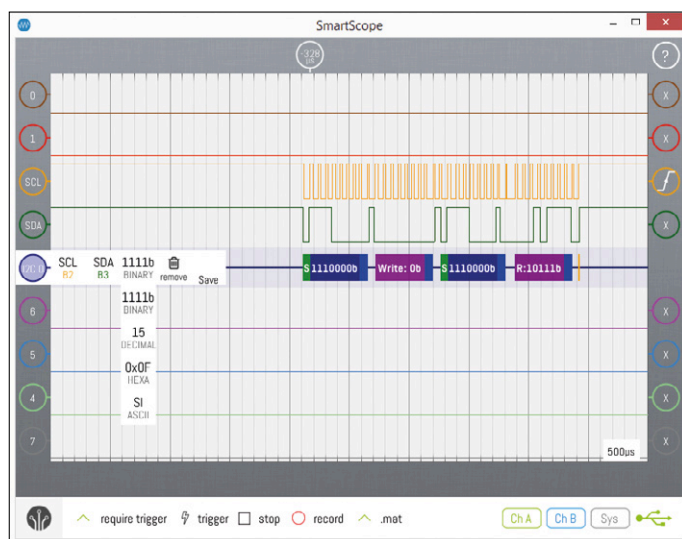


Figure 1. Mise en œuvre d'un décodeur de protocole.

pas besoin de connecter toutes les voies, et voulez libérer de l'espace, il suffit de cliquer sur une voie inutilisée et de choisir « Hide ». La voie est retirée et son pictogramme apparaît en bas à droite de l'écran (**fig. 5**). Il suffit de cliquer sur ce pictogramme pour restaurer le signal.

- **Configurez le déclenchement** : une fois le circuit à observer raccordé, apparaîtront probablement des signaux qui passent de façon aléatoire. Définissez les conditions de déclenchement pour demander au *SmartScope* d'attendre une combinaison donnée de signaux qui déclenche la capture et l'affichage. Pour cela, modifiez les symboles de l'indicateur de déclenchement à droite de chacun des signaux. La **figure 1** montre par ex. que le *SmartScope* attend un front montant sur D2.

La position de déclenchement est aussi réglable vers la droite et la gauche pour pouvoir rechercher ce qui vous intéresse. Notez que la plage de déclenchement est très vaste sur le *SmartScope* ; elle déborde largement de l'écran et le déclenchement peut même être situé au-delà des limites de la RAM. Remarquez qu'à l'intérieur de la RAM, la position de déclenchement est signalée par un trait vertical rouge en haut de l'écran.

Terminons avec le mode de déclenchement. En mode « Require trigger », le *SmartScope* attend la réalisation d'une condition de déclenchement et affiche ensuite seulement la séquence capturée. Le mode « Normal trigger » se comporte exactement de la même manière, mais si aucune condition de déclenchement n'est détectée au bout d'une seconde, le *SmartScope* se contente de capturer des séquences à volonté puis les affiche sur l'écran. Le mode « Single trigger » attend qu'une condition de déclenchement donnée se produise, effectue une seule acquisition puis l'envoie en totalité à l'écran (tout le contenu de la RAM est transféré).

## Accès aux informations de base

Avec une acquisition gelée sur l'écran, comme le montre la **figure 2**, il y a énormément d'informations auxquelles vous avez accès immédiatement.

- **Chronologie des intervalles** : l'écran affiche l'intervalle de temps entre tous les fronts pour la trace sélectionnée.



Figure 2. Interfaçage à un capteur d'image RVB 5 MP. Les lignes du bas sont HSync et VSync, les quatre lignes du haut les données des pixels eux-mêmes. D4 et D5 sont les données et l'horloge de la voie I<sup>2</sup>C, avec un décodeur ajouté pour décoder les données.



Figure 3. Message I<sup>2</sup>C décodé.

Un survol de la souris affiche ces informations pour n'importe quelle trace.

- **Curseur de temps** : faites glisser un curseur de temps latéral jusqu'à l'endroit voulu de la trace. En ajoutant un second curseur, un curseur de différence apparaît ; il indique l'intervalle de temps entre les deux. Si vous cliquez sur ce curseur de différence, le temps est converti en la fréquence correspondante.
- **Décodeurs de fronts** : pour connaître le nombre exact de fronts entre deux positions, sélectionnez : *Menu* → *Add decoder* → *Edge counter*. Sélectionnez la voie voulue et le décodeur comptera les fronts pour vous. En ajoutant le décodeur d'intervalles entre fronts « *Edge intervals* », vous obtiendrez le temps entre chaque intervalle avec une précision de 10 ns.

### Utilisation des décodeurs numériques

Dans un grand nombre de systèmes, les puces communiquent entre elles par des voies assez lentes (par ex. données de configuration ou mesures à faible vitesse).

La figure 2 contient une telle voie : le contrôleur envoie un

message à un capteur d'image RVB selon le protocole I<sup>2</sup>C. Chaque protocole est bien défini ; en I<sup>2</sup>C, le message commence par un octet qui contient l'adresse du circuit destinataire et indique si le maître veut lire ou écrire. Chaque bit de la ligne SDA doit être interprété au moment d'un front montant de la ligne SCL.

Sachant cela, vous pouvez commencer à décoder les données que vous voyez sur l'écran en écrivant la valeur du signal SDA au moment de chaque front montant du signal SCL. Vous pouvez aussi demander au *SmartScope* de le faire pour vous : sélectionnez *Menu* → *Add decoder* → *I2C Decoder*. Le décodeur est ajouté à l'écran et le logiciel tente de rechercher quel signal est le SDA et quel signal est le SCL. La **figure 3** montre une situation plus complexe avec des lignes multiples travaillant à haute et basse fréquence, il faut donner un coup de main au logiciel. Avec le menu décodeur ouvert par défaut, cliquez sur le bouton SCL et sélectionnez D4. Ensuite cliquez sur le bouton SDA et sélectionnez D5. Enfin cliquez sur le pictogramme lui-même pour faire disparaître le menu et vous verrez que le décodeur a décodé la totalité de la communication pour vous.

### Changement du type de sortie du décodeur

Parfois vous souhaitez connaître les valeurs décodées sous forme décimale simple tandis que d'autres fois vous aurez besoin de la notation hexadécimale. Vous pouvez le spécifier en cliquant sur le pictogramme de décodeur, et en cliquant sur le sous-menu *radix* (base) puis en sélectionnant le type de sortie souhaitée (fig. 1). De cette manière, vous pouvez par ex. traduire directement une communication UART en ASCII.

### Création de votre propre décodeur

Le sous-système de décodage du *SmartScope* est très souple. Vous pouvez même programmer votre propre décodeur en C#, le compiler et le mettre dans le même dossier que l'appli [1]. Cela vous permet de créer un décodeur pour vos propres protocoles et vous pouvez même mettre les décodeurs en cascade. De cette façon, vous pouvez par ex. créer un décodeur qui prend la sortie du décodeur I<sup>2</sup>C et traduit le registre d'adresses en registre de noms.

### Obtention de résultats optimaux

Les fils peuvent nécessiter quelques précautions pour obtenir des résultats corrects, en particulier avec les signaux à haute fréquence qui sont pilotés activement aux niveaux haut et bas. Comme les fils de liaison entre le *SmartScope* et l'appareil à observer ne sont pas blindés, une intermodulation entre fils peut se produire. Ainsi, un front montant sur une voie peut induire une très courte impulsion sur un fil voisin. Si les conducteurs ne sont pas blindés, comme il s'agit d'interférences d'origine électrique et magnétique, la meilleure stratégie est d'utiliser des liaisons aussi courtes que possible, des conducteurs assez espacés, mais proches d'un conducteur de masse.

En outre, un conducteur de mesure agit comme un petit condensateur ajouté sur le signal à observer et cela peut influencer le signal lui-même au point de causer des erreurs de communication, juste parce que vous avez connecté un fil au signal.

### Câble de connexion de l'analyseur logique LabNation

Eu égard au problème ci-dessus, LabNation a développé pour l'analyseur logique un câble spécifique (**figure 4**) à raccorder



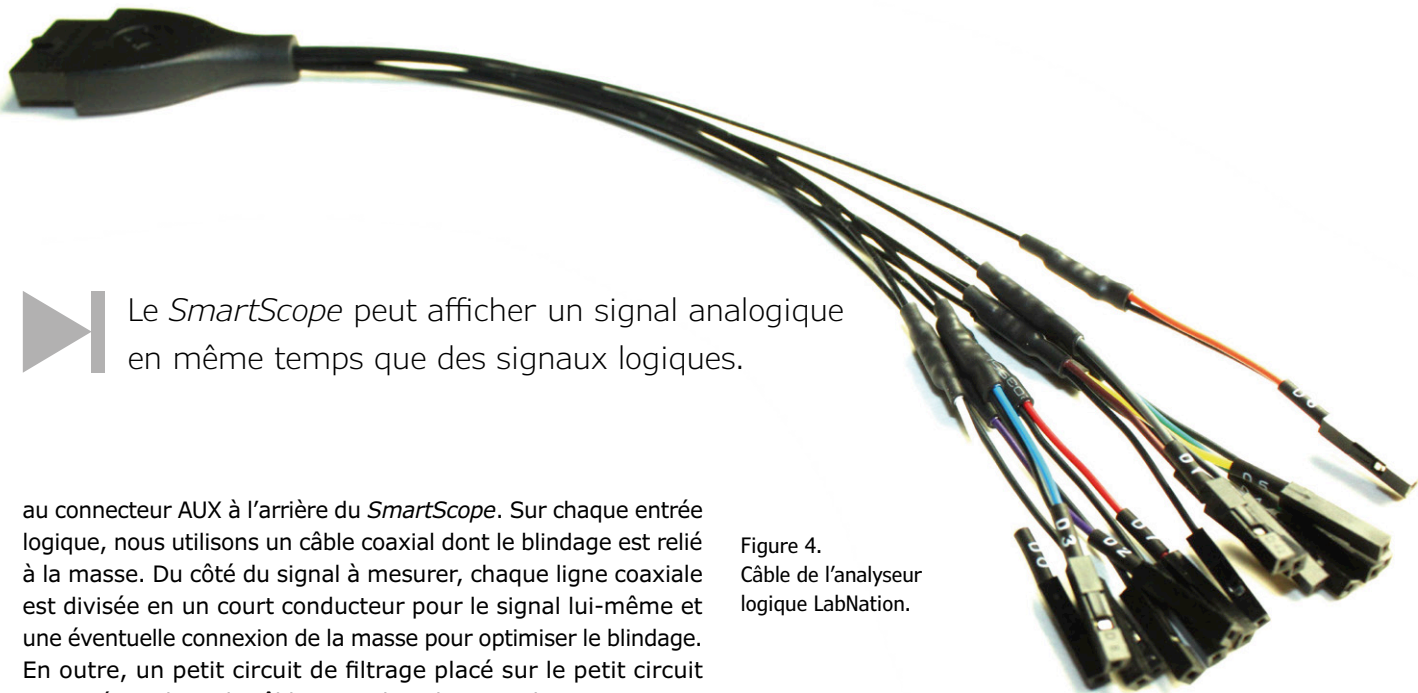


Figure 4.  
Câble de l'analyseur  
logique LabNation.

► Le *SmartScope* peut afficher un signal analogique en même temps que des signaux logiques.

au connecteur AUX à l'arrière du *SmartScope*. Sur chaque entrée logique, nous utilisons un câble coaxial dont le blindage est relié à la masse. Du côté du signal à mesurer, chaque ligne coaxiale est divisée en un court conducteur pour le signal lui-même et une éventuelle connexion de la masse pour optimiser le blindage. En outre, un petit circuit de filtrage placé sur le petit circuit imprimé qui divise le câble coaxial en deux conducteurs permet de réduire au mieux l'influence du câble sur le signal lui-même.

### Stratégie de décodage

Si vous voulez capturer un flux de communication relativement long pour le décoder et l'enregistrer dans un fichier de résultats, l'analyseur logique est incontournable. Voici le meilleur moyen de le faire.

Établissez une bonne connexion entre le *SmartScope* et l'appareil à tester. Préférez le câble pour analyseur logique ou assurez-vous que les fils des signaux sont séparés et que le fil de masse est au plus près du fil véhiculant la fréquence la plus haute. Assurez-vous que la connexion de masse est parfaite entre les deux appareils.

Définissez la condition de déclenchement, en général un front montant ou descendant sur l'un des deux signaux. Assurez-vous que les signaux non utilisés pour le déclenchement soient sur « X », sinon le déclenchement risque de ne jamais se produire. Assurez-vous que le *Panorama* (vue de la RAM en haut de l'écran) couvre l'ensemble de la communication à analyser. En général vous déplacerez la position de déclenchement (trait rouge vertical) tout à gauche du panorama. La manière optimale pour y arriver est de travailler en déclenchement automatique et d'écouter quelques communications.

Pour finir, sélectionnez le déclenchement monocoup et attendez la communication suivante. Dès que le déclenchement se produit, le contenu total de la RAM est transféré vers l'hôte et vous pouvez y naviguer à votre rythme et ajouter les décodeurs appropriés.

Si vous avez besoin d'un panorama si long que la fréquence d'échantillonnage tombe au-dessous de 100 Méc/s, sélectionnez *Menu* → *System* → *Acquisition depth* → *4MS*. Cette astuce vous permet d'utiliser la totalité de la RAM, mais le transfert vers l'hôte prend plus de temps.

### Le mode mixte en renfort

Dans certains cas, vous voyez des signaux parfaits sur votre écran, mais votre circuit ne réagit pas. Peut-être que la communication utilise une logique en 3,3 V alors que le circuit attend du 5 V. Bien que cela donne des signaux parfaits vus

de l'analyseur logique, il ne se passe rien pour le circuit 5 V. Parfois c'est l'inverse, tout fonctionne correctement, mais vous voyez des pointes indésirables sur l'écran de l'analyseur.

Alors vous aimeriez avoir en même temps un analyseur logique et un oscilloscope. C'est exactement ce que le *SmartScope* sait faire. Sélectionnez *Menu* → *Mixed mode*, et vous avez une voie analogique en plus de l'analyseur logique complet.

La voie analogique et les voies logiques sont mesurées de façon synchrone. Si vous connectez la sonde analogique sur l'une des sondes logiques vous pourrez voir la forme analogique du signal ainsi que le résultat logique (fig. 5). Notez que la portion visible des traces représente moins de 1 % des données contenues dans la RAM, et qu'elle détaille parfaitement les formes d'onde analogiques et logiques. ◀

(160282 – version française : Yves Georges)

### Lien

[1] [http://wiki.lab-nation.com/index.php/Creating\\_your\\_own\\_Protocol\\_Decoder](http://wiki.lab-nation.com/index.php/Creating_your_own_Protocol_Decoder)

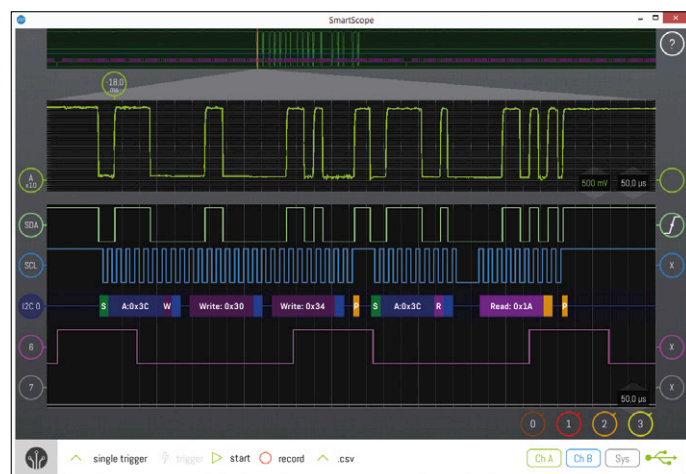


Figure 5. Mode mixte, avec voie A sur SDA.



# vol tous azimuts

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

## Comptoir ouvert du Troisième Type



Maker Store Berlin, Danziger Str. 22,  
10435 Berlin, Allemagne

Évènement surprenant à Berlin en décembre 2016.

Ni artistique, ni sportif, ni alimentaire (bien que boissons et pizzas y furent servies) : l'ouverture d'un magasin. Et pas juste « un magasin » mais le premier *Arduino Store* au monde. Mon expérience est que les boutiques d'électronique du type une vitrine et une caisse enregistreuse ferment, ceci depuis près de 35 ans. Les boutiques d'électronique virtuelles fleurissent, surgissant partout sur l'internet, mais ouvrir un magasin d'électronique avec un comptoir et payer quelqu'un pour rester derrière, en fait c'est un concept nouveau – voire révolutionnaire ? Dans ce nouveau magasin Arduino, vous pouvez acheter des produits Arduino (!), mais pas seulement, car il dispose d'un espace de travail pour

organiser des ateliers. Le début d'une nouvelle tendance ?

Les boutiques d'électronique vont-elles bientôt fleurir à tous les coins de rue comme avant ? Ou bien ce magasin va-t-il subir le même sort que ses prédécesseurs ?



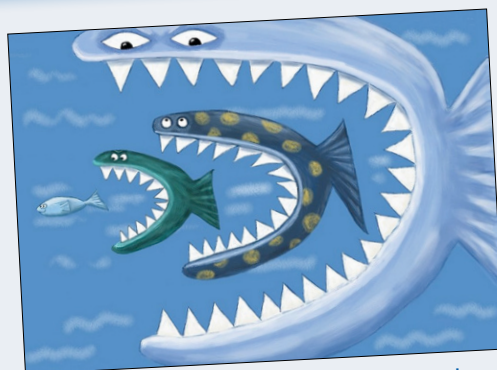
## Trucs & Astuces

**Q.** Comment pouvez-vous dire si vous travaillez sur du courant alternatif ou continu ?

**R.** Si c'est de l'alternatif, vous claquez des dents en tenant les conducteurs.  
Avec du continu, elles se bloquent.

## La jungle des semi-conducteurs : manger ou être mangé

Observateur de l'industrie électronique ou simple lecteur de journaux, vous aurez noté la vague de fusions et rachats survenus ces dernières années. Intel a acquis le fabricant de FPGA Altera. NXP a repris Freescale juste avant d'être absorbé lui-même quelques mois plus tard par Qualcomm. Microchip a englouti son concurrent Atmel ; ON Semiconductor a racheté Fairchild Semiconductor et Analog Devices a avalé Linear Technology. Le distributeur de composants Avnet a repris son collègue Farnell qui venait de vendre Cadsoft (Eagle) à Autodesk (AutoCAD) ; Mentor Graphics a fusionné avec Siemens et, juste après le Brexit, ARM a été rachetée par le Japonais Softbank. Saviez-vous qu'Amazon était aussi entré dans la danse ? Ils possèdent Annapurna Labs en Israël. Ce ne sont que les fusions très médiatisées ou les acquisitions à plusieurs centaines de millions, voire dizaines de milliards, d'Euros/Dollars/Livres, mais il y a eu beaucoup d'autres prises de contrôle moins coûteuses. Les petites compagnies fusionnent pour former des entités plus grandes, mais quand elles deviennent trop grosses, elles commencent à essayer de perdre du poids. NXP était issu de Philips, Freescale faisait partie de Motorola ; HP a décidé de se couper en deux, et fin 2016 Samsung a annoncé aussi une possible scission. C'est comme regarder les étoiles et les planètes se percuter et se désintégrer de nouveau. Un trou noir caché quelque part dans cet univers serait-il en train d'essayer d'aspirer toute l'industrie ?



Gros Poisson mange Petit Poisson,  
crédit image : [www.catherineswenson.com](http://www.catherineswenson.com)

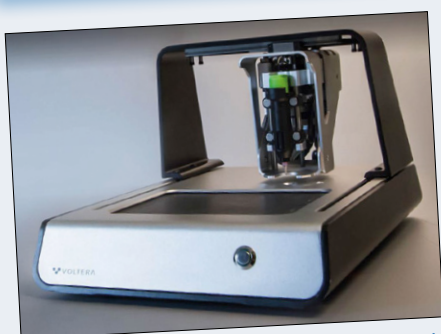


Color	1 <sup>st</sup> band	2 <sup>nd</sup> band	3 <sup>rd</sup> band (multiplier)	4 <sup>th</sup> band (tolerance)	Temperature Coefficient
Black	0	0	$\times 10^0$		
Brown	1	1	$\times 10^1$	$\pm 1\%$ (F)	100 ppm
Red	2	2	$\times 10^2$	$\pm 2\%$ (G)	50 ppm
Orange	3	3	$\times 10^3$		15ppm
Yellow	4	4	$\times 10^4$		25ppm
Green	5	5	$\times 10^5$	$\pm 0.5\%$ (D)	
Blue	6	6	$\times 10^6$	$\pm 0.25\%$ (C)	
Violet	7	7	$\times 10^7$	$\pm 0.1\%$ (B)	
Gray	8	8	$\times 10^8$	$\pm 0.05\%$ (A)	
White	9	9	$\times 10^9$		
Gold			$\times 10^{-1}$	$\pm 5\%$ (J)	
Silver			$\times 10^{-2}$	$\pm 10\%$ (K)	
None				$\pm 20\%$ (M)	

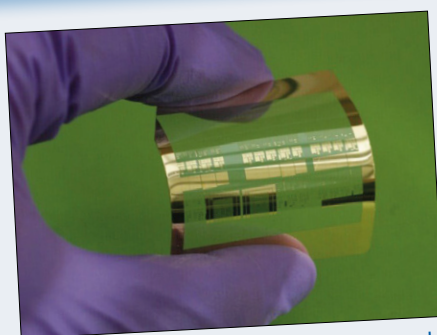
## RS-279 ?

La plupart des lecteurs d'Elektor connaissent la norme RS-232, beaucoup la RS-485, certains la RS-422, mais qui connaît la RS-279 ? Vous ! Surprenant, mais depuis 1963, la RS-279 que vous avez mémorisée inconsciemment au fil des années, est la norme EIA officielle pour le code de couleur des résistances. Élaborée il y a près d'un siècle et devenue une norme CEI en 1952, elle couvre aussi les condensateurs. Les CMS n'utilisent pas de code de couleur, mais des lettres et chiffres. La norme internationale actuelle définissant les codes de marquage pour les résistances et condensateurs est la CEI 60062:2016.

## L'imprimante de circuits imprimés — encore utile ?



Le système d'impression et d'assemblage de circuit imprimé, pour bureau, de Voltera allie design et fonctionnalité.



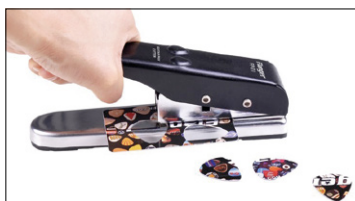
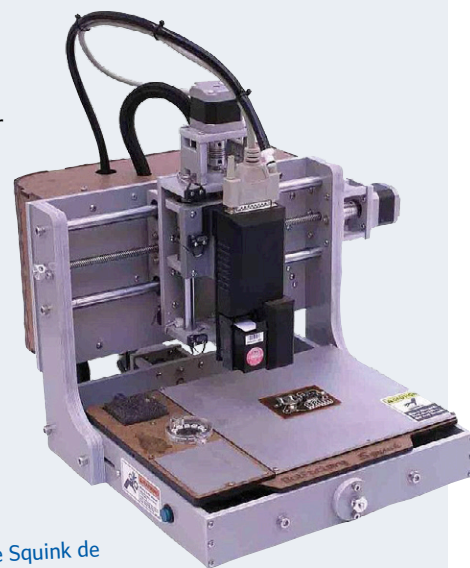
Ces minuscules transistors avec une bande passante de 38 GHz ont été gravés et imprimés sur un support flexible (crédit photo : Jung-Hun Seo).

L'électronique ne serait pas où elle en est aujourd'hui sans une technique développée il y a environ 500.000 ans ! Je ne parle pas de la découverte du silicium, encore plus ancienne, quelques milliards d'années au moins, mais de la gravure : « l'action d'inciser un dessin dans une surface dure, généralement plate en creusant des sillons dedans » (merci Wikipédia).

Les Circuits Imprimés, ou CI, sont réalisés avec différentes techniques de gravure (chimique,

mécanique ou optique), et les circuits intégrés sont gravés sur des galettes de silicium. L'impression, une invention âgée de plus de 2000 ans, est aussi devenue essentielle pour l'industrie électronique. Avant tout parce que sans elle vous ne pourriez pas lire cette belle prose, mais aussi parce que la fabrication des CI repose dessus. Le « I » de CI prend aujourd'hui une nouvelle dimension, à cause du développement d'imprimantes capables de produire des CI sans gravure, par simple jet d'encre conductrice sur une surface dure, généralement plane. Des imprimantes de circuits électroniques sont également à l'étude. On travaille à l'impression d'un simple microcontrôleur à 4 bits directement sur du carton. Le but est de créer des emballages intelligents, des vêtements connectés ou tout autre *bidule communicant*. Aujourd'hui on pourrait se demander qui voudrait d'un processeur à 4 bits, mais en 1971, quand Intel a sorti sa centrale 4004 à 4 bits, ce fut une révolution qui permit de faire voler des avions de chasse. Que préféreriez-vous avoir dans votre labo ? Une imprimante qui imprime des CI qu'il faut ensuite garnir, ou une imprimante qui crache des circuits prêts à fonctionner ? Faut-il abandonner l'imprimante de CI et nous concentrer sur l'imprimante de circuits électroniques ?

En plus d'imprimer les circuits imprimés, le Squink de BotFactory place et soude les composants.



## Outil de labo indispensable

Le médiateur de guitare ou l'onglet pour forcer l'ouverture de gadgets récalcitrants sans (trop) les abîmer. Vous les perdez toujours ? Procurez-vous un emporte-pièce à médiateur.

**Vous voulez participer ? Envoyez vos commentaires, suggestions, trucs et astuces à [labs@elektor.com](mailto:labs@elektor.com)**

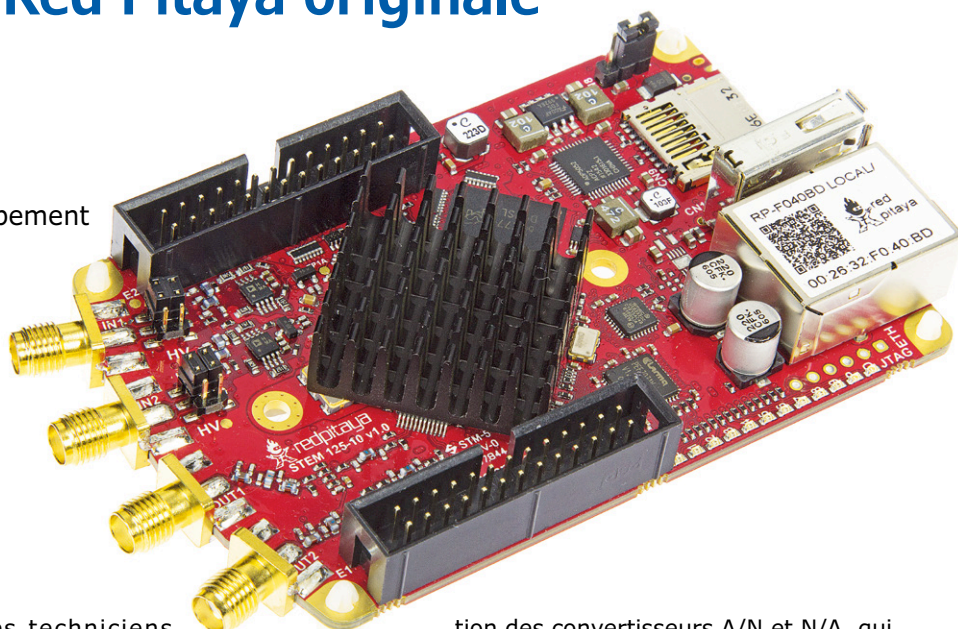


# banc d'essai : **STEMlab 125-10**

## la petite sœur de la Red Pitaya originale

Harry Baggen (labo d'Elektor)

La plateforme de mesure et de développement bien connue Red Pitaya change de nom et est désormais disponible en deux versions : 10 ou 14 bits. Nous avons testé la première, dénommée STEMlab 125-10.



La carte *Red Pitaya*, de couleur rouge et au format d'une carte de crédit, est bien connue des électroniciens. C'est un instrument de mesure puissant, avec un générateur intégré, et un outil de développement pour circuit logique programmable (*Field-Programmable Gate Array* – *FPGA*) polyvalent, ce qui permet la mise en œuvre de diverses applications. Elektor a publié quelques articles illustrant ces applications, ainsi qu'un livre (en anglais) [1].

Les techniciens de chez *Red Pitaya* ont renouvelé et étendu leur offre, et les cartes s'appellent désormais toutes *STEMlab* (de *Science, Technology, Engineering and Mathematics*, acronyme très utilisé dans l'enseignement des sciences exactes et appliquées aux États-Unis). L'« ancienne » carte s'appelle donc maintenant *STEMlab 125-14*, et elle a une petite sœur, la *STEMlab 125-10* ; cette dernière diffère par la résolu-

tion des convertisseurs A/N et N/A, qui est de 10 bits, et la présence de moins de mémoire vive. Dans la plupart des cas, ces différences n'ont pas d'importance.

### Moins de bits

Nous avons testé un prototype et un modèle de série de la *STEMlab 125-10* pendant quelque temps, et nous n'avons pas pu constater qu'il s'agissait d'une version « réduite » de notre carte *Red Pitaya* : les deux versions tournent

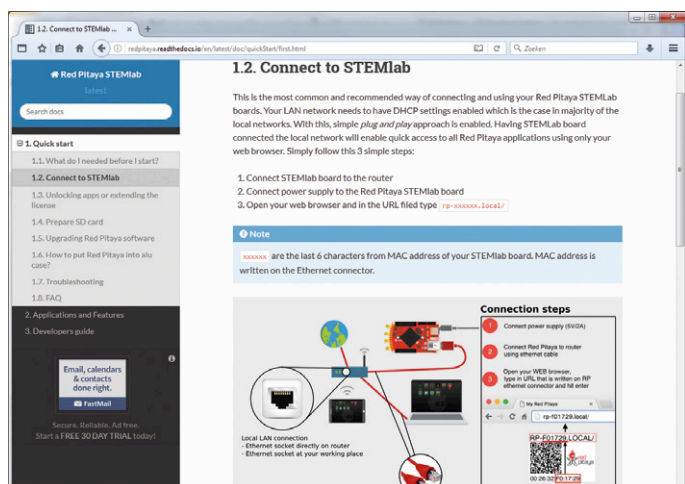


Figure 1. La documentation sur le site de *Red Pitaya*.

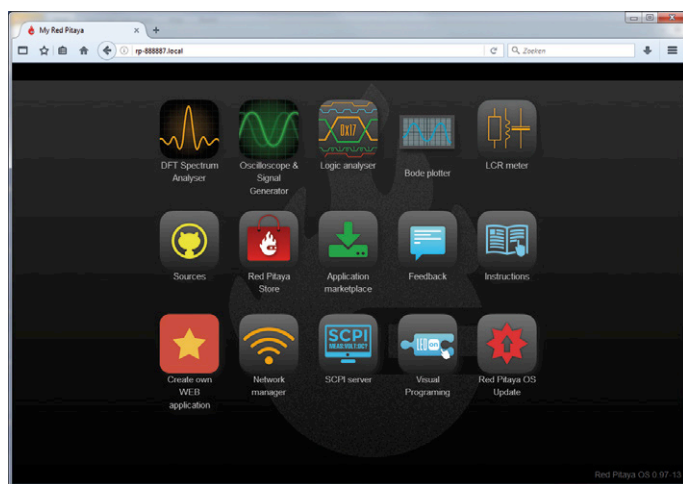


Figure 2. La nouvelle interface utilisateur, avec des icônes semblables à celles d'un ordiphone.



Une différence de 4 bits peut sembler importante, mais c'est très relatif : presque tous les oscilloscopes USB et autonomes d'un prix équivalent, voire plus chers, sont équipés de convertisseurs à 8 bits ! Notre plateforme à 10 bits a une résolution quatre fois plus élevée, ce qui permet la visualisation très détaillée d'un signal sur un écran d'oscilloscope. Une résolution de 14 bits n'est intéressante que pour ceux qui écrivent leurs propres applis et qui veulent des mesures vraiment très, très précises.

## Premiers pas

rubrique **HELP**. Cette documentation est plus conviviale que par le passé, et la première page vous montre comment vous connecter et vous lancer en deux temps trois mouvements. Le plus simple est de vous raccorder directement à un routeur avec un câble Ethernet ; la reconnaissance de la carte est immédiate, il vous suffit d'une seule commande avec l'adresse **MAC** (*Media Access Control*) dans le navigateur. Vous aurez alors accès à l'interface utilisateur, qui utilise des icônes semblables à celles d'un ordiphone (**figure 2**). Certaines applis sont préinstallées. Connectez-vous à *Marketplace* si vous voulez plus d'applis ou de nouvelles fonctions, comme un gestionnaire de réseau et la possibilité de mettre le système d'exploitation à jour à partir du navigateur. Le gestionnaire de réseau permet bien entendu d'utiliser un réseau Wi-Fi existant, il suffit d'y ajouter la carte et de saisir un mot de passe ; l'adresse IP de la carte apparaîtra alors.

## Conclusion

Plusieurs versions du kit sont disponibles, tant pour la *STEMlab 125-10* que pour la *125-14* [4]. L'adaptateur réseau et la carte mémoire sont bien sûr standard pour toutes les versions, les kits ne diffèrent donc que par les accessoires fournis (ou non). Ceux-ci sont disponibles séparément, et vous pourrez toujours les acquérir par la suite en fonction de vos besoins : boîtier, module LCR, câbles, etc.

(160366 – version française : Jean-Louis Mehren)

- [1] [www.elektor.fr/red-pitaya-for-test-and-measurement](http://www.elektor.fr/red-pitaya-for-test-and-measurement)
- [2] [www.redpitaya.com](http://www.redpitaya.com)
- [3] [www.elektormagazine.fr/150595](http://www.elektormagazine.fr/150595)
- [4] [www.elektor.fr/developpement/red-pitaya/](http://www.elektor.fr/developpement/red-pitaya/)

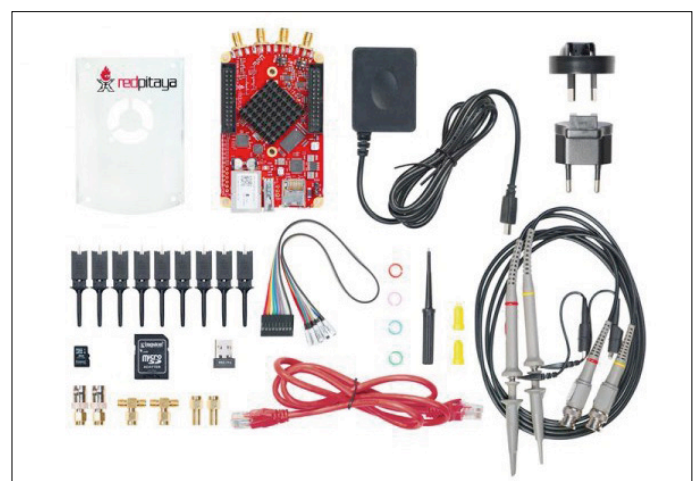


Figure 4. Avec la version « *Diagnostic* » du kit, vous disposez d'un banc de mesure complet chez vous.



# générateur de nombres réellement aléatoires

tests NIST & composants abordables



Lorsque j'utilise un logiciel générateur de nombres aléatoires (GNA), ces nombres calculés à partir de valeurs antérieures ne sont pas du tout aléatoires. J'ai donc cherché à réaliser un générateur de nombres vraiment aléatoires avec un circuit électronique.

## À PROPOS DU PROJET



ordinateur & internet

sécurité

cryptographie

jeux



débutant

connaissseur

→ expert



environ 4 h



soudage de CMS,  
programmeur AVR,  
alimentation variable



environ 150 €

Luka Matić (Croatie)

Plus tard quand j'ai appris les bases de la cryptographie, j'ai découvert d'autres bonnes raisons pour lesquelles il est utile d'avoir un générateur de nombres aléatoires (GNA). En fait il est impossible de crypter des données de manière sécurisée sans en utiliser un. Les applications de jeux et de paris nécessitent également un excellent GNA [2], et – pour les mathématiciens en herbe – il est possible d'estimer  $n$  en utilisant des nombres aléatoires. Pour savoir comment, cherchez *Buffon* et *Pi* sur l'internet.

Des générateurs de signaux aléatoires d'excellente qualité sont disponibles dans le commerce, mais à des coûts élevés, j'ai donc décidé de construire le mien en utilisant des composants bon marché et disponibles facilement.

## Circuit

Intéressons-nous au schéma de la figure 1.

Puisque le bruit est aléatoire par nature, une source de bruit est la base du GNA. Le caractère aléatoire de différents types de bruit électronique (grenaille, thermique, scintillation, en créneaux, avalanche) est bien connu. Mon GNA utilise le bruit d'avalanche – obtenu avec des diodes Zener – à cause de sa bande passante large et sa forte amplitude comparées à d'autres sources de bruit. Deux diodes Zener de 12 V (D7 et D10) produisent deux signaux de bruit. Un amplificateur différentiel bâti autour de T4, T5 et T6 amplifie la différence des deux signaux pour augmenter les chances d'avoir un signal de bruit fortement aléatoire. Cet amplificateur élimine aussi les signaux de mode commun dus aux interférences et perturbations qui peuvent affecter les deux diodes.



L'étage suivant (T7) amplifie encore le signal de bruit, jusqu'à 0,5 à 1 Vcc, niveau suffisant pour le comparateur analogique intégré au µC ATtiny2313. Il sert aussi de filtre correctif ajustable qui permet au GNA de satisfaire les critères de qualité des signaux aléatoires

définis par une organisation mondiale de standardisation (voir **encadré**). La sortie du comparateur analogique est échantillonnée à une fréquence d'env. 800 kHz. Par conséquent, certaines séquences du flux de données aléatoires peuvent être corrélées avec

des sous-harmoniques de la fréquence d'échantillonnage présente dans le signal de bruit. Les séquences de données 0x00 et 0xFF correspondent à un sous-harmonique de 50 kHz ; les séquences 0x0F et 0xF0 pour 100 kHz ; les séquences 0x33 et 0xCC

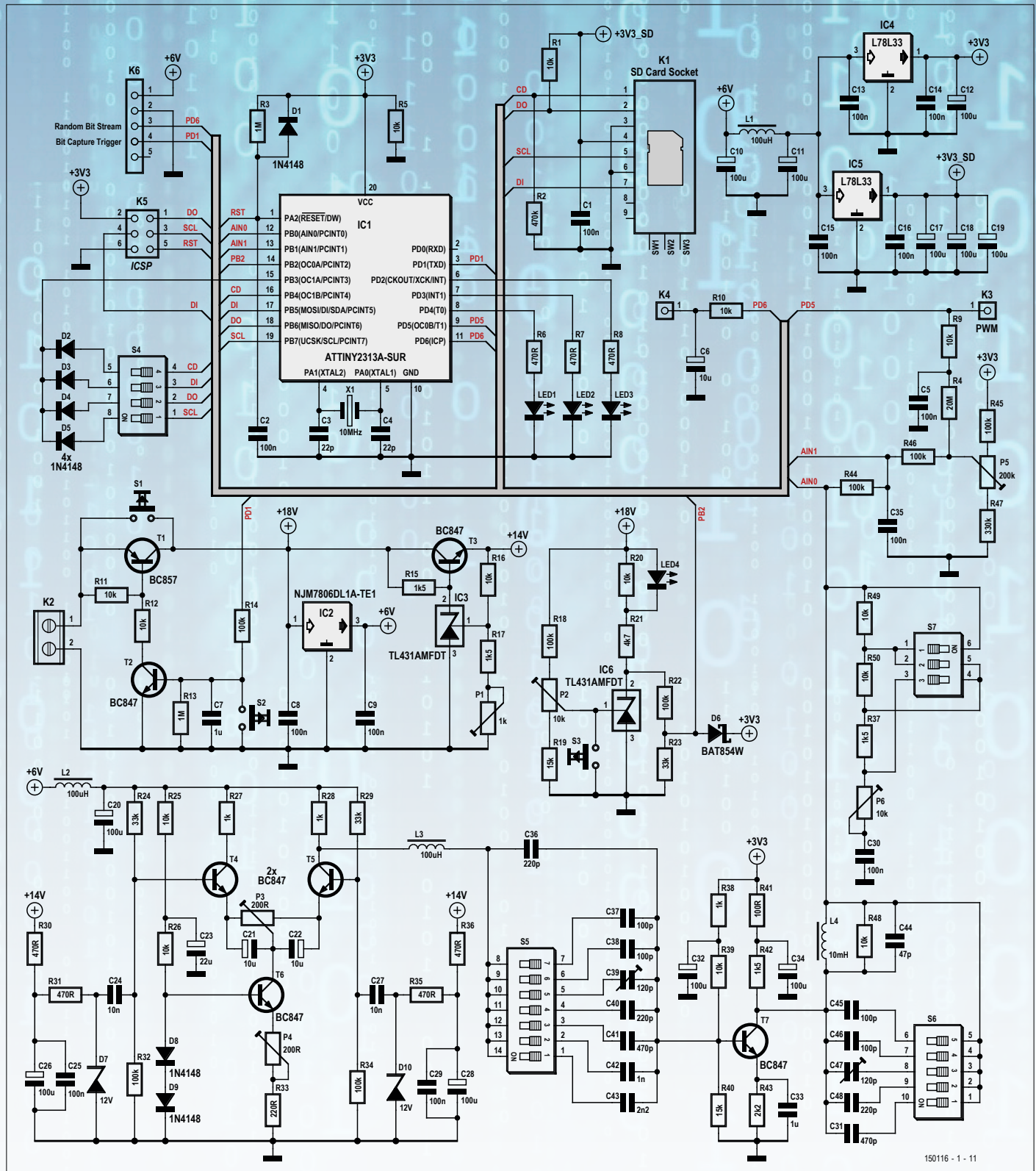


Figure 1. Le générateur de nombres aléatoires (GNA) combine habilement l'électronique analogique et l'électronique numérique pour produire un flux de données réellement aléatoires.



**Tableau 1.** Pour calculer la taille finale du fichier, additionnez 16 Mo et les valeurs de tous les commutateurs activés de S4. Si tous les commutateurs sont désactivés, la taille du fichier est de 16 Mo.

Commutateur	Taille du fichier
S4-1	2048 Mo
S4-2	1024 Mo
S4-3	512 Mo
S4-4	256 Mo

pour 200 kHz ; et les séquences 0x55 et 0xAA pour 400 kHz. Trop ou pas assez d'occurrences d'une certaine séquence de données indique que la réponse en fréquence du filtre correctif doit être corrigée afin de diminuer ou augmenter le gain de la fréquence « fautive ». C'est possible avec les commutateurs DIP S5, S6 et S7, et en ajustant C39, C47 et P6. La sortie échantillonnée du comparateur

analogique est enregistrée dans un grand fichier sur une carte SD formatée en FAT32 (voir **encadré**). Avant la mise sous tension du GNA, définissez avec le commutateur S4 la taille du fichier, entre 16 Mo et 3,8 Go (voir **tableau 1**). Après un enregistrement fructueux, un fichier nommé *random.hex* est créé sur la carte SD.

Le flux de données aléatoires est



## LISTE DES COMPOSANTS : GNA

### Résistances

Par défaut : 0805, 100 mW, 5%  
 R1,R5,R9,R10,R11,R12,R16,R20,R25,R26,R39,  
 R48,R49,R50 = 10 kΩ  
 R2 = 470 kΩ  
 R3,R13 = 1 MΩ  
 R4 = 20 MΩ  
 R6,R7,R8,R30,R31,R35,R36 = 470 Ω  
 R14,R18,R22,R32,R34,R44,R45,R46 = 100 kΩ  
 R15,R17,R37,R42 = 1,5 kΩ  
 R19,R40 = 15 kΩ  
 R21 = 4,7 kΩ  
 R23,R24,R29 = 33 kΩ  
 R27,R28,R38 = 1 kΩ  
 R33 = 220 Ω  
 R41 = 100 Ω  
 R43 = 2,2 kΩ  
 R47 = 330 kΩ  
 P1 = ajustable, 1 kΩ, 11 tours, CMS  
 P2,P6 = ajustable, 10 kΩ, 11 tours, CMS  
 P3,P4 = ajustable, 200 Ω, 11 tours, CMS  
 P5 = ajustable, 200 kΩ, 11 tours, CMS

### Condensateurs

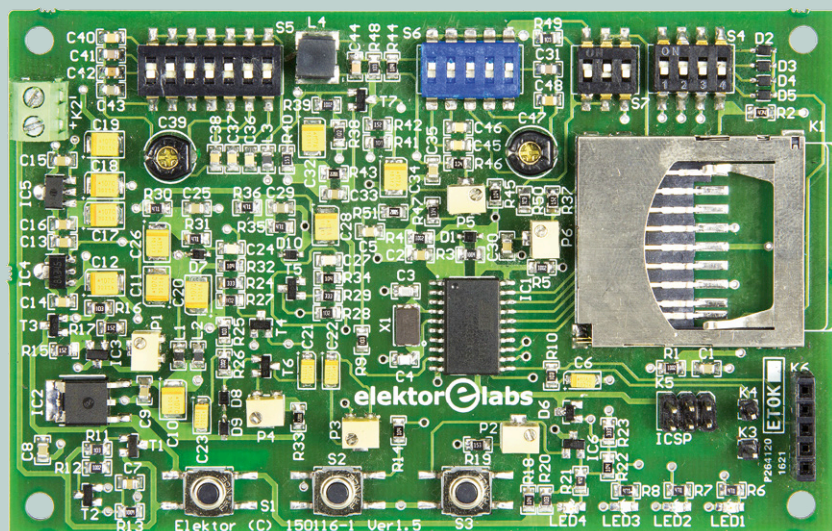
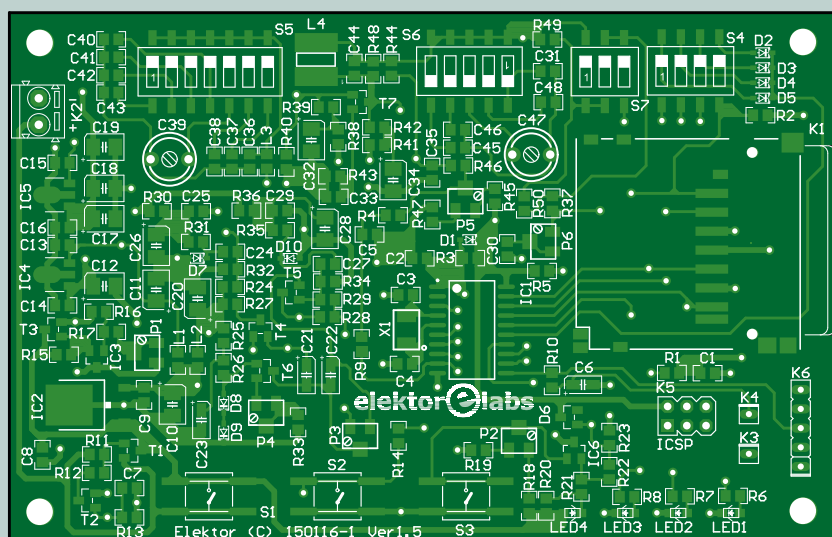
Par défaut : 0805  
 C1,C2,C5,C8,C9,C13,C14,C15,C16,C25,C29,  
 C30,C35 = 100 nF  
 C3,C4 = 22 pF  
 C6,C21,C22 = 10 µF, 6,3 V, boîtier A  
 C7,C33 = 1 µF  
 C10,C11,C12,C17,C18,C19,C20,C26,C28,  
 C32,C34 = 100 µF, 10 V, boîtier B  
 C23 = 22 µF, 10 V, boîtier A  
 C24,C27 = 10 nF  
 C31,C41 = 470 pF  
 C36,C40,C48 = 220 pF  
 C37,C38,C45,C46 = 100 pF  
 C39,C47 = 120 pF condensateur ajustable  
 (voir sur eBay)  
 C42 = 1 nF  
 C43 = 2,2 nF  
 C44 = 47 pF

### Inductances

L1,L2,L3 = 100 µH, 0805  
 L4 = 10 mH, 2220

### Semi-conducteurs

IC1 = ATtiny2313V-10SUR, programmé  
 IC2 = L7806CD2T-TR (ou NJM7806DL1A-TE1)



IC3,IC6 = TL431AMFDT  
 IC4,IC5 = L78L33ACUTR  
 D1,D2,D3,D4,D5,D8,D9 = 1N4148WS  
 D6 = BAT854W  
 D7,D10 = BZX384-C12  
 T1 = BC857CW  
 T2,T3,T4,T5,T6,T7 = BC847BW  
 LED1,LED4 = vert, 2×1,25 mm  
 LED2 = rouge, 2×1,25 mm  
 LED3 = jaune, 2×1,25 mm

### Divers

X1 = quartz à 10 MHz, 18 pF, 5,0×3,2 mm  
 S1,S2,S3 = interrupteurs tactiles

S4 = commutateur DIP, 4 positions  
 S5 = commutateur DIP, 7 positions  
 S6 = commutateur DIP, 5 positions  
 S7 = commutateur DIP, 3 positions  
 K1 = connecteur pour carte SD,  
 type SDBMF-00915B0T2  
 K2 = bornier à vis encartable à 2 pôles,  
 pas de 3,5 mm  
 K3,K4 = barrette mâle, 1 contact  
 K5 = barrette mâle, 6 contacts (2×3),  
 pas de 2,54 mm  
 K6 = barrette mâle, 5 contacts,  
 pas de 2,54 mm



www.elektormagazine.fr mars/avril 2017 71

## Toujours mieux...

Il est possible d'améliorer le degré aléatoire des données produites par le GNA : il faut effectuer un XOR (ou exclusif) bit par bit sur deux séquences enregistrées de même longueur. C'est similaire au cryptage d'une séquence par une autre, également appelé cryptage *One-Time Pad* (OTP). Le résultat est une séquence avec un histogramme beaucoup plus plat (c.-à-d. avec une variation de seulement  $\pm 2-3\%$ ) qui permet de satisfaire certains tests NIST pour des séquences beaucoup plus longues. Par ex. le test *runs* réussit pour des séquences de plus de 500.000 bits, contrairement aux 50.000 bits qui peuvent être atteints sans traitement ultérieur.

pour observer les réglages en temps réel peut vous venir en aide : l'analyseur de flux de données (AFD).

L'AFD est constitué d'un  $\mu\text{C}$  ATmega8 et d'un écran alphanumérique de  $2 \times 16$  caractères (**fig. 2**) connecté à K6

du GNA. En entrée, il utilise le flux de données aléatoires produit par le GNA sur PD6 ainsi que le signal de synchronisation de données sur PD1. C'est une sorte d'analyseur de spectre ajusté sur les quatre premiers sous-harmoniques de la fréquence d'échantillonnage (800 kHz) du GNA. Il affiche les amplitudes notées  $f_1$  (50 kHz),  $f_2$  (100 kHz),  $f_4$  (200 kHz) et  $f_8$  (400 kHz), et affiche le rapport 0/1.

Le  $\mu\text{C}$  est secondé par une sorte « d'ordinateur analogique » (IC2) qui détermine les niveaux des quatre sous-harmoniques parce que l'ATmega8 n'est pas assez rapide pour tout faire tout seul. Le  $\mu\text{C}$  s'occupe uniquement du rapport 0/1, et « soustraite » les sous-harmoniques à l'ordinateur analogique. Le  $\mu\text{C}$  délivre une impulsion courte de durée constante sur PB2 à PB5 pour chaque occurrence d'une séquence caractéristique (0x00, 0x0F, 0x33 ou 0x55) et lit la valeur analogique traitée sur PC0 à PC3.

Le nombre dans le coin supérieur droit de l'écran correspond au rapport 0/1 qui devrait être autour de 512. Il est contrôlé par P5 du GNA (non de l'AFD) et l'auto-offset de la MLI. Les quatre sous-harmoniques doivent avoir des amplitudes du même ordre (c.-à-d. dans  $\pm 10\%$ ), la valeur exacte n'a pas d'importance.

Pour utiliser l'AFD, il faut équilibrer les offsets en CC des amplificateurs opérationnels :

1. Mettez le cavalier JP1 (PC5 relié à GND), mettez sous tension ou faites une RàZ de l'AFD. Le  $\mu\text{C}$  passe en mode réglage de l'offset zéro, indiqué par le caractère « Z » sur l'écran.
2. Ajustez P2 à P5 afin de régler  $f_1$ ,  $f_2$ ,  $f_4$  et  $f_8$  à des valeurs un peu au-dessus de zéro, entre 10 et 20.
3. Ôtez le cavalier JP1, remettez à zéro l'AFD (ou mettez-le hors tension puis sous tension).

### Régler le GNA

Construire un circuit avec des composants bon marché implique en général le réglage manuel d'un ou de plusieurs paramètres. Le GNA en a plusieurs :

### Potentiomètre ajustable P1

Réglez afin d'obtenir 14 V sur l'émetteur de T3.



## LISTE DES COMPOSANTS : AFD

### Résistances

Par défaut : 0805, 100 mW, 5%

R1 = 1 k $\Omega$

R2, R4, R6 = 22 k $\Omega$

R3 = 100 k $\Omega$

R5, R7, R8, R9, R20, R21, R23, R24, R26, R27 = 10 k $\Omega$

R10, R19, R22, R25 = 470 k $\Omega$

R11, R12, R13, R14, R15, R16, R17, R18 = 1 M $\Omega$

P1 = 10 k $\Omega$ , ajustable, CMS

P2, P3, P4, P5 = ajustable, 200 k $\Omega$

### Condensateurs

Par défaut : 0805

C1, C2, C23 = 22 pF

C3, C4, C5, C6, C7, C10, C13, C16, C19 = 100 nF

C8 = 100  $\mu\text{F}$ , 16 V, radial CMS

C9, C12, C15, C18 = 22  $\mu\text{F}$ , 16 V, radial CMS

C11, C14, C17, C20 = 1  $\mu\text{F}$

C21, C22 = 10  $\mu\text{F}$ , 25 V, radial CMS

### Inductance

L1 = 10  $\mu\text{H}$ , 0805

### Semiconducteurs

D1, D2 = 1N4148WS

D3 = MRA4007T3G

IC1 = ATmega8-16AU

IC2 = LM324MX

IC3 = TL7660CD

### Divers

X1 = quartz à 16 MHz, 18 pF, 5 $\times$ 3,2 mm

JP1 = barrette mâle, 2 contacts, pas de 2,54 mm

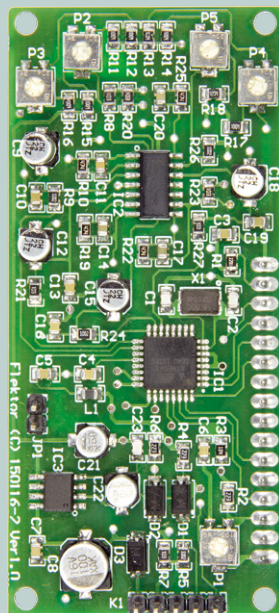
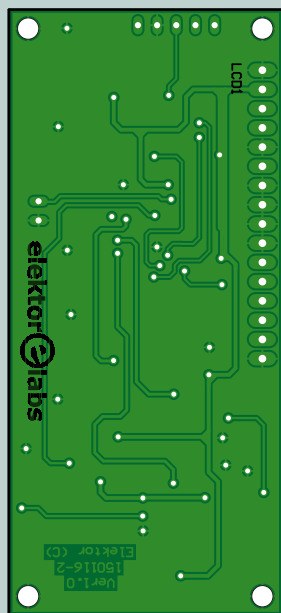
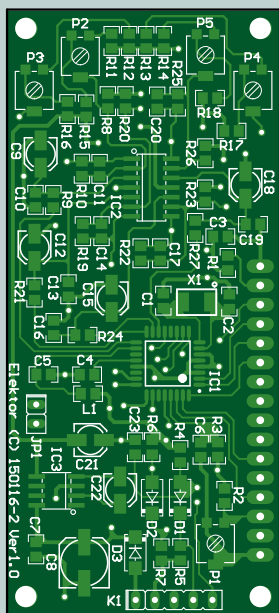
Cavalier, pas de 2,54 mm

K1 = barrette femelle, 5 contacts, pas de 2,54 mm (côté soudure)

LCD1 = alphanumérique, 16 col., 2 lignes

Barrette mâle, 16 contacts (pour l'écran LCD), pas de 2,54 mm

Barrette femelle, 16 contacts (pour l'écran LCD), pas de 2,54 mm





### Potentiomètre ajustable P2

Détermine la tension qui déclenche l'alarme *piles faibles* (LED4 éteinte). Si le GNA utilise deux accus NiMh de 9 V, l'alarme se déclenche à 15,4 V (ces accus contiennent des cellules NiMh de 1,2 V ; quand chaque cellule est déchargée à 1,1 V, la tension totale est de 15,4 V). Connectez une alimentation variable sur K2 ; faites varier la tension entre 15 V et 16 V. Réglez P2 afin d'éteindre la LED4 quand la tension sur K2 chute à 15,4 V.

### Potentiomètres ajustables P3 et P4

Le courant collecteur des transistors T4 et T5 doit être de 1 mA. Réglez P3 (symétrie) et P4 (courant total) afin d'obtenir une chute de tension de 1 V aux bornes des résistances R27 et R28.

### Potentiomètre ajustable P5

1. Coupez la tension (appuyez sur S2, arrêt matériel).
2. Restez sur S3 (arrêt en douceur), puis appuyez sur S1 (départ) afin d'accéder au mode *test*. La LED3 s'allume.
3. Connectez un oscilloscope (ou un voltmètre en continu) à PD5 afin de surveiller le signal MLI qui compense la dérive en continu. Réglez P5 afin de stabiliser le signal MLI vers 50%.
4. Le mode *test* s'exécute pendant env. 40 s, puis la valeur MLI est écrite dans l'EEPROM interne au µC et la LED1 s'allume. Relancez le mode *test* s'il vous faut plus de temps. Si la MLI sature à 0 ou 100% la LED2 s'allume et la valeur MLI n'est pas écrite dans l'EEPROM. Si cela se produit, relancez le mode *test*.

### S5, S6 et S7

La **figure 3** montre la fonction de transfert du circuit autour de T7. S5 avec son condensateur contrôle la fréquence de coupure basse. Régler S6 afin d'ajuster la fréquence de résonance du filtre comme suit :

$$f_{\text{res}} = 1 / (2\pi \times \sqrt{L4 \times C_{S6}}) \text{ [Hz]}$$

où S7 positionne le pic maximal à :

$$20 \times \log_{10}(R_{S7} / (R_{S7} + R48)) \text{ [dB]}.$$

### Autoréglages

La dernière valeur MLI est stockée dans l'EEPROM interne. Pendant l'enregistrement normal des données aléatoires, le rapport cyclique de la MLI est ajusté en permanence. En fin

d'enregistrement (arrêt normal ou alarme *piles faibles*), la valeur actuelle du rapport cyclique de la MLI est écrite dans l'EEPROM afin d'être utilisée à la prochaine mise sous tension du GNA.

Recommandation : enregistrez d'abord un petit fichier de 16 Mo (et effacez-le) avant d'enregistrer un fichier à la taille désirée afin que le µC règle le signal MLI. Ensuite, choisissez la taille désirée

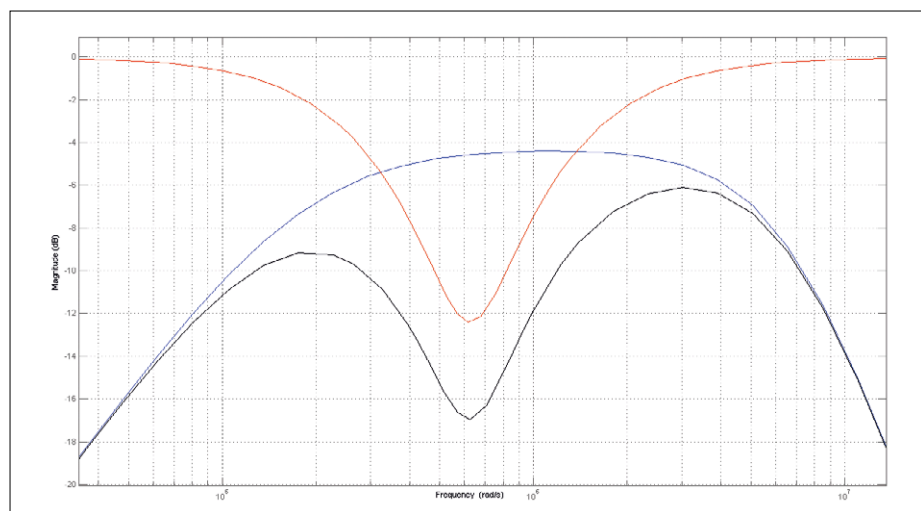


Figure 3. Simulation de la fonction de transfert du filtre construit autour de T7 (courbe noire). La courbe bleue est la fonction de transfert de T7, définie par sa bande passante. La fréquence de coupure basse est contrôlée par S5 et ses condensateurs C36 et C43. La courbe rouge correspond à la fonction de transfert du filtre correctif L4-R48-CS6-RS7. Pour être plus clair, on néglige le gain du transistor, les courbes réelles noire et bleue se situent 30 dB plus haut.

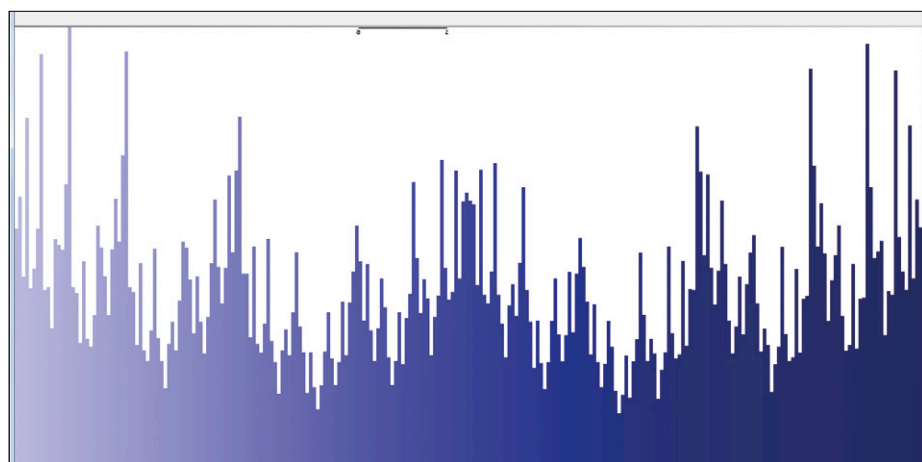


Figure 4. L'histogramme des données créées par le GDA montre un filtre correctif mal réglé. Si l'on regarde uniquement les barres à l'extrême gauche et à l'extrême droite (données respectives de 0x00 et 0xFF) le rapport 0/1 semble bon, toutefois comme le filtre correctif n'est pas bien réglé, les occurrences de 0x0F et 0xF0 sont trois fois plus fréquentes que celles de 0x55 et 0xAA.

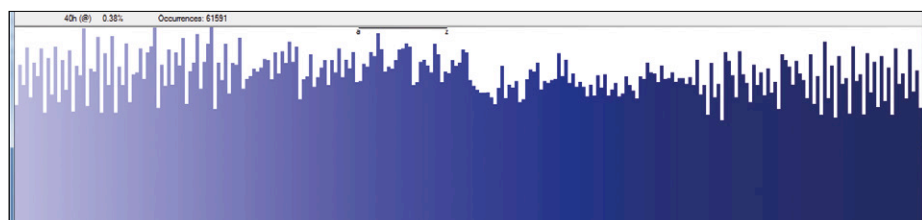


Figure 5. Après un réglage précis du filtre correctif, l'histogramme est presque plat. À présent vous avez des données aléatoires d'excellente qualité.

## Quelques astuces pour la carte SD

Le code (assembleur) pour l'ATtiny2313 ne travaille qu'avec des cartes SDHC formatées en FAT32 : pas de FAT16, ni de cartes SD autres que SDHC. Il ne supporte pas les cartes SDHC de taille supérieure à 4 Go parce que ses pointeurs à 32 bits ne peuvent adresser que 4 Go. Ce n'est pas un problème, car le FAT32 ne supporte pas de fichier supérieur à 4 Go et que de toute façon l'écriture de tels fichiers prendrait trop de temps.

Comme un pilote FAT32 sachant faire tout et plus encore consommerait trop de mémoire de programme de notre petit (*tiny*)  $\mu$ C, seules les opérations de base FAT32 nécessaires à l'écriture d'un simple fichier dans le dossier racine sont implémentées. À cause de cette simplicité et de certaines mesures de sécurité pour gérer correctement l'arrêt en douceur, la taille du fichier final n'est pas exactement celle annoncée. Il est recommandé de formater la carte SD avec des clusters de 8 Ko pour deux raisons :

- Plus la taille du cluster est petite, plus la mise à jour de la table FAT sera fréquente (toutes les 2 s avec des secteurs de 1 Ko). Toutefois le signal vers l'AFD est arrêté pendant la mise à jour de la table FAT, ce qui entraîne des sauts de lecture de l'analyseur parce que ses condensateurs d'entrée se déchargent. Si cela arrive trop souvent, le réglage du GNA sera difficile à cause des lectures fluctuantes.
- Plus la taille du cluster est grande, plus le temps entre les mises à jour de la table FAT est long (plus de 60 s pour des secteurs de 32 Ko). La correction de l'auto-offset de la MLI est exécutée à chaque mise à jour de la table FAT, et si le temps entre les mises à jour est trop long, la correction de l'auto-offset n'est plus efficace.

avec le commutateur DIP S4 et relancez le GNA immédiatement. Les données aléatoires auront maintenant une meilleure qualité parce que l'offset en continu sera correctement réglé dès le début.

### Test de bon fonctionnement

Après les réglages, enregistrez un petit fichier (16 Mo) et analysez-le avec un programme tel que WINHEX afin de créer un histogramme des occurrences des 256 séquences possibles de 8 bits. La **figure 4** montre un histogramme de données non uniforme avec des pics insuffisamment aléatoires pour un cryptage de qualité, même si le rapport 0/1 est correct. Avec des réglages plus précis, on obtient un histogramme comme celui de la **figure 5**.

### Attention aux interférences électromagnétiques

Nota : sans un boîtier adéquat, des instruments très précis comme le GNA sont sensibles aux rayonnements électromagnétiques et RF. Recommandation importante : montez le GNA dans un boîtier métallique (aluminium) avec une connexion correcte de la masse du circuit au boîtier.

### Tests NIST

Les normes d'évaluation de la qualité d'un flux de données aléatoires ont été définies par des instituts de standardisation comme le *National Institute Standards and Technology* (NIST). Le NIST a défini quinze tests du caractère aléatoire, résumés en trois règles simples :

1. un flux de données vraiment aléatoires comprend un nombre égal de zéros et de uns ;
2. un flux de données vraiment aléatoires comprend des fréquences d'oc-

currences égales pour des séquences de données différentes ;

3. aucune partie du flux de données n'est mathématiquement corrélée avec une autre partie du flux.

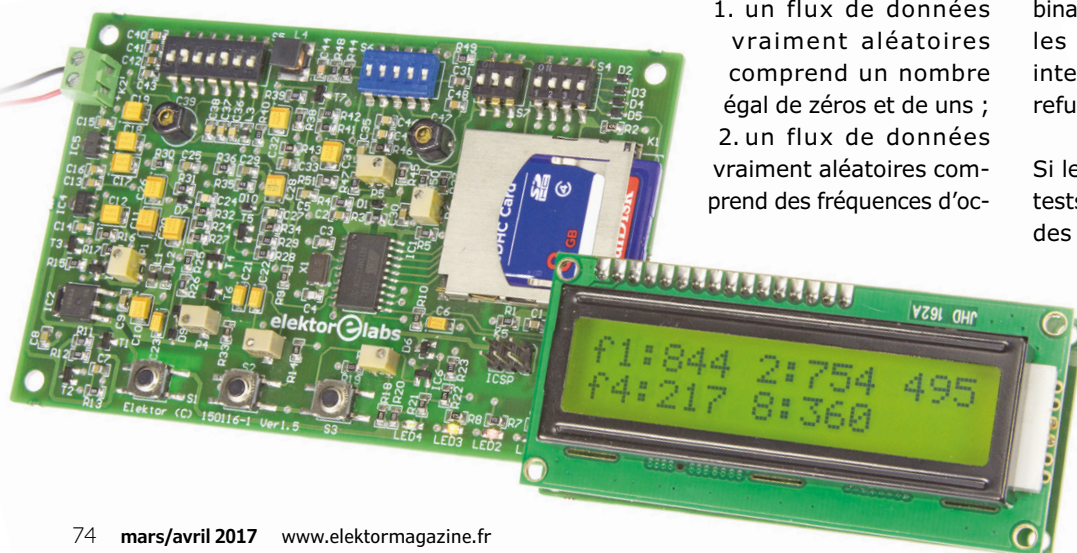
Quand ses réglages sont corrects pour produire des histogrammes comme à la **figure 5**, notre GNA réussit les quinze tests NIST. Le 1<sup>er</sup> critère est satisfait en réglant correctement la polarisation en continu à l'aide de P5. Le 2<sup>e</sup> critère est rempli en réglant la réponse en fréquence du filtre correctif avec ses condensateurs et résistances (DIP S5, S6 et S7) et ajustables C39, C47 et P6. Si la source de flux de données aléatoires est un bruit aléatoire, le 3<sup>e</sup> critère est automatiquement satisfait.

Avant de passer aux tests NIST, faites un test simple avec un utilitaire de compression comme 7-Zip ou similaire. Les algorithmes de compression de fichiers essaient de réduire la redondance ou corrélation des données pour diminuer leur volume. Comme les données réellement aléatoires ne sont pas du tout corrélées, elles ne peuvent pas être compressées. Dans ce cas, le fichier compressé sera plus gros que l'original ; l'embonpoint est dû à des questions de format de fichier.

Vous trouverez en [2] une application Windows qui exécute les tests NIST, des générateurs de nombres pseudo-aléatoires (PRNG) qui produisent un bon flux de données pseudo-aléatoires à des fins de comparaison, et des échantillons de flux de données.

L'application de test NIST nécessite un fichier texte en entrée, il faut donc convertir les fichiers binaires. L'application est assez sensible sur le format exact du fichier d'entrée – seuls les caractères ASCII « 0 » (code ASCII 0x30), « 1 » (0x31), et l'espace (0x20) sont autorisés. Tous les caractères binaires seront séparés par des espaces ; les espaces de début et de fin sont interdits. Une seule erreur suffit pour refuser le fichier.

Si le flux de données échoue à certains tests, c'est toujours en rapport avec l'un des trois critères énoncés ci-dessus, et un réglage supplémentaire du GNA devrait résoudre le problème. Par ex. si un flux de données réussit le test *monobit* (un simple test de rapport 0/1), mais échoue au





test *runs* (c.-à-d. comptage du nombre de *runs* ou séquences de tous les uns ou tous les zéros) à cause d'une valeur excessive, alors la fréquence analogique moyenne est un peu trop haute. Diminuez alors légèrement la fréquence de coupure basse (en augmentant la valeur du condensateur sélectionné avec S5).

Avec quelques réglages des ajustables, il est facile de faire passer ces quinze tests à des flux de données de longueur beaucoup plus grande que le minimum requis.

## Conclusion

Réalisé avec des composants bon marché et faciles à trouver, notre générateur de nombre aléatoire (GNA) est suffisamment bon pour satisfaire les normes sur les données réellement aléatoires édictées par des organisations comme le NIST. L'approche *fais-le toi-même* est payante – dans le pur esprit du magazine et du labo d'Elektor ! Ce GNA est plus qu'un jouet expérimental, c'est un instrument de précision. Toutefois des systèmes de chiffrement qui l'utiliseraient (ou qui utilisent de meilleurs GNA) peuvent, en théorie, être percés avec un ordinateur quantique. Donc ni Elektor ni l'auteur



## L'auteur

Luka Matić est diplômé du FER Zagreb (Croatie) où il a obtenu un master d'ingénierie en automatisme. Son expérience professionnelle comprend aussi bien l'automatisation de sites industriels (à terre et en mer), ainsi que de véhicules submersibles pilotés à distance (ROSV). En tant qu'ingénieur indépendant, il connaît toutes sortes de composants et d'appareils électroniques. Luka est auteur pour le magazine et le labo d'Elektor, mais également coauteur d'un article scientifique sur la compensation de la friction pour les systèmes de positionnement commandés électriquement. Écrivez-lui : luka.matic@gmail.com

ne peuvent, ni ne pourront, être tenus responsable de la confidentialité de vos données – vous êtes prévenus.

(150116 – version française : Patrick Bechler)

## Liens

- [1] [http://csrc.nist.gov/groups/ST/toolkit/rng/documentation\\_software.html](http://csrc.nist.gov/groups/ST/toolkit/rng/documentation_software.html)
- [2] [www.elektormagazine.fr/labs/random-number-generator-150116](http://www.elektormagazine.fr/labs/random-number-generator-150116)
- [3] [www.elektormagazine.fr/150116](http://www.elektormagazine.fr/150116)

**DANS L'E-CHOPPE**

- 150116-1  
Circuit imprimé
- 150116-2  
Circuit imprimé
- 150116-41  
Microcontrôleur programmé
- 150116-42  
Microcontrôleur programmé

Publicité

# EX PÉRIENCE, C OMPÉTENCE, I DÉES, T ALENT & E NVIES

**Vous avez de l'expérience, des compétences, des idées, du talent et des envies d'aller de l'avant dans la jungle des réseaux et des protocoles ?**

Les fauves et les moustiques appelés *LoRa*, *SigFox*, *Neul*, *Qowisio*, *ThingSpeak*, *Brillo*, *Nest Weave*, *Thread*... ne vous effrayent pas ?  
Ce ne sont pas ces quelques réseaux à bas débit sur longue distance qui vous font perdre le nord ?

Pour vous orienter dans le monde de l'Internet des Objets, vous avez déjà pratiqué une ou plusieurs cartes d'expérimentation : Elektor vous invite à partager vos connaissances maintenant en publiant **votre livre !**

Forts de notre expérience d'éditeur, nous vous accompagnerons dans l'écriture de votre ouvrage et nous chargerons de la mise en page, de l'impression, de la commercialisation en librairie et bien sûr sur l'internet (notamment chez Amazon).

La relation éditeur-auteur est encadrée par un contrat d'édition qui stipule vos droits d'auteur sous la forme d'un pourcentage déterminé du prix de vente.

Présentez-nous votre projet en nous envoyant une description et un projet de sommaire par courriel à **redaction@elektor.fr**.





# projet 2.0

## corrections, mises à jour et courrier des lecteurs



### préamplificateur SUPRA 2.0

Elektor 6/2016, p. 54 (150616)

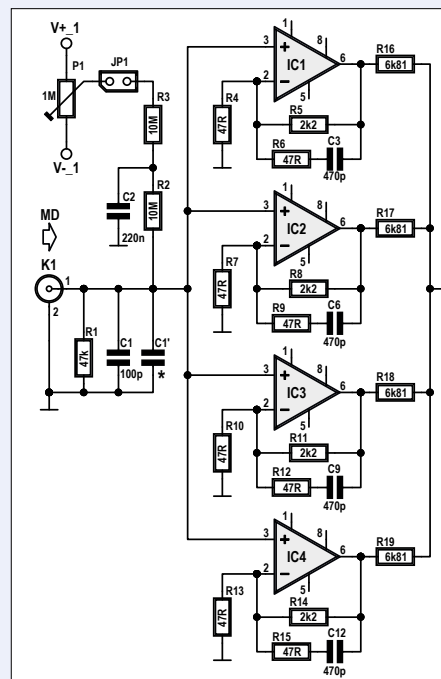
Pour obtenir le meilleur rapport signal/bruit avec un phonocapteur MM, je recommande fortement de n'utiliser qu'un seul ampli op LT1028 dans l'étage d'entrée. Les phonocapteurs MM ont une impédance d'entrée de quelques centaines à quelques milliers d'ohms, qui dépend de la fréquence. Avec un courant d'entrée de bruit de 1 pA/√Hz, le bruit correspondant dépasse le bruit de la tension d'entrée. Avec quatre amplis ops LT1028 et une impédance de capteur de 1 kΩ, le bruit dû au courant serait de 2 nV/√Hz. Pour une version à un seul ampli op, la résistance R16 devrait être réduite à 1,7 kΩ. Les phonocapteurs MC sont à basse impédance, ce qui donne un sens à la configuration à quatre amplis ops.

Andreas von Ow

Bonjour M. von Ow

Vous avez entièrement raison, mais ici, il s'agissait de concevoir un préamplificateur qui donne de bons résultats pour les deux types de phonocapteurs MC et MM. Nous n'avons pas trouvé d'autres types d'amplis ops qui fournissent pour les cellules MM et MC un meilleur compromis pour à la fois la tension et le courant de bruit.

Harry Baggen et Ton Giesberts



### carte de commande Wi-Fi : le retour

Elektor 11/2016, p. 32 (150402)

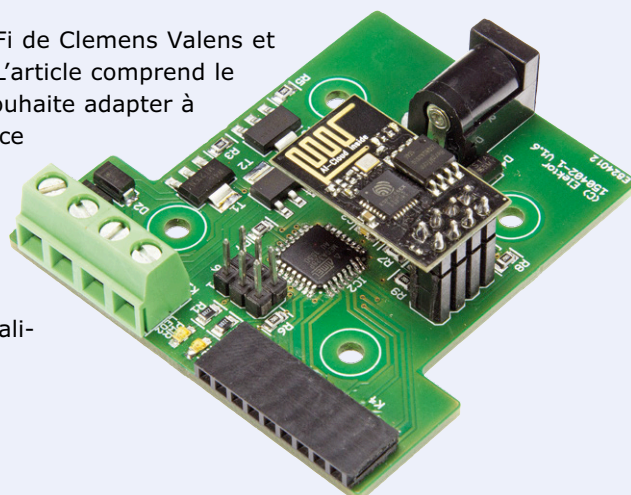
Ma question concerne la carte de commande Wi-Fi de Clemens Valens et Roy Aarts, que par ailleurs, je trouve excellente. L'article comprend le croquis Arduino « wifi\_controller\_v1.3.ino » que je souhaite adapter à mon application. Comme je n'ai encore aucune expérience de programmation in situ de croquis Arduino, je vous demande conseil. Je pense acheter le module Arduino bon marché « Arduino ISP », utilisable avec la version 1.5 et suivantes d'Arduino. On le monte sur les broches ISP de la carte cible (ici K6) et, après un paramétrage adéquat par *tools/board*, on peut lui transférer le croquis par USB. À quoi faut-il prendre garde, à la tension d'alimentation de 3,3 V de la carte par ex. ? Ou bien y a-t-il une meilleure méthode ?

Herbert Nitsche

Bonjour M. Nitsche

La carte de commande Wi-Fi est alimentée en 3,3 V alors que l'Arduino ISP est conçu pour des systèmes en 5 V. Néanmoins, il est possible de l'utiliser : il ne faut pas relier l'alimentation de l'Arduino ISP au connecteur ISP (K6) de la carte de commande Wi-Fi (il y a pour cela un cavalier soudable sur l'Arduino ISP). Vous devrez ensuite alimenter la carte sous sa tension normale. En outre, c'est peut-être une bonne idée d'insérer des petites résistances de 100 Ω en série dans les lignes MOSI, MISO, SCK et RESET, même si ce n'est sans doute pas indispensable.

Clemens Valens







## hors-circuits - le filtrage numérique sans stress - filtres FIR

Elektor 01/2016, p. 98 (150635)

Votre article sur la théorie des filtres FIR m'a beaucoup intéressé. Magnifique article de Robert Lacoste ! Un tuyau : le regretté Tony Fisher, hélas trop tôt disparu, a programmé un outil en ligne génial de conception des filtres numériques, que fort heureusement l'université de York maintient toujours en activité : <https://www-users.cs.york.ac.uk/~fisher/mkfilter/>

Comme résultat, on obtient du code source C qu'on peut utiliser pour réaliser non seulement des filtres FIR, mais aussi des filtres IIR de n'importe quel ordre.

Matthias Weingart

**Butterworth / Bessel / Chebyshev Filters**

This is an interactive filter design package. For designing digital filters by the bilinear transform or matched z-transform method. Fill in the form and press the "Submit" button, and a filter will be designed for you.

1. Select filter type:
 

<input checked="" type="radio"/> Butterworth	<input checked="" type="radio"/> Lowpass
<input type="radio"/> Bessel	<input type="radio"/> Highpass
<input type="radio"/> Chebyshev	<input type="radio"/> Bandpass
	<input type="radio"/> Bandstop
2. If you specified "Chebyshev" above, enter ripple in dB here:  (The ripple, if specified, must be a negative number. For other filter types, leave this field blank.)
3. Enter the filter order:  (For lowpass and highpass, this is the number of poles. For bandpass and bandstop, the number of poles is twice the order.)
4. Sample rate, in samples per second:
5. Enter corner frequencies, in Hz.
 

Corner frequency 1: <input type="text"/> Hz
Corner frequency 2: <input type="text"/> Hz

(For Butterworth and Bessel lowpass designs, the corner frequency is the frequency at which the magnitude of the response is -3 dB. For Chebyshev lowpass designs, the corner frequency is the highest frequency at which the magnitude of the response is equal to the specified ripple. For highpass, bandpass and bandstop, the above definition is modified in an obvious way.)

(For lowpass and highpass, one corner frequency is required; enter this in the first slot and leave the second one blank. For bandpass and bandstop, two corner frequencies are required.)

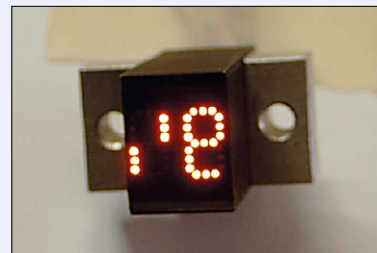


## bienvenue dans la section Partager - Nixie « Canada Dry », épisode 3

Elektor 12/2016, p. 68 (160223)

Cette idée n'est pas neuve. Des afficheurs de ce genre existaient déjà il y a plus de 40 ans dans les environnements à forte luminosité ambiante, comme les postes de pilotage des avions. Les sources de lumière étaient des petites lampes à incandescence spéciales. La photo montre l'un de ces afficheurs, qui n'est malheureusement plus totalement en ordre de marche.

Max Hiltmann



## nombres aléatoires en analogique

Elektor 12/2016, p. 33 (150831)

Pour contribuer au succès de l'ésotérisme (plus les mesures sont précises, moins on observe d'événements positifs) avec cette méthode de mesure déjà vieille de 25 ans, il faut prendre garde aux points suivants :

1. Tous les amplificateurs, comparateurs et surtout la diode Zener génératrice de bruit doivent être alimentés par des tensions produites à partir du réseau 50 Hz, pour garantir que le bruit contient bien un minuscule ronflement à 100 Hz (les harmoniques sont un avantage).

2. Échantillonner le mélange bruit-ronflement à un multiple entier de la fréquence du réseau (à la précision du quartz, ou mieux, d'une horloge atomique).

On a réalisé ainsi un beau redresseur synchrone (*Lock In Amplifier*). Si l'on échantillonne toujours au milieu de l'alternance positive (invisible) du ronflement, on obtient déjà un minuscule écart positif. Il en est de même pour l'alternance négative. Si l'on observe la phase/fréquence de notre réseau 50 Hz pendant 24 h, on constate que ces valeurs ne sont pas constantes, mais varient en fonction de la charge du réseau. L'horloge réseau retarde le jour et rattrape son retard la nuit. S'il y a des soi-disant « événements » comme une finale de foot, les compagnies d'électricité doivent faire tourner leurs alternateurs plus longtemps et à plus forte charge et ne peuvent rattraper le retard de phase qu'après le coup de sifflet final. Ce processus est observable sous forme d'un minuscule saut à la sortie de l'appareil de mesure ésotérique. Si vous voulez en savoir plus sur l'ésotérisme, vous devriez consulter la page d'accueil de la GWUP.

Heinz Lenk

N.d.T. – GWUP : *Gesellschaft zur wissenschaftlichen Untersuchung von Parawissenschaften*, société pour l'étude scientifique des sciences parallèles. Équivalent français : Laboratoire de zététique, [www.unice.fr/zetetique](http://www.unice.fr/zetetique).

Cher Monsieur Lenk

Merci pour votre lettre de lecteur pleine d'humour, dont l'ironie sous-jacente nous a bien fait sourire. Vous avez tout à fait raison, la recherche d'une « conscience globale » peut rapidement déboucher sur des dérapages ésotériques. Nous pouvons vous assurer que l'ésotérisme n'est pas notre tasse de thé et que notre seule intention était de présenter à nos lecteurs un circuit utilisable dans beaucoup d'applications. Comme vous le savez certainement, les signaux de bruit interviennent dans beaucoup de domaines de l'électronique. De plus, le croquis pour Arduino que nous proposons peut servir à l'enregistrement d'autres données (par ex. des températures).

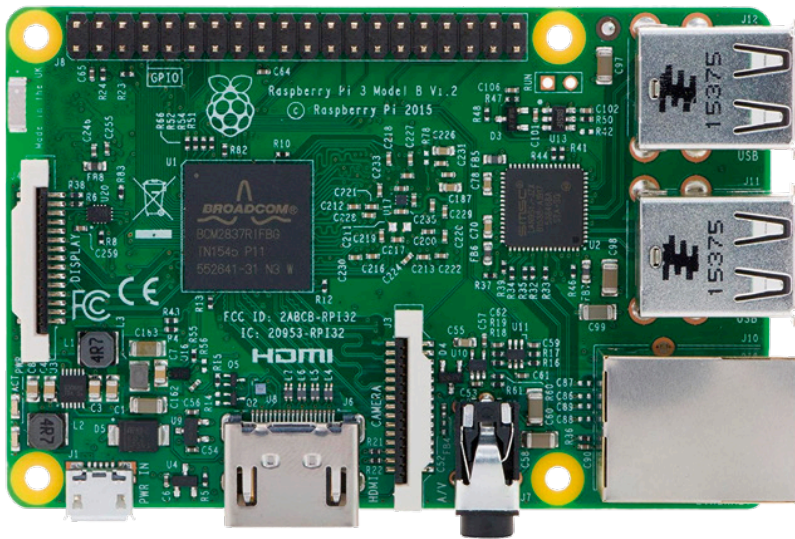
Avec notre référence au projet de Princeton (GPC), qui existe réellement, nous ne prétendons nullement cautionner les vues de ses promoteurs, nous voulons simplement offrir au lecteur la possibilité de se faire sa propre opinion.

Elektor a depuis toujours malicieusement pratiqué l'électronique parascientifique ; le sujet eut même droit jadis à un numéro spécial parce que l'éditeur était d'avis que même les électroniciens amateurs avaient de l'humour et ne prenaient pas tout avec un sérieux imperturbable.

Kurt Diedrich

# Android sur Raspberry Pi (1)

mesurer, commander, réguler : les broches GPIO



Google sort une nouvelle version d'Android, spécialement conçue pour les ordinateurs monocarte ainsi que la mesure, la commande et la régulation. Nous avons testé Android sur le Raspberry Pi et nous vous montrons pas à pas comment réaliser un premier projet.

**Tam Hanna** (Slovaquie)

Les progrès rapides de Microsoft dans le domaine du Raspberry Pi ont certainement dû donner des maux de tête à Google : après l'ordinateur de bureau, la tablette et l'ordiphone, l'Internet des

Objets est bien parti pour devenir le prochain champ de bataille.

Convergence, c'est le maître-mot – Google est contraint de se lancer dans le domaine de l'Internet des Objets pour ne pas laisser à Microsoft la liberté d'en profiter pour une nouvelle attaque du marché des portables.

Le système « Brillo », annoncé il y a un an, a eu peu de succès. Pour ce deuxième essai, on ne fait pas dans le détail et on appelle le nouveau produit simplement « Android Things » (en français *Objets Android*).

## Accès plus simple au matériel

Commençons par le plus évident. Malgré tout le tapage – les analogies avec « Windows 10 IOT Core » ne sont que

des coïncidences – *Android Things* n'est en gros qu'une version standard d'Android, avec une extension pour l'interaction avec le matériel. La **figure 1** montre la structure du système d'exploitation.

La bibliothèque « Things Support Library » se compose de deux modules. Le premier est l'Interface de Programmation des Périphériques (*Peripheral I/O API*) qui, à l'heure où nous écrivons ces lignes, offre un accès aux périphériques suivants (il n'est pas clair si *OneWire* sera ajouté par la suite) :

- GPIO avec MLI
- I<sup>2</sup>C
- SPI
- UART

## Attention à la qualité !

Si vous achetez des cartes en Chine, soumettez-les à des contrôles rigoureux. Beaucoup de fabricants vendent des lots de matériel de mauvaise qualité qui annoncent au contrôleur la capacité nominale, mais qui en pratique perdent des données.



Le deuxième est l'Interface de Programmation des Pilotes Utilisateur (*User Driver API*). Il s'agit d'une interface de programmation qui permet aux développeurs de mettre à la disposition du reste du système les données fournies par leurs capteurs maison. Vous trouverez en [1] davantage d'informations sur les modules contenus dans la bibliothèque *Things Support Library*. Il est temps de nous lancer dans nos premiers essais avec le matériel.

## Images complètes

À l'heure où nous rédigeons cet article, Google, qui a apparemment tiré les leçons de l'échec de Brillo, propose des images d'*Android Things* pour les plateformes Edison, NXP Pico et Raspberry Pi 3. Ici, nous allons nous intéresser au Raspberry Pi 3, de loin le plus répandu. Pour commencer, téléchargez l'image disponible en [2] et copiez-la sur une carte SD selon la procédure habituelle (voir aussi l'**encadré** « Attention à la qualité »). Dans la suite, l'auteur utilise une carte de 8 Go. L'utilisation de cartes de plus grande capacité ne devrait pas poser de problèmes, alors que la taille de l'image, à savoir 4,6 Go, interdit l'usage de cartes plus petites.

Raccordez ensuite l'écran et un câble réseau au RPi et démarrez en mettant sous tension (le clavier et la souris ne sont pas nécessaires, car *Android Things* n'affiche qu'un écran de bureau sans possibilité d'interaction directe). Nous n'insisterons pas sur le débogage en mode liaison Wi-Fi, parce que la latence plus élevée due à la liaison sans fil peut introduire des délais parasites dans la chasse aux erreurs. Si vous voulez néanmoins vous y risquer, suivez les instructions de connexion au réseau Wi-Fi en [3]. Même sur un Raspberry Pi 3, le premier démarrage d'*Android Things* peut prendre une minute ou deux. Tout au début apparaît un message d'absence d'Ethernet, qu'on peut ignorer si le câble réseau est effectivement branché.

Ceci fait, le système affiche l'écran de démarrage représenté sur la **figure. 2**. L'adresse IP – ici 10.42.0.44 – va nous être utile dans un instant.

## Environnement de développement

Ne serait-ce que par manque de place, il nous faut supposer ici que vous disposez sur votre PC de développement d'un environnement *Android Studio* opé-

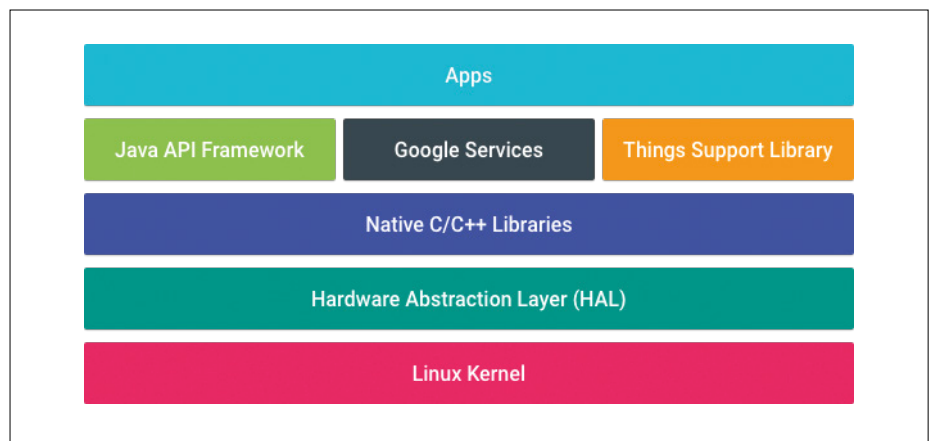


Figure 1. *Android Things* est une variante d'Android (source : Google).

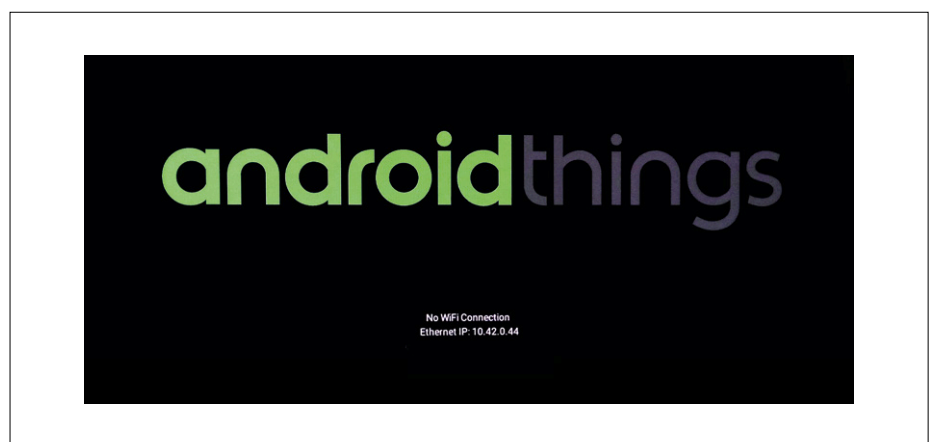


Figure 2. Notre *Android Thing* est prêt à l'emploi – notez son adresse IP.

rationnel, sinon, vous trouverez conseils et soutien en [4]. Pour les étapes suivantes, l'auteur utilise une station de travail AMD à 8 cœurs sous Ubuntu 14.04 ; sous Windows et Mac Os, cela ne devrait être guère différent.

Prochaine étape : positionnez-vous sur le PC de développement sous le répertoire racine de la Passerelle de Débogage d'Android (*Android Debug Bridge, ADB*). Puis connectez-vous au RPi au moyen des commandes de console (en gras) suivantes :

```
tamhan@TAMHAN14:~/Android/Sdk/
platform-tools$ ./adb connect
10.42.0.44
```

```
connected to 10.42.0.44:5555
```

```
tamhan@TAMHAN14:~/Android/Sdk/
platform-tools$ ./adb devices
```

```
List of devices attached
```

```
10.42.0.44:5555 device
```

```
tamhan@TAMHAN14:~/Android/Sdk/
platform-tools$
```

La commande `adb devices` n'est pas absolument indispensable, nous la mentionnons ici parce qu'elle permet de lister tous les périphériques connectés à la passerelle.

Notez qu'une connexion entre ADB et RPi peut être perdue, par exemple si le PC passe en veille.

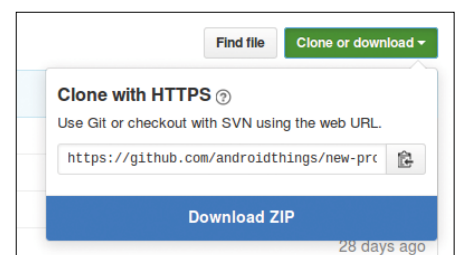


Figure 3. GitHub est et reste une pénibilité.

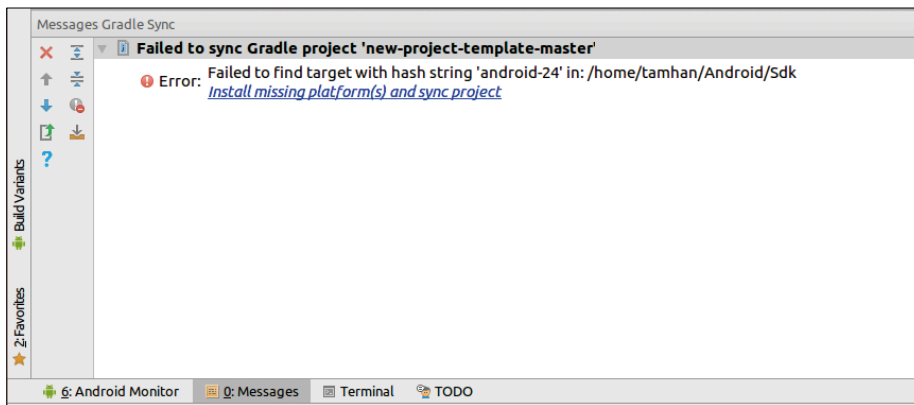


Figure 4. Ici, il manque un kit de développement logiciel compatible.

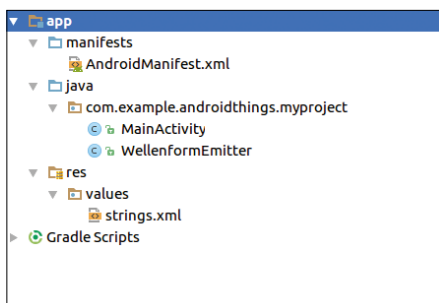


Figure 5. Cette application se passe de XML.

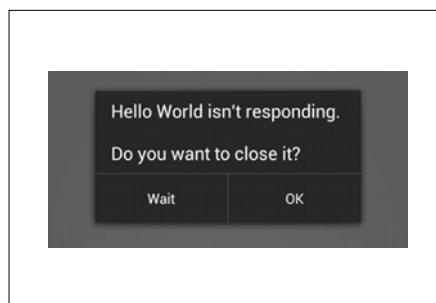


Figure 7. Celui qui bloque le fil GUI est sanctionné (source : Google).

Vu l'état de développement précoce d'*Android Things*, il n'y a actuellement pas de modèle disponible dans le générateur de projet d'*Android Studio*. À la place, vous devez accéder au dépôt GitHub en [5] et cliquer sur le bouton apparaissant sur la

**figure 3** pour télécharger un exemple de squelette de projet.

À ce moment, si vous avez un projet ouvert dans *Android Studio*, fermez-le en cliquant sur *File -> Close Project*. *Android Studio* affiche alors la fenêtre d'accueil.

Vous devez alors extraire l'archive téléchargée à l'étape précédente dans un répertoire facile d'accès. Cliquez ensuite sur *Open existing Android Studio Project* et indiquez ce répertoire. Un clic sur *OK* et *Android Studio* démarre la construction du projet.

Dans le cadre de la synchronisation du projet avec *Gradle* (moteur de production intégré), *Android Studio* essaie automatiquement de télécharger les composants manquants. Un éventuel problème est signalé par un message d'erreur dans la fenêtre des messages (**fig. 4**). Un clic sur le lien conduit habituellement à sa résolution.

Ne vous étonnez pas si *Android Studio* se plaint fréquemment de versions périmées. *Android Things* exige la toute dernière version de divers composants du système d'exploitation, qui doivent être installés en plusieurs étapes.

Après un démarrage réussi d'*Android Studio*, il est bon de cliquer encore une fois sur *Build -> Make Project* pour lancer une compilation complète. Ceci est important dans la mesure où une connexion à l'internet est indispensable pour une première compilation. Ensuite, vous avez une bonne chance de pouvoir travailler aussi hors connexion avec le squelette de votre projet.

## De menus changements

Le code de *MainActivity*, qui sert de point d'entrée, se distingue des applications normales d'Android par le fait que Google y insère des appels supplémentaires à *Log*. Ils sont nécessaires, car *Android Things* peut travailler sans écran. Dans ce cas, on obtient, grâce à ces appels, un minimum d'informations sur la console de débogage.

Assurez-vous aussi que le répertoire *res* est vide, comme le montre la **figure 5** : derrière cette *Activity*, il n'y a pas de procédure XML, ce qui se traduit dans le code (**listage 1**) par l'absence d'appels XML de chargement de ressources.

À noter aussi la structure du fichier de manifeste, reproduit partiellement dans le **listage 2**, qui présente deux modifications intéressantes : d'abord, la partie *Library* assure que les bibliothèques mentionnées plus haut seront bien liées au projet, ensuite, la *MainActivity* est pourvue d'un filtre additionnel *intent-filter* qui la désigne comme point d'entrée pour les appareils *Android Things*.

```
pi@raspberrypi:~$ gpio readall
```

Pi 1					Pi 2					Pi 3						
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM	Physical	V	Mode	Name	wPi	BCM
2	8	3.3v			1	2		5v								
3	9	SDA.1	IN	1	3	4		5V								
4	7	SCL.1	IN	1	5	6		0v								
		GPIO. 7	IN	1	7	8	1	ALTO	TxD	15	14					
		0v			9	10	1	ALTO	RxD	16	15					
17	0	GPIO. 0	IN	0	11	12	0	IN	GPIO. 1	1	18					
27	2	GPIO. 2	IN	0	13	14		0v								
22	3	GPIO. 3	IN	0	15	16	0	IN	GPIO. 4	4	23					
		3.3v			17	18	0	IN	GPIO. 5	5	24					
10	12	MOSI	IN	0	19	20		0v								
9	13	MISO	IN	0	21	22	0	IN	GPIO. 6	6	25					
11	14	SCLK	IN	0	23	24	1	IN	CE0	10	8					
		0v			25	26	1	IN	CE1	11	7					
0	30	SDA.0	IN	1	27	28	1	IN	SCL.0	31	1					
5	21	GPIO.01	IN	1	29	30		0v								
6	22	GPIO.22	IN	1	31	32	0	IN	GPIO.26	26	12					
13	23	GPIO.23	IN	0	33	34		0v								
19	24	GPIO.24	IN	0	35	36	0	IN	GPIO.27	27	16					
26	25	GPIO.25	IN	0	37	38	0	IN	GPIO.28	28	20					
		0v			39	40	0	IN	GPIO.29	29	21					

Figure 6. Ce tableau présente les broches du Raspberry Pi.



Ceci a son importance dans la mesure où *Android Things* doit pouvoir se passer d'un point de début de programme. Le RPi est en fait reconverti en un simple outil asservi à son application hôte.

### Entrées/Sorties

Pour une première démonstration, intéressons-nous au moteur GPIO. Un vieil adage stipule que les performances en temps réel d'un système d'exploitation diminuent linéairement ou même exponentiellement avec sa complexité : c'est particulièrement désagréable avec les systèmes basés sur Java où le ramasse-miettes fait de temps en temps des siennes. Rappelons au passage que le Raspberry Pi est une plateforme en 3,3 V. Lui raccorder des périphériques en 5 V conduit à coup sûr au désastre !

Nous pouvons maintenant nous intéresser au code de l'application dans la *MainActivity* (**listage 3**). Commençons par ajouter deux variables membres. L'API des périphériques est réalisée sous la forme d'un service système. Un service est une ressource généralement disponible au niveau le plus large : les applications qui veulent l'utiliser n'ont qu'à se procurer un identifiant pour y accéder. La classe *Gpio* sert à contrôler la commutation des broches.

C'est la fonction *onCreate* (**listage 3**) qui procure l'accès au service.

Même si Google ne traite pas les fonctions GPIO à 100% de la même manière qu'Arduino, on voit néanmoins clairement ce qui se passe ici. Le tableau de la **figure 6** donne les identifiants BCM associés aux diverses broches. On obtient ce tableau en entrant la commande *GPIO Readall* sur un RPi (sous Raspbian standard, pas sous Android).

### Mise au point

Pour évaluer la fiabilité ou la performance en temps réel d'un système d'exploitation, on peut effectuer un changement d'état périodique d'une sortie GPIO aussi rapidement que possible. Si l'on observe le spectre obtenu sur un analyseur de domaine de modulation, on peut en tirer quelques conclusions. Théoriquement, il n'y a pas d'objection à commuter la broche directement dans la fonction *onCreated*, qui est appelée au démarrage du programme. Mais si l'on essaie cela, on obtient le message d'erreur de la **figure 7**.

#### Listage 1. Squelette de la *MainActivity* – Point de démarrage du programme sans interface utilisateur

```
public class MainActivity extends Activity {
    private static final String TAG = MainActivity.class.
        getSimpleName();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Log.d(TAG, „onCreate“);
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        Log.d(TAG, „onDestroy“);
    }
}
```

La raison de ce comportement plutôt étrange tient à une spécificité du système Android : la partie interface utilisateur des programmes est exécutée dans un fil (*thread*) dédié, appelé *GUI*. Une application qui bloquerait ce fil se ferait « sortir » sans pitié par le système. Une boucle sans fin serait un blocage classique qui n'aurait à coup sûr droit à aucune pitié.

Pour contourner ce problème, nous pouvons transférer notre routine dans un autre fil.

Pour créer un nouveau fil, le moyen le plus simple est d'ajouter un *Runnable* au moyen d'une classe avec la structure suivante :

```
public class WaveformEmitter
    implements Runnable {
    @Override
    public void run() {
    }
}
```

Un *Runnable* n'est en somme rien d'autre

#### Listage 2. Fichier de manifeste avec l'inclusion de la bibliothèque

```
<manifest . . .>
    <application
        - - ->
        <uses-library android:name="com.google.android.Things"/>

        <activity android:name=".MainActivity">

            . . .

            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.
IOT_LAUNCHER"/>
                <category android:name="android.intent.category.
DEFAULT"/>
            </intent-filter>

        </activity>
    </application>
</manifest>
```

### Listage 3. La classe Gpio gère la commutation des broches

```
public class MainActivity extends Activity {

    private static final String TAG = MainActivity.class.
getSimpleName();
    PeripheralManagerService service;
    Gpio myGPIO;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Log.d(TAG, „onCreate“);

        PeripheralManagerService service = new
PeripheralManagerService();
        try{
            myGPIO = service.openGpio(„BCM6“);
            myGPIO.setDirection(Gpio.DIRECTION_OUT_INITIALLY_LOW);
            myGPIO.setValue(true);
        }
        catch (Exception e){Log.d(TAG, „Fehler:“ + e.getMessage());}
```

qu’une espèce de conteneur qui fait s’exécuter le code dans un nouveau fil. Outre la méthode `run` spécifiée ici, on peut bien entendu ajouter d’autres membres afin de stocker les informations nécessaires à l’exécution du fil.

Au complet, la classe résultante se présente comme dans le **listage 4**.

Outre le constructeur chargé de prendre en compte l’instance de la classe `Gpio`, nous avons peuplé la méthode `run()`,

### Listage 4. La boucle sans fin de production du signal est exécutée dans son propre fil – un Runnable sert de conteneur

```
public class WaveformEmitter implements Runnable {
    Gpio myGpio;
    public WaveformEmitter(Gpio _which){
        myGpio=_which;
    }
    @Override
    public void run() {
        try {
            while(1==1){
                myGpio.setValue(true);
                myGpio.setValue(false);
                myGpio.setValue(true);
                myGpio.setValue(false);
                myGpio.setValue(true);
                myGpio.setValue(false);
            }
        }
        catch (Exception e){}
    }
}
```

qui est responsable de la forme d’onde émise. Nous produisons ici une forme d’onde caractéristique composée de trois rectangles, ce qui est intéressant dans la mesure où l’exécution de la boucle `while` est mise en évidence de manière séparée.

Maintenant la question est de savoir comment activer le `Runnable` dans `onCreate`. Une erreur classique de débutant serait l’appel direct de `run()`. Ce faisant, on exécute le code dans le contexte de la méthode appelante (c’est-à-dire à nouveau dans le fil *GUI*). La manière correcte d’activer le second fil est de créer une nouvelle classe de fil à laquelle on passe l’objet « charge utile » (*payload*) `Gpio` en paramètre (**listage 5**). La méthode `start()` permet de démarrer le fil.

Nous sommes maintenant prêts à transférer le programme sur le RPi. Comme l’*ADB* fonctionne comme une couche d’abstraction, il suffit d’un clic sur *Run*, le RPi se comporte alors comme un ordi-phone connecté au PC par une liaison USB. Comme *MainActivity* n’inclut aucune interface utilisateur, l’écran connecté au RPi devient blanc, ce qui montre ainsi que notre *Activity* s’est mise joyeusement au travail.

### Exploitation

Connectez maintenant un analyseur de domaine de modulation au RPi pour bénéficier de la copie d’écran de la **figure 8** (l’auteur propose en [6] une vidéo en anglais qui fournit des informations complémentaires sur le rôle et l’intérêt d’un analyseur de domaine de modulation).

Outre une instabilité occasionnelle, on remarque deux pics : la plage à 2,073 kHz montre les deux « passages linéaires » alors que l’exécution de la boucle `while` se traduit par une fréquence moindre de « seulement » 2,062 kHz. Il est évident qu’Android ne peut en aucun cas concourir avec un Unix classique ; la machine virtuelle Java intermédiaire réclame son tribut. On trouvera sur l’internet divers articles sur le comportement en temps réel d’Android, par exemple en [7].

### Conclusion

Le programme que nous venons de réaliser est un exemple classique de gaspillage de ressources dans le domaine de l’embarqué. Un programmeur raisonnable



obtiendra une forme d'onde stable sans système en temps réel, rien qu'avec une poignée de lignes de code et un PIC.

Il faut reconnaître que le « *bit banging* » (exécution d'un protocole série par logiciel) ne fait pas vraiment partie du domaine d'application favori d'*Android Things*. Pour que la plateforme puisse montrer sa vraie puissance, il faut des applications plus exigeantes. Dans le prochain article, nous vous montrerons comment exploiter des données de capteurs et les afficher sous forme de diagramme. D'ici là, je vous souhaite un bon codage ! ◀

(160361 – version française : Helmut Müller)

#### Listage 5. Démarrage du fil de production du signal dans la MainActivity

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    . . .

    WaveformEmitter myEmitter=new WaveFormEmitter(myGPIO);
    new Thread( myEmitter ).start();
}
```

#### Liens

- [1] <https://developer.android.com/things/sdk/index.html>
- [2] <https://developer.android.com/things/preview/download.html>
- [3] <https://developer.android.com/things/hardware/raspberrypi.html>
- [4] <https://developer.android.com/studio/install.html>
- [5] <https://github.com/androidthings/new-project-template>
- [6] [www.youtube.com/watch?v=IBLEfVUVGyU](http://www.youtube.com/watch?v=IBLEfVUVGyU)
- [7] [www.utdallas.edu/~cxl137330/courses/fall14/RTS/papers/4a.pdf](http://www.utdallas.edu/~cxl137330/courses/fall14/RTS/papers/4a.pdf)

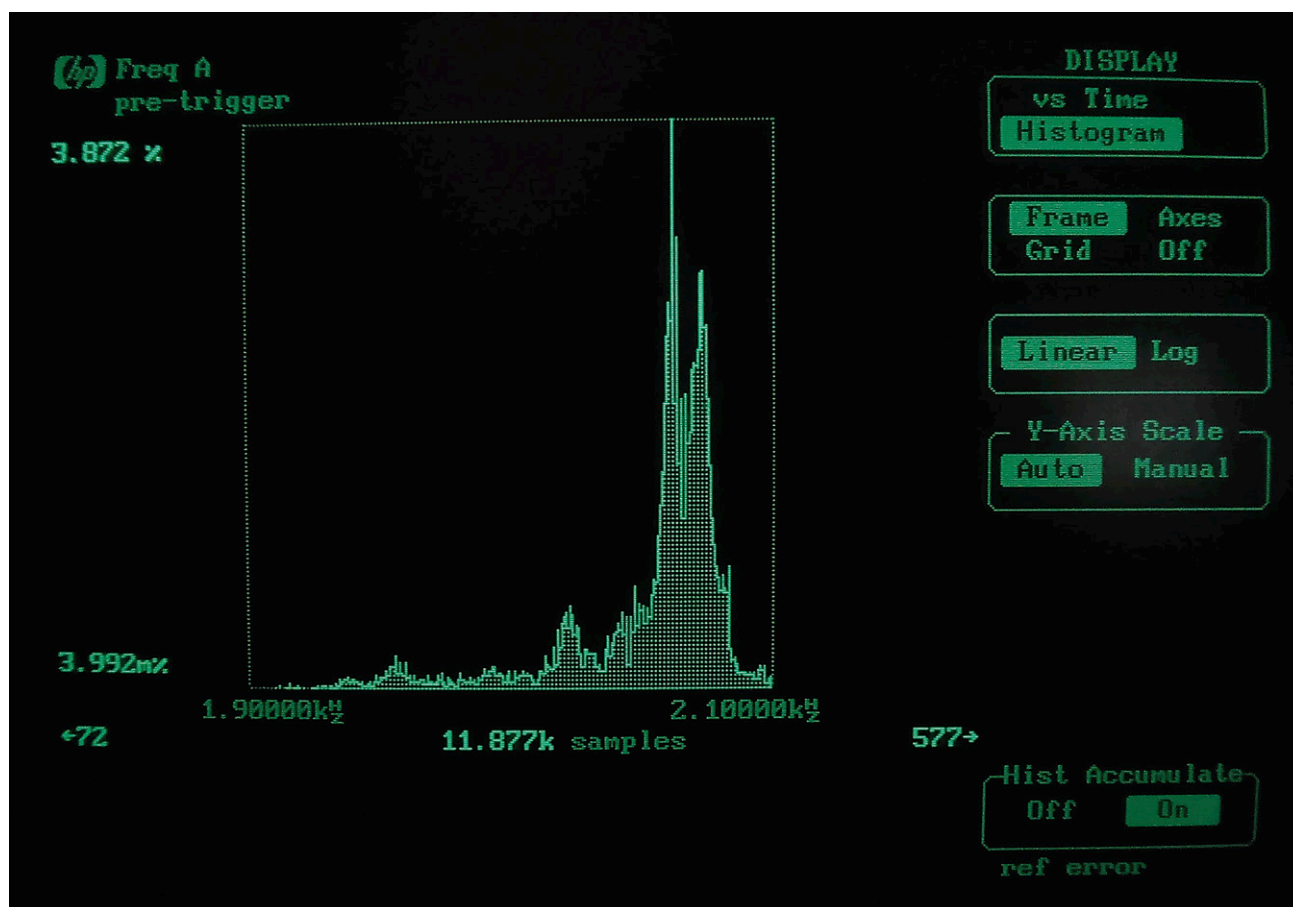


Figure 8. Les Arduinos produisent de bien plus beaux histogrammes.

# FORUM LED EUROPE

## de l'Internet des Objets à l'Internet de la Lumière

Pour la 8<sup>e</sup> année consécutive, le congrès ForumLED Europe a réuni en décembre 2016, à Lyon, les meilleurs spécialistes internationaux en matière de LED et de leurs applications. Ils ont dressé un état des lieux et partagé leur vision de l'avenir du marché et de la technologie LED. Les conférences étaient complétées par une exposition consacrée aux LED (modules, alimentation, outils de mesure et test, composants optiques, thermiques...) et une autre aux systèmes et matériels d'éclairage à LED.

Mariline Thiebaut-Brodier (Elektor)

### État des lieux

Ces deux journées de conférence ont commencé par un état des lieux et des tendances du marché de l'éclairage. Pour Pars Mukish de *Yole Développement* (conseil en stratégie), l'année 2015 avait été la première année de baisse du prix des LED [1]. En dépit de l'augmentation de la demande, le chiffre d'affaires des fabricants de LED avait diminué. La situation s'est améliorée en 2016 avec une augmentation du volume des ventes et l'augmentation du prix de certaines LED. Après le rétroéclairage des téléviseurs et des ordiphones, les LED se répandent dans l'éclairage domestique ainsi que l'éclairage extérieur et architectural. On constate que le nombre de points lumineux augmente parce que les sources à LED sont efficaces, bon marché et polyvalentes. Enfin comme les LED ont une durée de vie moins longue que prévu (20.000 h plutôt

que 50.000 h), les ventes pour le remplacement grossissent également les chiffres d'affaires.

Certains fabricants ont décidé de se diversifier pour se distinguer de leurs concurrents :

- Nouvelles formes de LED : filament (imitation des lampes à incandescence, **fig. 1**), COB (plafonds, vitrines, projecteurs), *flip chip*, CSP (boîtiers miniatures faciles à intégrer dans les montages avec des CMS).
- Applications spécialisées : automobile, horticulture (**fig. 2**), domaine de l'invisible (UV et IR).

Il y a quelques années, l'automobile – surtout de luxe – ne représentait qu'une portion congrue du marché de l'éclairage. Aujourd'hui, les feux d'une *Audi* conservent leur signature unique, mais les modèles de luxe ne sont plus les seuls équipés de LED. C'est maintenant l'éclairage général qui bénéficie des progrès du monde automobile.



Figure 1. LED à filaments pour remplacer les lampes à incandescence aujourd'hui disparues (modèles evoZiled).



Figure 2. LED optimisées pour l'horticulture avec des longueurs d'onde qui favorisent la photosynthèse et améliorent la croissance.





En 2008, une quinzaine de sociétés proposaient des LED UV, aujourd'hui elles sont plus d'une soixantaine. Auparavant les LED UV étaient surtout utilisées pour le séchage, maintenant elles ont une seconde application : la purification et la désinfection de l'air, des surfaces et de l'eau. Un marché en plein essor qui devrait dépasser celui du séchage.

La société suisse Exalos est venue présenter sa SLED (*Superluminescent Emitting Diode*), composant hybride entre la LED et la diode laser. La SLED est une source de lumière à large bande (comme la LED), mais avec un faisceau bien défini (comme la diode laser). Elle peut être considérée comme une diode laser incohérente au sens ondulatoire (alors que la lumière laser est cohérente) (**fig. 3**).

N'importe quelle source de lumière peut interférer avec sa propre lumière ou celle d'une autre source, mais seulement dans sa longueur de cohérence. Les réflexions par des surfaces qui ne sont jamais parfaites peuvent même causer des « tavelures » (ou chatoiements) quand l'écart entre les longueurs d'onde des lumières parasites est inférieur à la longueur de cohérence. Les tavelures sont considérées comme du bruit. Généralement les diodes laser produisent ce type de bruit parce que leur longueur de cohérence est élevée. Les SLED peuvent être considérées comme des diodes laser sans tavelures. Les principales applications des SLED sont les capteurs à fibre optique, les gyroscopes à fibre optique (effet Sagnac), les instruments de diagnostic pour la cornée et la rétine (tomographie), les outils de test des fibres optiques (dispersion chromatique et de la polarisation).

Du côté des afficheurs, c'est la course aux pixels (résolution 4K), aux économies d'énergie (fonctionnement sur pile/batterie) et à la luminance (essentiellement pour l'extérieur). La prochaine rupture technologique devrait donc venir de la microLED (diamètre < 10  $\mu\text{m}$ ), inventée par le professeur Hongxing Jiang de l'université Texas Tech. Diminuer la taille des LED permet d'éviter la pixellisation qui provoque un éblouissement (phares de voiture par ex.). Les  $\mu\text{LED}$  présentent d'autres avantages : une luminance (> 10<sup>7</sup> cd/m<sup>2</sup>) et un rendement

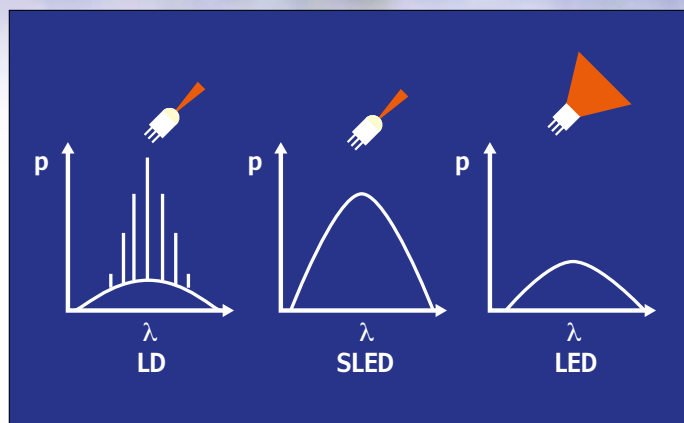
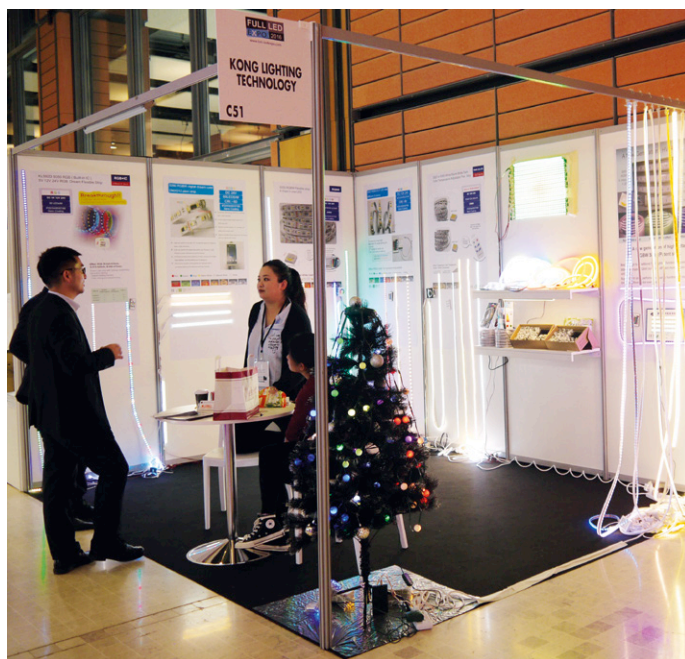


Figure 3. Comparaison entre SLED, LED et diode laser (source : Exalos).



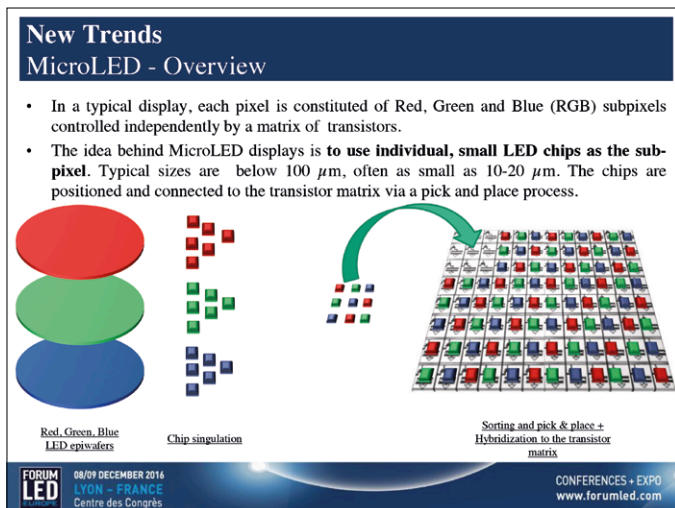


Figure 4. Structure d'un écran couleur à microLED (source : Yole).

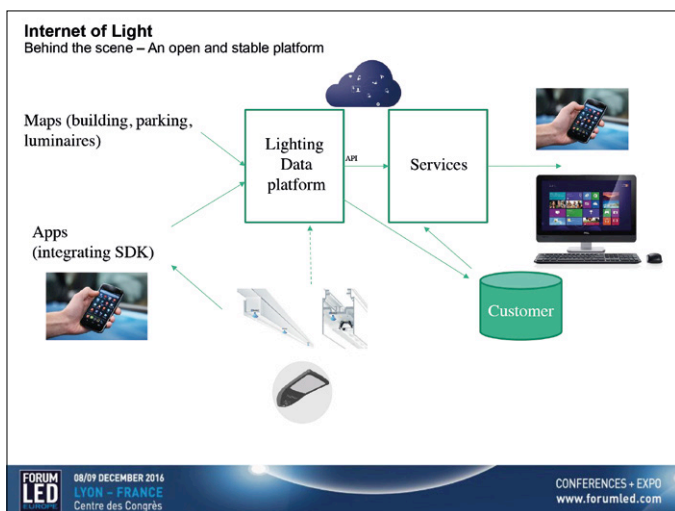


Figure 5. De l'Internet des Objets à l'Internet de la Lumière (source : Zumtobel).

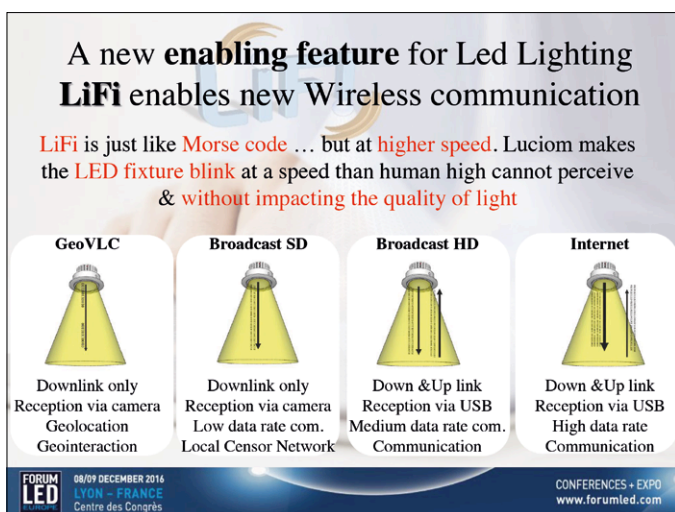


Figure 6. Li-Fi, le réseau de communication sans fil lumineux (source : Luciom).

quantique (EQE = 17%) supérieurs à ceux des OLED.

Un afficheur à  $\mu\text{LED}$  est constitué d'un réseau de LED rouges, vertes et bleues microscopiques où chaque groupe de LED RVB correspond à un pixel (**fig. 4**). Plusieurs fabricants travaillent sur ces très petits écrans ( $2,5 \times 2,5 \text{ mm}$ ) destinés aux casques de réalité virtuelle, à l'électronique vestimentaire ou aux petits projecteurs. La commercialisation pour le grand public ne devrait pas commencer avant 2020. Sony a déjà présenté un prototype de téléviseur baptisé CLEDIS (*Crystal LED Integrated Structure*) où les *crystal LED* sont des  $\mu\text{LED}$ .

### L'éclairage en tant que service

L'industrie européenne de l'éclairage est soumise à une concurrence internationale féroce. Comme elle ne peut pas réduire drastiquement le coût des produits « made in Europe », elle doit développer de nouvelles « solutions d'éclairage » avec des valeurs ajoutées pour l'utilisateur (design unique, maintenance facile, commande automatique...) et combinées à des technologies extérieures à l'éclairage (gestion de données, connectivité...). Le monde de l'éclairage doit s'ouvrir pour rejoindre la grande famille des objets connectés. C'est pourquoi les acteurs de l'éclairage se regroupent (voir [2] et [3]), organisent des groupes de réflexion et rencontrent les professionnels de différents domaines (santé, ville, transport, éducation, architecture...) pour connaître leurs besoins. Ils travaillent par exemple avec les fabricants de serres pour optimiser la position des lampes qui stimulent la croissance des plantes ou pour intégrer l'alimentation de l'éclairage aux structures (mise en œuvre et utilisation plus simples). Ils doivent devenir partenaires des différents corps de métier du bâtiment (volets, air conditionné, chauffage...).

Les éclairagistes disposent d'un atout pour obtenir un bout du gâteau de l'IdO (**fig. 5**) : ils sont déjà présents à l'intérieur comme à l'extérieur ; n'importe quel bâtiment contient un réseau qui relie les différents points d'éclairage et leur alimentation (en moyenne un point lumineux pour  $5 \text{ m}^2$ ) ; les rues sont jalonnées de lampadaires (environ tous les 50 m).

Pour permettre aux réseaux intérieurs d'accueillir les capteurs de tiers et de transmettre des données, les fabricants rendent leurs modules d'éclairage plus intelligents, c'est-à-dire qu'ils ne sont plus limités au protocole DALI, mais parlent également Bluetooth (Bluetooth Mesh, Eddystone, iBeacon...). Petit à petit, ils deviendront polyglottes : Wi-Fi HaLow, ZigBee, Thread... Ils peuvent donc être commandés depuis une appli sur un ordi- phone ou une tablette. Si un luminaire détecte la présence d'un téléphone, il allume les lampes en fonction des réglages sauvegardés dans le téléphone. Est-ce la fin des interrupteurs et autres boîtiers de commande ? Comme la communication est bidirectionnelle, les modules intelligents transmettent des données sur leur propre fonctionnement (heures de fonctionnement, température).

Le réseau d'éclairage pourrait donc jouer un rôle de passerelle entre objets connectés et serveurs dans le nuage. Comme pour tous les acteurs de l'IdO, les éclairagistes gagneront de l'argent en vendant du matériel, mais aussi des services (logiciels). Dans un immeuble de bureaux, ce sera la gestion de l'éclairage pour réaliser des économies d'énergie, mais aussi l'analyse du comportement des occupants pour mieux gérer l'occupation de l'espace. Des sociétés comme Zumtobel proposent de louer un réseau d'éclairage avec capteurs. Le propriétaire de



l'immeuble n'investit rien au départ. Son contrat de location définit le niveau de services : maintenance uniquement (dont les coûts sont souvent sous-estimés), éclairage automatisé, éclairage personnalisé, efficacité énergétique garantie, quantité de lux garantie... Bien entendu ces contrats ne sont rentables que sur une longue période (5 à 10 ans) et si le coût du kWh est élevé (comme c'est le cas en Allemagne, 16 ct contre 6 à 7 ct en France). Que se passe-t-il si on ne renouvelle pas le contrat (ou si le prestataire disparaît) ? Il faudrait alors démonter le réseau d'éclairage... Est-on prêt à louer des luminaires comme on le fait pour des voitures ?

En extérieur, les éclairagistes espèrent bien jouer un rôle essentiel dans la construction des « villes intelligentes » (*smart cities*) : les réseaux urbains d'éclairage pourraient être l'épine dorsale de l'infrastructure de connexion. Chaque point d'éclairage peut alimenter plusieurs capteurs : environnement (pollution, eau...), trafic, sécurité, services d'urgence... Les lampadaires, en facilitant l'intégration dans la base de données de la ville intelligente, présentent l'avantage d'une durée de vie très longue, plusieurs dizaines d'années. Alors que l'éclairage public avait un coût jusqu'à présent, il pourrait à l'avenir être une source non seulement d'économies grâce à une meilleure gestion de l'éclairage, mais aussi de revenus grâce aux services rendus.

### Communication éclairée : Li-Fi

Au fil des présentations consacrées à l'infrastructure d'éclairage transformée en infrastructure de services, on a vu réapparaître un *serpent de mer* : le Li-Fi, acronyme construit avec *light*, pour lumière, et *fidelity*, pour fidélité. C'est une technologie de connexion sans fil à l'internet : l'amplitude de la lumière d'une LED est modulée à haute fréquence (scintillement imperceptible à l'œil nu) pour coder et transmettre des informations (fig. 6). Le Li-Fi travaille dans la partie visible du spectre électromagnétique (spectre optique entre 480 nm et 650 nm) alors que le Wi-Fi exploite la partie radio de ce spectre. Un ordiphone ou un ordinateur équipé d'un émetteur-récepteur et placé dans le cône de lumière Li-Fi bénéficie d'une connexion bidirectionnelle à l'internet, à 10 Mbits/s, voire plus.

Le Li-Fi est un sujet régulièrement abordé ces dernières années, mais qui peine à prendre une forme concrète. En 2016, la société française *Lucibel* a mis sur le marché le premier luminaire Li-Fi. Relié à l'internet par un câble RJ45, il comporte un émetteur-récepteur pour la modulation et la réception du signal émis par le terminal connecté. Pour que le Li-Fi soit un jour largement adopté, il faudrait que téléphones, tablettes et ordinateurs soient tous équipés de l'émetteur-récepteur adéquat. Les promoteurs du Li-Fi arguent que l'utilisation de la lumière au lieu des ondes radio est bénéfique pour les personnes électrosensibles. Toutefois aucune étude n'a été menée sur l'impact du scintillement des LED. Un autre avantage du Li-Fi par rapport au Wi-Fi est de ne pas traverser les murs opaques. Cela limite les risques de piratage.

### Impact de la lumière sur le vivant et l'environnement

#### Nos amies les bêtes

Les présentations sur l'impact de la lumière artificielle sur les écosystèmes étaient très intéressantes et permettaient d'aborder l'éclairage à LED sous un angle original. La pollution lumi-



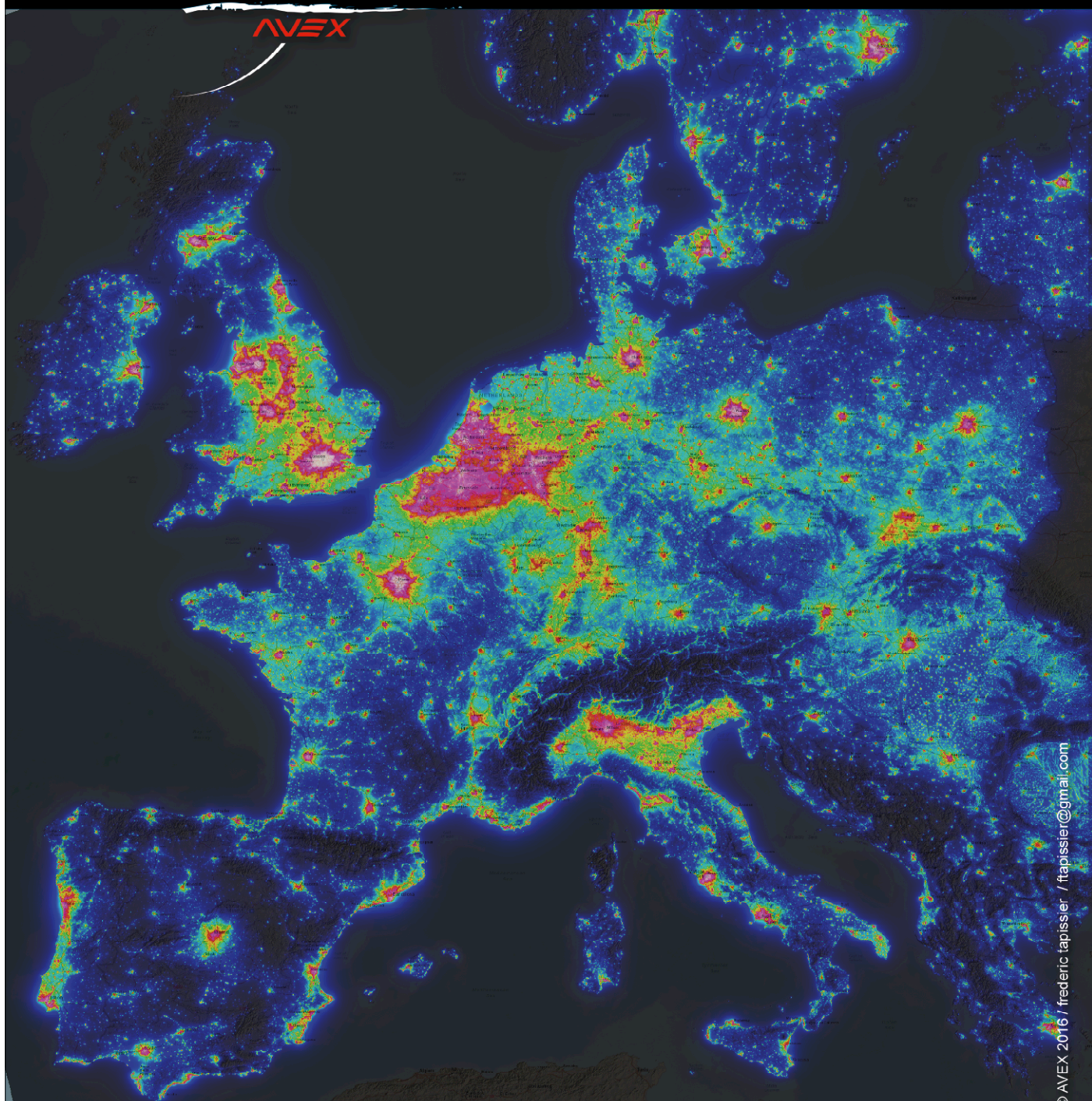
neuse (fig. 7) met en danger plusieurs espèces d'animaux. Plus de 75% des lepidoptera (papillons) ont une vie nocturne, tout comme 60% des mammifères. Certains poissons sont lucifuges et ne sortent qu'à la nuit tombée pour se nourrir. Si leur cours d'eau est éclairé en permanence (ponts, rives), ils n'ont plus de période propice à leur alimentation. Ils disparaissent. Vous avez tous observé des insectes irrémédiablement attirés par un lampadaire. Ils tournent autour de la lumière jusqu'à l'épuisement. Les papillons de nuit sont ainsi détournés de leur travail de pollinisation. Les petites chauves-souris qui volent lentement ne peuvent pas traverser les couloirs fortement éclairés.





Carte de Pollution lumineuse 2016  
European Light pollution Map 2016  
Kart der lichtverschmutzung 2016

AVEX



© AVEX 2016 / frederic tapisier / ftapisier@gmail.com

Blanc : 0-50 étoiles  
Magenta : 50-100 étoiles  
Rouge : 100-200 étoiles  
Orange : 200-250 étoiles  
Jaune : 250-500 étoiles  
Vert : 500-1000 étoiles  
Cyan : 1000-1800 étoiles  
Bleu : 1800-3000 étoiles  
Bleu nuit : 3000-5000 : étoiles  
Noir : + 5000 étoiles

Weiss : 0-50 sterne  
Magenta : 50-100 sterne  
Rot : 100-200 sterne  
Orange : 200-250 sterne  
gelb : 250-500 sterne  
Grün : 500-1000 sterne  
Cyan : 1000-1800 sterne  
Blau : 1800-3000 sterne  
Mitternachtsblau : 3000-5000 : sterne  
Schwarz : + 5000 sterne

White : 0-50 stars  
Magenta : 50-100 stars  
Red : 100-200 stars  
Orange : 200-250 stars  
Yellow : 250-500 stars  
Green : 500-1000 stars  
Cyan : 1000-1800 stars  
Blue : 1800-3000 stars  
Night blue : 3000-5000 : stars  
Black : + 5000 étoiles

rés, leur zone d'éparpillement de graines est alors limitée. On constate que la population de lucioles a d'autant diminué que la pollution lumineuse a augmenté. En avez-vous vu récemment ?

### Nous, les humains

Chez les mammifères, et donc les humains, les cellules ganglionnaires à mélanopsine situées dans la rétine détectent la lumière, en particulier la lumière bleue ( $\lambda_{\text{sensibilité max}} = 479 \text{ nm}$ ). Elles transmettent un signal à notre horloge biologique qui régule le sommeil, les hormones ou encore la température corporelle. On parle de rythme circadien.

La lumière, en particulier l'alternance jour-nuit, est le facteur de synchronisation le plus puissant. L'exposition à une lumière artificielle pendant la phase nocturne peut perturber le rythme circadien et la production des hormones, dont la mélatonine et le cortisol.

La mélatonine, appelée « hormone du sommeil », est sécrétée en grande quantité la nuit. Le manque de mélatonine augmente le risque de cancer, altère la mémoire, provoque des troubles neurocognitifs... Le cortisol, sécrété en permanence, régule le métabolisme des glucides, lipides, protéides ou encore de l'eau. Une perturbation des quantités de cortisol sécrétées tout au long de la journée peut être à l'origine de polyarthrite rhumatoïde, de dépression...

### Comment atténuer le côté sombre de la lumière ?

Le nombre de lumens installés à l'extérieur a progressé de 5% par an au cours des cinquante dernières années. Faut-il devenir un activiste de la lutte contre la pollution lumineuse et arracher tous les lampadaires ? Non, il faut plutôt moduler l'éclairage :

- réduire la quantité de lumens émis
- éviter d'envoyer de la lumière directement vers le ciel
- travailler sur le spectre utilisé. Le bleu par ex. se disperse plus que les autres couleurs, il faut donc éviter les longueurs d'onde inférieures à 500 nm.

Si on installe un point lumineux puissant dans une rue, ce qui se trouve autour paraîtra sombre. Il serait possible de réduire la puissance d'éclairage si on travaillait sur l'uniformité de l'éclairage en rapprochant les points lumineux par exemple. Si l'œil n'est pas ébloui, il s'adapte facilement, même si l'éclairage est faible.

À l'intérieur aussi, nous devons réfléchir à moduler notre éclairage. Dans un rapport d'octobre 2010, l'ANSES soulignait déjà les risques liés à la lumière bleue, soupçonnée de constituer un facteur de risque important dans certains cancers cutanés, certaines tumeurs de l'œil et le vieillissement de la peau. Elle indiquait également : « L'arrivée des LED en éclairage sur le marché marque un tournant sans précédent : c'est la première fois que des sources classées dans le groupe de risque 2 sont accessibles au grand public, en vue d'applications domestiques

Figure 7. Carte européenne de la pollution lumineuse. Dans les zones en blanc, on ne peut voir que 0 à 50 étoiles. La pollution lumineuse est très puissante et omniprésente. Dans les zones en bleu foncé, on peut voir 3.000 à 5.000 étoiles ainsi que la Voie lactée (source : Frédéric Tapissier, [www.avex-asso.org](http://www.avex-asso.org)).

et qui plus est, sans marquage de ce risque ». Inquiétant, non ? Avez-vous installé *f.lux* sur vos ordinateurs ? C'est un logiciel (à télécharger en [6]) qui adapte les couleurs de votre écran tout au long de la journée, froides le matin, chaudes le soir. Vous pouvez également essayer les paires de lunettes qui filtrent la lumière bleue.

Il faudrait aussi que les luminaires à LED puissent produire plus de lumière bleue le matin et moins le soir. Les étiquettes énergétiques des lampes devraient indiquer dans quelle mesure elles agissent sur la production de mélatonine. Des études sont en cours sur les effets des couleurs du spectre sur notre santé.

### Conclusion

Ces quelques lignes ne donnent qu'un aperçu des différents thèmes abordés pendant ces deux jours. Il y a également été question de normes, d'outils de mesure, de retours d'expérience sur l'éclairage urbain... Dans tous les cas, il est certain que l'éclairage à semi-conducteur est en perpétuelle évolution et soulève encore beaucoup de questions.

(160376)

### Liens

- [1] Rapports de Yole Développement sur le marché et les technologies des LED : [www.i-micronews.com/reports/led-reports.html](http://www.i-micronews.com/reports/led-reports.html)
- [2] SSL-erate, projet de coordination pour accélérer l'adoption de l'éclairage à semi-conducteurs en Europe : [ssl-erate.eu](http://ssl-erate.eu)
- [3] Plateforme d'information sur l'éclairage à semi-conducteurs en Europe : [lightingforpeople.eu](http://lightingforpeople.eu)
- [4] STARS4ALL, projet européen pour informer sur les effets négatifs de l'éclairage artificiel : [www.stars4all.eu](http://www.stars4all.eu)
- [5] *Loss of the Night Network* (LoNNe), informations sur les multiples effets de l'éclairage artificiel : [www.cost-lonne.eu](http://www.cost-lonne.eu)
- [6] Logiciel *f.lux* : [justgetflux.com](http://justgetflux.com)



### Glossaire

**COB** : *chip on board*, puces montées directement sur un circuit imprimé

**CSP** : *chip scale package*, boîtier-puce

**EQE** : *external quantum efficiency*, rendement quantique externe, à savoir rapport entre le nombre de photons émis par une LED et le nombre d'électrons qui traverse le composant - c'est-à-dire l'efficacité de la conversion d'électrons en photons.

**flip chip LED** : LED avec substrat monté sur (et non pas sous) le semi-conducteur émissif

**SLED** : *superluminescent LED*, LED superluminescente

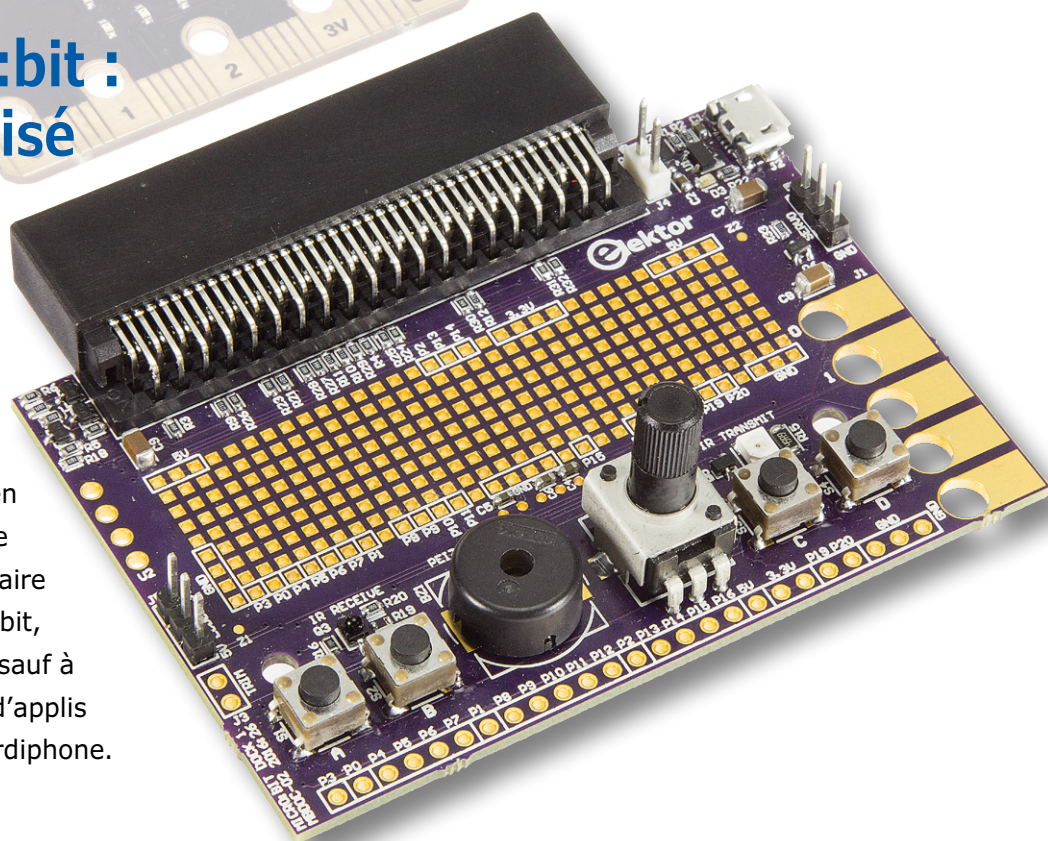


# station d'accueil pour BBC micro:bit

« base à micro:bit :  
vous êtes autorisé  
à vous poser »

Jason Long (Canada)

Aujourd'hui il y a au moins un million de cartes BBC micro:bit en circulation au Royaume-Uni et de par le monde. Même si on peut faire beaucoup de choses avec micro:bit, l'interface utilisateur est limitée sauf à devenir très bon dans l'écriture d'applis Bluetooth pour profiter de son ordiphone.



## À PROPOS DU PROJET

### Microcontrôleurs

BBC micro:bit Bluetooth  
prototypage

débutant

→ connaisseur

expert

environ 2 h

Soudage de CMS

environ 20 €

Mais les jeunes (et vieux) passionnés ne veulent pas être collés en permanence à un téléphone, et il y a sans doute une multitude de projets à réaliser avec une carte micro:bit autonome.

Nous avons donc créé une carte « station d'accueil » (il y en aura peut-être toute une série) pour ajouter de nouvelles caractéristiques et fonctions au système de base micro:bit. Ce premier *dock* fournit quelques E/S de plus et un espace de prototypage, et facilite l'éclatement de l'ensemble des broches de micro:bit sur une plaque d'essais.

### Connecteur de bord de carte !

Le premier défi à relever est le connecteur imprimé qui semble sympathique et surtout ne coûte rien. Toutefois il est pratiquement impossible de brancher micro:bit dans quoi que ce soit. Les embases pour connecteurs imprimés sont chères et leur profil est incompatible avec

les plaques d'essais. Les premières tentatives pour trouver des modèles abordables ont presque conduit à l'abandon du projet avant même son démarrage. Amusant comme un problème simple peut avoir un impact considérable ! Bienvenue dans l'ingénierie. Heureusement nous avons trouvé un connecteur bon marché.

Nous, vieux briscards des cartes de développement, avons nos options préférées. La première version du dock les a TOUTES plus la connexion pour une carte fille (au standard Engenuics), ce qui permet de connecter une infinité d'autres cartes. Nous faisons alors des suppositions sur le fonctionnement de micro:bit (difficile d'en avoir une à cette époque). Comme nous n'avions pas encore de connecteur de bord de carte, cette carte utilise notre idée initiale de connecteur à deux étages (fig. 1).

Nous lançons un prototypage rapide et

jouons un petit moment avec. Certains composants sont chers et la carte semble trop grande. Finalement nous retournons à la planche à dessin et nous nous focalisons sur un coût aussi bas que possible et les fonctions essentielles.

La seconde version élimine quelques composants coûteux et rajoute l'embase pour connecteur imprimé enfin trouvée. Le brochage de la version finale de micro:bit a un peu changé, nous appliquons aussi ces changements. Nous décidons de rendre le LCD optionnel et de le placer au-dessus de l'aire de prototypage pour minimiser la taille du circuit imprimé (**fig. 2**). Ne sachant pas comment évoluera micro:bit dans le futur, on peut déconnecter toutes les broches utilisées par le dock en ôtant une résistance de shunt. Ce serait dommage qu'un formidable projet se comporte bizarrement à cause d'interactions imprévues avec le dock.

Après quelques ajustements mineurs du placement et de la sérigraphie pour la production, nous sommes prêts à publier le projet. Examinons le fruit de notre travail (**fig. 3**).

### Plus de boutons

Les boutons sont la façon la plus simple de transmettre de l'information à un système embarqué et il n'y en a jamais assez ! Après débat sur la façon d'en ajouter, nous décidons de dupliquer les deux boutons existants (A & B) et d'en ajouter deux de plus (C & D). Vous dis-

posez maintenant de quatre boutons, faciles à consulter avec le code de la base micro:bit.

### Buzzzzzzzzzz et servocommande

Si les LED sont en tête des indicateurs de sortie numérique simples et pas chers, le son vient sans doute juste derrière. Avec un simple périphérique MLI ou la manipulation de bits, vous pouvez rendre les gens (et les chiens) fous avec des tonalités énervantes dans le spectre de fréquences audibles. Même si un buzzer piézoélectrique ne produit pas le meilleur son (soyons honnêtes, c'est horrible), il ajoute une nouvelle dimension fantastique à vos projets. Désolé pour votre entourage, en particulier les enseignants qui ont une classe pleine de cartes micro:bit.

Il y a un connecteur pour servocommande standard et avec l'alimentation USB, le niveau de la ligne SERVO passe du 3,3 V de micro:bit à 5 V. La ligne SERVO est multiplexée avec la ligne IR\_OUT, donc en laissant les deux résistances en place vous pilotez le moteur et émettez des bits IR aléatoires. Nous supposons que vous n'utilisez pas un moteur tout en jouant avec le laser IR (quoiqu'en écrivant cela nous imaginons des combats de robots avec laser...).

### Transmission et réception IR

La carte micro:bit supporte le Bluetooth

basse consommation (BLE) dont l'apprentissage sera difficile pour la majorité du public visé. Mais la communication sans fil c'est tellement bien ! Le dock la facilite en utilisant un émetteur-récepteur IR pour des tâches simples comme l'interruption de faisceau entre deux cartes, jusqu'à des communications complexes comme l'IrDA. On peut ainsi créer nombre d'applications qui exploitent la magie de la transmission d'informations par d'invisibles bits transitant dans les airs.

### Analogique

Que serait un système embarqué sans analogique ? Nous vivons dans un monde analogique ! Un potentiomètre moniteur de 10 k $\Omega$  permet d'appliquer à l'entrée analogique toute tension comprise entre les rails d'alimentation. Le  $\mu$ C de micro:bit a un convertisseur A/N à 10 bits, ce qui offre de nombreuses possibilités.

Comme cette entrée analogique est partagée avec la broche principale 1, nous avons ajouté un cavalier qui déconnecte l'ajustable pour éviter de charger un éventuel autre circuit analogique. Ceci facilite les essais soit en mode autonome, soit en raccordant l'ajustable pour tester toute la plage d'entrée de votre système.

### Puissance

La plupart des circuits du dock fonctionneront sur l'alimentation de micro:bit, mais l'idée est d'avoir une alimentation

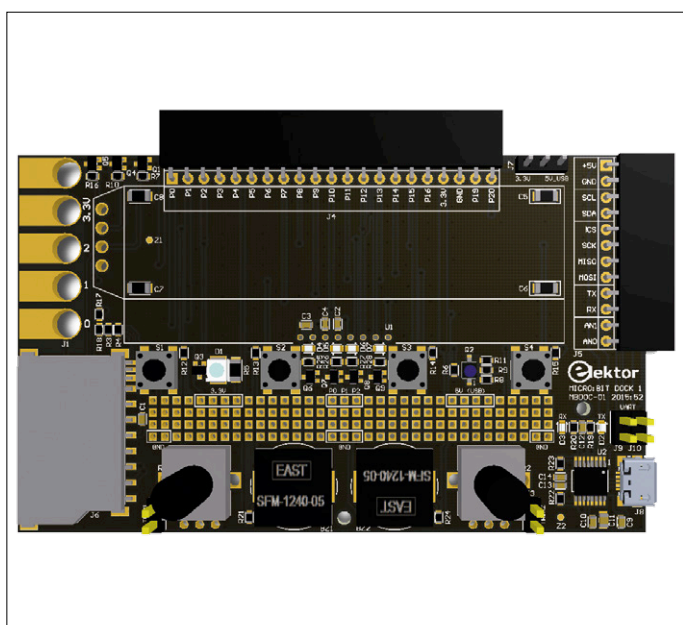


Figure 1. Vue 3D de notre premier jet de la station d'accueil pour la carte BBC micro:bit.

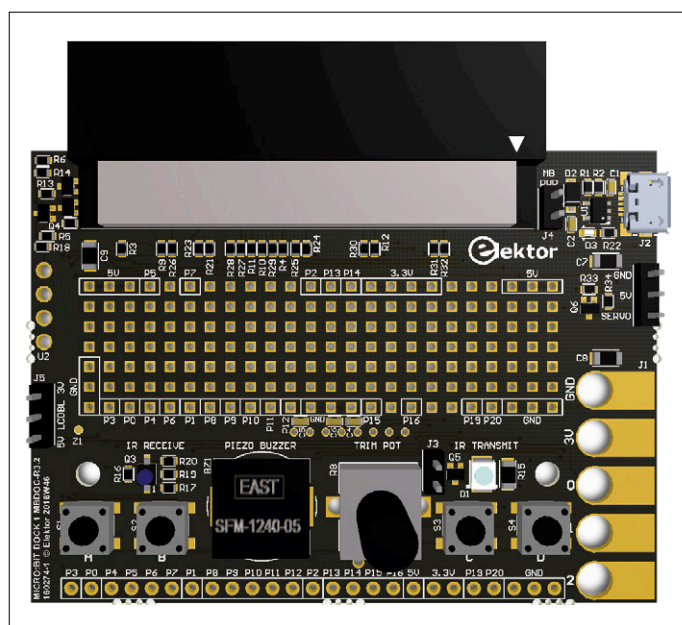


Figure 2. La version finale du dock pour BBC micro:bit ressemble à ceci.



extérieure. Le dock comporte un connecteur micro-USB qui permet d'utiliser la ligne de 5 V et un régulateur à faible chute de tension (LDO) de 3,3 V peut alimenter le dock et micro:bit. Un cavalier permet de les séparer. N'importe quelle alimentation micro-USB de 5 V devrait fonctionner, ne jetez pas vos vieux char-

geurs de téléphone !

### LCD (optionnel)

Le roi de l'interface utilisateur est le LCD, il est donc indispensable sur le dock. Ce sera un afficheur à 2×20 caractères ASCII avec un rétroéclairage RVB. Nous ajoutons donc une LED RVB au système ;

ainsi même si vous n'écrivez jamais une ligne de code pour allumer les caractères du LCD, vous pourrez créer de magnifiques effets lumineux avec votre carte. Le pilote du LCD est compatible avec le standard HD44780, mais utilise une interface I<sup>2</sup>C pour économiser une cargaison de broches. Le cavalier J5 per-

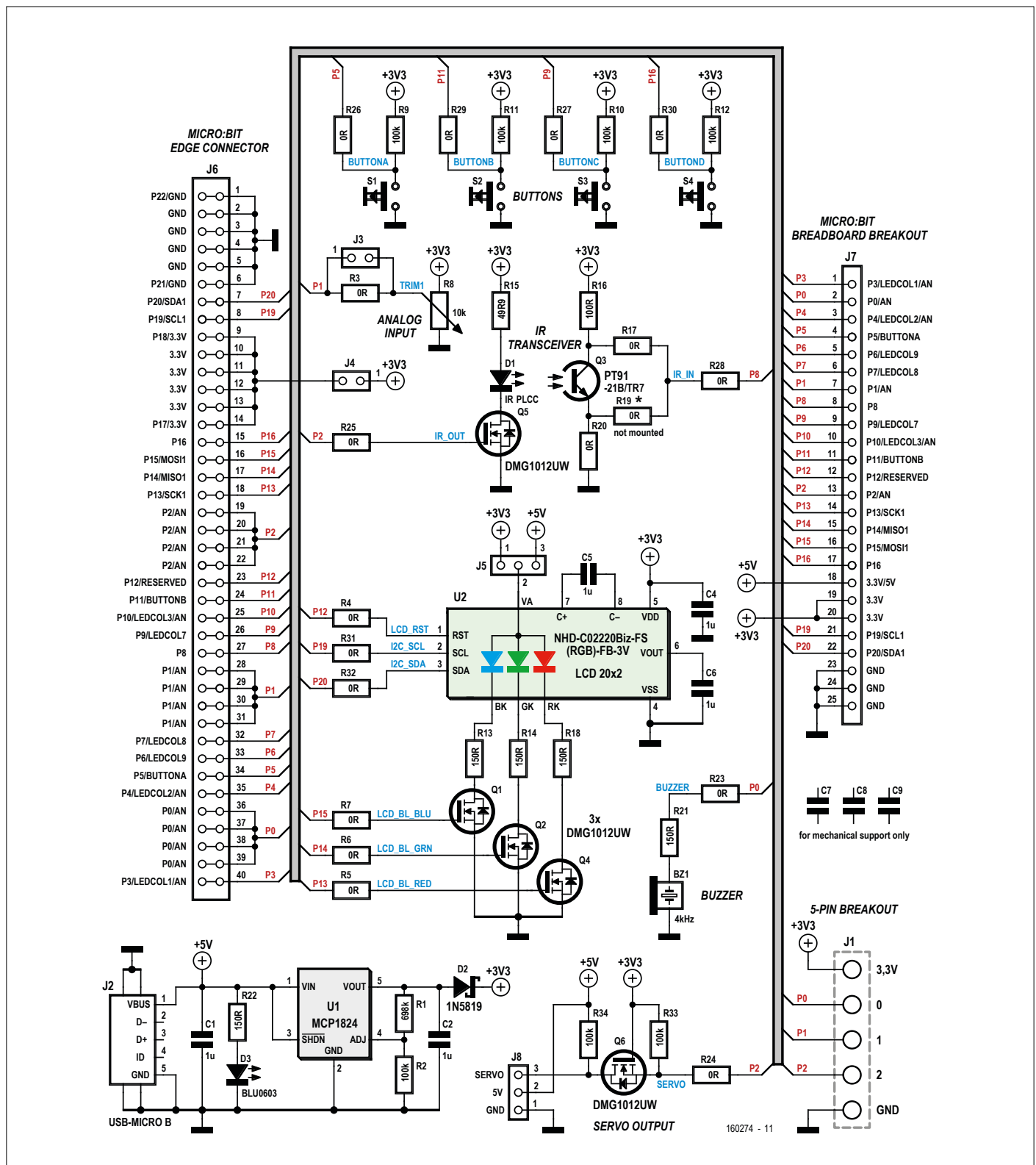
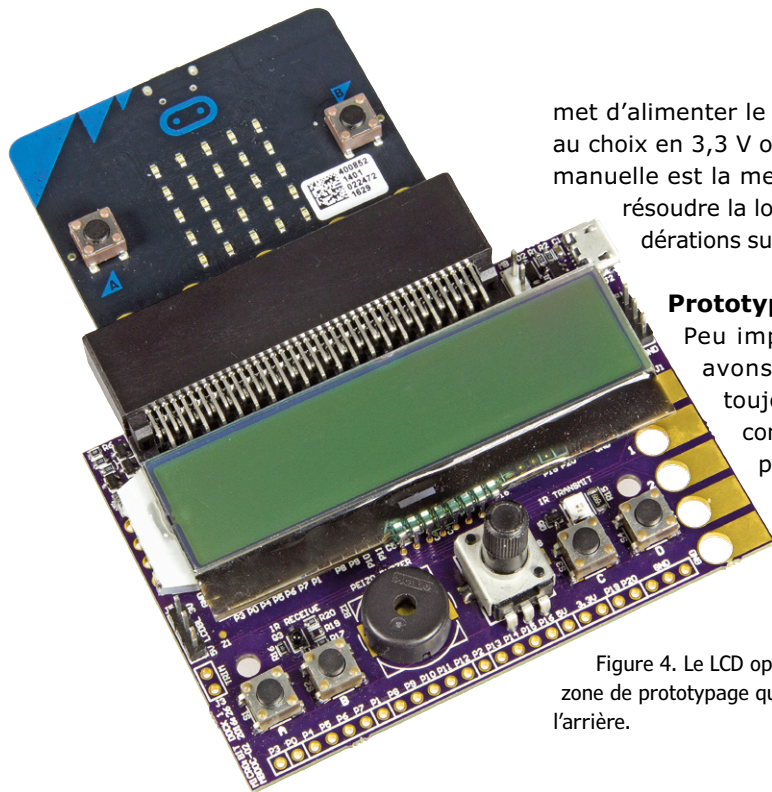


Figure 3. Schéma de notre dock.



met d'alimenter le rétroéclairage RVB au choix en 3,3 V ou 5 V – la sélection manuelle est la meilleure option pour résoudre la longue liste de considérations sur cette alimentation.

### Prototypage

Peu importe ce que nous avons, nous en voulons toujours plus ! Le dock comporte une zone de prototypage suffisante pour y souder ses propres

Figure 4. Le LCD optionnel recouvre la zone de prototypage qui reste accessible par l'arrière.

circuits. Les pastilles sont conçues pour des composants CMS 0603, si vous les préférez aux composants traversants de la vieille école. Notez que toutes les broches de micro:bit sont disponibles avec nombre d'alimentations et de masses. Les deux côtés de la carte étant annotés, vous pouvez ajouter le LCD au-dessus et toujours faire des circuits de l'autre côté. Les connexions 0, 1, 2, 3, GND et 3,3 V pour pinces crocodiles sont également ramenées, car ces trous ne sont pas accessibles quand la carte micro:bit est enfichée. Le LCD recouvre partiellement le 3,3 V et GND (**fig. 4**), mais vous pouvez encore mettre une pince dessus et les trois lignes d'E/S restent totalement accessibles par des fiches banane. Enfin, le brochage complet de la micro:bit est éclaté sur une embase mâle au pas de 2,54 mm en bas de la carte, vous pou-

Tableau 1. Brochage de la carte BBC micro:bit.

Broche micro:bit	Broche(s) du connecteur imprimé	Fonctions sur micro:bit	Fonctions (principales) sur dock	Fonctions (secondaires) sur dock	Commentaire
3.3V	10-13	3.3V	3.3V		J4 relie 3,3 V de micro:bit à 3,3 V du dock
GND	2-5	GND	GND		
P0	36-39	P0/AN	BUZZER	Accès P0	Déconnecté par R23
P1	28-31	P1/AN	TRIM1	Accès P1	Déconnecté par cavalier J3 ou R3
P2	19-22	P2/AN	SERVO	IR_OUT	SERVO déconnecté par R24 ; IR_OUT déconnecté par R25
P3	40	P3/LEDCOL1/AN			
P4	35	P4/LEDCOL2/AN			
P5	34	P5/BUTTONA	BUTTONA		Déconnecté par R26
P6	33	P6/LEDCOL9			
P7	32	P7/LEDCOL8			
P8	27	P8	BUTTONC	IR_IN	BUTTONC déconnecté par R27 ; IR_IN déconnecté par R28
P9	26	P9/LEDCOL7			
P10	25	P10/LEDCOL3/AN			
P11	25	P11/BUTTONB	BUTTONB		Déconnecté par R29
P12	23	P12/RESERVED	LCD_RST*		Déconnecté par R4
P13	18	P13/SCK1	LCD_BL_RED*		Déconnecté par R5
P14	17	P14/MISO1	LCD_BL_GRN*		Déconnecté par R6
P15	16	P15/MOSI1	LCD_BL_BLU*		Déconnecté par R7
P16	15	P16	BUTTOND		Déconnecté par R30
P17	14	P17/3.3V	3.3V		
P18	9	P18/3.3V	3.3V		
P19	8	P19/SCL1	LCD_SCL*		Déconnecté par R31
P20	7	P20/SDA1	LCD_SDA*		Déconnecté par R32
P21	6	P21/GND	GND		
P22	1	P22/GND	GND		

\*Seulement avec LCD optionnel.



vez ainsi ajouter un connecteur sur les broches dont vous avez besoin et brancher l'ensemble sur votre plaque d'essais. Nous avons conservé l'ordre des broches du connecteur imprimé avec en plus un contact 5 V.

## Assemblons le tout

Une fois toutes les fonctions nécessaires définies, il est aisé de dessiner le schéma puis de créer le circuit imprimé. Nous

avons fait très attention au brochage (cf. **tableau 1**) et exclu toutes les broches de la matrice de LED, car utiliser une de ces lignes aurait perturbé le fonctionnement de toutes les LED. La marge de manœuvre est donc réduite, c'est pourquoi nous avons dû doubler quelques fonctions.

## Alors, et les projets ?

Nous commençons à apprécier la carte

micro:bit (**fig. 5**) et codons de magnifiques projets pour la carte seule et maintenant pour le dock. Voici quelques sujets d'étude :

- **Laser game** : utilisez la fonction émission-réception IR pour « faire un carton sur vos amis ». Des tonnes d'options ici, mais nous attendons de voir la distance atteignable de façon fiable avec la relative basse puis-



## LISTE DES COMPOSANTS

### Résistances (1%, 0,1 W, 0603)

R3-R7, R17, R20, R23-R32 = 0 Ω  
 R19 = 0 Ω, pas montée  
 R16 = 100 Ω  
 R2, R9-R12, R33, R34 = 100 kΩ  
 R13, R14, R18, R21, R22 = 150 Ω  
 R15 = 49,9 kΩ (1206)  
 R1 = 698 kΩ  
 R8 = pot. de 10 kΩ, linéaire

### Condensateurs

C1, C2, C4, C5, C6 = 1 µF (0603)  
 C7, C8, C9 = n'importe quelle valeur, 1206, hauteur 0,85 mm

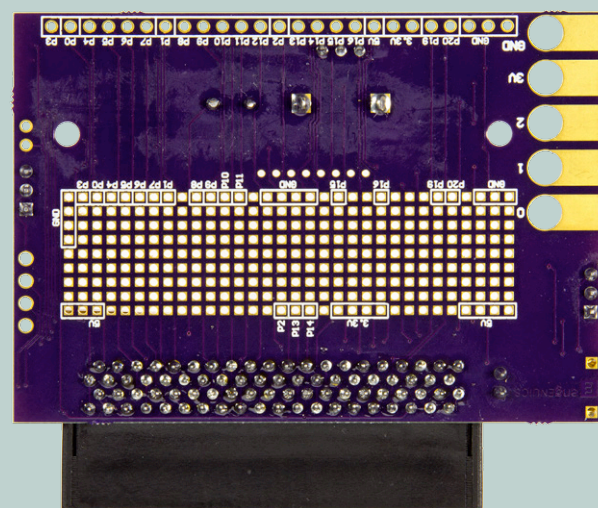
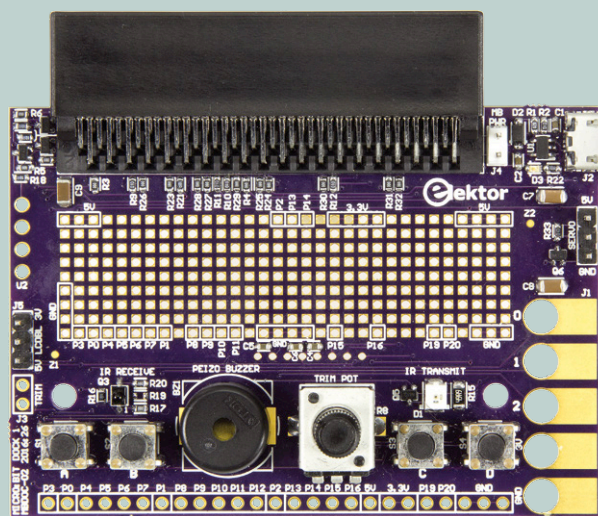
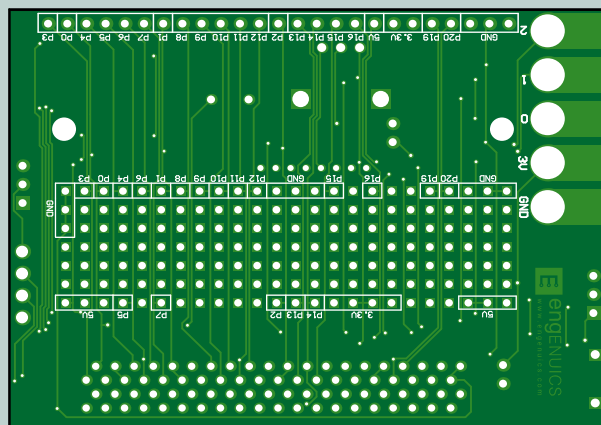
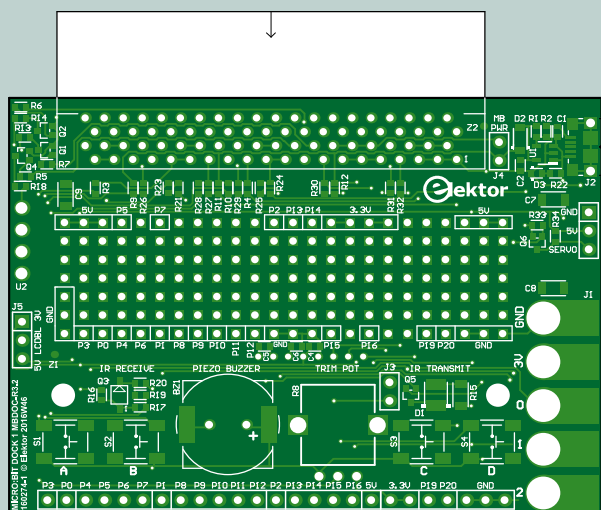
### Semi-conducteurs

D2 = 1N5819HW-7-F  
 D1 = IR67-21C/TR8 (diode IR)  
 U1 = MCP1824-ADJ/OT  
 U2 = LCD NHD-C0220BiZ-FS(RGB)-FB-3V NNP (en option)  
 D3 = LED, bleue, 0603  
 Q1, Q2, Q4, Q5, Q6 = DMG1012UW  
 Q3 = PT91-21B/TR7

### Divers

J6 = embase pour connecteur imprimé, 2×40, au pas de 1,27 mm  
 J7 = barrette mâle à 25 broches, au pas de 2,54 mm  
 J3, J4 = barrette mâle à 2 broches, au pas de 2,54 mm  
 J5, J8 = barrette mâle à 3 broches, au pas de 2,54 mm  
 J2 = connecteur USB micro-B  
 BZ1 = buzzer piézo  
 S1, S2, S3, S4 = bouton-poussoir, 6×6 mm

Circuit imprimé, réf. 160274-1





sance de la LED IR.

- Alarme à faisceau IR : sécurisez votre chambre avec un faisceau de lumière invisible qui incrémente un compteur ou déclenche une alarme à chaque passage !
- Commande d'éclairage : utilisez le potentiomètre et la connectivité BLE pour régler l'éclairage d'une pièce.
- Simple messagerie (avec LCD) : l'IU serait un peu difficile (imaginez composer caractère par caractère), mais vous pourriez envoyer des messages de carte à carte ou même de carte à téléphone. Vous pourriez coder vos messages standards favoris et les sélectionner avec les boutons.
- Réveil (LCD en option) : construisez votre propre réveil multifonctions.
- Buzzer énervant 1 : utilisez le potentiomètre et le buzzer pour composer n'importe quelle tonalité. Si vous avez un oscilloscope, observer le signal MLI est une excellente façon d'expliquer ce qu'est vraiment le son.
- Buzzer énervant 2 : réglez une tonalité haute fréquence et cachez

la carte dans le bureau d'un ami.

Déclenchez-le par BLE ou juste avec un temporisateur et regardez-le le chercher. Arrêtez-le quand il est trop près.

- Télécommande IR : utilisez votre micro:bit et son dock pour apprendre et programmer les principales fonctions de votre télécommande TV. Ou codez une communication carte à carte simple et réalisez des jeux ou fonctions sympathiques.

Si un de ces projets vous intéresse, dites-le-nous, nous nous y mettrons en priorité. Comme toujours, vos commentaires et suggestions sont les bienvenus. Bon accostage ! ◀

(160274 - version française : Denis Lafourcade)

### Lien

[1] [www.elektormagazine.fr/160274](http://www.elektormagazine.fr/160274)



## DANS L'E-CHOPPE

→ 160274-91

Module prêt à l'emploi

→ 17773

BBC micro:bit

→ 17589

Adaptateur 5 V AC/DC 2 A avec micro-USB

ou

→ 18012

Support pour 2 piles AA avec connecteur JST

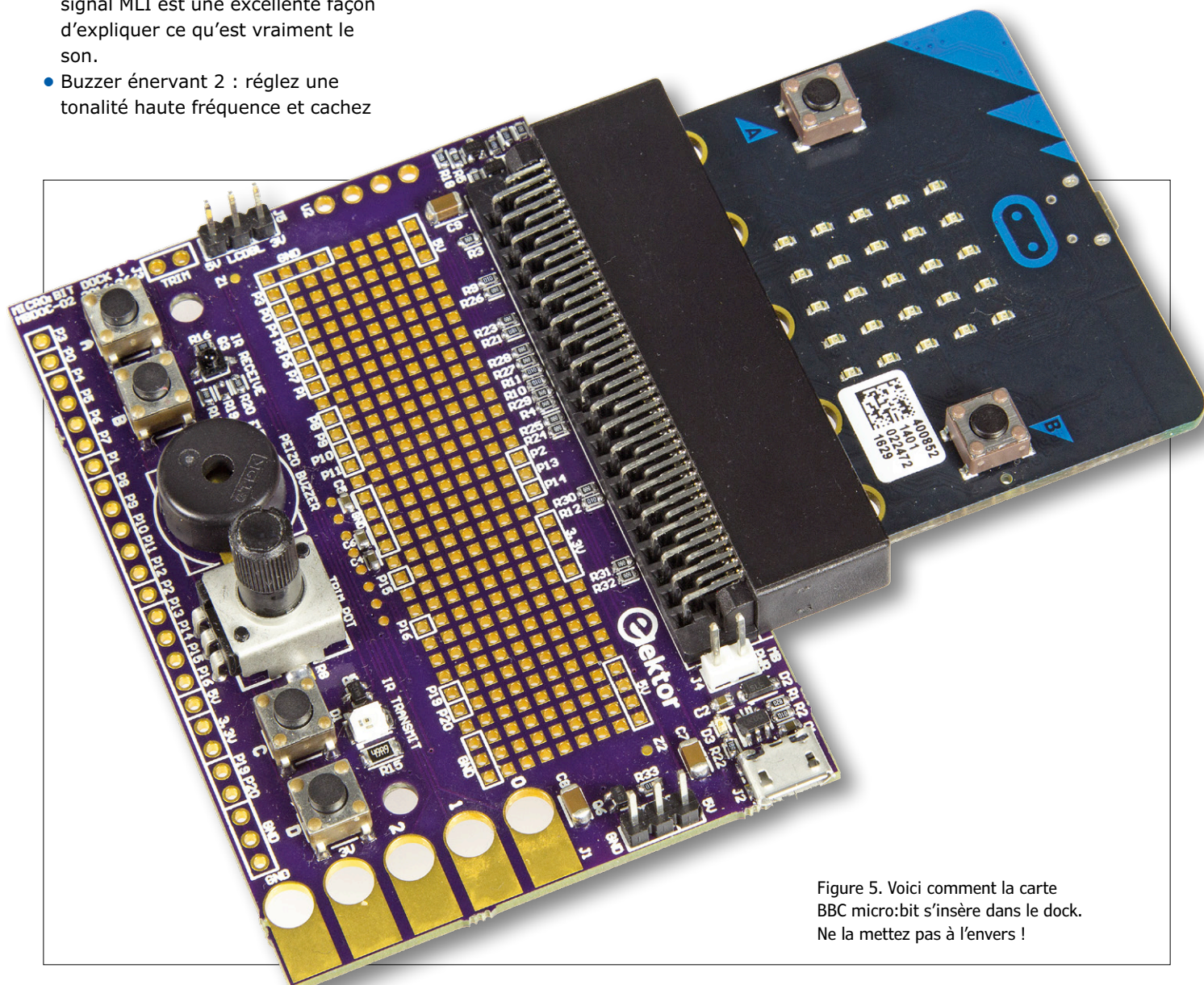


Figure 5. Voici comment la carte BBC micro:bit s'insère dans le dock. Ne la mettez pas à l'envers !



# le Fetron, un tube à transistors

## drôle de composant n°29

Il est brillant et tubulaire, 25 mm de haut et 15 mm de diamètre avec sept broches plaquées or à une extrémité. C'est manifestement un composant actif, mais est-il à vide ou à semi-conducteur ? Un peu des deux à la fois : c'est bien un tube, mais à semi-conducteurs, c'est un Fetron

**Andrew Emmerson** (Royaume-Uni)

En réalité, c'est un substitut de tube réalisé avec des transistors JFET (jonction à effet de champ) pour coïncider avec les caractéristiques de transfert de tubes à vide particuliers, principalement des pentodes, mais il y a eu quelques triodes aussi. Présenté par Teledyne Semiconductor en 1972 et destiné à un usage militaire ou à des appareils « indispensables » (pas pour le commun des mortels), le Fetron permettait d'allonger la durée de vie d'appareils très chers à tubes, en repoussant leur remplacement. Comme les Fetrons sont des semi-conducteurs, ils consomment moitié moins de puissance que les tubes et se passent de climatisation. Ils éliminent les réglages fréquents et le remplacement périodique des tubes, donc, malgré leur prix exorbitant, ils étaient rentabilisés dans les six mois après installation.

Teledyne a développé le Fetron pour le remplacement par insertion directe de certains tubes à vide et c'est ce qui causa leur faiblesse. Le grand spécialiste des Fetrons, le Dr Hugo Holden du Queensland en Australie, explique que les vendre comme substitut des tubes était une grave erreur. Dans de nombreuses applications, comme l'oscillateur local qui compte sur le courant de grille pour sa polarisation automatique, il fonctionne mal et dévie le courant de grille, ce qui cause la saturation et introduit de forts harmoniques à la sortie.

Néanmoins, un « tube » qui ne dégage pas de chaleur, ne dérive pas et ne consomme pas de courant de filament, était une solution séduisante. Les compagnies de téléphone lui ont donné un éclat particulier et les exemples de la **figure 1** viennent de Nouvelle-Zélande où ils ont été utilisés dans les amplificateurs d'attaque des câbles téléphoniques sous-marins vers l'Australie. Ils ont aussi servi dans les appareils de haute fiabilité pour la radiodiffusion et les amplis de guitare (!) pendant que Teledyne prospectait avec ses kits de conversion pour améliorer les voltmètres électroniques et les oscilloscopes chez Hewlett Packard et Tektronix, pour assurer même stabilité, faible bruit et fiabilité que les transistors bipolaires (**figure 2**). Le passage aux semi-conducteurs a écourté la carrière du Fetron, cet ingénieux composant ne laisse que quelques souvenirs. ◀

(160265 – version française : Robert Grignard)

### Liens

<https://en.wikipedia.org/wiki/Fetron>

<http://hrsasa.asn.au/docs/Fetron.pdf>

[www.philipstorr.id.au/radio/eleven/fetron.htm](http://www.philipstorr.id.au/radio/eleven/fetron.htm)

[www.worldphaco.com/uploads/WORLDFETRON.pdf](http://www.worldphaco.com/uploads/WORLDFETRON.pdf)

[www.elektormagazine.fr/160049](http://www.elektormagazine.fr/160049)

Envoyez-moi vos idées en quelques lignes ou sous forme d'un article déjà rédigé : [neil@gruending.net](mailto:neil@gruending.net)

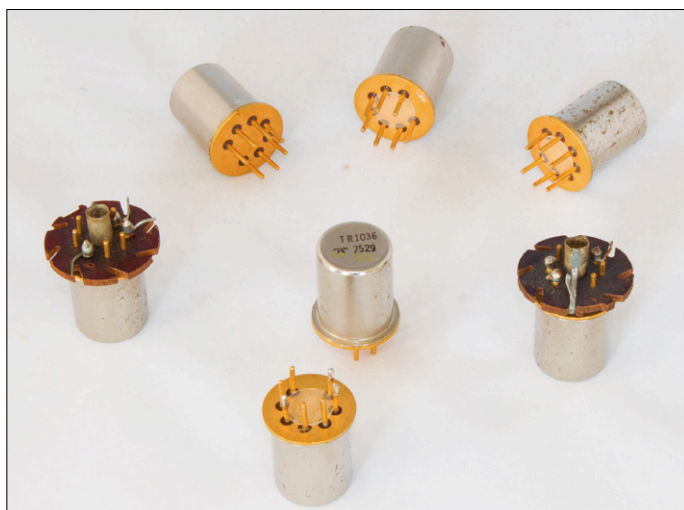


Figure 1. Des Fetrons sens dessus dessous.



Figure 2. Publicité de l'époque du Fetron.

# commande de vitesse par MLI avec régulation du rapport cyclique

Rolf Blijleven (Pays-Bas)

Le but de ce circuit est de commander un moteur à courant continu qui tourne lentement. La tension d'alimentation du moteur est un train d'impulsions modulées en largeur (MLI), dont le rapport cyclique augmente proportionnellement à la charge du moteur.

Pour certaines applications, on utilise un moteur à courant continu qui tourne à vitesse réduite. La solution classique est d'alimenter le moteur par un train d'impulsions, avec un rapport cyclique faible ; la tension moyenne est par conséquent relativement basse, et le moteur peut se bloquer si la charge augmente. Une alternative est de modifier le rapport cyclique, c'est-à-dire moduler la largeur des impulsions (MLI).

Rien de neuf sous le soleil ! De tels circuits sont légion. Toutefois nous voulons une

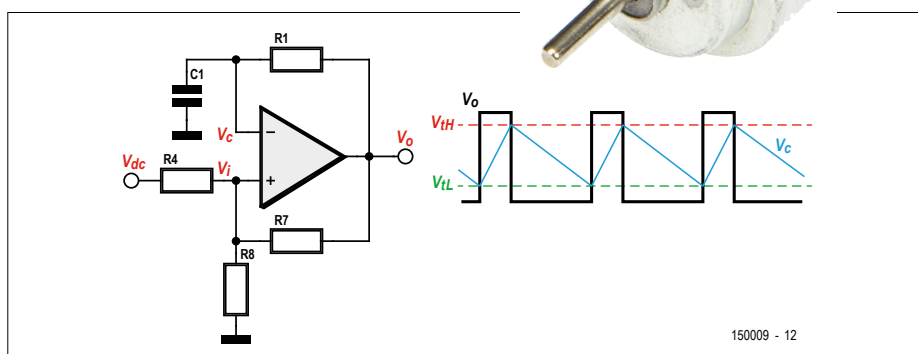
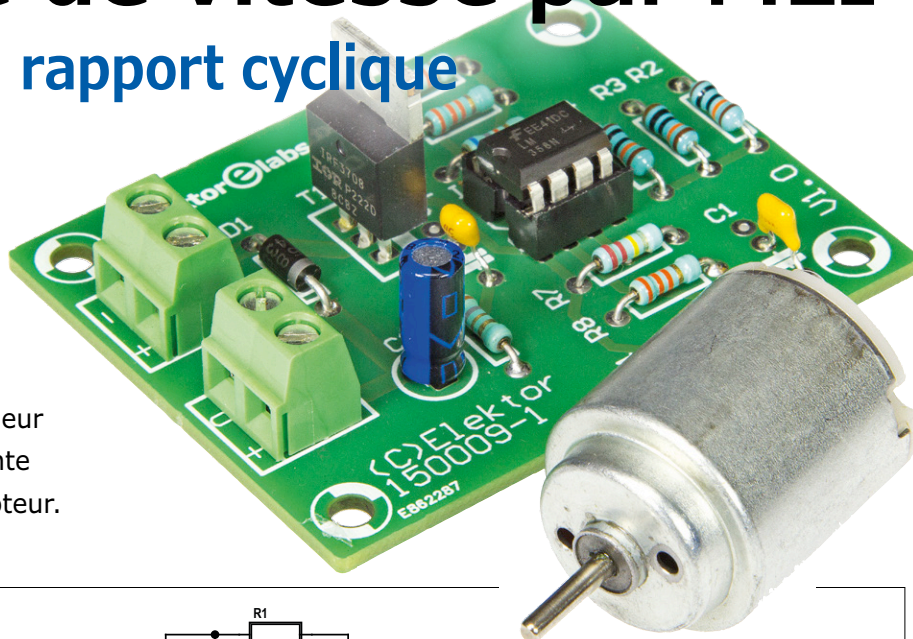


Figure 1. Schéma de principe d'un générateur d'impulsions avec un seul ampli op.

variation automatique du rapport cyclique avec un circuit simple (sans microcontrôleur) ! On prend un générateur de signaux à MLI pour alimenter le moteur, on mesure le courant d'alimentation, et on utilise le résultat pour réguler la MLI. Quelques amplis op, un MOSFET pour la commande du moteur, des résistances et des condensateurs, le tout alimenté sous 5 V, ça devrait suffire. Le circuit est une extension d'une application typique du LM358 (**fig. 1**), tirée d'une feuille de caractéristiques de Texas Instruments [1].

## Principe

Vous connaissez sans doute le principe de fonctionnement du circuit (**fig. 1**) ; sinon cherchez « rapport cyclique » (*duty cycle*) sur Wikipedia [2]. À la mise sous tension, l'entrée + de l'ampli op ( $V_i$ ) est positive ( $V_{dc} \leq$  tension d'alimentation), tandis que la tension à l'entrée - ( $V_c$ ) est nulle. La

sortie de l'ampli op sature à sa tension maximale ( $V_{o,max}$ ). Cette tension charge C1 via R1, jusqu'à ce que  $V_c$  soit supérieure à  $V_i$  ; nous appelons cette tension  $V_{th}$ . La tension de sortie passe alors à 0, et C1 se décharge via R1 jusqu'à ce que  $V_c$  redevienne inférieure à  $V_i$  (tension  $V_{tl}$ ). La tension de sortie repasse à son maximum, et le cycle recommence. C'est donc une combinaison entre un générateur de dents de scie et une bascule de Schmitt.

Quelques formules permettent de décrire formellement ce comportement et de déterminer la valeur des composants pour un cycle donné. La tension  $V_{o,max}$  peut atteindre la tension d'alimentation avec un ampli op à étage de sortie CMOS comme le CA3130, mais ne peut s'en approcher à moins de 1,5 V avec un 741. Continuons avec l'approche intuitive. Plus  $V_{dc}$  est élevée, plus  $V_i$  le sera, et donc le

### À PROPOS DU PROJET

loisirs

moteur MLI rapport cyclique

débutant

→ connaisseur expert

environ 30 min

Moteur à courant continu de récupération (lecteur de CD/DVD)

environ 15 €



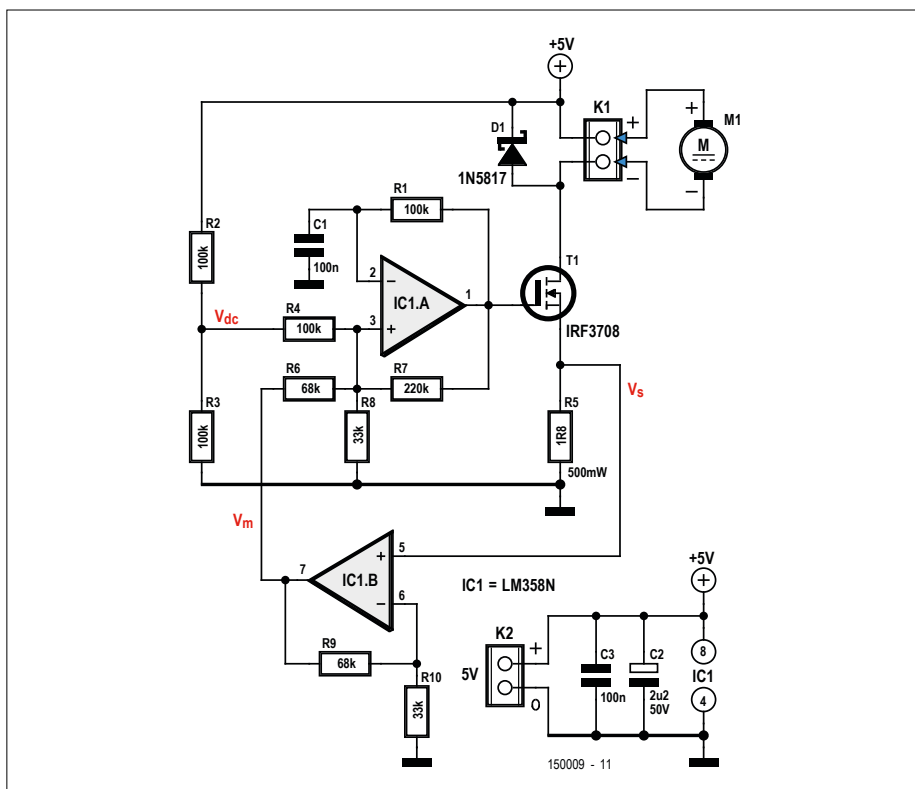


Figure 2. Le générateur IC1A commande le moteur via le MOSFET T1 ; IC1B et R5 (capteur de courant) assurent une certaine contre-réaction.

rapport cyclique aussi. Si  $V_i$  est toujours supérieure à  $V_c$ , le rapport cyclique est de 100%, c'est-à-dire une tension continue en sortie.  $V_i$  est le paramètre prédo-

minant dans le fonctionnement du circuit. Pour mieux comprendre, raisonnons en termes de conductance ( $1/R$ ) plutôt que de résistance ( $R$ ) :

- plus  $R4$  est élevée (plus la conductance  $V_{dc} \rightarrow V_i$  est faible ; faible contribution de  $V_{dc}$ ), plus le rapport cyclique est faible ;
- plus  $R8$  est faible (plus la conductance  $V_i \rightarrow$  masse est élevée), plus le rapport cyclique est faible ;
- plus  $R7$  est faible, plus l'influence de  $V_{o\max}$  sur  $V_{th}$  est grande, et plus la différence entre  $V_{th}$  et  $V_{tl}$  est grande, mais comme c'est la somme de ces deux tensions qui importe, l'influence de  $R7$  sur le rapport cyclique est bien moindre que celle de  $R4$  et de  $R8$ .

Il faut choisir des valeurs de  $R4$ ,  $R7$  et  $R8$  qui fixent un rapport cyclique suffisamment élevé pour faire tourner le moteur. Cependant nous voulons démarrer avec le rapport cyclique le plus bas possible, puis l'augmenter automatiquement si nécessaire, en fonction de la charge du moteur.

## Schéma

Le schéma complet du circuit est en **figure 2**. La mise au point n'est pas si simple qu'il y paraît, mais nous y reviendrons plus tard.

La tension  $V_{dc}$  est fixée à la moitié de la tension d'alimentation par  $R2$  et  $R3$ . Le signal MLI à la sortie d'IC1A commande le moteur via le transistor MOSFET de puissance T1. La résistance  $R5$  de  $1,8 \Omega$  mesure le courant d'alimentation du moteur ; son intensité est d'environ 450 mA dans notre cas et en fonctionnement normal, ce qui nous donne une tension  $V_s$  de 0,8 V aux bornes de  $R5$ . Si la charge augmente, le courant grimpe à 700 mA, et la tension à 1,2 V. La puissance à dissiper par  $R5$  serait donc de 0,9 W ( $I^2R = (0,7 A)^2 \times 1,8 \Omega = 0,9 W$ ), mais comme le fonctionnement est en impulsions, la puissance à dissiper est réduite par le rapport cyclique et n'est que d'environ 20% de sa valeur en continu ; une résistance prévue pour une dissipation de 0,5 W ne risque pas de s'échauffer.

La tension  $V_s$  est amplifiée par IC1B, dont la tension de sortie  $V_m$  se retrouve à l'entrée + d'IC1A via  $R6$ .  $V_i$  augmentera donc lorsque la charge du moteur croîtra. Ne crions pas trop vite victoire, car il y a bien quelques écueils, notamment le fait que la fréquence du signal MLI diminue lorsque le rapport cyclique augmente. C'est flagrant si on examine la formule

### LISTE DES COMPOSANTS

**Résistances**  
(5% / 0,25 W, sauf indication contraire)  
 $R1-R4 = 100 \text{ k}\Omega$   
 $R5 = 1,8 \Omega$ , 0,5 W  
 $R6, R9 = 68 \text{ k}\Omega$   
 $R7 = 220 \text{ k}\Omega$   
 $R8, R10 = 33 \text{ k}\Omega$

**Condensateurs**  
 $C1, C3 = 100 \text{ nF}$   
 $C2 = 2,2 \mu\text{F} / 16 \text{ V}$ , pas de 2 mm

**Semi-conducteurs**  
 $D1 = 1N5817$   
 $T1 = \text{IRF3708PBF}$   
 $IC1 = \text{LM358}$

**Divers**  
 $K1, K2$  = bornier à vis, 2 contacts, pas de 5,08 mm  
Moteur à courant continu avec réducteur (p. ex. récupéré dans un lecteur de CD/DVD d'ordinateur)  
Circuit imprimé, réf. 150009-1

du rapport cyclique ; les temps de charge ( $t_1$ ) et de décharge ( $t_2$ ) du condensateur C1 sont déterminants.

$$\text{rapport cyclique} = \frac{t_1}{t_1 + t_2} \cdot 100\%$$

Le rapport cyclique augmente proportionnellement au temps de charge, mais la fréquence  $[=1/(t_1 + t_2)]$  est inversement proportionnelle à cette durée. Si le rapport cyclique augmente, la fréquence diminue, et par conséquent la vitesse de rotation du moteur (que nous voulons garder constante). Si on donne plus d'influence à  $V_m$  en diminuant R6, le rapport cyclique augmente, mais la fréquence diminue ; c'est un cercle infernal !

Une constante de temps  $R1 \times C1$  plus faible n'est pas une solution non plus, car le moteur émet un bruit inquiétant lorsque la fréquence dépasse 75 Hz, bruit qui devient un crissement aigu au-delà de 120 Hz.

Nota : la tension aux bornes de R5 est une tension continue à laquelle se superpose une ondulation, due aux commutations des balais du moteur. Si la charge augmente, la tension augmente, mais

la fréquence de l'ondulation diminue, le moteur est ralenti.

Enfin, il faut signaler que des fractions de  $V_m$  et  $V_{dc}$  s'additionnent à l'entrée + d'IC1A. Il s'agit donc d'une réaction positive, ce qui d'après les manuels n'est pas une bonne idée ; entre autres parce que l'impédance des sources de tension a une influence sur la sommation. Ce rôle est négligeable dans le cas présent : l'impédance de sortie d'IC1B est beaucoup plus petite que R6, et l'impédance du diviseur de tension R2/R3 est de 50 k $\Omega$ , la moitié de R4. Nous avons expérimenté avec un suiveur de tension pour diminuer l'impédance du diviseur de tension, mais cela n'a eu presque aucun effet.

Ah ! Si nous avions un tableur avec les valeurs tirées des formules du fonctionnement du circuit de la **figure 1**, et celles issues des phénomènes liés au circuit de réaction bâti autour d'IC1B... Nous préférons l'expérimentation, c'est à cela que servent les platines éponymes !

Nous avons finalement obtenu un résultat satisfaisant avec les valeurs du circuit de la **figure 2** : un rapport cyclique d'environ 17% pour une fréquence de 55 Hz, allant jusqu'à 21% et une fréquence

de 45 Hz pour la charge la plus élevée du moteur. C'est un résultat acceptable, et comme le mieux est l'ennemi du bien, nous en sommes restés là.

Le labo d'Elektor a dessiné un petit circuit imprimé (**fig. 3**). C'est pratique, mais n'oubliez pas que vous devrez peut-être revoir la valeur de certaines résistances, en fonction du moteur utilisé. Pour obtenir une vitesse absolument constante, il faut sans doute se résoudre à utiliser un microcontrôleur ; ce sera pour une autre fois ! **◀**

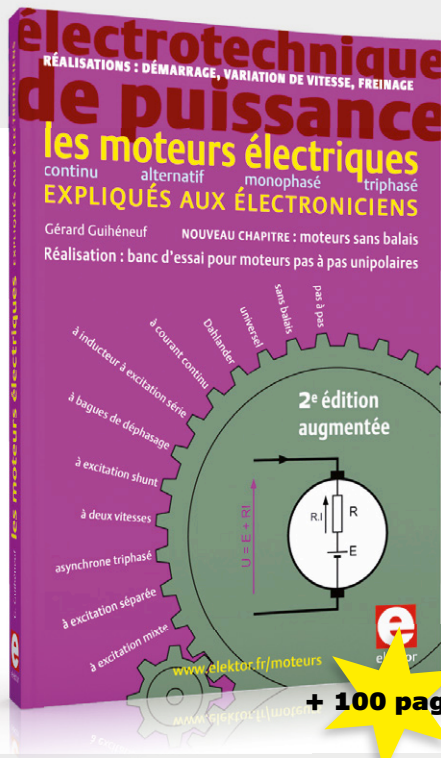
(150009 – version française : Jean-Louis Mehren)

## Liens

- [1] [www.ti.com/lit/ds/symlink/lm158-n.pdf](http://www.ti.com/lit/ds/symlink/lm158-n.pdf)
- [2] [en.wikipedia.org/wiki/Duty\\_cycle](http://en.wikipedia.org/wiki/Duty_cycle) et [fr.wikipedia.org/wiki/Rapport\\_cyclique](http://fr.wikipedia.org/wiki/Rapport_cyclique)



Publicité



## Nouvelle édition augmentée

**100 pages supplémentaires !**

## les moteurs électriques


expliqués aux électroniciens

Ce livre en quatre chapitres offre une information accessible et digeste : constitution, fonctionnement, caractéristiques, domaines d'utilisation, pour proposer aussi des réalisations électroniques simples et concrètes.

Le premier détaille les principes de variation de la vitesse des moteurs à courant continu. Les moteurs à alimentation alternative monophasée du 2<sup>e</sup> chapitre font appel à une électronique de puissance. Dans le 3<sup>e</sup> chapitre sont décrits les moteurs asynchrones triphasés, le moteur électrique le plus utilisé dans l'industrie.

Cette nouvelle édition s'enrichit d'un chapitre consacré aux moteurs sans balais (brushless), du moteur synchrone triphasé de plusieurs centaines de kW au moteur à courant continu de quelques centaines de watts, en passant par les moteurs pas à pas ou encore le surprenant moteur linéaire. Le lecteur découvrira leurs modes de commande : codeurs incrémentaux ou absolus associés à un onduleur commandé en courant ou en tension ou bien capteurs à effet Hall pour l'autopilotage, commande en pas entiers, demi-pas, micro-pas... L'auteur propose également de réaliser un banc d'essai pour moteurs pas à pas unipolaires.

ISBN 978-2-86661-194-1 – 432 pages – 46,50 €

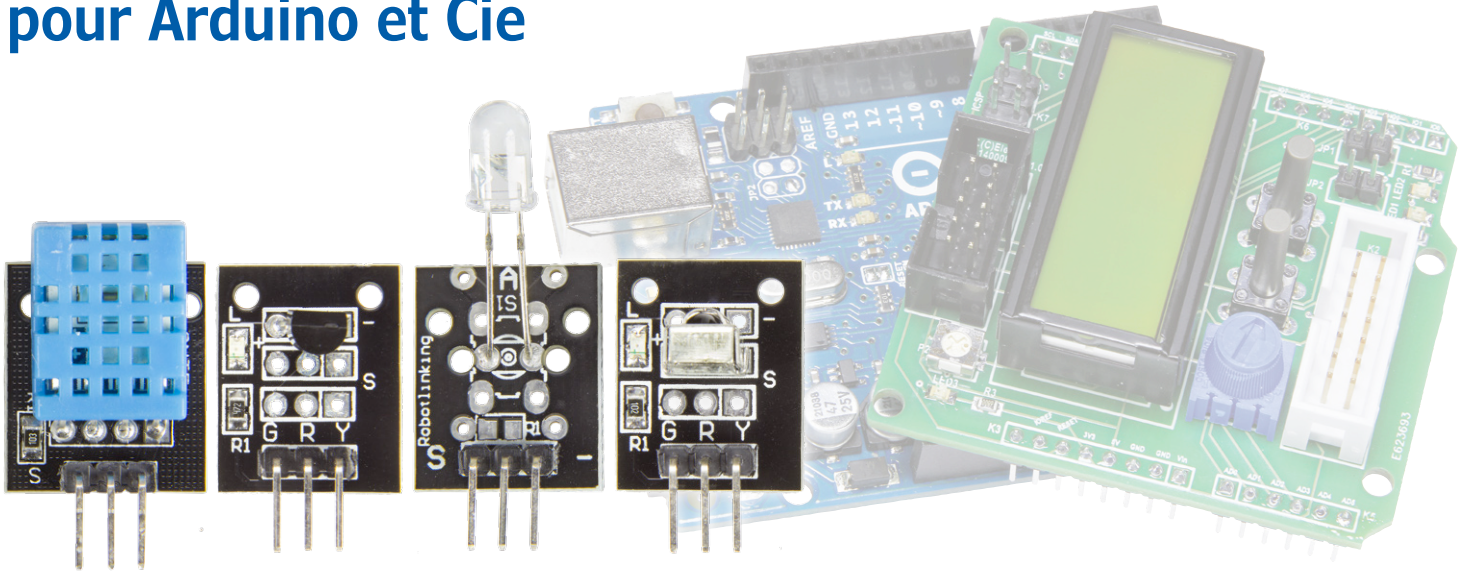


[www.elektor.fr/moteurs](http://www.elektor.fr/moteurs)



# capteurs (3)

## pour Arduino et Cie



Burkhard Kainka

Depuis les débuts de la télégraphie, la communication électrique passe par un seul fil. Le bus *1-wire* et bien d'autres protocoles se servent eux aussi d'un unique fil, et se passent de ligne d'horloge. Les télécommandes IR fonctionnent selon un principe similaire. Nous nous pencherons ici, entre autres sur le capteur de température et d'humidité ainsi que l'émetteur-récepteur IR du kit de 35 capteurs.

Nous commençons par le capteur de température « 18B20 Temp » issu du kit disponible dans l'e-choppe d'Elektor [1]. Le capteur est doté d'un DS18B20 qui a l'air d'un transistor ordinaire logé dans un boîtier TO92, mais en fait il s'agit d'un circuit intégré complexe avec une interface particulière. Le bus *1-wire* a été conçu par l'entreprise Dallas. À l'exception du fil de masse (GND), il suffit d'un seul fil pour piloter un ou plusieurs capteurs. Non seulement le flux de données, mais aussi l'alimentation du circuit passent par ce fil. Cependant,

la ligne VDD du DS18B20 est généralement utilisée pour l'alimentation, de sorte qu'on dispose en fait de trois fils au total. Le circuit imprimé du capteur est doté en plus d'une LED avec sa résistance série sur la ligne de données.

### Logiciel Arduino pour le 18B20

Tout d'abord, il convient d'inclure deux bibliothèques dans l'EDI Arduino : *OneWire* et *DallasTemperature*. Les deux se trouvent sur le CD fourni avec le kit de capteurs. Il faut les copier dans

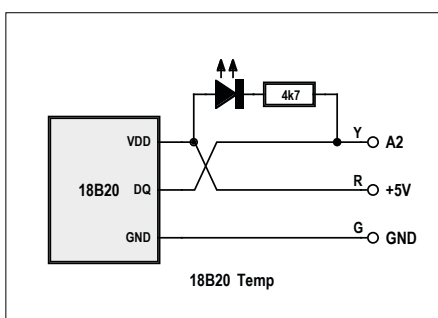


Figure 1. Connexion du capteur de température 18B20.

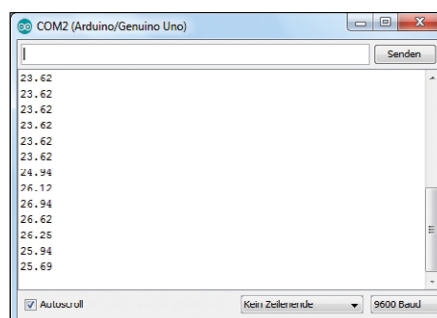


Figure 2. Affichage de la température dans le moniteur série.

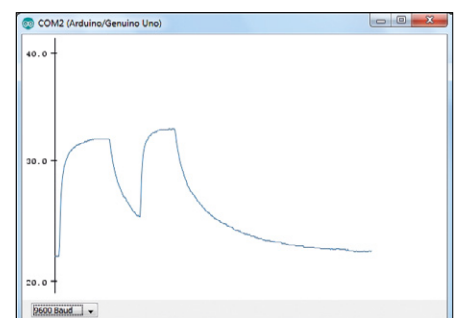


Figure 3. Courbe de température.

### Listage 1. Mesure de température avec le DS18B20

```
#include <OneWire.h>
#include <DallasTemperature.h>
#include <LiquidCrystal.h>
#define ONE_WIRE_BUS A2
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

float temp;
int minTemp;
int maxTemp;
LiquidCrystal lcd(2,3,4,5,6,7);

void setup(void)
{
  Serial.begin(9600);
  sensors.begin();
  lcd.begin(16, 2);
  minTemp = 100;
  maxTemp = -100;
```

```
}

void loop(void)
{
  sensors.requestTemperatures();
  temp = sensors.getTempCByIndex(0);
  Serial.println(temp);
  lcd.setCursor(0, 0);
  lcd.print(temp);
  lcd.print(" C ");
  if (temp < minTemp) minTemp = temp;
  if (temp > maxTemp) maxTemp = temp;
  lcd.setCursor(0, 1);
  lcd.print(minTemp);
  lcd.setCursor(5, 1);
  lcd.print(maxTemp);
  delay (500);
}
```

le dossier « libraries » du répertoire de croquis d'Arduino. On y trouve respectivement un fichier d'en-tête \*.h et un fichier en C++ \*.cpp. Je pense qu'il est intéressant d'examiner ces fichiers. Toutefois si vous n'y comprenez rien, ce n'est pas grave. En effet, il s'agit uniquement de reprendre les fichiers d'en-tête pour les mettre dans notre propre programme afin de n'avoir plus qu'à faire appel à quelques fonctions. Avec Arduino, c'est facile comme bonjour. Le programme du **listage 1** montre comment la température est interrogée toutes les 500 ms. La ligne de données est raccordée à A2 (= PC2) (**fig. 1**). Bien sûr, tous les codes liés à cet article peuvent être téléchargés sur le site d'Elektor [4].

Le thermomètre dispose non seulement d'une sortie série, mais il permet également d'afficher la température sur l'écran à cristaux liquides, à condition d'utiliser le *shield* d'extension Elektor [2] [5]. Comme il y avait de la place sur la deuxième ligne du LCD, le thermomètre affiche aussi les températures minimale et maximale.

La température affichée est une valeur réelle avec deux décimales. La résolution effective est de 0,06 °C par pas. La précision absolue est indiquée avec 0,5 °C. Les données peuvent être observées sur le moniteur série (**fig. 2**). Pour modifier la température, touchez le capteur.

Le traceur série montre la courbe de température (**fig. 3**). Durant cette mesure, le capteur a été réchauffé deux fois avec un doigt. On voit clairement que les constantes de temps diffèrent au moment du réchauffement et du refroidissement. Il est intéressant de constater que le doigt était plus chaud lors du deuxième contact. Quelque chose s'est donc passé entre ces deux mesures, ce qui a provoqué l'augmentation de la température. La haute résolution du capteur permet de détecter les moindres changements.

### 18B20 en Bascom

Certes Bascom permet d'utiliser le bus 1-wire, mais nous n'avons toujours pas de solution complète pour le 18B20.

Regardons de plus près la fiche technique et ajoutons quelques lignes au programme. C'est du travail, mais cela nous ouvre de nouvelles perspectives. En effet, nous pourrions nous servir des propriétés spéciales du capteur, p. ex. pour modifier la résolution ou lire le numéro d'identification unique contenu dans le circuit intégré.

L'exemple en Bascom (**listage 2**) montre comment lire la température. Il faut envoyer deux commandes ([Skip ROM](#), &HCC et

### Le protocole 1-wire

Avec 1-wire, tout passe par un seul fil : la ligne DQ. Enfin presque, puisque nous avons quand même GND et VCC. Au repos, la ligne de données DQ est ramenée à l'état haut à l'aide d'une résistance de rappel. Le dispositif maître peut alors envoyer une impulsion de remise à zéro et démarrer ainsi une communication, afin de pouvoir envoyer des instructions ou recevoir des données. Les deux partenaires peuvent envoyer des informations sur le bus. Un bit 0 est représenté par une impulsion à l'état bas d'une durée de 15 µs suivie d'un état haut d'une durée de 45 µs. Un bit 1 est symbolisé par une impulsion à l'état bas d'une durée de 60 µs. Entre les différents bits, il y a un état de repos durant lequel la ligne de données est ramenée à l'état haut.

Le contrôleur envoie à chaque fois une remise à zéro et ensuite une ou plusieurs instructions. Une puce de capteur répond en envoyant les données requises. Si on regarde attentivement la fiche technique de la puce, on y trouve non seulement le protocole général du bus, mais aussi les nombreuses instructions ainsi que la structure des données en retour. Les choses se compliquent, car on voit qu'il est possible d'avoir plusieurs esclaves sur le même bus. On se doute que la programmation ne va pas être du gâteau. Mais on n'est pas obligé de réinventer la roue, il doit bien y avoir du code quelque part.



### Listage 2. Mesure de température en Bascom (extrait)

```
'DS18B20LCD AD2, PORTC.2
...
Do
  lwreset
  lwwrite &HCC
  lwwrite &H44
  Waitms 800
  lwreset
  lwwrite &HCC
  lwwrite &HBE
  Dat(1) = lread(2)
  lwreset
  Temp = 256 * Dat(2)
  Temp = Temp + Dat(1)
```

```
Temp = Temp * 0.0625
```

```
Print Temp
  Tempint = Round(temp)
  If Tempint > Maxtemp Then Maxtemp = Tempint
  If Tempint < Mintemp Then Mintemp = Tempint
  Locate 1 , 1
  Lcd Temp ; " C  "
  Locate 2 , 1
  Lcd Mintemp
  Locate 2 , 5
  Lcd Maxtemp
  Waitms 200
Loop
End
```

Convert T, &H44). Après un temps d'attente pour la mesure proprement dite, on envoie la commande [Read Scratchpad](#) (&HBE) et ensuite on lit deux octets. Ceci permet de calculer un nombre de 16 bits et la température par pas de 0,0625 °C. Finalement, ce n'est pas si compliqué, même si le programme est un peu plus long que celui écrit pour l'Arduino. Une fois de plus, nous avons un thermomètre à minimum et maximum avec un afficheur à cristaux liquides.

### Détection de température et humidité de l'air avec le DHT11

À première vue, le détecteur de température et d'humidité DHT11 ressemble à un simple capteur d'humidité résistif. La présence d'une broche analogique renforce cette impression (**fig. 4**). Mais en fait derrière ce boîtier bleu se cache un capteur complexe avec une interface numérique.

L'entreprise chinoise Aosong est allée au-delà de ce qui ressemble, au premier abord, au bus 1-wire de Dallas. Chaque

mesure est déclenchée par le maître avec une impulsion à l'état bas d'une durée d'au moins 18 ms. Ensuite, le maître demande la lecture de 40 bits, avec une impulsion à l'état bas de 80 µs. Le capteur répond par des impulsions à l'état haut ; pour un 0, on a une durée maximale de 28 µs et pour un 1 on a 70 µs. Les 40 bits contiennent respectivement un octet haut et un octet bas pour l'humidité et la température, et un octet de parité supplémentaire pour vérifier que la transmission est correcte. Avec le DHT11, les octets bas sont tous à 0, si bien que les décimales ne sont pas transmises. Le modèle DHT22 lui transmet les décimales tout en utilisant le même protocole. La bibliothèque permet de lire les données fournies par ces deux modèles.

Et une fois de plus, nous sommes chanceux, car quelqu'un a pris la peine de transposer ce protocole compliqué dans une bibliothèque Arduino. Le DHT11 est très facile à utiliser dès que la bibliothèque DHT a été copiée dans le dossier de bibliothèques Arduino. Le **listage 3** montre la lecture de ces deux

### Listage 3. DHT11 en langage C Arduino

```
//DHT11LCD, pin AD2

#include <dht.h>
#define dht_apin A2
#include <LiquidCrystal.h>
LiquidCrystal lcd(2,3,4,5,6,7);
float temperature;
float humidity;

dht DHT;

void setup(){
  Serial.begin(9600);
  delay(500);
  delay(1000);
  lcd.begin(16, 2);
}
```

```
void loop(){
  DHT.read11(dht_apin);
  humidity = DHT.humidity;
  temperature = DHT.temperature;
  Serial.print("Humidity = ");
  Serial.print(DHT.humidity);
  Serial.println(" % ");
  Serial.print("Temperature = ");
  Serial.print(temperature);
  Serial.println(" C ");
  lcd.setCursor(0, 0);
  lcd.print(temperature);
  lcd.print (" C  ");
  lcd.setCursor(0, 1);
  lcd.print(humidity);
  lcd.print (" %  ");
  delay(2000);
}
```

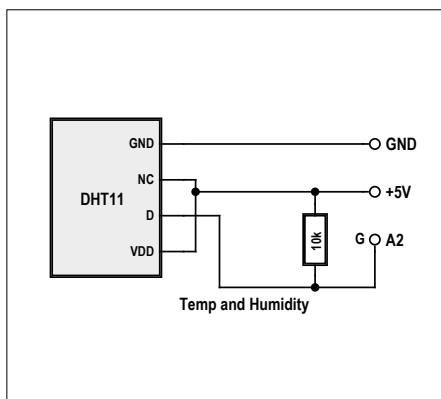


Figure 4. Connexion du capteur de température/humidité DHT11.

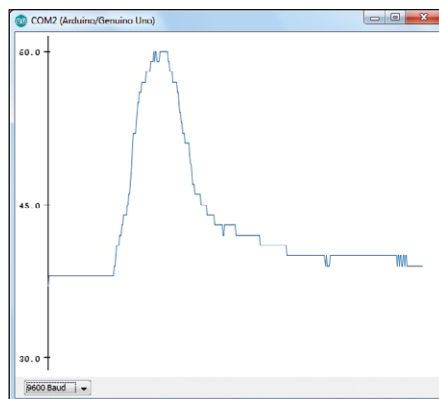


Figure 5. Mesure de l'humidité de l'air au contact du doigt.

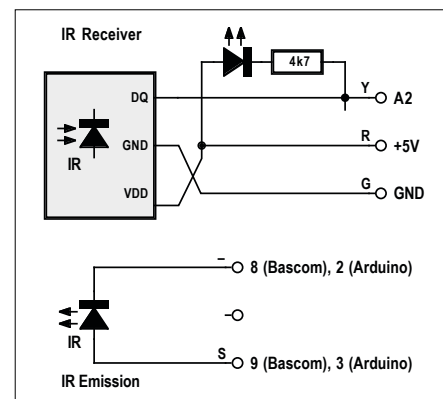


Figure 6. Émetteur IR et récepteur IR.

valeurs de mesure et leur affichage sur le LCD. La ligne de données est toujours connectée à la broche AD2.

Selon la fiche technique, la mesure de la température a une précision absolue de  $\pm 2^\circ\text{C}$ , celle de l'humidité de l'air de 5 %. L'humidité de l'air est difficile à déterminer, car la plupart des hygromètres ne livrent que des ordres de grandeur. Les choses ont été différentes avec la température, en tout cas avec notre exemplaire. Notre thermomètre numérique nous a indiqué  $23,8^\circ\text{C}$ , tandis que le DHT11 mesurait  $24^\circ\text{C}$  et le DS18B20 a mesuré  $23,37^\circ\text{C}$ , même s'il nous a fallu attendre longtemps après avoir touché le capteur.

### DHT11 et Bascom

Certes, Bascom dispose de nombre d'instructions et de fonctions toutes prêtes, mais on ne peut pas utiliser le DHT11 directement. Comme d'habitude, une recherche sur l'internet ne fera pas de mal. Qui sait, peut-être que quelqu'un a déjà travaillé sur le sujet ? Bingo ! Sur le forum Bascom, je trouve le programme DHT11LCD d'un utilisateur nommé Grütze, qui fait exactement ce que je veux. Le code est facile à lire et reprend exactement les données fournies dans la fiche technique. Il suffit de procéder à quelques modifications pour qu'il convienne au *shield* d'extension et pour que le capteur puisse être connecté à la broche AD2. J'en profite pour y ajouter une sortie série pour les données relatives à l'humidité parce que j'ai l'intention de les représenter dans le traceur série de l'EDI Arduino (**fig. 5**).

L'essentiel se passe dans la fonction `Get_dht11()` qui exécute à la lettre ce qui se trouve dans la bibliothèque correspondante pour l'Arduino. Voilà de quoi réaliser une station météo complète également en Bascom (**listage 4**).

### Télécommande IR

Vous utilisez tous des télécommandes IR avec vos téléviseurs et autres appareils électroniques domestiques. Il existe une multitude de fabricants et de protocoles (pas compatibles entre eux). C'est pourquoi la télécommande doit être adaptée à l'appareil. Toutefois, toutes les télécommandes ont un point commun : le signal d'une diode émettrice est modulé avec une fréquence comprise entre 30 et 40 kHz. Ce signal est ensuite commuté conformément à un certain modèle afin d'envoyer des paquets

d'impulsions. Un récepteur IR intégré reçoit les signaux via une photodiode, les amplifie, les filtre et les démodule pour obtenir un signal numérique. Certes, les filtres internes sont conçus pour une certaine fréquence dans une plage comprise entre 30 et 40 kHz, mais la bande passante suffit pour recevoir les signaux d'autres télécommandes qui se trouvent à proximité. Une télécommande IR peut être utilisée pour des tâches de commutation et de télécommande générales. Le kit de capteurs contient une diode émettrice IR (*IR Emission*) et un récepteur IR (*IR Receiver*, **fig. 6**). Une carte Arduino permet de recevoir ou d'envoyer des signaux IR. Notre programme fait les deux. Deux touches déclenchent l'envoi d'ordres qui sont ensuite analysés par le récepteur pour commander une sortie. Nous réaliserons cette tâche une fois en Bascom et une fois en langage C Arduino. Les deux programmes seront « compatibles » : le contrôleur Bascom pourra dire à la carte Uno programmée en Arduino ce qui doit être commuté, et vice-versa.

Les télécommandes IR utilisent très souvent le standard Philips RC5 (voir **encadré**). Il existe des instructions simples en Bascom de sorte que le décodage est facile. Le **listage 5** montre la programmation d'un émetteur-récepteur RC5 en

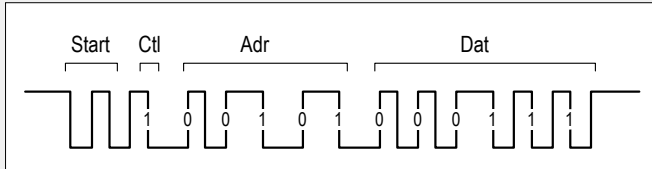
#### Listage 4. Utilisation du DHT11 en Bascom (extrait)

```
'DHT11LCD an AD2, PORTC.2
...
Do
  If Get_dht11() = 1 Then
    Print Humidity
    Locate 1 , 1
    Lcd "H: " ; Humidity ; " % "
    Locate 2 , 1
    Lcd "T: " ; Temperature ; " C "
  End If
  Waitms 1000
Loop
End
```



## Le protocole RC5

La plupart des télécommandes IR de téléviseurs, magnétoscopes et autres appareils électroniques domestiques suivent en partie la norme RC5 définie par Philips. Ce protocole utilise des signaux lumineux modulés dans une plage comprise entre 30 et 40 kHz. La télécommande envoie des salves d'impulsions d'une durée de 0,888 ms



ou 1,778 ms. Pour une fréquence de modulation de 36 kHz, une petite salve contient 32 impulsions tandis qu'une salve longue en contient 64. Le paquet de données qui dure environ 25 ms est répété toutes les 100 ms, tant qu'une touche est pressée.

Le protocole utilise un signal biphase. Un bit a une durée de 1,778 ms. Si les signaux modulés à 36 kHz arrivent dans la première partie, cela signifie un 0 logique, alors que le 1 logique est représenté par la deuxième moitié. Le signal est toujours lancé avec la même séquence de démarrage. Suivent trois plages de données :

- le bit de contrôle (Ctl) passe de 0 à 1 à chaque fois qu'on appuie sur une touche. Le récepteur peut ainsi décider si une touche a été pressée une fois longuement ou plusieurs fois brièvement.
- L'adresse d'appareil (Adr) contient 5 bits ; les bits de poids fort sont transmis en premier. Les adresses d'appareil usuelles sont 0 pour les téléviseurs et 5 pour les magnétoscopes. Ceci permet d'utiliser plusieurs télécommandes dans la même pièce.
- La plage de données (Dat) contient 6 bits pour jusqu'à 64 touches différentes. Les touches numériques de 0 à 9 produisent les codes 0 à 9. Ici également, on envoie d'abord les bits de poids fort.

Bascom. Toutes les données reçues sont affichées sur le LCD. De plus, l'instruction est envoyée à l'interface série. Enfin l'ordre est évalué afin de commuter une sortie du port B. La LED2 est connectée à la broche B.2 du *shield* d'extension. Ainsi, la touche 2 allume la LED et la touche 0 l'éteint.

Le récepteur RC5 peut être relié à n'importe quelle broche d'entrée. Nous utilisons donc une nouvelle fois l'entrée PC3 (AD3). Pour l'instruction [Getrc5](#) en arrière-plan, Bascom utilise l'interruption *timer0* que l'on doit autoriser de manière globale. De plus, la résistance de rappel vers le haut interne est activée pour l'entrée PC3. En effet, cela permet d'utiliser l'émetteur même si le récepteur IR n'est pas connecté. Si l'on avait une entrée à haute impédance sans résistance de rappel vers le haut, l'entrée ouverte pourrait prendre un état bas ce qui pourrait bloquer le programme.

En principe, on utilise PB1 (broche 9 sur l'Arduino) comme sortie pour l'émission (instruction [Sendrc5](#) en Bascom) parce que la sortie OC1A de *timer1* se trouve sur cette broche, généralement utilisée pour produire le signal d'émission à 36 kHz (attention, dans le logiciel en langage C++ Arduino, on utilise une autre sortie pour exécuter cette tâche, de sorte que cette fois-ci, on ne peut pas utiliser la même affectation des broches). En Bascom, il faut prédéfinir, avec une instruction de port, l'état au repos de cette broche de port ; elle doit être à l'état bas durant les pauses. Pour chaque paquet d'impulsions, le logiciel passe via le port à la sortie du *timer*. Comme la diode IR n'a pas de résistance série, il est logique de l'attaquer via un autre port de sortie afin d'utiliser deux résistances internes pour limiter le courant. Ici, on a choisi la broche PB0 qui, elle aussi, peut être commutée à l'état bas en sortie. Il reste encore quatre sorties au port B comme sorties de commutation potentielles pour les données reçues.

L'émission comporte deux ordres. Si l'on appuie sur la touche S1 du *shield* d'extension, le code de touche « 2 » de la télécommande est envoyé. La sortie est ainsi activée sur le récepteur et la LED2 devrait s'allumer. La touche S2 permet d'envoyer un « 0 » et de désactiver la sortie sur le récepteur.

## Arduino et IR

Pour la programmation en langage C Arduino, il faut d'abord inclure la bibliothèque *IRremote* ([listage 6](#)). Cette dernière utilise le *timer2* et sa sortie OC2B (PD3, broche 3 de l'Arduino) pour produire des impulsions qui seront envoyées à la LED IR. Malheureusement, c'est précisément la ligne électrique du LCD monté sur le *shield* d'extension. C'est la raison pour laquelle le programme ne peut pas travailler en même temps avec l'afficheur à cristaux liquides. Rien de grave, puisqu'il nous reste les sorties série pour analyser les données.

L'entrée peut être choisie librement, mais une fois de plus je choisis A2. Pas la peine de programmer la résistance de rappel vers le haut d'entrée puisque la bibliothèque écrasera ce réglage. De plus, c'est inutile puisque la réception ne bloque pas le programme même si l'entrée est ouverte à l'état bas. Il est donc possible d'utiliser ce programme tel quel soit en récepteur soit en émetteur, ou encore en émetteur-récepteur. L'avantage décisif de la bibliothèque *IRremote* est qu'elle permet non seulement de respecter la norme RC5, mais aussi un grand nombre d'autres normes. Comme il n'est pas rare d'avoir plusieurs télécommandes différentes à la maison, pourquoi ne pas viser maintenant le récepteur IR pour savoir quelle norme elles utilisent et quel type de données elles reçoivent. Le RC5 est défini comme type 3. Si l'on appuie plusieurs fois sur la touche 2 d'une télécommande compatible RC5, on obtient les messages suivants :

3  
382  
3  
B82

Les données sont des nombres à 12 bits. Les 5 bits du bas contiennent le code de touche. À ceci s'ajoutent les 5 bits d'adressage du système, dans ce cas l'adresse 14 correspond à un récepteur TNT. Le bit de poids fort est le bit de basculement, il n'a pas besoin d'être évalué et change à chaque fois qu'on appuie sur une touche. Je propose que vous notiez ces

données, car il sera possible de les renvoyer telles qu'elles. En revanche, nous pouvons négliger le bit de basculement. Si l'on envoie 382<sub>hex</sub> au travers de la diode IR, cela correspond à la touche 2 dans le code RC5.

Comme dans ce cas, l'Arduino doit se passer du *shield* d'extension, la sortie de commutation est reliée à la broche 13. Ceci me permet de commander la LED de l'Arduino. La broche 13 du port peut désormais recevoir un relais ce qui permettra de commander un récepteur avec une télécommande RC5 ou une deuxième carte Arduino dotée d'une diode IR.

En sens inverse, on utilise une fois de plus les commandes des touches 2 et 0. Comme dans la version en Bascom, les touches correspondantes sont reliées à A0 et A1, les résistances de rappel vers le haut internes y sont activées. Il est possible d'y raccorder beaucoup d'autres capteurs contenus dans le kit. Il n'y a pas que les boutons-poussoirs. Pourquoi ne pas y rac-

corder un capteur magnétique, un capteur de position ou un capteur de lumière ? On imagine les applications : le téléviseur qui s'éteint automatiquement au lever du soleil, le capteur de position qui détecte l'entrée d'une personne dans une pièce et allume aussitôt la lumière et la radio pour lui souhaiter la bienvenue. ◀

(160210 – version française : Pascal Duchesnes)

## Liens

- [1] [www.elektor.fr/arduino-sensor-kit](http://www.elektor.fr/arduino-sensor-kit)
- [2] [www.elektormagazine.fr/160152](http://www.elektormagazine.fr/160152)
- [3] [www.elektormagazine.fr/160173](http://www.elektormagazine.fr/160173)
- [4] [www.elektormagazine.fr/160210](http://www.elektormagazine.fr/160210)
- [5] [www.elektormagazine.fr/140009](http://www.elektormagazine.fr/140009)

### Listage 5. Émetteur et récepteur RC5 en Bascom

```
'RC5LCD In AD2, PORTC.2, Out OC1A, PORTB1
...
Do
  If S2 = 0 Then
    Togbit = 0 : Address = 0 : Command = 2
    Do
      Rc5send Togbit , Address , Command
      Waitms 100
    Loop Until S2 = 1
  End If
  If S1 = 0 Then
    Togbit = 0 : Address = 0 : Command = 0
    Do
      Rc5send Togbit , Address , Command
      Waitms 100
    Loop Until S1 = 1
```

```
End If
Getrc5(address , Command)
If Address < 255 Then
  Locate 1 , 1
  Lcd Address ; " "
  Locate 2 , 1
  Togbit = Command / 128
  Lcd Togbit ; " "
  Locate 2 , 5
  Command = Command And &B01111111
  Lcd Command ; " "
  Print Command
  If Command = 2 Then Portb.2 = 1
  If Command = 0 Then Portb.2 = 0
End If
Loop
```

### Listage 6. Commande IR en langage C Arduino

```
#include <IRremote.h>
int RECV_PIN = A2;
IRrecv irrecv(RECV_PIN);
IRsend irsend;
decode_results results;
int d;
int S1 = A0;
int S2 = A1;
int LED = 13;
int kathode = 2;
void setup()
{
  Serial.begin(9600);
  irrecv.enableIRIn();
  pinMode(S1, INPUT_PULLUP);
  pinMode(S2, INPUT_PULLUP);
  pinMode(RECV_PIN, INPUT_PULLUP);
  pinMode(LED, OUTPUT);
  pinMode(kathode, OUTPUT);
```

```
}

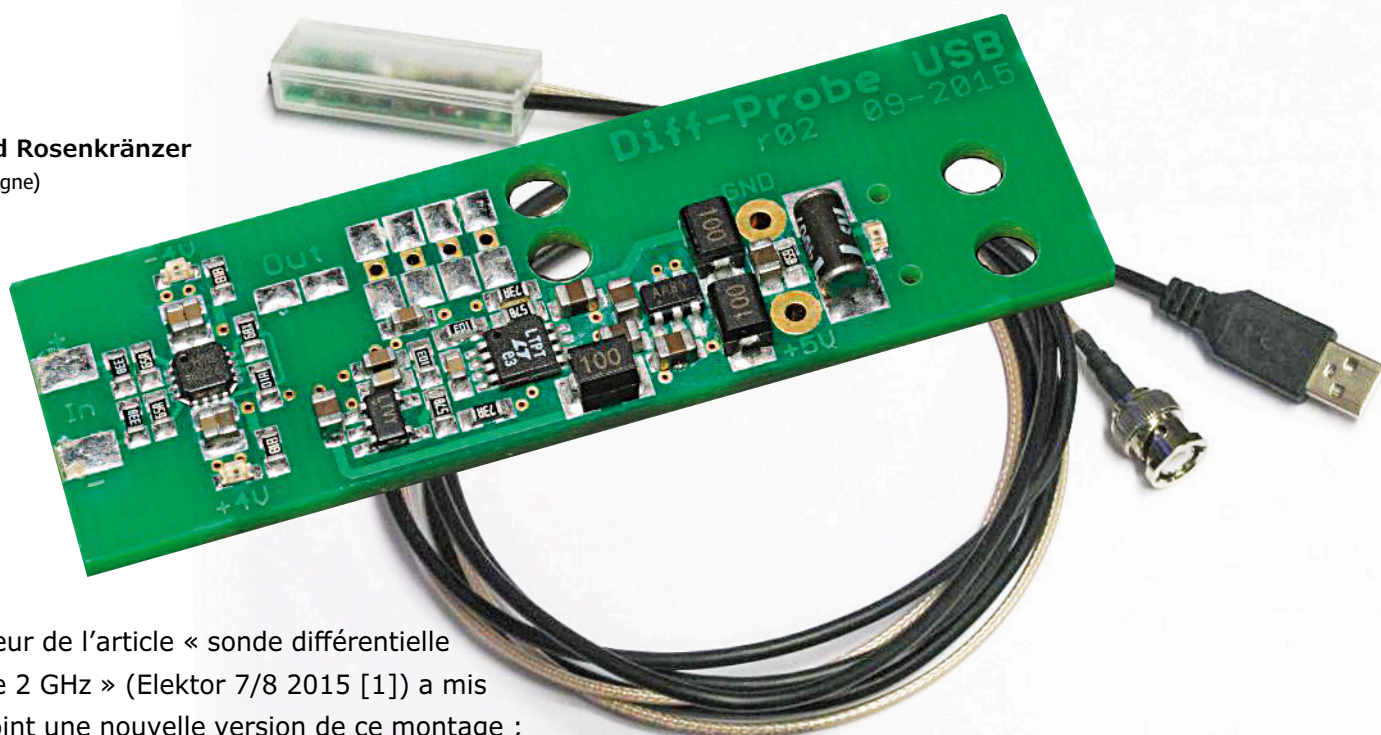
void loop() {
  if (irrecv.decode(&results)) {
    Serial.println(results.decode_type);
    Serial.println(results.value, HEX);
    d = results.value & 15;
    Serial.println(d);
    if (d==2) digitalWrite(LED,1);
    if (d==0) digitalWrite(LED,0);
  }
  if (digitalRead(S1) == 0)
    irsend.sendRC5(0x382, 32);
  if (digitalRead(S2) == 0)
    irsend.sendRC5(0x380, 32);
  irrecv.enableIRIn();
  delay(100);
}
```



# sonde différentielle active V2

## alimentée maintenant par USB

Alfred Rosenkränzer  
(Allemagne)



L'auteur de l'article « sonde différentielle active 2 GHz » (Elektor 7/8 2015 [1]) a mis au point une nouvelle version de ce montage ; elle est maintenant alimentée par le 5 V d'un port USB.

La **figure 1** montre le synoptique de la nouvelle alimentation. Après filtrage de la tension de 5 V fournie par la prise USB, une « pompe à charge » (*charge pump*) délivre une tension négative non régulée de valeur (presque) égale à celle de la tension d'entrée. Après un nouveau filtrage, un régulateur

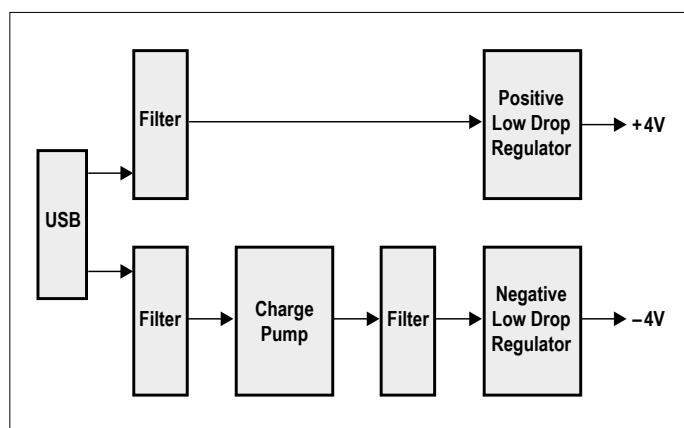


Figure 1. Synoptique de la nouvelle alimentation par USB.

négatif à faible tension de déchet (*low drop*) la régule à  $-4$  V. Un régulateur positif à faible tension de déchet fournit directement  $+4$  V à partir de la tension USB filtrée.

On retrouve, en **figure 2**, le schéma complet. Le circuit IC1, un LT1962EMS8, délivre la tension d'alimentation positive régulée. Le circuit IC2, un LT1964ES5, fait de même pour la tension négative. La pompe à charge prend la forme d'un MAX1697 (IC3). Les diodes (D2, D3 et D4) signalent la présence de la tension USB ainsi que celle des deux tensions régulées. Le filtrage de la tension USB « brute » est l'affaire des bobines L1 et L2. Le circuit de l'amplificateur différentiel (IC4) n'a guère changé. Seules les deux résistances d'entrée en série des deux entrées ont été combinées en un seul composant, ce qui permet d'avancer l'ampli op et de donner plus de place à l'alimentation. Le circuit, comparé à celui publié précédemment (voir [1]), comporte certaines différences. En raison de l'absence de convertisseur de tension pour assurer l'isolement galvanique, la masse du port USB est connectée à la masse du signal.

Malgré un filtrage sophistiqué, on peut, avec un analyseur de spectre, identifier, à un niveau  $< -80$  dBm, la fréquence d'horloge de quelque 200 kHz de la pompe à charge. Sur un oscilloscope, elle reste cependant invisible.

Dernier point, le coût des régulateurs a entraîné une augmentation de prix.

Le circuit trouve toujours place dans le boîtier USB utilisé auparavant, le câble BNC lui aussi reste inchangé. ◀

(150801 - version française : Guy Raedersdorf)

L'auteur propose aux lecteurs intéressés la sonde différentielle prête à l'emploi. Écrivez à [alfred\\_rosenkraenzer@gmx.de](mailto:alfred_rosenkraenzer@gmx.de)

## Lien

[1] [www.elektormagazine.fr/130538](http://www.elektormagazine.fr/130538)

## Spécifications

Atténuation : 10:1 pour des signaux différentiels et bouclage de 50  $\Omega$

Impédance d'entrée différentielle : 5 100  $\Omega$ ,  $\pm 1\%$

Impédance d'entrée asymétrique : 2 550  $\Omega$ ,  $\pm 1\%$

Plage d'entrée en mode commun (CMR) :  $\pm 12$  V

Impédance de sortie : 50  $\Omega$ ,  $\pm 1\%$

Largeur de bande : env. 1,9 GHz ( $-3$  dB)

Temps de montée/descente : 300 ps

Alimentation : USB 5 V, env. 70 mA

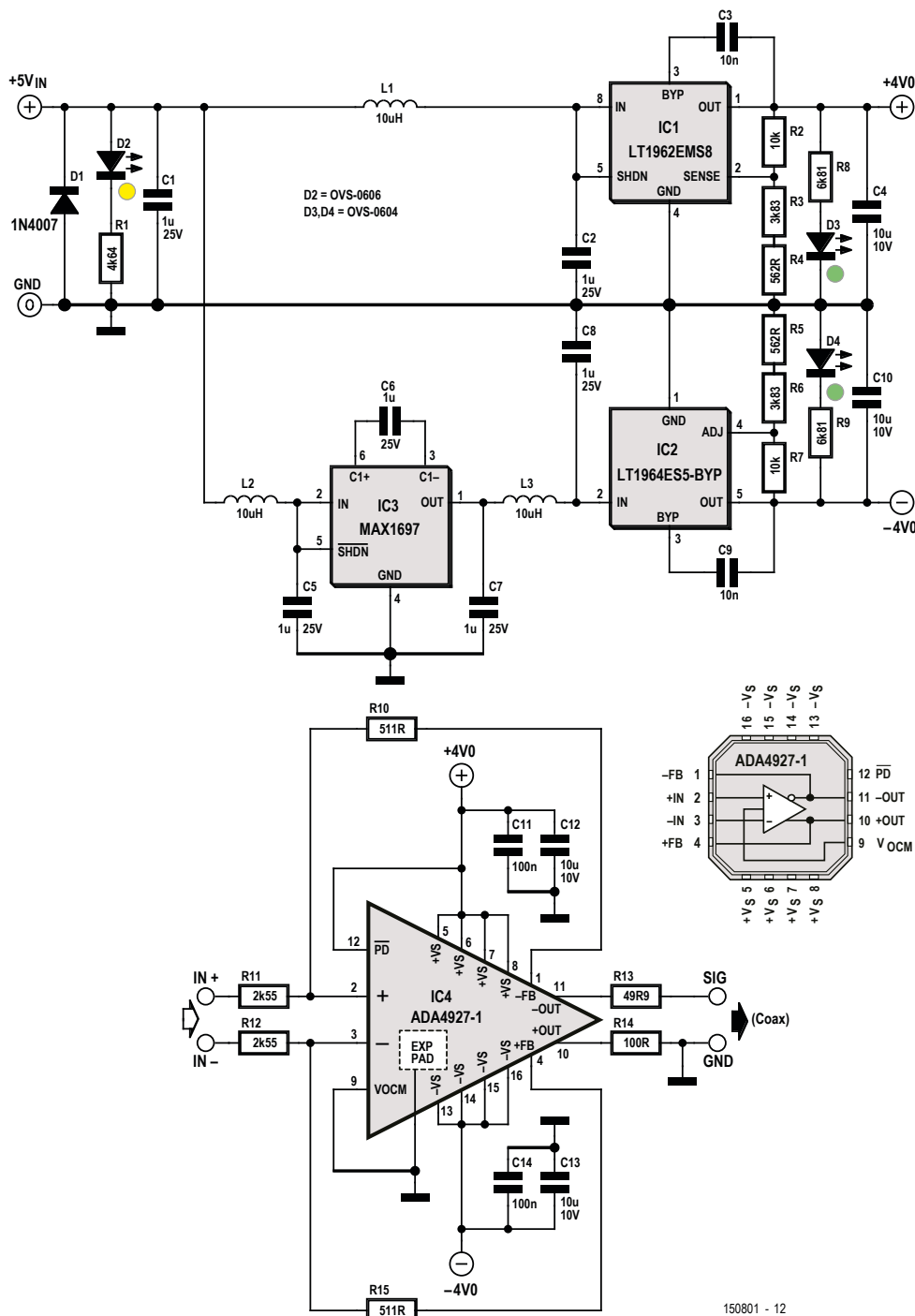


Figure2. Le schéma complet. La partie autour de l'amplificateur différentiel (IC4) n'a guère changé.



# kit d'initiation RFID pour Arduino Uno

## thermomètre et contrôle d'accès

Avec ce kit d'initiation RFID, Elektor vous propose le matériel de base pour des expériences intéressantes avec la carte Arduino Uno. Nos deux exemples vous montrent combien il est facile d'aborder de nouvelles applications de la carte Arduino Uno et consorts.

La carte Arduino et ses répliques, dits clones, sont depuis de nombreuses années des incontournables dans la communauté des électroniciens et des inventeurs de tout poil. Dans le passé,

on parlait d'un microcontrôleur ; aujourd'hui, on fait de plus en plus appel à Arduino. Accéder au monde de la commande est ainsi devenu un jeu d'enfant. Plutôt que de se casser la tête avec un EDI pour remplir des pages de programmes complexes, on travaille de manière intuitive sur un outil de développement que même un débutant maîtrise vite.

Mais après, quand il faut réaliser une extension matérielle, c'est une autre histoire, le concepteur se retrouve isolé. Il existe bien une quantité de *shields* pour des tâches spéciales, mais leur domaine d'application est assez restreint. Si l'on veut vraiment innover, il faut se coltiner tous les composants électroniques, ce qui peut donner des soucis au débutant.

Les kits sont une solution, car ils mettent à disposition tous les composants nécessaires à la réalisation de différents projets personnels. Dans les années 70 et 80, ces ensembles étaient présentés dans de séduisantes boîtes de construction d'électronique. C'est alors que de nombreux amateurs se sont mis à réaliser des sirènes et des radios en ondes moyennes ; maintenant, ce sont les projets à microcontrôleur qui ont la cote.

Le « kit d'initiation RFID pour Arduino Uno » est une variante particulièrement intéressante ! On y trouve plus de trente composants, des éléments de montage et des modules pour tous les domaines de l'électronique moderne. La dénomination du kit est un peu trompeuse, car la boîte renferme non seulement une carte de réception RFID avec deux puces de transpondeur et un porte-clef, mais également de nombreux autres composants.

Outre la carte Arduino Uno elle-même, le coffret en plastique, très pratique, contient entre autres :

- un capteur hygrométrique
- une LED multicolore
- une grande matrice de LED à 64 points lumineux
- un afficheur à quatre chiffres à sept segments
- une télécommande infrarouge avec le récepteur assorti
- un module LCD complet avec connectivité I<sup>2</sup>C.

Vous trouverez des informations détaillées sur le contenu du kit via [1]. Les deux exemples qui suivent ne représentent évidemment qu'une infime partie des multiples possibilités

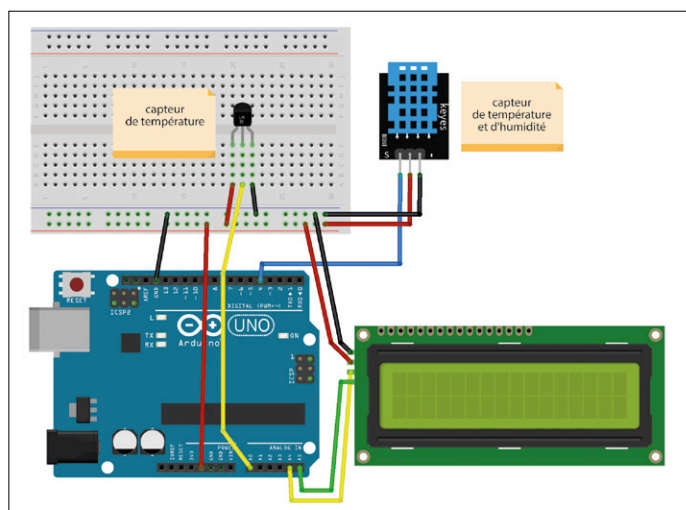


Figure 1. Station météorologique avec deux capteurs et LCD.

du kit qui permet de réaliser un nombre illimité d'expériences et de montages.

### Station météo à affichage LCD

Nous commençons par mettre en œuvre plusieurs capteurs pour recueillir différentes mesures météorologiques. Ce seront d'abord le capteur de température et d'humidité DHT11 ainsi que le capteur de température LM35DZ de manière à réaliser un thermomètre/hygromètre pour l'atmosphère intérieure et un thermomètre pour l'extérieur.

C'est le LCD qui est mis à contribution pour afficher les mesures. La **figure 1** montre le câblage complet du montage, tandis que le **listage 1** reproduit le croquis Arduino.

Le **capteur LM35** a été étalonné en usine pour une pente précise de 10 mV par degré Celsius. En outre, le zéro de la tension de sortie correspond exactement à 0,0 °C. Compte tenu de la résolution du convertisseur A/N à 10 bits utilisé (valeurs de 0 à 1 023) et de la tension d'alimentation du LM35 à 5 V, on peut écrire :

```
temp = (5.0 * analogRead (tempPin)
* 100.0) / 1023;
```

Ce qui entraîne une faible résolution, puisque le LM35 ne délivre alors qu'une tension entre 0 V et un peu plus de 1 V. Tel quel, avec la référence interne de 5 V du CA/N d'Arduino, on perd 80 % de la plage de valeurs possible. En revanche, fixer la tension de référence à 1,1 V permet presque d'atteindre la résolution maximale.

Changer la tension de référence à 1,1 V oblige évidemment à modifier la formule de conversion. Divisez 1,1 par 1 023, vous obtiendrez des pas de 1,0752 mV pour la sortie numérique. Pour un accroissement de 10 mV/°C, le facteur de conversion devient :

```
float TempCal = 0.1075; // 1.075/10
```

La ligne de programme suivante calcule la température :

```
tempLM35 = (LM35val * TempCal);
```

Activer la tension de référence interne de 1,1 V se fait sur Arduino par l'instruction :

```
analogReference (INTERNAL);
```

Avec cette formule, on atteint une résolution d'un dixième de degré Celsius. La tension de référence peut bien sûr présenter une certaine tolérance,

on l'ajuste en modifiant un peu la valeur **0.1075**. La bonne valeur trouvée, on peut s'attendre à un affichage précis à long terme, du fait de la stabilité de la tension de référence.

La plage de température avec ce calcul va de 0 à 110 °C. Comme un dixième de degré correspond à seulement 1 mV, il est fort probable que des tensions thermoélectriques sur les contacts et les soudures peuvent avoir une certaine influence. Pour un thermomètre d'intérieur, l'affichage au degré près peut suffire, mais pour des mesures plus précises, mieux vaut tenir compte de la remarque ci-dessus.

On lit les mesures du capteur combiné **de température et d'humidité DHT11** par le bus I<sup>2</sup>C. Il existe une bibliothèque qui facilite l'acquisition des valeurs mesurées, il y en a aussi pour l'afficheur et le bus I<sup>2</sup>C. Pour voir du texte ou des mesures, utilisez la fonction :

```
lcd.print()
```

#### Listage 1. Grandeurs environnementales des capteurs DHT11 et LM35

```
// DHT11_LM35_w_LCD_display.ino

#include <dht.h>
#include <LiquidCrystal_I2C.h>
#include <Wire.h>

LiquidCrystal_I2C lcd(0x27,16,2); // LCD address 0x27
dht DHT;
const int DHT11_PIN= 4;
const int A0 = 0;
float tempLM35 = 0;
long LM35val = 0;
float TempCal = 0.1075;

void setup()
{ lcd.begin();
  lcd.backlight();
  analogReference(INTERNAL);
}

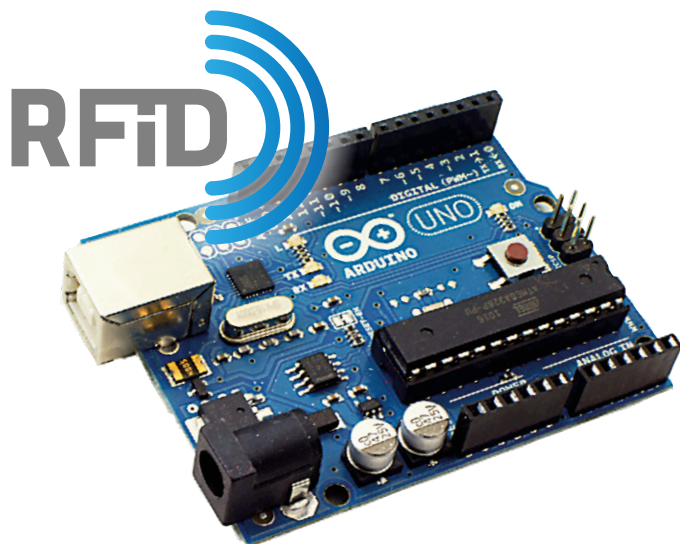
void loop()
{ DHT.read11(DHT11_PIN);

  LM35val = analogRead(A0);
  tempLM35 = (LM35val * TempCal);

  lcd.setCursor(0, 0); lcd.print("Ti="); lcd.print(DHT.temperature,0);
  lcd.print(char(223)); lcd.print("C"); // print unit " °C "

  lcd.setCursor(8, 0); lcd.print("Ta="); lcd.print(tempLM35,0);
  lcd.print(char(223)); lcd.print("C"); // print unit " °C "

  lcd.setCursor(0, 1);
  lcd.print("Humidity: ");
  lcd.print(DHT.humidity,0); lcd.print(" %");
  delay(200);
}
```



Une commande de curseur intégrée facilite la mise en forme de l'état de sortie. Les bibliothèques contiennent des croquis étonnamment compacts tels que `LiquidCrystal_I2C.h` [2], `DHT-Lib` [3] et `Wire.h`. La bibliothèque `Wire` a servi à la communication par le bus I<sup>2</sup>C et appartient aux bibliothèques standard, tout comme les bibliothèques SPI et Servo dans le croquis suivant, qui sont déjà dans l'EDI Arduino. Les deux autres

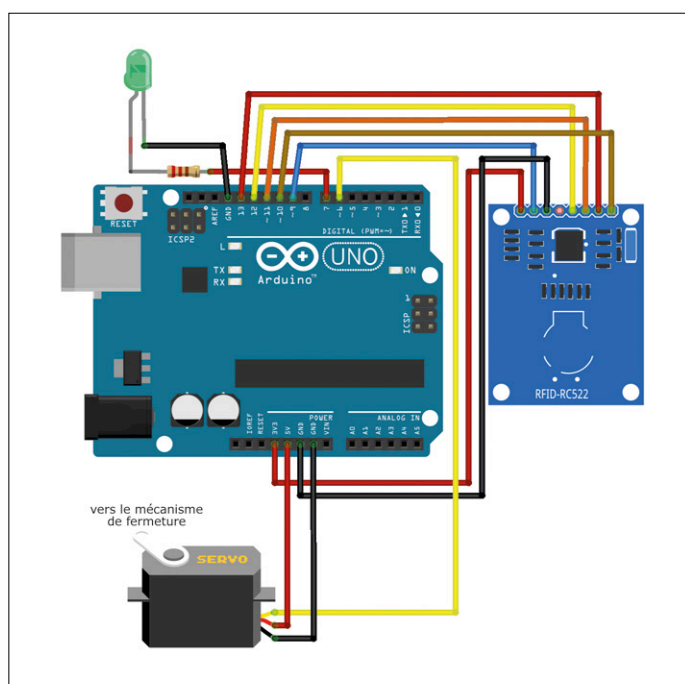


Figure 2. Commande de servo par module RFID.

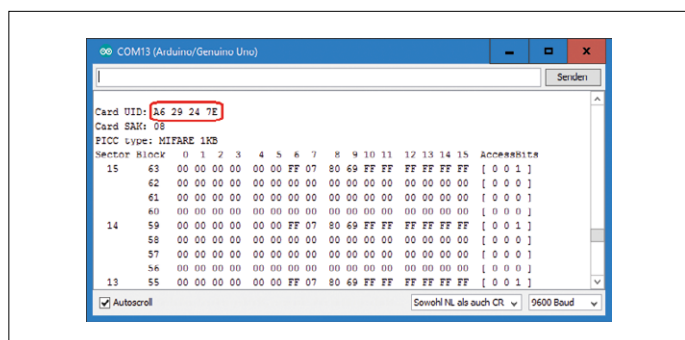


Figure 3. Les données du transpondeur RFID sur le moniteur série.

bibliothèques sont à télécharger gratuitement sur l'internet. Si un lien ne donne pas le résultat escompté, vous trouverez vite un remplaçant avec un moteur de recherche.

Le kit RFID vous permettra d'étoffer votre station météo d'autres capteurs. Le détecteur de niveau d'eau et la photo-résistance (LDR) peuvent fournir d'autres données. Le détecteur d'eau permet de savoir immédiatement quand il pleut. La LDR peut servir à mesurer la luminosité actuelle, à partir de laquelle vous pourrez établir l'intensité de l'ensoleillement et sa durée et même en tracer le graphe.

### Serrure de porte avec sécurité RFID

Des composants du kit permettent aussi de commander une serrure électronique à ouvrir au moyen de la puce RFID. Le module récepteur de RFID, qui a donné son nom au kit d'initiation, n'est en fait qu'un composant intéressant parmi bien d'autres.

La communication entre le module RFID et Arduino passe par SPI, *Serial Peripheral Interface*. L'interface possède trois lignes auxquelles tous les participants au bus sont reliés en commun. La ligne SCK (*Serial Clock*) fournit le signal d'horloge fixe du maître pour la synchronisation. Sur les lignes MOSI (*Master Output, Slave Input*) et MISO (*Master Input, Slave Output*), les données sont envoyées et reçues.

Outre la ligne d'horloge et celles de données, chaque esclave est relié à une ligne particulière de sélection, *Chip Select* notée CS ou parfois SS (*Slave Select*). Toutes ces lignes sont commandées par le maître. Une ligne RST permet la RàZ du module. Il peut encore y avoir une ligne d'interruption IRQ, nous ne l'utilisons pas ici.

Les liaisons entre le module RFID et Arduino sont répertoriées dans le **tableau 1**.

Question logiciel, deux autres bibliothèques sont encore nécessaires, SPI-Lib de l'IDE Arduino et MFRC522-Lib de [4] pour le module RFID.

Quand le matériel est câblé comme à la **figure 2**, on peut charger dans l'Arduino le croquis `RFID-RC522_data` (**listage 2**) du paquet de téléchargement [5]. Il permettra de lire les données d'un transpondeur RFID quand le moniteur série de l'EDI Arduino sera ouvert. Dès qu'un transpondeur sera tenu à proximité du module RFID, les données enregistrées sur la puce apparaîtront (**fig. 3**).

Le paramètre le plus important, celui que le croquis utilise, est l'UID (*Unique IDentification number*) du transpondeur. Les autres données stockées sur la puce ne nous intéressent pas pour ce projet.

Un des services les plus appréciés d'un module RFID, c'est le contrôle d'accès à un système. Quiconque est autorisé à accéder à une pièce réservée possède alors une étiquette sous forme d'une carte ou d'un porte-clef. Devant la porte, il l'expose au module RFID et la serrure s'ouvre pour autant que son étiquette soit valide. Par rapport aux clefs classiques, ce système a de nombreux avantages. Une étiquette perdue, il suffit d'en radier le code de la liste des identités valides, plutôt que de devoir



**Tableau 1. Liaisons entre module RFID et Arduino**

broche RC522	broche Arduino Uno
VCC (3V3)	3,3 V
GND	GND
RST	9
SDA (SS)	10
MOSI	11
MISO	12
SCK	13

remplacer la serrure et toutes les clefs distribuées. Pareil pour toute personne qui perd son accréditation, qui serait encore en possession de son étiquette et refuserait de la rendre pour quelque raison que ce soit. Et puis surtout, les étiquettes sont nettement meilleur marché que les clefs de sécurité.

Pour reconnaître une identité donnée, il est plus commode de convertir la séquence hexadécimale de l'UID en un nombre décimal, la vérification de la validité d'une étiquette en est simplifiée. Bien sûr, cela rend le système plus vulnérable, parce que tous les nombres des quatre blocs hexadécimaux ne peuvent être convertis en des nombres décimaux du même format, mais si vous tenez à la sécurité maximale, vous pouvez modifier en ce sens le logiciel, de manière à interroger chaque bloc en particulier, mais le travail sera nettement plus lourd.

Le croquis `RFID-RC522_servo_lock.ino` se trouve également dans le paquet de téléchargement relatif à cet article. Alors l'UID ainsi que le code correspondant s'affichent sur le moniteur série en décimal. Le nombre décimal s'inscrit comme code valide avec la ligne :

```
long validCode = 938350;
// enter valid code
```

On peut utiliser le système comme serrure électronique de sécurité puisque la séquence de déverrouillage n'est activée que si l'étiquette placée à proximité du module récepteur contient un code correct.

Le servomoteur de modélisme compris dans le kit peut servir à déplacer le pêne de la porte. La figure 2 montre également comment relier le servomoteur à la carte Arduino. En supplément, on peut connecter une LED verte pour annoncer que la porte ou le mécanisme de fermeture d'un casier a été déverrouillé. Pour cela, le code de la bibliothèque du servo a été allongé d'une simple instruction à la fin du croquis.

### Montage et perspectives

La plupart des nouveaux kits d'électronique ne contiennent plus de composants, mais des modules prêts à l'emploi. On semble ainsi s'éloigner de la technique, alors qu'il est possible de mettre au point des projets très utiles et pratiques en quelques étapes simples. Puis, grâce

à des bibliothèques bourrées de ressources, il est possible de programmer vite et bien les croquis nécessaires.

Ce kit s'est inspiré des boîtes d'expérimentation classiques des débuts de l'électronique pour permettre à de jeunes bidouilleurs et ingénieurs en herbe, mais aussi aux vieilles tiges de la guilde de l'électronique, de s'adonner à toutes les options qu'offre la technique moderne.

Dans le cadre de ce petit article, on ne peut présenter qu'une infime fraction des possibilités du kit. Laissez parler votre imagination ! ◀

(160322 – version française : Robert Grignard)

### Liens

- [1] Kit d'initiation RFID pour Arduino Uno dans l'e-choppe (réf. 150396-71) : [www.elektor.fr/rfid-starter-kit-for-arduino-uno](http://www.elektor.fr/rfid-starter-kit-for-arduino-uno)
- [2] Bibliothèque *LiquidCrystal-I2C* : <https://github.com/fdebrabander/Arduino-LiquidCrystal-I2C-library>
- [3] Bibliothèque pour le capteur DHT11 : <http://arduino.cc/playground/Main/DHTLib>
- [4] Bibliothèque pour le module RFID : <https://github.com/miguelbalboa/rfid>
- [5] Page du projet avec paquet de téléchargement : [www.elektormagazine.fr/160322](http://www.elektormagazine.fr/160322)

### Listage 2. Affichage de l'UID RFID sur le moniteur série

```
// RFID-RC522_data.ino

#include <SPI.h>
#include <MFRC522.h>

#define RST_PIN      9
#define SS_PIN       10

MFRC522 mfrc522(SS_PIN, RST_PIN);

void setup()
{
  Serial.begin(9600);
  SPI.begin();           // Start SPI bus
  mfrc522.PCD_Init();     // Initialise MFC522 Reader
  mfrc522.PCD_DumpVersionToSerial(); // Show RC522 details
  Serial.println("Place RFID TAG in range!");
}

void loop()
{
  // Card present?
  if ( ! mfrc522.PICC_IsNewCardPresent() )
  { return; }

  // Select card
  if ( ! mfrc522.PICC_ReadCardSerial() )
  { return; }

  // Send data to serial interface
  mfrc522.PICC_DumpToSerial(&(mfrc522.uid));
}
```

# conception d'enceintes

## programmes gratuits pour le calcul d'enceintes

Harry Baggen (labo d'Elektor)

Nous avons abordé de nombreux sujets dans cette rubrique introduite en mars 1996 sous le titre « électronique en ligne ». Certains sont désormais désuets, mais d'autres ont bien supporté l'usure du temps. C'est le cas de la conception et la construction d'enceintes ; nous avons passé en revue quelques sites et programmes en décembre 1997, et j'ai voulu voir ce qu'il en est près de 20 ans plus tard.

À cette époque, j'étais un fervent constructeur d'enceintes, que j'avais conçues à l'aide des programmes disponibles sur ces sites. Il m'a semblé amusant – et instructif – de voir quels programmes étaient disponibles gratuitement de nos jours (il existe pas mal d'offres commerciales, mais en tant qu'audiophiles et électroniciens, c'est surtout les enceintes elles-mêmes qui nous intéressent). Ce qui m'a frappé de prime abord, c'est que de nombreux logiciels n'ont plus été mis à jour depuis des lustres, et qu'on n'en trouve guère de récents. Les anciens programmes peuvent certes tourner dans une DOSBox ou un autre émulateur, quoique ce ne soit pas toujours évident, mais on préférerait un environnement Windows 7, 8 ou 10 natif.

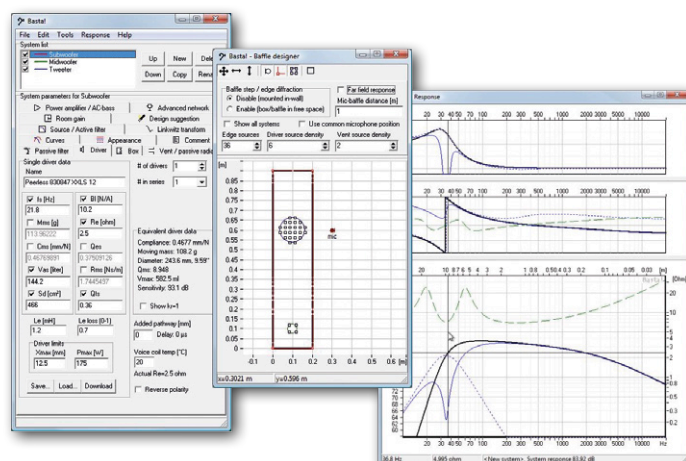
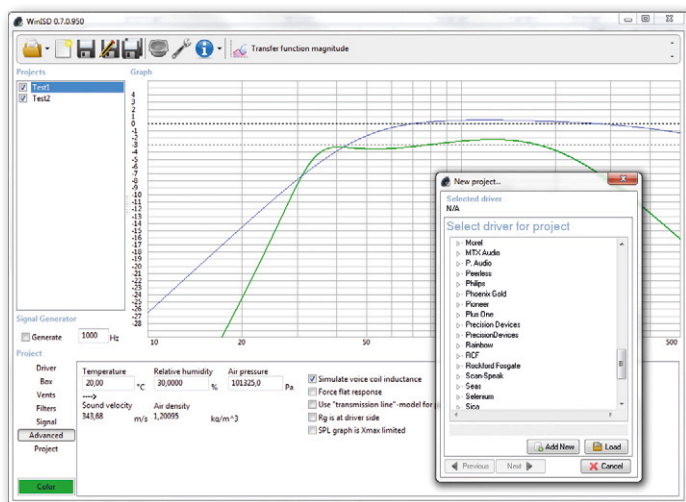
Un des programmes qui existait en 1997, et qui est toujours mis à jour, est [WinISD](#) [1], ISD pour *Interactive Speaker Design*. Il est relativement lent, mais il permet le calcul d'enceintes closes, *bass-reflex* et à charge passe-bande. La possibilité de calculer divers types de filtres passifs et égaliseurs est aussi offerte, mais pour un nouveau type vous devrez introduire les valeurs vous-même. C'est un bon programme, avec de nombreuses possibilités, et il tourne sous Windows 7.

Un autre programme intéressant est [Basta!](#), de *Tolvan Data* [2]. Il a 8 ans, et était initialement un logiciel commercial ; il

n'est plus mis à jour depuis quelques années et est proposé aujourd'hui gratuitement (*freeware*) ; suivant les concepteurs il tourne sous Windows 8 ou 10. Il permet le calcul et la simulation de baffles et d'enceintes closes, *bass-reflex* ou à charge passe-bande (à événements ou à radiateur passif). Le programme offre d'autres possibilités : calcul de la diffraction, des résonances, de la vélocité et de l'excursion des cônes, etc.

Sur le même site, vous trouverez aussi le programme [Sirp](#) [3], qui permet diverses mesures sur les enceintes à l'aide d'un signal sinusoïdal wobulé (*chirp*) à rampe logarithmique : réponses en fréquence et impulsionnelle, distorsion, etc. Ce n'est certes pas aussi précis qu'une analyse basée sur la technique MLS (*Maximum Length Sequence*), qui élimine les influences de l'environnement, mais les résultats sont exploitables. La courbe d'impédance peut également être mesurée avec *Sirp*, et les résultats importés dans *Basta!* comme métadonnées.

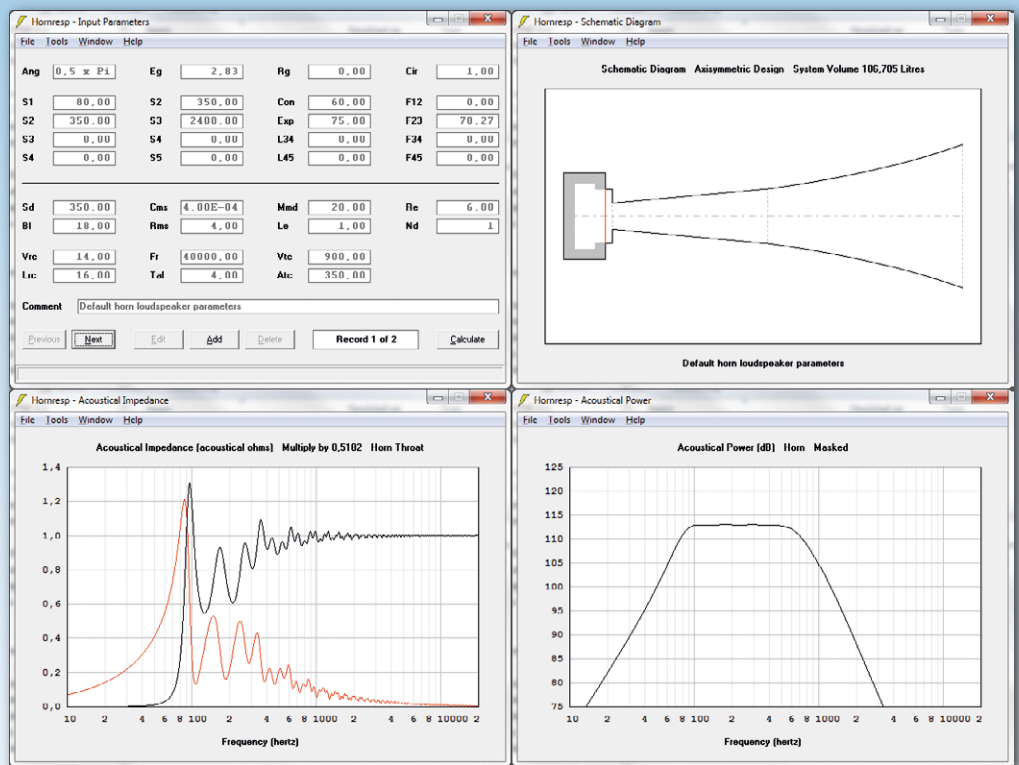
Un programme plus récent est [VituitxCAD](#) [4], de Kimmo Sau-nisto ; il offre de nombreuses possibilités et tourne sous Windows 7, 8 ou 10. Il permet la conception et la simulation de systèmes à plusieurs voies, avec divers types de filtres et d'enceintes, dont celles à radiateur passif ou charge passe-



bande (trois modèles). Le programme peut aussi calculer la diffraction d'angle et la directivité du système. Il vaut certainement la peine qu'on s'y attarde, mais il demande pas mal de temps et d'efforts pour explorer ses possibilités (ce qui est aussi vrai pour les autres logiciels).

Je terminerai avec *Hornresp* [5], un des rares programmes gratuits pour la conception de haut-parleurs à pavillon ; le site est très dépouillé : il n'y a que quelques lignes de texte, le lien pour télécharger le programme, et l'adresse de courriel du développeur. Vous trouverez néanmoins des explications concernant le programme et les HP à pavillon sur divers forums de discussion, dont ceux de *Home Theater Shack* [6] et d'*Audiofanzine* [7], ce dernier est en français. Si vous voulez vous lancer dans l'aventure, il est indispensable d'en apprendre plus sur le sujet, non seulement pour la conception ou la construction, mais aussi pour le calcul à l'aide du programme. *Hornresp* permet bien entendu le calcul d'autres types d'enceinte. Il existe depuis pas mal de temps, mais son concepteur, David J. McBean, le maintient à jour ; il en est à sa version 40.20, et tourne parfaitement sous Windows 7. ◀

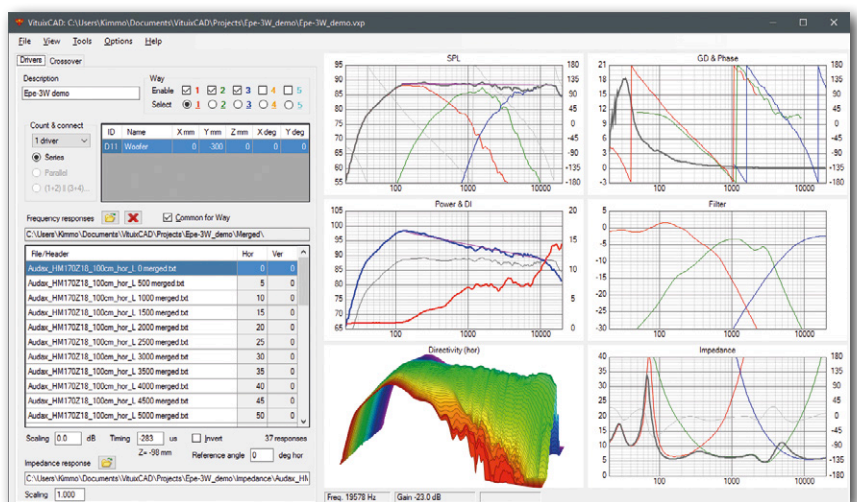
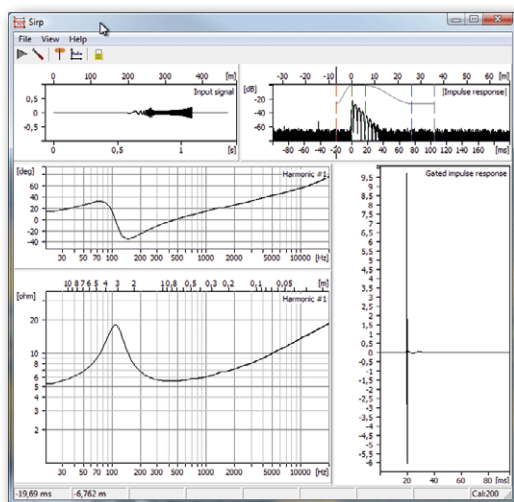
(160301 – version française : Jean-Louis Mehren)



▶ *Hornresp* est un des rares programmes gratuits pour la conception de haut-parleurs à pavillon

## Liens

- [1] [www.linearteam.dk/](http://www.linearteam.dk/)
- [2] [www.tolvan.com/index.php?page=/basta/basta.php](http://www.tolvan.com/index.php?page=/basta/basta.php)
- [3] [www.tolvan.com/index.php?page=/sirp/sirp.php](http://www.tolvan.com/index.php?page=/sirp/sirp.php)
- [4] [kimmosaunisto.net/Software/Software.html#VituixCAD](http://kimmosaunisto.net/Software/Software.html#VituixCAD)
- [5] [www.hornresp.net/](http://www.hornresp.net/)
- [6] [www.hometheatershack.com/forums/diy-subwoofers-general-discussion/36532-hornresp-dum-hmm-everyone.html](http://www.hometheatershack.com/forums/diy-subwoofers-general-discussion/36532-hornresp-dum-hmm-everyone.html)
- [7] [fr.audiofanzine.com/construction-d-enceintes/forums/t.525885,hornresp-introduction-et-tutoriels.html](http://fr.audiofanzine.com/construction-d-enceintes/forums/t.525885,hornresp-introduction-et-tutoriels.html)

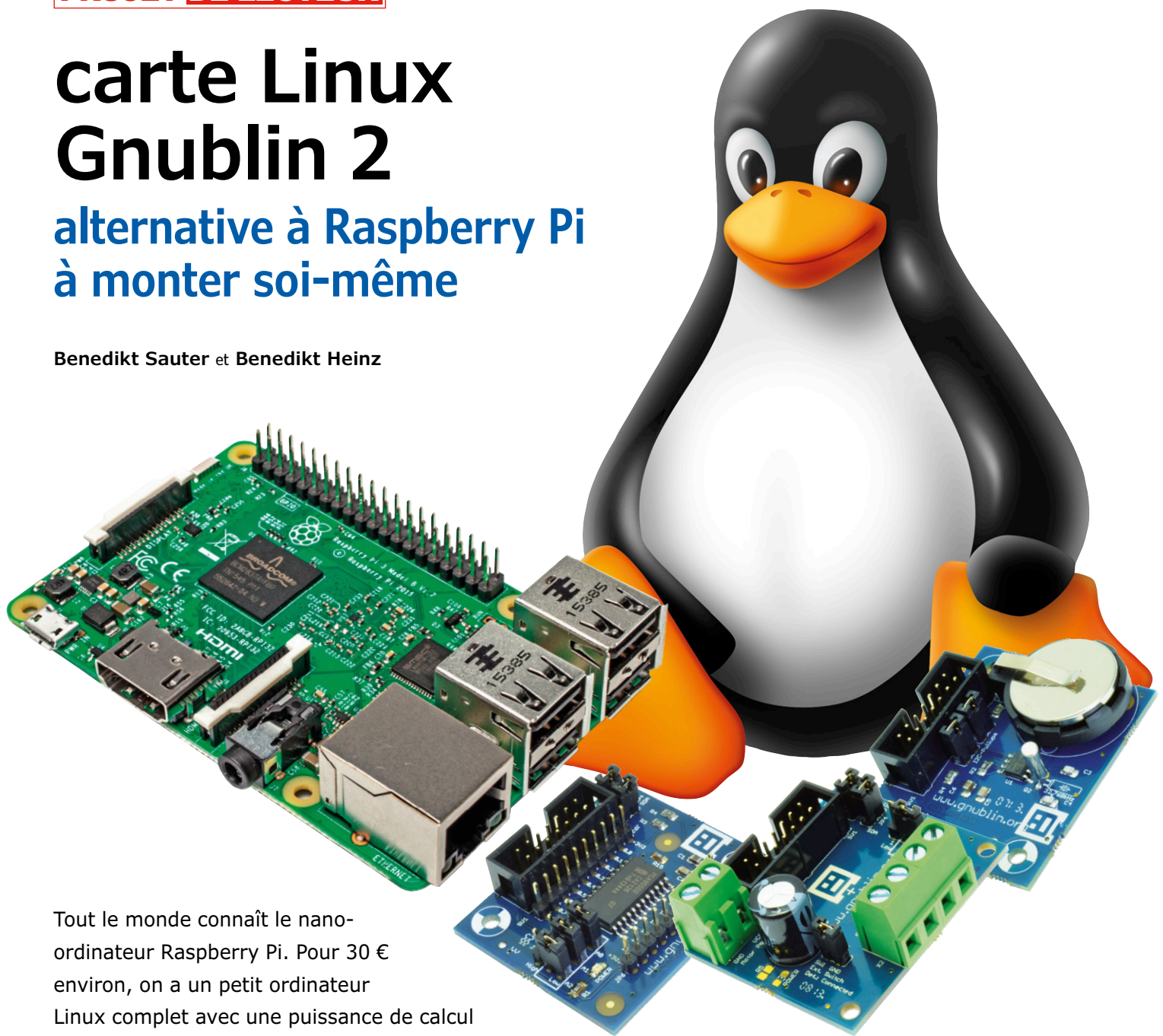




# carte Linux Gnublin 2

alternative à Raspberry Pi  
à monter soi-même

Benedikt Sauter et Benedikt Heinz



Tout le monde connaît le nano-ordinateur Raspberry Pi. Pour 30 € environ, on a un petit ordinateur Linux complet avec une puissance de calcul appréciable. Un petit problème toutefois : si son prix est très intéressant, sa conception matérielle n'est pas très ouverte. C'est pourquoi les auteurs ont conçu leur propre ordinateur monocarte avec une puce Cortex-A5. Son prix de revient n'est peut-être pas aussi attractif, mais son matériel et son logiciel sont adaptables et extensibles à volonté. On peut ainsi réaliser soi-même une carte Linux pour l'enseignement, le développement ou un usage industriel.

Bien entendu, la carte Gnublin 2 a un passé ; elle porte l'indice 2 car il y a eu une première version. La carte Gnublin 1 a servi de base à la carte Linux Elektor et au cours sur Linux embarqué publié par Elektor [1], qui a été suivi de quelques autres articles et de toute une série de modules dont certains sont

toujours disponibles dans l'e-choppe [2]. Notre société *embedded projects* leur a consacré un site web [3].

Depuis, cinq années ont passé, une petite éternité en microélectronique. Pendant cette période sont apparus non seulement le RPi avec tous ses descendants

et concurrents, mais aussi une série toujours renouvelée de cartes Linux sans cesse améliorées, qui s'adressent plus ou moins aux mêmes utilisateurs. La question est donc : est-ce que cela a vraiment un sens de sortir une nouvelle version de Gnublin ? La réponse « oui » devra évidemment l'emporter sur le simple

« nous aussi », d'autant que Gublin 1 jouit toujours d'une coquette popularité. À notre avis, c'est parce que les autres cartes satisfont insuffisamment les critères suivants :

- La carte Gublin est totalement ouverte. Le schéma, le dessin du circuit imprimé, le logiciel, tout est disponible.
- Les composants utilisés sont faciles à se procurer même en petite quantité et le circuit imprimé à « seulement » six couches est relativement bon marché même en petite série.
- Au lieu de vouloir briller avec des performances incroyables, comme la plupart des cartes Linux, Gublin 1 se distingue par une consommation d'énergie particulièrement réduite. Beaucoup de systèmes embarqués n'ont que faire d'une fonction graphique 3D complexe ; pour un fonctionnement permanent, l'intérêt se porte plutôt vers une faible consommation d'énergie.

Il y a là suffisamment de raisons pour nous pousser à explorer le marché à la recherche d'un système sur puce (*System on a Chip, SoC*) ARM pour donner vie à la version 2 de la carte Gublin, en nous appuyant sur deux critères essentiels : faible consommation et support optimal de Linux.

### Choix du système sur puce

Comme la faible consommation est primordiale, notre attention s'est concentrée sur les systèmes monocœurs de types Cortex-A5 et Cortex-A7. Nous avons finalement arrêté notre choix sur l'ATSAMA5D41 [4] d'Atmel avec un noyau Cortex-A5. Quant au support de Linux, nous en avons déjà acquis une large expérience avec Gublin 1 et d'autres projets, non sans quelques déboires frustrants, en particulier avec la LPC3131 de NXP sur Gublin 1. La dernière version du noyau pour cette puce est la 2.6.33, âgée de plus de 6 ans ! Quoique la LPC3131 soit toujours donnée comme active, c'est-à-dire « en production », cette puce n'est plus supportée par le noyau de Linux depuis des années.

Il en va tout autrement pour Atmel. Presque tous les systèmes sur puce Atmel – même ceux qui ont déjà quelques années – sont toujours couramment supportés. Il n'est nul besoin

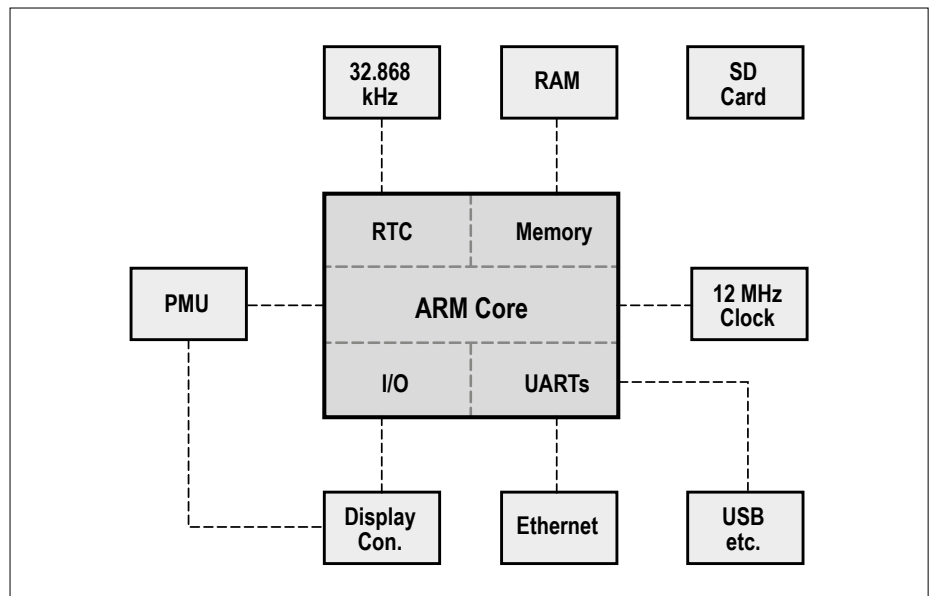


Fig 1. Le système sur puce encadré en trait gras est représenté sur ce diagramme avec ses principaux sous-ensembles et les composants périphériques connectés suivant leurs fonctions.

d'appliquer des correctifs spécifiques du fabricant, car le noyau officiel de Linux lui-même en fournit le support. De plus, Atmel met à disposition une documentation complète, sans aucune clause de confidentialité, et collabore étroitement au développement avec la communauté Linux, de sorte que l'équipe de développement du noyau maintient à niveau le code relatif à ces puces lors des modifications apportées au noyau. Voilà la plus sûre garantie d'un support à long terme de ces SoC par le noyau de Linux.

En outre, la puce ATSAMA5D41 offre quelques possibilités qui font défaut aux autres cartes Linux. On dispose par ex. de deux ports série, l'un pour la console Linux, l'autre pour des fonctions de l'utilisateur. De plus, Gublin 2 peut être relié directement à un PC par USB à travers le port à haute vitesse (*high speed device port*). C'est impossible avec Raspberry Pi. Avec les systèmes embarqués, on a souvent besoin d'une horloge en temps réel qui conserve l'heure exacte, même en cas de coupure de courant. Sur Gublin 2, l'horloge en temps réel peut être facilement secourue par une pile additionnelle. Même un afficheur tactile résistif peut être connecté directement sans alimentation extérieure. Voilà suffisamment de raisons pour une version 2 de Gublin basée sur cette puce. Comme la version précédente, la carte Gublin 2 est destinée à servir de base pour des projets ou produits personnels. Pour cela, les données habituelles sont complétées par

des rapports de mesure du laboratoire de CEM (Compatibilité Électromagnétique).

### Pour quoi faire ?

La carte Gublin 2 est à sa place partout où il est question de commande ou de mesure, c'est-à-dire dans toute application ou tout produit où il est avantageux de remplacer le microcontrôleur classique par un système Linux de puissance suffisante. Le prix de la configuration de base (processeur, mémoire et alimentation inclus) varie entre 15 et 20 € en fonction du nombre d'exemplaires. Gublin 2 est donc compétitif, même pour des petites séries. Grâce à la disponibilité de la liste des composants, du schéma et du dessin du circuit imprimé ainsi que des mesures de CEM et des outils de calcul, on fait l'économie, surtout pour les petites séries, de la quantité d'efforts qu'il aurait fallu investir dans l'intégration de cartes « toutes faites » comme RPi et consorts. De plus, avec la documentation, il n'est pas sorcier de connecter des cartes d'extension. Dans le processus de développement classique qui passe par des prototypes montés à la main, on n'arrive à la première série qu'après passage au laboratoire de CEM. Que ces mesures soient déjà disponibles est donc un raccourci bienvenu. Et même pour des pièces à l'unité, pour lesquelles des tests de CEM seraient hors de proportion, il est rassurant de savoir à quoi s'en tenir quant au comportement en haute fréquence.



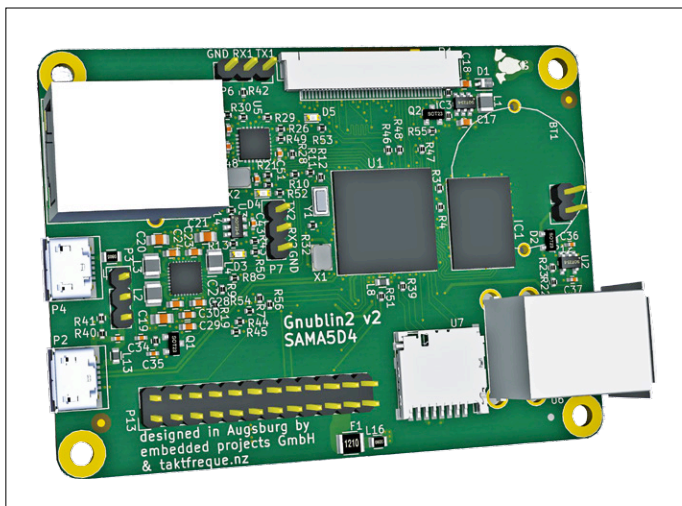


Fig 2. Représentation en 3D du circuit de Gnublin 2.

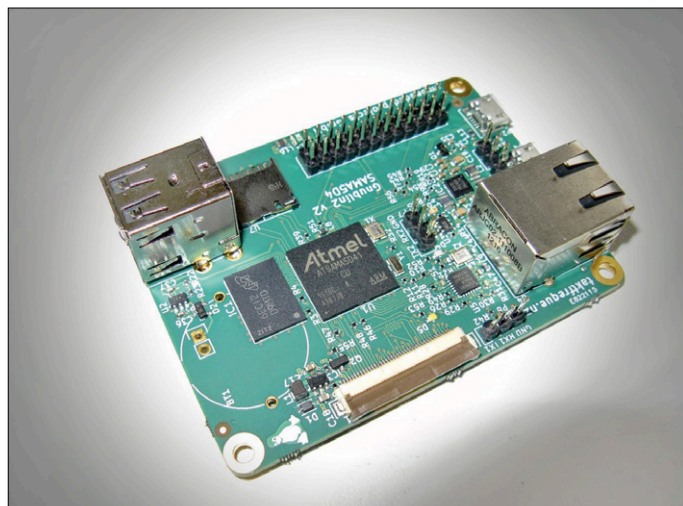


Fig 3. Le prototype terminé de Gnublin 2.

## Structure et composants

Par bonheur, la carte Gnublin 2 ne consomme que 0,3 W au repos, c'est-à-dire quand Linux est inoccupé, et cela à une fréquence d'horloge de 600 MHz et avec 256 Mo de RAM. Sur le diagramme fonctionnel de la **figure. 1**, le système sur puce est représenté au centre, encadré en trait gras. On distingue autour du cœur les principaux sous-ensembles (horloge en temps réel, mémoire, etc.) et leurs différentes liaisons avec des composants périphériques. Le schéma détaillé

peut être téléchargé sur [6], il est trop grand pour être reproduit ici. La mémoire DDR2 de 256 Mo (IC1) est reliée par un bus à 16 bits à la puce ATmega328P (U1). Par rapport à d'autres cartes, qui utilisent un bus à 32 bits, cela permet de réaliser un circuit imprimé plus simple, moins coûteux et moins énergivore. Grâce à la mémoire DDR2 cadencée à 300 MHz, on atteint un débit de plus de 1 Go/s. La puce requiert des tensions différentes pour les entrées/sorties et le cœur du

processeur. Elles sont fournies par l'unité d'alimentation (*Power Management Unit, PMU*) de type ACT8865 (IC2) [5]. Cette unité comporte plusieurs régulateurs linéaires et à découpage et peut être configurée par I<sup>2</sup>C depuis l'unité centrale. L'alimentation de l'horloge en temps réel intégrée à la puce est assurée par le régulateur linéaire à faible tension de déchet (*low dropout, LDO*) et très faible consommation TPS78001 (U2), lui-même alimenté par une pile bouton ou une source externe connectée à une paire

## Naviguer sur l'internet avec un afficheur et un navigateur

Bien que plusieurs distributions Linux puissent tourner sur la carte Cortex-A5, nous préférons Debian Jessie [1] pour le développement, mais Yocto ou Ubuntu conviennent aussi bien. Toutefois, dans sa version de base, Debian est nettement moins volumineuse qu'Ubuntu. De plus, Debian bénéficie d'une énorme quantité de paquets logiciels qu'il est facile d'installer à travers le réseau par l'outil *apt*. Les paquets déjà installés peuvent être facilement mis à jour à travers le réseau, ce qui est bien pratique et utile pour les mises à jour de sécurité. Même le compilateur GNU-C peut être installé de cette manière.

Le logiciel peut également être compilé directement sur la carte s'il n'est pas trop volumineux. Avec LXDE [2], on dispose d'une interface graphique simple, peu consommatrice de ressources et qui se laisse installer avec la commande « *apt-get install lxde* ». Pour l'affichage des sites web les plus élaborés, les 256 Mo de RAM peuvent ne pas suffire, mais avec Midori [3], on dispose d'un navigateur de petite empreinte, capable d'afficher beaucoup de sites utilisant HTML5, CSS et Javascript. On l'installe avec la commande « *apt-get install midori* ».



La **figure** montre que Gnublin 2 peut charger et afficher sans problème des pages web « normales ».

[1] Jessie : [www.debian.org/releases/jessie](http://www.debian.org/releases/jessie)

[2] LXDE : [http://wiki.lxde.org/de/LXDE\\_Debian](http://wiki.lxde.org/de/LXDE_Debian)

[3] Midori : <http://midori-browser.org>



de broches. L'horloge en temps réel est bien entendu complétée par un quartz de la fréquence typique de 32 768 Hz (Y1). Le cadencement normal est dérivé d'un quartz à 12 Mhz (X1).

À la différence de Gnublin 1, la deuxième version peut gérer directement un afficheur à cristaux liquides, ce qui ouvre la voie à beaucoup de nouvelles applications. Pour que la carte puisse se contenter d'une tension d'alimentation unique de 5 V, même en présence d'un afficheur, nous avons prévu un convertisseur élévateur de tension AP5724 (IC3). Celui-ci alimente le rétroéclairage de l'afficheur avec un courant constant de 40 mA sous la tension de 24 V. Le convertisseur A/N intégré permet la connexion directe d'un afficheur tactile résistif.

La puce KSZ8081RNA/D (U5) gère la connexion de la carte à un réseau Ethernet 10/100 Mbit/s. Le transformateur de couplage requis est directement intégré à la prise U4. Une EEPROM 24AA025E48 (U3), connectée au port I<sup>2</sup>C, est chargée de fournir une adresse MAC valide.

Comme la plupart des cartes Linux, Gnublin 2 possède une mémoire de masse : le chargeur d'amorce (*bootloader*) et le système Linux sont lus au démarrage sur une carte micro-SD (U7).

La puce comprend trois ports USB à haute vitesse. Deux d'entre eux, en mode hôte, utilisent la double prise USB U6. Le troisième, en mode périphérique, passe par la prise micro-USB P2. De cette manière, la carte peut être vue par un PC comme pratiquement n'importe quel périphérique USB. On peut par ex. établir une liaison USB entre un ordinateur hôte et Gnublin 2. L'alimentation de la carte passe alors directement par la prise micro-USB P2. Il y a aussi une deuxième prise micro-USB P4 qui sert à brancher une alimentation plus puissante (comme celle d'un ordiphone), au cas où plusieurs périphériques USB seraient connectés à Gnublin 2. Le cavalier P3 sert à sélectionner la prise d'alimentation active, P2 ou P4. Les broches d'entrées/sorties de la puce travaillent avec une tension de 3,3 V. Beaucoup d'entre elles sont accessibles sur l'embase mâle P13. Comme sur Gnublin 1, on dispose ici d'un bus SPI avec quelques sorties de sélection de puce (*chip select*). P13 fournit aussi l'accès à un bus I<sup>2</sup>C et, avec P7, au second port série (UART) de la puce. Ce port n'est pas utilisé par Linux comme console ; il est donc disponible pour des fonctions

## Financement participatif

Pour ce projet, embedded projects et Elektor ont lancé une campagne de financement participatif. Si vous désirez y participer ou prendre une option sur une carte Gnublin 2, rendez-vous sur le lien : [www.gnublin.org](http://www.gnublin.org)



propres à l'utilisateur. Enfin, P6 donne accès au premier port série (UART), qui met à disposition une console et affiche les messages du noyau au démarrage. On peut aussi contrôler le chargeur d'amorce par ce port.

La **figure 2** présente une vue 3D de la carte créée avec KiCAD. Les fichiers de conception, le schéma correspondant, le logiciel ainsi que la documentation et des compléments d'information sont disponibles en téléchargement gratuit sur [6]. La **figure 3** montre une photo du circuit imprimé terminé, garni de ses composants. L'**encadré** fournit les informations de base sur les logiciels à installer pour faire de Gnublin 2 avec afficheur un ordinateur Linux complet.

### En résumé

Cet article présente une courte description de ce dont il est question avec Gnu-

blin 2. Avec les composants de [6], il y a déjà de quoi réaliser de superbes projets personnels. Par ailleurs, au cours de notre étude se sont révélés des aspects secondaires tellement intéressants qu'ils méritent leurs propres articles. Vous pouvez donc compter sur la parution prochaine d'une description détaillée de la conception d'un circuit imprimé avec le logiciel KiCAD. Projet qui ne tournera pas autour d'une minable carte avec trois composants, mais autour de la carte Gnublin 2 elle-même. Et les mesures de rayonnement parasite alimenteront un article sur la compatibilité électromagnétique, pour lequel vous vous doutez déjà de quel exemple on se servira.

Enfin, il y aura un article qui montrera, pas à pas, comment réaliser sa propre version de cette carte Linux. ◀

(160090 – version française : Helmut Müller)

### Les auteurs

Benedikt Sauter est informaticien diplômé. Après ses études, il s'occupa de divers projets à microcontrôleurs et décida rapidement de devenir autoentrepreneur. Peu après, il fonda la société *embedded projects*. Passionné d'informatique depuis l'adolescence, il a publié plusieurs projets ouverts dans Elektor. Benedikt Heinz, informaticien diplômé, conçoit depuis plus de quinze ans de l'électronique numérique et du logiciel pour microcontrôleurs et ordinateurs. Utilisateur de Linux sur PC depuis de longues années, il privilégie l'usage des logiciels ouverts et contribue même à leur développement. Il travaille actuellement sur un logiciel qui permettra de simplifier la conception avec KiCAD de cartes complexes (PCI, DDR-RAM, etc.).

### Liens

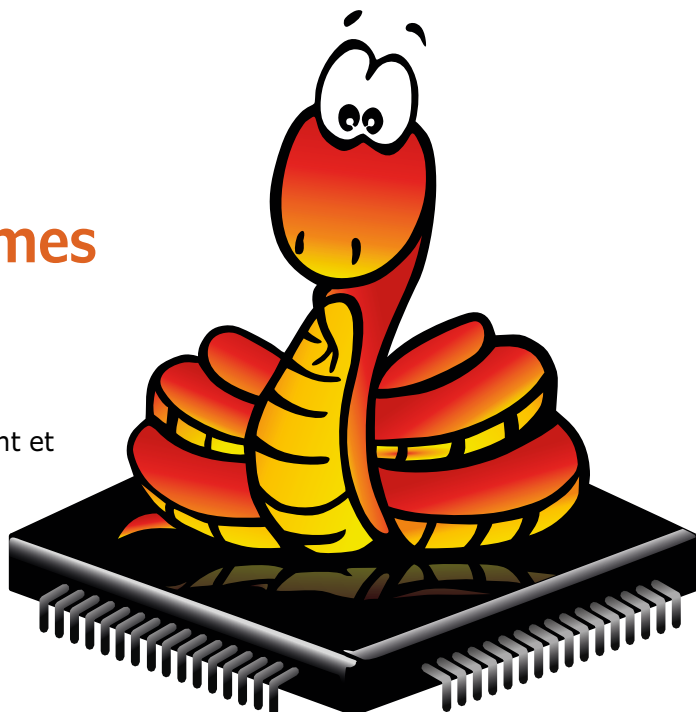
- [1] *Embarquez Linux! (1) - Linux pour les électroniciens*, Elektor 05/2012 : [www.elektormagazine.fr/120026](http://www.elektormagazine.fr/120026)
- [2] Produits Gnublin : [www.elektor.fr/devtools/gnublin-linux](http://www.elektor.fr/devtools/gnublin-linux)
- [3] Site web Gnublin : <http://gnublin.embedded-projects.net>
- [4] ATSAM5D41 : [www.atmel.com/devices/ATSAMA5D41.aspx](http://www.atmel.com/devices/ATSAMA5D41.aspx)
- [5] ACT8865 : [www.active-semi.com/products/power-management-units/act88xx/](http://www.active-semi.com/products/power-management-units/act88xx/)
- [6] Composants Gnublin 2 : [www.gnublin.org/index2.html](http://www.gnublin.org/index2.html)

# MicroPython

## le Python des petits systèmes

Dogan Ibrahim (Royaume-Uni)

MicroPython est un langage de programmation, puissant et efficace, dérivé du Python et qui a hérité d'une petite collection de bibliothèques. MicroPython n'occupe que 256 Ko de mémoire de programme et 16 Ko de RAM, c'est idéal pour des microcontrôleurs et autres systèmes embarqués limités en ressources.



Python est utilisé dans de nombreuses universités et écoles techniques à travers le monde comme langage d'initiation à la programmation. Ses nombreuses et puissantes bibliothèques et sa facilité d'utilisation en font un langage idéal pour les débutants. MicroPython, à l'origine conçu

pour les  $\mu$ C ARM à 32 bits, est compatible avec Python et, grâce à sa petite taille, est un excellent choix pour les processeurs embarqués. MicroPython permet d'écrire du code complexe et gérable pour piloter des systèmes embarqués qui nécessitent autrement des langages tels que le C ou C++. MicroPython permet aux utilisateurs aguerris tout comme aux débutants de programmer de petits systèmes embarqués.

Les principales raisons pour lesquelles MicroPython ne supporte pas entièrement Python 3 sont le manque de mémoire et l'absence de fonctions matérielles et logicielles des processeurs embarqués (comme le multitâche et le multiprocesseur), voir en [1] les différences entre Python 3 et MicroPython.

### Pourquoi MicroPython ?

Comparés aux autres langages de programmation, MicroPython et Python offrent des fonctions uniques. En résumé :

- (Micro)Python est interactif : le programme n'est pas compilé et chargé dans le processeur cible, mais traité et interprété lors de l'exécution. Malgré un léger ralentissement du programme, l'utilisateur a l'avantage de pouvoir tester facilement son code : par ex. faire de simples calculs interactifs comme avec une calculatrice, ou bien expérimenter avec des parties du programme jusqu'à obtenir le résultat souhaité. Cette fonction appelée *Read Evaluate Print Loop* (REPL, boucle lire-évaluer-afficher) n'est pas disponible dans les langages compilés

comme le C ou le C++.

- En plus de maintes fonctions natives, quantité de bibliothèques de fonctions sont disponibles pour être incluses (ou importées) dans un programme : génération de nombres aléatoires, trigonométrie, audio, réseau, traitement de chaînes et de fichiers, graphisme et jeux...
- MicroPython peut être mélangé avec d'autres langages comme le C ou le C++. Cela donne de la puissance et de la souplesse supplémentaires, car les parties de codes qui doivent être rapides sont développées avec des langages mieux adaptés à ces tâches.
- Les exceptions et le traitement d'erreurs sont supportés, c'est particulièrement important pour la programmation en temps réel. Sans une gestion appropriée des erreurs, un dysfonctionnement du programme arrêterait le processeur dans un état inconnu, ce qui peut avoir des effets indésirables.
- C'est à code source ouvert : la dernière version est disponible en [2] et exécutable gratuitement. Les codes source MicroPython peuvent être modifiés et adaptés à un processeur spécifique.
- Enfin, le langage est humainement lisible et sa syntaxe est facile à apprendre et à comprendre.

### Ce que MicroPython peut et ne peut pas faire

MicroPython peut, en général, faire tout ce que les autres langages de programmation peuvent faire, comme commander du matériel (LED, écrans, commutateurs et boutons, capteurs, moteurs...). Les

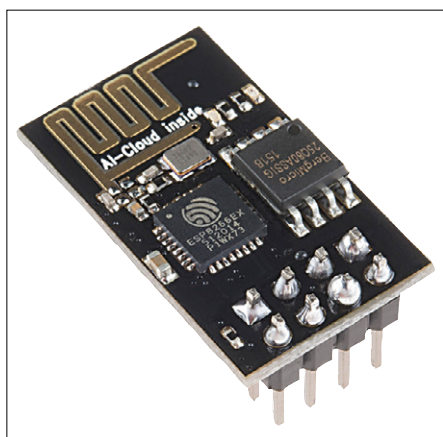


Figure 1. L'ESP01, un module très pratique et populaire, basé sur l'ESP8266.

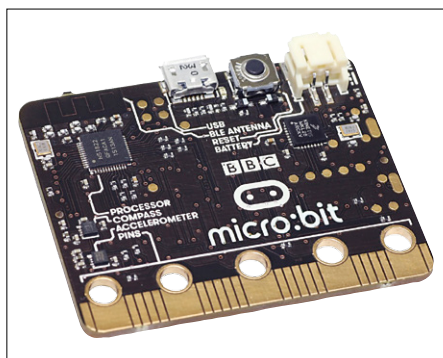


Figure 2. La carte BBC micro:bit peut être programmée en ligne avec MicroPython.

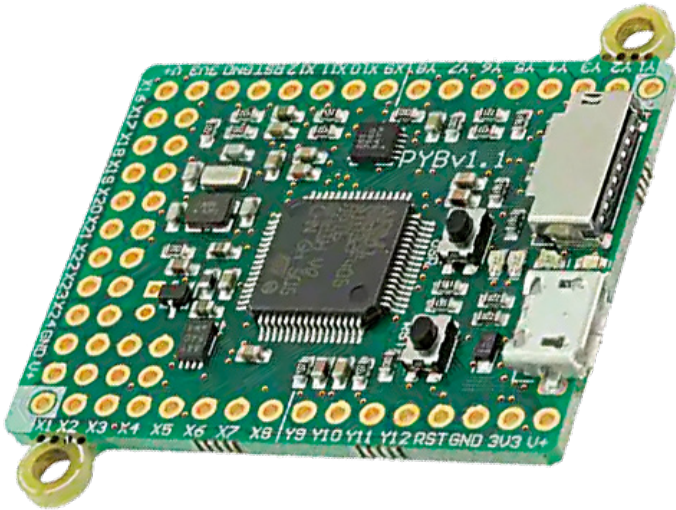


Figure 3. Pour information, la pyboard est la carte à µC officielle de MicroPython.



Figure 4. La WiPy est similaire à la carte NodeMCU basée sur l'ESP8266, mais avec un ARM Cortex-M4 CC3200 à la place.

bus de communication comme RS-232, CA/N, I<sup>2</sup>C, SPI, et autres sont faciles à utiliser grâce aux bibliothèques natives et externes. Il est possible d'écrire des programmes qui utilisent le réseau et le Wi-Fi pour communiquer avec d'autres appareils sur un réseau, ou pour développer des systèmes de type IdO. Comme MicroPython est un langage interprété, il est plus lent que d'autres langages de programmation embarqués, par conséquent ce n'est pas un bon choix pour du traitement rapide de signaux numériques ou des applications en temps réel pour lesquelles une grande vitesse d'exécution est critique. En plus, bien que de nos jours ce soit moins important, MicroPython utilise légèrement plus de mémoire que la plupart des autres langages embarqués. Comme MicroPython est une branche de Python qui ne supporte pas toutes les bibliothèques Python, un programme écrit en Python risque de ne pas fonctionner sur un système embarqué sous MicroPython.

### Cartes supportées par MicroPython

Le nombre de cartes de développement supporté par MicroPython augmente avec sa popularité. En voici quelques-unes.

#### ESP8266

Les cartes conçues autour de ce µC populaire, compatible Wi-Fi (fig. 1), avec pile TCP/IP intégrée et interface USB, peuvent être programmées en MicroPython. Le µC est basé sur un processeur RISC à 32 bits et MicroPython offre le support pour les E/S d'usage général (GPIO), SPI, I<sup>2</sup>C,

UART, CA/N et I<sup>2</sup>S. C'est une des solutions les plus économiques pour se lancer dans l'expérimentation en MicroPython.

#### BBC micro:bit

Cette carte au format carte de crédit (fig. 2) supportée par MicroPython est bien équipée : 25 LED, deux boutons-poussoirs, un accéléromètre, un magnétomètre, GPIO, I<sup>2</sup>C, UART et CA/N. Il est possible de créer les programmes en ligne, ce qui évite de devoir régler ou configurer quoi que ce soit. La BBC micro:bit est vivement recommandée pour les novices en programmation.

#### pyboard

MicroPython est préchargé dans cette carte de développement (fig. 3) basée sur un processeur Cortex-M4 de la série STM32F. La pyboard peut être connectée à un PC via son port USB. La carte embarque horloge en temps réel, accéléromètre, GPIO, CA/N, 4 LED et logement pour carte micro-SD.

#### WiPy

Comme la pyboard, cette carte (fig. 4) est aussi livrée avec MicroPython intégré. Ses broches sont adaptées au branchement sur une plaque d'essai. La carte est basée sur le processeur Cortex-M4 CC3200 qui tourne à 80 MHz et inclut UART, SPI, I<sup>2</sup>S, plusieurs CA/N, Wi-Fi, GPIO, compteurs ainsi que hachage et moteurs de cryptage.

MicroPython est supporté par d'autres cartes de développement embarqué intéressantes : Teensy 3.x, SAMD21, LoPy,

STM32F4-Discovery, Raspberry Pi et BeagleBone (les deux avec code Python 3 complet).

### Exemple de programme MicroPython

Voici un programme MicroPython simple qui s'exécute sur la BBC micro:bit. C'est un thermostat où la température du processeur est lue continuellement et un message approprié est affiché sur la matrice de LED de la carte. Quand la température est égale ou supérieure à 25 °C, le message « HIGH » est affiché. Si la température est entre 20 °C et 25 °C, c'est « MEDIUM ». Sinon, « LOW » est affiché.

```
#Simple CPU thermostat program
from microbit import *
```

```
while True:
    temp = temperature()
    If temp >= 25:
        display.scroll("HIGH")
    elif temp >= 20 and temp < 25:
        display.scroll("MEDIUM")
    else:
        display.scroll("LOW")
```



(160315 – version française : Alexandre Roy)

### Liens

- [1] <https://github.com/micropython/micropython/wiki/Differences>
- [2] <https://github.com/micropython/>
- [3] <http://micropython.org/>



# régénérateur de batteries au plomb 0-30 V

avec contrôle de la qualité de l'accumulateur

Jan Lichtenbelt (Pays-Bas)



Une entreprise de vente de matériel et de composants bien connue propose un *activateur de batterie au plomb* bon marché depuis quelques années [1]. Le principe est simple : toutes les 30 s, l'accumulateur doit débiter 100 A pendant 100  $\mu$ s. C'est difficile à prouver, mais il semble bien que ce traitement prolonge la durée de vie des batteries. Il y a moyen d'étendre les fonctions d'un tel appareil, c'est ce que nous vous proposons dans cet article.

## À PROPOS DU PROJET



mesure & test

loisirs

modélisme

batteries



débutant

→ connaisseur

expert



environ 3 h



pas d'outils spéciaux requis



environ 60 €

La théorie est que des impulsions de forte intensité empêchent la sulfatation des électrodes [2]. Nous utilisons l'activateur (auss appelé régénérateur) de Conrad en parallèle avec une alimentation de 13,8 V à LM317. L'accumulateur au plomb a une capacité de 7 Ah. L'alimentation ne semble pas gênée par les impulsions de courant.

Les impulsions de courant empêchent non seulement la sulfatation des électrodes, mais elles permettent aussi le contrôle de l'état de la batterie : nous faisons circuler un courant intense pendant un court laps de temps dans un shunt externe, et nous mesurons la chute de tension à ses bornes. L'intensité du courant dans le circuit est égale à :

$$I = \Delta V_{\text{shunt}} / R_{\text{shunt}}$$

La valeur de la résistance interne de la batterie est égale à :

$$R_i = (V_0 - V_t) / I,$$

où  $V_0$  est la tension aux bornes de la batterie à vide, et  $V_t$  avec charge. La valeur de la résistance interne est représentative de la qualité d'un accu : plus elle est basse, meilleur est l'accu. Ce principe est appliqué dans notre appareil, qui fonctionne en régénérateur pour les batteries au plomb de 2, 6, 12 ou 24 V, et peut mesurer la valeur de la résistance interne de piles, batteries et accus jusque 30 V. Nous voulions utiliser la carte de Conrad en la modifiant, mais nous nous sommes aperçus d'un risque pour la sécurité et la fiabilité : le shunt utilisé est une résistance de 0,1  $\Omega$ , avec une dissipation maximale de 2 W. C'est bien trop peu pour des impulsions de 80 à 100 A, et la résistance risque de claquer sans que l'on s'en aperçoive.

La littérature, p. ex. [3], nous apprend que la résistance interne d'une batterie





dépend de sa température, de son état de charge (*State of Charge – SOC*) et de son âge. Elle augmente lorsque la température diminue, lorsque la batterie est déchargée, et avec l'âge de la batterie.

Pour pouvoir comparer les valeurs de la résistance interne dans diverses circonstances, il faut veiller à ce que les conditions de la mesure restent identiques, et ne faire varier qu'un seul paramètre lors des mesures successives. Une résistance élevée à température ambiante indique que la batterie est soit vieille,

soit déchargée, mais dans tous les cas, une mesure de cette résistance interne et de la tension aux bornes de la batterie donnera une bonne indication de l'état de la batterie elle-même.

### Régénérateur de batteries au plomb de 2, 6, 12 et 24 V

Le circuit dont le schéma est reproduit en **figure 1** est bâti autour d'un microcontrôleur, qui envoie toutes les 30 s une impulsion de 100 µs sur la grille du MOSFET T1. Un shunt de 50 mΩ est branché aux bornes de la batterie, il est

en série avec la jonction drain-source de T1 et la diode Schottky D9, qui protège aussi T1 contre les inversions de polarité. Il en résulte un courant de pointe théorique d'environ 100 A pour un accu au plomb de 6 V.

La mesure de tension aux bornes de la batterie s'effectue via un circuit séparé (technique des quatre points), afin qu'elle soit le moins possible affectée par les impulsions de forte intensité ; seules les bornes de la batterie sont donc communes aux circuits de mesure et de décharge. L'intensité du courant est

**Tableau 1. Les trois LED donnent une indication de l'état de charge des accumulateurs au plomb.**

Plage	0-10 V			0-20 V	0-30 V
Tension	V < 2,5	2,5 < V < 5	5 < V < 10		
LED	accu 2 V	accu 4 V	accu 6 V	accu 12 V	accu 24 V
rouge	< 1,98 V	2,5 à 4 V	5 à 5,95 V	< 11,9 V	< 23,8 V
orange	1,98 à 2,08 V	4 à 4,16 V	5,95 à 6,25 V	11,9 à 12,5 V	23,8 à 25 V
verte	2,08 à 2,5 V	4,16 à 5 V	6,25 à 10 V	> 12,5 V	> 25,0 V





**Tableau 2. Aperçu des alarmes et arrêts du programme.**

	LED	Clignotement	Action
Vraisemblablement circuit ouvert	verte	Lent (1 s)	Attendre la résolution du problème
Pas de tension aux bornes*	verte	Rapide (400 ms)	Attendre la résolution du problème
Tension aux bornes hors de la plage de mesure	orange	Rapide (400 ms)	Arrêt du programme, <b>il faut déconnecter la batterie immédiatement</b>
Courant trop intense	rouge	Rapide (400 ms)	Arrêt du programme, il faut ajouter une résistance en série $R_{ext}$ et redémarrer
* Ceci ne peut se produire que lorsque le microcontrôleur est alimenté par une source externe, pour des tensions inférieures à 5,5 V.			

déduite de la mesure de la tension aux bornes du shunt.

Les impulsions sont produites par le microcontrôleur ; elles attaquent la grille de T1 via un photocoupleur et sont matérialisées par LED 1, de couleur bleue. Des LED de couleur rouge, orange et verte donnent une indication de la tension aux bornes de batteries au plomb de 6, 12 ou 24 V ; le **tableau 1** récapitule ces indications.

Notre régénérateur pourrait aussi être utilisé pour des accumulateurs de 2 V, mais il faudrait une alimentation externe pour délivrer les impulsions. Un convertisseur de 2 à 5 V pourrait s'en charger, mais les LED ne pourraient de toute manière pas donner d'indication sur l'état de l'accu.

### Mesure de la qualité de batteries de 0,2 à 30 V

À partir de maintenant nous n'utiliserons que le terme batterie(s), ce qui inclut bien entendu les accus.

Avec un shunt unique de 50 mΩ, le courant peut en théorie varier de 40 à 600 A. Dans la pratique, ce courant est limité par la résistance – bien plus élevée – de l'ensemble du circuit, qui comprend les liaisons et contacts, la diode Schottky D9, le MOSFET T1 et la résistance interne de la batterie. Pour des batteries de 0,2 à 10 V, des impulsions de 10 A sont adéquates, 100 A pour des tensions plus élevées. Le courant maximal peut encore être réduit par l'ajout d'une résistance externe en série avec le circuit de décharge. On peut aussi choisir un shunt de valeur différente : 50, 100 ou 220 mΩ pour des

plages de respectivement 0,2 à 10 V, 5,5 à 20 V, et 5,5 à 30 V. Rappelons que pour des tensions inférieures à 5,5 V, une alimentation externe est nécessaire.

L'entrée du convertisseur A/N (CA/N) du microcontrôleur ne peut dépasser 5 V, nous utilisons donc un diviseur de tension de rapport 1:2, 1:4 ou 1:6 pour les trois plages de tension ci-dessus. Les rapports sont sélectionnés à l'aide de cavaliers. Le programme peut s'arrêter dans diverses circonstances, reprises dans le **tableau 2**. La cause est affichée sur l'écran LCD, et est également indiquée par des LED. La cause principale de dysfonctionnement est un courant trop élevé ; si c'est le cas, la LED rouge s'allumera et T1 sera bloqué. Le circuit devra être redémarré.

Des valeurs négatives sont possibles si la batterie est connectée à l'envers. Les valeurs dépendront de la charge.

Au démarrage, les LED rouge, orange et verte clignotent alternativement.

Il peut être utile d'accroître la fréquence des impulsions de courant si vous souhaitez tester plusieurs batteries à la suite. Un cavalier permet de sélectionner une période de 10 ou 30 s, mais pour des accumulateurs au plomb cette dernière est impérative.

### Sécurité

Lorsque l'on travaille avec des courants de 100 A ou plus, il est indispensable de prendre quelques mesures de précaution. Celles-ci concernent essentiellement le shunt, pour lequel nous devons veiller à trois aspects : deux actions et un contrôle.

1. Les connexions du shunt, entre autres les pistes de circuit imprimé de l'ensemble du circuit de décharge, doivent pouvoir supporter les impulsions de courant. Les pistes de cuivre doivent être épaissies avec de la soudure. Les longs fils entre la diode Schottky,

le shunt et le MOSFET doivent être reliés directement entre eux. L'espace qui reste entre les fils et les pistes de cuivre doit être rempli avec de la soudure.

2. Les fils dans lesquels un courant intense circule doivent être le plus courts possible. Il ne faut surtout pas faire de boucles, ce qui risquerait d'entraîner la création d'un champ magnétique qui induirait un courant additionnel. Le programme s'arrête bien lors d'un courant trop élevé, mais seulement après la mesure dudit courant. Donc trop tard : le mal est déjà fait !
3. Si le MOSFET conduit trop longtemps ou est défectueux, le fusible F1 fondra. Pour des batteries de 1,2 V, nous recommandons une valeur nominale de 3 A ; pour des tensions supérieures, 10 A.

### Schéma

La partie du circuit où le courant intense circule est constituée du MOSFET T1, de la diode D9, qui protège aussi T1, et du shunt R27, une résistance non inductive. Une tension de grille  $V_{gs}$  de 5,5 V est nécessaire pour la commutation de 100 A par T1, 9 à 10 V pour un courant jusque 200 A. Le temps de commutation de T1 est de l'ordre de 10 ns, mais il sera en pratique plus long, suite à l'utilisation d'une résistance de grille.

Le MOSFET choisi a une faible charge de grille  $Q_g$  : 7,6 nC à 4,5 V. Ceci donne une capacité de

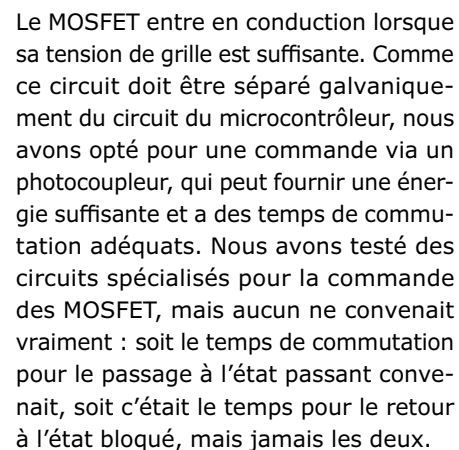
$$C = Q / V = 7,6 / 4,5 \approx 2 \text{ nF.}$$

Avec une résistance de 1 kΩ (R22), la constante de temps RC sera de

$$T_{on} = 10^3 \times 2 \times 10^{-9} = 2 \text{ } \mu\text{s.}$$

En ajoutant R21, nous aurons un temps de décharge  $T_{off}$  de 4 μs. Ces valeurs sont parfaites pour des impulsions de 100 μs.

Figure 1. Le schéma montre les divers ensembles du circuit : alimentation, circuit MOSFET, diviseur de tension et cavaliers, microcontrôleur et écran.



## 124



stance R22 et la diode Zener D10 protègent la grille de T1 contre les surtensions. R21 réduit le temps de retour à l'état bloqué.

Pour permettre les mesures sur des batteries au plomb de 6 V sans source externe, l'alimentation 5 V est fournie via un régulateur à faible tension de déchet (IC1, un LP2950C), qui ne nécessite que 5,3 V en entrée pour un débit de 30 mA. Pour des tensions inférieures à 5,5 V (5,3 V plus une chute de 0,2 V aux bornes d'une diode Schottky), une alimentation externe est nécessaire, via K2 qui est en parallèle avec le circuit de mesure de tension (K1). Attention à ne pas interchanger K1 et K2, ni à se tromper dans le sens de connexion de la batterie : une tension négative à l'entrée du microcontrôleur pourrait avoir des conséquences fâcheuses.

Les LED 2 (rouge), 3 (orange) et 4 (verte) donnent une indication de l'état de charge de la batterie au plomb. La LED 1 (bleue) suit les impulsions de courant ; la durée d'illumination est accrue, sinon les impulsions ne seraient pas visibles. Ces quatre LED ne sont commandées que par deux sorties du microcontrôleur. En mettant deux LED en série avec deux diodes Zener de 2,4 V (D3 + D4 ou D5 + D6, chaque couple de 4,8 V), aucune ne sera allumée sous 5 V (sorties du microcontrôleur à l'état haute impédance). Lorsque les deux sorties du microcontrôleur sont hautes ou basses, la tension est suffisante pour la ou les LED à allumer. L'écran LCD donne bien entendu plus d'informations que les LED.

IC2, le microcontrôleur utilisé, est un PIC16F1847, le plus puissant dans sa catégorie (8 bits, 18 broches) chez Microchip. La sortie B3 commande T1 via IC4. Les tensions sont mesurées grâce aux trois CA/N du microcontrôleur : juste avant l'émission d'une impulsion la tension aux bornes de la batterie hors charge ( $V_0$ ), et pendant les impulsions la ten-

sion aux bornes de la batterie en charge ( $V_c$ ) et aux bornes du shunt ( $V_h$  et  $V_l$ ). Lors de la première utilisation, la tension entre le drain (TP1) du MOSFET et sa source (contact - de K3) doit être mesurée ; celle-ci NE peut PAS chuter lors de la mise en marche.

Le courant nécessaire à l'alimentation du microcontrôleur, de l'écran LCD, des LED, et du régulateur de tension 5 V est d'environ 7 mA. Le convertisseur CC-CC a lui besoin de 16 à 19 mA ; ce n'est pas très économe, mais il peut encore fournir 9 V lors de courtes mesures sur des batteries de 1,2 ou 1,5 V.

### Mesures de tension et cavaliers

Les cavaliers JP2, JP3 et JP4 permettent de sélectionner le rapport des diviseurs de tension (R2 à R19), la tension à l'entrée des CA/N (mesure de  $V_c$ ,  $V_h$  et  $V_l$ ) qui ne peut pas dépasser 5 V. Les tensions jusqu'à 30 V seront divisées par 6 grâce à six résistances en série (cavaliers en position ouverte). Pour des tensions n'excédant pas 20 V, une division par quatre est suffisante (deux résistances court-circuitées par les cavaliers en position H), et pour celles inférieures à 10 V une division par deux (quatre résistances court-circuitées par les cavaliers en position L).

Le cavalier JP1 fonctionne en entrée

à trois états : haut, bas ou ouvert. Il indique au microcontrôleur l'état de JP2 à JP4. La résistance de 10 k $\Omega$  est nécessaire, car l'entrée peut fonctionner brièvement en sortie pour déterminer l'état, et ne peut donc être reliée directement au + 5 V ou à la masse.

Les cavaliers JP1 à JP4 sont parfaitement alignés les uns à côté des autres, et nous avons « construit » un cavalier quadruple pour les mettre tous ensemble dans la même position (**fig. 2**).

Nous l'avons déjà signalé, JP5 permet de sélectionner une période de répétition de 10 s (avec cavalier) ou 30 s (sans cavalier) ; pour des batteries au plomb, cette dernière est impérative.

Il est important que la mesure des trois tensions en charge s'effectue pendant la période de production des impulsions de 100  $\mu$ s. Ceci dépend tant du logiciel que du matériel, et les composants critiques sont le photocoupleur, le MOSFET et le microcontrôleur. Le photocoupleur choisi, un VO615A, a des temps de commutation de 9 à 10  $\mu$ s pour une charge de 1 k $\Omega$  ; il reste donc 80  $\mu$ s (100 - 2 $\times$ 10) pour la mesure.

Avec une fréquence d'horloge de 16 MHz, il faut au minimum 1  $\mu$ s pour convertir un bit ( $T_{ad}$ ), soit 11,5  $\mu$ s pour les 10 bits du CA/N. Le temps d'acquisition  $T_{acq}$  est

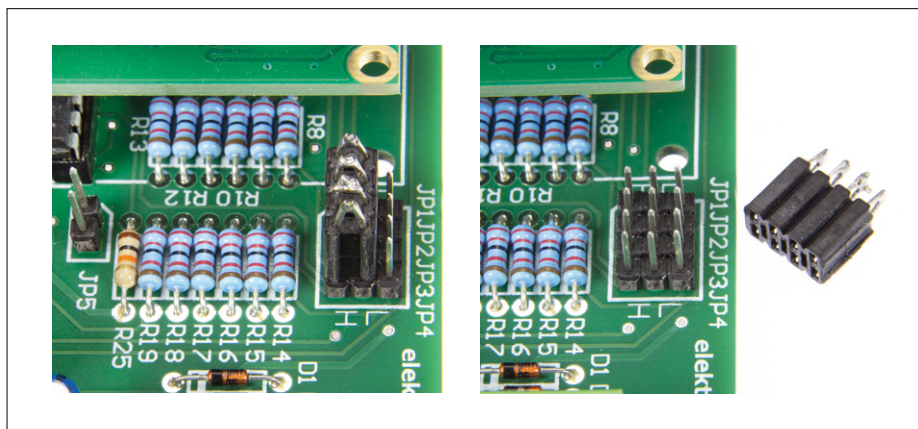


Figure 2. Pour la sélection du rapport du diviseur de tension, nous avons utilisé une combinaison de quatre cavaliers.



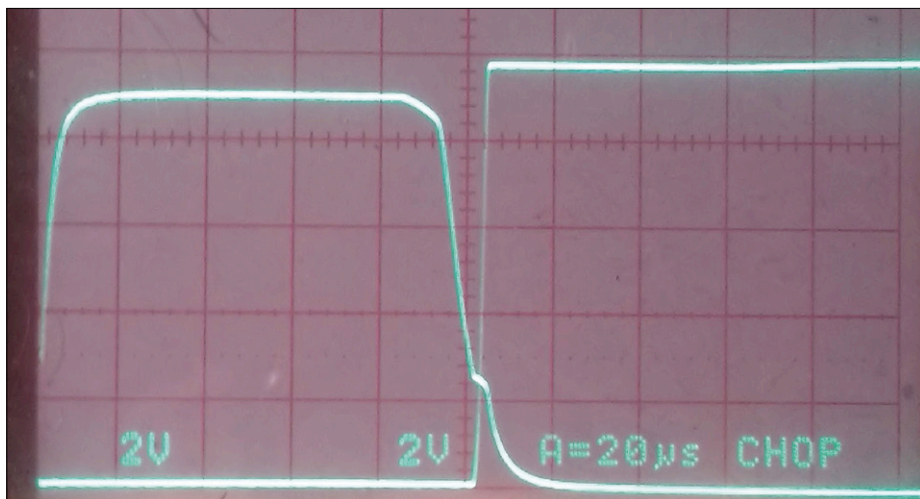


Figure 3. Courbe de gauche : tension de grille  $V_{gs}$  de T1 (impulsion de 94  $\mu$ s). L'autre courbe montre la tension drain-source  $V_{ds}$  de T1 pour une durée d'env. 105  $\mu$ s ( $V_{gs} > 3$  V pour un courant de shunt de 0,4 A et  $V_0 = 10$  V).

égal à la somme du temps d'établissement de l'ampli, du temps de charge du condensateur de l'échantillonneur ( $T_c$ ), et d'un retard dépendant de la température :  $2 \mu$ s +  $T_c$  +  $0,05 \Delta T$  (par rapport à 25 °C).  $T_c$  dépend entre autres de la résistance de source  $R_s$  ; pour ce microcontrôleur-ci :

$$7,62 \times 10 \text{ pF} \times (8 \text{ k}\Omega + R_s).$$

Pour une valeur optimale de  $R_s$  de 10 k $\Omega$ ,  $T_c$  est de 1,4  $\mu$ s. Dans notre circuit,  $R_s$  varie entre 11 k $\Omega$  (22 k $\Omega$  // 22 k $\Omega$ ) et 18 k $\Omega$  (22 k $\Omega$  // (5x22 k $\Omega$ )),  $T_c$  est donc de 1 à 2  $\mu$ s. Le retard dû à la différence de température est négligeable.  $T_{acq}$  est donc d'environ 16  $\mu$ s ( $11,5 + 2 + 2$ ), mais nous avons prévu dans le programme 10  $\mu$ s pour  $T_c$ , ce qui nous donne finalement un  $T_{acq}$  de 24  $\mu$ s. Il faut donc 72  $\mu$ s pour trois mesures, ce qui est infé-

rieur aux 80  $\mu$ s disponibles !

Le programme a néanmoins besoin de 0,25  $\mu$ s par pas en plus (activation du CA/N, lecture, sauvegarde du résultat, désactivation). Avec les procédures standard de *Flowcode* 6, nous avons besoin de 150  $\mu$ s pour les trois mesures, ce qui est trop long. Nous avons remplacé les procédures de *Flowcode* par des procédures en C, ce qui a permis une réduction du temps nécessaire de 33% ; il faut dès lors 100  $\mu$ s, ce qui est tout juste.

Les temps de commutation estimés du circuit de mesure sont conformes aux temps effectifs (**fig. 3**). Les temps de montée et de descente de  $V_{ds}$  (respectivement 0,8  $\mu$ s et 0,3  $\mu$ s, ce qui n'est pas mal du tout vu les courants en jeu) sont inférieurs à ceux de  $V_{gs}$ , ce qui est aussi favorable. Le MOSFET n'a pas besoin de radiateur.

### Logiciel

Le programme a été écrit avec *Flowcode* 6. Les fichiers *Flowcode* et *.hex* peuvent être téléchargés sur la page de l'article [4]. Au démarrage, l'écran montre le rapport du diviseur de tension, la plage de tension, la valeur du shunt, et la fréquence de répétition ; on attend ensuite la mesure d'une tension aux bornes du shunt. Un écran standard, après le démarrage, est constitué comme suit : tension hors charge  $V_0$  et tension en charge  $V_t$  à la première ligne, courant en A et résistance interne en m $\Omega$  à la seconde ligne, et à droite à la dernière ligne le temps restant jusqu'à l'impulsion suivante. Les valeurs sont toutes affichées avec minimum trois chiffres ; la précision de la mesure dépend bien entendu de divers facteurs, mais elle est suffisante pour établir une relation entre résistance interne et état de charge de la batterie.

Après connexion d'une batterie, le programme attend 1 s pour permettre aux tensions de se stabiliser ; les LED 2 à 4 clignotent alternativement.

Si vous souhaitez modifier le programme *Flowcode*, il est indispensable de positionner l'horloge sur « interne » et à 16 MHz, et de désactiver les fonctions tension basse et MCLR (*Master Clear Pin External Reset*) externe. Codes de configuration : *config1 0x09A4* et *config2 0x1CFF*.

### Quelques résultats

Nous avons testé l'appareil sur la batterie d'une de nos voitures, dont le chauffeur ne fait souvent que de courts trajets. La résistance interne était de 6 m $\Omega$  ; ce n'est pas ce qu'on fait de mieux, mais la voiture démarre toujours sans problème. L'exemple de la **figure 4** illustre bien



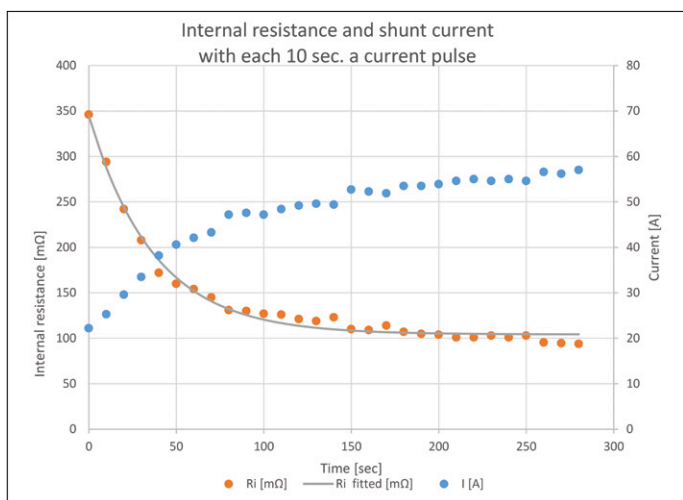


Figure 4. Effet de régénération pour une batterie au plomb défectueuse (12 V, 7 Ah), dont la tension aux bornes était de 10,3 V.

l'effet des impulsions de courant sur une autre batterie : après quatre à cinq impulsions, la valeur de la résistance interne a déjà diminué de moitié.

Pour un accu plomb-acide de 12 V et 7 Ah, la valeur de la résistance interne diminue d'environ 0,7 mΩ par °C : 34 mΩ à température ambiante, et 62 mΩ dans un congélateur (-18 °C).

Un accu plomb-acide neuf de 12 V et 7 Ah présente, après charge complète, une résistance interne de 34 mΩ (± 2%) ; le même type, âgé de 8 ans, a une résistance interne de 52 mΩ (± 3%). La tension aux bornes des deux accus est la même : 13,2 V.

Un ensemble de douze accus NiMH Sub-C de 4 600 mAh, reliés entre eux par des cosses à souder, a une résistance interne de 101 mΩ lorsqu'ils sont déchargés,

86 mΩ après charge jusque 2 000 mAh. Quelques remarques sur la précision des mesures. La variation des résultats des mesures sur les accus au plomb est inférieure à 1%, ce qui est excellent. Pour les accus NiMH et les piles alcalines (AA ou AAA), nous avons observé des variations de 10% et plus. Il faudra donc effectuer plusieurs mesures et calculer la moyenne, mais pas de panique : en une minute, on a déjà six résultats !

Nous avons aussi mesuré les piles alcalines AA de quatre marques différentes (figure 5). Ce qui frappe c'est la disparité entre les marques, et c'est les piles d'une marque très connue qui affichent les moins bons résultats. C'est peut-être une coïncidence, mais on voit en tous cas très bien l'influence de la décharge sur la résistance interne. Une chose est sûre, il faudrait une étude plus poussée

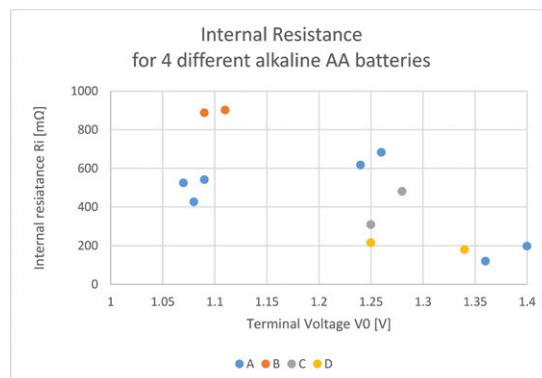


Figure 5. Résistance interne de piles alcalines AA (LR6) en fonction de leur tension à vide. Les piles de quatre marques différentes ont été mesurées : A, B, C et D.

pour expliquer les divergences entre les marques. Alors, au charbon avec notre appareil ! ◀

(160064 – version française : Jean-Louis Mehren)

## DANS L'E-CHOPPE

→ 160064-1  
Circuit imprimé

→ 160064-41  
Microcontrôleur programmé

→ 160064-71  
Kit de composants

→ 120061-74  
Écran LCD de 2×16 caractères

## Liens

- [1] [www.conrad.fr/ce/fr/product/191123/Contrôleur-de-batterie-kit-monte-H-Tronic-191123-12-VDC-1-pcs](http://www.conrad.fr/ce/fr/product/191123/Contrôleur-de-batterie-kit-monte-H-Tronic-191123-12-VDC-1-pcs)
- [2] [www.chargingchargers.com/tutorials/battery-desulfation.html](http://www.chargingchargers.com/tutorials/battery-desulfation.html)
- [3] [www.mpoweruk.com/performance.htm](http://www.mpoweruk.com/performance.htm)
- [4] [www.elektormagazine.fr/160064](http://www.elektormagazine.fr/160064)



# bienvenue dans votre e-choppe

la rédaction recommande



## Aide à l'inspection : microscope USB Andonstar

Si vous réalisez des prototypes avec des composants CMS, vous savez qu'il est difficile de débusquer à l'œil nu les mauvaises soudures, ponts.... Une loupe permet de détecter certains défauts, mais pas tous. De plus si votre instrument grossissant est très près du circuit imprimé, vous n'avez plus de place pour travailler avec le fer.

Oubliez le contrôle des soudures aux rayons X, optez pour le microscope USB Andonstar V160. Les

images capturées par la caméra entourée de LED sont envoyées vers un ordinateur ou une tablette où elles sont stockées, imprimées... Le modèle Andonstar V160 se distingue par le fait qu'il est livré avec un support stable, ainsi l'image n'est pas floue à la moindre vibration. Les boutons permettent de régler avec précision la position de la caméra.

Luc Lemmens (labo d'Elektor)



[www.elektor.fr/andonstar-usb-microscope](http://www.elektor.fr/andonstar-usb-microscope)

## Vos favoris :

### 1. Déwattteur

[www.elektor.fr/d-watter](http://www.elektor.fr/d-watter)



### 2. Red Pitaya STEMLab

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

### 3. Elektor Uno R4

[www.elektor.fr/elektor-uno-r4](http://www.elektor.fr/elektor-uno-r4)

### 4. Horloge de sable Arduino

[www.elektor.fr/horloge-de-sable-arduino](http://www.elektor.fr/horloge-de-sable-arduino)

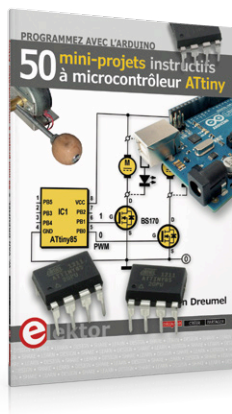
### 5. Carte de commande LEGO pour Raspberry Pi

[www.elektor.fr/lego-rpi-board](http://www.elektor.fr/lego-rpi-board)

### 6. 36 Expériences de Physique avec Arduino

[www.elektor.fr/arduino-36-experiences](http://www.elektor.fr/arduino-36-experiences)

## 50 mini-projets instructifs à µC ATtiny



Pour mettre au point des montages à microcontrôleur, la carte Arduino est parfaite. Ensuite, on peut obtenir le même résultat avec un autre µC, plus petit et moins cher, comme l'ATtiny. Ce livre montre comment faire à l'aide de 50 applications originales, voire déjantées. Dans ces pages, ni théorie ni mathématiques, que de la pratique ! ATtiny programmé avec une carte Arduino ; croquis disponibles sur cette plateforme transposables pour l'ATtiny.



Prix (membres) : 22,41 €

[www.elektor.fr/50-mini-projets-attiny-arduino](http://www.elektor.fr/50-mini-projets-attiny-arduino)

## Red Pitaya for Test & Measurement



Red Pitaya est un instrument de mesure à code source ouvert, de la taille d'une carte de crédit. Il peut remplacer nombre d'instruments de mesure de laboratoire coûteux. Avec ce livre (en anglais), les débutants découvriront les principes de l'électronique et leurs applications, tout en expérimentant avec Red Pitaya : théorie, mais aussi utilisation des composants électroniques, le tout accompagné d'expériences amusantes et intéressantes.



Prix (membres) : 31,45 €

[www.elektor.fr/red-pitaya-livre](http://www.elektor.fr/red-pitaya-livre)

## DVD Elektor 2016



Ce DVD-ROM réunit tous les articles d'ELEKTOR parus au cours de l'année 2016, en français (mais aussi en anglais, allemand et néerlandais). Il contient le texte des articles ainsi que les schémas, et tous les dessins des circuits imprimés, sous forme de fichiers à haute résolution. Une fonction de recherche dans la table des matières vous permet de trouver immédiatement l'article souhaité.



Prix (non-membres) : 27,50 €

[www.elektor.fr/dvd-2016](http://www.elektor.fr/dvd-2016)





# livre et kit pour **BBC micro:bit**

## 35 projets éprouvés avec des programmes écrits en microPython ou TouchDevelop

Ce livre (en anglais) permet de prendre en main la carte micro:bit conçue par la BBC et plusieurs acteurs de l'industrie électronique, et destinée à l'enseignement de la programmation.

La carte animée par un ARM Cortex-M0+ (Nordic) est bien équipée : émetteur-récepteur BLE, matrice de 25 LED, accéléromètre, magnétomètre, deux boutons-poussoirs, connecteur d'extension pour accéder au bus I2C et à plusieurs broches GPIO, plages de connexion pour fiches banane.

Le kit de plus d'une trentaine de composants permet de réaliser les 35 projets détaillés dans le livre et programmés avec TouchDevelop ou en MicroPython.

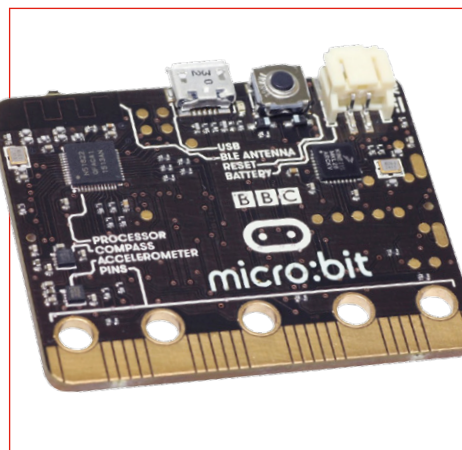
**Attention** : carte BBC micro:bit à commander séparément.



Prix (membres) : € 49,65

[www.elektor.fr/bbc-micro-bit-offre-groupee](http://www.elektor.fr/bbc-micro-bit-offre-groupee)

### BBC micro:bit



La carte BBC micro:bit est compacte et programmable dans différents langages, idéale pour faire ses débuts dans l'électronique programmée sans être intimidé. Elle combine capteurs (magnétomètre, accéléromètre), boutons, LED et Bluetooth. Chaque élément est programmable sans restriction grâce à un logiciel facile à utiliser, disponible sur un site consultable à partir d'un PC/téléphone. Concevez des jeux, connectez-vous à l'Internet des Objets...



Prix (membres) : 15,26 €

[www.elektor.fr/bbc-micro-bit](http://www.elektor.fr/bbc-micro-bit)

### Tuto Arduino Uno 45 projets électroniques originaux



Alarme silencieuse, détecteur de présence, luxmètre, commande de moteur, accès depuis l'internet, commande sans fil sont quelques-uns des projets Arduino Uno brillants et passionnants de ce livre, vérifiés et testés par l'auteur. Ces 45 projets sont faciles à réaliser et bien décrits (schémas, photos). Les explications et la description des composants utilisés facilitent l'adaptation des montages à ses besoins.



Prix (membres) : 31,32 €

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

### Horloge de sable Arduino



Ce fascinant gadget construit autour d'une carte Arduino Uno donne l'heure en traçant les chiffres dans le sable comme on le fait avec le doigt sur la plage. Les afficheurs à 7 segments et les aiguilles habituelles d'une horloge sont remplacés par un bac de sable, un stylet actionné par un pantographe et deux petits moteurs. Ce sont leurs vibrations qui toutes les minutes effacent en quelques secondes les chiffres de l'heure précédente.



Prix (membres) : 107,96 €

[www.elektor.fr/horloge-de-sable-arduino](http://www.elektor.fr/horloge-de-sable-arduino)

# Hexadoku casse-tête pour elektorniciens

Votre magazine se termine toujours et encore par une grille de sudoku particulière. Attrapez un crayon pour remplir la grille d'hexadoku. N'oubliez pas d'envoyer votre réponse pour tenter de gagner un bon d'achat.

À cause du passage au rythme bimestriel, nous annonçons ici les gagnants de l'hexadoku de décembre 2016. La solution de la grille du numéro de décembre 2016 est : **C246E**  
Les trois bons Elektor d'une valeur de 50 € vont à Frank **Hüllmandel** (Allemagne), Philippe **Monnard** (Suisse) et Christoph **Niessen** (Allemagne)

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 !

Envoyez votre réponse (les chiffres sur fond grisé) avec vos coordonnées par courriel, avant le **23 mars 2017** à l'adresse **hexadoku@elektor.fr**

## Les gagnants

La solution de la grille du numéro de janvier/février est **28D45F**

Les trois bons Elektor d'une valeur de 50 € vont à :

Jean-Claude **Carré** (France), Stefan **Falcke** (Allemagne) et Harald **Metzen** (Allemagne)

Bravo à tous les participants et félicitations aux gagnants !

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

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1					F														9
2		8	7													0	F		
3		B	C	4											7	E	1		
4		1	6	2	3			9	0	4			E	A	5	D			
5		D	5	9	E	C	B	7	1	6	A	8	4	3	2				
6		0	A	8	2	E	4	3	7	9	F	B	6	C	D				
7		4	0	1	5	6	D	F	8	B	3	2	9	A	7				
8		5	E	B					1	0	2		D			4	C	6	
9		2	6	D															
10		6	C	2															
11		F	7	9	A														
12		A	3	4	6	F	B	1	C										
13		0	B	2	8	D	4	5	F	9									
14	D	1	9	E	B	7	A	8	0	2									
15		4	3	A	1	E	B	0	5										
16		9	F		C	D	7	8	A	E	3	B	6	1					
17					F	9	1	3	6	C	5	2	0						
18																			
19																			
20																			

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.

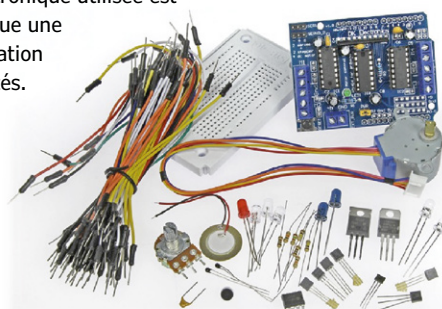
## 36 Expériences de Physique avec Arduino + Kit de démarrage



La rencontre de la physique et du microcontrôleur ne devrait plus étonner personne. Il existe d'excellents enregistreurs de données, ainsi que de nombreux programmes pour les traiter et les présenter sous forme de graphiques colorés et attrayants. La physique rébarbative, c'est fini !

J'ai choisi l'Arduino, car cette plate-forme est d'un accès facile et sa documentation abondante. La famille Arduino offre des ressources extraordinaires à un prix dérisoire. Ajoutez-y le logiciel gratuit CoolTerm, et vous pouvez enregistrer toutes les données de mesure pour les retravailler sous Excel et créer aisément des tableaux ou des graphiques.

Ce livre n'est pas un manuel de physique. Vous n'y trouverez ni équations différentielles ni courbes abstraites. Nous étudierons des phénomènes physiques de la vie de tous les jours. Sans chercher à être exhaustif, mon modeste ouvrage apporte aux leçons de physique une approche nouvelle et créative grâce aux techniques modernes de mesure et de traitement des données. L'électronique utilisée est simple, et constitue une belle démonstration des possibilités.



Un kit a été spécialement conçu pour réaliser les expériences décrites dans ce livre.

Contenu détaillé du kit, voir [www.elektor.fr/159009-71](http://www.elektor.fr/159009-71)

[www.elektor.fr/arduino-36-experiences](http://www.elektor.fr/arduino-36-experiences)

Après avoir magnifiquement expliqué **l'électronique aux débutants qui sèchent les cours**, dans son livre qui porte le même titre, Rémy Mallard revient avec un nouveau livre dans lequel il présente ...



## les microcontrôleurs PIC pour les débutants qui veulent programmer sans patauger

Ce livre initie à la programmation des microcontrôleurs PIC avec des applications pratiques qui vont bien au-delà du simple chenillard à LED et couvrent un grand nombre de besoins. Après une introduction (pas trop longue) aux principes essentiels de la programmation, tu apprends à interfacer des capteurs avec un microcontrôleur, à acquérir et stocker ces données, ou encore à établir une liaison USB ou Ethernet pour transmettre des données. Rémy regroupe par chapitres les informations théoriques et pratiques nécessaires à la réalisation de chaque montage décrit.

Après la lecture, gagné par la bonne humeur communicative de l'auteur et fort de sa longue expérience (qu'il partage volontiers), tu n'auras qu'une seule envie : aller plus loin, créer toi-même des montages encore plus ambitieux (transposer le code dans d'autres langages, le porter sur d'autres plates-formes de développement, t'attaquer aux PIC32)...



**Le premier pas coûte, après ça va tout seul.**

**elektor**

ISBN 978-2-86661-193-4 • 48,50 €

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



# La conception d'interfaces graphiques embarquées séduisantes n'a jamais été aussi simple

Des outils de développement visuels et bibliothèques de graphismes gratuits pour vos applications d'interfaces graphiques



Microchip propose un ensemble complet d'outils de développement d'interfaces graphiques utilisateur, de bibliothèques graphiques logicielles et d'outils matériels couvrant tous vos besoins pour la conception de graphismes en 32 bits.

Nos solutions graphiques sont compatibles avec le framework logiciel gratuit MPLAB® Harmony et laissent le choix aux développeurs entre deux outils de pointe :

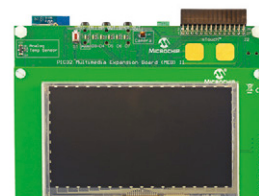
1

Notre configurateur graphique MPLAB® Harmony Graphics Composer (MHGC) fonctionne avec la bibliothèque graphique MPLAB Harmony et permet de créer des interfaces graphiques de qualité professionnelle sans avoir à écrire une seule ligne de code !

OU

2

Vous pouvez choisir SEGGER emWin Pro comme bibliothèque graphique et profiter de sa liste complète de widgets ainsi que de la suite logicielle SEGGER.



Carte de développement multimédia II (DM320005-2)

Démarrez votre développement dès aujourd'hui en téléchargeant le matériel d'essai, la documentation et les outils !

**microchip**  
**DIRECT**  
[www.microchipdirect.com](http://www.microchipdirect.com)



**MICROCHIP**

[www.microchip.com/MCU32GFX](http://www.microchip.com/MCU32GFX)

Le nom et le logo de Microchip, le logo Microchip et MPLAB sont des marques déposées de Microchip Technology Incorporated aux États-Unis et dans d'autres pays. Toutes les autres marques commerciales ci-dessus mentionnées sont la propriété exclusive de leurs propriétaires respectifs. © 2017 Microchip Technology Inc. Tous droits réservés. D560001462A. MEC2006Fre01/17