

station météo en réseau ouvert V.2

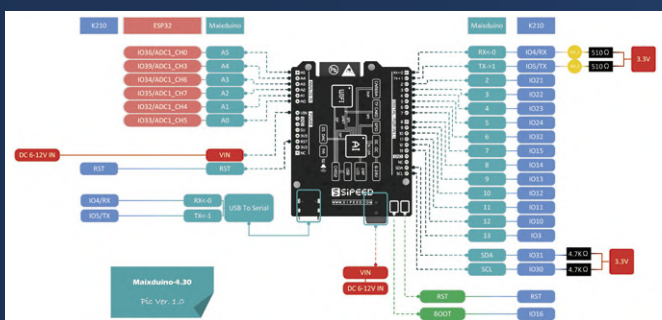
mesurer – enregistrer – transmettre

- capteurs multiples
- données météo sur *ThingSpeak*
- *openSenseMap*



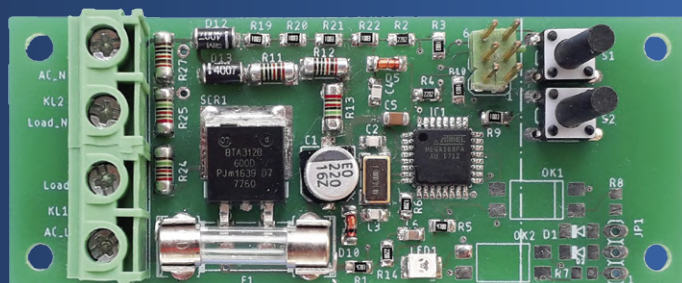
nouvelle série

l'intelligence artificielle pour les débutants



Reconnaissance d'objets avec carte Maixduino

commande de triacs avec ATmega



Commutation et gradation de charges variées

nouvelle horloge Nixie de précision révisée ✂ bus CAN + Arduino pour la surveillance de panneaux solaires ✂ sablier motorisé en BASIC avec ESP8266 et Annex WiFi RDS ✂ boîte inviolable (avec témoin d'effraction) ✂ fonctions FFT des oscilloscopes numériques ✂ banc d'essai : alim de labo PeakTech 6080 A ✂ banc d'essai : charge électronique USB Joy-IT HD35 ✂ SigFox : un renard sur l'internet des objets (4)

L 19624 - 483 - F: 15,50 € - RD



ABONNEZ-VOUS ET RECEVEZ

TOUS LES 2 MOIS, LES DERNIÈRES NOUVELLES DU RASPBERRY PI ET LES MEILLEURS PROJETS !



Vos avantages :

- Prix au numéro réduit
- Chaque numéro directement dans votre boîte aux lettres
- Tous les numéros disponibles sous forme numérique (PDF)
- Découverte de chaque nouveau numéro avant sa sortie en kiosque

SEULEMENT
54,95 €
PAR AN
(6 NUMÉROS)

ABONNEZ-VOUS : WWW.MAGPI.FR

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

Pour toutes vos questions :
service@elektor.fr

www.elektor.fr | www.elektormagazine.fr

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

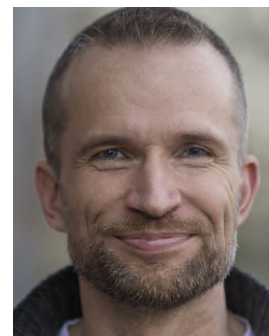
Publicité :
Margriet Debeij
Tél. : +49 (0)241 955 09 174
margriet.debeij@elektor.com

DROITS D'AUTEUR :
© 2020 Elektor International Media B.V.

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

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

Imprimé aux Pays-Bas par
Pijper Media – Groningen
Distribué en France par M.L.P.
et en Belgique par A.M.P.



Soudés, plus que jamais !

Il y a quelques semaines, mon intention était encore de vous parler dans cet éditorial des produits intéressants découverts au dernier salon du monde de l'embarqué que j'avais visité fin février. Aujourd'hui, tout cela semble appartenir à un passé fort lointain. Entre-temps, nous avons dû nous habituer à ne plus nous rencontrer autrement que par le truchement d'images vidéo, sans parler des autres contraintes du confinement. On finit par trouver surréalistes maintenant les vidéos de l'an passé sur des conférences tenues dans des salles bondées. La vie continue, mais sans les marées humaines enthousiastes des festivals de musique.

Avec un peu de créativité, il est possible de créer de nouveaux réseaux. C'est si important en ces temps difficiles. Un collègue a récemment fêté son anniversaire sur la plate-forme vidéo Zoom, avec de nombreux électroniciens, dont certains participaient activement, directement depuis leur labo. Moi-même, avec quelques amis et un équipement technique adéquat pour la musique de DJ, je m'apprête à diffuser en direct, mais nous serons chacun dans son salon respectif. Je suis sûr que vous aussi vous n'êtes pas à court d'astuces pour poursuivre l'échange des idées avec d'autres personnes afin de faire avancer vos projets communs. C'est d'ailleurs aussi la mission d'Elektor Labs d'offrir sur www.elektor-labs.com une plate-forme pour les projets électroniques les plus divers. Je vous invite à découvrir la toute nouvelle présentation du site, la navigation est simplifiée et ses fonctions étendues.

Avec ce numéro, l'édition allemande d'Elektor fête ses 50 ans. La longévité de notre magazine est un phénomène plutôt rare. La proportion de nos lecteurs restés fidèles pendant des décennies y est pour beaucoup. C'est notre plus belle récompense, mais certainement pas une excuse pour nous reposer sur nos lauriers. Au contraire, nous travaillons sur de nouvelles offres et de nouveaux formats pour informer encore mieux tous les électroniciens.

Restons soudés !

Jens Nickel

Rédacteur en chef

Notre équipe

Rédacteur en chef :	Jens Nickel
Rédaction :	Eric Bogers, Jan Buiting, Rolf Gerstendorf, Denis Meyer (traduction), Thomas Scherer, Clemens Valens
Service aux lecteurs :	Ralf Schmiedel
Correcteur technique :	Malte Fischer
Laboratoire :	Mathias Claussen, Ton Giesberts, Luc Lemmens, Clemens Valens (responsable), Jan Visser
Ont participé à ce numéro :	Robert Grignard, Denis Lafourcade, Helmut Muller
Maquette :	Giel Dols

3 Editorial

Restons soudés !

Savoir-faire

- 26 **GitHub pour les ±nuls**
Comment télécharge-t-on depuis GitHub ?
- 36 **interactive**
Corrections - Questions - Réponses
- 42 **bureau d'études – Zone D**
D comme développement, débrouille et dur-à-cuire ! | Trucs et astuces, bonnes pratiques et infos utiles
- 46 **démarrer en électronique (3)**
... est plus facile qu'on ne l'imagine !
- 48 **le retour des petits circuits**
... et des bonnes petites pétiotes d'Elektor
- 86 **électronique analogique (3)**
Étude de cas n° 1 - section 3 : Optimisation de la réponse du préampli et compromis
- 89 **bruits de labo**
Une sélection de projets en gestation qui méritent votre attention
- 90 **analyse de protocole et décodage de données série**
(I²C, SPI, SMBUS, UART, CAN/LIN, I²S, USB 2.0, DMX, etc.)
- 96 **fonctions FFT des oscilloscopes**
Représentation du spectre avec les oscillos à mémoire numérique
- 100 **dis-moi comment tu ranges ton labo...**
Visite guidée dans le saint des saints
- 106 **grand défi : internet des objets ou internet des déchets ?**
Des produits sûrs pour l'IdO
- 114 **hexadoku**
Grille Sudoku pour les elektorniciens

Infos

- 12 **intelligence artificielle pour débutants (1)**
Reconnaissance d'objets à l'aide de la carte Maixduino
- 35 **drôle(s) de composant(s)**
MC14500B : microprocesseur CMOS à un seul bit

station météo en réseau ouvert V.2

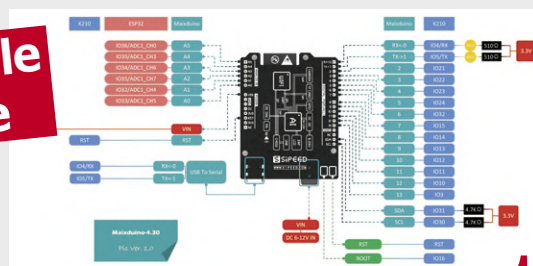
mesurer – enregistrer –

- capteurs multiples
- données météo sur *ThingSpeak*
- *openSenseMap*



intelligence artificielle pour débutants

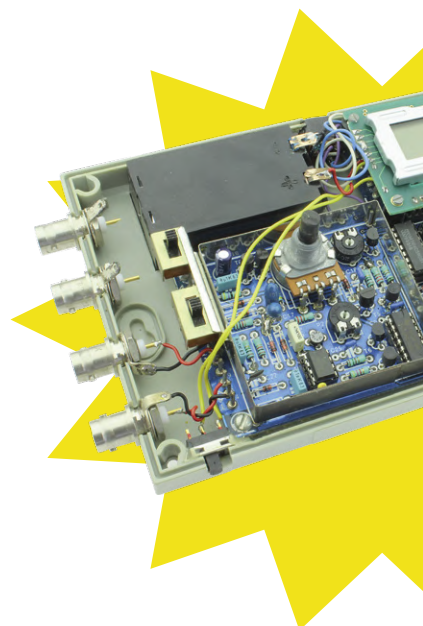
nouvelle série



reconnaissance d'objets à l'aide de la carte Maixduino

12

- 45 **drôle(s) de composant(s)**
Microcontrôleurs TMS1000
- 77 **drôle(s) de composant(s)**
TMS0280 : synthèse vocale
- 78 **banc d'essai :**
charge électronique USB Joy-IT HD35
Pour tester la capacité de charge des ports USB
- 102 **banc d'essai :**
alim de labo PeakTech 6080 A
Une alimentation nourrissante à tout petit budget

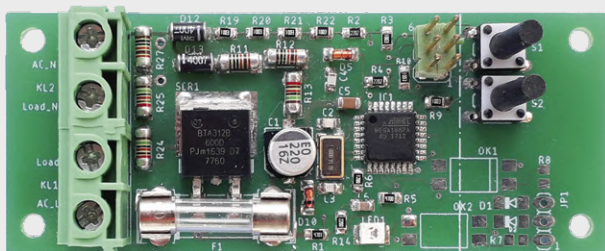


- transmettre

6



commande universelle de triacs avec ATmega



18

commutation et gradation de charges variées



108

- 104 e-choppe Elektor en ligne**
Bonnes affaires de la boutique
www.elektor.fr
- 108 Rétronique : fréquencesmètre 1,2 GHz & générateur de signaux carrés d'Elektor (1992/93)**
Il a franchi le mur du gigahertz
- 113 Start-ups :**
Le programme d'investissement Elektor

Projets

- 6 station météo en réseau ouvert V.2**
1^{ère} partie : présentation et matériel
- 18 commande universelle de triacs avec ATmega**
Commutation et gradation de charges variées
- 28 dessine-moi un bouton pour l'IdO (2)**
Boutonnière n° 2 : Prototypage entre la matière et le nuage
- 38 sablier motorisé en BASIC**
avec ESP8266 et Annex WiFi RDS
- 50 boîte inviolable protégée par un témoin d'effraction**
Envoyer des données en toute sécurité par la poste
- 58 nouvelle horloge Nixie révisée**
Un projet porté par l'enthousiasme de ses utilisateurs
- 64 multitâche en pratique avec l'ESP32 (3)**
Temporisateurs logiciels
- 68 SigFox :**
un renard sur l'internet des objets (4)
Composition d'un tableau de bord
- 82 bus CAN + Arduino pour la surveillance des cellules solaires**
Détecter et localiser les panneaux défectueux dans les grands réseaux photovoltaïques

bientôt dans ces pages

La rédaction et le labo travaillent sur le prochain numéro dont le menu sera copieux. En voici un avant-goût :

Capteur de température sans fil pour le thermomètre bargraphe Nixie - Système de domestique - Nouvelle version du LCR-mètre (avant-première) - Logiciel pour la station météo Mk.2 - Convertisseur S/PDIF - Conception électronique analogique (4) - Alimentation HT avec traceur de courbes - Booster de LED pour micro-contrôleurs - Domotique simplifiée - Petits circuits et astuces - Rétronique.
... et bien davantage

Sous réserve de modifications.

L'édition de juillet-août 2020 d'Elektor paraîtra le 2 juillet 2020.

station météo en réseau ouvert V.2

1^{ère} partie : présentation et matériel

Mathias Claußen et Luc Lemmens (Elektor Labs)

Que vous soyez climatosceptique ou pas, la météo est un sujet de préoccupation mondiale. Une station météo puissante et extensible est intéressante à titre individuel, pour vous et vos proches, mais elle a aussi sa place dans des réseaux mondiaux ouverts comme *ThingSpeak* et *openSenseMap*. La station décrite dans cet article en deux voire trois parties a été conçue dans un souci de flexibilité pour la mesure et l'enregistrement des données, avec une consommation minimale et une grande fiabilité en cas d'exposition aux éléments. Le projet s'articule autour d'un module ESP32 connecté à des capteurs du commerce et fonctionnant avec un logiciel gratuit et truffé de possibilités. Il ne laisse pas grand-chose à désirer, si ce n'est un bon soleil et une douce brise !

Caractéristiques

- Système à 3 cartes : principale ; connecteurs ; particules
- Composants TH et modules commerciaux
- Capteurs (de base) : température, humidité relative, pression de l'air, poussières/particules
- Basé sur ESP32 (Pico Kit)
- Bus I²C
- Alimentation par batterie 12 V (solaire) avec arrêt automatique en cas de sous-tension
- Connecteurs ou borniers à vis pour la plupart des câblages
- Coffret en classe IP66 pour la carte principale et le capteur de particules
- Logiciels libres
- Compatible avec *ThingSpeak*, *openSenseMap*
- Extensible avec des capteurs personnalisés
- Extension LoRa RFM95 (optionnel, en cours de développement)
- Compatible avec *ESPHome* et *Home Assistant*
- Accessible par WiFi
- Carte SD pour la commande locale

Les circuits imprimés de ce projet correspondent aux pièces mécaniques du kit de station météorologique disponible dans la boutique en ligne d'Elektor, composé d'un support et de capteurs (vitesse et direction du vent, pluie, température et humidité relative). Le V2 du titre de cet article est trompeur. Ce projet n'est pas une version remaniée, mais une conception nouvelle de notre précédente station météo ESP32 [1] désormais dépassée. Pour de nouvelles réalisations, nous recommandons exclusivement ce nouveau projet qui comporte trois cartes :

- carte mère (191148-1) pour installation à l'extérieur dans un boîtier étanche monté relativement près de votre domicile.
- carte de connexion (191148-2)

conçue pour s'insérer dans le boîtier d'un capteur thermo/hygro, en remplacement de la carte d'origine, avec prises RJ11 pour la connecter à des modules de capteurs (vent, pluie). Les cartes -1 et -2 sont reliées par 7 fils d'un câble Ethernet classique (qui en a 8 en tout).

- carte de capteur de particules (191148-3) pour le capteur HPM115 (Honeywell), permettant de le monter à distance dans le même type de boîtier que celui utilisé pour la carte principale de la station météo.

Ces trois cartes seront abordées dans cet article, le logiciel dans la 2^e partie. Un 3^e article sera peut-être nécessaire.

Carte mère 191148-1

La carte principale contient un certain nombre d'éléments communs aux capteurs utilisés, facilement identifiables sur le schéma (fig. 1).

Alimentation avec protection de la polarité et verrouillage en cas de sous-tension

Normalement, la station météo est alimentée par une batterie rechargeable

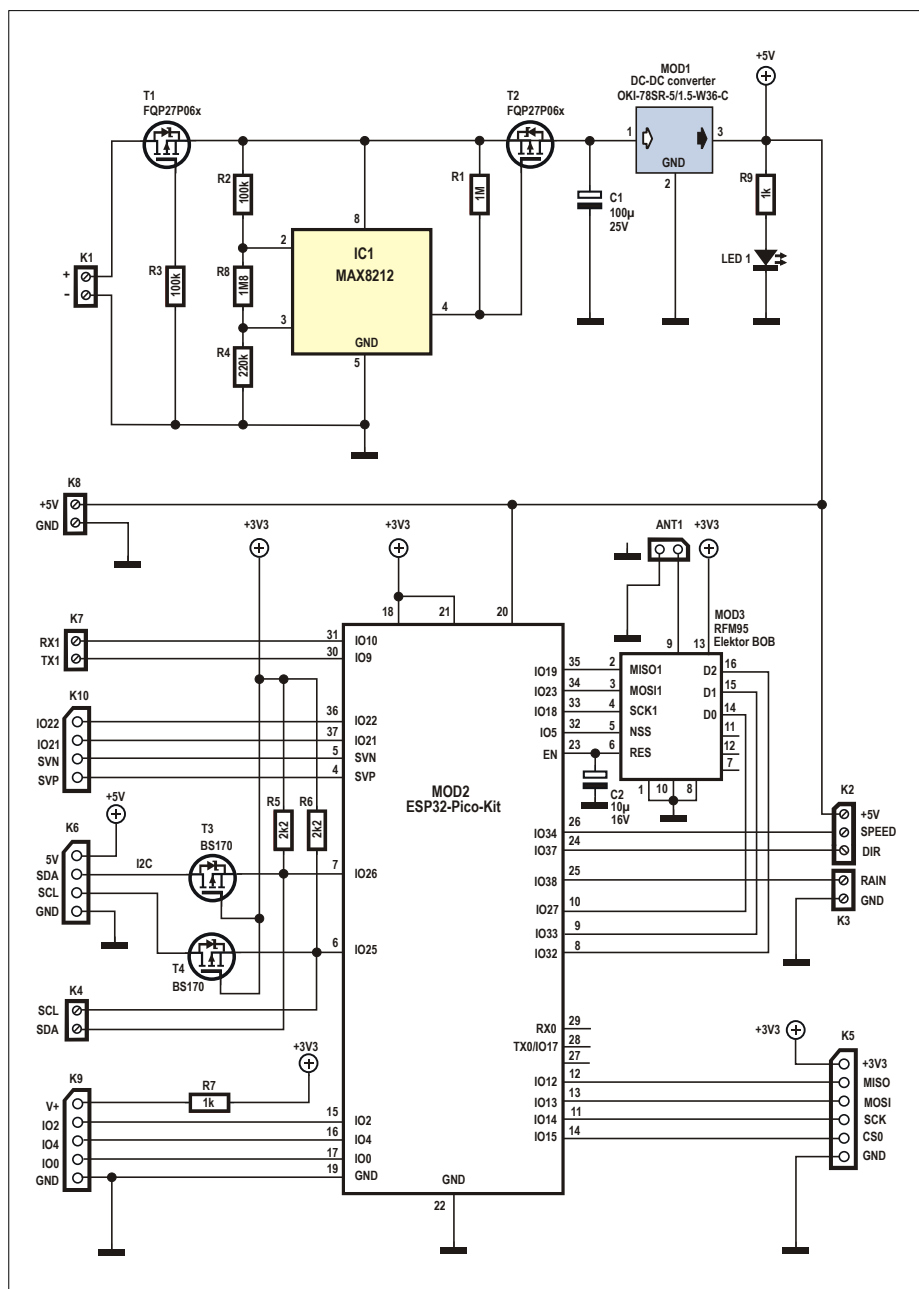


Figure 1. Schéma de la carte principale de la station météorologique en réseau ouvert V.2, compatible avec les plateformes de capteurs mondiales telles que *ThingSpeak* et *openSenseMap*. Notez sur le bus I²C l'adaptation de niveau bidirectionnelle 3,3 V/5 V avec des FET, qui fonctionne parfaitement.

INFOS SUR LE PROJET



capteurs météo

ESP32 ThingSpeak

openSenseMap



débutant

→ connaisseur

expert



3 h environ (assemblage du circuit imprimé)



outils de labo, petite perceuse



60 € (carte mère avec boîtier),
40 € (cône avec BME280),
50 € (capteur de poussière avec boîtier) (prix approximatifs)

de 12 V, qui tire son courant de charge d'un panneau solaire combiné à un chargeur solaire externe. La consommation maximale est de 1,5 A pour la station météorologique entière, en fonction du type et du nombre de capteurs utilisés. La plage de tension d'entrée utilisable est d'environ 11 à 14 V continus.

Le FET T1 et la résistance R3 assurent la protection de la polarité de la batterie, le MOSFET n'étant conducteur que si les bornes de la batterie sont connectées dans le bon sens.

Afin d'éviter d'endommager la batterie, si l'énergie requise par la station météo n'est pas disponible, la batterie devra être déconnectée automatiquement pour éviter une décharge profonde. Nous avons choisi pour cette tâche le bon vieux détecteur de tension programmable ICL8212 dans sa version CMOS moderne de type MAX8212. Avec seulement trois résistances externes pour le réglage des seuils/hystérésis et un MOSFET externe comme dispositif de commutation, le 8212 fait exactement ce que nous voulons : éviter la décharge profonde de la batterie.

Avec les valeurs données pour R2, R8 et R4, le verrouillage de sous-tension déconnecte la batterie lorsque la tension à ses bornes tombe en dessous de 10,5 V. Il la reconnecte quand elle repasse au-dessus de 11,1 V.

Un convertisseur CC/CC (MOD1) abaisse à 5 V la tension de la batterie de 12 V.

La LED1 sert d'indicateur de mise sous tension. Le module Picokit ESP32 est équipé d'un régulateur interne de 3,3 V dont la sortie est reliée aux broches 18 et 21 et qui fournit 3,3 V aux autres éléments de la carte principale.

Cerveau et biceps d'ESP32

L'ESP32 est bon marché mais puissant, avec une offre énorme de (bribes de) logiciel prêt à l'emploi et d'autres briques de logiciel disponibles sur la toile. Ses propriétés d'interfaçage de capteurs sont excellentes aussi. Un module Picokit ESP32 (MOD2) est utilisé pour commander les capteurs et lire leurs données, pour afficher les informations (LCD), pour enregistrer les données (carte SD) et pour communiquer avec un ordinateur (dans la maison) par WiFi (ou LoRa, une fois qu'il est en place). Sur la carte principale, il y a également un bouton de remise à zéro et un bouton *utilisateur*, ainsi qu'une connexion USB pour flasher le micrologiciel de la station.

Carte SD

Pour recevoir la carte micro-SD (K5), nous avons opté pour un petit adaptateur disponible dans l'e-choppe d'Elektor. Son connecteur au pas de 100 mil est facile à souder. Cet adaptateur est monté perpendiculairement à la carte principale pour pouvoir accéder à la carte SD. Soit vous remplacez le connecteur à broches droites de l'adaptateur par un connecteur à broches coudées, soit vous coudez ces broches à 90 ° avec une pince... et du doigté, pour ne pas endommager cette petite carte.

Bus I²C & LCD

Le bus I²C embarqué est utilisé pour communiquer avec l'écran LCD (sur K6) et les capteurs de pression, de température et d'humidité de l'air (sur K4). Il peut accueillir d'autres capteurs ou appareils compatibles I²C que vous souhaiteriez ajouter. Les résistances de polarisation R5 et R6 sont nécessaires sur le bus.

Si vous connectez sur K4 un appareil I²C qui contient déjà des résistances de polarisation, omettez R5 et R6 ou supprimez-les. Le module LCD suggéré [2] est muni de résistances de polarisation sur sa tension d'alimentation de 5 V. Les FET T3 et T4 décalent les niveaux 3,3 V / 5 V dans les deux sens pour les adapter aux broches d'entrée/sortie de l'ESP32 parce que celles-ci ne supportent pas 5 V.

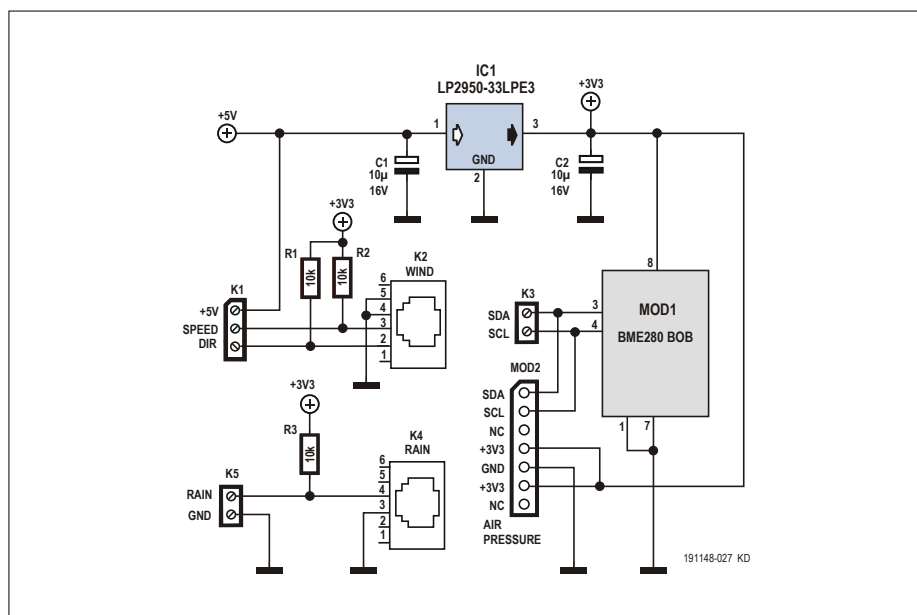


Figure 3. Schéma de la carte de connexion pour les capteurs de pluie et de vent. Il accueille également le module capteur de pression/humidité/température.

Sans oublier...

Le MOD3 est équipé d'un adaptateur RFM95 optionnel pour LoRa. Cette extension n'est pas encore prise en charge dans la version actuelle du logiciel. Les connecteurs K7 et K8 s'interfaçent avec un capteur de particules *Honeywell*.

Comme celui-ci est sensible au gel, nous ne l'avons pas monté dans le boîtier du capteur thermo/hygro qui doit être installé dans un espace plus ouvert. Le circuit imprimé principal est adapté au boîtier Spelsberg de classe IP66 (cf. liste des composants).



LISTE DES COMPOSANTS

Carte de connexion 191148-2

Résistances

R1,R2,R3 = 10 kΩ, carbone, 5%, 0,25 W, 250 V

Condensateurs

C1,C2 = 10 μF, 16 V, radial 5 mm (hauteur: 6,5 mm)

Semi-conducteurs

IC1 = LP2950-33LPE3, 3,3V LDO, TO-92

Divers

MOD1 = BoB BME280, e-choppe Elektor

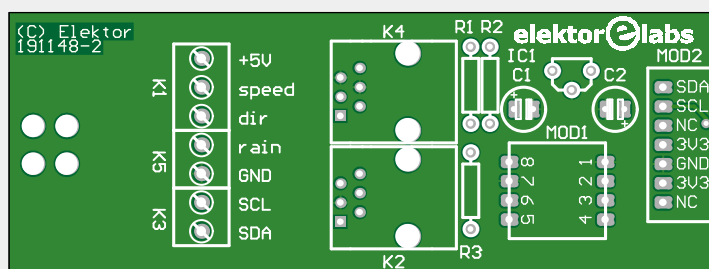
MOD2 = BoB WSEN-PADS (Würth)

K1 = bornier à vis pour PCB, 3 voies, pas 3,81 mm (0,15"), 300 V

K2,K4 = connecteur modulaire RJ-11, 6P6C, vertical, type Molex 95503-2661

K3,K5 = bornier à vis, 2 voies, pour PCB, pas 3,81 mm (0,15"), 300 V

Circuit imprimé 191148-2 V1.1, e-choppe Elektor



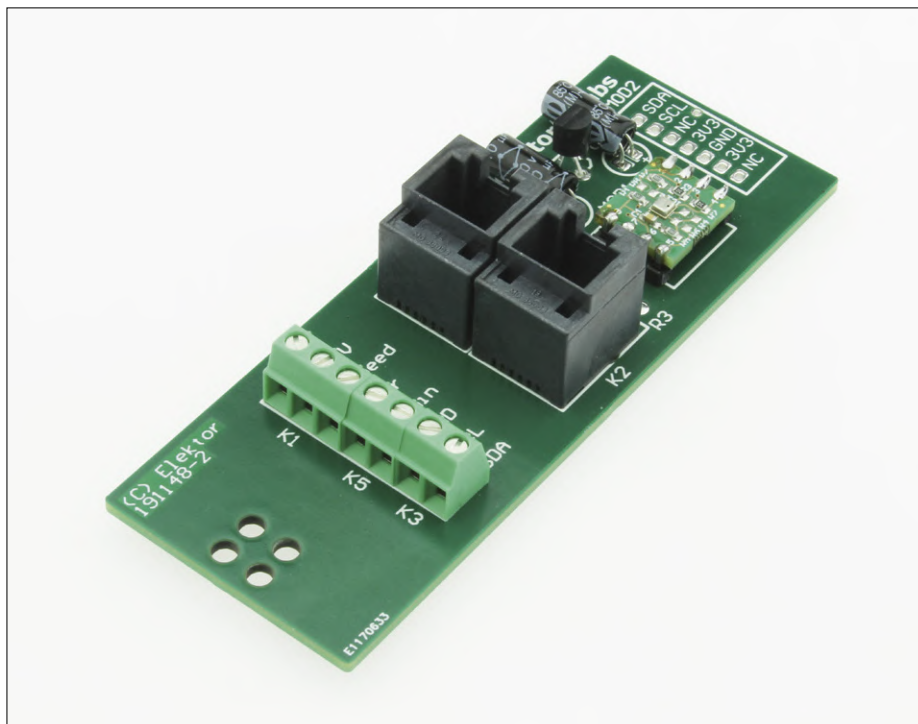


Figure 4. Carte de connexion finie vue de dessus. Sur cette carte se trouve également le module combiné capteur de pression/humidité/température.

C'est le prototype du labo que montre la **fig. 2**. On a fait pivoter de 180° les borniers de 3,81 mm (0,15 pouce), les connexions tournées vers l'intérieur, une solution insolite qui exploite bien l'espace libre dans le boîtier pour le raccordement des fils !

Carte de connexion 191148-2

Cette carte accueille également le capteur de pression/humidité/température, mais sa principale fonction est d'interconnecter le μ C ESP32 et les capteurs (cf. schéma du circuit **fig. 3**). Ce «circuit» remplace la carte d'origine de l'unité thermo/hygro fournie avec le kit WH-SP-WS02, et se glisse parfaitement dans le boîtier.

Le régulateur à faibles pertes (LDO = *low dropout*) IC1 de 3,3 V est le seul composant actif sur cette carte. Hormis les trois résistances de polarisation R1, R2 et R3 pour les interrupteurs des capteurs de pluie et de vent, les autres composants se limitent à des connecteurs pour les modules de capteurs utilisés dans la station.

Les prises RJ-11 se connectent à l'anémomètre (K2) et au pluviomètre (K4). Leurs positions correspondent aux trous dans l'unité thermo/hygro «évacuée» et au texte sur le boîtier.

La position MOD2 est destinée à d'autres capteurs et à l'accès des utilisateurs au bus I²C.

Le populaire adaptateur d'Elektor pour le BME280 est utilisé comme module de capteurs 3 en 1 pour la température, l'humidité relative et la pression atmosphérique. La numérotation des broches sur la carte pour la position MOD1 correspond à celle du BoB BME280. Si vous préférez un capteur T/RH/P compatible I²C d'une autre marque, vous pouvez le connecter à la position MOD2 du circuit imprimé et supprimer le BoB pour le BME280. La **figure 4** montre la carte de connexion telle qu'elle a été fabriquée et testée par Elektor Labs. Les quatre trous permettent de fixer le câble à la carte à l'aide de deux colliers.



@ WWW.ELEKTOR.FR

→ Carte mère, PCB nu
www.elektor.fr/191148-1

→ Carte de connexion, PCB nu
www.elektor.fr/191148-2

→ Carte de capteur de particules, PCB nu
www.elektor.fr/191148-3

→ Kit de la station météo V.2
www.elektor.fr/191148-71

→ Wireless Outdoor Unit for Professional Weather Station WH-SP-WS02
www.elektor.com/professional-outdoor-weather-station-wh-sp-ws02

→ Micro SD-Board BoB
www.elektor.com/1925

→ Afficheur LCD 2×16 caractères avec carte d'interface I²C*
www.elektor.fr/two-wire-lcd-071035-93

* interface I²C : retards de livraison possibles dus à des problèmes d'importation.

Liens

- [1] station météo ESP32, magazine Elektor, janvier 2019 : www.elektormagazine.fr/magazine/elektor-71/42320/
- [2] afficheur LCD 2×16 caractères avec I²C : www.elektor.fr/two-wire-lcd-071035-93
- [3] page de cette publication : www.elektormagazine.fr/191148-01
- [4] page du projet : www.elektormagazine.com/labs/remake-elektor-weather-station

Carte de capteur de particules 191148-3

Le principal objectif de cette carte est de faciliter la connexion et le montage du capteur de particules *Honeywell* HPM115S0-xxx. (fig. 5). Celui-ci est muni d'un connecteur à 8 voies, au pas de 1,25 mm (0,05 pouce), difficile à connecter sans outils professionnels. Le capteur est plus facile à implanter si vous utilisez le connecteur *Würth Elektronik* de la liste des composants. Il suffit de souder les quatre fils sur K3 (deux pour l'alimentation 5 V, et deux pour Rx et Tx) et de brancher soigneusement le connecteur au pas de 1,25 mm dans le capteur. Les connecteurs K1 et K2 assurent ensuite une connexion facile à manipuler entre capteur de particule et carte mère. La fig. 6 montre la carte complète avec le capteur connecté.

Utilisez trois vis à métaux M2x10 mm pour fixer le capteur au circuit imprimé. Il faut percer quelques orifices de ventilation dans le boîtier afin d'obtenir un flux d'air suffisant pour des mesures correctes, tout en empêchant l'humidité dans le capteur installé à l'extérieur.

Produits

Les trois circuits imprimés nus de la station météo en réseau ouvert V.2, ainsi qu'un kit de composants sont disponibles dans la boutique en ligne d'Elektor (cf l'encadré @ www.elektor.fr). Nos lecteurs qui préfèrent commander leurs cartes auprès d'un fabricant de circuits imprimés peuvent le faire en utilisant les fichiers Gerber et les fichiers BOM mis à leur disposition [3].

À suivre

Dans la prochaine édition d'Elektor (juillet-août), nous poursuivrons avec vous le développement du logiciel de la station météo en réseau ouvert V.2. En attendant, nous vous invitons à suivre les progrès de ce projet sur le site Elektor Labs [4].

191148-02

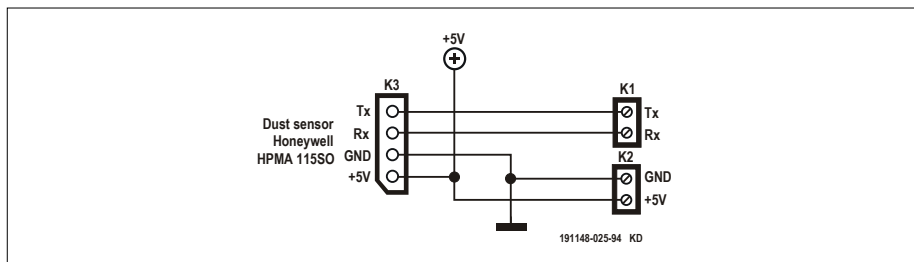


Figure 5. Schéma de la carte du capteur de poussière/particules accueillant le capteur *Honeywell* HPM115S0.

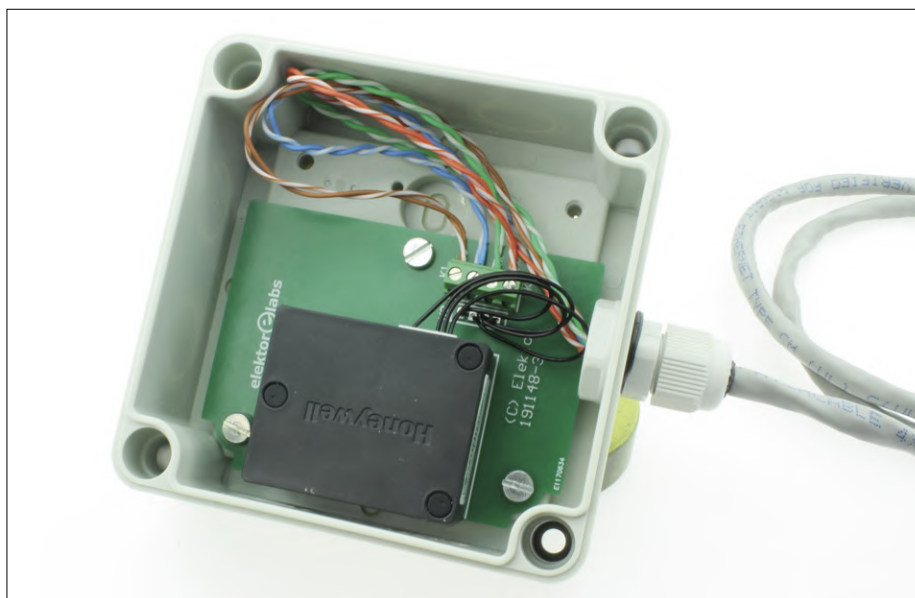


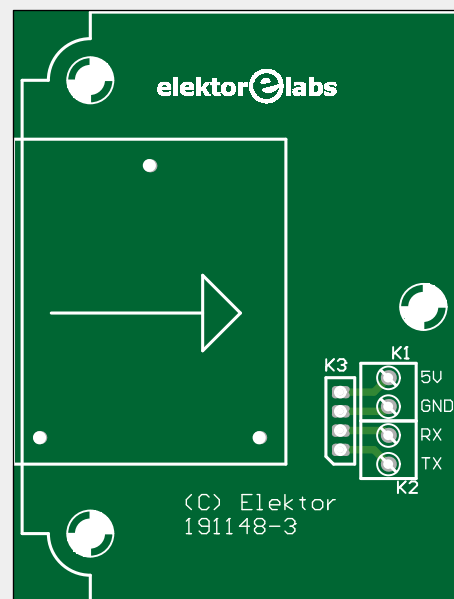
Figure 6. Carte de capteur de poussière/particules prête à l'emploi montée dans le boîtier de classe IP66.



Liste des composants

Carte du capteur de particules 191148-3

MOD1 = capteur de particules HPM115S0-xxx (*Honeywell*), Farnell # 2770767
K1, K2 = bornier à vis, 3 voies, pour PCB, pas 3,81 mm, 300 V
embase WR-PHD, 8 voies, pas 1,25 mm, *Würth Elektronik* # 653008113322
2 câble pré-serti, *Würth-Elektronik* # 653153128130
3 vis à métaux M2x10
coffret = TK PS 94x94x57 mm, classe IP66 (*Spelsberg*)
circuit imprimé 191148-3 V1.0, e-choppe Elektor





intelligence artificielle pour débutants (1)

Reconnaissance d'objets à l'aide de la carte Maixduino

Walter Trojan (Allemagne)

Il n'y a pas de sujet plus brûlant en informatique que l'intelligence artificielle (IA). La carte *Maixduino* est une bonne introduction peu coûteuse, à la reconnaissance de la parole et de l'image ainsi qu'à d'autres aspects de l'IA. Équipée d'une caméra et d'un petit afficheur LCD, elle ne coûte qu'environ 30 €. Elle est programmable dans le célèbre IDE Arduino. Le premier épisode de cette série présente la richesse matérielle du Maixduino et propose quelques programmes de démonstration. L'un d'entre eux reconnaît 1000 objets différents !

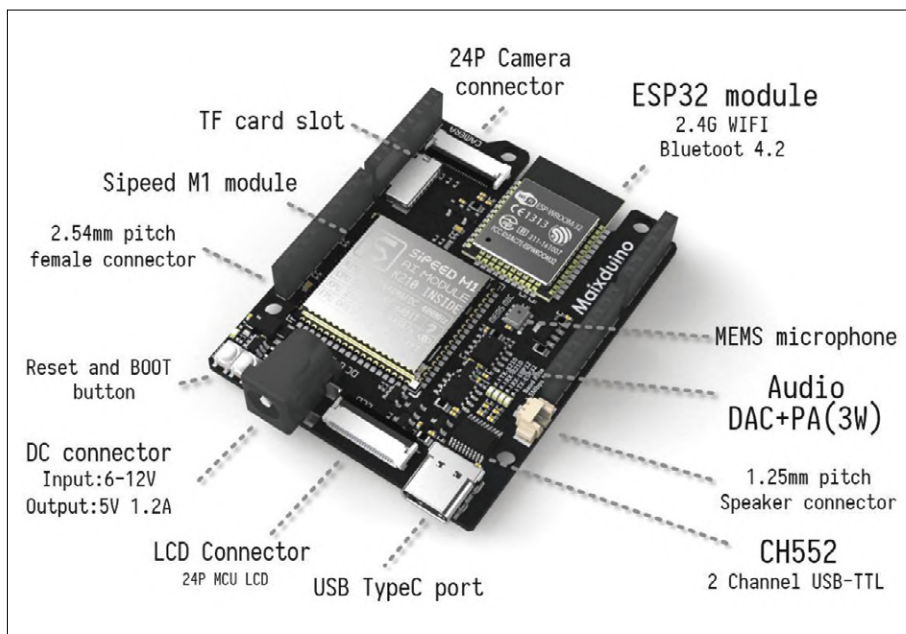


Figure 1. Le matériel disponible sur Maixduino.

L'utilisation de l'intelligence artificielle progresse partout. On trouve de l'IA dans de nombreuses applications. Pour l'année en cours, les ventes mondiales d'applications d'entreprise dans le domaine de l'IA devraient atteindre 4,8 milliards de dollars US, et en 2025, ce chiffre devrait dépasser les 30 milliards de dollars US. L'expertise en IA est très recherchée. Toute personne dotée de connaissances dans ce domaine peut s'attendre à de meilleures perspectives d'emploi. Il est donc utile de prendre le temps de se

familiariser avec le sujet et les nouveaux horizons qu'il ouvre. Ce qui ne gâche rien, c'est qu'en plus c'est très amusant ! On entend par intelligence artificielle, entre autres, la reproduction d'un comportement *intelligent*. L'apprentissage *machine* est un domaine particulier, dans lequel une application n'est plus pensée dans tous ses détails par le développeur ; elle est un cadre de programme d'application universelle, sous la forme d'un réseau neuronal (NN) qui s'enseigne lui-même les fonctions

nécessaires à l'aide de nombreuses données d'exemple. Une catégorie particulière de ce programme est l'apprentissage *approfondi*, dans lequel des structures de programme complexes et optimisées sont utilisées pour améliorer encore les résultats obtenus. Nous reviendrons sur cette terminologie dans la suite de cette série d'articles.

Cependant, l'IA n'est pas un sujet nouveau : ses débuts remontent aux années 1950. La percée récente résulte de l'augmentation vertigineuse de la puissance de calcul disponible. Des cartes graphiques avec des centaines de cœurs de processeur fonctionnant en parallèle. Des puces d'IA hautement spécialisées ont permis de mettre en place et de former de véritables systèmes d'IA. Cela a ouvert la porte à la reconnaissance automatique de la parole, utilisée dans de nombreux assistants personnels, et à la reconnaissance d'images d'objets de toutes sortes. L'IA joue également un rôle important dans la conduite de véhicules autonomes. Dans certains domaines, l'IA a dépassé les capacités humaines : des systèmes d'IA peuvent battre les meilleurs champions aux échecs et au go, et détecter les tumeurs de manière plus fiable que les spécialistes humains. Les systèmes ne sont supérieurs aux humains que dans les domaines particuliers dans lesquels ils ont été formés, et les domaines dans lesquels les systèmes sont formés continueront à être choisis par les humains.

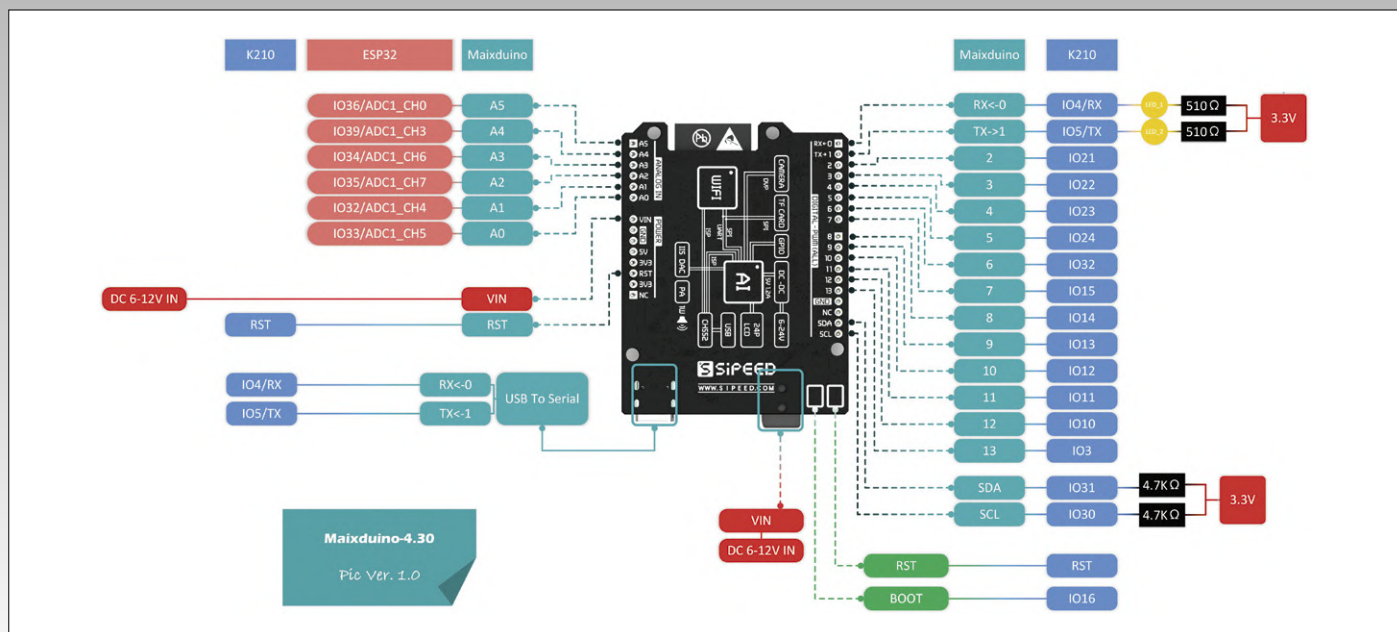


Figure 2. Brochage de Maixduino.

Une initiation qui ne ruinera personne

Maixduino est une manière peu coûteuse de s'initier à l'IA. Le MAix BiT, qui comprend la carte, l'appareil photo et le petit afficheur LCD, est disponible chez Elektor (cf l'encadré en fin d'article). La carte est du format Arduino Uno, mais le matériel est beaucoup plus complet. Elle est fabriquée par Sipeed. Parmi les alternatives au Maixduino, on trouve le Nvidia Jetson Nano, le ROCK PI N10 Model A, le Intel Neural Compute Stick 2 et d'autres, mais ils sont tous beaucoup plus chers. Dans ce 1^{er} épisode, examinons le riche environnement matériel de Maixduino et sa programmation dans l'IDE Arduino. Outre quelques applications typiques de l'Arduino, nous examinerons comment utiliser la caméra et l'afficheur. Nous nous quitterons sur une démonstration de reconnaissance d'objets.

Dans les 2^e et 3^e volets, nous approfondirons le thème de l'intelligence artificielle, nous décrirons la structure des réseaux neuronaux, nous installerons MicroPython et l'IDE qui l'accompagne et nous montrerons comment fonctionne la reconnaissance faciale. Nous examinerons également comment programmer vos propres applications d'IA et comment communiquer avec l'internet.

Super Arduino

Les caractéristiques de Maixduino font saliver ! Sa taille et sa construction sont alignées sur celles de l'Arduino Uno, mais

vous remarquerez immédiatement la densité plus élevée des composants et les nombreuses connexions supplémentaires disponibles. Au cœur de la carte, il y a deux grands modules. Le premier est le processeur Sipeed M1 AI basé sur le dispositif Kendryte K210, dont nous examinerons plus tard le fonctionnement. Le second est un module ESP32 pour communiquer sur WLAN et Bluetooth et pour acquérir des signaux analogiques. L'ESP32 contient deux cœurs de processeur cadencés à 240 MHz et offre donc à lui seul une puissance de traitement considérable, et il peut être utilisé pour décharger les processeurs principaux de leurs fonctions de communication. L'ESP32 a déjà été bien présenté dans Elektor, je n'entrerai donc pas dans les détails.

Les connecteurs correspondent à ceux de l'Arduino original, et l'affectation des broches est largement la même. Attention, les entrées et les sorties sont conçues pour fonctionner sous 3,3 V, voire 1,8 V. **L'application aux circuits d'entrée d'une tension de 5 V serait irrémédiablement destructrice !**

La **figure 1** donne plus de détails à ce sujet. Deux connecteurs à 24 broches sont prévues pour l'interface avec la caméra et l'afficheur LCD. Une carte microSD peut être utilisée pour étendre la capacité de stockage de la carte Maixduino. Le connecteur USB, de type C, est utilisé pour la programmation et le contrôle. Pour le traitement des données

audio, la carte comprend un microphone numérique et d'un convertisseur N/A suivi d'un ampli audio de 3 W. Le matériel riche et varié de la carte vous permettra de vous lancer dans toutes sortes de projets sans recours à des cartes supplémentaires.

Le brochage de la carte (**fig. 2**) ressemble beaucoup à l'original. La tension d'alimentation peut être fournie par l'entrée DC ou en utilisant la broche V_{IN} entre 6 V et 12 V. On peut aussi l'alimenter sous 5 V par le connecteur USB. Six des GPIO de l'ESP32 sont accessibles (32 à 36 et 39), utilisables comme entrées analogiques A0 à A5. Sur le bord opposé de la carte se trouvent les entrées et les sorties du module K210. Celles-ci répondent aux commandes Arduino habituelles, le numéro de broche Arduino étant le numéro du bit d'entrée ou de sortie du K210.

```
#define LED 12 // K210 IO12,
Maixduino broche 10
pinMode(LED, OUTPUT) ; //
configurer le port comme sortie
digitalWrite(LED, HIGH) ; // port
au niveau haut (3,3 V)
```

La broche RST fonctionne sous 1,8 V. Il est hors de question d'y appliquer une tension supérieure. Les sorties IO36 à IO47 sont également conçues pour un fonctionnement sous basse tension : elles ne sont pas sur les connecteurs, mais

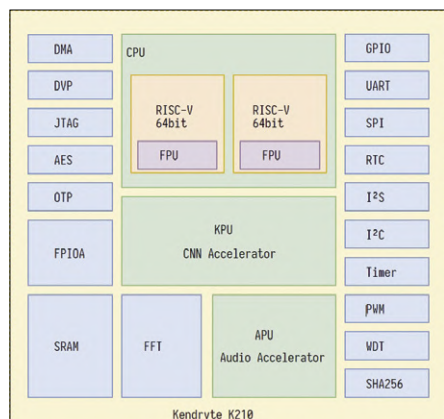


Figure 3. Synoptique des fonctions du Kendryte K210.

utilisées en interne, p. ex. pour piloter l'afficheur LCD. Les ports série RX et TX et l'interface I²C sont équipés des résistances de polarisation appropriées. Malheureusement, la fiche technique du K210 ne donne pas le courant de sortie maximal disponible. Il devrait y avoir de quoi piloter une LED (<10 mA) ; pour des

courants de plus forte intensité, prévoir un étage de puissance.

La grande marmite : Kendryte K210

Au cœur du Maixduino, le SoC K210 (système sur puce) de la société chinoise Kendryte, en technologie 28 nm à faible consommation, disponible depuis septembre 2018. Pour une charge de travail normale, ses deux cœurs de processeur à 64 bits sont cadencés à 400 MHz, une fréquence qui au besoin peut être forcée jusqu'à 800 MHz. Tout ceci répond à la spécification ouverte RISC-V pour éviter au fabricant les frais de licence ARM et contribue ainsi au faible coût total de l'appareil. Les deux processeurs sont équipés d'un FPU (unité de calcul à virgule flottante) qui fonctionne à la fois en simple et en double précision (**fig. 3**).

La particularité du K210 est son unité de traitement des connaissances KPU (avec le k de *knowledge*) pour construire et exécuter des réseaux de neurones.

La puissance de calcul totale disponible est étonnante à ce prix : 0,46 Tops, soit 460 milliards d'opérations par seconde. Avec le surfréquentage de l'horloge, ce nombre déjà astronomique pourra même être doublé, ce qui permet de reconnaître jusqu'à 60 objets par seconde. La performance exceptionnelle est obtenue grâce à 64 unités arithmétiques en parallèle et un bus de 576 bits. Comparée à celle d'autres systèmes d'IA, la dissipation de puissance de seulement 0,3 W est très faible : *Nvidia* recommande d'utiliser une alimentation 5 V 4 A (donc 20 W) pour son *Jetson Nano*, comparable par sa puissance de calcul de 0,4 Tops.

La KPU peut mettre en œuvre des architectures avancées de réseaux neuronaux, y compris des réseaux convolutifs. Ceux-ci ont une structure de filtrage particulièrement efficace dans les applications de traitement d'images : nous y revenons bientôt dans le deuxième volet de cette série. La mémoire principale incluse dans le SoC a une capacité de 8 Mo, divisée en 2 Mo pour les processeurs principaux et 6 Mo pour la KPU. Jusqu'à 5,9 Mo sont donc disponibles pour stocker la configuration du réseau neuronal, ce qui est suffisant pour mettre en œuvre un réseau de taille moyenne.

La puce K210 a d'autres atouts matériels. Notamment une unité de traitement audio (APU), particulièrement utile pour le prétraitement dans les applications de reconnaissance vocale. L'unité peut traiter jusqu'à 8 canaux (ou 4 canaux stéréo) à des fréquences d'échantillonnage d'entrée jusqu'à 192 kHz. Une unité FFT est disponible pour analyser le spectre du signal par transformée de Fourier rapide.

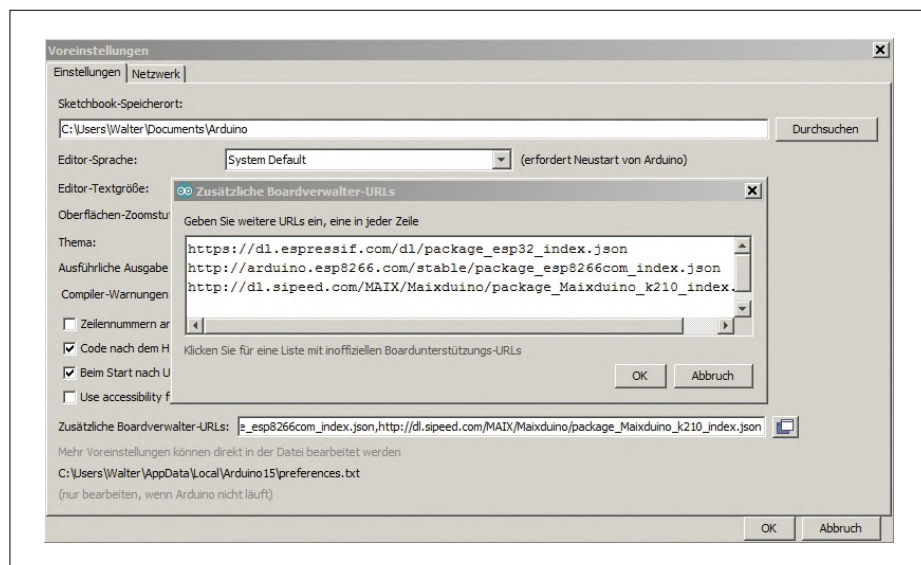


Figure 4. Reconnaissance et configuration de Maixduino.

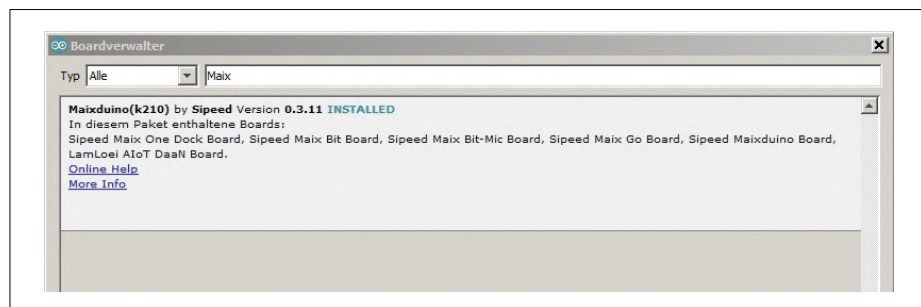


Figure 5. Installation des outils de base de Maixduino.

Notez la présence d'accélérateurs AES et SHA-256 pour les fonctions cryptographiques.

Sans oublier la collection des périphériques habituels : UART, I²C, SPI, I²S, temporisateur, RTC et PWM.

Sipeed a ajouté 16 Mo de mémoire flash sur la carte, ainsi que le circuit d'alimentation, le microphone et l'amplificateur de puissance de 3 W, et l'excellent module ESP32.

Que souhaitez de plus ? Vous trouverez de plus amples informations sur le fonctionnement de la carte, le schéma et d'autres détails en suivant les liens [1], [2] et [3].

Programmation en terrain connu

Ce n'est pas seulement par sa forme que notre carte est similaire à l'Arduino original : elle utilise l'environnement IDE Arduino, dans lequel le noyau Maixduino est intégré de la même manière que l'ESP8266 ou l'ESP32. Sous *Fichier* -> *Préférences*, il est nécessaire d'ajouter une nouvelle URL pour le gestionnaire de cartes : si vous faites un clic droit sur le bouton à droite de la zone de saisie de texte, une petite fenêtre s'ouvrira pour faciliter la saisie de l'URL (**fig. 4**). Si vous souhaitez programmer uniquement le Maixduino, il suffit d'ajouter la ligne «sipeed» ; la programmation de l'ESP32 embarqué passera par là aussi.

Pour installer les cœurs Maixduino, sélectionnez l'option de menu *Outils* -> *Gestionnaire de cartes*. Saisissez le terme de recherche «Maix» (**fig. 5**) et procédez à l'installation.

Ça y est, nous pouvons déjà commencer à programmer ! Plutôt que le programme classique *hello world*, nous allons passer directement à l'essai de la caméra et de l'afficheur. Connectez le Maixduino par USB et dans le menu *Outils*, réglez les paramètres suivants.

Board : *Sipeed Maixduino*
CPU Clock Frequency : 400 MHz
Burn Tool Firmware : *open-ec*
Burn Baud Rate : 1.5 Mbps
Tool Install Location : *Standard*
Port : *<port COM utilisé>*
Programmer : *k-flash*

Un programme de démonstration permettant de capturer une image et de l'afficher sur l'afficheur LCD est déjà disponible dans l'IDE. Appelez-le : *Fichier* -> *Exemples* -> *Sipeed_OV2640* -> *selfie* et une fois qu'il est téléchargé sur la carte (vous devrez peut-être appuyer sur le bouton de réinitialisation à ce stade), vous pouvez l'exécuter. Le code est indiqué dans la **figure 6**.

Le programme commence par intégrer les fonctions des bibliothèques Sipeed pour le pilotage de la caméra et du LCD, connectées à la carte à l'aide d'un bus SPI. Le format d'image choisi est la résolution QVGA (320 x 240 pixels) avec une couleur RGB565. La routine de configuration initialise les deux appareils, et dans la boucle principale, les images capturées sont transférées directement au LCD : on ne pourrait pas faire plus simple.

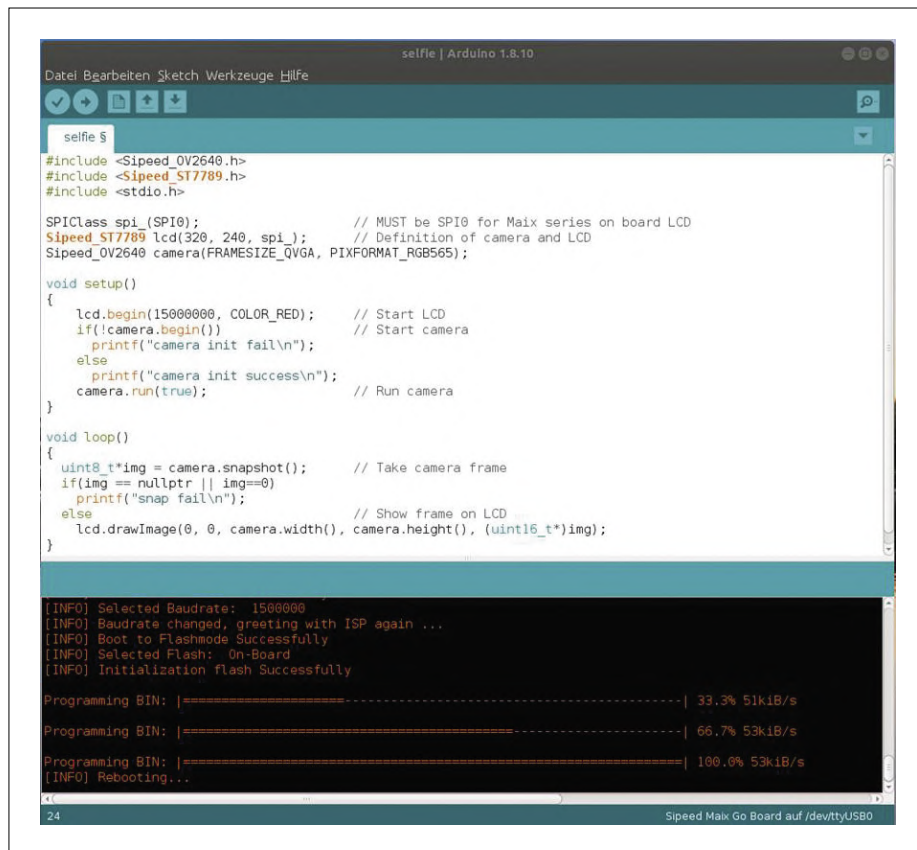


Figure 6. Arduino IDE avec le programme «selfie».

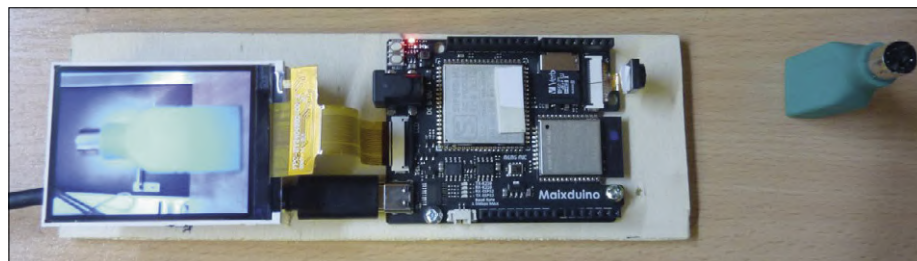


Figure 7. Image (à gauche) capturée d'une fiche d'adaptation (à droite).

L'image n'est pas particulièrement contrastée, mais elle est nette et le rafraîchissement fluide (**fig. 7**).

Maixduino est donc aussi facile à utiliser qu'un Arduino Uno, mais il offre beaucoup plus de possibilités. Nous pouvons commencer à construire des applications plus exigeantes.

Premier modèle d'IA

L'apprentissage approfondi de l'intelligence artificielle sera au menu du prochain article de cette série, mais nous pouvons commencer par une application simple, avec *MobileNet*. Rien à voir avec les téléphones portables, mais plutôt un classificateur d'images qui reconnaît et

identifie 1000 types d'objets quotidiens. Il utilise un réseau neuronal, qui est une structure logicielle construite à partir de nœuds organisés en couches formées par un long processus au cours duquel on lui présente des milliers d'images. Comme le fichier contenant le jeu d'images d'apprentissage occupe environ 200 Go et que le processus de formation lui-même est très long, il n'est pas pratique de le tenter sur le Maixduino. Cependant, il est possible d'y installer un modèle prêt à l'emploi pour le mettre immédiatement au travail de reconnaissance d'objets. C'est ce que je propose maintenant. Le logiciel nécessaire [4] se compose des éléments suivants :

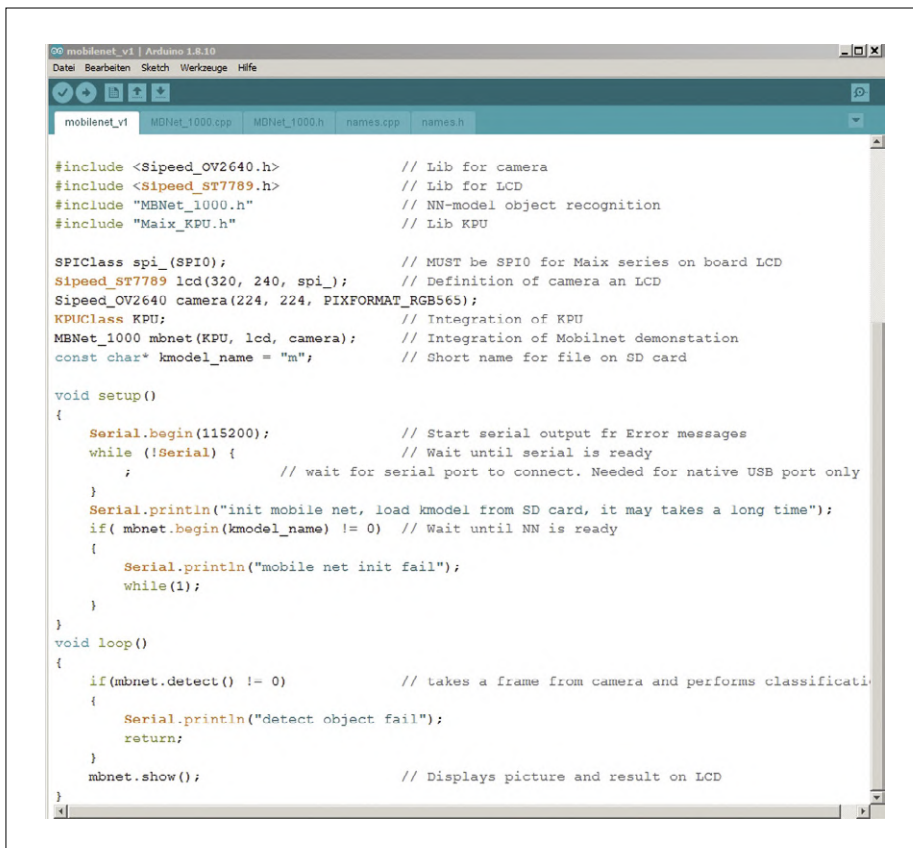


Figure 8. Programme de démonstration MobileNet.

- **mobilenet_v1.ino** : programme C++ principal pour la démonstration
- **MBNet_1000.h** : fichier d'en-tête pour les routines de démonstration
- **MBNet_1000.cpp** : routines C++ pour l'acquisition, la reconnaissance et l'affichage d'objets
- **names.h** : fichier d'en-tête avec description des objets
- **names.cpp** : routine C++ pour décrire les objets reconnus

Téléchargez ces fichiers et placez-les tous dans le même répertoire. Le modèle d'IA préformé mobilenet_0x300000.kfpgk peut être téléchargé [5] : compressé, il ne fait que quelques mégaoctets. Après l'avoir décompressé avec 7zip, vous devriez trouver un dossier mobilenet_0x300000 dans lequel il y a deux fichiers : le modèle est celui appelé 'm'. Copiez ce fichier sur une carte microSD dans le répertoire racine et insérez la carte sur le Maixduino.

Un coup d'œil au code du programme (**fig. 8**) montre qu'il est possible de programmer facilement des applications même très complexes, grâce aux puissantes bibliothèques : dans le domaine de l'IA, il existe de nombreuses bibliothèques très efficaces. La première partie du programme déclare la caméra, le LCD et la KPU ainsi que leurs paramètres. La résolution de la caméra est configurée au même format que les images d'entraînement, soit 224 par 224 pixels. Enfin, l'objet **mbnet** rassemble les ressources de la KPU, de la caméra et du LCD.

La routine de configuration initialise le code de démonstration, puis dans la boucle principale, nous avons la classification des objets dans les images acquises par la caméra et leur affichage sur le LCD. Imaginez-vous qu'un programme de reconnaissance d'objets puisse être aussi simple ?

Comme objet de test, j'ai présenté au Maixduino une photo de chat (**fig. 9**), immédiatement identifié comme chat tigré ou égyptien (**fig. 10**). L'application est capable de traiter environ cinq images par seconde et peut donc produire des résultats utiles même si la caméra est en mouvement, surtout sous un bon éclairage et devant un arrière-plan pas trop chargé. Comme le montre la figure 10, la palette de couleurs est limitée (bleu-blanc) pour augmenter le taux de reconnaissance, selon une méthode courante. Au lieu d'utiliser les trois canaux de couleur, on n'en utilise qu'un seul pour réduire la quantité de données et donc la puissance nécessaire au traitement des pixels. La technique consistant à utiliser un réseau déjà formé, c'est-à-dire à effectuer la formation et la classification sur des plateformes différentes, est courante aussi dans ce type d'applica-



Figure 9. Acquisition de l'image d'un « objet ».

tion. De nombreux utilisateurs profitent de la grande puissance de traitement disponible sur AWS, Microsoft Azure ou Google Cloud pour former leurs réseaux de neurones, pour les faire fonctionner – ce qui nécessite beaucoup moins de ressources que l'apprentissage – sur des plateformes à plus petite échelle. L'industrie des semi-conducteurs répond déjà à cette demande : Intel a récemment annoncé ses processeurs pour réseaux de neurones Nervana NNP-T et NNP-I, dont la version «T» offre une puissance de traitement plus élevée pour la formation, et la version «I» est un dispositif moins puissant, destiné aux applications d'inférence et de classification.

Vous allez voir ce qu'on va voir...

L'application de démonstration examinée dans cet article effleurer la surface du sujet de l'intelligence artificielle en matière de reconnaissance d'objets, mais j'espère avoir suscité votre intérêt. Dans le prochain épisode, nous nous pencherons sur la structure et le fonctionnement d'un réseau de neurones. Avec un nouvel environnement de développement, nous goûterons aux tentations de Linux et du langage Python, car c'est là que l'on trouve les bibliothèques et les

infrastructures les plus prometteuses. Dans une telle infrastructure, la création d'un réseau neuronal est aussi simple que l'assemblage de briques Lego. Nous essayerons une application de reconnais-

sance des visages et vous développerez vos propres structures de réseau neuronal. Aux impatients, je recommande le livre *Make Your Own Neural Network* de Tariq Rashid. Si vous doutez encore de

leur intérêt, sachez que la traduction de l'article que vous venez de lire a largement bénéficié d'outils de traduction automatique basés sur des réseaux neuronaux. ◀

200023-03



Figure 10. L'objet est identifié et classé.

Liens

- [1] Description de Maixduino : <https://wiki.sipeed.com/en/maix/board/maixduino.html>
- [2] Forum Sipeed : <https://bbs.sipeed.com/>
- [3] Schéma de la carte Maixduino : <http://dl.sipeed.com/MAIX/HDK/Maixduino/Maixduino-4.30/>
- [4] Démonstration MobileNet : https://github.com/sipeed/Maixduino/tree/master/libraries/Maix_KPU/examples/mobilenet_v1
- [5] Fichier de données modèle MobileNet : http://dl.sipeed.com/MAIX/MaixPy/model/mobilenet_0x300000.kfpgk

Publicité



**Assemblage en ligne
de carte électronique**

www.emsproto.com



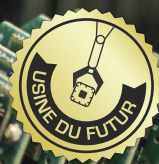
CHIFFREZ
VOTRE CARTE
ÉLECTRONIQUE
EN LIGNE



DÉLAIS
2 à 12
JOURS



QUANTITÉ
1 à 50
CARTES

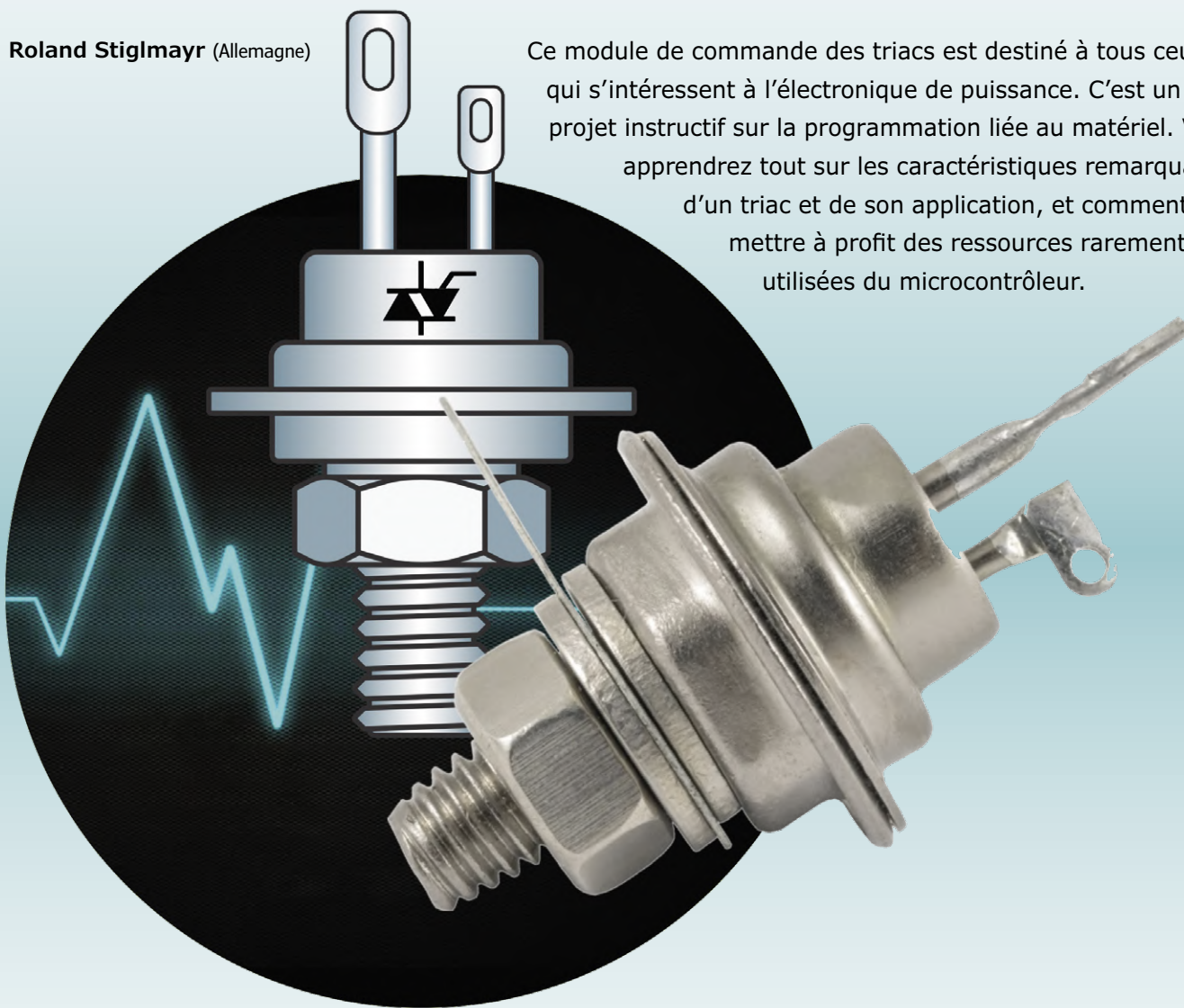


commande universelle de triacs avec ATmega

Commutation et gradation de charges variées

Roland Stiglmayr (Allemagne)

Ce module de commande des triacs est destiné à tous ceux qui s'intéressent à l'électronique de puissance. C'est un projet instructif sur la programmation liée au matériel. Vous apprendrez tout sur les caractéristiques remarquables d'un triac et de son application, et comment mettre à profit des ressources rarement utilisées du microcontrôleur.



Tout électronicien a une idée de ce que sont un thyristor et un triac, comment ils fonctionnent et où les utiliser. En fait, leurs possibilités sont si variées qu'elles ne sont jamais pleinement exploitées que par des spécialistes. Cet article se cantonne aux triacs, car les thyristors ne sont plus utilisés désormais que dans des cas particuliers.

Nous commencerons par la fonction de base d'un triac et ses paramètres pertinents, puis passerons à l'électronique du module de commande. La **figure 1** montre le prototype fini. Le couple ATmega - triac permet de réaliser de nombreuses

fonctions : selon le microprogramme, le module peut être utilisé comme simple interrupteur (à distance), minuterie, gradateur ou module de démarrage progressif pour charges inductives (transformateurs ou moteurs).

L'essentiel

Un triac est un interrupteur à semi-conducteur bidirectionnel. C'est donc bien un interrupteur pour courant alternatif. Le courant (de charge) circule par les deux électrodes T1 et T2 (c'est le T de *terminal*). Ces électrodes sont parfois désignées

MT1/MT2 (de *main terminal*) ou anciennement A1 et A2 (anode). La gâchette G est l'électrode de commande. La **fig. 2** donne le symbole du triac tel qu'il est dessiné dans les schémas d'Elektor (à gauche) et un circuit équivalent simple (à droite).

Le triac ne conduit pas en l'absence d'un courant de gâchette insuffisant entre G et T1. Une fois le triac amorcé, il conduit tant que l'intensité du courant entre T1 et T2 ne passe pas sous un certain seuil qui dépend du type de triac. En dessous de ce seuil de courant, le triac ne conduit plus. Sous une tension alternative, s'il n'est pas commandé en permanence par sa gâchette, la conduction du triac est donc coupée à chaque passage par zéro du courant (lequel courant n'est pas nécessairement en phase avec la tension). Le triac peut donc être amorcé par une courte impulsion de gâchette de faible intensité, mais on ne peut pas en forcer le blocage instantanément.

T1 est le point ou potentiel de référence pour la gâchette G. Selon la polarité du potentiel de la gâchette et de T2 par rapport à T1, il y a quatre flux de courant possibles, illustrés par le modèle à quatre quadrants (**fig. 3**).

L'axe des X représente la polarité de la tension de gâchette par rapport à T1. L'axe des Y représente la polarité de la tension sur T2 par rapport à T1. Les quatre quadrants qui en résultent sont marqués de chiffres romains. Les valeurs positives sont indiquées en rouge et les valeurs négatives en bleu. Dans sa fiche technique, les paramètres d'un triac sont indiqués pour chacun des quatre quadrants, car ils sont souvent différents.

Paramètres

Pour amorcer ou ouvrir le triac, la valeur de seuil de l' I_{GT} (courant d'amorçage de la gâchette) doit être dépassée. Après la fin de l'impulsion de commande, le triac ne reste conducteur que si le courant de charge est supérieur à I_L (*Latching Current* = c. de verrouillage). S'il est plus faible, le triac se verrouille après la fin de l'impulsion d'amorçage. Dans les fiches techniques, la spécification des courants d'amorçage et des autres courants est donnée séparément pour chaque quadrant. Si le courant de charge à travers le triac une fois amorcé tombe en dessous du courant de maintien I_{HT} , le triac se bloque.

Avec les charges inductives, il y a entre courant et tension un déphasage qui dépend de la composante inductive. Par conséquent, la tension aux bornes de la charge lors du passage par zéro du courant n'est pas nécessairement nulle. C'est pourquoi, lors du blocage du triac, la tension sur T2 peut sauter brusquement à des valeurs élevées, ce qui pourrait causer des problèmes : si le saut de tension est supérieur à la valeur limite dv_{COM}/dt , le triac redevient conducteur. Selon la charge, cela peut avoir un effet destructeur. On amortit cette montée en tension trop rapide au moyen d'un réseau RC (dit *snubber*) en parallèle sur T2 et T1. Les triacs dits Hi-Com sont beaucoup moins sensibles à cet égard.

Il faut considérer aussi le paramètre dv_D/dt . Si cette valeur limite est dépassée sur le triac bloqué, celui-ci devient conducteur. Si le courant de suivi dure une demi-onde, un transformateur mis sous tension de cette manière peut entrer en saturation et présenter brusquement une faible impédance. Le triac et éventuellement les pistes du circuit imprimé ne survivent généralement pas à ce genre de traitements. Un amortisseur ou un triac Hi-Com sont de bons remèdes dans ce cas.

Le paramètre dI_T/dt , ou valeur limite du taux maximal d'augmentation du courant après le déclenchement du triac, est également essentiel. Un dépassement entraîne la destruction

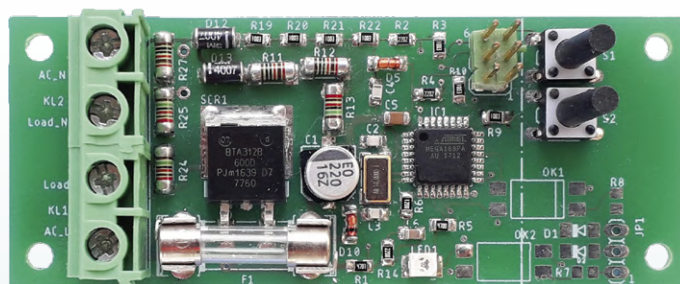


Figure 1. Prototype du module de commande de triac réalisé par l'auteur.

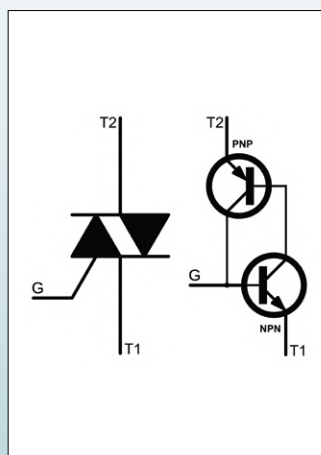


Figure 2. À gauche, symbole du triac et à droite circuit équivalent composé d'un transistor PNP et NPN pour le premier quadrant.

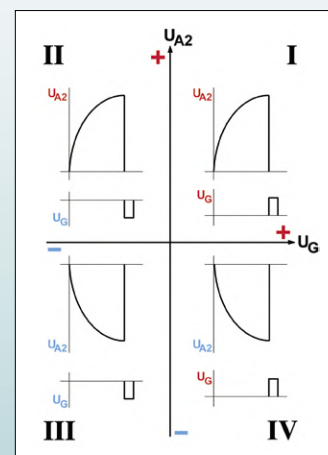


Figure 3. Les quatre quadrants résultant de la polarité des tensions sur la gâchette ou sur T2 par rapport à T1.

du triac avec un court-circuit entre T2 et T1. La raison en est la densité de courant non homogène autour de l'électrode de grille, qui entraîne une surchauffe locale. Comme la valeur de dI_T/dt est faible, en particulier dans le 4e quadrant, ce mode de fonctionnement doit être évité. Le problème du dI_T/dt reste mineur lors de la commutation au passage par zéro de la tension – mais seulement pour les charges à prédominance résistive. Si nécessaire, le taux d'augmentation du courant peut également être limité en connectant une petite self à air en amont. Le paramètre dI_{COM}/dt définit la valeur limite du taux de montée du courant au passage par zéro actuel. Si cette valeur est dépassée, le triac ne se bloque pas mais reste conducteur. Le triac est mal amorcé si la tension inverse maximale périodique V_{DRM} (*Repetitive Peak Off-state Voltage*) est dépassée. On remédie à cela avec en parallèle sur T2/T1, une varistance qui limite la tension maximale.

Le dernier paramètre à expliquer est l'intégrale de la charge limite I^2t ou l'intégrale du courant au carré sur 10 ms. Cette valeur limite correspond à l'absorption maximale d'énergie du triac. Elle doit être prise en compte pour dimensionner la sécurisation.

Les autres grandeurs de la fiche technique d'un triac devraient en grande partie s'expliquer d'elles-mêmes. Si vous souhaitez concevoir vous-même des circuits à triacs, vous devriez maintenant pouvoir sélectionner le triac idéal pour votre application en fonction des paramètres.

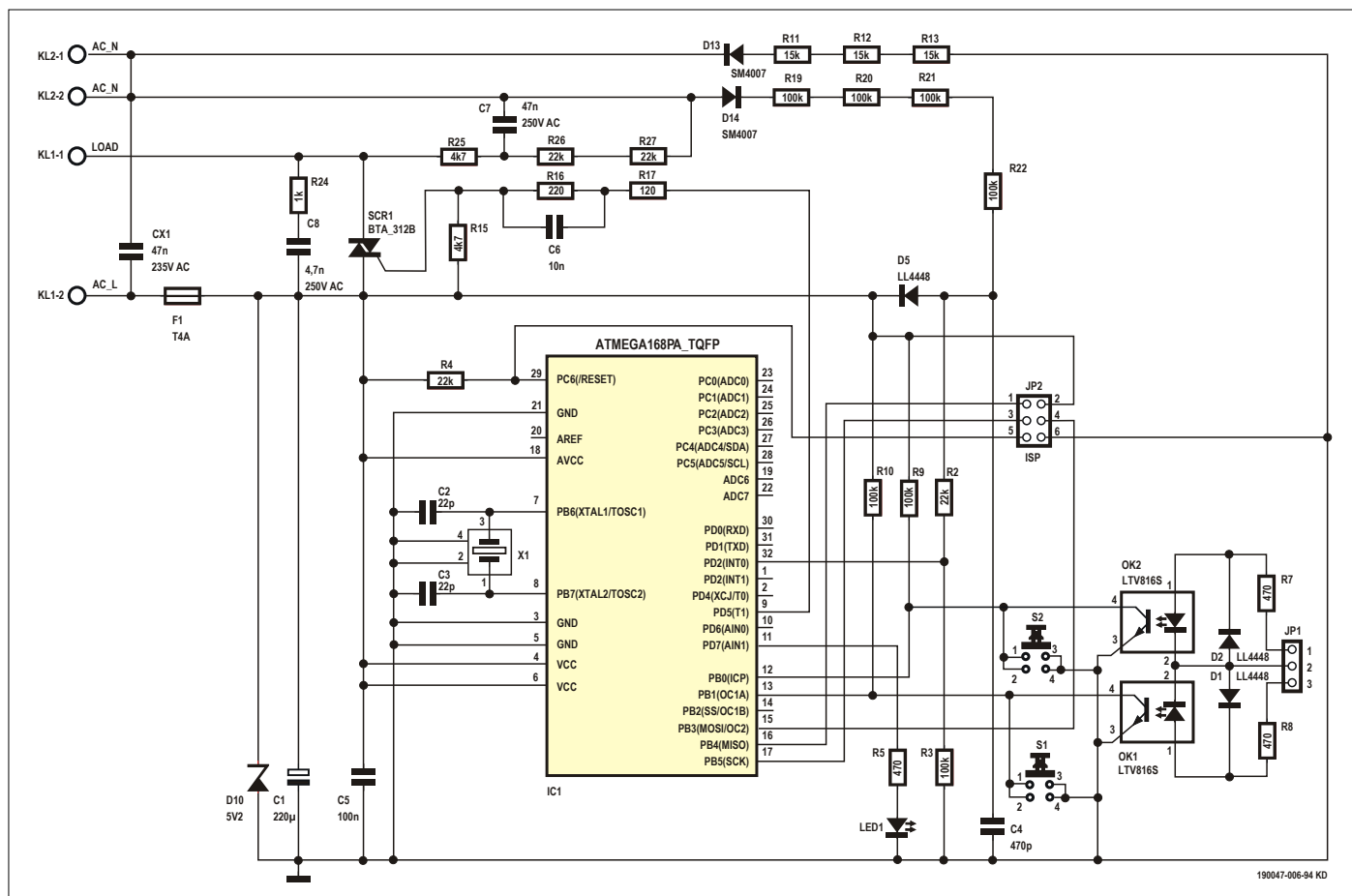


Figure 4. Le schéma du module de commande de triac. La fréquence du quartz est de 11,0592 MHz.

Module de commande du triac

Passons à la pratique. Grâce à son intelligence, notre module de commande universel pour triacs pourra remplir de nombreuses fonctions : simple interrupteur, temporisateur avec arrêt temporisé, gradateur ou commande de charges inductives : il sait tout faire ! Ici microcontrôleur et intelligence vont de pair. Le choix s'est porté sur un ATmega.

Pour choisir le bon triac, les principaux critères sont le courant à commuter et la tension maximale appliquée au triac. Pour un fonctionnement sur réseau monophasé, on peut se contenter de $230\text{ V} \times \sqrt{2} + \text{réserve} \approx 600\text{ V}$. Pour un usage industriel, il est préférable de choisir des triacs à tension inverse de 800 V en raison des transitoires élevés qui se produisent. Le courant réel possible dépend non seulement de la valeur nominale du triac, mais aussi de la puissance dissipée et donc du refroidissement. Cette puissance est le produit de la tension directe par le courant commuté.

Il faut également tenir compte du courant de charge minimal. Les seuils I_H et I_L doivent être dépassés franchement, sinon la charge ne sera pas correctement mise en marche ou le flux de courant ne sera pas continu. Les types BT134-600E (4 A) et BTA312B (12 A) de 600 V sont universels et conviennent à de nombreux usages simples.

Alimentation

L'alimentation d'un tel module n'a rien d'anodin. Une petite alimentation à découpage serait bienvenue, mais elle serait trop encombrante. Une résistance capacitive en série serait

plus simple, mais les condensateurs MKT avec une capacité et une rigidité diélectrique convenables ne sont pas bon marché. Une résistance ohmique en série serait encore plus simple. Le triac doit être exploité dans les quadrants II et III. Il faut donc une tension de déclenchement de grille négative par rapport à l'électrode T1 directement connectée au secteur. Le moyen le plus simple de l'obtenir est le redressement simple alternance. La praticabilité d'un tel circuit dépend toutefois directement de sa consommation de courant.

Il faut donc compter le courant du microcontrôleur lui-même et l'énergie d'amorçage du triac. Le μC a très peu à faire et peut donc roupiller la plupart du temps, de sorte qu'il ne consomme pas grand-chose. Les impulsions d'amorçage sont courtes, l'énergie moyenne reste faible. Jusqu'à quel point est-il possible de réduire la consommation. Le μC est en veille prolongée la plupart du temps et ne se réveille que pour produire l'impulsion d'amorçage.

La situation la plus défavorable est celle du déclenchement intempestif par un parasite RF sans détection de passage par zéro, avec un signal de déclenchement produit toutes les $278\text{ }\mu\text{s}$. Sous une tension $V_{CC} = 4,7\text{ V}$, un ATmega a besoin de $1,9\text{ mA}$. Avec un courant de gâchette de crête de 11 mA pendant $10\text{ }\mu\text{s}$ chacun à une période de $278\text{ }\mu\text{s} = 3,6\text{ kHz}$, le courant de gâchette moyen est de $400\text{ }\mu\text{A}$. Un total d'environ $2,3\text{ mA}$ circule alors. Cela détermine la valeur de la résistance en série et sa dissipation de puissance. Pour le redressement simple alternance, la valeur effective est $I_{pk} / 2$ et la valeur moyenne est I_{pk} / π . Il en découle :

$$I_{pk} = 2,3 \text{ mA} \times n = 7,3 \text{ mA},$$

$$R = 325 \text{ V} / 7,3 \text{ mA} = 45 \text{ k}\Omega,$$

$$I_{eff} = I_{pk} / 2 = 3,7 \text{ mA}$$

et enfin une dissipation de puissance à la résistance R de :

$$P_{vv} = I_{eff}^2 \times R = (3,7 \text{ mA})^2 \times 45 \text{ k}\Omega = 0,62 \text{ W}.$$

Si R est réparti en trois résistances individuelles de 1 W, ni la dissipation ni la tension ne posent pas de problème.

Un problème supplémentaire se pose lors de la mise en marche. Tant que le μC est en cours d'initialisation, il lui faut beaucoup de courant, et la tension V_{cc} n'augmente que très lentement. Ce problème peut être résolu en réglant le niveau de DBO à 2,7 V et en activant immédiatement le mode veille pendant 250 ms. Nous y reviendrons.

Sécurité

Les dangers d'un module relié directement au réseau électrique sont à prendre au sérieux. Quand le circuit est sur le secteur, aucun instrument de mesure et certainement aucun PC ne doit jamais y être connecté. De telles manipulations ne doivent jamais être faites que par un électricien habilité. La présence d'un transfo de séparation galvanique est obligatoire. Il faut aussi vous prémunir contre tout contact accidentel lorsque le module est branché sur le secteur. Il sera donc forcément installé dans un boîtier parfaitement isolant !

Avant tout test ou modification, le module doit être débranché du secteur et alimenté sous 5 V par un bloc d'alimentation. Un signal de 50 Hz d'une amplitude de 4 V issu d'un générateur de signaux carrés sera nécessaire sur C4 pour la détection du passage par zéro.

Matériel

L'alimentation du μC (**fig. 4**) et de ses périphériques passe par les résistances de limitation R11 à R13 et le redressement monoalternance par D13 ainsi que le lissage par C1. Les résistances MELF (*metal electrode leadless face*) R11 à R13 offrent une rigidité diélectrique et une capacité de charge élevées. Du fait de l'impédance élevée de l'alimentation, celle-ci se comporte en source de courant, et la zener D10 fait bien l'affaire comme source de tension constante.

Le quartz X1 cadence le μC avec précision ce qui assure une chronologie rigoureuse des impulsions de déclenchement. D12 associée au diviseur de tension de R2, R3 et R19 à R22 délivre un signal adapté à la détection du passage par zéro, et « raccroché » au potentiel V_{cc} par D5.

R16 et R17 limitent le courant de gâchette. C6 en parallèle sur R16 assure un déclenchement rapide du triac. En parallèle sur T2 et T1 du triac, on trouve l'amortisseur fait de R24 et C8, qui empêche les amorçages intempestifs. Le triac utilisé est de type Hi-Com, ce qui permet d'associer un condensateur de faible capacité à une résistance assez forte.

Une particularité de ce circuit est la charge auxiliaire optionnelle composée de R25 à R27 et C7. Elle n'est nécessaire que si la charge proprement dite présente une inductance élevée, comme c'est le cas d'un transfo. La plupart du temps, ces composants ne sont pas implantés. Le condensateur X CX1 supprime les interférences.

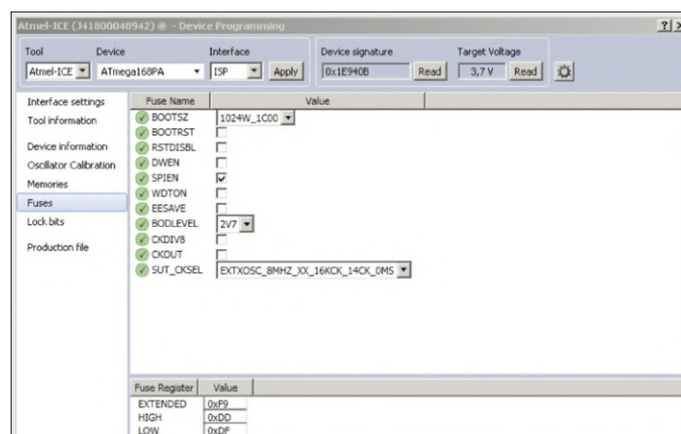


Figure 5. Réglage des fusibles lors de la première programmation de l'ATmega.

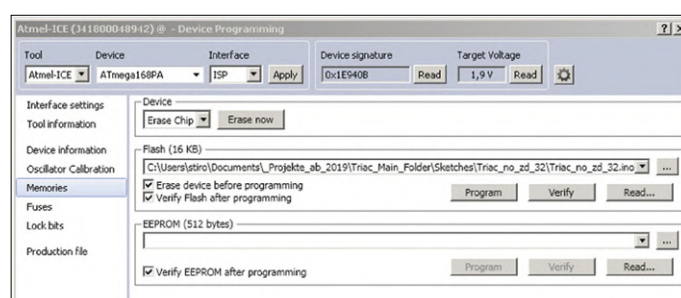


Figure 6. Paramètres de la programmation par ISP.

Le module est commandé par les boutons S1 et S2. Il est possible d'utiliser les optocoupleurs OK connectés en parallèle. Pour respecter les distances de sécurité, utilisez des boutons poussoirs d'un modèle approprié à ces conditions. La LED1 sert de témoin de fonctionnement. IC1 est programmé via l'interface ISP (JP2). F1 protège le module en cas de court-circuit.

Logiciel

Le logiciel [1] a été créé avec l'IDE Arduino. En raison de la lenteur des fonctions d'Arduino, l'auteur ne les a pas utilisées. Le programme est très rapide et fonctionne de manière déterministe. C'est pourquoi les sketches n'ont délibérément pas été portés dans Atmel Studio.

Malheureusement, la programmation du μC n'est pas possible via le chargeur de démarrage (*bootloader*), car à la mise en marche la consommation de courant serait si forte que la tension d'alimentation nominale ne serait pas atteinte. Les fusibles du μC doivent donc être réglés de façon à inactiver le chargeur de démarrage. La programmation se fait directement depuis Atmel Studio par un programmeur ISP approprié. Dans le répertoire du projet, vous trouverez le fichier HEX pour chaque application en plus du croquis correspondant. Lors de la première programmation, il faut que les fusibles soient programmés comme indiqué (**fig 5**). Puis on charge le fichier HEX souhaité à partir du répertoire du projet et on le programme dans le μC (**fig. 6**). Après avoir modifié un croquis, il faut le recompiler (Vérifier/Compiler). L'indicateur d'état (*status*) de l'IDE indique le répertoire du fichier HEX

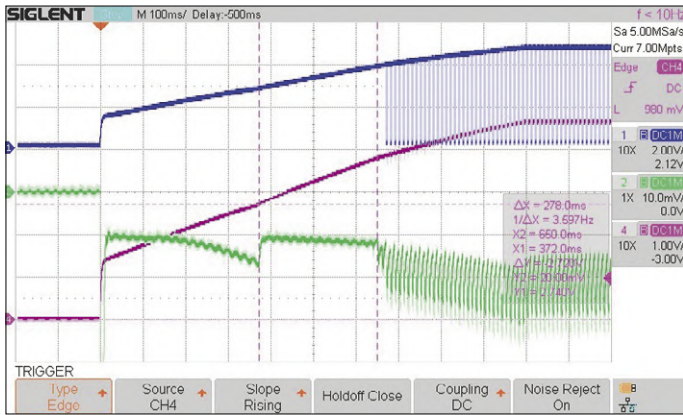


Figure 7. Conditions à la mise sous tension. CH1 (bleu) : impulsions de gâchette; CH2 (vert) : courant; CH4 (rouge) : V_{CC} . X1, Y1 : début de la minuterie WDT et mode de veille, niveau de DBO 2,7 V; X2 : fin de minuterie WDT, réveil.

produit. Un chemin de fichier typique serait :

C:\Users\name\AppData\Local\Temp\buildxyz.tmp\

Triac...ino.hex

Comme le répertoire et le numéro de *build* sont réaffectés quotidiennement, le fichier doit également être mis à jour quotidiennement dans Atmel Studio. Si le répertoire *AppData* n'est pas visible dans l'explorateur/gestionnaire de fichiers, la fonction «Afficher tous les fichiers et dossiers» doit être définie dans l'explorateur sous *Outils/Options de dossier/Vue*.

Dans Atmel Studio, vous chargez le fichier HEX souhaité et vous programmez l'ATmega.

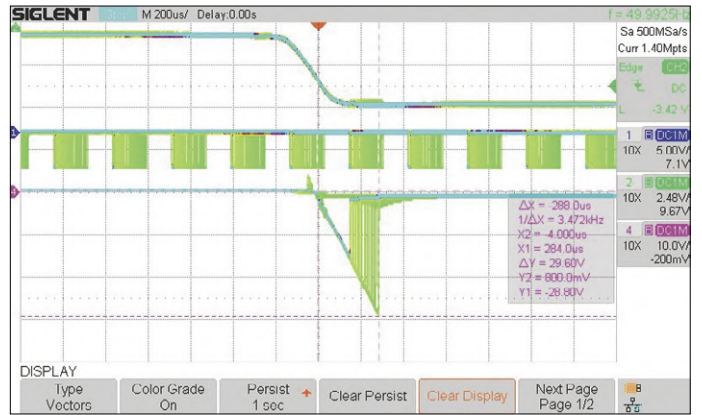


Figure 8. Tension aux bornes du triac avec des impulsions de déclenchement non corrélées pendant 1 s. Courbe supérieure (CH2) : passage par zéro du réseau; courbe médiane (CH1) : impulsions de gâchette; courbe inférieure (CH4) : tension aux bornes du triac.

Attention : lors de la programmation, le module doit être complètement déconnecté du secteur et alimenté en toute sécurité électrique par une source basse tension.

Applications

La procédure de démarrage est la même pour toutes les apps. À la mise sous tension, c'est le niveau BOD de 2,7 V qui détermine la fin de la réinitialisation. Après le démarrage du programme, la minuterie (*WatchDogTimer*) est activée et le programme est mis en mode veille. Cela permet à V_{CC} de s'établir rapidement et en toute sécurité (fig. 7). L'expiration de la minuterie déclenche l'interruption WDT qui met fin au mode veille. Ensuite commence l'exécution du programme proprement dit.

L'extrait du sketch (listage 1) montre comment configurer le minuteur et le mode veille. Le listage contient une partie du code de la procédure de démarrage. Tout d'abord, le chien de garde est réglé et activé. Ensuite, le μC est mis en mode de veille. L'expiration du WDT déclenche l'interruption du WDT, qui réveille à nouveau le μC .

Un croquis distinct est disponible pour chaque exemple de demande. Chaque application a des fonctions différentes que l'on peut sélectionner par des boutons ou des cavaliers. De plus amples informations figurent dans les commentaires de ces sketches.

Interrupteur et minuterie

L'application *Sketch Triac_no_zd_32* ne nécessite pas de détection du passage par zéro, elle montre la programmation de base du module et ses particularités. Le circuit fonctionne sur une alimentation électrique de 5 V pour la programmation et les tests. Si vous vous intéressez à la programmation matérielle de la minuterie WD, aux différents modes de veille, aux interruptions, aux fonctions de la minuterie ou aux commandes d'E/S, plongez dans le code qui, comme tous les exemples, est téléchargeable gratuitement sur le site d'Elektor [1].

Listage 1

```
void setup()
{
  //after reset, start with sleep mode to stabilize VCC.
  //prepare the watchdog for wakeup

  asm ("WDR"); //clear wd-timer
  WDTCR= 0x1c; //set WDCE, WDE to enable setting
  WDTCR= 0x1c; //wd-timer 0.25s
  WDTCR= 0x14; //clear WDE
  WDTCR= (WDTCR | 0x40); //set WDIE, enable WD-interrupt
  interrupts(); //enable interrupts

  SMCR= 0x0f; //prepare xtd-standby
  asm ("SLEEP"); //enter sleep mode
  SMCR= 0x00; //finish, not necessary
  noInterrupts(); //disable interrupts, after wakeup

  MCUSR= (MCUSR & 0xf7); //clear WDRF, wd-reset active
  WDTCR= WDTCR | 0x18; //set WDCE, enable setting WDE
  WDTCR= 00; //clear WDIE, disables wd-timer

  . . .
}
```

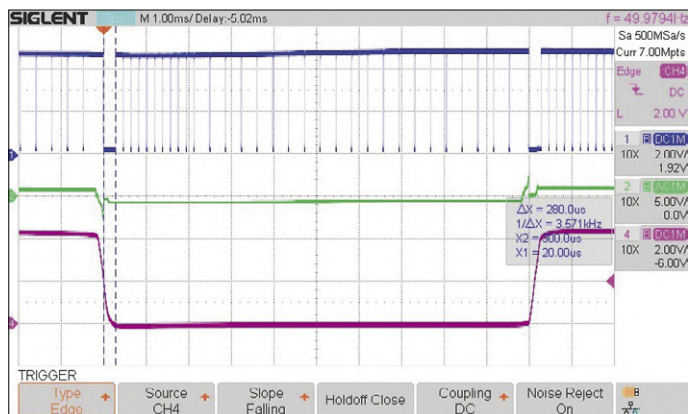


Figure 9. Allumage d'une lampe à LED au passage par zéro de la tension secteur. CH1 (bleu) : impulsions de gâchette; CH2 (vert) : tension aux bornes du triac ; CH4 (rouge) : signal de passage par zéro.

Le choix de la fonction (interrupteur ou minuterie) est fait au moyen d'un bouton. Avec le mode de fonctionnement *Autostart*, la minuterie démarre immédiatement après la mise en marche puis coupe la charge après expiration. Cette fonction convient pour limiter la durée d'éclairage d'une lampe. L'heure d'extinction est enregistrée dans le programme.

Le programme délivre des impulsions d'amorçage du triac d'une durée de 10 μ s, séparées par des intervalles de 278 μ s. Comme ces impulsions ne sont pas liées à la fréquence du secteur, la commutation est asynchrone par rapport aux alternances. Dans le pire des cas, la charge n'est activée que 278 μ s après le passage par zéro. À cet instant, la tension du réseau sera déjà de 29 V, ce qui se traduit par des pics de tension problématiques pour la CEM. L'oscillogramme (**fig. 8**) montre la séquence des événements durant 1 s. Les impulsions d'amorçage se déplacent sans corrélation avec le passage par zéro. Selon le délai entre impulsions de déclenchement et passage par zéro, la tension aux bornes du triac n'est pas la même.

Dans la prochaine application, avec détection du passage par zéro, l'amorçage du triac sera synchrone. Contrairement aux mesures suivantes, c'est V_{CC} qui est ici le potentiel de référence. Comme le potentiel de T1 du triac est celui de V_{CC} , la polarité des impulsions d'amorçage de la gâchette est négative.

Interrupteur et minuterie synchrones

Cette fois c'est un signal synchrone avec les alternances qui signale les passages par zéro. Appliqué à l'entrée INTO, le front descendant de ce signal déclenche une interruption et marque ainsi le début de la première demi-onde. La fin de cette demi-onde et le début de la demi-onde suivante sont déterminés par *Timer2*, pour faire l'économie d'une détection plus complexe du deuxième passage par zéro.

Pour un amorçage sûr, même en présence d'un courant de charge faible, la première impulsion de déclenchement de chaque demi-onde dure 280 μ s. Elle est suivie d'impulsions de déclenchement de 10 μ s supplémentaires. Cela garantit que le seuil d'intensité I_t du triac sera dépassé même en présence de charges à haute impédance (**fig. 9**). On remarque les pics de tension (environ 3 V) aux bornes du triac à la fin de chaque demi-onde. C'est parce que, à la fin de la demi-onde, le faible courant de la lampe à LED connectée pour ces essais tombe en dessous de I_H . Ceci ne cause pas de perturbations notables.

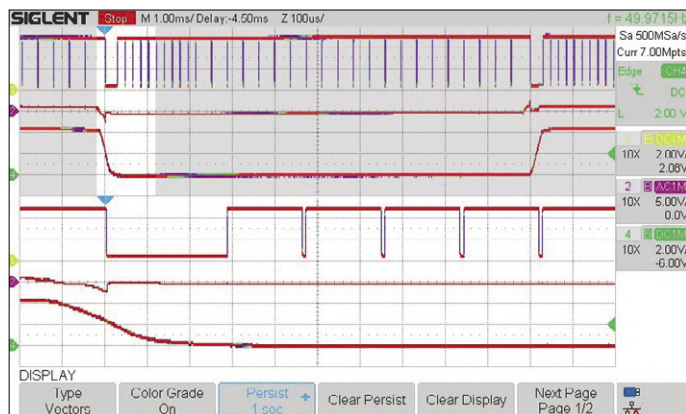


Figure 10 : Allumage d'une lampe à LED au passage par zéro de la tension. Progressions sur 1 s. CH1 : impulsions de gâchette ; CH2 : tension aux bornes du triac ; CH4 : signal de passage par zéro.

La superposition des courbes durant 1 s sur la **fig. 10** montre la synchronisation des impulsions d'amorçage avec la tension du réseau.

Gradateur

L'application par excellence du triac est le gradateur d'intensité lumineuse par découpage de l'angle de phase. La durée de mise sous tension de la charge varie par rapport au passage par zéro de chaque demi-onde du secteur (\approx angle de phase). Les gradateurs peuvent causer des problèmes de CEM considérables, car des flancs abrupts et élevés se produisent en fonction de l'angle de découpage de phase. Dans le cas le plus défavorable, l'amorçage a lieu quand la tension du secteur atteint sa valeur de crête (angle de phase = 90°). Lors de l'amorçage du triac, la tension à ses bornes chute rapidement de sa valeur maximale à zéro (**fig. 11**). Selon la nature de la charge, cela peut se traduire par des courants très élevés et donc des niveaux de bruit élevés. Un filtre CEM en amont (récupéré p. ex. sur un ancien gradateur HS) est donc indispensable.

De nombreuses lampes à LED ne fonctionnent pas avec les variateurs conventionnels en raison de leur mode d'alimentation. C'est aussi le cas avec ce module qu'il faudra donc utiliser avec des luminaires à LED qui acceptent les gradateurs. Ce qui est intéressant ici, c'est qu'après l'impulsion initiale

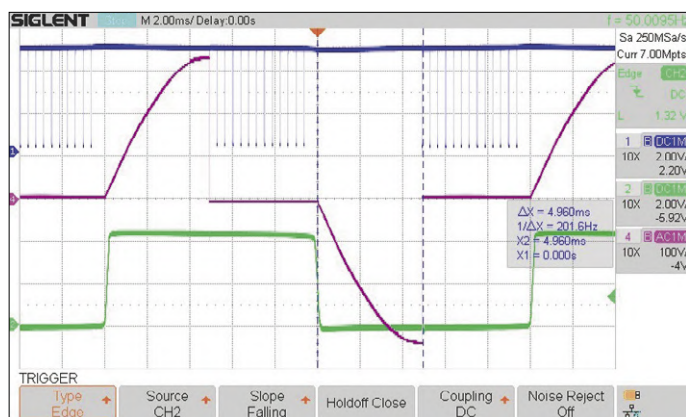


Figure 11. Gradateur avec découpage de l'angle de phase. CH1 (bleu) : impulsions de gâchette ; CH2 (vert) : signal de passage par zéro ; CH4 (rouge) : tension aux bornes du triac.

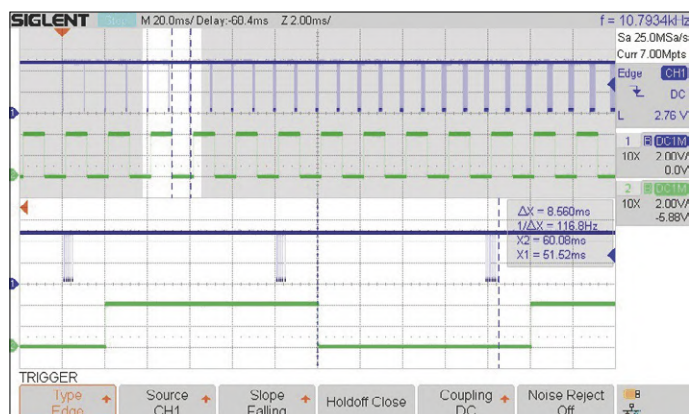


Figure 12. Impulsions de commande après un redémarrage. CH1 (bleu) : impulsions de gâchette ; CH2 (vert) : signal de passage par zéro. La zone agrandie ci-dessous montre l'augmentation de 5 à 7 impulsions.

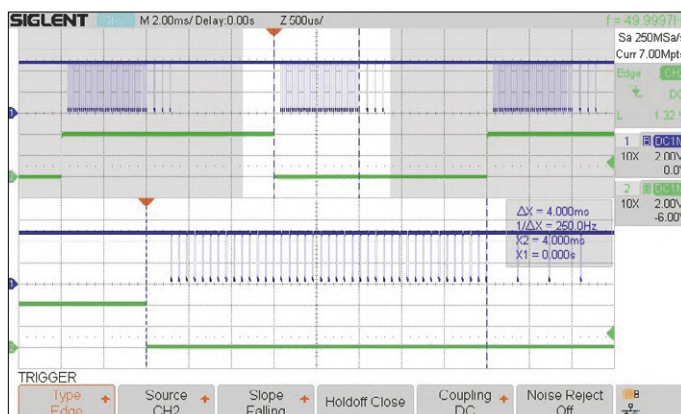


Figure 13. Impulsions de commande en régime permanent avec une portée élargie dans le bas. CH1 (bleu) : impulsions de gâchette ; CH2 (vert) : signal de passage par zéro. Des impulsions de remplissage sont produites pendant 5 ms après chaque passage par zéro.

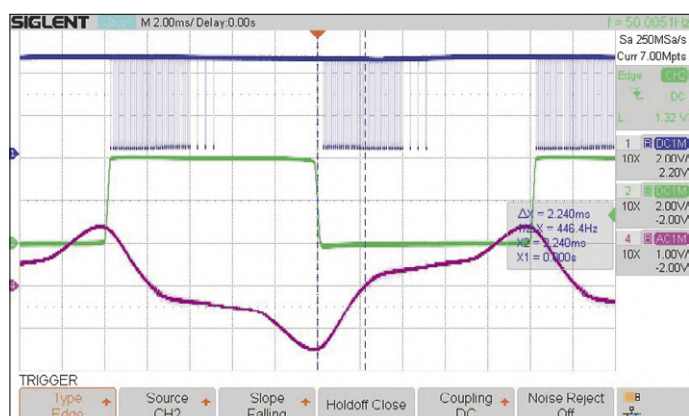


Figure 14. Courant de transformateur à vide. CH1 (bleu) : impulsions de gâchette ; CH2 (vert) : signal de passage par zéro ; CH4 (rouge) : courant de transfo

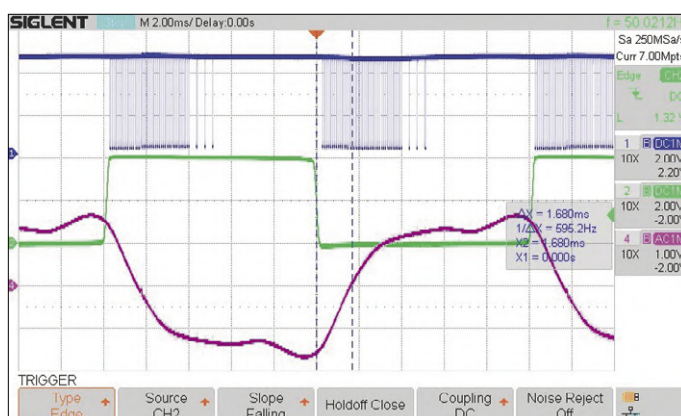


Figure 15. Courant de transformateur à faible charge. CH1 (bleu) : impulsions de gâchette ; CH2 (vert) : signal de passage par zéro ; CH4 (rouge) : courant de transfo

d'amorçage, d'autres impulsions sont produites jusqu'à la fin de la demi-onde, ce qui permet d'utiliser des charges plus complexes. À pleine puissance, la première impulsion après le passage par zéro dure 190 μ s pour être sûr que le seuil d'intensité I_H est dépassé sans risque, même avec des charges à haute impédance. La puissance de sortie est réglée à l'aide des deux boutons ou au moyen d'un optocoupleur.

Charges inductives : amorçage en douceur

La mise sous tension des charges inductives, en particulier des transfos, donne lieu à un appel de courant de forte intensité. Le passage par zéro de la tension est le moment le plus critique. Le magnétisme résiduel du noyau peut également poser des problèmes, car il entraîne une asymétrie du courant d'appel par rapport à la ligne du zéro. La courbe B de la boucle ou du cycle d'hystérésis [2] du noyau est décalé dans une direction et atteint ainsi facilement la zone de saturation. Ensuite, l'enroulement perd soudainement son inductance et le courant dont l'intensité continue d'augmenter n'est presque plus limité que par la (faible) résistance du fil, etc. Dans cette situation, les transfos toriques sont mal lotis. C'est pourquoi il faut augmenter la marge de précaution bien au-delà de l'intensité nominale

du courant de charge.

Parfois, on limite le courant d'appel par une thermistance ou une résistance en série avec relais temporisateur, jusqu'à ce qu'un champ magnétique symétrique et donc un courant alternatif propre soient établis. Théoriquement, ce problème peut également être résolu par une croissance lente du courant d'appel par déphasage de l'angle d'amorçage. On amorce le transfo avec un angle de phase d'environ 180° que l'on réduit lentement jusqu'à 0° . C'est ce qui est fait dans cette application. La **figure 12** montre la séquence d'impulsions après redémarrage. La première impulsion d'amorçage intervient environ 8,6 ms après le passage par zéro. Après chaque alternance complète, une impulsion supplémentaire est ajoutée avant, jusqu'à ce que le temps soit réduit à environ 0,3 ms. La **figure 13** montre l'état d'équilibre. Au cours des trois premières ms, le courant de charge passe par zéro en raison du déphasage. Pour obtenir un flux de courant continu, les impulsions de commande doivent être rapprochées durant cet intervalle. La **figure 14** montre le courant du transfo à vide. Le signal de passage par zéro indique la référence à la tension du secteur. Vous pouvez voir le courant induit, qui atteint son intensité maximum lors du passage par zéro de la tension. L'autre part

résulte des pertes dans le fer et le cuivre du transfo. Le fait que le courant ne soit pas déphasé exactement de 90° est dû aux pertes et aux défauts de linéarité du noyau. La **figure 15** montre le courant du transfo sous charge. Le courant de charge et le courant induit sont superposés. La charge réduit le déphasage. Une condition préalable à un fonctionnement fiable est la symétrie de l'amorçage du triac. Les impulsions d'amorçage doivent être équidistantes du passage par zéro dans les deux demi-alternances, de sorte que l'angle de circulation du courant soit lui aussi symétrique. Les intensités I_{GT} et surtout I_H jouent un rôle important ici. En amplifiant le flanc montant du courant de gâchette avec le réseau RC, on assure au triac un amorçage sûr. Pour que le triac reste conducteur, le courant du transfo doit être supérieur au seuil d'intensité I_H . Les transfos toriques sont problématiques en raison de leur inductance élevée. Heureusement, une petite charge résistive supplémentaire (R25 à R27, en option) suffit pour résoudre ce problème. Certains transfos toriques nécessitent même une charge externe supplémentaire.

La programmation de la commande du triac uniquement par impulsions était un défi, surtout avec les transfos. J'espère que vous en capterez les astuces grâce aux commentaires

À propos de l'auteur

Roland Stiglmayr est un ingénieur en électronique spécialisé dans les technologies de l'information. Il a 40 ans d'expérience professionnelle en recherche et développement dans les domaines de l'ingénierie de contrôle, de la technologie HF, des communications mobiles, des fibres optiques et de la transmission de données. Il exerce une fonction de consultant.

détaillés de mon code.

Amusez-vous bien avec les applications décrites ! Si vous envisagez de nouvelles, n'hésitez pas à modifier le microprogramme comme bon vous semble. ◀

190047-02



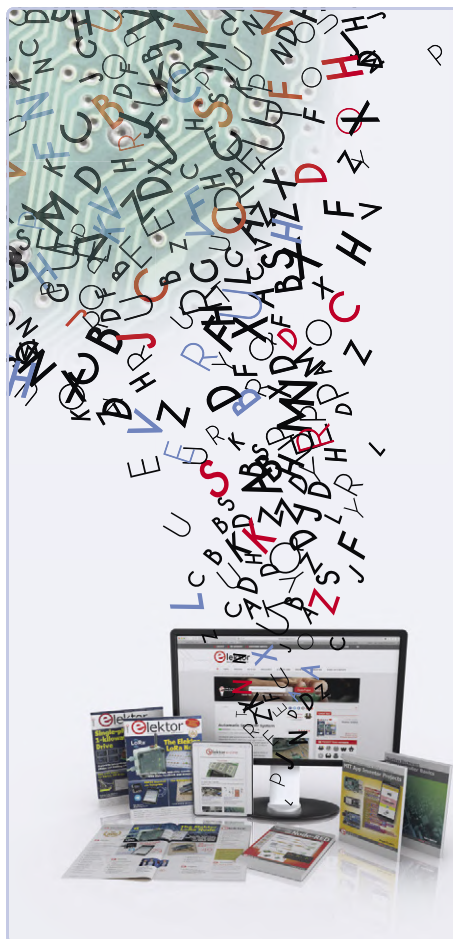
@ WWW.ELEKTOR.FR

→ Elektor Arduino Uno R4
www.elektor.fr/elektor-uno-r4

Liens

- [1] Page du projet sur le site d'Elektor Labs : www.elektormagazine.fr/190047-01
- [2] Boucle d'hystérésis magnétique:
www.goudsmitmagnets.com/fr/wiki/199/hysteresis-dans-les-materiaux-ferromagnetique-courbe-bh

Publicité



Elektor cherche des auteurs

Le coronavirus bouleverse nos vies, avec parfois des conséquences positives.

Le temps libéré, vous pouvez l'utiliser pour **partager** vos connaissances en **électronique** avec d'autres. Selon vos talents, le plus simple consiste à donner des cours **vidéo** ou à écrire un **article** ou un **livre**. Vous avez une bonne idée ? Action !

Elektor vous assistera. Outre la satisfaction de cette expérience, il y aura des recettes pécuniaires.

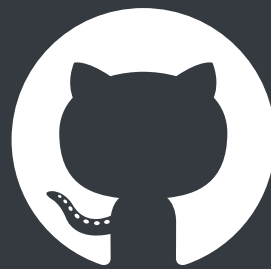
Faites-nous part de votre idée, nous vous répondrons.

elektor.fr/cherche-des-auteurs

elektor
design > share > sell

COMMENT télécharger-t-on depuis GitHub ?

GitHub pour les nuls



Clemens Valens (Elektor Labs)

Si vous utilisez des microcontrôleurs, si vous pratiquez l'Arduino ou d'autres logiciels libres, vous êtes forcément déjà tombé sur *GitHub*.

GitHub est le lieu par excellence du partage de code source ouvert. Ça va du simple croquis Arduino qui tient en un seul fichier à d'énormes applications déployées dans le nuage, dont les nombreux dépôts (*repository* en anglais). Ceux-ci contiennent des milliers de fichiers de logiciel, mais aussi des fichiers de conception électronique ou simplement du texte.

Ci-gît le code

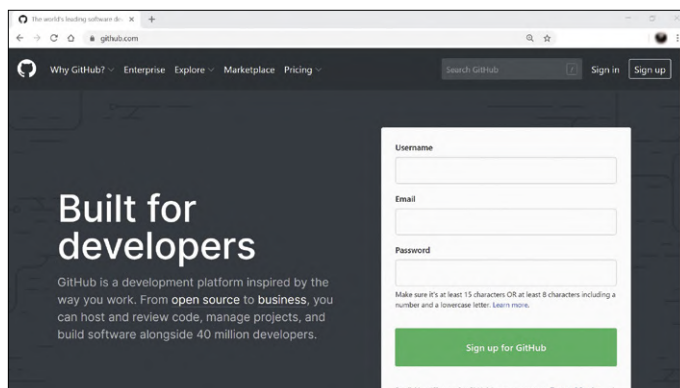
En anglais, le mot *git* désigne ce que nous appelons un connard (ou une conne), Cette espièglerie terminologique, nous la devons à l'inventeur de Linux qui a recyclé ce mot pour désigner un système de gestion des versions des logiciels, lequel garde la trace des modifications apportées à tout document dont il a connaissance. *GitHub* est une plateforme pour *gits* qui permet à des groupes de personnes de coopérer à un même logiciel. Comme outil pour logiciels professionnels et de production, il dispose de nombreuses fonctions et options dont la plupart des usagers n'ont jamais besoin et qui en compliquent l'utilisation.

Comment télécharger depuis *GitHub* ?

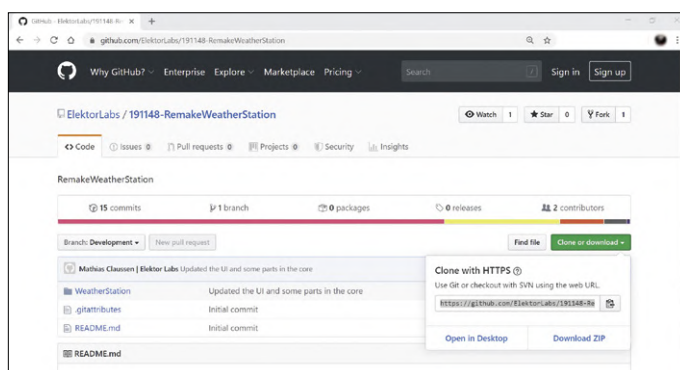
Elektor publie son code sur *GitHub* d'où nous invitons nos lecteurs à le télécharger. On nous demande souvent *comment télécharger quelque chose de GitHub*, une question qui peut paraître stupide à l'utilisateur expérimenté. Il est vrai que vous ne pouvez pas télécharger les fichiers que vous voyez dans un dépôt, faute de bouton de sauvegarde ou de téléchargement. Si vous cliquez sur un fichier, il s'ouvre dans un navigateur, mais n'est pas téléchargé. Avec un clic droit sur un fichier et un enregistrement, vous n'obtenez que du code HTML, pas le contenu escompté.

Mode brut

Pour le télécharger, il faut d'abord ouvrir un fichier donné en mode *Raw*, puis faire un clic droit pour l'enregistrer en tant que fichier. Cette méthode est inutilisable avec des dossiers multiples.



Tout le monde est censé pouvoir l'utiliser, mais *GitHub* est fait pour les développeurs de logiciels professionnels et de nombreuses fonctions sont donc inutiles pour la plupart des utilisateurs.



Le bouton vert «Cloner ou télécharger» n'apparaît que dans la racine du dépôt.

Le bouton vert : cloner ou télécharger

Le bouton de téléchargement facile n'est visible que si vous êtes dans la racine du dépôt. Là, les choses deviennent faciles : il suffit de cliquer sur le bouton vert *Cloner ou télécharger*. Cependant, ne vous laissez point dérouter par les options de «clonage avec HTTPS», ou lors de l'ouverture sur votre bureau, ou encore lors du téléchargement d'un fichier ZIP. Ici, il suffit de cliquer sur «Télécharger ZIP», ignorez le reste. En fait, vous n'avez besoin ni d'un compte ni de vous connecter.

Arduino & GitHub

Le nom du fichier téléchargé peut se terminer par «-master». Une fois décompressé, vous obtenez un dossier du même nom. La plupart du temps, cela n'a pas vraiment d'importance, mais cela en a quand le téléchargement se trouve être un croquis (*sketch*) Arduino. Pour l'IDE Arduino, le nom du fichier .INO doit être le même que celui du dossier. Il suffit de supprimer

la partie «-master» en renommant le dossier.

En principe, ceci est inutile si le croquis se trouve dans un **dossier imbriqué dans le fichier .ZIP** car le dossier imbriqué portera le même nom que le fichier .INO.

Si le téléchargement est une bibliothèque Arduino, alors l'IDE sait quoi en faire. Vous pouvez l'importer en utilisant l'option *Add .ZIP library* du menu *Sketch / Include library*.

Versions stables

De nombreux projets sur GitHub évoluent sans cesse. Si vous téléchargez le fichier .ZIP comme nous l'avons fait ci-dessus, vous obtenez un instantané d'une version en cours de développement qui peut ne pas fonctionner ou ne pas répondre à vos attentes. C'est pourquoi on publie des versions stables, dites *releases*, fonctionnelles et testées. Ces versions autorisées sont souvent hébergées sur un autre site, par exemple sur le site officiel du projet, mais elles ont leur place aussi sur GitHub. Consultez l'onglet *Releases* pour voir s'il y en a et, le cas échéant, téléchargez la version ou le fichier le plus récent dont vous avez besoin. Vous pouvez également consulter l'onglet *Releases* si le code source du projet ne nous intéresse pas, mais que vous voulez les fichiers de code binaire précompilés à exécuter sur votre plate-forme. C'est souvent le cas lorsque la compilation du projet est trop complexe pour vous ou même impossible faute d'outils adéquats.

Clonage d'un dépôt

Enfin, vous pouvez utiliser la méthode officielle de *GitHub* pour télécharger un projet, qui consiste à le cloner. Pour cela, il faut

d'abord installer un client *git* sur votre ordinateur. Il en existe avec de belles interfaces graphiques, d'autres contrôlés par ligne de commande. Dans les deux cas, il vous faudra l'URL complète du dépôt à cloner. Pour l'obtenir, cliquez sur le bouton de téléchargement, mais cette fois en copiant l'URL dans la case *Cloner avec HTTPS*. Pour cela, cliquez sur le bouton *Copier* voisin.

Clonage en ligne de commande (CLI)

Dans l'interface en ligne de commande, naviguez d'abord jusqu'au dossier où vous voulez votre clone du dépôt, puis tapez la commande *git clone* et collez derrière l'URL du dépôt puis appuyez sur la touche Entrée. Si tout va bien, le processus de clonage commencera. Exemple :

```
git clone https://github.com/ElektorLabs/180468-ESP32-Weatherstation.git
```

Ouvrir dans le bureau

Dans le menu de l'interface du bureau, sélectionnez *Fichier / Cloner le dépôt...*, naviguez jusqu'au dossier de destination et collez l'URL que vous avez copiée auparavant dans la case *URL du dépôt*, puis cliquez sur *Cloner*.

Clonage, plus facile que télécharger

Une fois que vous savez y faire, le clonage est probablement plus facile que le téléchargement. Inutile de déballer quoi que ce soit, vous clonez directement dans le dossier de votre choix. Et vous ne serez pas embêté non plus par l'ajout de «-master» au nom du dépôt. ◀

200047-03

Publicité



Vous souhaitez publier votre montage dans le magazine ?

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

Cliquez sur « Créer un projet ».

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

Remplissez les différents champs du formulaire.

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

Labo d'Elektor : www.elektor-labs.com

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

ÉTAT DU PROJET

- ☒ TOUT
- ☐ PROPOSITION (743)
- ☐ EN COURS (41)
- ☐ TERMINÉ (330)
- ☐ BROUILLON (1)

DATE DE PUBLICATION

- ☒ TOUT
- ☐ CETTE SEMAINE (6)
- ☐ CE MOIS-CI (18)
- ☐ CETTE ANNÉE (160)
- ☐ PLUS D'UN AN (1115)

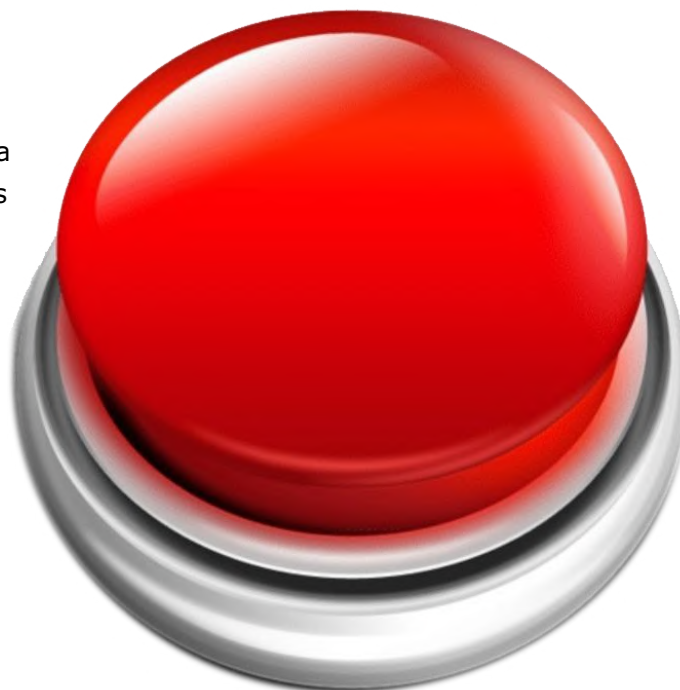
dessine-moi un bouton pour l'IdO

Boutonnière n° 2 :

Prototypage entre la matière et le nuage

Veikko Krypczyk (Allemagne)

L'idée d'un bouton pour l'IdO repose sur le principe de la communication entre un matériel réduit à peu de choses et un service distant dans le nuage, virtuellement sans limite. Il faut pour cela établir une connexion au réseau local. Pour cette tâche, les petites cartes à microcontrôleurs pas chères comme l'ESP8266 NodeMCU sont idéales. Nous mettrons en lumière ici les aspects de l'IdO pertinents pour cet usage et présentons des idées pour des projets personnels. Tout est relativement facile à gérer et à la portée de chacun. À vous ensuite de prolonger...



L'internet des objets (IdO) rend potentiellement intelligents toutes sortes de dispositifs pourtant bien modestes, en les reliant au monde extérieur grâce à l'internet. Dans la première partie de cet article publiée dans le précédent numéro, nous avons décrit l'architecture de base d'un paysage d'IdO, composé de capteurs/actionneurs et d'une infrastructure (*backend*) de serveur. Vous n'aurez ni à construire ni à programmer vous-même l'infrastructure. Il suffira de tirer parti de l'offre

diversifiée dématérialisée dans le nuage. La communication passe par l'internet. C'est simple si vous vous en tenez à des normes établies. Pour les demandes de renseignements (= requêtes), la communication basée sur *REST* a fait ses preuves. Le client, dans ce cas le dispositif IdO, peut transmettre des données à des services (sur un serveur) ou les interroger. Cette deuxième partie de l'article traitera du matériel et des logiciels du côté des dispositifs de l'IdO.

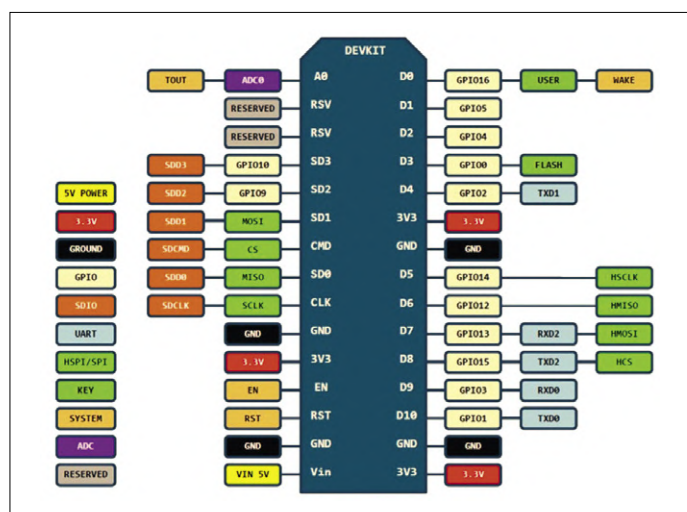


Figure 1. Affectation des broches de la carte microcontrôleur ESP8266 NodeMCU.

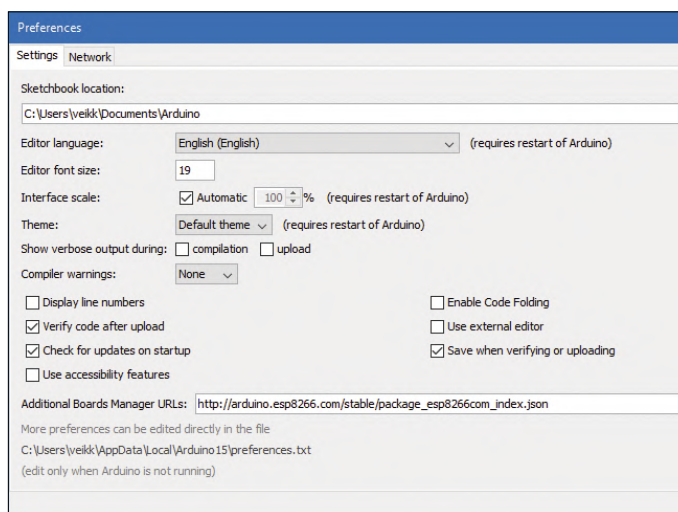


Figure 2. Configuration de l'EDI Arduino pour la programmation de l'ESP8266.

Sélection du matériel

Quel matériel utiliser ? Il existe diverses options, p. ex. un nano-ordinateur comme le **Raspberry Pi**. Outre le WLAN, le RPi offre une puissance de calcul suffisante pour toutes les applications. Si nous voulions actionner, dans un même lieu, plusieurs boutons, capteurs et actionneurs ayant des fonctions différentes, tous ces dispositifs seraient connectés aux broches GPIO du Raspberry Pi. Le RPi utiliserait le réseau pour se connecter au service dans le nuage. Pour des tâches réduites, un Raspberry Pi serait trop encombrant et il exigerait une alimentation permanente et un bloc d'alimentation. Trop gourmand.

Voyons l'**Arduino Uno** ! Comme on sait, ce n'est pas un ordinateur, mais une carte à µC. Ses possibilités matérielles et logicielles sont vastes et les bibliothèques et les exemples d'application sont innombrables. La programmation est aisée grâce à l'IDE Arduino. Cependant, l'Arduino ne sait rien faire du WLAN, de sorte que pour cette fonction cruciale pour l'IdO, il faudrait lui adjoindre un module supplémentaire. Trop lourd.

Voyons alors une carte à microcontrôleur **ESP8266 NodeMCU** ([1] et encadré). Elle réunit toutes les fonctions nécessaires, dont l'interface WLAN (**fig. 1**). Il suffit de connecter un poussoir (c'est notre bouton IdO) entre la borne D7 sur la carte et GND. Quand on appuie sur ce bouton, un signal doit être transmis par WLAN. Il y a une LED d'état sur la carte, nous n'avons donc besoin d'aucun autre composant. L'alimentation est fournie par le port micro-USB de l'ordinateur, pour l'instant.

Préparation du NodeMCU ESP8266

Comme les cartes Arduino, les µC ESP8266 peuvent être programmés dans l'IDE Arduino. Il faut d'abord connecter la carte avec un câble de données au PC via le port USB et, si nécessaire, installer le pilote pour le jeu de puces CH340 [2]. Si c'est déjà fait chez vous, sautez cette étape, sinon installez l'EDI Arduino sur votre ordinateur [3]. L'IDE doit être complété par un élément supplémentaire : sous *Fichier* -> *Préférences*, entrez le lien [4] pour les *URL supplémentaires des gestionnaires de cartes* (**fig. 2**). Fermez la boîte de dialogue avec OK. L'installation proprement dite s'effectue sous le menu *Outils*, où se trouve l'entrée *Modules ESP8266 génériques*. Sélectionnez-la et cliquez sur le *gestionnaire de cartes* dans le menu contextuel en haut, réduisez la zone de recherche avec l'entrée *NodeMCU* et effectuez l'installation (**fig. 3**).

Un simple programme «Hello World» (**listage 1**) permet de vérifier si tout a bien fonctionné. Copiez le code et chargez-le dans le µC à l'aide de *Sketch|Upload*. La LED sur la carte est éteinte par la petite boucle avec `digitalWrite(LED, HIGH)` pendant 3 s et allumée par `digitalWrite(LED, LOW)` pendant 1 s.

Pour interroger un bouton (relié à la broche D7), un peu de code suffit. Celui-ci vérifie par exemple si le bouton a été fermé :

```
int bouton=D7 ;
status=digitalRead(button) ;
if (status==LOW)
{ ..... }
```

avant de déclencher des opérations ultérieures.

Il ne reste plus qu'à accéder à l'internet. Le matériel néces-

Listage 1. «Hello World» (LED clignotante) sur l'ESP8266

```
#define LED D4

void setup()
{
    pinMode(LED, OUTPUT);
}

void loop()
{
    digitalWrite(LED, HIGH);
    delay(3000);
    digitalWrite(LED, LOW);
    delay(1000);
}
```



Figure 3. Installation des bibliothèques pour la carte ESP8266 NodeMCU.

La carte Node ESP8266MCU

Sur la carte ESP8266 NodeMCU on trouve la puce ESP8266 avec une vitesse d'horloge de 80 MHz et jusqu'à 4 Mo de mémoire flash. Le NodeMCU ESP8266 contient un module WLAN et une interface USB avec convertisseur UART/USB (CH340G).

La petite carte présente les caractéristiques techniques suivantes :

- soutient le WPA/WPA2
- puce : ESP8266
- processeur : Tensilica L106 – horloge : 80 MHz
- architecture du système : 32 bits, Flash : jusqu'à 4 Mo
- protocoles pris en charge : SPI, UART, I²C, I2S, IR Remote Control, PWM, GPIO
- niveau logique : 3,3 V
- consommation : < 10 µA en veille et jusqu'à 200 mA en charge
- alimentation et connexion de programmation via Micro-USB : 5 V
- puce de programmation : CH340G
- taille : 5,5 x 3,1 x 1,2 cm ; poids : 42 g

Ces spécifications montrent que la carte ESP8266 NodeMCU est bien adaptée pour faire fonctionner de petits dispositifs d'IdO particulièrement économes en énergie

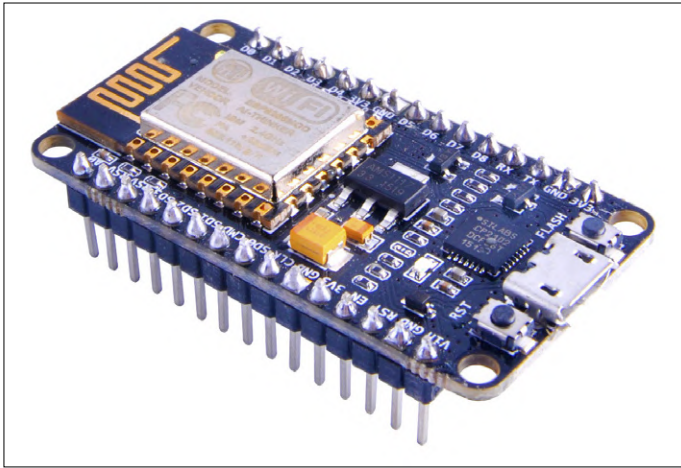


Figure 4. ESP8266 NodeMCU avec module WLAN

saire se trouve déjà sur la carte de l'ESP8266 sous la forme d'un minuscule module WLAN (gris-vert) (**fig. 4**).

Si vous voulez utiliser le WLAN de l'ESP8266, il faut inclure une bibliothèque avec la clause `#include ESP8266WiFi.h` dans l'en-tête du code source. La bibliothèque attend les données correctes pour la connexion à l'internet, le SSID et le mot de passe. Ces données sont (d'abord statiquement) stockées dans le code source, plus tard l'utilisateur devrait bien sûr pouvoir configurer les données. L'établissement d'une connexion WLAN est merveilleusement simple (**listage 2**).

Il faut maintenant procéder de telle sorte que lorsqu'un événement se produit, nous envoyons un message sur le réseau. Un terminal REST, tel que décrit dans la première partie de cet article, servira de récepteur dans cette communication sur le réseau. Ce point de terminaison reçoit les données ou le message et peut en faire un traitement ultérieur et déclencher une réaction. Avec un simple bouton IdO, il n'y a aucune autre

Listage 2. Se connecter à l'internet

```
include <ESP8266WiFi.h>
void setup()
{
    Serial.begin(115200);

    WiFi.begin(<SSID>, <Password>);

    while (WiFi.status() != WL_CONNECTED);
    {
        delay(1000);
        Serial.print("Connecting ...");
    }
}
```

donnée à transmettre que l'événement «bouton actionné». Si l'application était étendue en conséquence, des données supplémentaires pourraient être transmises en tant que paramètres par la requête REST. Essayons !

Données dans le nuage

Un terminal REST est nécessaire pour envoyer des données dans le nuage, c'est-à-dire vers un serveur distant. Vous n'avez pas besoin de programmer cela vous-même, vous pouvez utiliser les services existants. Pour les tests et les applications privées, ces services en nuage sont généralement gratuits. Nous voulons procéder de manière expérimentale, étape par étape. Sous [5], vous pouvez créer une API REST gratuite à des fins de test, à laquelle vous pouvez accéder à partir de notre µC via le WLAN. Cela fonctionne en quelques clics de souris, même sans enregistrement préalable. Regardez [6] ; nous avons même déjà créé un point de terminaison (**fig. 5**) ! Pour vérifier si et comment ce point d'arrivée est atteint à

Listage 3. Envoyer la demande GET au point de terminaison REST

```
void loop()
{
    if (WiFi.status() == WL_CONNECTED)
    {
        status=digitalRead(button);
        if (status==LOW)
        {
            digitalWrite(LED, LOW);

            BearSSL::WiFiClientSecure client;
            client.setInsecure();
            HTTPClient https;
            https.begin(client, "https://iotbutton.free.
                               beeceptor.com");

            int httpCode = https.GET();
            if (httpCode == 200)
            {
                String payload = https.getString();

                Serial.println(payload);
            }
            else
            {
                Serial.print("Erreur: ");
                Serial.println(httpCode);
            }
            https.end();
        }
        else
        {
            digitalWrite(LED, HIGH);
        }
    }
}
```

partir du µC, nous envoyons une demande GET au service lorsque le bouton est enfoncé. Il est inséré dans la boucle du programme (**listage 3**).

La ligne de code suivante vérifie si le µC est connecté à l'internet :

```
if (WiFi.status() == WL_CONNECTED)
{ }.....
```

Toutes les lignes incluses du code source ne sont donc traitées que si une connexion à l'internet a été établie.

L'étape suivante

```
status=digitalRead(button) ;
if (status==LOW)
```

vérifie si le bouton a été actionné. Si c'est le cas, un client http est initialisé et l'option de transmission de *commandes https* est donc activée. Cela nécessite une bibliothèque, qui doit être incluse dans l'en-tête avec `#include`. Pour spécifier l'URL de notre point terminal REST, la valeur `https://iotbutton.free.beeceptor.com` est définie. Elle est suivie d'une simple requête GET sur le point final, dont le résultat peut être spécifié par la ligne

```
String payload = https.getString() ;
```

comme une chaîne de caractères.

En réaction directe à l'appui sur le bouton, la LED de la carte doit être allumée, puis éteinte à nouveau dès que le bouton sera relâché (voir ci-dessus). Il est possible de vérifier en ligne si les données sont également arrivées dans le nuage. Allez au point terminal de la page Beeceptor [6] et suivez-y les requêtes au point terminal (**fig. 6**).

Notez que l'envoi gratuit de requêtes par ce service est limité à 50 par jour. Au cours de nos essais, nous avons dépassé ce nombre et avons reçu un message d'erreur avec le code 429 sur la requête GET comme valeur de retour. Toutefois, tant que vous ne dépassez pas le nombre maximum de requêtes, le code de statut 200 (OK) et les données de la demande REST sont renvoyés au µC. Dans l'environnement de développement, vous pouvez suivre dans le moniteur série (*Outils|Moniteur série*) les valeurs de retour reçues par le µC (**fig. 7**).

En appuyant sur un bouton, nous avons donc transmis une demande REST par l'internet. C'est le côté technique du bouton IdO. Le **listage 4** indique le code source complet. Les zones à adapter individuellement sont indiquées en gras.

Courant et sommeil

Nous avons choisi la carte ESP8266 en raison notamment de sa faible consommation. On peut se fier aux valeurs de consommation mentionnées dans [7]. Avec un simple programme Arduino, le courant est d'environ 50 à 70 mA, en fonctionnement WLAN la consommation atteint environ 80...170 mA et pendant quelques millisecondes peut même culminer à 400 mA. D'où l'intérêt d'une réduction drastique de la consommation en mode de sommeil profond. Selon diverses sources, on parle de quelques microampères. Une batterie (rechargeable) devrait tenir des mois. La tension d'alimentation est connectée aux bornes +5V et GND. Il existe deux variantes du mode de sommeil profond.

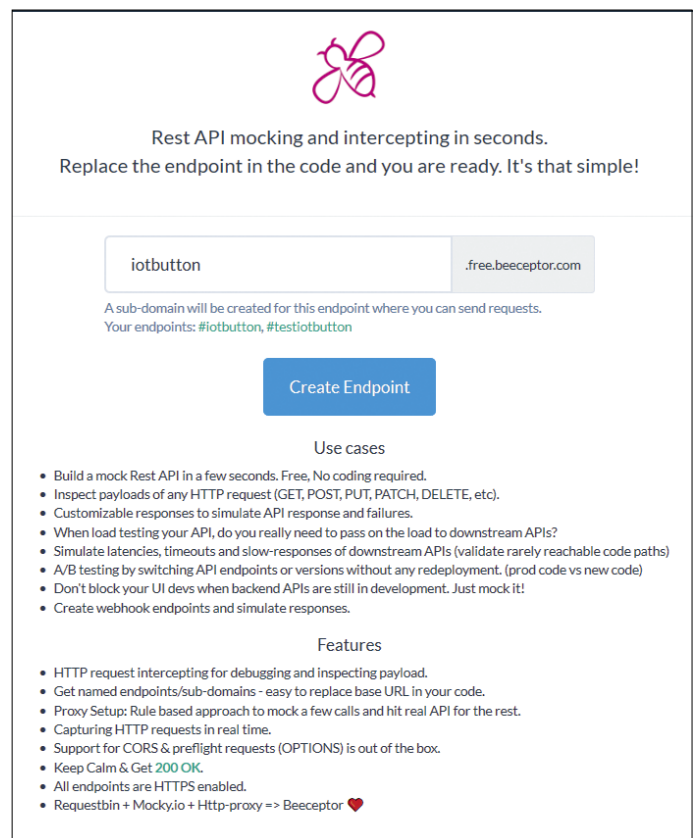


Figure 5. Point final REST pour le test sur le nuage de Beeceptor.

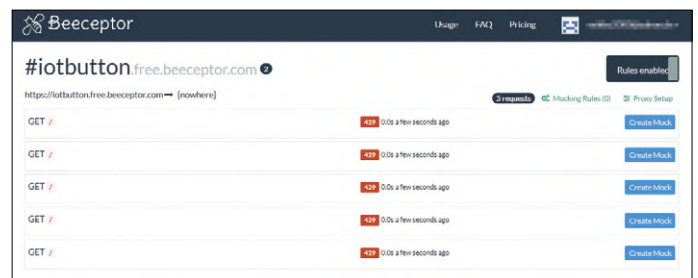


Figure 6. Demandes enregistrées vers l'extrémité REST sur le nuage Beeceptor

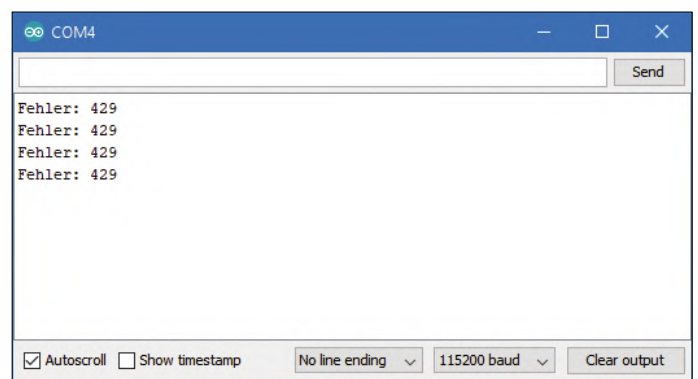


Figure 7. Le Serial Monitor enregistre les réponses du point final REST. Ici, le message d'erreur 429 est émis parce que le nombre de demandes a été dépassé.

Listage 4. Envoi d'une demande GET via REST

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#define LED D4
int button=D7;
int status=1;
void setup()
{
    Serial.begin(115200);
    pinMode(LED, OUTPUT);
    digitalWrite(LED, HIGH);
    WiFi.begin(<SSID>, <Password>);
    while (WiFi.status() != WL_CONNECTED)
    {
        delay(1000);
        Serial.print("Connecting ...");
    }
}

void loop()
{
    if (WiFi.status() == WL_CONNECTED)
    {
        status=digitalRead(button);
        if (status==LOW)
        {
            digitalWrite(LED, LOW);
        }
    }
}
```

```
BearSSL::WiFiClientSecure client;
client.setInsecure();
HTTPClient https;
https.begin(client, "https://iotbutton.free.
                beeceptor.com");

int httpCode = https.GET();
if (httpCode == 200)
{
    String payload = https.getString();
    Serial.println(payload);
}
else
{
    Serial.print("Erreur: ");
    Serial.println(httpCode);
}
https.end();
}

else
{
    digitalWrite(LED, HIGH);
}
}
```

Listing 5. Envoi d'une demande GET par le mode REST et le mode sommeil profond

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>

#define LED D4
void setup()
{
    Serial.begin(115200);
    pinMode(LED, OUTPUT);
    digitalWrite(LED, HIGH);
    WiFi.begin(<SSID>, <Password>);
    while (WiFi.status() != WL_CONNECTED)
    {
        delay(1000);
        Serial.print("Connecting ...");
    }
}

void loop()
{
    if (WiFi.status() == WL_CONNECTED)
    {
        digitalWrite(LED, LOW);
    }
}
```

```
BearSSL::WiFiClientSecure client;
client.setInsecure();
HTTPClient https;
https.begin(client, "https://iotbutton.free.
                beeceptor.com");

int httpCode = https.GET();
if (httpCode == 200)
{
    String payload = https.getString();
    Serial.println(payload);
    digitalWrite(LED, HIGH);
    delay(2000);
    ESP.deepSleep(0);
}
else
{
    Serial.print("Erreur: ");
    Serial.println(httpCode);
}
https.end();
}
```

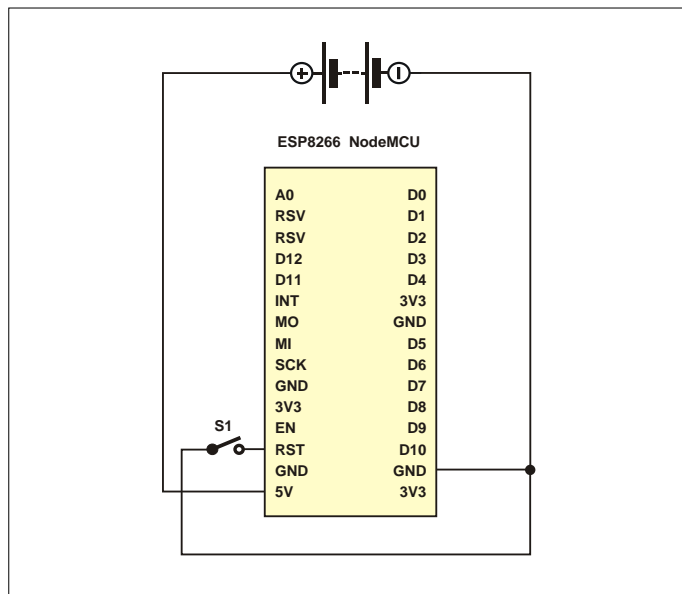


Figure 8. Le schéma de circuit ne contient qu'un seul bouton poussoir en plus du microcontrôleur.

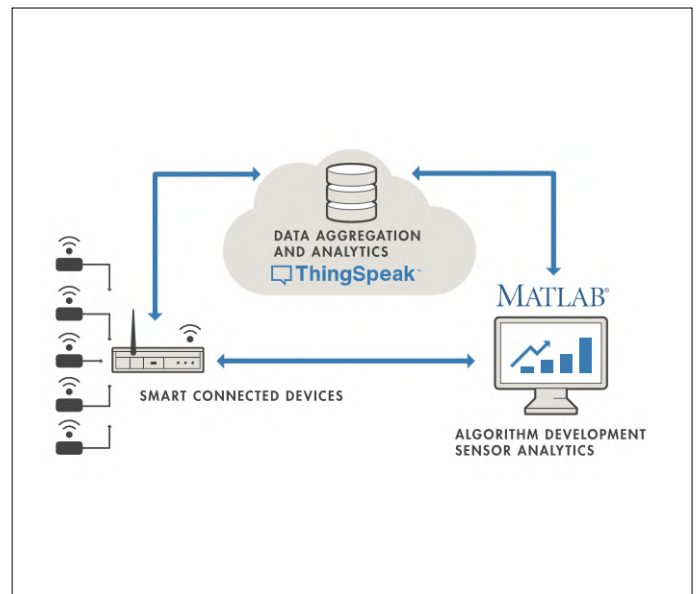


Figure 9. Collaboration entre l'IdO-Device, l'IdO-Cloud et d'autres services en utilisant l'exemple de ThingSpeak.

- **Le sommeil profond des logiciels** : ESP8266 se réveille automatiquement après un certain temps. Pour ce faire, une connexion doit être établie entre le GPIO16 (D2) et le connecteur RST. Après le délai fixé, un signal est envoyé qui redémarre automatiquement l'ESP. Toutefois, la durée maximale est limitée à environ 71 minutes. Cette option est un bon choix dans les scénarios qui transmettent des valeurs mesurées de manière cyclique. La commande correspondante est : `ESP.deepSleep(SLEEP_TIME * 1000000)`
- **Interrompre le sommeil profond** : Le ESP8266 ne redémarre pas automatiquement, c'est une interruption qui le tire de son sommeil profond. Ce mode de sommeil profond est appelé avec : `ESP.deepSleep(0)`. Toutefois, GPIO16 et RST ne doivent pas être reliées. Dès que l'événement est déclenché, on envoie un signal bas (*LOW*) à la broche RST. Cela permet de redémarrer l'ESP8266 et de faire exécuter le logiciel.

Pour le bouton IdO, cela signifie que le bouton n'est plus connecté à une broche GPIO, mais à une broche RST (et à la terre) (**fig. 8**). Dans le code, la vérification de la frappe peut donc être omise. Au lieu de cela, le μ C est mis en mode veille (**listage 5**) après avoir envoyé avec succès les informations par l'interface REST. Compilez le code source (*sketch*) et chargez-le dans le contrôleur. Après le démarrage, il se connecte au réseau, envoie un message et se met en mode veille. Si le microcontrôleur est réveillé par le bouton (*Reset*), le processus est relancé. Vous pouvez suivre en ligne comment l'API REST a été appelée avec succès par la demande GET.

Encore une fois dans le nuage

Techniquement, le bouton IdO est en place. Il est possible à présent de configurer en quelques étapes les actions effectuées à la réception du message du μ C. Pour ces scénarios également, il existe différents services en nuage, tels que *Microsoft*

Liens

- [1] Manuel de l'utilisateur du NodeMCU ESP8266 : www.handsontec.com/pdf_learn/esp8266-V10.pdf
- [2] Pilote CH340 : <https://sparks.gogo.co.nz/ch340.html>
- [3] IDE Arduino : <http://www.arduino.cc/en/main/software>
- [4] URL du gestionnaire de cartes : http://arduino.esp8266.com/stable/package_esp8266com_index.json
- [5] BEECEPTECTOR : <https://beecceptor.com/>
- [6] Console IoT-Button : <https://beecceptor.com/console/iotbutton>
- [7] Consommation de l'ESP8266 : <https://arduino-hannover.de/2018/07/25/die-tuecken-der-esp32-stromversorgung/>
- [8] Nuage IdO Microsoft : <https://azure.microsoft.com/en-en/services/iot-hub/>
- [9] ESP8266 dans le nuage Azure : <https://sandervandeveld.wordpress.com/2019/05/07/connection-a-cheap-esp8266-to-azure-iot-central/>
- [10] NodeMCU dans le nuage Azure : <https://www.thingforward.io/techblog/2018-05-22-connecting-nodemcu-to-microsoft-azure-iot-hub-part-1.html>
- [11] Thingspeak : <https://thingspeak.com/>

Figure 10. Configuration du point final REST (canal) pour *ThingSpeak*.

Figure 11. Les applications déterminent les actions de *ThingSpeak* pour une chaîne.

Azure IoT [8][9][10] ou *ThingSpeak* [11]. L'approche consiste essentiellement à configurer un point de terminaison en ligne directement dans le service en nuage et à l'adresser via l'internet à partir du dispositif IdO. Les actions souhaitées peuvent ensuite être lancées par le service (fig. 9).

Créez un compte gratuit avec le service *ThingSpeak*, lequel ne nécessite qu'une vérification par adresse électronique, puis créez un nouveau point de terminaison REST (canal).

Remplissez le formulaire détaillé (fig. 10) suivant les explications données dans la partie droite, ou donnez-lui d'abord un nom. Chaque canal se voit attribuer une URL unique et la possibilité d'envoyer des données supplémentaires au-delà de la demande à l'aide de paramètres URL. Pour un simple bouton IdO, aucun autre paramètre n'est nécessaire, mais sous *Mes canaux/clés API*, vous pouvez trouver les clés de lecture et d'écriture (et des exemples sur la façon d'inclure les clés dans les requêtes GET) du canal. Si un canal est correctement configuré, vous pouvez utiliser l'élément de menu *Apps* pour définir les actions à déclencher (fig. 11), p. ex. l'envoi d'un e-mail par un service web ou la publication d'un micromessage sur *Twitter*. L'infrastructure (backend) de l'IdO sert pour ainsi dire de couche intermédiaire.

Attention : pour communiquer avec les services du réseau, les dispositifs IdO doivent forcément avoir accès à ces services. Pour cela, ils doivent bien sûr s'identifier, p. ex. avec un nom d'utilisateur et un mot de passe ou avec d'autres caractéristiques uniques (clés). Ces valeurs doivent être stockées dans le code source. Veillez donc à protéger ces informations, notamment lorsque vous copiez le code source ou que vous le publiez sur l'internet (pour un usage gratuit).

Conclusion et perspectives

Ne sont-elles pas fascinantes, ces expériences avec le μ C ESP8266 ? Que vous manque-t-il pour mettre en œuvre des scénarios de l'IdO ? Pour rendre opérationnel le projet du bouton IdO ou pour le transformer en application commerciale, il ne manque plus grand-chose : mécanique (boîtier), alimentation (connexion permanente de la batterie). Des extensions sont envisageables : si par exemple quelqu'un s'approche sans autorisation d'un objet sécurisé, notre bouton IdO déclenché par un détecteur de mouvement pourrait directement déclencher l'alarme et communiquer via l'internet. Ce travail-là sera effectué en ligne, par une configuration appropriée dans le nuage. ◀

190303-B-02

@ **WWW.ELEKTOR.FR**

→ **ESP8266 NodeMCU**

www.elektor.fr/17952

MC14500B : microprocesseur CMOS à un seul bit drôle(s) de composant(s)

Neil Gruending (Canada)

La logique à relais utilisée jadis pour commander les machines a cédé sa place à une logique numérique discrète. Celle-ci restait difficile à modifier et à entretenir. Comme dans les années 1970, les ordinateurs étaient coûteux et complexes, Motorola a inventé l'unité de commande industrielle programmable MC14500 dont l'architecture à 1 bit mérite les honneurs de cette rubrique.

Lancé en 1977, le MC14500 mettait en œuvre le langage Ladder ou *langage à contacts* utilisé dans les automates programmables (API). Le principe est intéressant, car il fait usage de juste assez de logique pour implémenter 16 instructions avec un registre à un seul bit (**fig. 1**) [1]. Selon l'instruction en cours, la ligne de données sera soit entrée, soit sortie. Les sauts se font avec JMP/RTN, d'autres instructions font les opérations logiques de base entre registre de donnée et entrée/sortie de données, mais il n'y a pas d'instructions de calcul, impossibles avec seulement un bit. Tout ça sans compteur ordinal ni mémoire ?

Si, le MC14500 fait appel à un compteur ordinal et une mémoire externes (**fig. 2**) [1]. Selon les besoins de l'application, le compteur ordinal pointe sur l'adresse de mémoire où sont les instructions et autres signaux de commande pour assurer des fonctions telles que le verrouillage de donnée. Ce verrouillage permet au MC14500 de commander avec 1 bit à la fois autant d'entrées et de sorties que nécessaire.

Le schéma de principe de la figure 2 ne vous semble-t-il pas familier ? Il y a là tous les éléments de base distincts d'un ordinateur simple. La société allemande *DATANorf Hard and*

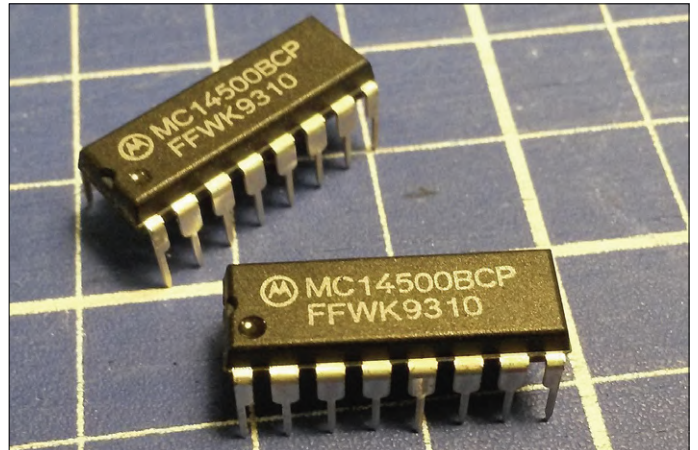


Image: JPL - Own work, CC BY-SA 4.0,
<https://commons.wikimedia.org/w/index.php?curid=48560865>

Software a fabriqué et vendu le kit de formation WDR-1 avec un MC14500 comme unité centrale rudimentaire [2]. Pour le kit, on a rajouté des LEDs en sortie et un clavier pour saisir les instructions. Dieter Mueller a même implémenté un MC14500 avec de la logique standard [3].

La production du MC14500 par Motorola s'est poursuivie environ 20 ans (!). L'information à son sujet est pléthorique. Commencez par sa fiche technique et son manuel d'applications. Ce drôle de paroissien est facile à trouver sur le marché de l'occasion. Vous pouvez aussi implémenter vous-même en logique programmable ce processeur à 1 bit ! ◀

(180575-E-04 VF LK)

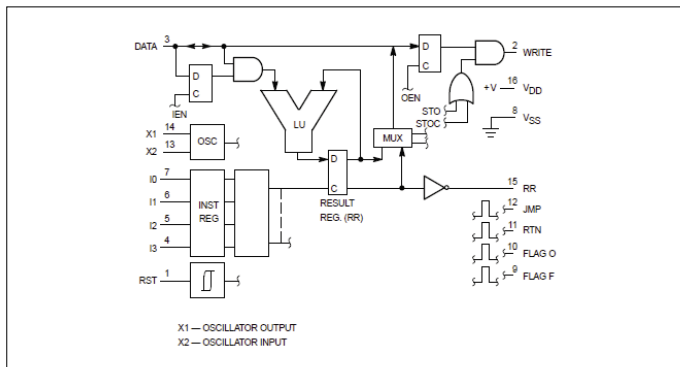


Figure 1 : Le MC14500 dans son biotope [1].

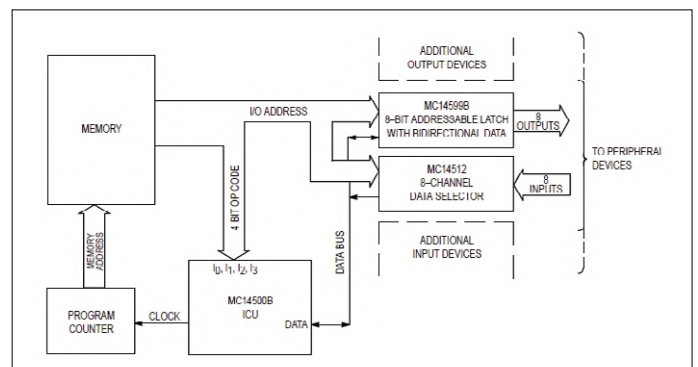


Figure 2 : Architecture interne du MC14500 [1].

Liens

- [1] MC14500 diagrams : www.brouhaha.com/~eric/retrocomputing/motorola/mc14500b/mc14500brev3.pdf
- [2] WDR-1 training educational kit : www.old-computers.com/museum/computer.asp?st=1&c=834
- [3] Dieter Mueller's website : www.6502.org/users/dieter/m14500/m14500.htm



corrections et mises à jour

Questions et réponses

Clemens Valens (Elektor Labs)

Mises à jour et compléments d'information sur des articles publiés par Elektor, avec des tuyaux, des astuces, des conseils ingénieux et des réponses à des questions d'intérêt général posées par des lecteurs.

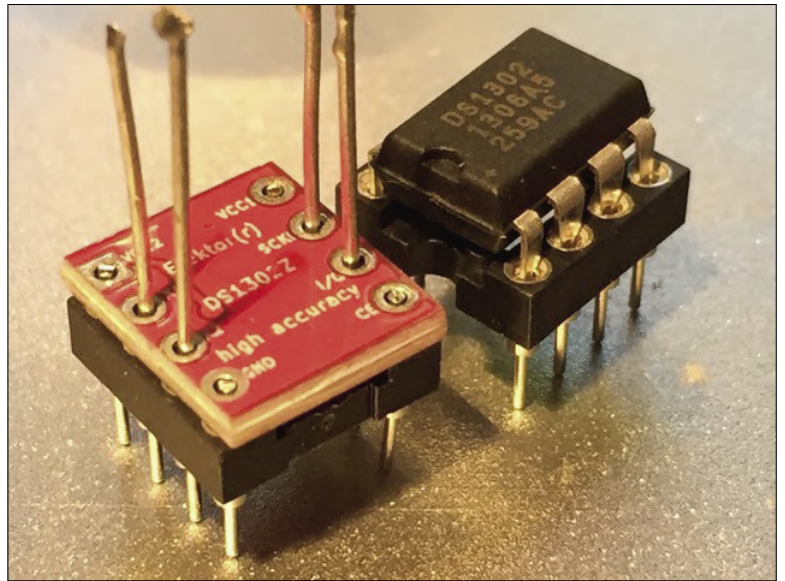
Mise à jour du réveil à triple affichage

Elektor mai-juin 2019, p. 44 (180620)

C'est encore moi avec un bon tuyau. Selon l'article, la passerelle RTC est trop grande pour remplacer simplement le DS1302 (IC2) de l'horloge. Il mentionne ensuite trois façons possibles de modifier l'horloge pour que la passerelle (BoB) s'adapte. J'en ai trouvé une quatrième, meilleure car elle remplace directement le CI existant sans modif. L'astuce consiste à ne pas utiliser de connecteurs mais des pattes de composants coupés, dont tout passionné d'électronique a toujours des provisions.

Une fois le BoB équipé de ses trois composants, placez-le – côté composants vers le bas – sur un support de circuit intégré à 8 broches. Faites passer huit fils par le haut à travers les trous de la carte pour les enficher dans les contacts du support (cf. photo). Une pince plate sera utile pour cette manip. Toutes les broches doivent descendre jusqu'aux contacts. Coupez les fils au ras de la petite carte et soudez-les. Retirez délicatement le BoB du support et insérez-le dans le support pour IC2 sur le circuit imprimé de l'horloge. La photo montre que la RTC est encore plus basse maintenant que le CI d'origine.

Hans Schneider (Allemagne)



Variateur sans scintillement pour LED

Elektor sept. – oct. 2019, p. 12 (190062)

J'ai apprécié l'article, mais n'y manque-t-il pas quelque chose ? Il ne mentionne pas les deux inconvénients de la méthode qui permet de réduire le scintillement et d'obtenir une meilleure CEM.

Primo, la couleur d'une LED dépend de son courant de fonctionnement. Le principe proposé par J. Waeghebaert modifie le courant de fonctionnement et donc la couleur de la lumière. Cela peut être gênant, par exemple, dans la photographie professionnelle. En cas de fonctionnement avec un courant rectangulaire pulsé (PWM), les impulsions de courant sont invariables, la couleur de la lumière résultante ne change donc pas. Elle reste constante sur toute la plage de gradation.

Secundo, une gradation fluide dans le bas de la plage de luminosité n'est pas garantie. Toutes les LED de

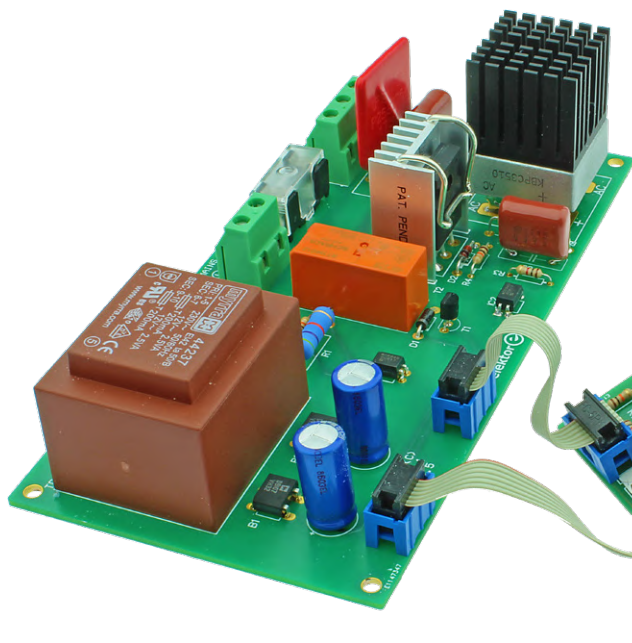
puissance n'ont pas un comportement défini dans le bas de la plage de courant. Certains fabricants spécifient un courant minimum supérieur de 10 % au courant nominal (par exemple). Avec un courant pulsé, il est toujours possible de réduire progressivement la luminosité jusqu'à zéro.

En pratique, il faudra trouver un compromis approprié pour chaque demande.

Daniel E. (Suisse)

variateur pour moteur monophasé 1 kW

Elektor janvier - février 2020, p. 70 (191142, 150199)



Dans le schéma de la figure 1, les symboles «+» et «-» dans le symbole du redresseur B4 doivent être permutés.

Dans le schéma de la figure 2, on ne voit pas tout de suite que les broches SDA et SCL de l'afficheur OLED1 sont connectées l'une à la broche 23 et l'autre la broche 18 du µC IC2.

Dans la liste des composants, la référence correcte du transfo de puissance Myrra TR1 aurait dû être 44237 et non 44327.

Enfin, contrairement à ce que dit l'article, la carte de puissance est à double face.

Aucun des conseils ci-dessus n'a de conséquence sur le fonctionnement du projet, ni sur les PCB associés disponibles dans la boutique en ligne d'Elektor.

www.elektor-labs.com/3980

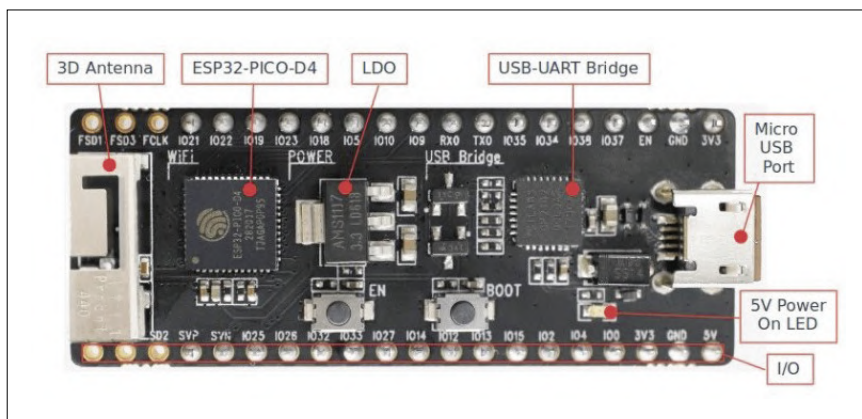
ESP32-PICO-Kit v4.1 est-il programmable avec l'EDI Arduino ?

Q : Les deux modules ESP32-PICO-Kit achetés le mois dernier portent un numéro de version 4.1. Les modules achetés plus tôt ont la version 4 et peuvent toujours être programmés avec l'EDI Arduino. Avec les nouveaux, je n'ai pu télécharger qu'un seul croquis, puis le téléchargement a cessé de fonctionner. Le sketch téléchargé fonctionnait. J'ai testé différentes vitesses de transmission. Quel peut être le problème ?

S. Kepper (Allemagne)

R : Les deux modules que vous nous avez envoyés ont pu être programmés sans aucun problème sur notre système de test. Cependant, nous avons découvert un problème de pilote ou d'autre logiciel avec certaines combinaisons de contrôleurs Windows 10 et USB (plateforme AMD Ryzen) sur lesquels nous n'avons aucun contrôle. Comme solution de contournement, on peut utiliser un vieux hub USB 2.0.

Elektor Labs



Épilogue : Merci beaucoup pour votre réponse. Votre réponse m'a incité à examiner de plus près les ports USB. USB 2.0 et USB 3.0 de mon PC. J'ai constaté que je ne peux pas utiliser l'USB 3.0 sous Windows 7 pour les nouveaux modules ESP32-PICO-Kit 4.1. Cette découverte intéressera d'autres utilisateurs. Reste le mystère de la programmation du module qui a fonctionné la première fois sur un port USB 3.0.

S. Kepper (Allemagne)

190379-C-03

sablier motorisé en BASIC

avec ESP8266 et Annex WiFi RDS

Peter Neufeld (Allemagne)

Dans le dernier numéro d'Elektor, l'Annex WiFi RDS a été décrite comme une plateforme de programmation en BASIC des microcontrôleurs ESP8266/ESP32. En voici un exemple pratique: le micrologiciel pour un sablier. Pas en C, ni en C#, ni en C++, mais rapidement, simplement et facilement en BASIC.

Figure 1. Le sablier terminé est doté d'une horloge numérique à quatre chiffres sur la gauche et un véritable sablier illuminé est tourné par un servo caché sur la droite.



Voici un projet pratique comme application de l'environnement de développement en BASIC appelé Annex WiFi RDS [2] décrit dans le dernier numéro d'Elektor [1]. Les différents modèles de modules ESP8266, disponibles pour quelques euros, offrent à l'électronicien des possibilités infinies pour des projets de plus en plus ambitieux, non seulement grâce à leur fonction WLAN, mais aussi parce qu'ils peuvent être facilement connectés sans fil à des téléphones tactiles, des tablettes et des PC. Divers capteurs et actionneurs peuvent être facilement connectés au microcontrôleur sans grands frais, de sorte que le bricolage pourrait redevenir un véritable plaisir si seulement il n'y avait pas le fichu obstacle de la programmation plutôt compliquée des SoC modernes.

Le choix est pourtant vaste entre langages de programmation, IDE, bibliothèques, pilotes, ressources et compilateurs ! Ou faudrait-il peut-être contourner l'obstacle en douce et récupérer du code écrit par d'autres ? Celui-ci est souvent difficile à comprendre et le transfert dans le module ne va pas de soi. Et si vous souhaitez commander votre projet avec un téléphone par l'intermédiaire d'une interface web, ça se complique. Sauf si adoptez l'environnement de développement Annex WiFi RDS programmé en BASIC. Vous devriez essayer, la réussite est rapide, avec zéro frustration.

Projet simple

Quelques lignes de BASIC suffisent pour réaliser une horloge numérique

avec ESP8266, qui affiche en même temps les prévisions météorologiques. Jugez-en vous-même par les exemples de projets sur le site de l'Annex [3]. J'y ai trouvé l'idée de combiner un affichage numérique et un affichage analogique : le projet *ESP8266 Sand Clock*, renverse un petit sablier toutes les minutes tout en donnant l'heure sous un format numérique conventionnel à quatre chiffres.

Ce n'est pas tout : le sable qui coule est éclairé par quatre LED NeoPixel pour la lumière de fond. La luminosité de l'afficheur doit être adaptée à la luminosité ambiante, grâce à une photorésistance (LDR) utilisée comme capteur de lumière. Si vous ajoutez un capteur de température à 1 fil, vous pouvez même afficher la température ambiante.

Ce sablier devra pouvoir être commandé par un réseau local sans fil (WLAN) grâce à l'interface web intégrée. Imaginez-vous le code qu'il faut pour ça ! Et la ribambelle de bogues qu'il faudra déboguer ? Le mot qui vient à l'esprit quand songe à tout ça commence et finit par *mañana...*

Heureusement il y a le BASIC ! Voyez donc le sablier fini de la **fig. 1** : il ne nécessite pas tant de logiciel qu'on pourrait le penser.

En parcourant le fichier d'aide de l'Annex WiFi RDS, détaillé et orienté vers la pratique, de même que le contenu du site associé, d'autres idées de projets inédits ont surgi. Le BASIC est à la fois puissant facile à maîtriser, même pour des programmeurs occasionnels. Comparé aux autres solutions courantes, l'Annex Rapid Development Suite simplifie considérablement le développement de logiciels.

L'électronique

La carte à μC D1-Mini utilisée offre un module ESP8266 et, grâce à ses broches, convient à l'expérimentation sur une plaque à trous, car toutes les connexions SoC pertinentes sont accessibles. Grâce à son interface USB et à son mode auto-flash, le micrologiciel peut être chargé très facilement et rapidement, sans intervention matérielle, à l'aide de l'Annex Toolkit [4]. N'oubliez pas de définir le fuseau horaire correct sur la page CONFIG et d'y saisir le script BASIC dans la ligne de démarrage automatique (le chemin de mon script actif est toujours «`/program/default.bas`»).

Pour mes tests, j'ai assemblé le circuit de la **fig. 2** sur une plaque d'expérimentation (**fig. 3**).

Si la tension de service est fournie par la prise micro-USB, Les tensions de +5 V et +3,3 V sont également accessibles de l'extérieur et peuvent être utilisées dans certaines limites du fait de la présence d'une diode Schottky et d'un régulateur de tension de 3,3 V. Le courant consommé par les périphériques utilisés ici ne dépasse pas ce que l'on peut attendre de ces lignes d'alimentation. Si vous y tenez, vous pouvez alimenter le servomoteur séparément sous +5 V. Les deux condensateurs électrolytiques entre +5 V et la masse lissent les pics de courant du servo.

Notez que les entrées d'un ESP8266 ne tolèrent pas de niveaux de plus 3,3 V. C'est pourquoi presque toutes les résis-

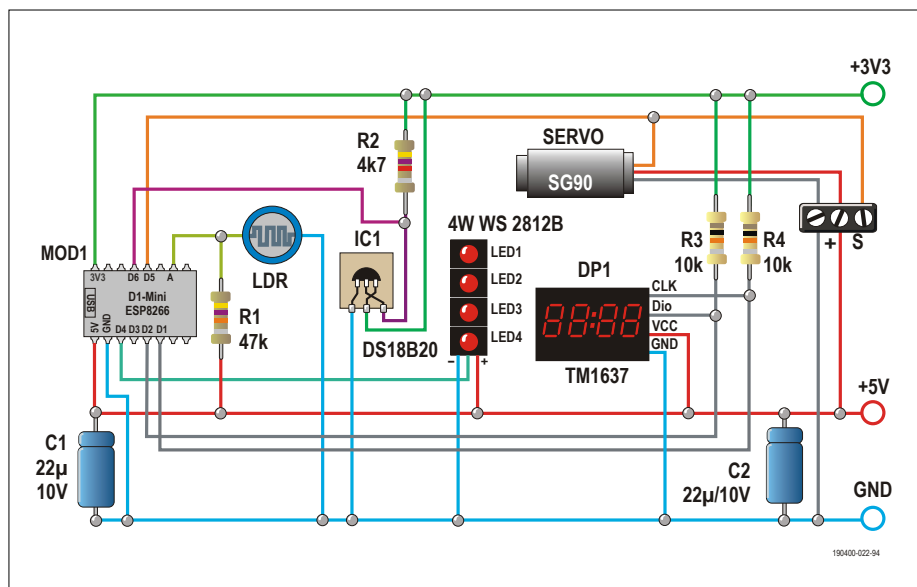


Figure 2. Le circuit du sablier n'est pas plus compliqué que ça : en plus du module ESP8266, quatre résistances, deux condensateurs, quatre LED, un capteur de température, un écran et un servo, c'est tout !

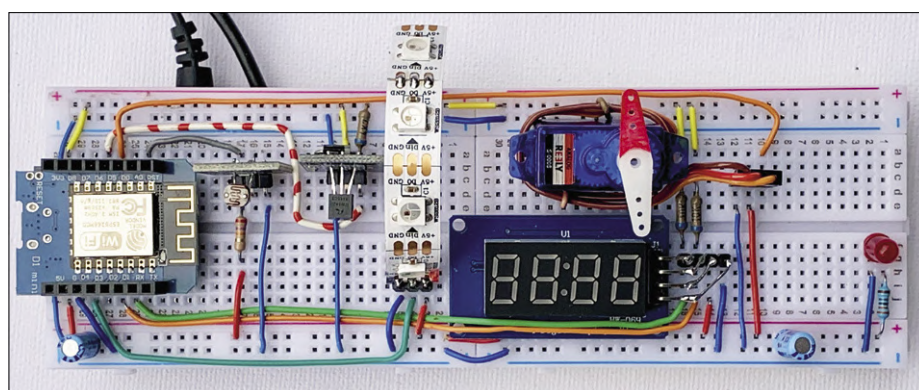


Figure 3. Le prototype du sablier sur une plaque d'expérimentation le circuit de la figure 2.

tances de polarisation externes sont connectées à +3,3 V. Grâce au diviseur de tension interne de l'entrée analogique, toute tension de 0 à 3,3 V est acceptée et même l'application de 5 V ne pose pas de problème. Le diviseur de tension de la photorésistance doit être dimensionné de manière à ce que, en pleine lumière, il règne au maximum 500 mV à l'entrée. Si l'entrée analogique reste déconnectée, la tension sera nulle ici et la luminosité de l'écran sera à son maximum. Attention lors du branchement du capteur de température DS18B20 ! Le brochage du capteur sur la carte est différent de celui du capteur en boîtier TO.

Logiciel BASIC

Une fois le micrologiciel installé sur

l'ESP8266 en utilisant le Toolkit d'Annex [4] et une fois les instructions [5] suivies, il suffit d'un navigateur pour accéder au serveur du module. Le nouveau code peut être rapidement écrit dans l'éditeur ou inséré dans *default.bas* par copier-coller.

Le fichier d'aide d'Annex (touche F2 dans l'éditeur) contient des explications, les brochages et des bribes de script pour l'afficheur à LED TM1637, le servo analogique, les LED NeoPixel et divers actionneurs et capteurs disponibles dans le commerce ainsi que pour les différents modules ESP. L'aide en ligne contient également une section utile sur les bases du langage.

Le **listage 1** présente le début du script BASIC utilisé. Le **listage 2** contient le

Listage 1. Début de `hourglass.bas`

```
##### HOURGLASS #####
' 4Digit digital clock combined with 10 minutes hourglass
VERSION$ = "V3.3"
' 10/2019 Peter.Neufeld@gmx.de min:ANNEX_1.39b6
' - ESP8266-Module: WEMOS-D1-Mini
' - Display: 4digit 7segment TM1637
' - Servo: analog servo (SG90)
' - Background light: 4 NeoPixel-LEDs
' - Temperature sensor: DS18B20
' - Light sensor: LDR @ analogue input
' GPIO-Mapping (PIN-Label) for WEMOS and NodeMCU boards
D0=16:D1= 5:D2= 4:D3= 0:D4=2
D5=14:D6=12:D7=13:D8=15:D9=3:D10=1
SERVO_PIN = D5 'Servo @ GPIO14=D5
TM_DATA = D2 'TM1637 Data @ GPIO4=D2
TM_CLOCK = D1 'TM1637 Clock @ GPIO5=D1
TM_BRIGHT = 7 'TM1637 brightness
TURN_TIME = 10 'Turn hourglass every xx minutes
SERVO_LEFT = 180 '1. Position of hourglass
SERVO_RIGHT = 0 '2. Position of hourglass
SERVO_POS = 0 'actual position of hourglass
BLINK = 0 '255= visible colon @ TM1637
ADC_DARK = 450 'max. ADC value for dark LDR,
' depends on LDR and Pull-up
LED1_STATUS = 0 'Background LEDs: 0=automatic, 1=manually
STATUS$ = "Modus: AUTOMATIK"
ADC_IN = 0:
```

Listage 2. Sous-programme pour une page web.

```
WEB_PAGE:
cls
a$ = "<center><h1> - S A B L I E R - "+ VERSION$ +" - </h1>"
a$ = a$ + "<br>" + textbox$(t$, "cssTB")
a$ = a$ + METER$(SS, 0, 60, "cssMET")
a$ = a$ + textbox$(TEMP$, "cssTB") + "<br><br>"
a$ = a$ + "<br>R: " + slider$(R, 0, 255) + textbox$(R, "cssTB")
a$ = a$ + "<br>G: " + slider$(G, 0, 255) + textbox$(G, "cssTB")
a$ = a$ + "<br>B: " + slider$(B, 0, 255) + textbox$(B, "cssTB")
a$ = a$ + "<br>" + LED$(LED1_STATUS)
a$ = a$ + "<br>" + textbox$(STATUS$)
a$ = a$ + "<br>" + BUTTON$("Mode auto/manuel", MAN_AUTO)
a$ = a$ + cssid$("cssTB", " width:70px;text-align:center")
a$ = a$ + cssid$("cssMET", " transform:rotate(-90deg);")
html a$
return
```

code de l'interface web délibérément simple (**fig. 4**).

Ce projet a rapidement compté environ 140 lignes BASIC. Comme il est amplement commenté, vous comprendrez un tel script même si vous ne l'avez pas écrit vous-même. Si vous le souhaitez, vous le prolongerez ou l'améliorerez dans quelque temps. Le code complet de cet article peut être téléchargé sur le site d'Elektor [6].

Au cours des manipulations du fichier *default.bas*, édition, sauvegarde (!), lancement, débogage, modification, extension, sauvegarde, nouveau lancement, etc., les erreurs courantes du code sont identifiées et marquées par l'interpréteur. Cela en facilite la correction. Le code BASIC est à la portée des programmeurs occasionnels, surtout si on prend soin d'attribuer aux variables des noms qui parlent d'eux-mêmes et si on a recours à des sous-programmes. Le code BASIC est facile à lire. Le codage graphique de la syntaxe améliore la lisibilité du code et en facilite la vérification. En cas de problème de syntaxe, la touche F2 permet d'effectuer une recherche contextuelle dans le fichier d'aide en ligne. Une petite coche verte dans la fenêtre de l'éditeur mène aussi au contrôle syntaxique.

Pour commencer, le scénario BASIC est assez linéaire. Après initialisation de certaines variables et un court mouvement du servo du sablier, l'adresse IP du serveur Annex est affichée par portions. C'est commode lorsqu'on accède au serveur avec un navigateur, surtout si le routeur a attribué une adresse IP différente.

La routine principale est exécutée une fois par seconde au moyen d'un temporisateur. Heures et minutes sont préparées pour l'affichage, puis sont exécutées successivement les sous-programmes de commande de lumière et du rétroéclairage, l'affichage de l'heure et de la température, et enfin, toutes les dix minutes, la servocommande.

J'ai abandonné l'idée initiale de tourner le sablier une fois par minute à cause du bruit que fait un servo de modèle réduit dans un salon silencieux. J'ai opté pour un sablier de dix minutes, qui tourne le verre lentement pendant environ quatre secondes et donc plus silencieusement et moins fréquemment. Si vous voulez, vous pouvez utiliser un sablier plus rapide pour

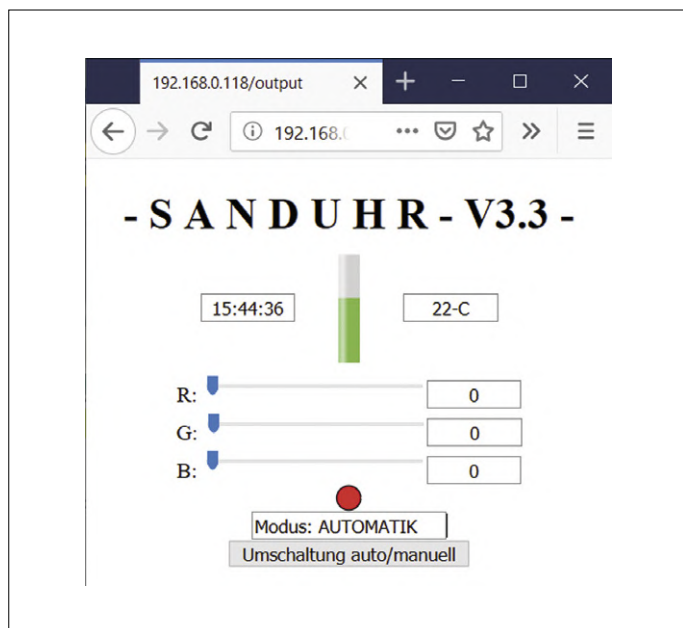


Figure 4. Un peu de code BASIC suffit pour créer une interface web simple.

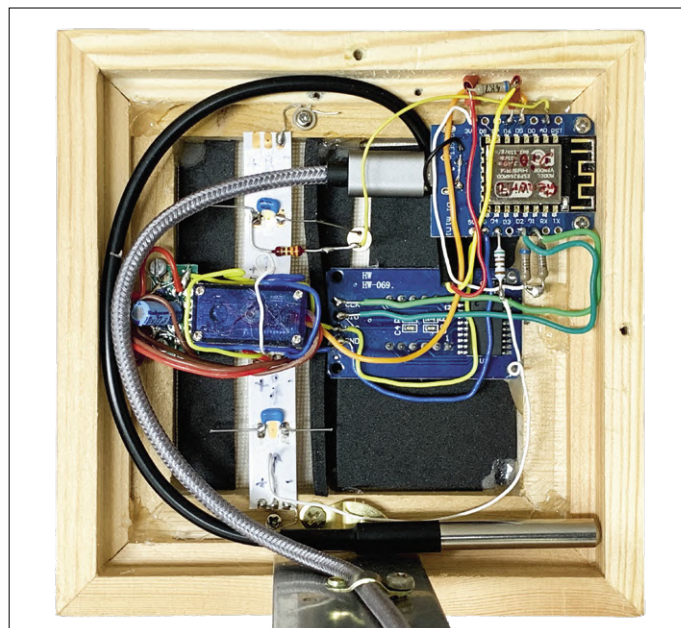


Figure 5. Les coulisses de mon sablier électronique ou les joies du bricolage en montage volant !

obtenir par exemple un rythme d'une ou cinq minutes. Il suffit de modifier la variable `TURN_TIME`.

Interface web

Lorsqu'un navigateur accède pour la première fois ou à nouveau, ou que les valeurs des variables utilisées changent, le contenu HTML de la page web ESP8266 est préparé et mis à jour dans le sous-programme `WEB_PAGE` (listage 2). L'interface affiche l'heure et la température actuelles. La simple animation d'arrière-plan à LED derrière le sablier est remplacée par une sélection manuelle de couleurs RVB par bouton-poussoir. En mode automatique, les contrôleurs RGB et les champs d'affichage suivent en direct les valeurs de l'animation calculées en continu.

Le site vise uniquement à montrer comment le micrologiciel d'Annex assure la communication bidirectionnelle avec le navigateur en arrière-plan. Grâce à l'utilisation par Annex de CSS (Javascript, AJAX et les sockets web sont également pris en charge), il n'y a pratiquement pas de limites aux expériences de communication avec (presque) tous les navigateurs. On en trouve également de nombreux exemples sur le site web d'Annex - notamment dans les fichiers qui peuvent être copiés du Toolkit d'Annex dans le dossier «`/program`» du module ESP lors de l'installation du

micrologiciel. Cela vaut la peine de fouiller ! Si vous trouvez un script intéressant, vous pouvez l'insérer grâce à la commande `Ctrl+A` (= tout sélectionner) suivie d'un copié-collé dans la fenêtre de l'éditeur d'Annex avec un éditeur de texte (p. ex. *GEANY.exe*), puis le sauvegarder et le lancer rapidement dans *default.bas* ! Des sections complètes du script peuvent être rapidement converties en lignes de commentaires à partir du menu et vice versa.

Emménager

Mon circuit dûment a finalement emménagé en montage volant dans un petit cadre en bois, recouvert à l'avant d'un écran translucide (**fig. 5**). Seuls les segments actifs et lumineux de l'affichage à LED sont visibles. Le capteur de lumière (LDR) reçoit suffisamment de lumière ambiante. L'axe du servo qui fait tourner le sablier passe par un petit trou

dans l'écran. Des plaques de mousse ou du papier noir découpé à la bonne taille permettent de limiter la transparence de l'écran aux seules zones nécessaires, derrière le sablier et sur la LDR. On peut également utiliser d'autres types de cadres avec façade en plexiglas (comme on en trouve au catalogue d'un célèbre magasin de meubles en kit). La face transparente sera recouverte à l'intérieur par du papier translucide ou une photo adéquate. À vous de bricoler ! ◀

190400-B-02

Liens

- [1] BASIC pour ESP32/ESP8266 : www.elektormagazine.fr/190400-02
- [2] Annex WiFi RDS : <https://sites.google.com/site/annexwifi/home>
- [3] Exemples de projets : <https://sites.google.com/site/annexwifi/projects>
- [4] Toolkit : <https://sites.google.com/site/annexwifi/downloads>
- [5] Instructions : <https://sites.google.com/site/annexwifi/home/first-steps>
- [6] Page du présent article : www.elektormagazine.fr/190400-B-02

bureau d'études – Zone D

D comme développement, débrouille et dur-à-cuire !

Trucs et astuces, bonnes pratiques et autres informations utiles

Clemens Valens (Elektor Labs)

DE L'IDÉE AU PRODUIT

5^E PARTIE

Il semble que la conception de nos nouveaux produits progresse assez lentement ici, mais il est peu probable que dans la vraie vie sa progression soit plus rapide. La création d'une entreprise prend du temps. Le précédent épisode nous a laissé une nomenclature (BoM) et une *net liste*, ce qui nous permet de commencer à travailler sur un ou plusieurs circuits imprimés, selon les besoins. La complexité de la conception des PCB dépasse le cadre de cette rubrique ; sur ce sujet, je vous recommande cet autre article [1].

COMMENCER PAR LA FIN

L'une des premières étapes de cette série d'articles a été la maquette de notre futur produit, à partir de laquelle nous avons conçu les circuits qui la composent, puis nous avons sélectionné les composants adéquats. Nous allons maintenant ajouter un ou plusieurs circuits imprimés qui eux aussi devront s'adapter. Avant de commencer à placer des composants et à imaginer un possible tracé des pistes, il faut réfléchir au contour de la carte, aux trous de montage ou à d'autres moyens de fixation à l'intérieur de son boîtier. Souvent, on commence par un contour rectangulaire, dans lequel on serre les composants les uns contre les

autres autant que le routage le permet, pour rétrécir l'encombrement de la carte. Puis c'est la gravure du PCB, l'implantation, le test, et l'installation dans le boîtier. On constate alors qu'il n'y a pas ou pas assez de trous de fixation, ou qu'ils ne sont pas au bon endroit ou pas du bon diamètre ; que les connecteurs sont du mauvais côté ou que le radiateur ne rentre pas. Bref, c'est un cauchemar mécanique. Il faut redessiner le PCB. La plupart de ces problèmes auraient pu être évités en posant le futur boîtier sur le bureau du concepteur du circuit imprimé avant de commencer.

PROFITEZ DE VOTRE LIBERTÉ

La conception des PCB combine diverses disciplines, notamment l'électronique et la mécanique : vous avez à mettre au point une pièce mécanique dotée de propriétés électroniques particulières. En tant que concepteur, vous en avez le contrôle (presque) total. Exploitez toutes les possibilités offertes par l'étude des circuits imprimés. Le PCB peut être une partie du boîtier, il peut être en aluminium, il peut être souple, il peut prendre presque n'importe quelle forme. Que la plupart des PCB que vous voyez (dans Elektor) soient rectangulaires n'implique nullement qu'il faille faire pareil.



Le boîtier du légendaire détecteur TAPIR d'Elektor est fait uniquement de PCB.

CONCEVOIR POUR PRODUIRE

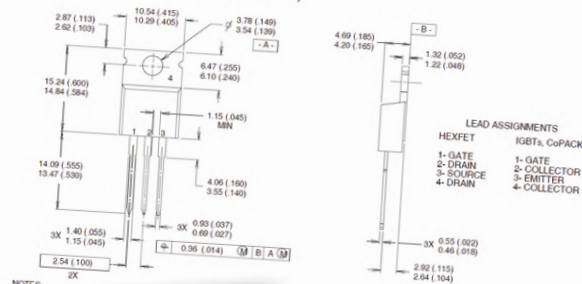
En dessinant une nouvelle empreinte, il est tentant de prévoir une marge de sécurité dans les dimensions : une pastille ou un trou un poil plus grand que prévu, ce genre de choses. Une bonne idée à première vue, mais vous risquez des tracas lors de l'assemblage automatisé. Les pastilles de CMS surdimensionnées peuvent faire dériver les composants, voire les faire pivoter lors du soudage par refusion. Un excès de pâte à souder peut provoquer des courts-circuits. Une broche traversante dans un trou trop grand risque d'être mal soudée à la vague.

FAITES-VOUS CONFIANCE À VOS COMPOSANTS ?

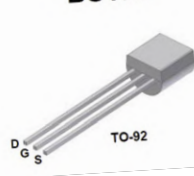
L'incompatibilité mécanique est un problème fréquent des PCB, les mauvaises empreintes en sont un autre. Il ressemble aux tracas que nous avons décrits pour la capture du schéma. Tant que vous vous en tenez aux composants passifs en boîtiers 0805 et que vous n'utilisez que des transistors et des diodes SOT-23, tout ira probablement bien, mais gare à vous s'il manque ne serait-ce qu'une seule empreinte inconnue des bibliothèques de votre logiciel de CAO. Méditez les sages propos du précédent épisode : *Les créer soi-même ? Ce serait peut-être plus vite fait, mais aussi plus risqué d'aller les glaner sur l'internet. Les acheter auprès d'un fournisseur de schémas de composants pour PCB ? Dans tous les cas, vérifiez soigneusement chaque composant avant de vous y fier. Idem pour votre bibliothèque standard (qui elle aussi peut contenir des erreurs).* Outre le téléchargement de la fiche technique de chaque composant de votre étude, il est fortement recommandé

TO-220AB Package Outline

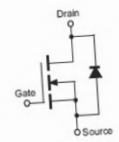
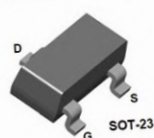
Dimensions are shown in millimeters (inches)



BS170



MMBF170



Certaines fiches techniques sont des modèles de confusion. Est-il donc si difficile de rester limpide ?

d'obtenir des échantillons de chaque composant avant de commencer le tracé de la carte. Les fiches techniques peuvent être étonnamment obscures : diamètre (broche ou trou ?), vue (de dessus ou de dessous ?), unités (pouces ou millimètres), numérotation des broches, etc. Sans parler des erreurs flagrantes. Tenir le composant entre les mains vous permettra de lever le doute.

PARLEZ AVEC L'ASSEMBLEUR

Le placement de certains composants risque aussi d'entraver l'assemblage. En raison de leurs exigences thermiques contradictoires, le voisinage d'un gros composant métallique et d'un minuscule composant en plastique peut entraîner une mauvaise soudure du gros, ou endommager le petit. La position de certains composants pourrait être incompatible

avec l'automate d'implantation, il faudrait peut-être des languettes détachables (*snap-off tabs*) au PCB. La plupart de ces problèmes peuvent être résolus par les professionnels de l'assemblage. Les consulter *avant* de finaliser votre carte permettra éventuellement de leur faciliter la tâche, d'augmenter la fiabilité et de réduire vos coûts.

COÛTEUSES ITÉRATIONS DE PCB

La première version de votre PCB nécessitera probablement des modifications. Pour ne pas vous y perdre dans les itérations, marquez et étiquetez-les bien. N'oubliez pas la mise à jour du numéro de version sur le PCB à chaque modification. Prenez des notes des changements et de leurs causes. À mesure qu'augmentera votre confiance dans le produit, vous serez tenté d'en augmenter les quantités. Bravo pour la réduction du coût, mais gare aux modifications éventuelles. L'élimination de dix prototypes est probablement moins coûteuse que le remplacement d'une résistance erronée sur 1.000 cartes produites en série. Surtout si vous ne vous rappelez pas bien les raisons de la modification de cette valeur de résistance...

Lien

[1] [1] concevoir un circuit imprimé ? pas si facile : www.elektormagazine.fr/160397elektor-201707/40522

MÉFIEZ-VOUS DES EMPREINTES EN MIROIR !

L'empreinte en miroir est une faute si fréquente qu'il fallait que j'en parle. Vous avez lu le titre de ce paragraphe, il n'y a rien à ajouter. Ne faites plus cette faute, point. Fin de paragraphe.

ET LA PROCHAINE ÉTAPE ?

Notre produit est-il prêt ? Presque, nous avons quelques prototypes à tester. Des tests ? Eh oui. Tester le fonctionnement, la certification, la qualité – il reste en effet beaucoup de tests à faire. Comment ? Quelles en sont les implications ? Cette série d'articles ne finira-t-elle donc jamais ?

190369-C-01

Thermomètre à grillon et loi de Dolbear



Ce que vous voyez sur cette photo n'est pas un *Oecanthus fultoni*, mais une sauterelle. Chez Elektor, on s'y connaît en puces, mais pas en grillons...

Au risque de me répéter, la température est sans doute la grandeur physique la plus observée au monde. Il existe plusieurs méthodes pour la mesurer. En électronique, les plus courantes sont :

- **thermistance** : composant dont la résistance augmente (coefficient de température positif ou CTP) ou diminue (CTN) avec la température ;
- **détecteur de température à résistance** (*Resistance Temperature Detector* RTD) : exploite la relation précisément connue entre la température et la résistance d'un matériau comme le platine ou le cuivre ;
- **thermocouple** : jonction de deux métaux différents qui engendre une tension qui varie selon la température par effet thermoélectrique (Seebeck) ;
- **capteurs à semi-conducteurs** : la tension aux bornes d'une diode ou d'une jonction PN diminue d'environ 1 à 2 mV par degré Celsius ;
- **thermomètre à infrarouge** : permet de déduire la température à partir du rayonnement thermique.

Cette dernière technique est souvent utilisée pour mesurer la température du corps humain en pointant un *pistolet à température* à l'intérieur d'une oreille. Rien de nouveau, direz-vous, en réprimant discrètement un bâillement. Mais saviez-vous que pour déterminer la température vous pouvez utiliser votre ouïe elle-même ?

La fréquence de la stridulation de certains grillons est proportionnelle à la température. Y auriez-vous pensé ? Tout comme on peut estimer la distance d'un orage en comptant les secondes entre un éclair et le tonnerre qui suit, la température de l'air peut être estimée d'après le nombre de stridulations par minute. Cette corrélation a été signalée en 1881 par Margarette W. Brooks, et c'est Amos Dolbear qui publiait en 1897 l'article « *The Cricket as a Thermometer* ». Depuis, le grillon arboricole vert (*Oecanthus fultoni*) est connu sous le sobriquet de grillon thermomètre. Le calcul de la température T à partir des stridulations est facile, surtout en degrés Fahrenheit :

$$T = N_{15} + 40 \text{ [}^{\circ}\text{F]}$$

Ici, N_{15} est le nombre de stridulations en 15 secondes. Converti (et arrondi) en degrés Celsius, cela donne :

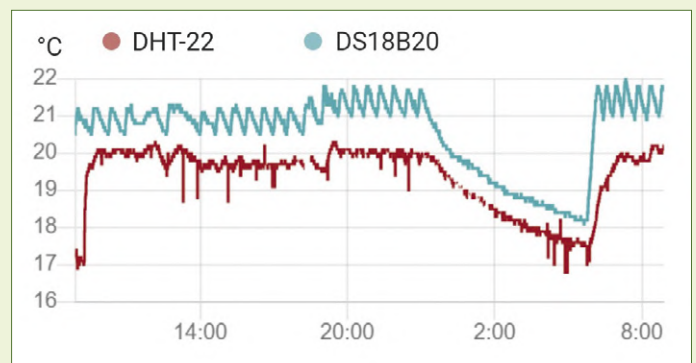
$$T = N_8 + 5 \text{ [}^{\circ}\text{C]}$$

Ici, N_8 est le nombre de stridulations en 8 secondes. Le thermomètre à grillon est assez précis entre 5 et 30 °C.

Émulateur de grillon

Pour rigoler, avec un Arduino Uno, un résonateur et un capteur de température à un fil DS18B20, j'ai inversé le processus et imité les stridulations d'un grillon variant selon la température ambiante. À première vue, ce projet paraît stupide et c'est comme ça que tout a commencé. Cependant, une fois mis en marche, j'ai été frappé par la sensibilité de l'ouïe aux *changements de rythme*. À tel point que je considère maintenant cet indicateur de température comme utile dans des circonstances où il est mal pratique de garder un œil sur un thermomètre.

Les détails du projet vous attendent à l'adresse : www.elektor-labs.com/3945



Les capteurs de température ne naissent pas égaux. Ce relevé journalier compare les températures mesurées par un DHT-22 et un DS18B20 en même temps au même endroit.

microcontrôleurs TMS1000

drôle(s) de composant(s)

Neil Gruending (Canada)

Les microcontrôleurs aujourd'hui omniprésents jusque dans la plus modeste lampe de poche constituent un marché de plusieurs milliards de dollars. Leurs débuts ont été modestes. Pour les obtenir, il a fallu condenser plusieurs puces LSI en une seule. Examinons certains des premiers μ C : la série TMS1000 de Texas Instruments.



Figure 1. Microcontrôleur TMS1000 sur une seule puce. Image : Christian Bassow, Wikimedia Commons, CC BY-SA 4.0. https://upload.wikimedia.org/wikipedia/commons/d/d1/TI_TMS1000NP_1.jpg

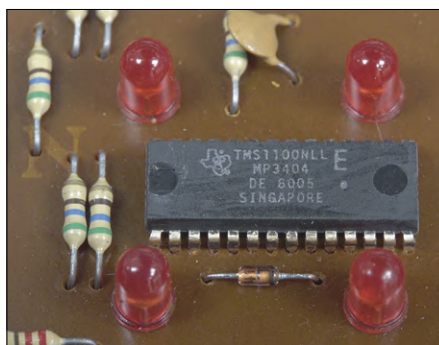


Figure 2. TMS1100 dans le jeu électronique Merlin de Parker Brothers. Image : Binarysequence. Wikimedia Commons, CC BY-SA 4.0. <https://upload.wikimedia.org/wikipedia/commons/a/ad/Tms1100-merlin.png>

accessible ainsi qu'un indicateur de retenue (*carry flag*) entre le programme principal et un seul niveau de sous-programme. Cinquante ans après, c'est bien peu de choses au regard des normes actuelles, mais la série TMS1000 était conçue délibérément comme système fermé à puce unique, sans UART ni autre interface.

Environ 100 millions de puces de la série TMS1000 ont été produites en une quarantaine de variantes, utilisées principalement dans des produits électroniques grand public comme les calculatrices, les jeux et les appareils électroménagers. Pour dénicher aujourd'hui un de ces composants pionniers de la série TMS1000, il faut dépiapter un de ces appareils si vous en trouvez un. ◀

190383-B-04

Les microprocesseurs sont apparus dans les années 1970. Ces μ P polyvalents allaient permettre de mettre en œuvre des fonctions complexes sans puces logiques conçues sur mesure. Certes puissants pour l'époque, ces μ P présentaient l'inconvénient de ne vivre qu'en troupeau : mémoire vive (RAM), mémoire morte (ROM), périphérie, horloge... c'était encombrant et cher. Texas Instruments cherchait pour une nouvelle calculatrice une solution économique pour laquelle Gary Boone et Michael Cochran ont conçu le TMS1802 (1971). Une seule puce regroupait toute la logique nécessaire à une calculatrice à quatre fonctions, sauf le clavier et l'écran, avec seulement 3000 bits de mémoire de programme et bits de mémoire vive.

En 1974, *Texas Instruments* commercialisait le premier microcontrôleur de la série TMS1000, une puce qui réunissait 1024×8 bits de ROM, 64×4 bits de RAM, un microprocesseur et une horloge. Sa mémoire morte n'était pas modifiable ;

TI proposait donc un TMS1000 sans ROM, utilisable pour mettre au point le programme en mémoire vive, qui servirait ensuite à produire la ROM masquée de la puce finale. Le coût initial restait élevé, mais pas les coûts de production en grandes quantités. Le format des données du TMS1000 était de 4 bits, avec des bus de données de ROM et de RAM séparés comme c'est encore le cas aujourd'hui des μ C à architecture Harvard. Un compteur ordinal de 6 bits supportait une ROM de 2048×8 bits max. L'unité centrale n'acceptait pas d'interruptions, mais le registre du compteur ordinal était

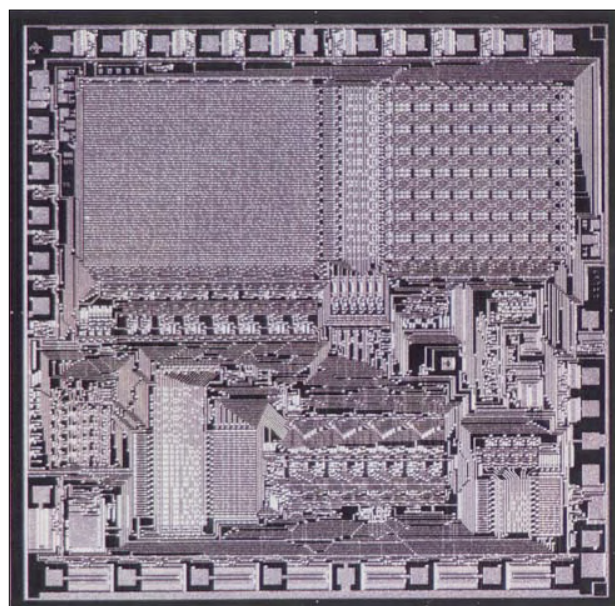
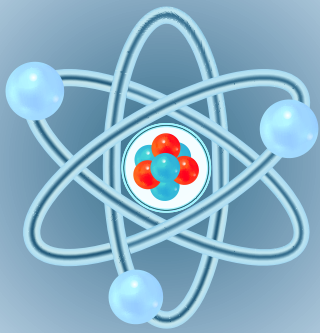


Figure 3. La puce du TMS1000. Source : *State of the Art*, par Stan Augarten. <http://smithsonianchips.si.edu/augarten/i38.htm>



démarrer en électronique (3)

... est plus facile qu'on ne l'imagine !

Eric Bogers (Elektor Pays-Bas)

Dans l'épisode précédent, nous nous sommes familiarisés avec la loi d'Ohm et avons fait quelques calculs élémentaires. Maintenant, cela devient un peu plus compliqué - mais n'ayez crainte : nous n'avons pas besoin de mathématiques supérieures...

Tout d'abord, une petite mise au point. Nous mentionnions dans le précédent épisode le fait que le préfixe servait souvent de point décimal : 8,2 kΩ devient 8k2 ou 8kΩ2 et 5,6 nF devient 5n6 ou 5nF6. Idem pour le préfixe μ (le *mu* grec pour "micro" ou 10⁻⁶) que beaucoup d'électroniciens remplacent par un u, car ils ne trouvent pas le signe μ sur leur clavier (ALT+0181) ou parce qu'ils le jugent malcommode : 4,7 μF = 4,7 uF = 4uF7 ou 4u7.

Résistances en série

Lorsque nous connectons deux résistances en série (**fig. 1**), que se passe-t-il ? Si nous voyons la résistance comme un composant qui s'oppose à la circulation du courant, nous pourrions raisonner comme suit : le courant est freiné dans une certaine mesure par R1, et la quantité réduite de courant qui passe par cette résistance est à nouveau freinée, dans une certaine mesure, par R2. Intuitivement, lorsque l'on connecte des résistances en série, la résistance totale semble augmenter. C'est peut-être bancal, mais ce n'est pas faux.

Voyons maintenant ce qui se passe exactement. La loi d'Ohm nous dit que la tension U aux bornes de la connexion

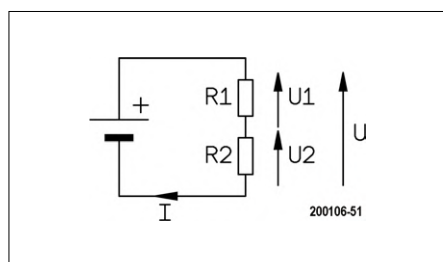


Figure 1. Deux résistances en série.

en série de R1 et R2 (que nous appelons R) est égale à

$$U = R I$$

Les résistances n'ont rien de mystérieux. Il ne s'y perd rien. Ainsi, l'intensité du courant I dans le circuit en série est-elle égale à celle du courant sortant du circuit en série. L'intensité du courant est donc la même dans les deux résistances, il y circule le même courant. Aux bornes de chacune des résistances, il se perd une certaine tension (une chute de tension régie par la loi d'Ohm). Et comme les résistances sont passives, aucune tension ne s'ajoute en route. De sorte que la somme des deux tensions partielles $U1$ et $U2$ est égale à la tension totale U . Nous pouvons écrire :

$$U1 = R1 I$$

$$U2 = R2 I$$

$$\text{Et comme } U1 + U2 = U :$$

$$U = (R1 I) + (R2 I) = (R1 + R2) I$$

Et il s'ensuit que la résistance totale R est égale à la somme des résistances R1 et R2. Vous l'aviez deviné, maintenant vous pouvez en être sûr. Prenez donc la peine de vérifier ce qui précède au moyen d'un exemple numérique, p. ex. une tension de batterie de 15 V, R1 = 10 Ω et R2 = 20 Ω.

Résistances en parallèle

Si au lieu de connecter les résistances en série nous les mettons en parallèle ("côte à côte") comme sur la **fig. 2**. Faisons d'abord comme s'il n'y avait pas R2 ; un certain courant circule à travers

R1. Ajoutons R2 ; un certain courant circule à travers elle aussi. Le courant total augmente ; et comme la tension aux bornes des résistances est restée la même, cela implique que la résistance totale de R1 et R2 en parallèle est devenue plus petite !

Deux choses sont claires sur la fig. 2 : sur chacune des résistances R1 et R2 règne la même tension U . Le courant total I est par conséquent la somme des courants à travers les deux résistances. Puisque ce sont des composants passifs, donc aucun courant ne disparaît ni ne s'ajoute en route.

Si on appelle R la résistance totale du circuit parallèle, on en est toujours à :

$$U = R I$$

Appliquons à présent la loi d'Ohm aux résistances individuellement :

$$U = R1 I1 \quad I1 = U / R1$$

$$U = R2 I2 \quad I2 = U / R2$$

Avec $I1 + I2 = I$, on écrit

$$I = (U / R1) + (U / R2) = U / R$$

et de cela :

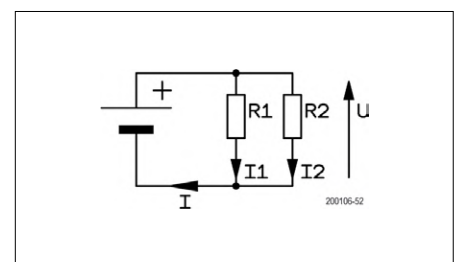


Figure 2. Deux résistances en parallèle.

$$1 / R = (1 / R_1) + (1 / R_2)$$

Que l'on peut réécrire ainsi :

$$R = 1 / [(1 / R_1) + (1 / R_2)]$$

Ici aussi, nous vous invitons le lecteur attentif à vérifier ces calculs avec des valeurs de tension et de résistance réelles.

Circuit mixte

En pratique, nous n'avons pas seulement des circuits "purs", soit en série, soit en parallèle, mais aussi à des "circuits mixtes" (fig. 3). On y voit une connexion en série d'une seule résistance (R_1) avec une connexion en parallèle de deux résistances ($R_2 || R_3$). (Note : cette dernière configuration est symbolisée par deux barres parallèles $||$).

Prenons un exemple numérique :

$R_1 = 10 \Omega$, $R_2 = 20 \Omega$ et $R_3 = 30 \Omega$. Il serait possible d'écrire une seule relation pour la résistance totale, mais une telle approche est susceptible d'entraîner des erreurs et des fautes de calcul. Il est préférable de calculer la résistance totale en deux étapes : d'abord le circuit parallèle, puis le circuit en série. Pour le circuit parallèle de R_2 et R_3 , on écrit :

$$1 / R_{\text{parallèle}} = (1 / R_2) + (1 / R_3)$$

Saisissons les valeurs de résistance de l'exemple :

$$1 / R_{\text{parallèle}} = (1 / 20 \Omega) + (1 / 30 \Omega) = (3 / 60 \Omega) + (2 / 60 \Omega) = 5 / 60 \Omega$$

Et inversons : $R_{\text{parallèle}} = 60 \Omega / 5 = 12 \Omega$. Cette résistance est connectée en série avec R_1 . La résistance totale R de ce circuit mixte est $R = 10 \Omega + 12 \Omega = 22 \Omega$.

Quelques remarques

Vous pouvez maintenant vérifier tout ça par vous-même. Les règles empiriques suivantes s'appliquent pour les estimations au jugé :

- Avec les résistances connectées en série, la résistance totale est toujours supérieure à la plus grande des résistances en série ;
- avec des résistances connectées en parallèle, la résistance totale est toujours inférieure à la plus petite des résistances connectées en parallèle.

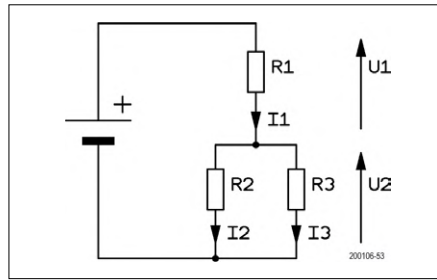


Figure 3. Circuit mixte simple.

Par exemple, si nous avons une connexion en série d'une très grande et d'une très petite résistance (disons $100 \text{ k}\Omega$ et 10Ω), dans la pratique quotidienne de l'électronique, nous ne cherchons même pas à calculer cela exactement, mais disons simplement que la résistance totale est égale à la plus grande résistance, donc $100 \text{ k}\Omega$. Et avec une connexion en parallèle des mêmes résistances, nous nous épargnons aussi la peine du calcul et disons que la résistance totale est égale à la plus petite résistance, donc 10Ω . La divergence de cette estimation par rapport aux valeurs réelles est négligeable.

Et une autre remarque avant de conclure : nous nous sommes limités ci-dessus à la connexion en série et en parallèle de deux résistances, mais tout ce que nous avons raisonné et calculé vaut aussi pour la connexion en série et en parallèle de plus de deux résistances :

- connexion en série de n résistances :
 $R_{\text{série}} = R_1 + R_2 + \dots + R_n$
- connexion parallèle de n résistances :
 $1 / R_{\text{parallèle}} = (1 / R_1) + (1 / R_2) + \dots (1 / R_n)$

Retour à la figure 2. Nous avons constaté que la somme des courants I_1 et I_2 est égale au courant total I . Cela semble aller de soi, mais en physique (dont l'électronique est une manifestation) il faut des preuves irréfutables. C'est le physicien Gustav Kirchhoff qui a établi en 1845 la loi qui porte son nom et stipule que la

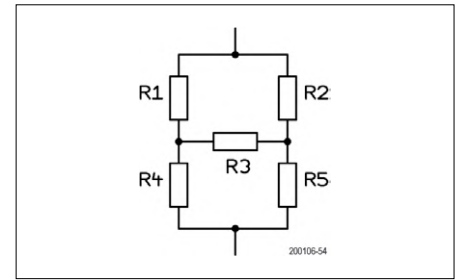


Figure 4. Le circuit en H, redouté dans les examens.

somme des courants dans un nœud est égale à la somme des courants provenant de ce même nœud.

Le circuit H – un avant-goût

Les connaissances acquises nous permettent de calculer aisément des circuits mixtes simples comme ceux de la fig. 3, mais regardez la **fig. 4** : un circuit mixte, en H, d'un genre plus délicat. Pas moyen de le décomposer par traiter séparément des circuits en série ou en parallèle. Comment allons-nous nous y prendre ? Voilà de quoi meubler vos insomnies en attendant le prochain épisode. Où il sera question d'étoiles et de triangles et de transformation. Et puis nous aborderons les courants et les tensions alternatifs. ◀

200106-03



@ WWW.ELEKTOR.FR

→ Basic Electronics for Beginners

www.elektor.fr/basic-electronics-for-beginners

→ L'électronique pour les débutants

www.elektor.fr/l-electronique-pour-les-debutants



le retour des petits circuits

... et des bonnes petites pétites d'Elektor

Eric Bogers (Elektor Pays-Bas)

Nos petits circuits évitent les CMS et les outils spéciaux et peuvent être construits sur des plaques d'expérimentation. Ils nous incitent à l'expérimentation et à l'improvisation, et peuvent devenir le tremplin d'une carrière en électronique. Voici une sélection de la boîte à suggestions d'Elektor.

Idée : Michael A. Shustov (Russie) & Andrey M. Shustov (Allemagne)

Relais photosensible à haut rendement

Dans le premier épisode de cette série (Elektor janv.-févr. 2020), nous avons décrit quelques variations sur le thème du « relais éco-énergétique » de l'équipe Shustov. Il s'agit en fait de variantes du fameux circuit d'interrupteur de cage d'escalier. Les Shustov sont de retour : la **fig. 1** montre une version complétée par un phototransistor, pour que la lampe ne s'allume pas seulement quand on appuie sur l'un des boutons SB1 à SBn, mais pour qu'elle puisse également être allumée à distance à l'aide d'une torche.

Si nous omettons T3 un instant, le circuit fonctionne comme indiqué dans le premier épisode. Au repos, les poussoirs SB1 à SBn connectés en série sont tous fermés (pour simplifier, nous n'en dessinons qu'un seul). Un faible courant peut circuler dans la lampe à travers R1 et la LED. Celle-ci s'allume pour indiquer que le circuit est armé. Pendant ce temps, le condensateur C1 est chargé à travers D1. Cependant, l'intensité du courant qui traverse la lampe est insuffisante pour qu'elle s'allume. Il s'agit là d'une petite lampe à incandescence avec un filament sous bulbe de verre.

Dès qu'un bouton est enfoncé, il ne circule plus de courant dans la lampe et la base de T1 est mise à la masse à travers R3, R2, R1 et la LED1. Ce transistor se met donc à conduire et C1 se décharge à travers R4. La tension aux bornes de R4 devient suffisamment élevée pour que T2 conduise à son tour. Dès lors, si le bouton est relâché, l'intensité du courant à travers la lampe et T2 la fait s'allumer.

Cet état est stable tant que la tension aux bornes de R4 est suffisante pour que T2 continue de conduire. Une fois que C1 s'est suffisamment déchargé pour que T2 s'éteigne, l'état initial est rétabli ; après un court instant, C1 est rechargé à travers D1 et le circuit est réarmé.

T3 a la même fonction que les poussoirs : dans l'obscurité, ce phototransistor est bloqué : le circuit est au repos. Lorsque la lumière frappe le transistor (p. ex. le faisceau d'une torche), il conduit et force la base de T1 à la masse. Le circuit se comporte ensuite précisément comme si on avait appuyé sur l'un des boutons.

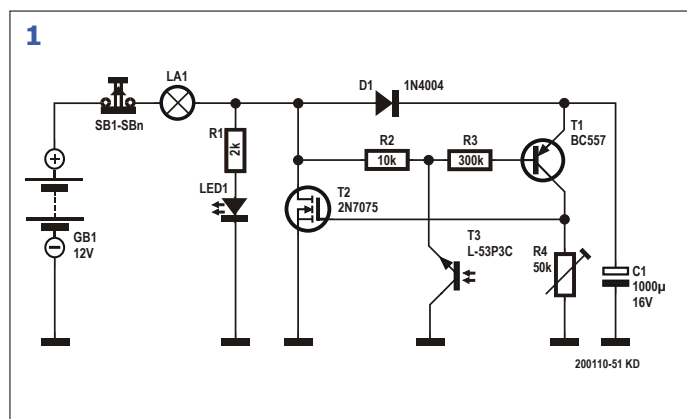
Ce circuit convient p. ex. pour une cave sombre. Plus besoin de chercher l'interrupteur à tâtons. Braquez la torche, la lumière s'allume....



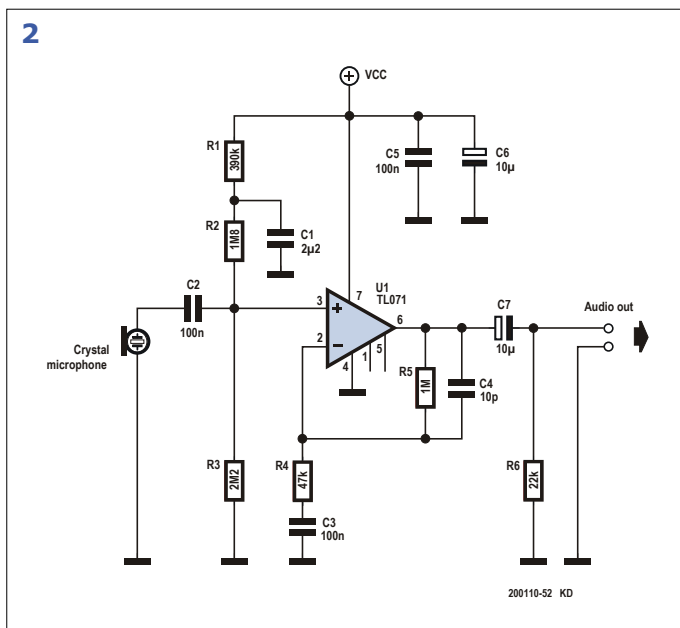
Idée : Joseph Kreutz (France)

Préamplificateur pour cellule piézo de tourne-disque

Vous souvenez-vous de ces consoles en bois des années 1970 ? Ces meubles imposants arboraient une radio incluant un amplificateur (encore à tubes) avec de gros boutons de sélection mécanique, sur lesquels il fallait appuyer fort, des gros boutons de volume et d'accord de part et d'autre d'un



2



panneau de verre éclairé portant le nom des stations, et, toujours sur le même meuble, un tourne-disque (on disait *pick-up* pour être dans le vent) pour disques phonographiques 33, 45 et 78 tours par minute ?

Sur ces successeurs de l'antique gramophone à pavillon, la courroie de transmission était rare, et l'entraînement direct restait à inventer. Le plateau était mis en rotation par un galet d'entraînement monté sur l'arbre du moteur. Pour transformer en électricité les vibrations mécaniques prélevées sur le sillon du disque phonographique, ce tourne-disque était le plus souvent équipé d'une cellule piézoélectrique reliée à une aiguille au bout de laquelle se trouvait un minuscule saphir. Leur tension de sortie était relativement élevée, mais ne pouvait pas être chargée avec une faible impédance. Pas de problème pour l'électronique à tubes de l'époque, mais sur les amplis transistorisés qui ont suivi, l'entrée dite « phono » était caractérisée par une impédance d'entrée normalisée de 47 kΩ, beaucoup trop faible pour une cellule phono. Les amplis modernes sont conçus pour des niveaux de signal *phono* de l'ordre de quelques millivolts seulement.

Pour pouvoir expérimenter avec de vieux tourne-disques, il nous faut donc un préampli/convertisseur d'impédance, facile à réaliser (fig. 2). Le composant central est un ampli op à FET à haute impédance TL071, un ancien qui a fait ses preuves et reste disponible. Avec R1, R2 et R3, l'impédance d'entrée est adaptée le mieux possible à la cellule de lecture. L'ampli op est non inverseur, C3 et C4 limitent sa bande entre environ 33 Hz et un peu moins de 16 kHz. Le gain est de l'ordre de 21, ce qui permet de le connecter directement à l'entrée CD d'un ampli moderne.

Ce circuit convient aussi aux micros piézoélectriques fournis autrefois avec les magnétophones.

Idée : Wolfgang Borst (Allemagne)

Surveillance externe des systèmes de microcontrôleurs

Un microcontrôleur qui fonctionne jour et nuit toute l'année peut se verrouiller pour une raison quelconque. Cela n'arrive

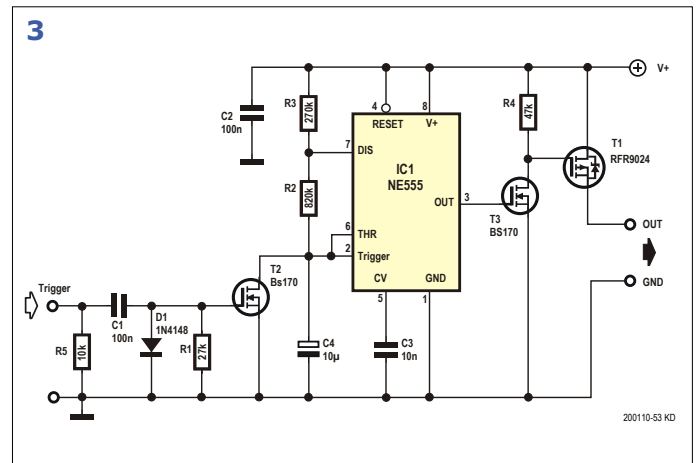
heureusement que rarement. Quand ça arrive, le signal d'initialisation normal produit par le chien de garde interne au μC peut ne pas suffire. Il ne reste qu'à couper l'alimentation du μC puis à la remettre en marche.

Comme le fait ce petit circuit (fig. 3) de chien de garde externe : s'il ne reçoit aucun signe de vie du microcontrôleur pendant 10 s environ, il coupe la tension d'alimentation (sortie OUT) pendant 5 s environ, puis la remet en marche.

Ce circuit est construit autour du fameux temporisateur 555. Son fonctionnement est simple : les impulsions de signe de vie régulières (entrée *Trigger*) émises par le μC (remarque : c'est au logiciel qu'il incombe de produire ces impulsions à intervalles réguliers, ce qui ne posera pas de problème à un programmeur expérimenté) mettront chaque fois T2 en état de conduction, ce qui entraîne la décharge du condensateur de synchronisation C4 du 555. Ce dernier est configuré ici en multivibrateur.

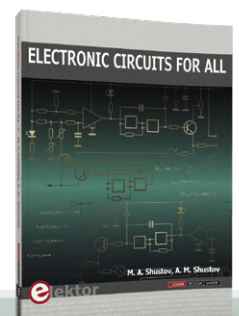
Normalement, la tension aux bornes de ce condensateur n'a jamais le temps d'atteindre le seuil de basculement du 555. La

3



sortie (broche 3) du 555 reste donc haute et la tension d'alimentation du μC lui est appliquée à travers T1 commandé par T3. En l'absence d'impulsions de signe de vie (*trigger*) pendant plus de 10 s, la tension aux bornes de C4 atteint le seuil de basculement du 555 : sa sortie devient basse, T1 ne conduit plus et l'alimentation du μC est coupée. Après environ 5 s, le 555 se remet en marche et l'alimentation électrique est rétablie. ◀

200110-04



@ WWW.ELEKTOR.FR

→ Livre en anglais : Electronic Circuits For All
www.elektor.fr/electronic-circuits-for-all

boîte inviolable protégée par un témoin d'effraction

Envoyer des données en toute sécurité par la poste

Luka Matić (Croatie)

Dans les systèmes de communication sécurisés basés sur des blocs-notes numériques à usage unique, les messages sont cryptés et décodés à l'aide d'une clé aléatoire. Pour que cela fonctionne, l'expéditrice, appelée Alice par convention, et le destinataire, appelé Bob ou Bernard, doivent utiliser la même clé. Si Alice produit la clé, comment l'envoie-t-elle à Bob de manière sécurisée ? C'est à ce problème de la poule et de l'œuf que s'attaque cet article. Jouissif !



Caractéristiques

- Code source ouvert sans astuces cachées
- Détection capacitive d'ouverture de boîte
- Détection lumineuse d'ouverture intempestive
- Vérification du temps de voyage
- Détection des attaques de température
- Communication infrarouge
- Réutilisable

Ce projet est la suite (logique) de mon TRNG [1] et de mon OTP Crypto Shield [2] et tente de résoudre le problème de l'échange sécurisé de clés OTP (*one-time pads* ou OTP), stockées p. ex. sur une carte SD. J'ai commencé par l'idée de concevoir un dispositif inviolable, mais j'ai ensuite appris qu'il est pratiquement impossible d'effacer en toute sécurité les données d'un dispositif à mémoire flash comme une mémoire de masse à semi-conducteurs (SSD) ou une carte SD [3].

De plus, même la mémoire SRAM présente des effets parasites de *brûlage* ou de rémanence, permettant de récupérer les données qu'elle contient [4]. C'est dans cet esprit que j'ai renoncé à construire une boîte inviolable, pour en concevoir une qui indiquera qu'elle a été crochétée.

Principe de fonctionnement

Alice stocke une clé OTP fraîchement produite sur une carte microSD et la met dans une boîte qu'elle ferme. Elle

l'arme en y stockant une chaîne dite de défi et de réponse (*challenge-and-response string*) au moyen d'une liaison sans contact (infrarouge). Elle envoie la boîte à Bob par la poste. Lorsqu'il la reçoit, il demande à Alice la chaîne de défi. Pour cela, il utilise n'importe quel canal *non sécurisé*, le téléphone p. ex. ou le courriel, car seule Alice connaît cette chaîne et Bob a la boîte. Bob déverrouille la boîte en lui envoyant la chaîne de défi par la liaison optique. La boîte répondra soit avec la chaîne de réponse, soit avec l'avertissement d'une éventuelle tentative d'intrusion. Seule Alice peut confirmer la validité de la chaîne de réponse. Si la chaîne de réponse est incorrecte, ou si la boîte émet un avertissement de sabotage, Bob jette simplement la carte SD et n'utilisera pas pour des communications sécurisées avec Alice la clé de ce bloc-notes OTP potentiellement dangereuse. Si la boîte est ouverte sans avoir été désarmée au préalable à l'aide de la chaîne de défi, ou si elle est crochétée d'une autre manière, elle effacera en

toute sécurité les chaînes défi-réponse de sa mémoire. Ève, qui écoute aux portes, ou Mallory, qui est *malveillante*, peuvent intercepter la boîte, lire et copier la carte SD, puis transmettre la boîte à Bob, mais ils ne peuvent pas réarmer la boîte avec les mêmes chaînes défi-réponse que seule Alice connaît, et Bob finira donc par le découvrir.

Description du matériel

Sur la **fig. 1** au cœur du schéma de la boîte, le μC IC1 est le cerveau de l'appareil. Toutes les deux secondes, il se réveille du mode de veille pendant moins de 50 ms pour vérifier plusieurs paramètres. Si ces paramètres indiquent une intrusion, le μC efface (met à zéro) de manière sécurisée le contenu de sa SRAM, ce qui n'est pas possible avec une mémoire flash ou une EEPROM. L'ouverture de la boîte est détectée de deux façons :

- Modification de la capacité de la boîte (il doit s'agir d'une boîte métal-

INFOS SUR LE PROJET

communication sécurisée

cryptage

communication IR

AVR

débutant

connaissseur

expert

±3 h

fer à souder (CMS),
programmeur AVR

±50 €.

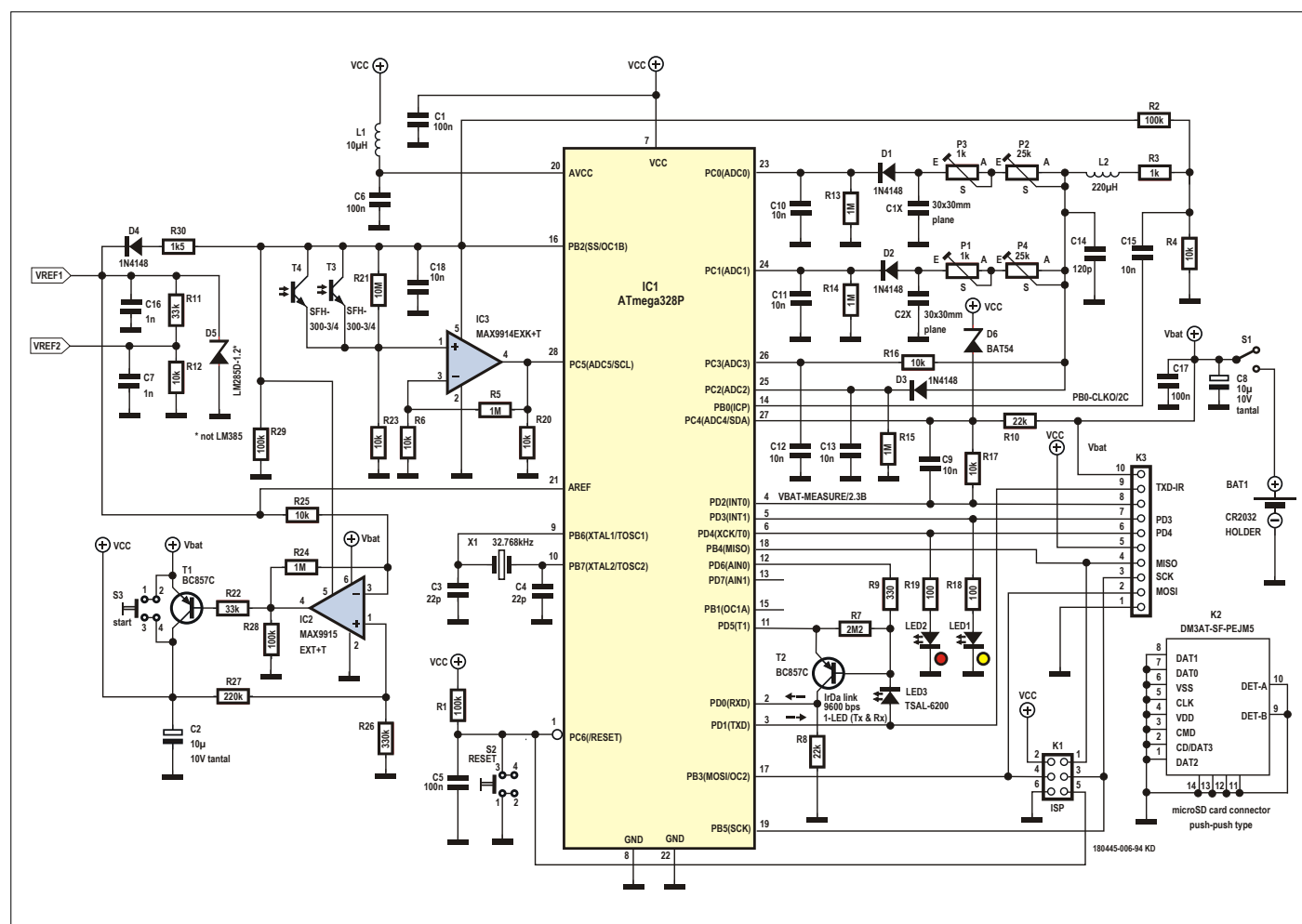


Figure 1. La boîte protégée par un témoin d'effraction utilise des phototransistors et des condensateurs à plaque ouverte pour détecter les effractions.

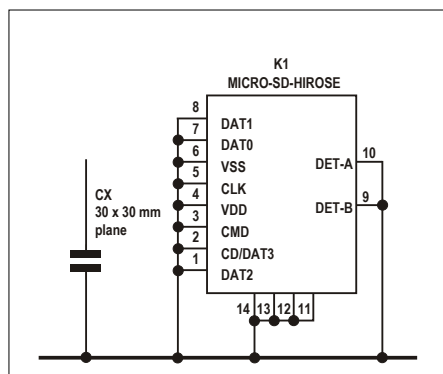


Figure 2. Les condensateurs à plaque ouverte C1x et C2x sont des plaques cuivrées de 30 x 30 mm (matériau pour PCB simple face). Un connecteur de carte microSD peut également être monté.

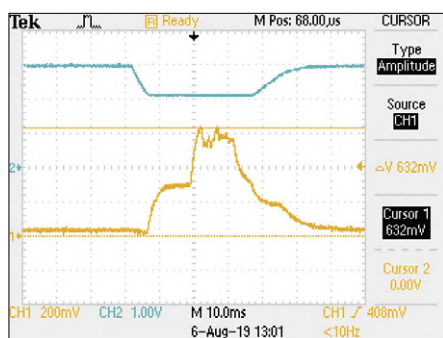


Figure 3. Pour limiter sa consommation, la boîte ne se réveille que toutes les deux secondes pendant environ 50 ms. Le régulateur de tension est activé (trace supérieure) pour mesurer les paramètres de surveillance d'intrusion. La trace du bas indique la consommation de courant.

lique). C1x et C2x (**fig. 2**) sont des condensateurs à plaque ouverte qui protègent la carte SD placée entre eux. Leur autre plaque est la paroi métallique (!) de la boîte : toute modification de leur capacité = intrusion.

- Détection de la lumière. La boîte scellée est étanche à la lumière ; il y fait noir. Les phototransistors T3 et T4 détectent l'un la lumière infrarouge et l'autre la lumière visible aussitôt que la boîte est ouverte.

La LED3 infrarouge IrDA assure les communications bidirectionnelles pour les codes défi-réponse. Sa portée est d'environ 1 cm, c'est peu mais c'est assez. Le circuit est alimenté par une pile CR2032 (3 V). Elle durera plus d'un mois (**fig. 3**), ce qui devrait suffire même si le facteur fait sa tournée à pied. La tension d'une telle pile neuve est généralement

de 3,3 V. Ce circuit fonctionnera jusqu'à 2,1 V, grâce au régulateur de tension à faible perte (LDO) autour de IC2, qui produit une tension V_{cc} stable de 2,0 V. D5 fournit une référence de tension précise pour le convertisseur analogique-numérique (ADC) du μC . Le LM285 est une variante à plage de température étendue du LM385, ce qui permet au circuit de fonctionner en toute sécurité avec une précision suffisante de -20 °C à +60 °C, c'est-à-dire les températures que l'on peut rencontrer lors d'un périple par courrier postal.

Le μC est cadencé par son oscillateur RC interne de 1 MHz. Pour réduire la consommation, cette horloge n'est allumée que toutes les 2 s pendant environ 50 ms. L'horloge à faible consommation de 32,768 kHz, calée sur le quartz horloger X1, fonctionne en continu et chronomètre ces moments de réveil. Elle est également utilisée pour régler l'horloge UART du μC à chaque réveil, et fournit la base de temps pour mesurer précisément la durée du voyage de la boîte.

R3, L2 et C14 filtrent l'onde carrée de 1 MHz dérivée de la sortie d'horloge du μC sur PB0, pour obtenir une onde sinusoïdale utilisable pour mesurer avec précision les capacités C1x et C2x. On mesure les valeurs de crête et moyennes et on calcule les rapports de la tension alternative sur C1x et C2x par rapport à la tension alternative sur C14. Un changement significatif de l'un de ces rapports est le signe d'une intrusion. LED1 et LED2 indiquent les différents états de la boîte. LED1 clignote lorsqu'un code de défi correct a été saisi. LED2 clignote s'il est incorrect. Elles clignent une fois toutes les deux après la mise sous tension. LED1 clignote une fois à chaque réveil du μC , car elle est utilisée pour aider la sortie LDO à atteindre rapidement 2 V. Si LED2 clignote une fois à chaque réveil du μC , c'est qu'un sabotage a été détecté.

L'interrupteur S1 est l'interrupteur d'alimentation du boîtier, et S2 est le bouton de réinitialisation du μC . Le bouton S3 permet de redémarrer le circuit au cas où la tension de la batterie serait un peu trop faible.

Notez qu'il y a des composants des deux côtés de la carte. L'idée est de cacher une partie du circuit à un intrus, notamment les bons endroits pour placer des sondes ou interférer autrement sans retirer le PCB, ce que le condensateur à plaque ouverte du bas détecterait aussitôt.

Interface IR

Une liaison infrarouge (IR) est disponible pour communiquer avec la boîte inviolable quand elle est hermétiquement fermée. Cette liaison n'est pas particulièrement rapide (9600 bauds) et sa portée est courte (1 cm). Elle fonctionne sans qu'il soit nécessaire d'ouvrir la boîte. L'adaptateur (**fig. 4** et **fig. 5**) qui rend cela possible est constitué d'un petit μC ATtiny45, équipé d'un émetteur-récepteur IR presque identique à celui de la boîte. Les circuits diffèrent légèrement en raison des contraintes de faible puissance de la boîte, lesquelles ne s'appliquent pas à l'adaptateur IR.

Côté utilisateur de l'adaptateur, il y a un port série. Comme la saisie des codes de défi-réponse est critique pour la sécurité, l'adaptateur IR peut se connecter à un ordinateur de différentes manières, p. ex. par un convertisseur USB-série (câble), par un «vrai» port RS-232, et même par Ethernet si vous ajoutez un adaptateur WIZnet WIZ107 Ethernet-série. Ces options permettent de réserver à la communication avec cette boîte un PC portable bon marché qui ne se connecterait jamais à l'internet.

Ajustement des détecteurs capacitifs

Pour garantir la fiabilité de la détection de sabotage capacitive, les ajustables P1, P2, P3 et P4 doivent être réglés. Mettez le boîtier sous tension (la LED2 clignotera une fois, suivie de la LED1) et placez la LED IR de l'adaptateur IR devant la LED3 du boîtier à moins de 1 cm. Connectez l'adaptateur IR à un ordinateur équipé d'un terminal série (9600n81). Après avoir tapé «U» (0x55) dans le terminal, vous verrez ce bloc de données arriver toutes les deux secondes :

```
Box unlocked. Status-1. Define
challenge string.
Capacitor Cx1 voltage ratio: 0.703
Capacitor Cx2 voltage ratio: 0.703
Phototransistor voltage [V]: 0.181
Box temperature [degC]: +22
Battery voltage [V]: 2.900
```

Pour une sensibilité maximale, P1 à P4 doivent être réglés de telle sorte que la résistance combinée [P2+P3] soit égale à la réactance C1x, et que la somme [P1+P4] soit égale à la réactance C2x. Cela semble compliqué, mais les données d'état contiennent des rapports de tension pour vous aider ici. Ils doivent

être proches de $1/\sqrt{2}$, c'est-à-dire 0,707. Comme la boîte elle-même sert de plaque de condensateur pour les condensateurs à plaque ouverte C1x et C2x, les rapports de tension doivent être corrects avec le couvercle de la boîte en place. Utilisez P2 et P4 pour les réglages grossiers et P1 et P3 pour les réglages fins.

Mesures de la température

Le μ C est équipé d'un capteur de température intégré. Sa précision est médiocre ($\pm 5^\circ\text{C}$) mais suffisante pour détecter des températures anormales et pour enregistrer les températures minimales et maximales approximatives pendant le voyage de la boîte. La température est mesurée par la lecture du canal 8 de l'ADC, qui est le thermomètre interne du μ C. Bien que conçu pour fonctionner avec la référence de tension interne du μ C (1,1 V et pas très stable), si l'on ajoute un peu de mathématiques (une fonction *affine*), une référence de tension externe (D5) peut être utilisée pour améliorer les choses. On obtient ainsi une précision bien meilleure que celle indiquée dans la fiche technique du μ C. Si nécessaire, la pente de la fonction *affine* peut être modifiée à l'intérieur du programme, dans la fonction `measure_signals()`. Pour qu'il puisse servir, le capteur de température doit être calibré avec la commande «TP», à envoyer depuis le terminal. Sa syntaxe est «TPsxx» où «s» est le signe «+» ou «-» (le signe est obligatoire), et «xx» est le décalage (sous forme de valeur à 2 chiffres, c'est-à-dire de «00» à «99»). Par exemple, si la valeur de la «température de la boîte» est de $+40^\circ\text{C}$ alors que la température ambiante est réellement de $+25^\circ\text{C}$, vous devez envoyer «TP-15» depuis le terminal. Pour remettre le décalage à zéro : «TP+00». Le décalage est stocké dans l'EEPROM interne du μ C.

Manuel pour les espions

Alice crée un fichier de nombres aléatoires avec son TRNG et le copie sur une carte SD jamais utilisée. Elle insère une pile neuve dans la boîte et l'allume avec S1. Comme la pile est fraîche, il ne sera pas nécessaire d'appuyer sur S3 pour faire démarrer l'appareil. Puis Alice met la carte SD dans la prise et ferme la boîte. Avec l'adaptateur IR en face de la LED IR du boîtier, elle envoie d'abord le caractère «U» (0x55) de son terminal. Cela permet au μ C de l'adaptateur IR de calibrer son horloge UART. Le terminal

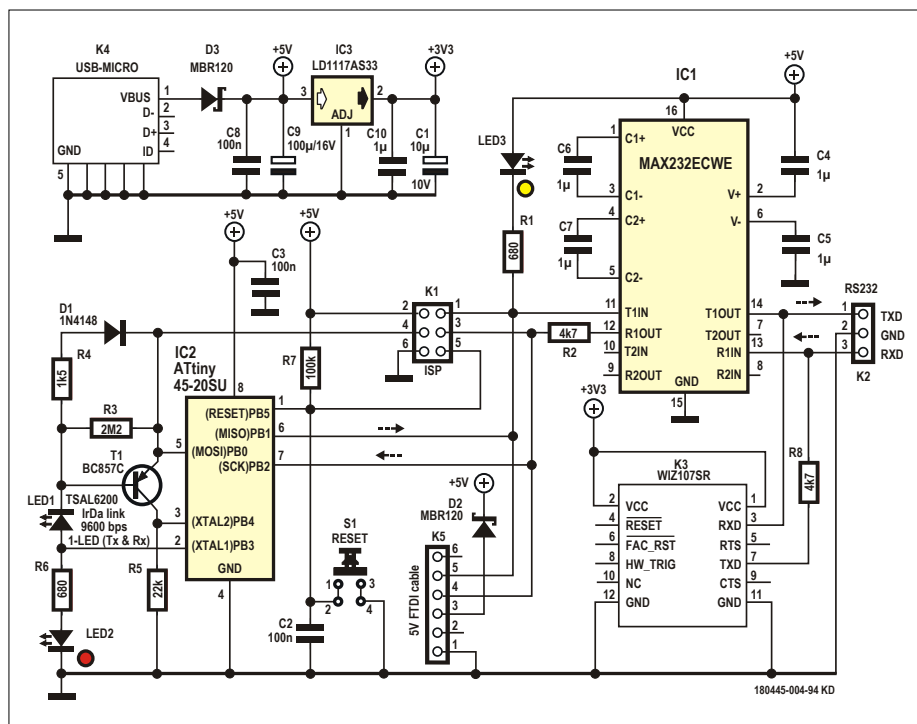


Figure 4. L'adaptateur de communication IR fournit une interface entre le boîtier inviolable hermétiquement fermé et un ordinateur.

commencera à recevoir des messages de la boîte, affichant quelque chose de ce genre :

```
Box unlocked. Status-1. Define
challenge string.
Capacitor Cx1 voltage ratio: 0.710
Capacitor Cx2 voltage ratio: 0.704
Phototransistor voltage [V]: 0.181
Box temperature [degC]: +25
Battery voltage [V]: 3.354
```

Le boîtier émet ces blocs de données

toutes les deux secondes sur son port infrarouge, sauf dans les états 3 et 5. Dans ces états, les transmissions de données s'arrêtent après 60 s pour économiser l'énergie.

Statut 1

L'état 1 est l'état initial du boîtier après la mise sous tension, et l'appareil attend qu'une chaîne de défi arrive sur son récepteur infrarouge. Alice saisit la chaîne de défi dans le terminal (jusqu'à 65 caractères, les chaînes plus longues sont

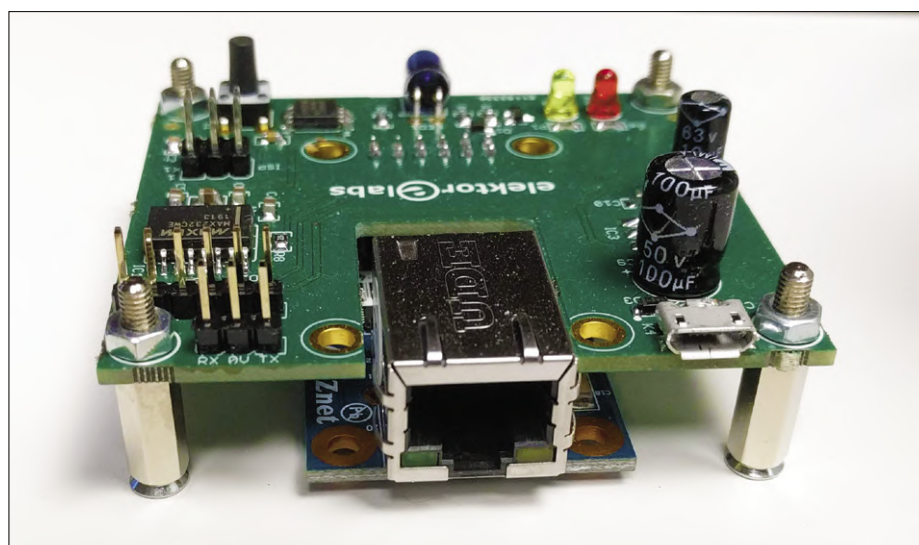


Figure 5. L'adaptateur de communication IR avec le module Ethernet (en option) monté.

tronquées). La boîte passe alors à l'état 2.
Remarque importante : toutes les chaînes de caractères, y compris les commandes à plusieurs caractères, doivent être copiées dans le terminal et non tapées caractère par caractère !

Box unlocked. Status-2. Define response string.

Capacitor Cx1 voltage ratio: 0.710
 Capacitor Cx2 voltage ratio: 0.704
 Phototransistor voltage [V]: 0.181
 Box temperature [degC]: +25
 Battery voltage [V]: 3.354

Statut 2

La boîte attend qu'une chaîne de réponse soit saisie. Les rapports de tension de Cx1 et Cx2 et la tension du phototransistor sont stockés dans la SRAM du µC. Si ces tensions changent trop pendant le voyage, une alerte de sabotage est donnée. Alice entre la chaîne de réponse (jusqu'à 65 caractères) pour armer et verrouiller la boîte et la faire passer à l'état 3. La pendule de voyage est également mise en marche. Alice note la date et l'heure exacte, ainsi que les chaînes défi-réponse, et cache le tout de manière sûre (dans sa

tête, si possible). Ces infos de vérification seront utilisées plus tard. Elle envoie ensuite la boîte à Bob par la poste.

Box locked. Status-3. Enter challenge string to unlock.

Capacitor Cx1 voltage ratio: 0.710
 Capacitor Cx2 voltage ratio: 0.704
 Phototransistor voltage [V]: 0.181
 Box temperature [degC]: +25
 Battery voltage [V]: 3.354

Statut 3

Les chaînes de défi et de réponse ont été



LISTE DES COMPOSANTS

Boîte inviolable

Résistances

Par défaut : SMD 0805

R18,R19 = 2 MΩ

R9 = 100 Ω

R3 = 22 kΩ

R30 = 1,2 MΩ

R4,R6,R12,R16,R17,R20,R23,R25 = 220 kΩ

R8,R10 = 33 kΩ

R11,R22 = 1 kΩ

R1,R2,R29,R28 = 22 kΩ

R27 = 330 kΩ

R26 = 1 kΩ

R5,R13,R14,R15,R24 = 220 kΩ

R7 = 2,22 kΩ

R21 = 10 MΩ

P3,P1 = 1 kΩ aj.

P2,P4 = 10 kΩ aj.

Condensateurs

Par défaut : SMD 0805

C3,C4 = 22 pF

C14 = 120 pF

C7,C16 = 1 nF

C9,C10,C11,C12,C13,C15,C18 = 10 nF

C1,C5,C6,C17 = 100 nF

C2,C8 = 10 µF 10 V, tantale, pas 2 mm

Inductances

L1 = 10 µH (0805)

L2 = 220 µH (1812)

Semi-conducteurs

D1,D2,D3,D4 = 1N4148WS (SOD-323)

D5 = LM285D-1,2 V

D6 = BAT54

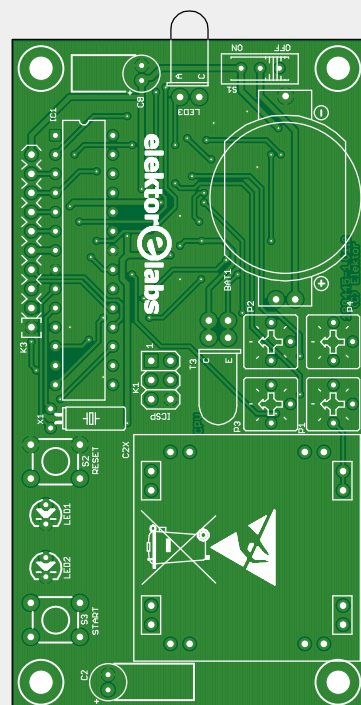
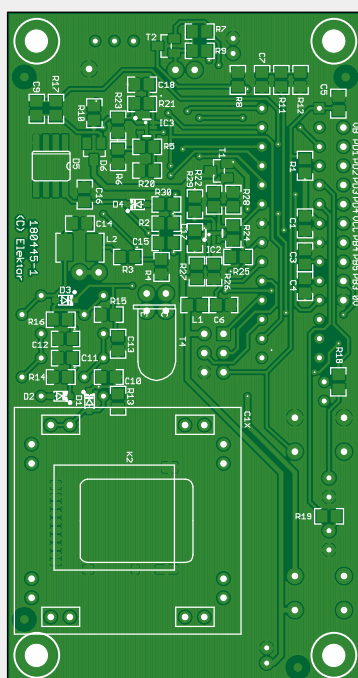
IC1 = ATmega328P-PU, programmé

IC2 = MAX9915EXT+T

IC3 = MAX9914EXK+T

LED1 = 3 mm, rouge

LED2 = 3 mm, jaune



LED3 = TSAL6200

T1,T2 = BC857C

T3,T4 = SFH-300-3/4

Divers

BAT1 = support de pile, 2032,
 p. ex. Renata HU2032

C1x,C2x = connecteur à 2 broches + prise
 pour C1x & C2x, pas 2,54 mm

K1 = connecteur à 6 broches (2x3),
 pas 2,54 mm

K2 = connecteur de carte microSD, poussoir

K3 = connecteur à 10 broches, pas 2,54 mm

S1 = interrupteur à glissière, SPDT S2,

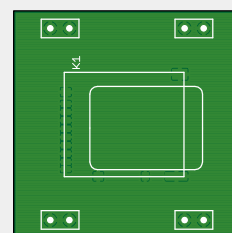
S3 = interrupteur tactile, 6x6 mm

X1 = cristal de quartz 32,768 kHz

support scellé pour LED3

circuit imprimé #180445-1 (1x)

circuit imprimé #180445-3 (2x)



saisies et la boîte est armée et verrouillée. Toute manipulation ou saisie d'une chaîne de défi incorrecte cinq fois de suite déclenchera la séquence de remise à zéro. La boîte reste en statut 3 pendant son voyage par la poste.

Toutes les deux secondes, les mémoires tampons SRAM contenant les chaînes défi-réponse sont complémentées (inversées bit par bit). De cette façon, puisque chaque bit passe le même temps à l'état «0» qu'à l'état «1», on évite de brûler des cellules SRAM [4].

Chez Bob

Après un long et périlleux voyage, la boîte arrive (espérons-le) chez Bob. Il connecte son adaptateur IR à un ordinateur, le place devant la LED IR du boîtier et appuie sur le bouton Reset de l'adaptateur. Cela produira une impulsion de trois secondes sur le port infrarouge pour indiquer à la boîte d'activer son port infrarouge et de commencer à communiquer. Au bout de trois secondes, Bob envoie un seul caractère «U» pour calibrer l'UART de l'adaptateur infrarouge et le terminal commence à afficher les données :

```
Box locked. Status-3. Enter
challenge string to unlock.
Capacitor Cx1 voltage ratio: 0.703
Capacitor Cx2 voltage ratio: 0.711
Phototransistor voltage [V]: 0.175
Box temperature [degC]:      +23
Battery voltage [V]:         2.752
```

Comme on le voit, la tension de la batterie a diminué, mais il n'y a aucun signe de sabotage car les rapports de tension Cx et les valeurs de tension du phototransistor n'ont pas beaucoup changé. Bob appelle Alice et lui demande la chaîne de défi qu'il saisit dans son terminal et, si tout va bien, il verra :

```
Box unlocked. Status-4. Challenge
string correct!!
Capacitor Cx1 voltage
ratio:      0.703
Capacitor Cx2 voltage ratio: 0.711
Phototransistor voltage [V]: 0.178
Box temperature [degC]:      +24
Battery voltage [V]:         2.752
Min temperature [degC]:      -2
Max temperature [degC]:      +31
Travel time [s]:             502,164
Response:
Trust no one 01a2g23w46e57f80g12r3
e34fv245hasdvfr4.
```

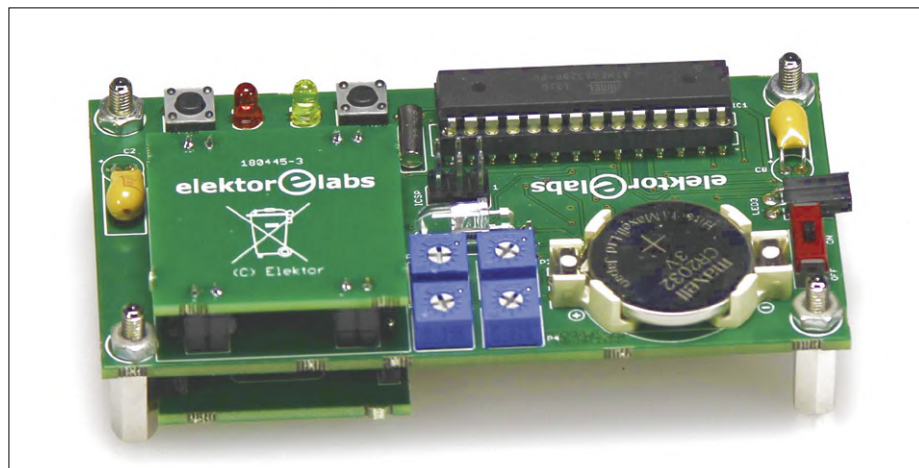


Figure 6. La carte entièrement assemblée avec des condensateurs à plaque ouverte installés des deux côtés.

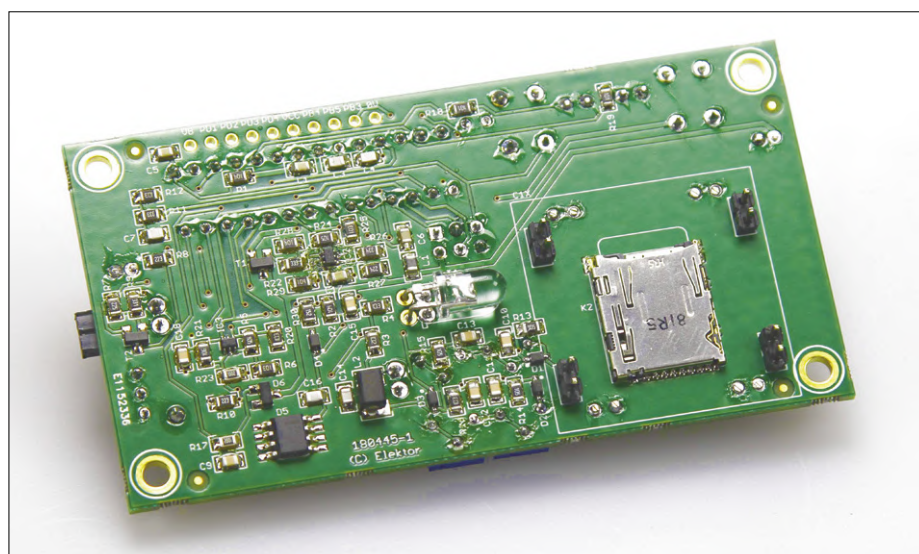


Figure 7. La face inférieure de la carte est peuplée de composants CMS et d'un transistor T4 traversant qui ressemble à une LED. Le condensateur C1x est monté sur le connecteur de la carte microSD.

Statut 4

Bob note immédiatement la date et l'heure actuelles et appelle à nouveau Alice pour vérifier la chaîne de réponse. Si Alice l'accepte, elle donnera à Bob l'heure et la date exactes qu'elle a notées lorsqu'elle a fermé la boîte. Cela permet à Bob de calculer le temps de trajet théorique et de le comparer au temps mesuré par la boîte. Si les deux correspondent, alors Bob est autorisé à utiliser la clé OTP stockée sur la carte microSD pour crypter et décrypter ses communications sécurisées avec Alice. En cas d'altération en cours de route, la boîte effacera la partie de la SRAM contenant les codes et la minuterie de voyage. Dans ce cas, la boîte affichera quelque chose comme ceci :

```
Box zeroized. Status-5. Tampering,
wrong password, or low battery!
Capacitor Cx1 voltage ratio: 0.712
Capacitor Cx2 voltage ratio: 0.713
Phototransistor voltage [V]: 0.179
Box temperature [degC]:      +12
Battery voltage [V]:         2.793
```

Statut 5

La séquence de mise à zéro a été exécutée en raison de la détection d'un événement de sabotage ou après avoir saisi cinq chaînes de défi invalides. Les tensions de contrôle de l'altération ne sont plus mises à jour, ce qui permet au détenteur de la boîte de voir le dernier état enregistré avant la détection de l'intrusion.

La séquence de mise à zéro est exécutée dans l'un des cas suivants :

- Après avoir saisi une chaîne de défi incorrecte cinq fois ;
- Le rapport de tension réel du condensateur diffère trop de la valeur enregistrée ;
- La tension réelle du phototransistor diffère trop de la valeur enregistrée ;
- Tension de la batterie trop faible (inférieure à 2,1 V) ;
- Température hors limites (inférieure à -20°C ou supérieure à +60°C).

Pour éviter que la lumière parasite ne déclenche l'événement d'intrusion 3, assurez-vous de la fermeture hermétique de la boîte. Appliquez une pâte noire autour de la LED3 à l'intérieur de la boîte et scellez les bords de la boîte avec du ruban adhésif noir. Testez-le en exposant la boîte fermée (dans l'état 3) à une forte lumière solaire directe. Maintenant, Bob peut entrer la commande «DE» pour voir ce qui a causé la mise à zéro :

```
Box zeroized. Status-6. Tampering
status display!
Capacitor Cx1 voltage ratio: 0.683
Capacitor Cx2 voltage ratio: 0.656
Phototransistor voltage [V]: 0.188
Box temperature [degC]: +10
Battery voltage [V]: 2.775
Min temperature [degC]: -5
Max temperature [degC]: +30
Cause of zeroization:
Capacitor Cx2. kCx2= 0.656
```

Statut 6

Après avoir reçu la commande «DE», la boîte transmet des détails sur la cause de la mise à zéro.

Les données affichées dans les états 5 ou 6 n'ont pas besoin d'être valides car elles dépendent de ce que Mallory a fait de la boîte. Elle peut l'avoir réinitialisée et avoir entré des chaînes défi-réponse avant de transmettre la boîte à Bob dans l'état 3. Cependant, même si elle parvient à entrer les bonnes chaînes, elle devra également usurper le temps de parcours, ce qui sera beaucoup plus difficile.

Retour au statut 1

La commande «RE» peut être utilisée pour redonner à la boîte le statut 1 sans l'ouvrir. En appuyant sur le bouton de réinitialisation S2 à l'intérieur de la boîte, vous obtiendrez la même chose.

Chevaux de Troie matériels

On ne s'inquiète généralement que des logiciels malveillants et des chevaux de Troie qui attaquent un ordinateur. Quelques mots sur les Troyens matériels ne seraient pas déplacés. Un système simple avec un seul μC , jamais connecté à l'internet, est à l'abri des logiciels malveillants typiques qui cibleraient votre PC ou votre téléphone. La référence [5] donne une explication de base des problèmes liés aux dispositifs de mémoire à semi-conducteurs à très haut niveau d'intégration.

Un Troyen matériel à porte dérobée

consiste en un circuit supplémentaire ajouté à la puce de silicium du μC . Cela peut faire autant de mal qu'un cheval de Troie logiciel ou même plus. Il représente un danger pour les μC d'aujourd'hui dont le processeur, la SRAM, l'EEPROM et la mémoire flash sont intégrés sur une puce de silicium et sont programmables à basse tension (3 à 5 V). Il en existe essentiellement deux types :

1. Troyens matériels à usage général

Mallory peut implanter un cheval de Troie matériel dans le μC , p. ex. pour copier périodiquement la SRAM et les registres du μC dans une partie secrète de la mémoire flash. Quelle que soit l'application du μC , un instantané «gelé» de la RAM sera toujours utile pour elle et Ève. Avec un tel cheval de Troie en place, Mallory peut facilement lire les chaînes de défi et de réponse définies par Alice. Il sera un peu plus difficile de pirater la minuterie de voyage.

2. Troyens matériels spécifiques à une application

Pour installer ce type de cheval de Troie matériel, Mallory doit d'abord connaître le micrologiciel de l'unité centrale de traitement. Ensuite, elle installe sur le μC un circuit capable de reconnaître le micrologiciel qu'Alice a flashé dans son μC . Au démarrage du μC , le cheval de Troie agira comme un chargeur de démarrage et modifiera quelques lignes du programme du μC (p. ex. les remplir de NOP en utili-



LISTE DES COMPOSANTS

Adaptateur IR

Résistances

Par défaut : SMD 0805

R1, R6 = 680 Ω
R4 = 1,5 k Ω
R2, R8 = 4,7 k Ω
R5 = 22 k Ω
R7 = 100 k Ω
R3 = 2,2 M Ω

Condensateurs

Par défaut : SMD 0805

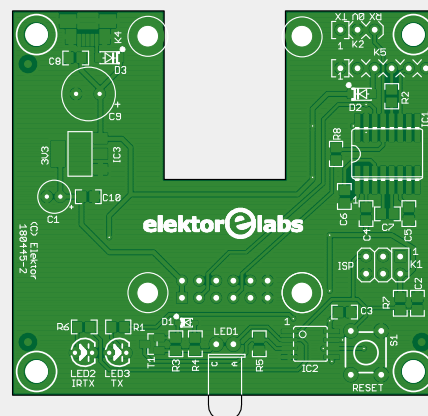
C2, C3, C8 = 100 nF
C4, C5, C6, C7, C10 = 1 μF , X7R, 50 V
C1 = 10 μF 10 V tantale, pas 2 mm
C9 = 100 μF 16 V, pas 3,5 mm

Semi-conducteurs

D1 = 1N4148WS (SOD-323)
D2, D3 = MBR120
IC1 = MAX232ECWE
IC2 = ATtiny45-20SU, programmé
IC3 = LD1117AS33
LED1 = TSAL6200
LED3 = 3 mm, jaune
LED2 = 3 mm, rouge
T1 = BC857C

Divers

K1 = connecteur à 6 broches (2x3), pas 2,54 mm
K2 = connecteur à 3 broches, pas 2,54 mm
K3 = douille à broche à 12 voies (2x6), pas 2,54 mm
K4 = connecteur micro USB de type B
K5 = connecteur à 6 broches, pas 2,54 mm



S1 = interrupteur tactile, 6x6 mm
module Ethernet WIZ107
circuit imprimé #180445-2

sant la fameuse commande SPM de l'AVR) pour désactiver par exemple certaines actions de protection cruciales. Les chances de succès d'une attaque de ce type sont beaucoup plus faibles qu'avec un cheval de Troie à usage général.

Quelle parade ?

Il existe de nombreux moyens de contrer ce genre de menaces matérielles. La plus simple est de revenir à la technologie des années 1980 et d'utiliser un simple système Z80 [6] avec SRAM et EPROM comme circuits intégrés séparés. L'EPROM est effacée par une lumière UV et a besoin d'une tension de 13 à 14 V sur sa broche V_{pp} pour l'écriture, ce qui rend beaucoup plus difficile l'installation d'une porte dérobée matérielle. Les architectures internes des EPROM de type Z80 et 27C (produites par millions durant plus de 20 ans) sont très connues, et il est facile de décapsuler le CI et de l'inspecter au microscope si nécessaire. Avec les millions de pièces qui traînent autour du globe, Mallory aurait fort à faire pour les trafiquer toutes.

Le portage du micrologiciel de l'unité centrale de traitement du TRNG ou de l'OTP Crypto Shield sur un Z80 peut se faire au détriment de la vitesse d'exécution et de la taille physique, mais les systèmes fonctionneraient quand même bien. Je signale ici une autre idée fausse de l'ingénierie de conception actuelle : les dispositifs mini, micro et nano ne sont pas forcément meilleurs !

Bien sûr, un système basé sur le Z80 serait trop lourd et trop gourmand pour être utilisé pour notre boîte postale, mais ce problème peut être résolu d'une autre manière, avec un peu de circuits supplémentaires, tout étant installé à l'intérieur de la boîte existante. Si vous avez étudié attentivement le schéma, vous vous

demandez peut-être quelle est la fonction de K3. Il est destiné à un circuit supplémentaire pour protéger la boîte contre les Troyens matériels. Vous en saurez plus dans un prochain article.

Portes dérobées

Le prototype de la boîte sur les photos a fait plusieurs voyages avec différents transporteurs et il a fonctionné correctement. J'ai eu l'occasion de constater que chez certains transporteurs les paquets sont plus chahutés que chez d'autres. J'ai compris qu'il fallait arrimer C1x et C2x pour qu'ils ne se détachent pas. Une des méthodes consiste à coller dessus de la mousse non conductrice ou des morceaux de plastique. Si le μC est monté sur support, vous pouvez également l'assujettir. La pile est en principe assez serrée dans son support, mais on ne sait jamais. Si vous avez d'autres suggestions d'améliorations, notamment de défense contre les portes dérobées, veuillez me contacter sur la page du projet sur Elektor Labs [7]. J'ai des idées, mais vous en avez peut-être d'autres et de meilleures.



Bob et Alice finissent par se rencontrer.

Proposez-les. Pour confirmer sa sécurité, tout dispositif de cryptage doit être examiné par nombre d'experts indépendants, alors n'hésitez pas !

Les fichiers de conception et le code source de ce projet peuvent tous être téléchargés sur la page du projet [7].

180445-02



WWW.ELEKTOR.FR

→ Boîte à témoin d'effraction - panneau de tous les PCB nus
www.elektor.fr/180445-1

→ Boîte à témoin d'effraction - μC ATmega328P-PU programmé
www.elektor.com/180445-41

→ Adaptateur IR - μC ATtiny45-20SU programmé
www.elektor.com/180445-42

Liens

- [1] Générateur de nombres aléatoires : www.elektormagazine.com/labs/random-number-generator-150116
- [2] Système de cryptage à bloc-notes à usage unique (OTP) : www.elektormagazine.com/labs/one-time-pad-otp-crypto-system
- [3] Effacer les données des clés USB : www.schneier.com/blog/archives/2011/03/erasing_data_fr.html
- [4] Rémanence des données dans les dispositifs à semi-conducteurs : www.usenix.org/legacy/events/sec01/full_papers/gutmann/gutmann.pdf
- [5] Entretien vidéo d'Elektor avec l'auteur sur la sécurité : www.elektormagazine.com/news/secure-communications-an-interview-with-luka-matic
- [6] Z80 : un petit ordinateur simple : www.sunrise-ev.com/z80.htm
- [7] Boîte aux lettres inviolable pour la distribution sécurisée de blocs-notes à usage unique : www.elektormagazine.com/labs/tamper-evident-paper-mail-box-for-secure-distribution-of-one-time-pads

nouvelle horloge Nixie révisée

Un projet porté par l'enthousiasme de ses utilisateurs, nouveaux et anciens

Andy Leitch (Royaume-Uni) et Luc Lemmens (Elektor Labs)

Qui se serait douté que parmi vos projets favoris au cours des cinq dernières années figure une horloge à tubes Nixie ! Les ventes de kits ne mentent pas, vos nombreuses suggestions d'améliorations non plus. Nous les avons compilées et vous les proposons en réponse à votre enthousiasme pour que tous en profitent. Ces modifications varient de « pas mal, mais optionnel » à « urgent : faites quelque chose pour la durée de vie des Nixies ». Tout cela a abouti à une mise à jour de l'horloge et à une nouvelle carte de rétroéclairage, mais il est possible d'adapter votre carte existante et d'utiliser le nouveau logiciel.



INFOS SUR LE PROJET



Nixie
PIC
horloge mise à jour



débutant
→ connaisseur
expert



1 h env. (mise à jour & mods) ;
3 h (assemblage semi-kit)



outillage courant,
programmeur de PIC



10 € env. (mise à jour & mods) ;
140 € env. (semi-kit sans coffret)

En mai 2016, Elektor publiait la nouvelle horloge Nixie [1] et depuis lors, de nombreux lecteurs ont acheté soit le kit complet soit les circuits imprimés nus. Ceux-là ont acheté leurs composants ailleurs. À en juger par le nombre de réactions, notamment sur le site d'Elektor Labs [2], c'est l'un des projets les plus populaires de ces dernières années, pour lequel nous avons reçu des demandes et des suggestions d'améliorations, en particulier du logiciel. Cette tâche a été confiée à Andy Leitch, l'un des plus fervents adeptes d'Elektor Labs. Il a réécrit en C le logiciel produit à l'origine en assembleur. Il a adapté le circuit permettant à l'horloge d'afficher la date et de servir de réveil. Pour exploiter pleinement les fonctions du nouveau progiciel [3], il faut quelques modifications (mineures) des circuits imprimés. Si vous vous contentez d'ajouter l'affichage de la date et de prévenir/limiter

l'empoisonnement de la cathode du tube, il suffira de remplacer ou de reprogrammer le microcontrôleur.

L'amélioration la plus accrocheuse apportée par Andy est le rétroéclairage multicolore des tubes Nixie, qui remplace les six LED bleues du semi-kit. Ce n'est pas seulement décoratif. Le nouveau rétroéclairage indique également quelles informations sont affichées par l'horloge. Andy a utilisé pour cela une bande de circuit à pastilles et des LED WS2812 RGB traversantes. Le labo d'Elektor en a fait un circuit imprimé facile à assembler, avec les mêmes LED en boîtier SMA. Ce circuit imprimé de couleur noire est disponible dans l'e-choppe d'Elektor comme les autres PCB de ce projet. Nous y reviendrons.

Nouvelles fonctions, nouveau schéma

Le nouveau schéma (fig. 1) tient compte de certaines des modifications du logiciel

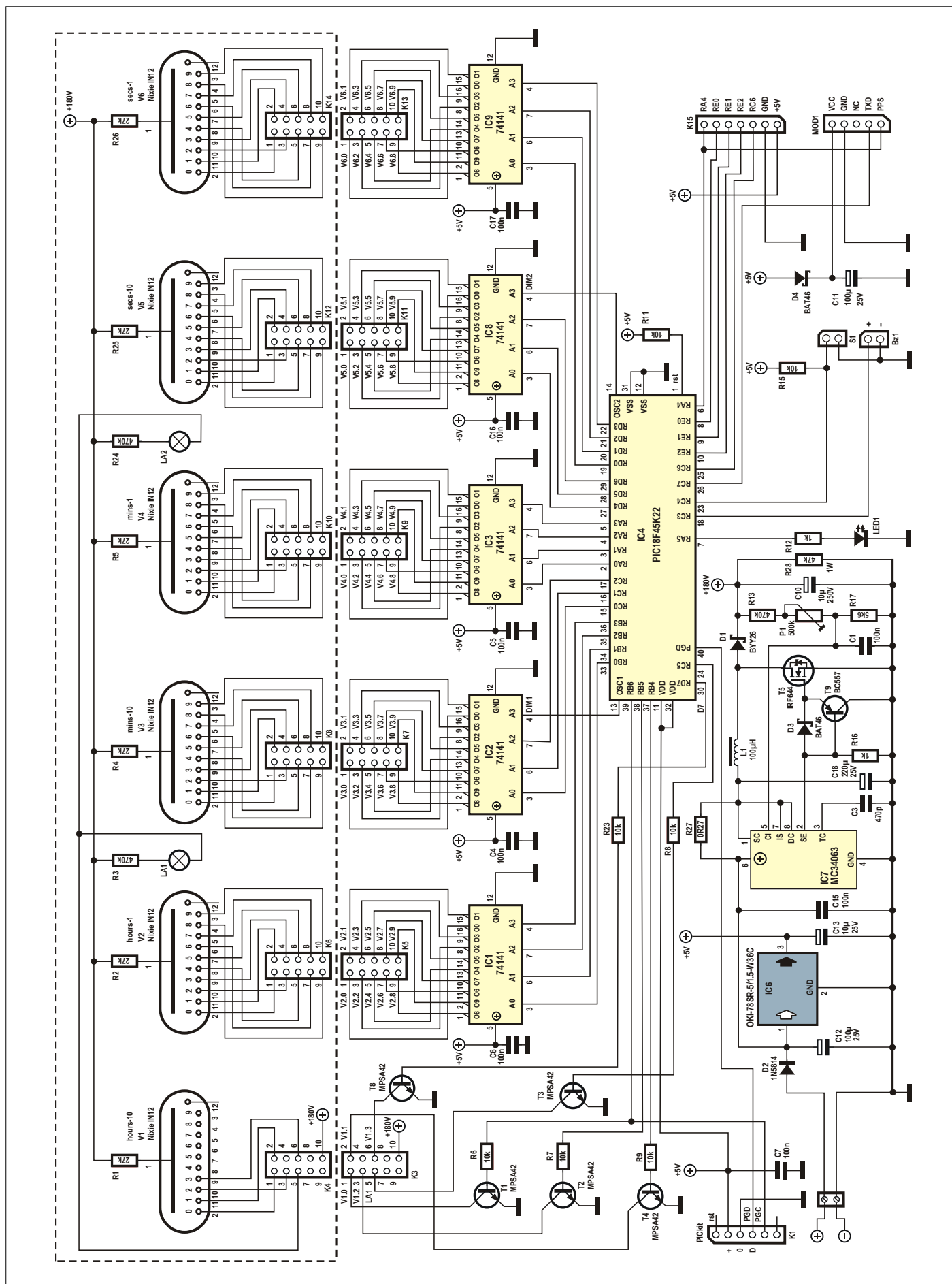


Figure 1. Schéma de la nouvelle horloge Nixie précise révisée. Cette version intègre toutes les modifications apportées au circuit original de 2016.

et correspond à la nouvelle carte désormais incluse dans les semi-kits de la nouvelle horloge Nixie de précision vendue dans l'e-choppe d'Elektor. Les instructions concernaient l'ancienne carte ; la nouvelle version V2.1 ne nécessite aucune modification.

Modification des boutons

Statut : obligatoire. Remplacez l'interrupteur à bascule existant par un modèle à contact fugitif. Cette modification très simple permettra d'ouvrir la plupart des fonctions du nouveau logiciel. En fait, il suffit de dessouder l'interrupteur existant et de le remplacer par un poussoir. Vous n'aurez rien à percer si vous en utilisez un avec un axe de 4,95 mm, il s'adaptera au trou de l'ancien.

Prévention de l'empoisonnement des cathodes

Statut : hautement recommandé.

Si un tube Nixie est utilisé pour n'afficher que certains chiffres, les fils non utilisés ont tendance à s'altérer. Cet effet appelé *empoisonnement de cathode* se traduit par des points ou des segments de chiffres sombres au lieu d'être allumés. Sur les Nixie d'une horloge, le chiffre des unités de secondes est rarement affecté par cet empoisonnement puisqu'il passe par tous les chiffres toutes les dix secondes, ce n'est pas le



Figure 4. Connexion de l'élément «3» du tube Nixie V1 à la broche centrale sur le dessus de K4 ((la modification ne s'applique qu'aux PCB existants !)



Figure 2. Élimination d'un tronçon de piste de PCB pour déconnecter un néon de son pilote (la modification ne s'applique qu'aux PCB existants !)

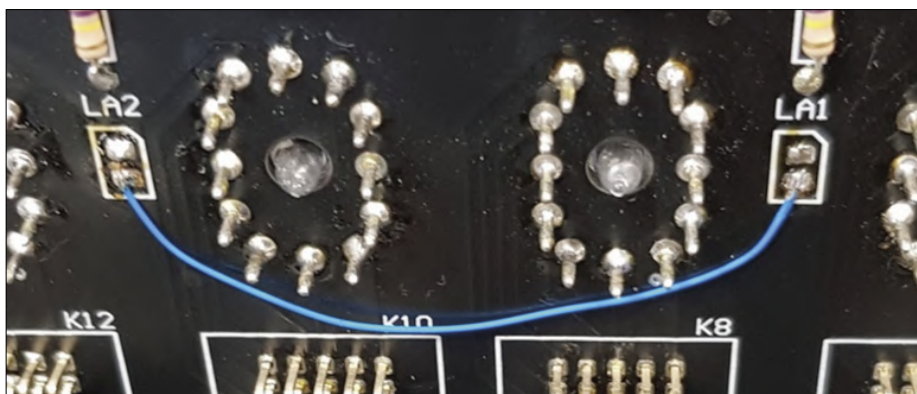


Figure 3. Interconnexion des lampes au néon avec un fil (la modification ne s'applique qu'aux PCB existants !)

cas des autres chiffres. Pour réduire le risque d'empoisonnement, le nouveau logiciel fait passer chaque tube par tous ses numéros de temps en temps. Le premier nettoyage intervient à la mise sous tension puis se répète toutes les 30 minutes (à 15 et 45 minutes après l'heure). Lors du nettoyage préventif, s'il est installé, le rétroéclairage à WS2812 clignotera et fera défiler les couleurs. Cette option peut être désactivée dans les réglages de l'horloge.

Modification du son du réveil

Statut : hautement recommandé.

Un signal sonore est essentiel comme retour d'information au cours du paramétrage, en tout cas pour la fonction de réveil. Il faut pour cela un ronfleur à faible courant qu'on trouvera chez divers fournisseurs : avec votre moteur de recherches habituel, cherchez *buzzer piezo 3-24V* et sélectionnez parmi les résultats un candidat dont la consommation ne dépasse pas 10 mA (le PIC ne peut fournir que 25 mA sur sa sortie). Respectez la polarité du ronfleur en

reliant son fil noir à la broche 1 de JP1 (au bord du PCB) et son fil rouge à la broche du côté du PIC. R14 peut être dessoudée ou coupée de la carte. Percez un petit trou à l'arrière du boîtier pour permettre la sortie du son émis par le ronfleur que vous collerez.

Modification de l'affichage de la date

Statut : facultatif.

Cette modification est nécessaire pour afficher la date au format européen (jj/mm/aa), car un 3 est nécessaire pour afficher les jours sur les chiffres les plus à gauche. Le circuit imprimé est conçu pour alimenter les deux néons séparément, et le chiffre «3» du tube de Nixie gauche n'est pas connecté. Il est modifié pour relier les deux lampes au néon à un pilote, puis pour relier le «3» au pilote maintenant inutilisé.

1^{ère} étape : déconnectez un néon de son pilote. À l'aide d'une lame tranchante, sectionnez la piste inférieure côté tube du circuit imprimé. Faites une deuxième coupure de la même piste environ 1 mm

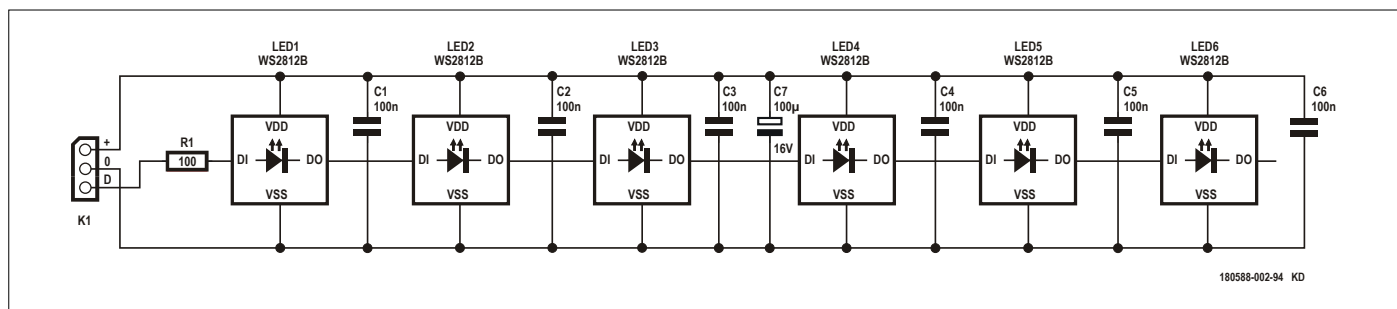


Figure 5. Schéma du circuit de rétroéclairage décoratif avec LED CMS. Les condensateurs de découplage suppriment le bruit causé par le courant relativement élevé.

plus loin, de manière à pouvoir enlever un tronçon (fig. 2).

2^e étape : Reliez les néons entre eux (fig. 3).

3^e étape : Connectez le «3» à la broche centrale, rangée du haut, de K4 (fig. 4).

WS2812 Rétroéclairage LED

«Neopixel

Statut : facultatif.

Attention : L'intensité du courant des LED alimentées sous 5 V est assez forte. Vous pouvez utiliser un second régulateur de 5 V alimenté par l'entrée 9 V, ou obtenir l'alimentation, la masse et les données à partir du connecteur de programmation. Si vous prélevez l'alimentation sur le connecteur de programmation, il faudra équiper IC6 d'un radiateur, ou mieux, opter pour un modèle à découpage (p. ex. Murata OKI-78SR-5/1.5-W36-C).

Le schéma du rétroéclairage est sur la fig. 5. Comme la broche RB7 n'est utili-

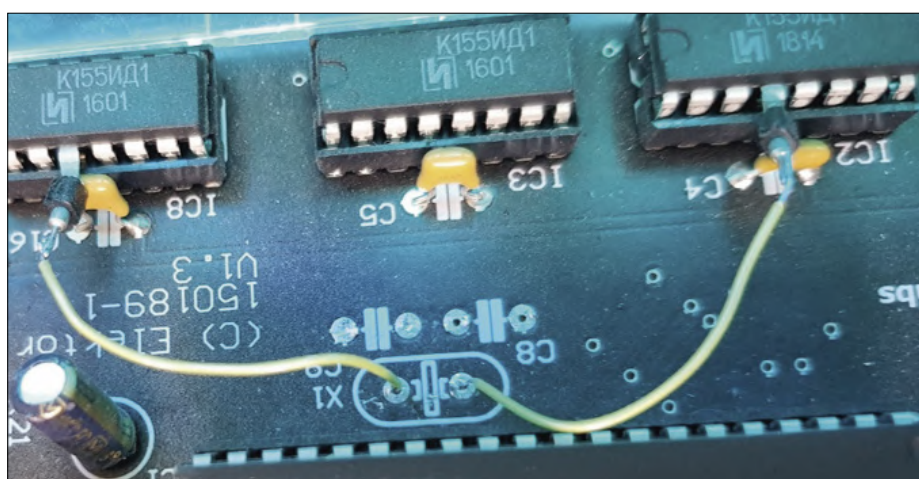


Figure 6. Les connexions du quartz libérées sur le circuit imprimé sont reliées aux broches du circuit intégré de commande tordues sur le côté (la modification ne s'applique qu'aux PCB existants)

sée dans le projet que pour la programmation, un petit code a été ajouté pour lui permettre de piloter les LED du rétroéclairage – le signal de pilotage sort sur

le connecteur K1, broche 4.

Les LED CMS WS2812 sont montées sur le circuit imprimé avec un condensateur de 10 µF/16 V au milieu du circuit et



LISTE DES COMPOSANTS

Carte de rétroéclairage 180588-1

Résistance :

R1 = 100 Ω, film épais, 5%, 0,1 W, 150 V, 0805

Condensateurs

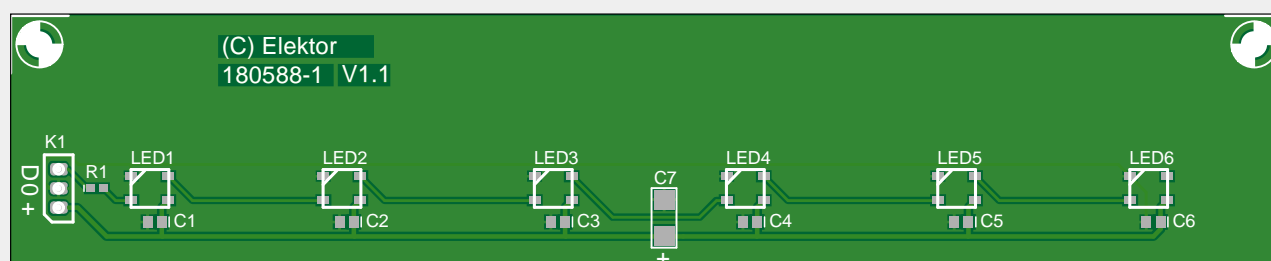
C1,C2,C3,C4,C5,C6 = 100 nF, 50 V, X7R, 0805
C7 = 100 µF, 16 V tantale, 10%, boîtier 2312

Semi-conducteurs

LED1 à LED6 = WS2812B LED RGB (CMS)

Divers

Câble à trois brins, d'une longueur d'environ 10 cm
Connecteur SIL à 3 voies
Circuit imprimé 180588-1 V1.1





LISTE DES COMPOSANTS

Circuit imprimé de l'horloge Nixie révisée 150189-1 V2.1

Résistances

R1,R2,R4,R5,R25,R26 = 27 k Ω 5% 250 mW
R6,R7,R8,R9,R11,R15,R23 = 10 k Ω 5% 250 mW
R3,R13,R24 = 470 k Ω 5% 250 mW
R12,R16 = 1 k Ω 5% 250 mW
R17 = 5 k Ω 6 5% 250 mW
R27 = 0,27 Ω , 5%, 2 W
R28 = 47 k Ω 5% 1 W
P1 = aj.500 k Ω

Inductance

L1 = 100 μ H, 1,5 A, radial, DxL 8x8 mm

Condensateurs

C3 = 470 pF, 50 V, Y5U
C1,C4,C5,C6,C7,C15,C16,

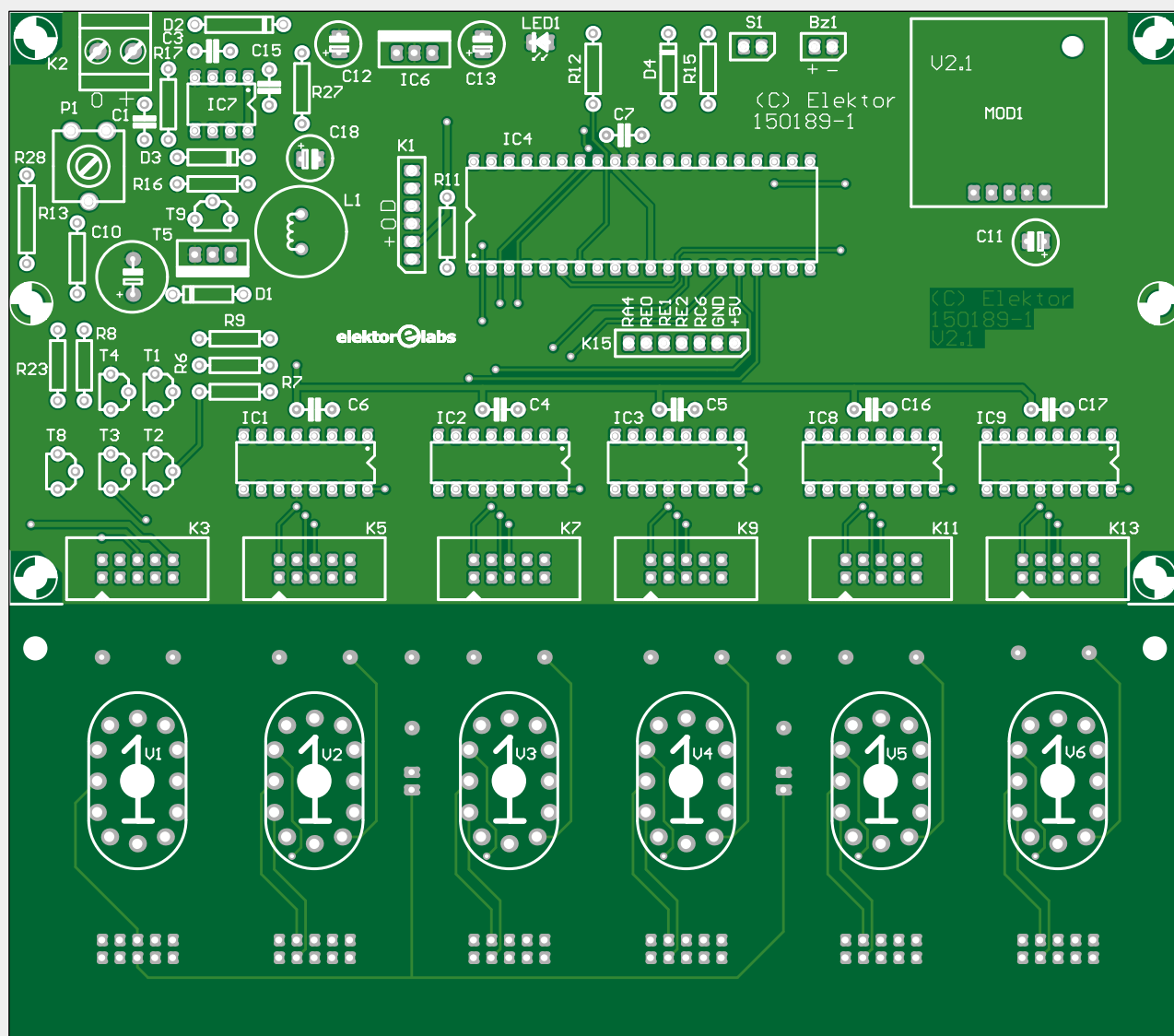
C17 = 100 nF 50 V 20%
C10 = 10 μ F, 250 V, 20%, radial, pas de 5 mm
C11,C12 = 100 μ F, 25 V, pas de 3,5 mm, 8x11 mm
C13 = 10 μ F, 25 V, pas de 2 mm, 5x1 mm
C18 = 220 μ F, 25 V, pas de 5 mm, 10x16 mm

Semi-conducteurs

D1 = BYV26, diode ultrarapide, 600 V, 1 A
D2 = 1N5814, diode Schottky, 40 V, 1 A
D3,D4 = BAT46, diode Schottky 100 V, 150 mA
LED1 = rouge, 3 mm
T1,T2,T3,T4,T8 = MPSA42, NPN, 300 V, 500 mA
T5 = IRF644PBF, NMOS, 250 V, 14 A
T9 = BC557 C, -45 V, -100 mA, 500 mW, h_{fe} =400
IC1,IC2,IC3,IC8,IC9 = K1551D1 (74141)
IC4 = PIC18F45K22, programmé (Elektor : 180588-41)
IC6 = OKI-78SR-5/1,5-W36-C, convertisseur CC/CC 5 V, 1,5 A, Murata
IC7 = contrôleur CC/CC MC34063

Divers

Bz1 = ronfleur 1,5-20 V_{cc} p. ex. LD-BZPG-2312
S1 = bouton-poussoir SPST, 20 V (4,95 mm)
K1 = barrette SIL à 6 voies, pas de 0,1 pouce
K2 = bornier à vis à 2 voies pour circuit imprimé, pas de 2,54 mm, 630 V
K3,K5,K7,K9,K11,K13 = embase à 10 broches (2x5), à sécable, verticale
K4,K6,K8,K10,K12,K14 = connecteur à 10 broches (2x5), sécable, coudé
K1 = SIL à 6 broches, pas de 0,1 pouce
LA1,LA2 = T1.1/4, lampe au néon, à fil
MOD1 = Module GPS, avec antenne GY-NEO6MV2
V1,V2,V3,V4,V5,V6 = tube Nixie IN-12
Circuit imprimé 150189-1 V2.1 de Elektor Store



des condensateurs de 100 nF sur les broches d'alimentation de chaque LED. La première LED possède une résistance de 100 Ω en série avec le fil de données RB7. Vous pouvez construire votre propre carte de rétroéclairage avec un morceau de circuit à pastilles et des versions traversantes du WS2812. Au labo nous avons préféré les CMS non seulement parce que ce type de LED est plus facile à aligner, mais aussi parce que leur profil plat s'intègre mieux dans le boîtier en acrylique sans contrainte mécanique.

Gradation

Statut : recommandé.

Les décodeurs compatibles avec le 74141 affichent un blanc pour les valeurs d'entrée supérieures à 9. Pour faire clignoter, et donc réduire la luminosité de tous les chiffres, deux puces doivent avoir leur «bit 3» (broche 4) connecté au processeur au lieu de la masse.

Si la gradation est activée dans les réglages de l'horloge sans que cette modification soit faite, les chiffres, au lieu de s'éteindre, afficheront un «7». Le quartz doit être dessoudé. Chauffez une extrémité à la fois à tour de rôle et soulevez doucement X1, les deux condensateurs C8 et C9 doivent également être retirés.

Deux des puces de décodage, IC2 et IC8, sont retirées. La broche 4 est soigneusement redressée, puis les circuits intégrés sont réinsérés. Attention de ne pas tordre de broches sous le circuit intégré lors de la réinsertion ! Utilisez un support de circuit intégré à tulipe ou un câble de raccordement femelle pour vous connecter à la puce – **ne soudez pas**, car la soudure provoque des fuites visibles pendant la gradation.

Reliez les broches aux connexions du quartz maintenant libres avec un fil à brin unique (fig. 6).

Module GPS

Statut : recommandé.

Au fil des mois, des lecteurs qui préféreraient acheter les composants eux-mêmes et séparément, ont buté sur le module GPS A2035H de *Maestro Wireless Solutions*. Ce composant n'est plus produit. Heureusement, *Maestro* propose le A2235H, un GPS plus récent qui tient sur l'ancienne carte bien que le nouveau module soit plus petit que son prédécesseur. Il lui faut toutefois deux résistances de polarisation supplémen-

taires de 2,2 k Ω des broches I²C à la borne 1,8 V du module. Pour éviter tout court-circuit avec le blindage du module, il est recommandé de placer un isolant adéquat (Kapton) entre la carte et le module [2].

Le débit par défaut de la sortie série des deux modules GPS *Maestro* est de 4800 bauds. De nombreux autres modules existent, mais ils fonctionnent généralement à 9600 bauds. Le logiciel a été remanié de sorte que l'horloge dispose désormais d'une détection automatique du débit en bauds sur le module GPS.

Mode d'emploi

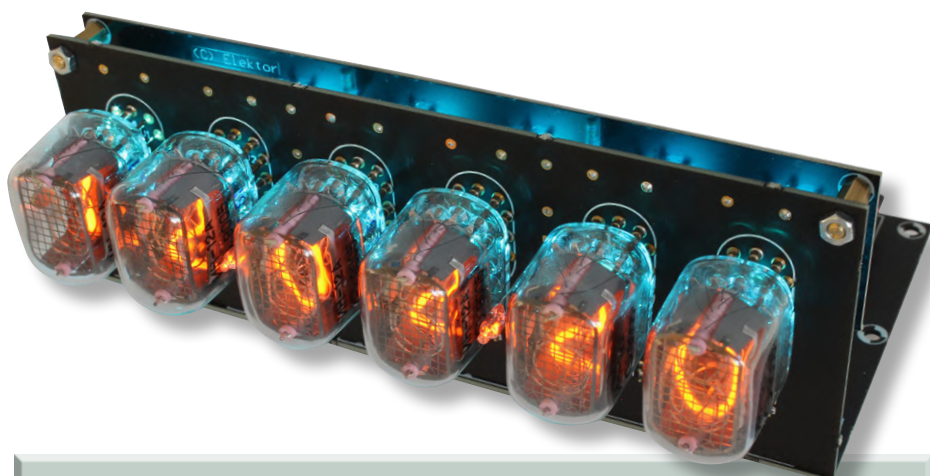
Avec le nouveau logiciel, il n'y a qu'un seul bouton pour commander l'horloge par pressions courtes qui modifient l'affichage ou le paramètre ; une pression longue change le mode, en fonction de l'affichage en cours. Toutes les options,

réglages et paramètres disponibles pour configurer et régler la nouvelle horloge Nixie révisée se trouvent dans le manuel d'Andy [2].

Conclusion

La version de 2016 a fait ses preuves, son fonctionnement est irréprochable, et pourtant cette nouvelle version décrite ici offre de nouvelles fonctions qui méritent la mise à jour. À vous de juger si vous effectuerez aussi les modifications du matériel... ou seulement une partie. Les auteurs vous recommandent chaudement la réception GPS améliorée, la fonction anti-empoisonnement et le rétroéclairage multicolore ! ◀

180588-03



@ WWW.ELEKTOR.FR

→ Nouvelle horloge Nixie de précision (révisée) PCB nu, 150189-1 V2.1
www.elektor.fr/180589-1

→ Nouvelle horloge Nixie de précision (révisée) PCB nu 180588-1 V1.1
www.elektor.fr/180588-1

→ μ C programmé, 180588-41
www.elektor.fr/180588-41

→ module GPS
www.elektor.fr/open-smart-gps

Liens

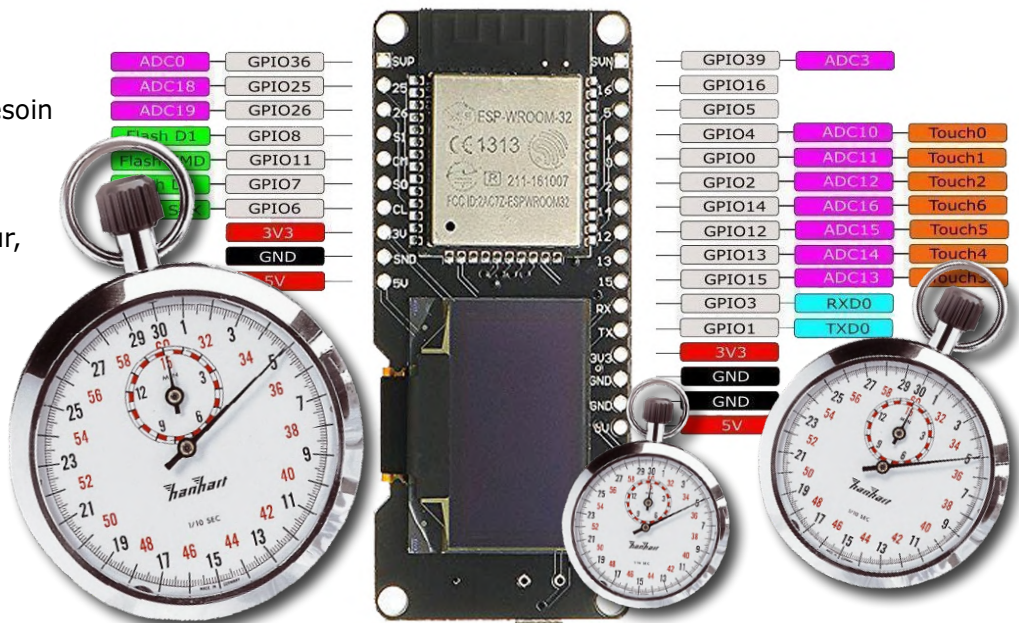
- [1] nouvelle horloge Nixie pilotée par GPS d'Elektor, mai & juin 2016 : www.elektormagazine.fr/magazine/elektor-201605/28938
- [2] page du projet sur Elektor Labs : www.elektormagazine.com/labs/150189-6-digit-nixie-clock
- [3] page de cet article : www.elektormagazine.fr/180588-01

multitâche en pratique avec l'ESP32 (3)

Temporisateurs logiciels

Warren Gay (Canada)

Les développeurs sur microcontrôleur ont souvent besoin de temporisateurs, pour les retransmissions et autres procédures de reprise sur erreur, la suppression des rebonds des poussoirs, ou un simple clignotement. L'ESP32 et l'EDI Arduino fournissent des temporisateurs logiciels qui utilisent la bibliothèque FreeRTOS intégrée. Intéressons-nous à la fonction de temporisation.



Pour montrer comment fonctionnent les temporisateurs de FreeRTOS, notre démonstration visera à construire une classe de LED d'alerte. Il ne s'agit néanmoins pas d'une LED clignotante ordinaire. Lorsqu'elle est activée, elle clignote rapidement cinq fois puis s'éteint pour le reste de la période et recommence. Visuellement, cette alerte attirera l'attention comme le ferait une sonnerie de téléphone.

La classe `AlertLED` va utiliser en interne un temporisateur logiciel FreeRTOS pour chaque pilote de LED. S'agissant d'une classe, vous pourrez en ajouter autant que nécessaire, sans surcharger la mémoire de l'ESP32. Le **listage 1** montre le programme complet [2]. La ligne 100 montre la création d'une instance de LED d'alerte avec une simple instruction de déclaration C++ :

```
static AlertLED alert1(GPIO_LED,1000);
```

Le nom de l'instance de classe est ici `alert1`, elle utilise la LED spécifiée (GPIO 12 en ligne 5), et la période de clignotement est de 1000 ms. Nous pourrions facilement en ajouter une autre pour le GPIO 13, juste en ajoutant la ligne :

```
static AlertLED alert2(13,1000);
```

Nous consacrerons cette démonstration à un seul indicateur, mais la structure de classe C++ en facilite l'usage.

La classe AlertLED

La définition de la classe est reproduite ci-dessous par commodité. Les données membres de la classe comprennent le membre `thandle` (ligne 11), qui est un index vers le temporisateur FreeRTOS (type `TimerHandle_t`). Au début il prend la valeur `nullptr` (NULL en langage C). La donnée membre `state` contiendra l'état de la LED (ligne 12), active à l'état haut (`true` = on). La donnée membre `count` (ligne 13) sera utilisée comme un sous-état dans la fonction de retour, comme expliqué plus loin. Finalement, les membres `period_ms` et `gpio` définissent la période de clignotement en ms et le GPIO à piloter pour la LED (lignes 14 et 15).

```
0010: class AlertLED {
0011:     TimerHandle_t    thandle = nullptr;
0012:     volatile bool     state;
0013:     volatile unsigned count;
0014:     unsigned          period_ms;
0015:     int               gpio;
0016:
0017:     void reset(bool s);
0018:
0019: public:
0020:     AlertLED(int gpio,unsigned period_ms=1000);
0021:     void alert();
```

Listage 1 : Le programme alertled.ino [2].

```
0001: // alertled.ino
0002: // MIT License (see file LICENSE)
0003:
0004: // La LED est active à l'état haut
0005: #define GPIO_LED      12
0006:
0007: //
0008: // Classe AlertLED pour pilotage de la LED
0009: //
0010: class AlertLED {
0011:     TimerHandle_t      thandle = nullptr;
0012:     volatile bool      state;
0013:     volatile unsigned   count;
0014:     unsigned            period_ms;
0015:     int                 gpio;
0016:
0017:     void reset(bool s);
0018:
0019: public:
0020:     AlertLED(int gpio,unsigned period_ms=1000);
0021:     void alert();
0022:     void cancel();
0023:
0024:     static void callback(TimerHandle_t th);
0025: };
0026:
0027: //
0028: // Constructeur :
0029: // gpio      broche GPIO pour activer la LED
0030: // period_ms période complète en ms
0031: //
0032: AlertLED::AlertLED(int gpio,
0033:                     unsigned period_ms) {
0034:     this->gpio = gpio;
0035:     this->period_ms = period_ms;
0036:     pinMode(this->gpio,OUTPUT);
0037:     digitalWrite(this->gpio,LOW);
0038: }
0039: //
0040: // Méthode interne
0041: //      pour réinitialiser les valeurs
0042: //
0043: void AlertLED::reset(bool s) {
0044:     state = s;
0045:     count = 0;
0046:     digitalWrite(this->gpio,s?HIGH:LOW);
0047: }
0048: //
0049: // Méthode pour démarrer l'alerte :
0050: //
0051: void AlertLED::alert() {
0052:
0053:     if ( !thandle ) {
0054:         thandle = xTimerCreate(
0055:             "alert_tmr",
0056:             pdMS_TO_TICKS(period_ms/20),
0057:             pdTRUE,
0058:             this,
0059:             AlertLED::callback);
0060:         assert(thandle);
0061:     }
0062:     reset(true);
0063:     xTimerStart(thandle,portMAX_DELAY);
0064: }
0065:
0066: //
0067: // Méthode pour arrêter une alerte :
0068: //
0069: void AlertLED::cancel() {
0070:     if ( thandle ) {
0071:         xTimerStop(thandle,portMAX_DELAY);
0072:         digitalWrite(gpio,LOW);
0073:     }
0074: }
0075:
0076: // Méthode statique, agissant comme callback
0077: // du temporisateur :
0078: //
0079: void AlertLED::callback(TimerHandle_t th) {
0080:     AlertLED *obj = (AlertLED*)
0081:         pvTimerGetTimerID(th);
0082:     assert(obj->thandle == th);
0083:     obj->state ^= true;
0084:     digitalWrite(obj->gpio,obj->state?HIGH:LOW);
0085:
0086:     if ( ++obj->count >= 5 * 2 ) {
0087:         obj->reset(true);
0088:         xTimerChangePeriod(th,pdMS_TO_TICKS
0089:             (obj->period_ms/20),portMAX_DELAY);
0090:     } else if ( obj->count == 5 * 2 - 1 ) {
0091:         xTimerChangePeriod(th,
0092:             pdMS_TO_TICKS(obj->period_ms/20+
0093:                 obj->period_ms/2),
0094:                 portMAX_DELAY);
0095:         assert(!obj->state);
0096:     }
0097: }
0098: // Objets globaux
0099: //
0100: static AlertLED alert1(GPIO_LED,1000);
0101: static unsigned loop_count = 0;
0102:
0103: //
0104: // Initialisation:
0105: //
0106: void setup() {
0107:     // delay(2000);
0108:     // pour permettre la connexion USB
0109:     alert1.alert();
0110: }
0111:
0112: void loop() {
0113:     if ( loop_count >= 70 ) {
0114:         alert1.alert();
0115:         loop_count = 0;
0116:     }
0117:     delay(100);
0118:
0119:     if ( ++loop_count >= 50 )
0120:         alert1.cancel();
0121: }
```

```

0022: void cancel();
0023:
0024: static void callback(TimerHandle_t th);
0025: };

```

La classe comporte aussi quelques appels à des méthodes publiques comme :

- `alert()` - active l'indicateur à LED (ligne 21)
- `cancel()` - désactive l'indicateur à LED (ligne 22)

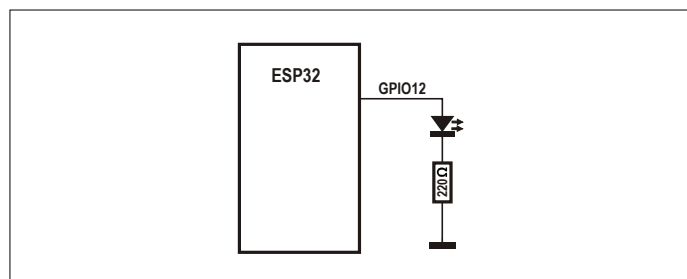


Figure 1. Le câblage du programme de démo *alertled.ino*.

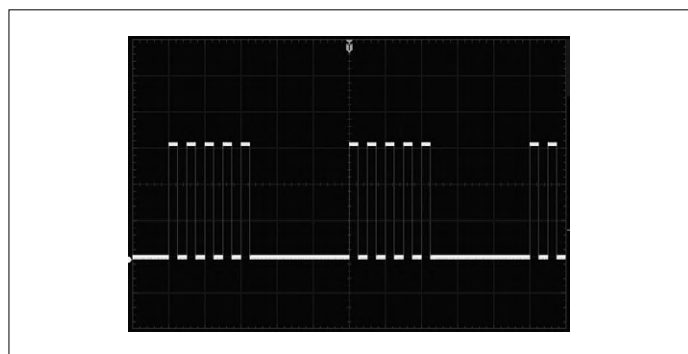


Figure 2. Le signal de pilotage de la LED pour *alertled.ino*, en abscisse 200 ms/div.

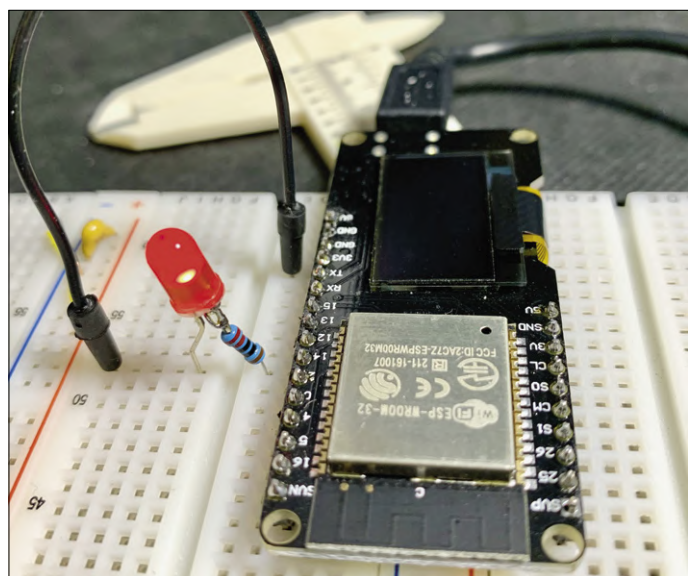


Figure 3. ESP32, LED et résistance de 220 Ω câblés pour la démo *alertled.ino*.

Il y a aussi une méthode C++ de type `static` nommée `callback()` à la ligne 24. Nous y reviendrons.

La méthode `AlertLED::alert()`

Les méthodes `alert()` et `callback()` sont les parties les plus intéressantes de la classe. Nous avons vu que l'index a été initialisé à `nullptr`. Lorsque la méthode `alert()` est appelée pour la première fois, elle va trouver que la valeur est nulle en ligne 53, ce qui entraîne la création du temporisateur.

Le temporisateur logiciel est créé avec `xTimerCreate()` aux lignes 54 à 59, ce qui retourne l'index du temporisateur nouvellement créé. La macro `assert()` à la ligne 60 s'assure du succès de la création. Les arguments de `xTimerCreate()` sont :

1. Un nom convivial pour notre temporisateur (ici c'est juste « `alert_tmr` »). FreeRTOS ne l'utilise que pour quelques fonctions de recueils statistiques. Il doit normalement être unique mais peut ne pas l'être si vous ne faites pas de recherches par nom.
2. Une période de temporisation en tops d'horloge système (convertis en ms à la ligne 56 avec la macro `pdMS_TO_TICKS()`).
3. Une indication si le temporisateur est simple (valeur `pdFALSE`) ou réarmable (valeur `pdTRUE`). Notre exemple crée un temporisateur réarmable (ligne 57).
4. `this` est un « identifiant de temporisateur » dans la terminologie FreeRTOS. On pourrait le voir comme un « paramètre de donnée utilisateur ». C'est un pointeur non typé, qui dans cet exemple est l'adresse de l'objet de classe qui utilise le mot réservé `this` en C++ (ligne 58).
5. L'adresse de la fonction de retour. Nous fournissons à la ligne 59 le nom de la méthode statique `AlertLED::callback()`. C'est la fonction qui sera appelée lorsque le temporisateur déclenche.

Certaines données membres sont réinitialisées par l'appel à la méthode interne `reset()` à la ligne 62, puis vient le démarrage du temporisateur à la ligne 63 avec `xTimerStart()`. Tous les temporisateurs sont créés « inactifs » jusqu'à ce qu'ils soient démarrés/redémarrés. Une fois démarrés, ils passent dans l'état « en cours ».

La méthode statique `AlertLED::callback()`

La classe `AlertLED` définit un appel à une méthode statique appelée `callback()`. Pour qui n'est pas familier avec le C++, cela signifie que cette méthode est juste une fonction. Elle n'a donc pas de pointeur d'objet propre (pas de pointeur « `this` »). C'est la même chose qu'une fonction C excepté son nom bizarre en C++ et le fait qu'elle bénéficie d'un accès spécial aux éléments internes de l'objet de classe.

Le temporisateur logiciel requiert un argument de type `TimerHandle_t` (ligne 79). Cela nous donne accès à l'index du temporisateur qui est déclenché mais il nous faut plus d'information — dans ce cas l'instance de LED qui doit être pilotée. La ligne 80 l'obtient avec la valeur du « Timer ID » - l'adresse de l'instance (`alert1`) de la classe (LED) avec l'appel à `pvTimerGetTimerID()`, qui nécessite l'index du temporisateur. Nous avons en retour la valeur fournie à la ligne 58, lorsque le temporisateur a été créé. Nous pouvons maintenant accéder à l'objet LED et à ses membres par la variable `obj`.

Les lignes 83 et 84 basculent le GPIO spécifié pour la LED.

Lorsque la valeur de `count` est incrémentée (ligne 86), et jusqu'à ce qu'elle atteigne 9, on ressort simplement de la fonction. Ceci produit le clignotement rapide simple pour la première moitié de la période d'alerte.

Lorsque `count` atteint 9, les lignes 90 à 93 sont exécutées. La fonction `xTimerChangePeriod()` modifie la période de temps utilisée par le temporisateur que nous avons créé. Notez l'ajout de `period_ms/20` à `period_ms/2` pour compenser la dernière moitié d'extinction du cycle de clignotement rapide.

Finalement, au dixième passage dans la fonction, les lignes 87 et 88 sont exécutées. Les données membres sont réinitialisées par appel à la méthode interne `reset()` (ligne 87), et la période du temporisateur rétablie pour un clignotement rapide à la ligne 88. Comme la donnée membre `count` est remise à zéro, le cycle complet recommence. Le caractère réarmable du temporisateur maintiendra la répétition du scénario jusqu'à l'appel de `AlertLED::cancel()`.

La méthose `AlertLED::cancel()`

On utilise la méthode `AlertLED::cancel()` pour arrêter le clignotement de l'alerte. Pour cela, on appelle `xTimerStop()` (ligne 71), et la LED est désactivée en écrivant LOW sur le GPIO de la LED à la ligne 72.

Nécessité de *volatile*

Le lecteur attentif aura noté que nous avons utilisé le mot clé C `volatile` aux lignes 12 et 13 de la définition de classe. Pourquoi est-ce nécessaire ? Ceci informe le compilateur que chaque fois qu'il accède aux valeurs des données membres `state` ou `count`, il doit aller directement à la mémoire de l'instance de classe (`alert1` dans notre démo) plutôt que de se fier à une valeur qui pourrait subsister dans un registre. Il y a deux tâches qui interagissent avec notre classe :

- la classe démon de FreeRTOS (via la fonction de retour)
- votre tâche qui invoque les méthodes `alert()` or `cancel()` de la classe.

Si ces valeurs n'étaient pas marquées comme *volatile*, une des tâches ou les deux pourraient utiliser des valeurs qui sont encore dans un registre, car c'est normalement le plus efficace. Toutefois, les valeurs en mémoire peuvent avoir été modifiées par l'autre tâche. Pour cette raison le mot clé *volatile* force le compilateur à ignorer cette possible optimisation et à générer à la place un code qui va directement chercher les valeurs en mémoire.

La démonstration

Pour une simple démo `alertled.ino`, nous pouvons de nouveau utiliser le Lolin ESP32 OLED Display Module. La **fig. 1** montre le câblage. La LED est câblée selon la configuration active-haut. Pour démontrer l'activation et l'annulation de la classe `AlertLED`, la fonction `loop()` maintient une variable de comptage nommée `loop_count` (ligne 101). La fonction `setup()` active initialement la LED, mais lorsque le décompte de boucle atteint 50, l'alerte est annulée à la ligne 120. Lorsque plus tard le décompte

atteint 70, l'alerte est réactivée à la ligne 113 et le compteur est remis à zéro.

Une fois le programme flashé et exécuté sur l'ESP32, vous devriez voir la LED d'alerte clignoter de façon à attirer l'attention. Elle s'éteindra quelques instants après, juste pour repartir de nouveau. La **fig. 2** monte le signal de pilotage de la LED. La **fig. 3** montre la configuration, avec une combinaison LED-résistance et un fil Dupont vers GND.

Limitations de FreeRTOS

Nous avons jusqu'ici appliqué l'API (Interface de Programmation d'Application) du temporisateur logiciel, sans trop parler de certaines limitations importantes de la fonction de retour. Parmi ces limitations :

- Ne jamais faire d'appel bloquant à l'intérieur d'une fonction de retour (comme `delay()` ou une mise en file bloquante par exemple).
- Évitez les longues durées d'exécution (cela perturbera l'API).
- Évitez d'utiliser trop d'espace de pile (l'Arduino Esp32 est à priori limité autour de 1500 octets). Des fonctions comme `printf()` et `snprintf()` exigent souvent une taille de pile considérable et devraient donc être évitées dans la fonction de retour.
- Certaines des fonctions API de FreeRTOS fonctionnent au moyen d'une file interne. C'est pourquoi notre programme de démo a utilisé `portMAX_DELAY` aux lignes 88 et 92. La profondeur de la file de temporisateur pour l'Arduino ESP32 étant de 10, il est peu probable qu'elle se remplisse, à moins que de nombreux temporisateurs soient actifs en même temps.

Résumé

Pourquoi ne pas simplement utiliser une tâche pour gérer chaque LED d'alerte ? La raison principale est que chaque tâche nécessite une pile et un bloc de contrôle (TCB). Si vous deviez allouer 32 indicateurs à LED, cela demanderait 32 x (356 + 1500) octets, soit 59392 octets ! Comparez cela avec l'approche à temporisateur FreeRTOS qui prend 40 octets pour chaque temporisateur, pour un total de 1200 octets (l'objet de type `StaticTimer_t` fait 40 octets). La pile est partagée avec la tâche démon FreeRTOS pour tous les temporisateurs (autrefois appelée tâche *timer service*).

Les applications peuvent mélanger les approches pour des besoins spécifiques, mais bien souvent un temporisateur FreeRTOS répond aux exigences. ◀

(200071-03 VF Denis Lafourcade)

Liens

- [1] Multitâche en pratique avec l'ESP32, Elektor Magazine 1/2020 : www.elektormagazine.fr/magazine/190182-03
- [2] Code source du projet : https://github.com/ve3wwg/esp32_freertos/blob/master/alertled/alertled.ino



SigFox : un renard sur l'internet des objets (4)

Composition d'un tableau de bord

Frank Schleking & Bernd vom Berg (Allemagne)

Notre courte série d'articles se terminera par la création d'un tableau de bord personnalisé pour afficher sous forme de graphiques et de chiffres les mesures envoyées sur le réseau Sigfox. Il existe pour cela un outil gratuit de tableau de bord basé sur la navigation HTML. Il permet à l'utilisateur de créer et d'utiliser une interface de visualisation et de surveillance personnalisée sur n'importe quel PC, ordinateur portable, tablette ou téléphone tactile.

Dans les précédents épisodes de cette série, nous avons examiné le matériel Sigfox de base pour acquérir et transmettre des données mesurées, puis les concepts qui sous-tendent la transmission de données dans le réseau Sigfox. Nos mesures se trouvent maintenant dans l'infrastructure Sigfox et attendent d'être collectées et traitées. Il existe un outil gratuit de tableau de bord basé sur la navigation HTML. Il permet à l'utilisateur de créer et d'utiliser une interface de visualisation et de surveillance personnalisée sur n'importe quel PC, ordinateur portable, tablette ou téléphone tactile. Et, s'il est correctement activé, ce tableau de bord peut être appelé et visualisé par n'importe quel utilisateur autorisé sur l'internet, partout dans le monde. Le point de départ des développements décrits ici est le croquis Arduino Sigfox-3_0.ino pour la carte MKR FOX 1200. Si vous choisissez le quatrième point du menu Send three readings (*envoyer trois relevés*), les trois grandeurs mesurées (pression atmosphérique, température de l'air et intensité lumineuse captée par la LDR) seront acquises en continu, les résultats étant affichés sur l'ePaper de la carte mère et transmis à l'IDE

Arduino par le moniteur série. Lorsque le bouton est pressé, ces relevés sont automatiquement envoyés sur le réseau Sigfox, reçus par l'infrastructure Sigfox et stockés. Et à partir de ce moment, nous pouvons procéder à la visualisation des données sur un tableau de bord.

Plateforme du tableau de bord *thinger.io*

Nous utiliserons *thinger.io*, une plateforme de visualisation de l'IdO à code source ouvert qui permet la construction rapide et simple d'affichages de données clairs et attrayants. Elle peut même être utilisée gratuitement pour des projets plus simples. Pour l'instant, le système ne prend pas en charge les opérations mathématiques sur les données reçues, ce qui signifie que les relevés doivent lui être fournis sous une forme traitée et prête à être affichée. Ce traitement préalable peut être effectué facilement par notre logiciel Arduino sur la carte MKR FOX 1200. Le modèle de visualisation des valeurs des données est basé sur un tableau de bord à la disposition de toute personne disposant d'un navigateur web.

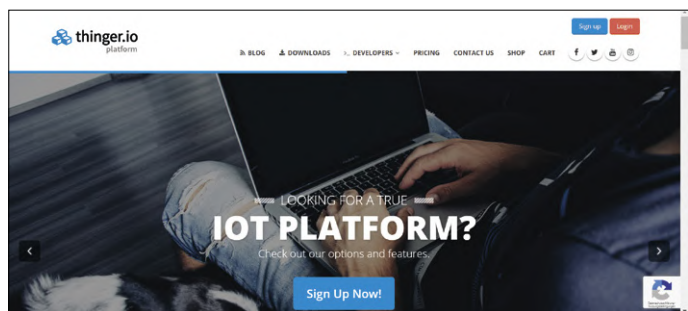


Figure 1. Page d'accueil de *thinger.io*.

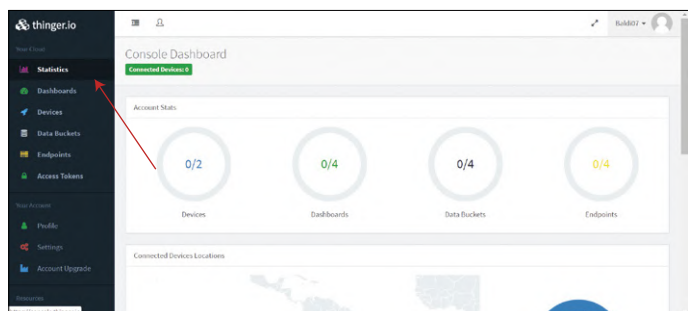


Figure 3. Écran principal du compte affichant les statistiques.

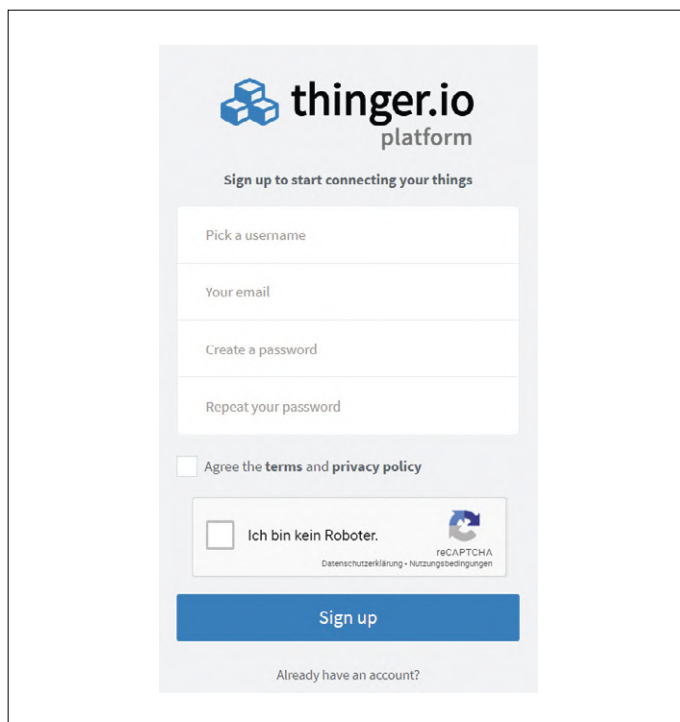


Figure 2. Création d'un nouveau compte.

Pas à pas

Quelques étapes simples sont nécessaires pour créer un tableau de bord personnalisé comme suit.

- Créez un compte d'utilisateur gratuit sur *thinger.io*.
- Créez un «bac de données» avec *thinger.io* pour accumuler les relevés envoyés par l'infrastructure Sigfox.
- Créez un point d'accès et un jeton avec *thinger.io*, et avec celui-ci, configurez l'infrastructure Sigfox de sorte qu'elle dispose des identifiants nécessaires pour transférer des données à *thinger.io*.
- Créez un système de rappel sur l'infrastructure de Sigfox pour envoyer des données sur l'internet depuis l'infrastructure vers le bac de données de *thinger.io*.
- Dessinez sur le site *thinger.io* un tableau de bord graphique adapté pour visualiser les données.

Création d'un compte utilisateur sur *thinger.io*

Un compte d'utilisateur (gratuit) doit être créé sur *thinger.io* pour pouvoir utiliser ses services. Pour ce faire, rendez-vous sur la page d'accueil de *thinger.io* [1] (**fig. 1**) et cliquez sur *Sign up* en haut à droite. Vous accéderez alors au formulaire d'enregistrement d'un nouvel utilisateur (**fig. 2**). Remplissez les champs du

formulaire et cliquez sur *Sign up*.

Les services offerts aux utilisateurs disposant d'un compte gratuit sont quelque peu limités, mais néanmoins plus que suffisants pour la plupart des petits projets expérimentaux. Vous trouverez tous les détails sous l'onglet *Pricing*.

Une fois votre adresse électronique confirmée, vous pouvez

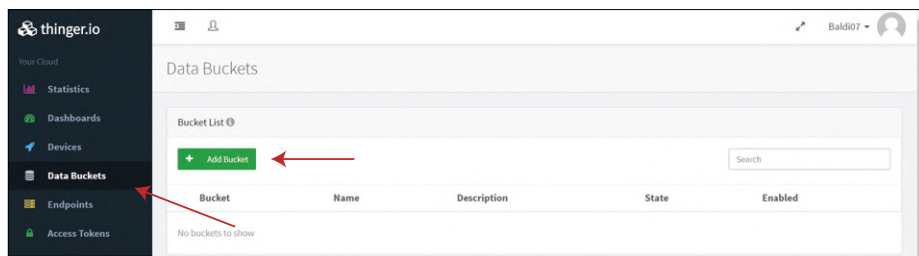


Figure 4. Mise en place d'un nouveau bac de données.

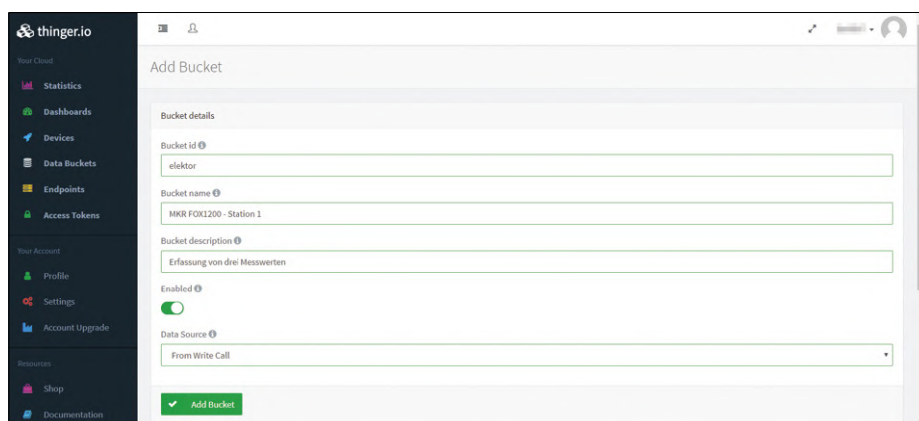


Figure 5. Spécification des propriétés du bac de données.

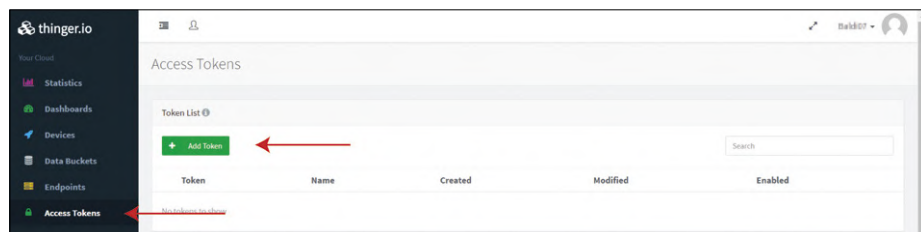


Figure 6. Mise en place d'un nouveau jeton (point d'accès).

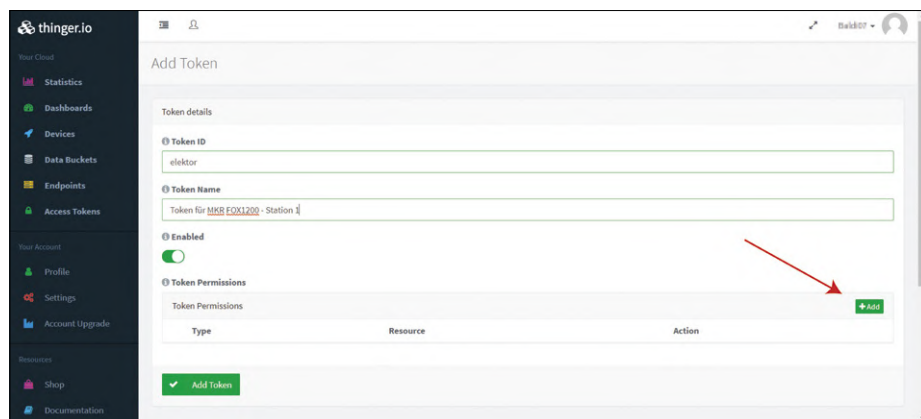


Figure 7. Spécification des propriétés du jeton.

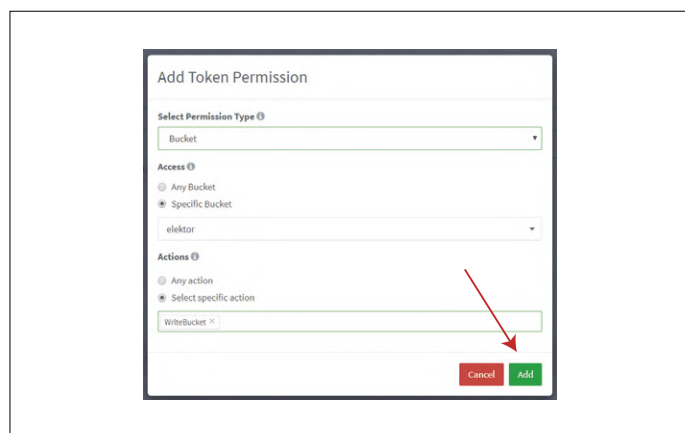


Figure 8. Configuration des autorisations de jetons.

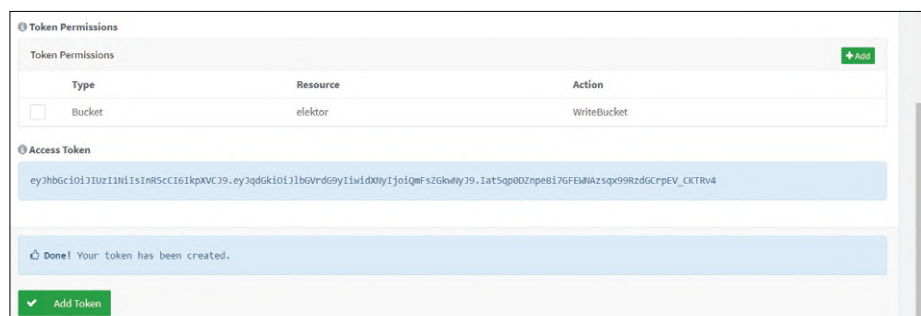


Figure 9. Le jeton est créé et activé avec les autorisations appropriées.

vous rendre sur la page de connexion et vous connecter. L'écran principal de votre compte s'appelle *Statistics* et c'est le point de départ (fig. 3). Le menu principal se trouve sur le côté gauche et contient toutes les fonctions nécessaires pour créer un tableau de bord pour votre projet IdO.

Création d'un bac de données chez *thinger.io*

Ensuite nous avons créé un **bac** (ou panier) **de données**, c'est-à-dire une base de données dans laquelle les relevés peuvent être stockés, et à partir de laquelle elles peuvent être relues. L'infrastructure Sigfox écrira automatiquement les valeurs mesurées dans cette base de données en utilisant le rappel (*callback*) Sigfox. Sur l'écran principal, cliquez sur *Data Buckets* sur la gauche, puis sur *Add Bucket* (ajouter un bac) (fig. 4). Une fenêtre apparaîtra pour vous permettre de configurer les caractéristiques du bac de données (fig. 5). L'identifiant *Bucket id* doit être unique pour l'ensemble du compte et sera utilisé ultérieurement lors de la configuration du bac pour fonctionner avec

Sigfox. Dans cet exemple, nous entrons «elektor». Les champs *Bucket name* et *Bucket description* peuvent contenir n'importe quel texte utile à l'utilisateur. Utilisez le menu déroulant dans le champ de sélection *Data source* (source de données) pour définir sa configuration sur *From Write Call* (à partir de l'appel d'écriture). Cela signifie que les données seront envoyées à *thinger.io* sur Internet à partir de l'infrastructure Sigfox en utilisant le rappel Sigfox. Dès que le rappel envoie les valeurs des données, elles seront stockées ici dans le bac. Enfin, cliquez sur *Add bucket* et le bac sera créé.

Mise en place d'un point d'accès chez *thinger.io* pour recevoir les données de l'infrastructure

L'infrastructure Sigfox doit bien sûr savoir où les mesures qu'il a acquises doivent être envoyées grâce à son rappel : en d'autres termes, nous devons indiquer à l'infrastructure Sigfox l'adresse du destinataire sur *thinger.io*. Cette étape se fait en créant un jeton avec les informations d'identification appropriées pour permettre l'accès au bac de données. Dans la fenêtre principale, cliquez sur *Access Tokens*, puis sur *Add Token* (fig. 6). Une fenêtre apparaîtra dans laquelle vous pourrez entrer les propriétés du jeton (fig. 7). Nous commençons par définir l'ID du jeton : pour cet exemple, nous choisissons «elektor». Sous la rubrique *Token Name*, vous pouvez ajouter une description du jeton en texte libre ou simplement un nom.

Cliquez ensuite sur *Add* (le petit bouton vert en bas à droite) et une autorisation de jeton (c'est-à-dire les informations d'identification pour accéder au bac de données) sera ajoutée. Dans l'étape suivante, vous pourrez configurer cela en détail (fig. 8).

Dans le champ *Select Permission Type*, choisissez *Bucket*. Dans le champ *Access*, cliquez sur *Specific Bucket* et choisissez ensuite «elektor» (l'identifiant que nous avons configuré) dans le menu déroulant. Sous *Actions*, choisissez *Select specific action* et dans le champ en dessous, choisissez *WriteBucket* dans le menu déroulant. Enfin, cliquez sur le bouton *Add* : cela va générer un jeton pour le bac cible que vous avez sélectionné (avec l'identifiant «elektor») et l'ajouter à la liste des permissions (fig. 7). Cliquez maintenant sur *Add Token* (fig. 7, en bas à gauche) et ce point d'accès, qui sera l'adresse de réception des messages de l'infrastructure Sigfox, sera créé dans *thinger.io* (fig. 9).

Notez que le champ bleu *Access Token*, qui contient les informations d'authentification générées par *thinger.io*, sera nécessaire pour l'infrastructure Sigfox. Celle-ci devra être configurée de manière à envoyer ce code d'accès à l'adresse de destination de *thinger.io* en même temps que les données utiles : cela garantira que les données utiles envoyées seront acceptées par *thinger.io* et correctement dirigées vers le bac de données «elektor». Il est donc nécessaire de copier-coller ce code d'accès et de le stocker en lieu sûr. Fin de la configuration des paramètres sur *thinger.io*. Retournons à l'infrastructure Sigfox.

Configuration du 'rappel' dans l'infrastructure Sigfox

Nous voulons faire en sorte que lorsque le MKR FOX 1200 envoie des données sur l'infrastructure Sigfox, les informations soient automatiquement transmises à *thinger.io* pour traitement et visualisation ultérieurs. Pour ce faire, nous utilisons une fonction de l'infrastructure Sigfox déjà mentionnée, sous le nom de *callback*. Un rappel n'est rien d'autre qu'un transfert automatique de données exécuté chaque fois que l'infrastructure Sigfox reçoit des données d'une station Sigfox telle que notre MKR FOX 1200. Pour permettre ce transfert, l'infrastructure doit bien sûr être correctement configurée, en s'assurant que les données sont envoyées au destinataire souhaité.

Afin de transférer les données utiles dans notre bac de données «elektor» sur *thinger.io*, nous devons utiliser notre jeton et le code d'accès que *thinger.io* a générés pour nous lors de l'étape

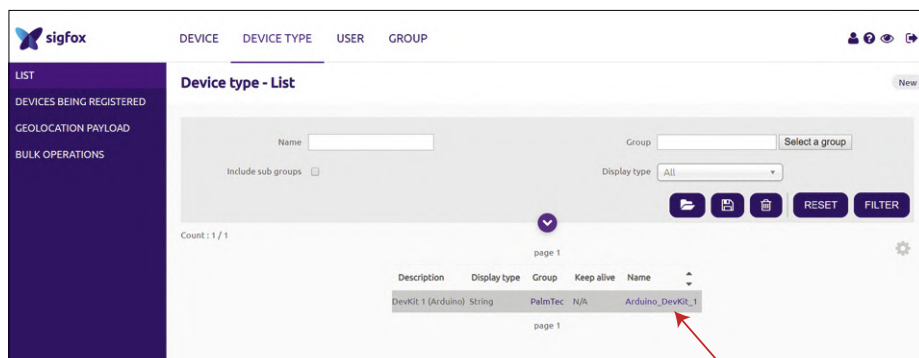


Figure 10. Choix du type d'appareil souhaité.

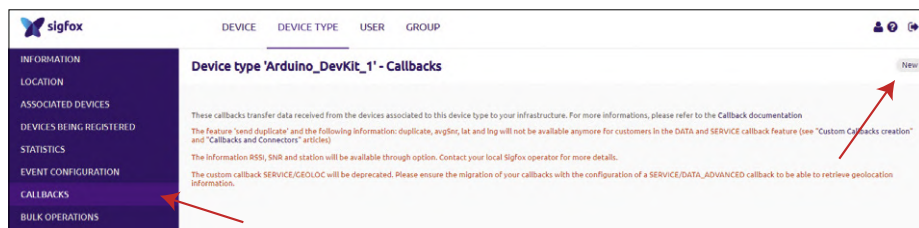


Figure 11. La fenêtre de rappel.

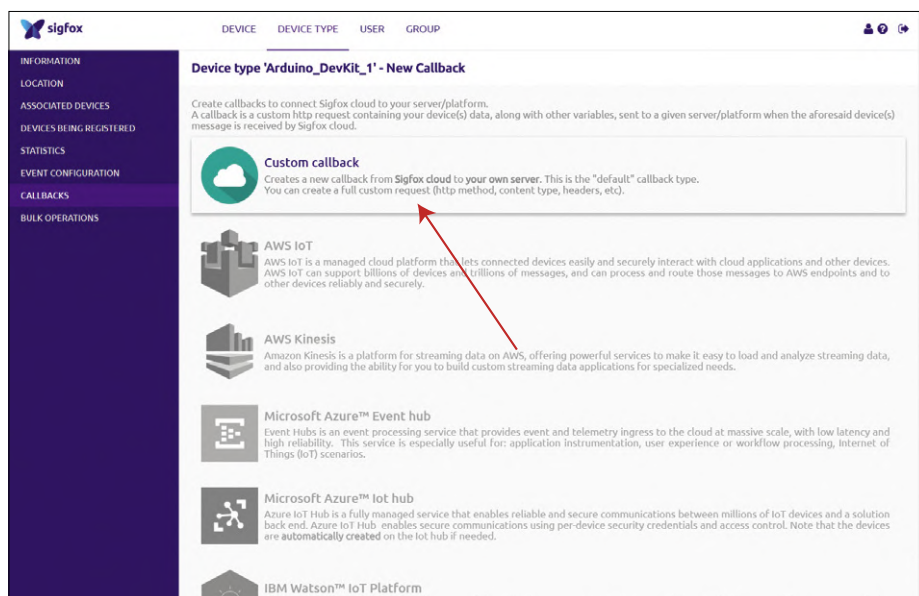


Figure 12. Il existe un vaste choix de types de rappel.

précédente. En d'autres termes, l'ordinateur central établira une connexion HTTP sur Internet pour accéder au jeton, puis utilisera le code du jeton d'accès pour s'authentifier. Une fois authentifié, il peut écrire des données dans le bac. Pour configurer le rappel, il faut d'abord se connecter à l'infrastructure Sigfox [2]. Sur la page principale, cliquez sur l'onglet *Device Type* (type d'appareil) et dans la liste qui apparaît alors (elle ne comportera qu'une seule entrée dans ce cas), cliquez sur le nom du type d'appareil *Arduino_DevKit_1* (fig. 10). La page d'aperçu de cet appareil apparaîtra alors. Dans la liste de gauche, cliquez sur *Callbacks* (fig. 11). Si vous cliquez maintenant sur le petit bouton *New* en haut à droite de

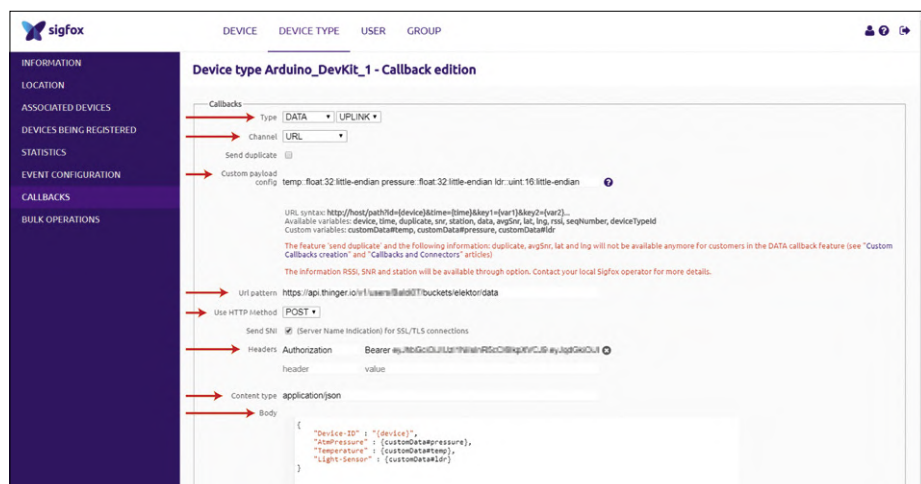


Figure 13. Configuration du rappel.

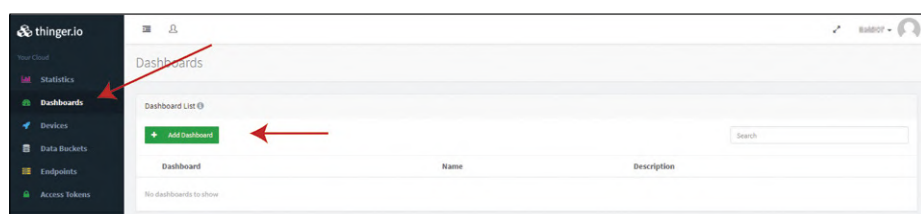


Figure 14. Début de la création d'un tableau de bord.

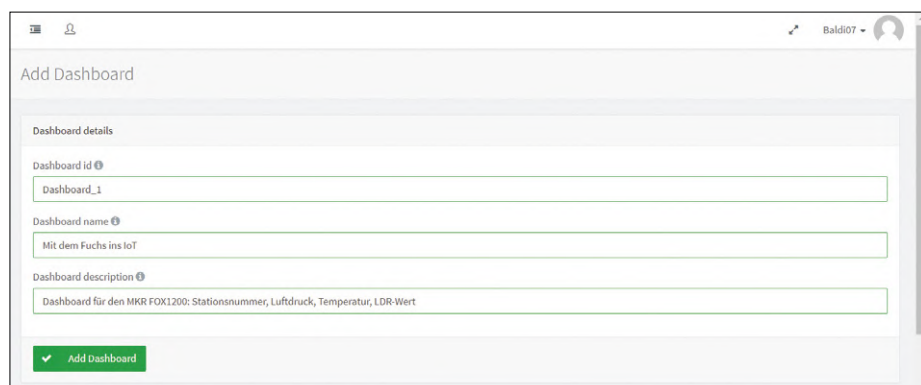


Figure 15. Paramètres de base du tableau de bord.

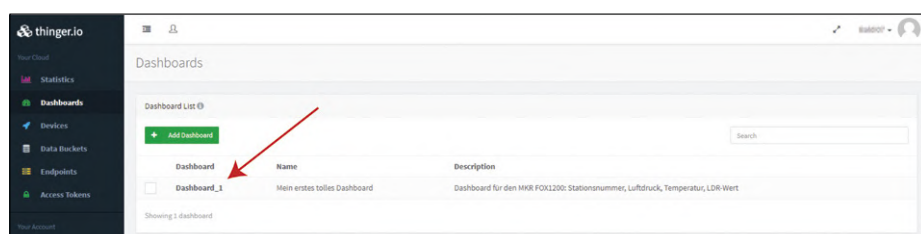


Figure 16. Liste des tableaux de bord disponibles.

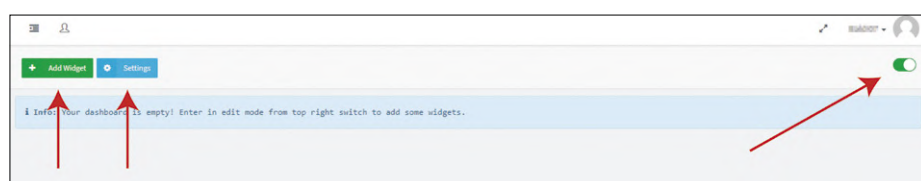


Figure 17. Le tableau de bord en mode édition.

cette fenêtre, une liste apparaîtra avec tous les types de rappel disponibles qui fonctionnent avec les programmes de tableau de bord et de cloud courants (**fig. 12**). Parmi ceux-ci, choisissez *Custom callback* (rappel personnalisé). Dans la fenêtre qui apparaît maintenant, vous pourrez configurer le rappel et la manière dont les données sont envoyées de l'infrastructure Sigfox à *thinger.io* (**fig. 13**).

Veillez à ce que tout ce que vous saisissez sur cette page apparaisse **exactement** comme décrit ci-dessous : la moindre co(q)uille entraînera des messages d'erreur et le système entier cafouillera.

Une description détaillée du fonctionnement des différents cadres dépasse le cadre de cet article – voir [3], [4] et [5]. Commençons par le haut de la figure 13. Sous *Type*, les paramètres corrects sont DATA et UPLINK, et pour *Channel*, choisissez URL, car c'est par une connexion HTTP que nous allons parler à *thinger.io*.

Maintenant, les choses se compliquent un peu. Les valeurs de données envoyées par le MKR FOX 1200 dans la charge utile vers l'infrastructure Sigfox doivent être décomposées en valeurs individuelles au sein de l'infrastructure Sigfox. En interne, un nom de variable librement choisi est attribué à chaque valeur. Ces noms sont utilisés pour spécifier le traitement ultérieur que l'infrastructure effectue sur les valeurs des variables. La manière dont la charge utile Sigfox est déballée est spécifiée dans le champ *Custom payload* (charge utile personnalisée) et doit refléter la manière dont les valeurs des données sont incorporées dans la charge utile sur le MKR FOX 1200. Pour que la charge utile Sigfox soit décomposée correctement en variables, il faut en spécifier précisément la séquence des données, les types de données concernés et le format de stockage.

Dans notre exemple, la *Custom payload* devrait contenir trois éléments, chacun apparaissant **précisément** comme suit.

```
temp::float:32::little-endian
pressure::float:32::little-endian
ldr::uint:16::little-endian
```

La première donnée de la charge utile envoyée par le MKR FOX 1200 est donc la température. Dans l'infrastructure

Sigfox, cette valeur porte le nom de «temp» et le type de données est «float», sur 4 octets (32 bits).

La deuxième valeur des données de la charge utile est la pression atmosphérique, également une variable flottante codée sur 32 bits.

La troisième et dernière valeur est la tension de la LDR. Dans l'infrastructure, elle porte le nom de «ldr» et est une variable de type «uint» stockée sur 2 octets (16 bits).

Le terme *little-endian* précise l'ordre des octets utilisés pour stocker une valeur donnée : *little-endian* est l'ordre utilisé par notre µC Atmel [6]. Vous devez entrer les trois éléments ci-dessus, concaténés sur une seule ligne et séparés par des espaces, dans le champ de configuration de la charge utile personnalisée. Vous pouvez maintenant configurer l'infrastructure Sigfox pour traiter ces 3 variables, ou bien vous pouvez les envoyer directement à *thinger.io* sous cette forme.

Sous *URL pattern*, vous spécifiez l'URL de destination sur *thinger.io*. L'URL est construite de la manière suivante.

```
https://api.thinger.io/v1/users/{user_id}/buckets/  
{bucket_id}/data
```

Dans l'adresse, remplacez *user_id* par le nom d'utilisateur de votre compte *thinger.io* (fig. 2) et *bucket_id* par l'identifiant du bac, ici «elektor» (fig. 5), dans les deux cas sans parenthèses ni guillemets.

Comme méthode HTTP, choisissez *POST*.

L'étape suivante consiste à mettre en place une authentification pour le transfert de données vers *thinger.io*. Cela se fait dans les deux champs situés à côté des *headers* (en-têtes). Dans le champ *header* entrez le mot *Authorization* et, à côté, dans le champ *value*, entrez *Bearer* suivi d'un espace et du jeton d'accès que nous avons créé (fig. 9).

```
Bearer eyJhbGciOiJI1NiIsIn...
```

Dans le champ *Content type*, entrez «application/json», ce qui indique que le corps du message est au format JSON.

Dans la dernière étape, nous précisons exactement ce qui sera envoyé à *thinger.io*. La saisie (au format JSON) dans le champ *Body* doit être la suivante. Là encore, il est essentiel que le texte apparaisse **exactement** comme indiqué.

```
{  
  "Device-ID" : "{device}",  
  "AtmPressure" : {customData#pressure},  
  "Temperature" : {customData#temp},  
  "Light-Sensor" : {customData#ldr}  
}
```

Le *Body* (corps) est entouré de parenthèses («{», «}»). Mais que signifie le texte dans le corps ?

Chaque ligne entre les accolades commence par des guillemets ; ensuite apparaît un nom de variable ad libitum (mais unique), suivi de nouveaux guillemets. Suivent un espace, un double-point et un autre espace. C'est là que la valeur de la variable sera insérée par l'infrastructure Sigfox. Le nom de la variable et sa valeur seront ensuite envoyés sous ce format à *thinger.io*, où ils seront stockés dans le bac de données approprié.

Nous avons donc

- "Device-ID" : "{device}" qui donnera le numéro d'appareil unique de la station Sigfox (notre MKR FOX 1200), qui est connu de l'infrastructure. Cette information est envoyée avec les données mesurées afin que la station

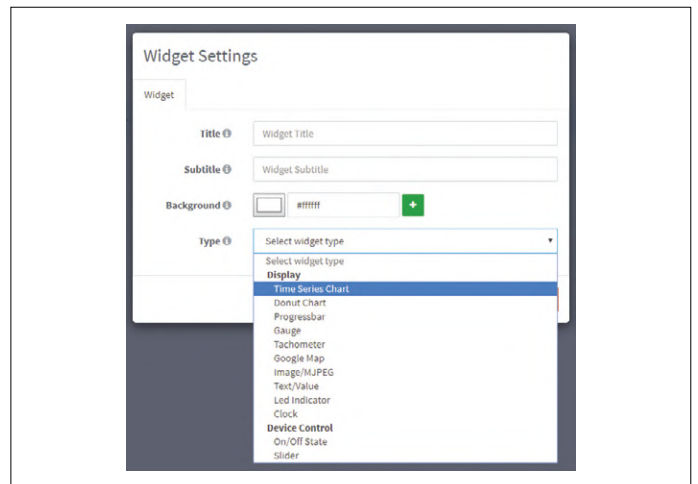


Figure 18. Première fenêtre pour la création d'un widget.

d'origine puisse être identifiée avec certitude par la suite.

- La ligne «AtmPressure» : {customData#pressure} envoie une variable avec le nom *AtmPressure*. La valeur de cette variable correspond à la valeur de la variable d'infrastructure appelée *pression*, que nous avons précédemment spécifiée dans le champ *Custom payload config*. Le mot-clé *customData#* indique que la valeur d'une variable déballée à l'aide du champ *Custom payload config* doit être insérée. C'est ainsi que les noms et les valeurs des variables sont reformatés pour être transmis à *thinger.io*.
- Les deux dernières lignes sont traitées de manière analogue. Chaque ligne, à l'exception de la dernière, doit être terminée par une virgule.

Cela complète la configuration du rappel dans l'infrastructure de Sigfox pour lui permettre d'envoyer les données souhaitées au bac de données «elektor» de *thinger.io*. Lorsque vous avez correctement effectué toutes les étapes ci-dessus, vous pouvez cliquer sur OK.

Désormais, lorsque le MKR FOX 1200 envoie une charge utile à l'infrastructure Sigfox, et en supposant qu'aucune erreur n'est rencontrée à la réception, l'infrastructure de rappel Sigfox transmet immédiatement les données au format spécifié ci-dessus à *thinger.io* pour traitement ultérieur.

Tableau de bord chez *thinger.io*

Nous voulons présenter les données sous forme graphique sur un tableau de bord. Pour le créer, allez dans la console principale de *thinger.io* (fig. 3) et cliquez à gauche sur *Dashboards* et dans l'écran suivant sur *Add Dashboard* (fig. 14).

Remplissez les champs de la fenêtre *Dashboard details* (fig. 15). *Dashboard id* est un identifiant unique pour l'ensemble du compte. Les champs *Dashboard name* et *Dashboard description* peuvent contenir n'importe quelle information, pourvu qu'elle soit lisible par l'utilisateur. Confirmez les entrées en cliquant sur *Add Dashboard*. Le nouveau tableau de bord sera ajouté à la liste des tableaux de bord que vous verrez chaque fois que vous cliquerez sur l'élément de menu *Dashboards* dans la fenêtre principale de la console (fig. 16).

Si vous cliquez sur le nom du tableau de bord, une fenêtre apparaîtra avec un tableau de bord vide (fig. 17). En haut à

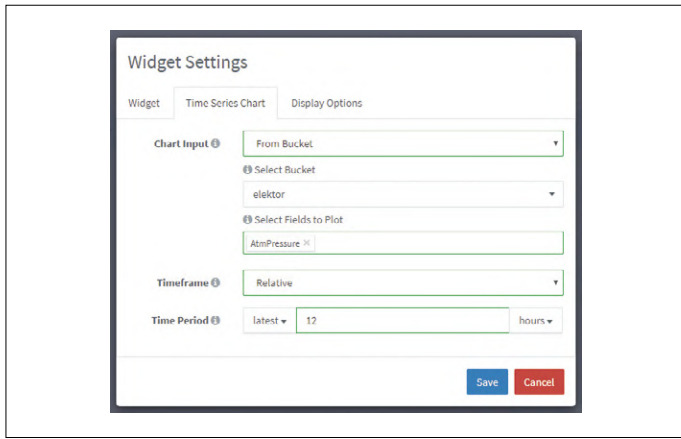


Figure 19. Configuration d'un graphique de séries chronologiques.

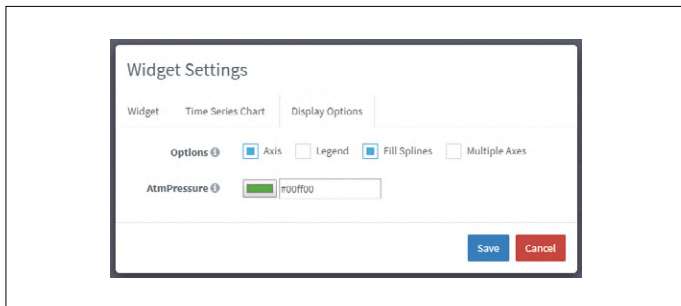


Figure 20. Réglage des caractéristiques de base de l'affichage.

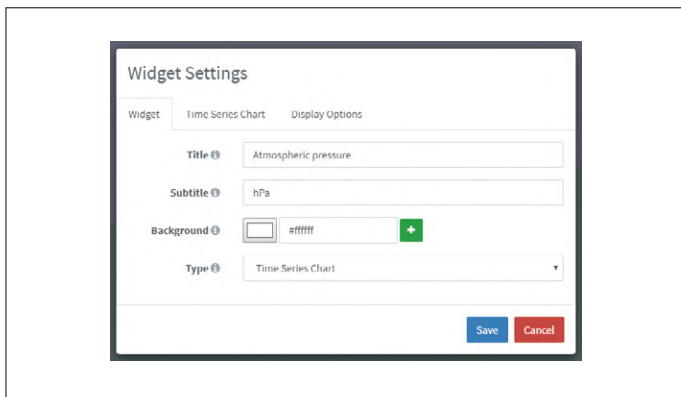


Figure 21. Paramètres dans l'onglet Widget.

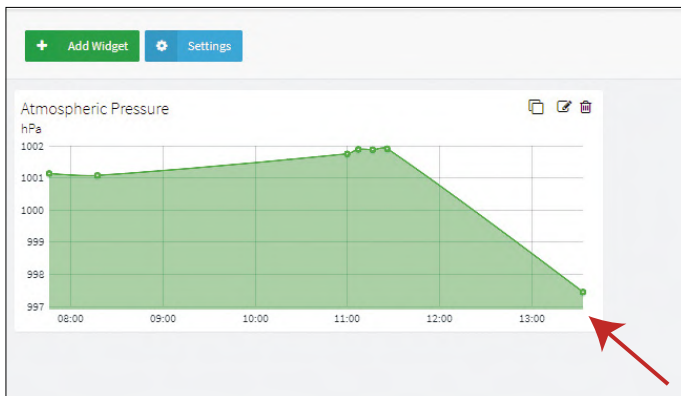


Figure 22. Le graphique des séries chronologiques ajouté au tableau de bord : quelques relevés sont déjà tracés.

droite, il y a un petit interrupteur : faites-le glisser vers le vert pour passer en mode édition. Il y a également deux boutons *Add Widget* et *Settings*. Dans *thinger.io*, les éléments d'affichage du tableau de bord qui font partie de l'interface utilisateur sont appelés *widgets*. Un large éventail de *widgets* est disponible, notamment des graphiques de séries chronologiques, des graphiques en beignet, des barres de progression, des affichages analogiques, des cartes Google avec des informations sur la position, des images, des affichages de texte, des DEL et une horloge. Dans notre application, nous souhaitons afficher un graphique chronologique pour chacune des trois valeurs mesurées et un affichage numérique pour chacune d'entre elles indiquant la valeur la plus récemment reçue.

Cliquez sur *Add Widget* et une fenêtre intitulée *Widget Settings* apparaîtra (fig. 18). Sous *Type*, vous pouvez choisir le type de *widget* souhaité. Dans ce cas, nous choisissons *Time Series Chart*. Deux autres onglets apparaîtront (fig. 19). Dans l'onglet *Time Series Chart*, sous *Chart Input*, choisissez l'option *From Bucket*. Au fur et à mesure que vous avancerez, plus de champs à remplir apparaîtront. Dans le champ *Select Bucket*, choisissez le bac «elektor». Dans le champ suivant, *Select Fields to Plot*, choisissez la valeur de la donnée à afficher.

Nous commençons par *AtmPressure*. Dans la rubrique *Timeframe* (cycle), choisissez *Relative* et réglez la fenêtre de temps à afficher sur 12 h. Le graphique affichera alors les données des 12 h les plus récentes ; ajustez ce cycle comme vous le souhaitez. Cliquez maintenant sur l'onglet *Display Options* (fig. 20) et désactivez l'option *Multiple Axes* : nous allons établir un graphique distinct pour chacune de nos variables, et nous n'avons donc pas besoin de plusieurs axes Y sur le même graphique. L'option *Axis* (déjà activée) permet de mettre automatiquement à l'échelle les axes X et Y du graphique. Dans le champ voisin de l'étiquette *AtmPressure*, nous pouvons choisir la couleur du graphique : un vert apaisant. Si vous activez l'option *Fill Splines*, la région entre le graphique et l'axe des X aura la couleur sélectionnée à côté de l'étiquette *Pression atmosphérique*. L'option *Legend* permet d'étiqueter séparément des séries chronologiques différentes tracées sur le même graphique.

Enfin, nous revenons à l'onglet *Widget* (fig. 21), où nous pouvons maintenant définir le titre et le sous-titre du graphique. Le titre est *Atmospheric pressure* et le sous-titre est l'unité de mesure : hPa. Le paramètre *Background* permet d'ajuster la couleur de fond du graphique. Nous terminons le paramétrage du graphique en sauvegardant : le graphique sera ajouté au tableau de bord (fig. 22). Vous pouvez agrandir ou réduire le graphique en cliquant dans le coin inférieur droit. Vous pouvez également faire glisser le graphique dans la zone du tableau de bord. Le changement de taille graphique ne se passe pas toujours comme on veut.

Nous voulons également afficher la lecture actuelle de la pression atmosphérique sous forme numérique (fig. 23). Le bouton *Add Widget* ouvre à nouveau le menu permettant d'ajouter un nouvel élément à l'affichage ; dans ce cas, dans le champ *Type*, nous sélectionnons *Text/Value*.

Dans le champ *Widget Value*, nous sélectionnons à nouveau *From Bucket*, choisissons le bac «elektor» et précisons à nouveau la valeur que nous voulons afficher (ici «AtmPressure»). Pour la *Bucket Value*, nous définissons *Latest Value*. L'onglet *Display Options* nous permet d'effectuer divers réglages de configuration. Par exemple, nous pouvons définir la précision

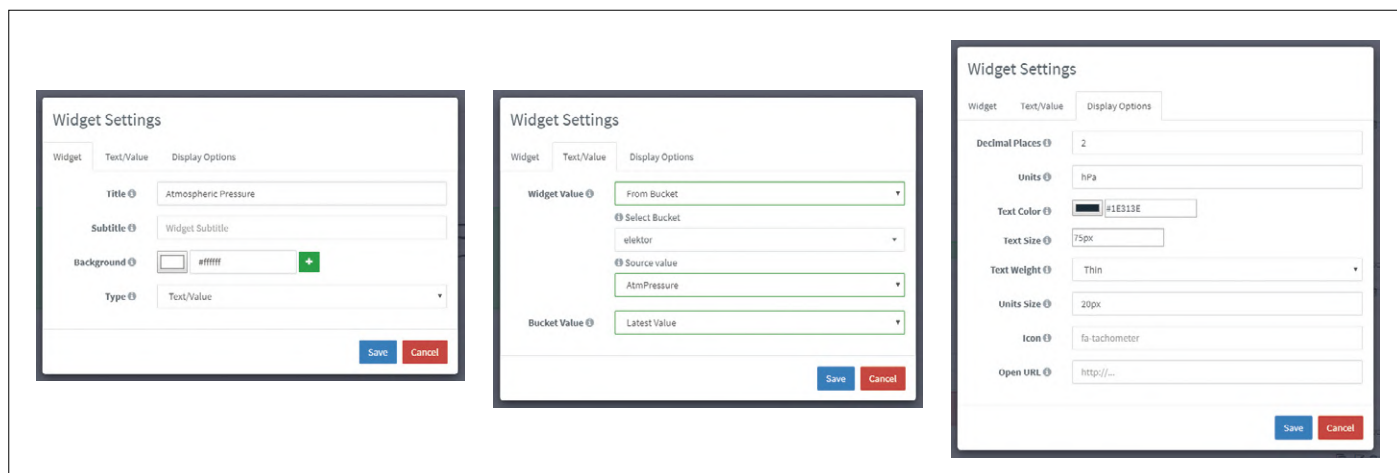


Figure 23. Configuration de l'affichage numérique pour la lecture de la pression atmosphérique.

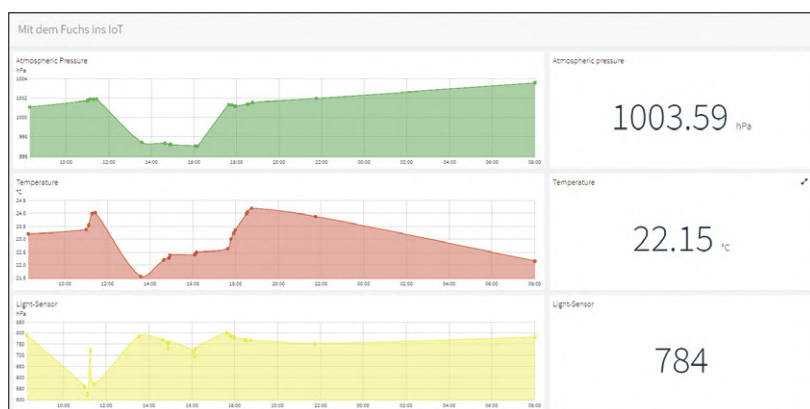


Figure 24. Le tableau de bord achevé.

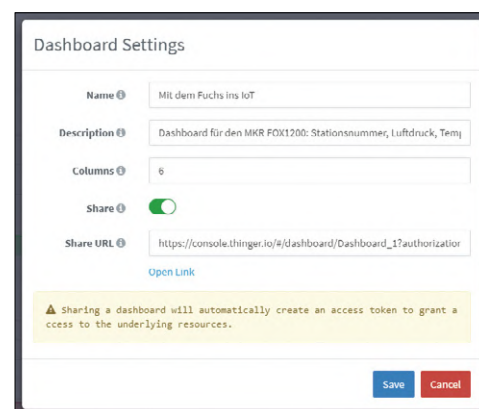


Figure 25. Paramètres généraux du tableau de bord.

avec laquelle la valeur est affichée (nombre de décimales) et nous pouvons définir 'hPa' pour le champ *Unités*. Cette chaîne sera affichée dans le *widget* à côté de la valeur mesurée. Il existe encore d'autres options. Dans l'onglet *Widget*, nous avons à nouveau défini *Atmospheric pressure* pour le champ *Title*. Nous pouvons laisser vide le sous-titre.

L'affichage des deux autres relevés est configuré de manière

similaire (**fig. 24**). Pour être sûr que tout est bien enregistré dans *thinger.io*, quittez le mode édition et retournez-y... D'autres d'informations sur les tableaux de bord sur *thinger.io* se trouvent ici [7].

Il ne nous reste plus qu'à permettre l'accès au tableau de bord «depuis l'extérieur». De retour dans le menu du tableau de bord, cliquez sur le bouton bleu intitulé *Settings*, qui vous amènera

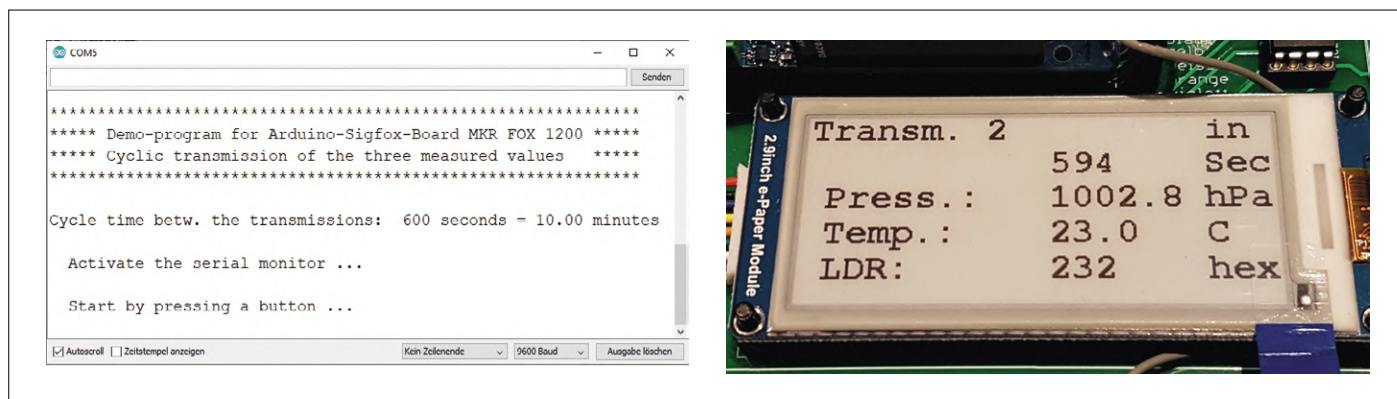


Figure 26. Sortie de Sigfox-Demo.ino sur le moniteur série et sur l'afficheur ePaper.



Figure 27. Le tableau de bord présente bien sur un téléphone !

à la page des paramètres du tableau de bord (**fig. 25**). Vous pouvez y donner un nom et une description à votre tableau de bord (ce n'est pas nécessaire dans ce cas, nous l'avons déjà configuré). Le bouton *Share* permet d'activer le tableau de bord pour un accès externe : le lien (assez long) nécessaire pour y accéder apparaîtra dans le champ «Partager l'URL». Grâce à ce lien, il est désormais possible d'accéder au tableau de bord depuis n'importe quel endroit dans le monde et d'examiner les données qu'il affiche. En raison de la longueur de l'URL, il est recommandé d'utiliser un raccourci d'URL (bitly.com) pour l'abrégé. L'URL raccourcie peut ensuite être convertie en un code graphique QR qui peut être scanné par un téléphone : cela évite la saisie de l'URL longue.

Lorsque sur *thinger.io* vous autorisez l'accès public au tableau de bord, celui-ci est automatiquement protégé en écriture. Les tiers ne peuvent pas modifier les paramètres ou effectuer d'autres ajustements : cela n'est possible qu'après avoir utilisé les informations d'identification appropriées pour se connecter au portail *thinger.io*.

Nous souhaitons tester le système dans son ensemble, mais pas avec le sketch *Sigfox-3_0.ino*, car il faut appuyer sur un bouton chaque fois que nous voulons envoyer un message. Il existe un autre sketch, appelé *Sigfox-Demo.ino* [8]. Toutes les dix minutes, ce programme acquiert automatiquement les trois relevés et les transmet (**fig. 26**). L'intervalle entre les transmissions est déterminé par la variable globale *w_zeit*. Réglez-la comme vous le souhaitez pour acquérir et afficher les lectures sur une longue période en n'oubliant pas que vous n'êtes autorisé à envoyer que 140 messages par jour sur le

réseau Sigfox, soit environ un message toutes les 10 min. Il n'est pas si critique qu'un compte gratuit chez *thinger.io* ne vous permette d'écrire des données dans un bac plus souvent qu'une fois par minute. Que dites-vous de la **fig. 27** avec le tableau de bord sur l'écran d'un téléphone ?

En conclusion

Voici donc la fin de cette série d'articles de présentation du réseau Sigfox 0G. La programmation très flexible et polyvalente de l'Arduino MKR FOX 1200 facilite permet l'acquisition de types de données très variées, puis leur transmission sur le réseau Sigfox dans le monde entier. Parallèlement, grâce aux possibilités d'affichage simples, mais puissantes offertes par *thinger.io*, la programmation de tableaux de bord personnalisés pour la communication d'informations aux utilisateurs finaux est rapide et efficace.

Bonne collecte de données dans le monde passionnant de l'Internet des objets ! ◀

190281-D-03

@ WWW.ELEKTOR.FR

→ Arduino MKR FOX 1200
www.elektor.fr/19096

→ Arduino Antenne 868 MHz
www.elektor.fr/19095

Liens

- [1] Thinger: <http://www.thinger.io/>
- [2] Infrastructure Sigfox : <https://backend.sigfox.com/>
- [3] Infos sur la configuration de Callback: <https://support.sigfox.com/docs/callbacks-documentation>
- [4] Infos sur la configuration de Callback: <https://support.sigfox.com/docs/custom-callback-creation>
- [5] Infos sur la configuration de Callback: <https://backend.sigfox.com/apidocs/callback>
- [6] Ordre des octets dans le codage des nombres : <https://de.wikipedia.org/wiki/Byte-Reihenfolge>
- [7] Infos sur la création du tableau de bord : <https://github.com/thinger-io/Docs/blob/gh-pages/console/dashboards.md>
- [8] La page de ce projet : <https://www.elektor.fr/190281-D-03>

TMS0280 : synthèse vocale

drôle(s) de composant(s)

Neil Gruending (Canada)

Vers 1976, *Texas Instruments* (TI) a fait plonger ses ingénieurs sur un appareil qui permette, comme outil d'apprentissage de l'orthographe, de prononcer un mot épilé sur un clavier. Il devait être portable, bon marché et connaître autant de mots que possible, contenus dans seulement 256 Kbits de mémoire. Un sacré défi à cette époque. En 1978, grâce au CI de synthèse vocale TMS0280, TI avait réussi à intégrer les performances d'un ordinateur de plusieurs milliers de dollars dans ce « jouet » à 50 \$ connu sous le nom de *Speak & Spell*.

TI disposait déjà du μ C TMS1000, pas assez puissant pour faire de la synthèse vocale à lui tout seul. Pour cela, TI s'est tourné vers Larry Brantingham, Paul Breedlove, Richard Wiggins et Gene Frantz qui, après des recherches approfondies, ont compris que les consommateurs associaient spontanément une voix monocorde à la parole synthétisée par ordinateur. D'où ils déduisirent qu'ils pouvaient réduire considérablement les exigences de qualité auxquelles devait répondre la voix synthétique. Ainsi naquit le TMS0280, avec son codage prédictif linéaire pour synthétiser la parole à un taux d'échantillonnage de 10 kHz à partir de données audio codées à 1200 bits/s.



Figure 2. TI Speak & Spell. Auteur : Bill Bertram 2006. Aucune modification. CC-BY-2.5 -

Attribution : <https://creativecommons.org/licenses/by-sa/2.5/deed.en>

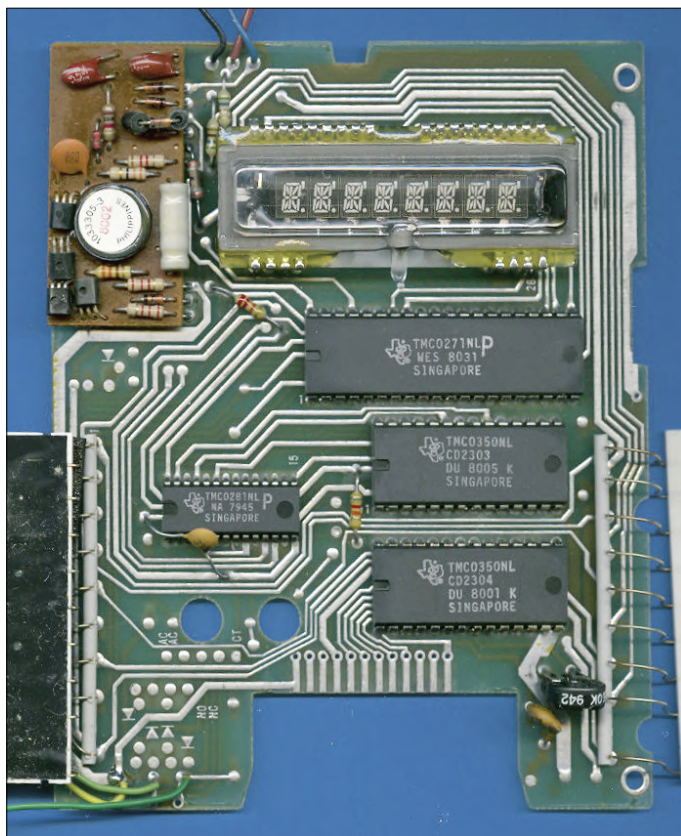


Figure 1. TMC0280 étiqueté TMC0281 dans un Speak & Spell.

Source : Datamath.

Les données codées étaient stockées sous forme de trames de données contenant des informations nécessaires aux algorithmes de la parole relatives à l'énergie, à la hauteur tonale et aux coefficients de réflexion. Parfois, les trames de données étaient produites à la main en labo, mais généralement, les paramètres de codage étaient le produit de grands programmes informatiques qui analysaient de vraies données vocales. Quoi qu'il en soit, l'objectif était de créer des données codées qui puissent être clairement comprises et qui devaient sonner autant que possible comme la parole humaine.

Le TMS0280 contenait un processeur de signal numérique avec un multiplicateur, un filtre en treillis à deux multiplications et une ROM masquée pour la mémoire du programme. La ROM permettait de modifier le débit de données, l'excitation (résonance) et les caractéristiques du filtre en fonction des variations de l'émission vocale. Le processeur vocal intégré contenait également un oscillateur, une logique d'interface et, pour la sortie audio, un CN/A de 200 mW.

Le TI n'a plus produit de *Speak & Spell* depuis 30 ans, ce qui n'empêche pas une vaste communauté d'enthousiastes de l'utiliser encore aujourd'hui. Le plus célèbre est sans doute E.T., l'extra-terrestre du film éponyme, qui utilisa un *Speak & Spell* pour appeler chez lui.

Si vous avez encore un *Speak & Spell* dans vos placards, il serait peut-être temps de le ressortir et voir ce qu'il y aurait à tirer de son TMC0280. ◀

190383-C-02



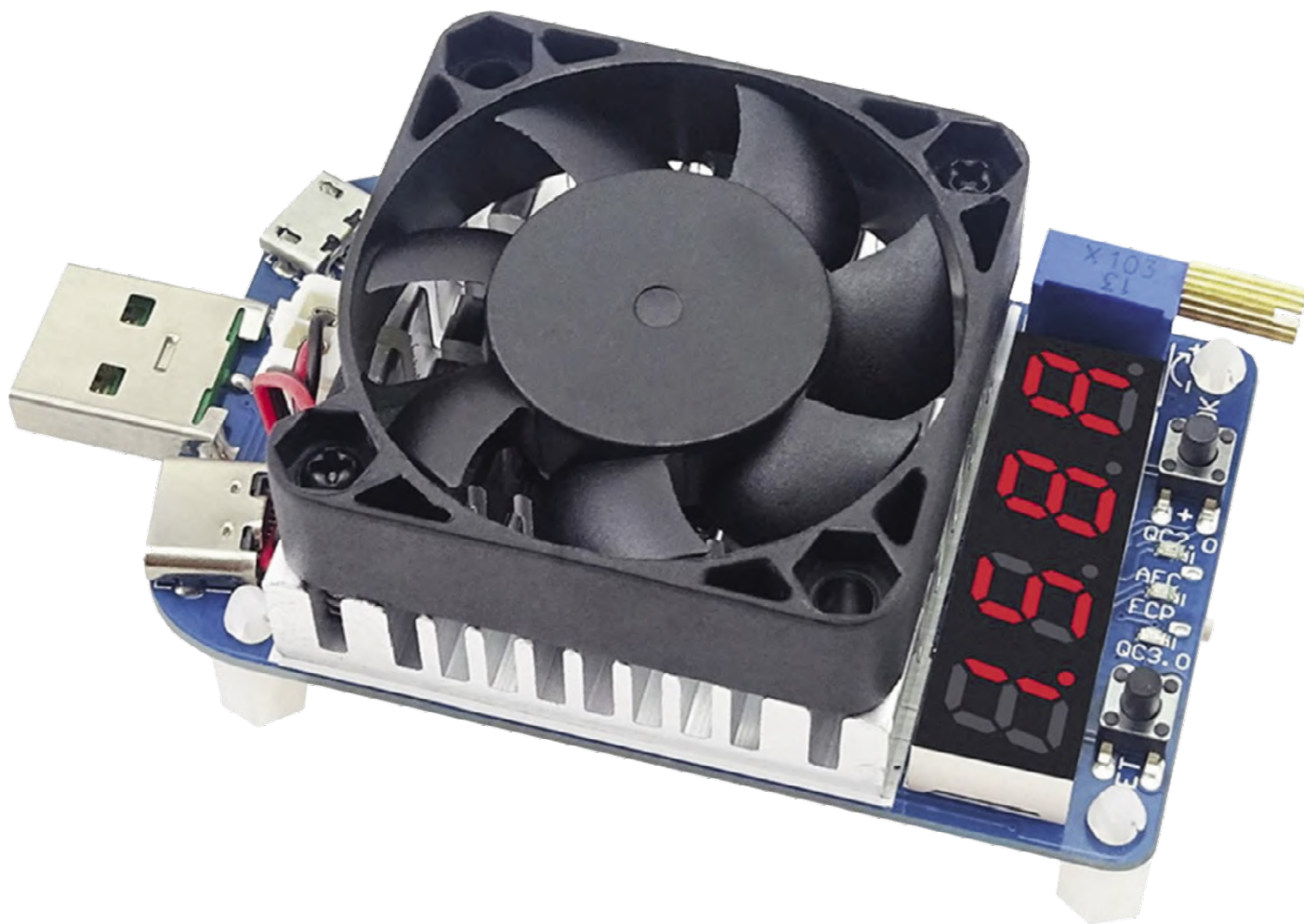
BANC D'ESSAI

charge électronique USB Joy-IT HD35

Pour tester la capacité de charge des ports USB

Thomas Scherer (Allemagne)

Cette charge électronique à brancher aux ports USB est une spécialité chinoise. Elle permet de vérifier les caractéristiques de ce super chargeur que vous venez peut-être de commander, en Chine aussi. Avec ses trois ports USB et son courant de sortie de 4 A selon la pub, tient-il vraiment ses promesses? Un tel accessoire devrait-il figurer dans l'inventaire de tout atelier d'électronique pour tester par exemple les ports USB de vos propres projets?



Cette charge électronique à brancher aux ports USB est une véritable spécialité. Elle permet de vérifier les caractéristiques de ce super chargeur que vous venez de recevoir de Chine avec ses trois ports USB et son courant de sortie de 4 A selon la pub tient vraiment ses promesses. Un tel accessoire devrait

figurer dans l'inventaire de tout atelier d'électronique pour tester p. ex. les ports USB de vos propres projets.

À première vue, j'ai surtout été amusé par ce HD35 proposé par Joy-IT. Une charge électronique pour port USB ? N'ont-ils pas



Figure 1. Cette petite boîte en plastique contient l'électronique du testeur de charge USB HD35.



Figure 2. La boîte en plastique vue de dessous. C'est là qu'on trouve les caractéristiques techniques et l'URL du fabricant.

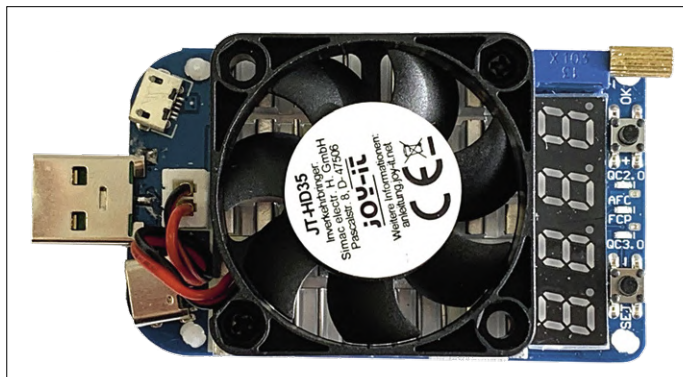


Figure 3. Mon testeur de charge USB dans toute sa gloire. Tout y est : 3 entrées USB, ventilateur, écran, 3 LED, 3 boutons (le troisième est à l'arrière) et potentiomètre.

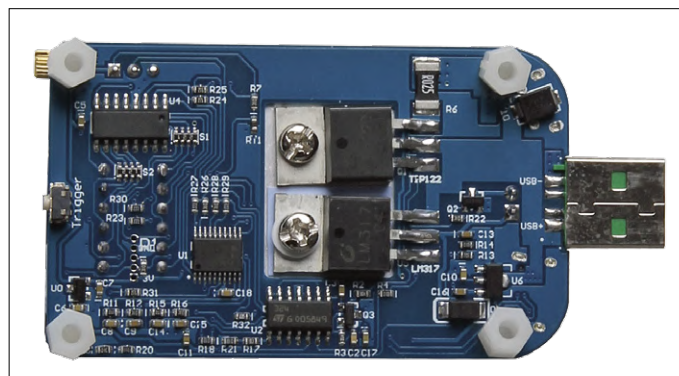


Figure 4. Dessous du testeur de charge USB avec l'électronique.

des drôles d'idées en Extrême-Orient ? Ce n'est qu'en regardant de près ses caractéristiques que son utilité m'est apparue clairement. Tester la puissance et la stabilité de toutes sortes de ports USB n'est pas seulement utile pour les appareils que nous concevons, mais aussi pour tous ces chargeurs achetés. J'avais déjà parlé d'un testeur USB [1] avec l'UM25C, mais ici c'est tout à fait autre chose.

Caractéristiques chargées

Voici ce que Joy-IT dit de son produit : «Le JT-HD35 est un ballast (une résistance de charge) électronique compacte pour port USB d'une puissance maximale de 35 W». Ça ne vous paraît pas un peu fort pour l'USB ? Ils précisent que la charge connaît plusieurs modes selon les méthodes de charge rapide, et que l'intensité du courant peut atteindre 5 A et qu'elle est réglable pour l'«adapter à presque toutes les applications» et qu'elle est équipée d'un dissipateur thermique ventilé «pour un refroidissement efficace de l'ensemble de l'appareil». En résumé les données techniques sont :

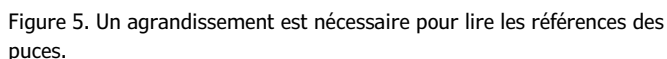
- Tension de fonctionnement : 4 à 25 V (CC)
- Courant de charge : 0,25 à 5 A

- Puissance : 35 W max. (d'où le nom de l'appareil)
- Indicateurs d'état : 3 LED (vert, bleu, rouge)
- Affichage : quatre chiffres à sept segments LED rouge
- Fonctionnement : 3 poussoirs, 1 potentiomètre multitour (courant)
- Connexions : USB 2.0, micro-USB, USB-C
- Ventilateur : 40 mm, 0 à 8 000 tr/min, thermostaté
- Dimensions : 84 x 41 x 28 mm
- Poids : 52 g

Déballage

Dans le colis, je n'ai trouvé qu'une petite boîte en plastique discrète (**fig. 1**) avec la photo du module qui doit se trouver à l'intérieur. L'électronique, c'est bien, mais pas sans manuel imprimé. Pas la moindre trace d'un bout de papier. Pas non plus de code QR ni de lien, ni dessus ni dessous (**fig. 2**). J'ai donc fait comme on fait maintenant quand on cherche quelque chose : on suit l'URL du fabricant, ici Joy-IT et on se met à fureter sur leur site jusqu'à ce qu'on trouve la page du produit, et ... le manuel [2] au format PDF.

En fait, j'admets volontiers que les manuels imprimés ne me manquent pas vraiment. Avec tous les appareils et les outils



Regardez le contenu déballé de la petite boîte en plastique (**fig. 3**) : le ventilateur n'est vissé au radiateur que par deux des quatre vis possibles. Ça doit faire une économie d'un centime entier, mais admettons que mécaniquement ça tient même si ça fait mal aux yeux. Sous la carte (**fig. 4**) vous pouvez voir le TIP122, un Darlington NPN de puissance de 65 W (probablement la charge proprement dite) et le régulateur de tension variable LM317. Je vois aussi le microcontrôleur U1, un N76E003AT20, cousin du 8051, extrêmement bon marché à 28 ¢ avec des puces périphériques et des composants passifs. Pour mieux lire les inscriptions, je les photographie (**fig. 5**). Tout est clair, on peut y aller...

Après la mise sous tension, la tension est affichée sur deux chiffres, soit «5,1U». Avec chaque pression sur le poussoir, on passe de l’affichage de la tension à celui de la puissance (d’abord «0.0P») puis à celui du courant (ici «1.00A» clignotant). Appuyez une fois sur le bouton OK, le clignotement s’arrête, le courant reste affiché. Le ventilateur, qui a démarré après 20 s, m’a fait comprendre pourquoi : le clignotement signifie «affichage du courant de consigne de la charge, mais charge absente». En revanche, quand l’affichage ne clignote pas, le courant de consigne circule. Il suffit de réappuyer sur «Ok» pour faire clignoter à nouveau l’indicateur de puissance. Quand j’augmente avec le potentiomètre l’intensité du courant, la tension de la petite alimentation noire s’effondre immédiatement : il ne reste que 3,5 V à 1,1 A. À ma grande surprise, ce petit bloc est donc doté d’une limitation de courant en cas de surcharge ! Notons au passage que si le connecteur micro-USB est peu encombrant, il est aussi totalement inadéquat comme connecteur d’alimentation ; même à 1 A, l’afficheur n’indique que 4,9 V. Dans les mêmes conditions mais avec un câble USB-A, cette alimentation tient bravement ses 5,0 V. Passons maintenant au gros cobaye : connecté par le câble USB-A à sa prise USB-A, sa limitation de courant ne se manifeste que vers 3 A. Connecté par le câble USB-C, j’ai même pu atteindre un courant de 3,88 A avant que la limitation de courant coupe l’alimentation (**fig. 7**). La vaillante *Ravpower* tient ses promesses.

Je ne suis pas arrivé plus loin que ça, ne trouvant pas tout seul comment tester les autres tensions selon OC 2.0 ou 3.0.

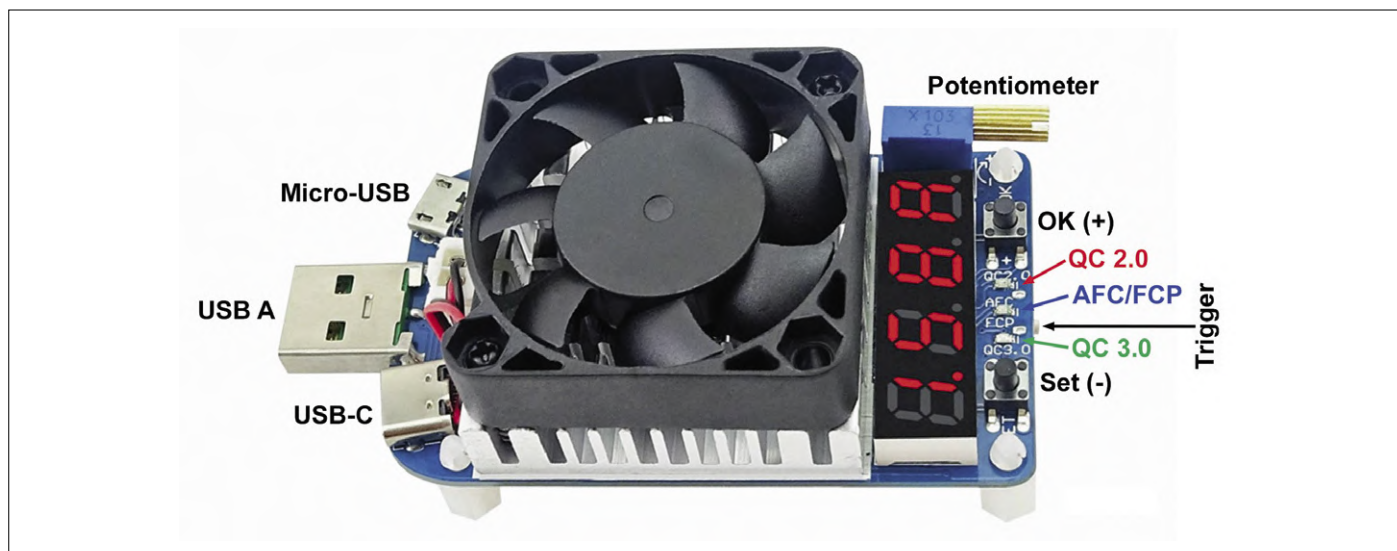


Figure 8. Connexions, affichages et commandes du HD35 (illustration de Joy-IT modifiée).

J'ai donc téléchargé le manuel pour l'étudier. Ah, je n'avais pas remarqué le petit bouton discret, en dessous, avant de le voir dans la notice (**fig 8**). Voyons ce qu'on peut faire de ce discret bouton en bas, au milieu, à droite. Une pression de 2 s permet de passer du mode par défaut (5 V «normal») aux autres modes. Vous devriez pouvoir lire «-5.0» et vous êtes en mode QC 2.0. Si la LED rouge clignote, vous pouvez régler la tension par paliers de 5/9/12/15/20 V avec les touches «+» et «-». Avec *Trigger*, vous pouvez également accéder aux autres modes. À défaut de chargeur adéquat, je n'ai pas pu tester les fonctions de charge rapide du mode AFC9 de Samsung ni du FCP9 de Huawei.

Sans le manuel, je n'aurais pas compris ce changement de mode. L'utilisation de cette charge électronique USB ne m'a donc pas paru assez intuitive pour mon tempérament impatient. Je me connais, quand je voudrai de nouveau l'utiliser dans quelque temps, j'aurai oublié comment elle marche et ça m'agacera. Cela dit, elle fait ce qu'elle promet une fois que vous savez comment la faire fonctionner.

Conclusion

Cette charge électronique USB n'est pas chère et fait exactement ce qu'elle est censée faire. Pour rester positif, je dirai que l'absence de manuel imprimé est écologique. Il ne compte que sept pages, mais la proportion de ce que j'appellerai des faiblesses de formulation et des fautes de frappe est élevée. Le recours à de mauvais logiciels de traduction est regrettable. Un appareil de cette qualité mérite mieux. Même en anglais, c'est consternant. Si l'électronique était aussi mal conçue que la langue dans laquelle elle est décrite, l'appareil exploserait

dès sa mise sous tension. Ce n'est heureusement pas le cas. L'ergonomie est améliorable aussi. Pourquoi cacher le troisième bouton ? Ça facilite peut-être les choses pour le concepteur, mais pas pour l'utilisateur. Non seulement il est difficile à trouver, mais il faut les deux mains pour l'activer.

Cette charge pour ports USB tient ses promesses, par son prix modéré, sa ventilation suffisante, ses performances qui se laissent voir, aussi en test prolongé. J'ai essayé jusqu'à 30 W. La puissance de crête de 35 W est plus que suffisante pour tester toutes sortes de sorties USB. Si vous avez l'usage d'une telle charge électronique, il est improbable que la construire vous-même revienne moins cher.

200049-03



@ WWW.ELEKTOR.FR

→ Charge USB HD35 de Joy-IT

www.elektor.fr/joy-it-jt-hd35-usb-load-resistor-35-w

Liens

[1] Banc d'essai : Testeur USB UM25C de Joy-IT :

www.elektormagazine.fr/news/banc-d-essai-testeur-usb-um25c-avec-ecran-couleur-oled-bluetooth

[2] JT-HD35 Manual English - 933,04 KB : <https://www.joy-it.net/files/files/Produkte/JT-HD35/JT-HD35-Manual.pdf>

bus CAN + Arduino pour la surveillance des cellules solaires

Détecter et localiser les panneaux défectueux dans les grands réseaux photovoltaïques

image: Shutterstock

Somnath Bera (Inde)

Pourquoi restreindre l'excellent bus CAN aux seuls capteurs embarqués dans les véhicules ? Voici comment ce bus a été mis à contribution pour lire le rendement de chaque panneau photovoltaïque individuellement ainsi que son numéro afin de localiser et d'identifier la source de problèmes. Cette idée de détournement de bus est venue à l'auteur quand il a été confronté à des clients, des installateurs et des autorités en désaccord à propos des résultats décevants d'un réseau de cellules solaires de 25 kW en exploitation commerciale.

L'installation à surveiller consiste en un réseau de cellules photovoltaïques sur un toit : 55 panneaux (24 V) délivrent 25 kW pour alimenter un grand onduleur monophasé à 220 V qui à son tour charge un essaim de batteries. Le consommateur est un conglomérat de magasins ayant besoin d'électricité le soir pour les ventes et d'autres activités. Ces commerces étaient censés bénéficier de 6 h d'éclairage, de 17h30 à 23h30. L'installation a été inaugurée en grandes pompes.

Laver et essuyer

Le réseau photovoltaïque, l'onduleur et les batteries ont fonctionné pendant... un certain temps. Puis, sur deux mois environ, la performance s'est progressivement détériorée : même avec des périodes de charge de 10 à 12 h, l'alimentation du soir ne tenait pas plus de 3 à 4 h. Ce fut le début des chicanes. L'administration locale a incriminé le travail des installateurs, la médiocre qualité des panneaux, etc. Il a suffi de jeter un coup d'œil aux panneaux sur le toit pour comprendre. Les 55 panneaux d'une puissance de 500 W chacun répartis sur trois toits sont survolés du matin au soir par de nombreux volatiles attirés par un grand marché à ciel ouvert à proximité : la plupart étaient couverts de fiente.

En attendant de déplacer les panneaux vers un endroit moins fréquenté par les oiseaux, j'ai proposé qu'on lave chaque panneau tous les deux jours. Ce nettoyage régulier a permis

de rétablir les performances initiales. Peu après, il s'est avéré qu'un nettoyage sélectif suffisait, rendant inutile le lavage systématique. Le préposé au lavage devait néanmoins d'abord vérifier visuellement tous les panneaux, puis procéder au nettoyage sélectif. Cet examen prend beaucoup de temps, car le préposé doit faire le tour de tous les panneaux. Pour cette installation de seulement 55 panneaux, ce n'est pas un problème, mais pour une autre installation similaire, prévue avec une capacité de 100 kW, un cauchemar se préparait pour le préposé au nettoyage. C'est là qu'est née l'idée d'un système de surveillance individuelle des panneaux qui permettrait à la personne chargée du nettoyage de se rendre directement aux seuls panneaux dont le rendement était affecté par des déjections.

Détournement du bus CAN

La combinaison du réseau CAN (*Control Area Network*) avec l'Arduino est une excellente solution pour mesurer les performances des différents panneaux photovoltaïques. Inventé par Bosch pour l'automobile, le bus CAN permet de collecter les signaux de centaines de capteurs connectés dans un véhicule. Parmi les paramètres mesurés sur une voiture, il y a la température int/ext., la pression des pneus, le niveau du liquide de frein, la température du moteur, la vitesse, la température de la climatisation et bien d'autres. Tous les capteurs sont connectés par deux fils seulement dans une structure de bus et reliés

à l'ordinateur central qui surveille les signaux de sortie des capteurs et, si nécessaire, répond en émettant des commandes aux capteurs concernés, là encore par le biais du bus à deux fils. Le bus CAN va de nœud en nœud avec un câble blindé à paires torsadées d'une impédance de 120 Ω. Pour une bonne adaptation et pour réduire la réflexion du signal, les deux nœuds d'extrémité du bus CAN sont équipés d'une résistance de terminaison de 120 Ω.

Comme émetteur-récepteur CAN bon marché, nous utilisons la puce MCP2515, qui vient habituellement de Chine à moins de 1,50 \$. Du côté du microcontrôleur, elle est prête pour le SPI et du côté CAN, il y a un séparateur entre les fils de bus CAN H (haut) et CAN L (bas). On utilise un câble de cuivre à paires torsadées blindées de bonne qualité et durable. Pour le µC, nous proposons l'ATMEGA328 sur un Arduino UNO, qui reçoit les signaux du capteur sur ses broches d'E/S numériques et analogiques. Pour commander les actionneurs, vous pouvez connecter des relais, des servos et d'autres dispositifs de sortie sur les autres broches d'E/S.

Par souci de simplicité, nous nous contentons du CAN 2.0 qui accepte des données de 0 à 8 octets à la fois d'un nœud à l'autre. Tous les nœuds, y compris le maître, reçoivent un identifiant (0 à 255) pour l'émission et la réception. Plus cet identificateur est bas, plus la priorité pour l'envoi et la réception par cet appareil est élevée. Habituellement, c'est le maître de bus qui reçoit l'identifiant le plus bas. La vérification et la correction des erreurs par le protocole CAN sont excellentes. Le nombre de nœuds peut varier de 0 à 130 sur le même bus CAN dont la longueur varie de quelques mètres à quelques kilomètres. Vitesse de transmission et distance entre les nœuds finaux sont inversement proportionnelles. Au-dessus de quelques centaines de mètres, pour assurer l'immunité au bruit et l'intégrité du signal, il convient d'utiliser un répéteur de bus CAN. Le nombre maximum de nœuds autorisé sur un bus CAN 2.0 typique est d'environ 120, ce qui reste bien supérieur aux 32 que propose une alternative comme le MODBUS RS485.

débit	bus	tronçon de câble
1000 kbit/sec	40 m	0,3 m
500 kbit/sec	110 m	0,3 m
250 Kbit/sec	240 m	0,3 m
125 Kbit/sec	500 m	0,3 m
100 Kbit/sec	500 m	0,3 m
50 Kbit/sec	1300 m	0,3 m
20 Kbit/sec	3300 m	0,3 m
10 Kbit/sec	6600 m	0,3 m
5 Kbit/sec	30 km	0,3 m

Surveillance des panneaux photovoltaïques (PV)

Le bus CAN permet de surveiller jusqu'à 120 panneaux et lire leurs performances individuelles, c'est-à-dire la tension et le courant contribués par chacun d'eux au courant continu total fourni à l'onduleur. Le schéma de principe (**fig. 1**) du système de surveillance des performances des cellules photovoltaïques connectées en parallèle : elles produisent la même tension, mais leurs courants de sortie divergent. Un simple capteur de courant est connecté à la borne de charge de chaque

panneau photovoltaïque. L'ACS712 est une sonde de courant à effet Hall. La version -30 peut mesurer jusqu'à 30 A. La bande de barrage permet de connecter l'appareil en série avec le circuit d'alimentation PV. En Chine, l'ACS712-30 coûte environ 2 \$/pièce. Pour lire l'intensité du courant continu d'un capteur de courant ACS712-30 dont la sortie est connectée à la broche A0 de l'Arduino UNO, il suffit de programmer :

```
load_current = 0.075714*(analogRead(A0) - 512)
```

L'alimentation sous 5 V de l'UNO est assurée par un petit régulateur abaisseur (2 \$/pièce) connecté au bus des panneaux (24 V). Son principal avantage est sa très faible dissipation de puissance.

Outre les saletés qui leur font de l'ombre, les cellules photovoltaïques souffrent des petites bulles d'humidité qui se forment à leur surface, de la décoloration de la surface, des microfissures et des points chauds (dits *hot spots*). La perte de puissance de la cellule résultant de ces facteurs peut atteindre 35 %, ce qui se traduit par des pertes d'intensité du courant. À défaut d'approfondir la mesure de la contribution *absolue* du panneau au courant, on peut se contenter de ce calcul :

```
load_current = analogRead(A0) - 512
```

Cette approximation est comparée à une référence définie ainsi qu'aux valeurs d'autres panneaux PV en vue de décider de déclencher ou non une alarme. Après avoir vu le signal d'alarme et le lieu (numéro), le préposé au nettoyage se rendra au panneau en question pour en vérifier l'état. Soit un bon nettoyage suffit, soit il faut chercher d'autres causes (points chauds, humidité excessive, etc.). Dans le pire des cas, le panneau entier doit être remplacé.

Bus Can 4*4

Le projet permet d'intégrer un clavier 4*4 sur le bus CAN, à des fins de commande et de saisie locale de chiffres. Les **figures 2** et **3** montrent le schéma et le dispositif expérimental sur l'établi de l'auteur. Vous verrez que le clavier est utilisé en option dans le logiciel du projet.

Le logiciel du projet

Le logiciel de ce projet prend la forme d'un ensemble de croquis Arduino hébergé sur le site d'Elektor Labs [1] avec des mises à jour et des notes de l'auteur. Le logiciel doit être considéré comme un point de départ, et non comme une application figée. Il entend stimuler votre propre ingéniosité et votre créativité maintenant que le premier pas est fait. On distingue maître et esclave. Un esclave lit l'intensité du courant continu sur le port A0 et la transmet au maître. Le maître a deux modes de fonctionnement : affichage et commande. Le maître CAN reçoit l'ID «1», les autres unités se voient attribuer «254» et «255» comme dans la figure 1, ou tout autre nombre entre 2 et 253. Le nombre maximum de nœuds est de 120.

Maître en mode d'affichage : Ceci est obtenu en utilisant la broche D10 des entrées/sorties numériques. Lorsque D10 est haute, le maître passe en mode d'affichage, dans lequel il collecte les données de tous les esclaves et pour les afficher en une structure de grille. Il détermine ensuite si un paramètre (p.ex. la puissance) est inférieur au niveau de consigne et émet une alarme pour le nettoyage de ce panneau particulier. Ce mode convient bien pour la surveillance des panneaux photovoltaïques.

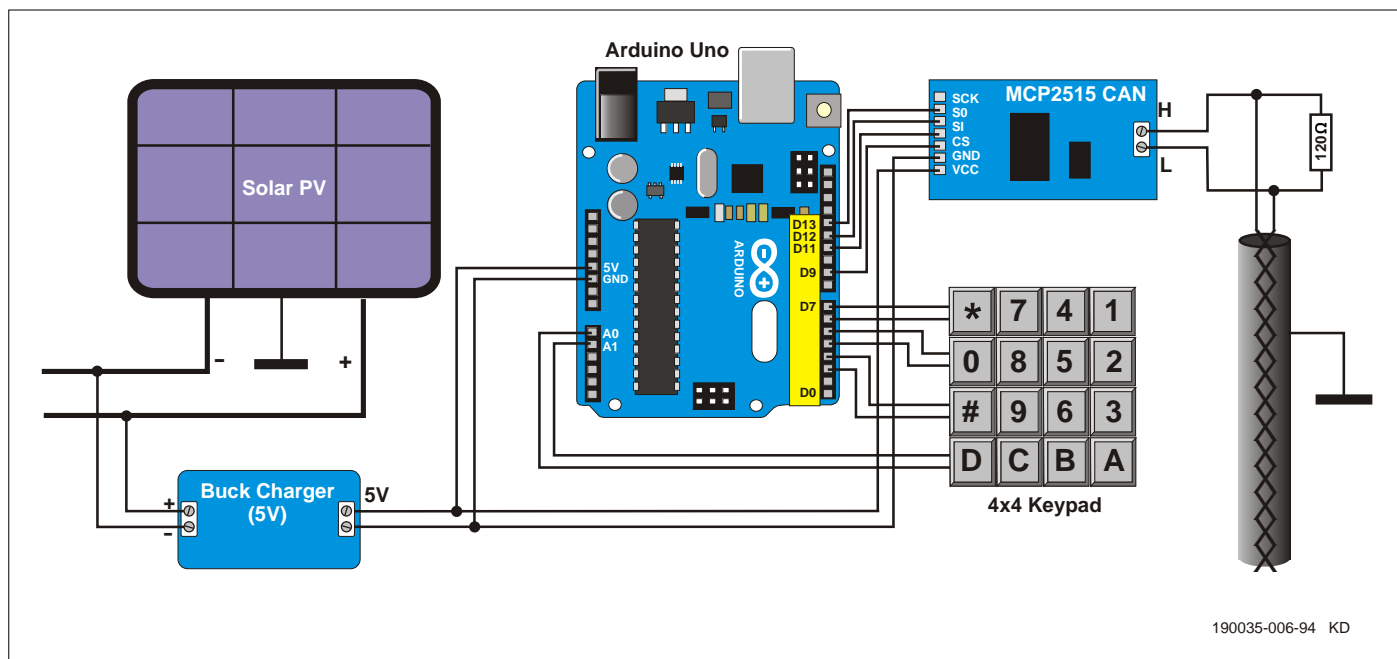


Figure 2. Schéma de principe du CAN Bus Master, avec un clavier 4x4 pour une utilisation en mode de commande.

Maître en mode de commande : Lorsque D10 est en l'air ou au niveau bas, le maître passe en mode de commande. Dans ce cas, les nœuds sont connectés à des relais commandés par les broches numériques D0 à 8. Un clavier CAN 4*4 est connecté sur l'appareil CAN maître (ID = 1). Le maître émet une chaîne de commande comme

25581#,25580#,13011#

etc. et les relais des nœuds respectifs sont activés ou désactivés pour commander, par exemple, une douche et/ou un essuie-glace connecté aux cellules PV. Exemples :

25580# : 255 = n° de nœud ; 8 = n° de broche numérique ;
0 = désactivé ; 1 = activé

25471# : 254^e nœud, relais connecté à la broche
numérique 7 et 1 = activé

25130# : 251^e nœud, relais connecté à la broche 3 et
0 = désactivé

Le caractère dièse (#) clot la commande.

Le maître du bus CAN peut avoir son propre capteur de courant PV, connecté à sa broche A0. Il peut tout aussi bien être dépourvu de tout panneau PV.

Votre application ?

Une autre utilisation de ce type de configuration consiste à déterminer, dans une petite ville ou un parc, combien de lampadaires sont effectivement allumés ou éteints. Un système tel que celui décrit ici permet également de contrôler de nombreuses

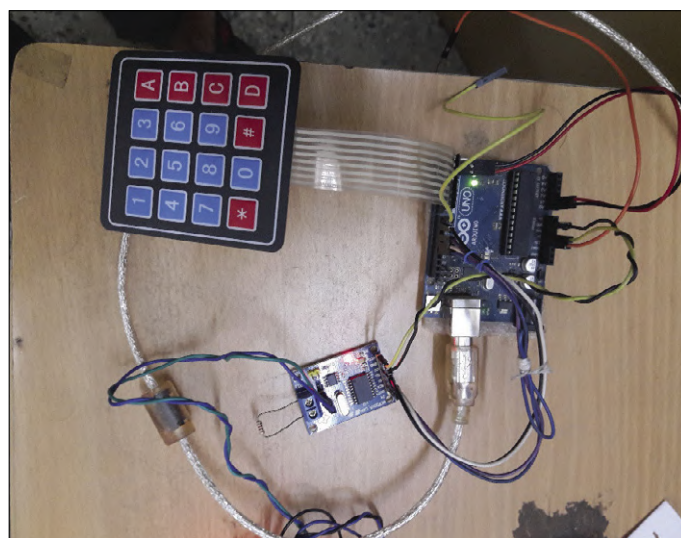


Figure 3. Version expérimentale du maître du bus CAN complétée par un clavier 4x4.

petites unités connectées sur le bus CAN à partir d'un bureau central. Parmi d'autres utilisations possibles, citons les boutons d'appel des chambres d'hôtel ou la commande de l'éclairage d'un hôtel. Au lieu de poser toutes sortes de câbles à l'intérieur du bâtiment, un seul bus CAN à paires torsadées suffit pour une commande et des mesures à la fois étendues et fiables. ◀

190035-02

Lien

[1] Projet sur Elektor Labs : www.elektormagazine.com/labs/can-bus-for-solar-pv-cell-monitoring



électronique analogique

Étude de cas n° 1 - section 3 :

Optimisation de la réponse du préamplificateur et compromis

Ton Giesberts (Elektor Labs)

Dans cette série d'articles, des experts de l'électronique analogique en éclairent divers aspects susceptibles de profiter à des lecteurs nés dans le numérique. L'analogique n'est pas de la magie noire. Dans la 3e et dernière section de cette étude, nous parlons du préampli de micro plutôt raffiné utilisé dans un projet d'Elektor appelé *détecteur de chauve-souris^{PLUS}*.

Si vous découvrez cette série, il est indispensable de lire les épisodes précédents [1] [2]. Pour que tout le monde puisse suivre, le circuit du préampli de micro dont nous allons parler a été extrait du schéma complet du *détecteur de chauve-souris-PLUS* ou Bat Detector^{PLUS} (fig. 1). Allumez vos lampes de poche et analysons la conception de ce circuit.

Plage dynamique

En raison du gain fixe élevé et de la faible tension d'alimentation du préampli, il y a un risque que la sortie sature et écrête les signaux. D'autre part, quand le signal est trop faible à certaines fréquences, le gain risque d'être insuffisant. La plage dynamique est déterminée d'un côté par le signal de sortie maximal sans distorsion, de l'autre par le plus petit signal utilisable distinct du niveau de bruit intrinsèque. Contrairement aux convertisseurs N/A et A/N, ici *rapport signal/bruit (S/B)* n'est pas synonyme de plage dynamique. Le rapport *S/B* des micros est généralement spécifié à une pression acoustique de 94 dB (1 pascal), mais pour la plupart des micros, il existe également une relation linéaire entre pression acoustique et tension de sortie, généralement jusqu'à 120 dB. Il reste de la distorsion (minime) et la plage dynamique est donc bien plus étendue que le rapport *S/B* spécifié. Pour notre détecteur, l'ajout d'une commande de gain rapide et automatique adaptée (voire limitée) au signal source serait un bon complément pour étendre cette plage.

Solution assez simple : le circuit classique avec un JFET comme résistance variable. Cela permettra également d'améliorer la détection de la modulation. Il n'y a pratiquement aucun danger d'accrochage (larsen) si les hautes fréquences sont bien filtrées au niveau de l'ampli de puissance et n'arrivent pas au haut-parleur – si toutefois cela devait poser un problème. Malheureusement, malgré un filtrage approprié, il peut arriver que les harmoniques du signal de sortie de l'ampli de puissance ainsi que le signal issu du haut-parleur donnent lieu à un

accrochage entre haut-parleur et micro. Il s'agit d'un problème d'ordre mécanique, car même le meilleur haut-parleur a ses défauts et colore le son. Avec ce détecteur de chauve-souris, il n'y a aucune exigence pour des signaux faibles contrairement à ce qui serait le cas avec un ampli audio. Un rapport *S/B* de 20 dB dans le signal injecté au comparateur devrait suffire.

Courants de polarisation

La résistance de 10 kΩ sur chaque entrée non-inverseuse peut vous faire sourciller. On fait ça habituellement pour compenser un décalage supplémentaire causé par les courants de repos à l'entrée de l'ampli op. Même dans la plage de température maximale d'un TL06x, ces courants à travers les entrées JFET n'atteignent pas 5 nA. La faible chute de tension (0,5 mV) de ces 5 nA à travers une résistance de 100 kΩ est donc presque négligeable, d'autant qu'il n'y a aucun couplage continu entre les étages d'amplification. Ce serait différent avec des amplis op bipolaires ; les courants de repos peuvent s'élever à plusieurs centaines de nA ou même de μA. Dans ce cas, il faudrait en tenir compte, en fonction des valeurs de résistance bien sûr. En outre, ces résistances de 10 kΩ augmentent la sensibilité des étages d'amplification aux parasites. Envisagez de les remplacer par du fil.

Ampli op comparateur

Le circuit utilise également la moitié d'un TL062 comme comparateur. C'est un choix tentant quand il reste encore un ampli op inutilisé, tout comme le recours à une version double au lieu d'un ampli op unique. Un vrai comparateur ferait un circuit intégré de plus sur la liste des composants. Les comparateurs ont généralement une sortie à collecteur ouvert et ne sont pas toujours recommandés. Il faut commencer par s'inquiéter de la plage de tension d'entrée différentielle maximale. Est-elle suffisante ?



Ici oui. Il en va de même pour les amplis op à entrées CMOS et JFET. Avec les amplis op bipolaires comme les amplis op audio, cette plage est souvent limitée par un réseau supplémentaire pour protéger les transistors d'entrée, par exemple une paire de diodes en anti-parallèle ou des diodes Zener en anti-série. En outre, si les spécifications sont dépassées, la tension de sortie peut basculer. Un problème avec le TL062 vient de son décalage d'entrée maximal, qui peut atteindre 15 mV, voire 20 mV sur toute la plage de température. Le signal appliqué devra dépasser ce décalage avant que la sortie ne bascule. Dans le cas du détecteur, la sensibilité dépend de l'exemplaire du circuit intégré utilisé. Or nous ne voulons pas d'une telle dépendance. Un autre problème est le gain de tension différentielle qui est de 3000 au moins selon la fiche technique et habituellement de 6000, ce qui est insuffisant. Pour fonctionner correctement en comparateur, il faut au moins 10, voire 30 fois plus de gain. Sur un vrai comparateur, un LM311 ou un LM339, il est de 200.000. L'effet de la tension d'alimentation sur le gain différentiel n'est pas précisé dans les fiches techniques, mais il serait intéressant de savoir s'il affecte les $\pm 4,5$ V plutôt faibles de la batterie de 9 V. Pour produire des ondes rectangulaires correctes à des fréquences allant jusqu'à 120 kHz, il faut une plus grande largeur de bande et certainement une pente $V/\mu s$ (temps de montée = *slew rate*) plus raide. Avec un comparateur comme le LM311, cette pente dépend du signal différentiel et, en fait, de la surcharge. Plus l'amplitude du signal d'entrée est grande, plus la tension de sortie change rapidement, mais même à 2 mV, ce sont 30 $V/\mu s$ principalement dus au transistor de sortie, et non à la résistance de polarisation haute (*pull up*) ou basse (*pull down*). En comparaison, la pente de 3,5 $V/\mu s$ du TL062 dans notre circuit est modeste.

Électret à 2 ou 3 broches ?

La capsule du microphone à électret nous est familière. La plupart n'ont que deux connexions. L'une est la masse (GND), reliée au boîtier métallique (blindage). L'autre est le signal de sortie prélevé sur le drain (d) d'un JFET intégré. La source (s) de JFET est reliée à la masse. Une membrane polarisée convertit le signal sonore en tension sur la grille du JFET (g). Une résistance entre l'entrée de l'amplificateur du micro et l'alimentation (qui est généralement assez faible, entre 1,5 et 5 V) polarise le FET. Une amplification de l'ordre de 10 dB se produit déjà à l'intérieur du micro à électret. Son inconvénient est la

pente non linéaire d'un FET. La pente ou transconductance g_m est le rapport I_{ds}/U_{gs} soit la relation entre le courant de drain et la tension de grille. Comme cette relation est non linéaire, le signal de sortie s'écarte de la tension de la membrane, ce qui provoque une légère distorsion.

Avec de petits signaux, cet effet est faible, mais à mesure que la pression sonore augmente, la distorsion aussi, et devient même visible sur un oscilloscope. De plus, la phase du signal de sortie peut être inversée, mais cela ne pose généralement pas de problème. Il suffit d'inverser la polarisation de la membrane. Avec les micros à électret à 3 broches, la source du FET est également disponible sur la troisième connexion (**fig. 2**). Nous ne parlons pas ici des types avec amplificateur intégré qui nécessitent une véritable tension d'alimentation plutôt qu'une polarisation. En utilisant cette troisième connexion, le FET peut être configuré comme un suiveur de source, ce qui réduit la distorsion. Malheureusement, dans cette configuration, le signal de sortie est plus faible. La connexion de drain est directement reliée à l'alimentation positive et la source est mise à la masse. La valeur de la résistance de polarisation vers le haut dans la configuration à 2 broches, et de la résistance de polarisation vers le bas dans la configuration à 3 broches, dépend bien sûr du type de micro à électret et de la tension d'alimentation. Vérifiez toujours sur la fiche technique. Dans un micro MEMS, il y a toujours un circuit intégré, pour lequel il faut une vraie alimentation. Il existe des astuces pour établir une connexion à 2 fils avec un micro MEMS en plaçant une charge

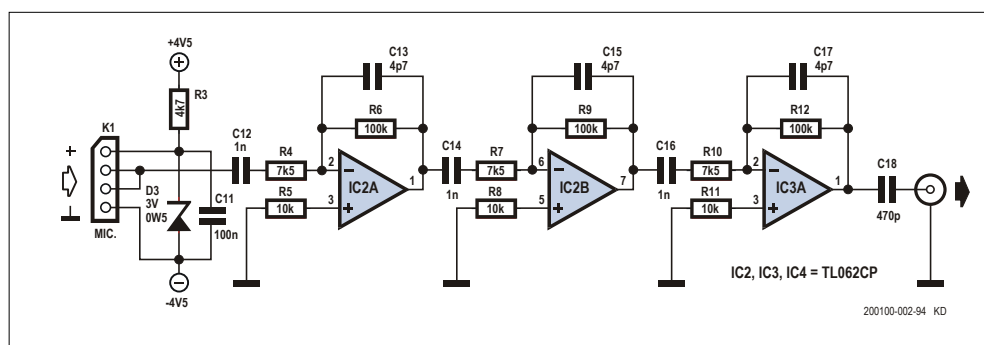


Figure 1. Le préamplificateur de microphone à 3 étages intégré dans le *détecteur de chauve-souris^{PLUS}* d'Elektor présente des caractéristiques remarquables.

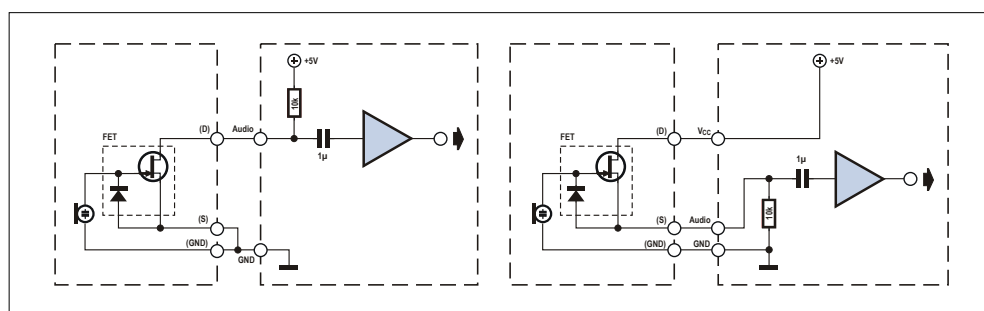


Figure 2. Schéma de connexion d'un microphone à électret à 2 broches (à gauche) et d'une version à 3 broches (à droite). Dans un électret à 2 broches, la source du FET et la masse sont déjà interconnectées.

à la sortie pour faire varier le courant d'alimentation, mais ne me demandez pas ce que deviennent les caractéristiques du micro dans ce cas....

Calcul du bruit d'entrée

Pour connaître le bruit, nous devons calculer en partant d'une largeur de bande de 120 kHz, au lieu de 20 kHz, la plage par défaut pour l'audio. Souvent le paramètre Bruit n'est mentionné qu'en pondération A, ce qui ne facilite rien. La formule bien connue qui représente le bruit en fonction de la température, de la résistance et de la largeur de bande nous dit :

$$\text{Bruit} = \sqrt{4 k T B R} \quad (\text{Eq. 1})$$

avec

k = constante de Boltzmann ($1,38 \times 10^{-23} \text{ J K}^{-1}$)

T = température absolue (en K, on suppose 270 K)

B = largeur de bande applicable (en Hz)

R = résistance (en Ω)

Le bruit total à l'entrée est composé du bruit déjà présent dans le signal source (U_{bruit1}) et du bruit produit par les résistances (U_{bruit2}) et par l'ampli op utilisé (U_{bruit3}).

$$U_{\text{bruit(total)}} = \sqrt{(U_{\text{bruit1}})^2 + (U_{\text{bruit2}})^2 + (U_{\text{bruit3}})^2} \quad (\text{Eq. 2})$$

Selon la fiche technique, le bruit du micro que nous avons utilisé a été mesuré avec un filtre pondéré A. Ce filtre n'est malheureusement pas spécifié pour les fréquences supérieures à 20 kHz, or c'est justement la zone ultrasonique dont nous avons besoin. Il est même recommandé d'utiliser un filtre à forte pente de 20 kHz dans les sonomètres combiné avec le filtre pondéré A. Pour nous, les indications de la fiche technique n'ont aucune pertinence. Ce qui est sûr, c'est que le bruit sera considérablement plus élevé pour notre calcul. Formulons une hypothèse. Si nous comparons le filtre pondéré A avec un filtre passe-bas de 10 kHz du premier ordre, le bruit dans une largeur de bande de 120 kHz est au moins multiplié par $\sqrt{(120 / 10)} = 3,5$ fois, soit une estimation (forcément) approximative de 11 dB de plus pour notre calcul que ce qui est indiqué dans la fiche technique. Par conséquent, un rapport S/B de 59 dB(A) à un niveau de pression acoustique de 94 dB (1 pascal, Pa) et une bande passante de 120 kHz est alors estimée à 48 dB si la mesure n'est pas filtrée. À cette pression sonore, un niveau de sortie de -42 dBV est donné pour 1 kHz, soit 8 mV. À 1845 fois le gain, il donne 14,76 V, ce qui signifie que l'ampli est amplement surmodulé. Mais dans les ultrasons, au-dessus de 40 kHz, le niveau de sortie de notre micro MEMS diminue. Entre 10 et 20 kHz, l'augmentation est de plus de 10 dB. Le degré de surmodulation varie donc fortement selon la fréquence et devient presque imprévisible. Un calcul basé sur un comportement linéaire du bruit n'est donc pas exactement ce dont nous aurions besoin pour le fonctionnement du détecteur, mais nous nous contenterons de cette estimation.

Il serait certes préférable d'ajuster la réponse en fréquence de l'amplificateur au micro. À quoi ressemblerait alors le calcul du bruit ? S'il devait être possible de changer de micro, seule une solution numérique serait envisageable. Mais revenons au calcul, maintenant pour le bruit par source à la sortie. Le bruit du micro à une pression sonore de 94 dB est alors de :

$$8 / (10^{(48/20)}) \times (100 / 7,5) = 0,425 \text{ mV}$$

Pour la contribution des résistances au bruit, nous devons calculer avec le circuit parallèle des deux qui déterminent le gain, c'est-à-dire celles de 100 k Ω et 7,5 k Ω . Nous omettons

la résistance dans le chemin de l'entrée non-inverseuse, en la supposant remplacée par un fil :

$$\sqrt{(4 \times 1,38 \times 10^{-23} \times 270 \times 120000 \times 6977) \times (100 / 7,5)} = 0,047 \text{ mV}$$

Le bruit de l'ampli op est amplifié comme avec un amplificateur non inverseur. Le bruit d'un TL06x est généralement de 42 nV/ $\sqrt{\text{Hz}}$, soit :

$$(42 \text{ nV} \times \sqrt{120000}) \times (1 + 100 / 7,5) = 0,21 \text{ mV}$$

Le niveau de bruit total à la sortie du premier étage d'amplification du détecteur de chauve-souris s'élève à :

$$\sqrt{(0,425^2 + 0,047^2 + 0,21^2)} = 0,476 \text{ mV}$$

Ce qui donne un rapport S/B de

$$(100/7,5) \times 8 \text{ mV} / 0,476 \text{ mV} = 224,1$$

soit 47 dB. En d'autres termes, le bruit à la sortie de l'amplificateur de microphone est presque exclusivement déterminé par le micro MEMS. Avec une amplification suffisante, la contribution des étages suivants est négligeable. Si le gain par section diminue, les étages d'amplification suivants auront également un impact. C'est une autre raison pour laquelle huit sections à faible gain ne sont pas le meilleur choix. Et encore une fois, ce calcul n'est valable que si la distribution du bruit est linéaire et non comme approximativement la caractéristique de fréquence.

Micro MEMS à même le circuit imprimé

Petite digression. À quoi ressemblerait l'application pratique d'un microphone MEMS monté directement sur le PCB ? Les réflexions et questions suivantes peuvent servir de point de départ.

1. Le montage d'un microphone MEMS sur le circuit imprimé doit ménager un bon passage du port acoustique vers l'air ambiant.
2. Un joint en caoutchouc ou autre chose est-il nécessaire ? Est-il possible d'obtenir l'étanchéité du boîtier du détecteur afin que l'humidité, la poussière, etc. ne pénètrent pas dans le microphone MEMS ?
3. Qu'est-ce qui peut affecter la réponse en fréquence et la sensibilité ? Il est préférable d'utiliser un microphone MEMS dont l'orifice se trouve sous le composant, à condition bien sûr de prévoir un trou à cet endroit dans le circuit imprimé à même lequel sera monté le micro. On aura le chemin acoustique le plus court possible vers l'extérieur. Ceci implique l'utilisation de composants CMS.
4. Si on renonce à utiliser d'autres microphones, il y a lieu de rectifier la caractéristique de fréquence du MEMS (de sorte qu'aucun autre microphone ne soit nécessaire). Un traitement numérique sera plus judicieux qu'une approche analogique. Il est si facile de modifier un microprogramme pour apporter des améliorations, obtenir des caractéristiques de filtrage différentes (par exemple, une bande de fréquence spécifique ou pour compenser des tolérances).

Puisque cette étude analogique se termine paradoxalement ci-dessus sur une incitation à recourir au traitement numérique, voici une question pour vous : comment vous y prendriez-vous pour déterminer avec une certaine précision la caractéristique de fréquence réelle du préampli pour microphone MEMS sur le fameux *détecteur de chauve-souris*^{PLUS} dans sa version actuelle ? Dans la prochaine édition d'Elektor, nous aborderons un autre sujet *de manière totalement analogique*. ◀

200100-02

bruits de labo



Plus d'un siècle d'histoire de l'électronique ! Ça a laissé des traces. Même si nombre de montages sont rapidement tombés aux oubliettes, d'autres continuent à vivre et à occuper des passionnés....

Du plus simple au plus complexe, de la mesure à la commande, le labo d'Elektor est un lieu pour tous les projets électroniques. Jetez un coup d'œil à cette sélection, il y a peut-être quelque chose que vous pouvez utiliser.

Construisez un « voleur » à transistor

Le « voleur de joule » est un circuit très simple qui vous permet d'exploiter jusqu'à l'ultime limite l'énergie stockée dans une batterie monocellule. Comme ce circuit est très simple, il passionne beaucoup d'amateurs d'électronique. D'après Wikipédia, son concept remonte à près d'un siècle. Voici une variante un peu plus compliquée sous la forme d'un « voleur » à transistor. (Lire les commentaires.)



@ Elektor Labs: <https://goo.gl/ZDEfM3>

Réalisez un thérémine entièrement analogique

Inventé il y a une centaine d'années, le thérémine reste un instrument de musique électronique très populaire. Les versions les plus performantes disponibles dans le commerce sont coûteuses, ce qui rend leur réalisation « maison » extrêmement séduisante. Pourtant, il n'est pas si facile de parvenir à réaliser un circuit reproductible entièrement analogique. Avez-vous des idées à nous soumettre ?



@ Elektor Labs: <https://goo.gl/pq8Dc5>

Construisez-vous un micro-chargeur solaire de batteries

Même si l'utilisation du soleil pour recharger des batteries n'a rien de nouveau, un circuit simple pouvant être assemblé rapidement un jour de pluie où il n'y a rien d'autre à faire est toujours le bienvenu. En voici un, basé sur un petit microcontrôleur ATtiny13 qui se charge d'orchestrer le tout...



@ Elektor Labs: <https://goo.gl/a6eZnM>

Sauvons la lampe à lave

« Si vous achetez ma lampe, vous n'aurez plus besoin de médicaments ». Tel était le slogan accrocheur d'Edward C. Walker pour commercialiser l'Astro Lamp qu'il avait inventée au début des années 1960. Plus de 50 ans après, cette invention, plus souvent appelée « lampe à lave » à cause de ses bulles multicolores montant et descendant lentement dans un liquide, a encore son contingent d'adeptes.... ◀

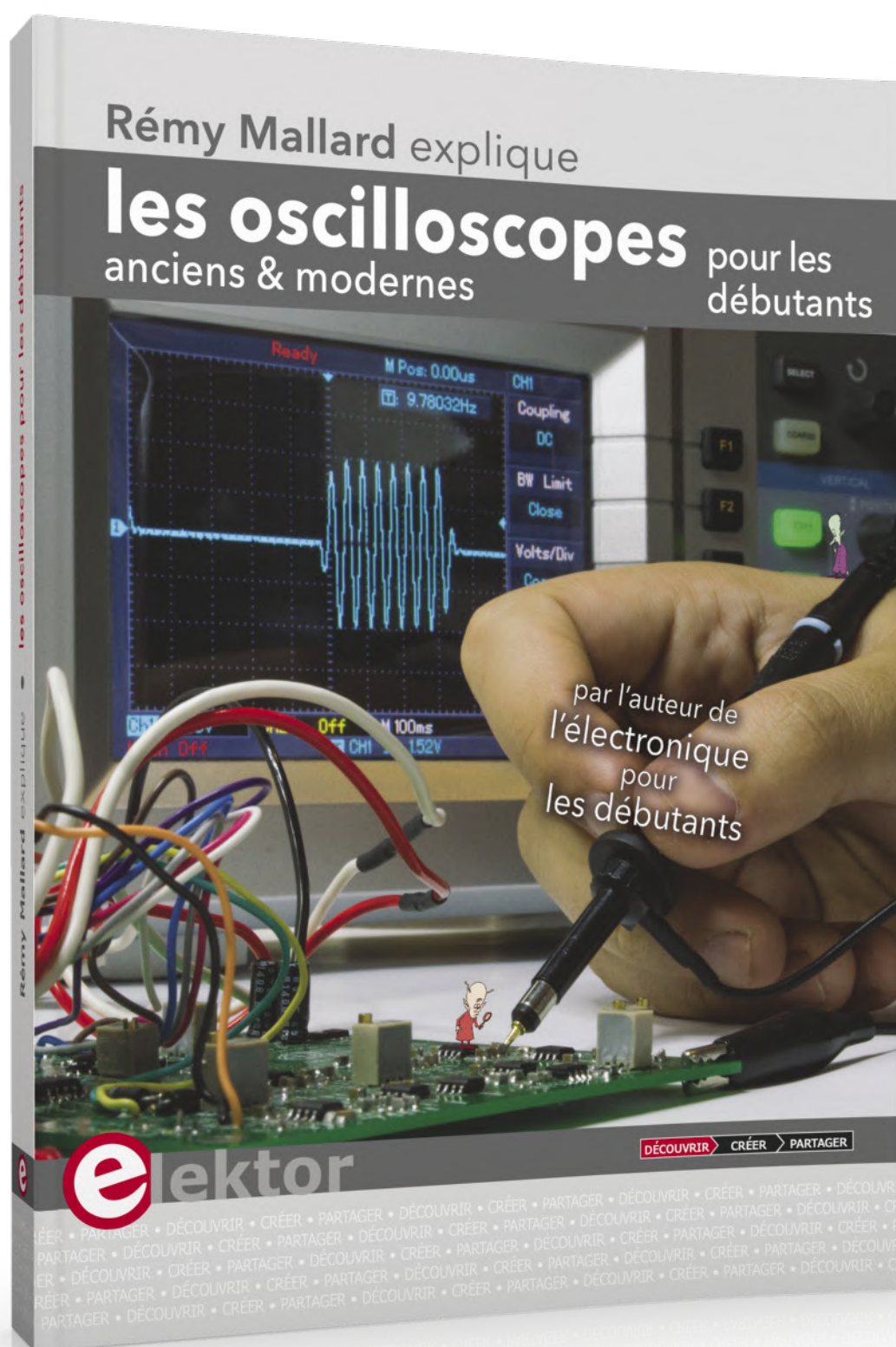


@ Elektor Labs: <https://goo.gl/a6eZnM>

(180461-03)

Livre

« Oscilloscopes anciens et modernes » de Rémy Mallard



Pour rendre compte du contenu impressionnant du livre de Rémy Mallard sur les oscilloscopes paru récemment, il nous a semblé intéressant d'en reproduire un chapitre pour nos lecteurs. En accord avec l'auteur, nous avons retenu un extrait représentatif du ch. 6 (voir la table des matières complète ci-dessous pour situer l'extrait). Il traite de l'analyse de protocole et du décodage de données série à l'aide d'un oscilloscope. L'idée de la citation d'un long passage est bonne en elle-même, mais comporte en germe le risque de donner du contenu une impression faussée. En effet, comment un extrait aussi condensé pourrait-il rendre compte d'un livre aussi riche ? Les paragraphes de la section 6.7 condensés ici en quelques pages comptent 13 pages en réalité!

6.7 – Analyse de protocole et décodage de données série (I²C, SPI, UART, etc.)

(I²C, SPI, SMBUS, UART, CAN/LIN, I²S, USB 2.0, DMX, etc.)

Dans ce chapitre, nous verrons comment « mesurer », « valider » et « décoder » des trames de données numériques véhiculées sur un bus de données série tel que RS232, I²C ou SPI. Les données peuvent être transmises sur une liaison de type asymétrique (RS232, I²C ou SPI par exemple) ou sur une liaison de type différentielle (RS485, USB ou Ethernet par exemple), et nous devons identifier les ressources nécessaires pour ces deux types de liaison. [...] Si les signaux électriques objets de l'analyse sont trop déformés, ou si vous indiquez à l'oscilloscope qu'il va recevoir des données USB alors que vous lui fournissez un signal de type SPI, il y a fort à parier que le pauvre ne va pas s'y retrouver ! Dans ce chapitre, nous aborderons ces deux points. Une fois la validité électrique établie, nous verrons comment « décoder » des signaux numériques émis sur des bus de données tels que bus I²C ou SPI, de façon manuelle ou de façon automatique (avec les outils de décodage automatique inclus dans les oscilloscopes numériques Rigol DS1054Z et Picoscope 2204).

6.7.1 – Vérification de la validité électrique des signaux

Ce chapitre est consacré à la mesure de signaux numériques, qui permet de vérifier la validité de trames de données série (amplitude globale et forme des transitions). Il peut sembler curieux de parler de mesures sur un bus de données numériques, car le terme « mesure » pourrait faire penser à « tension » ou « fréquence ». Pourtant, une analyse de données en vue d'un décodage ne peut être fiable que si la voie de transmission (la liaison) est fiable ; les données doivent être assez « propres » pour être correctement interprétées.

Tous les signaux électriques que l'on peut observer sur un écran d'oscilloscope sont par nature analogiques. Un signal qu'on dit numérique ne fait pas exception à la règle, **c'est** un signal analogique. Si on l'appelle signal numérique, c'est parce qu'on se contente d'utiliser un nombre réduit de valeurs de tension, parmi toutes les valeurs que ce signal pourrait prendre. Avec des circuits logiques (numériques) TTL par exemple, il est commun de travailler avec les deux valeurs 0 V et +5 V ou 0 V et +3,3 V. Quand la tension de sortie d'un circuit logique passe de 0 V (état logique bas) à +5 V (état logique haut) cela ne se fait pas de façon instantanée, mais de façon progressive (même si la transition est très rapide), nous l'avons vu au chapitre décrivant la méthode de mesure du temps de montée ou de descente d'un signal numérique. [...] Comme pour les signaux analogiques, les signaux logiques sont représentés en fonction du temps. Mais au lieu de stocker des valeurs numériques qui correspondent à l'amplitude des points acquis, on conserve seulement des états logiques binaires (0 ou 1). Comparé à une acquisition analogique effectuée sur 8 bits, cela est beaucoup plus rapide et demande 8 fois moins d'espace de stockage. Dit autrement, un système d'acquisition 8 bits permet l'acquisition et l'affichage simultané de 8 signaux numériques indépendants. Une autre manière encore d'exprimer les choses est de dire que chaque voie numérique possède une résolution verticale (d'amplitude) de 1 bit. [...]

Sur un bus série, on trouve un ensemble d'informations binaires (bits) qui se suivent à la queue leu leu. Ces informations peuvent être simplifiées à l'extrême, par exemple un seul bit transmet pour un ordre marche ou arrêt d'un unique appareil. Bien évidemment, cette solution très simple est également très limitée, et dans la

pratique on rencontrera de très nombreux cas de figure avec transmission d'un ou plusieurs octets (par exemple 8 bits de données éventuellement encadrés de bits de START et STOP). [...]

L'écran D de la figure 6.7.1.a met en évidence un problème de taille : le signal numérique montré sous sa forme analogique est

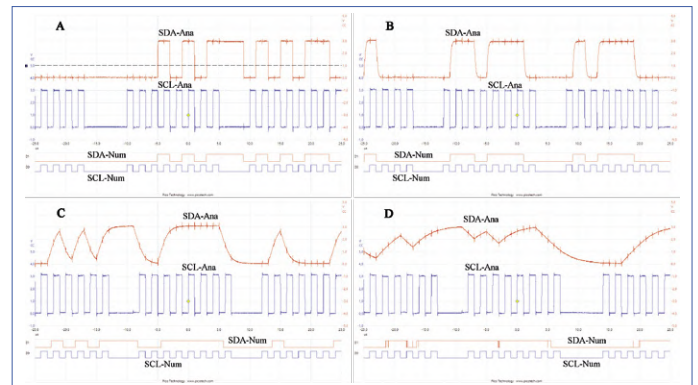


Figure 6.7.1.a – Dans chaque écran A à D, on peut voir un même signal SDA (ligne de données) sous ses formes analogique (SDA-Ana) et numérique (SDA-Num), et un même signal SCL (ligne d'horloge) sous ses formes analogique (SCL-Ana) et numérique (SCL-Num). L'écran A montre des signaux valides, l'écran B montre un signal de données légèrement déformé, et l'écran C montre qu'un signal moyennement dégradé peut encore être exploité. L'écran D quant à lui met en évidence un sérieux problème, le signal de données n'est plus exploitable du tout !

tellement déformé que les valeurs numériques (bits) qu'il transporte ne peuvent plus être correctement reconnues. Regardez attentivement la trace SDA-Num de l'écran D, on y voit à certains endroits des états bas ou hauts très brefs, de durée bien inférieure à une demi-période du signal d'horloge SCL-Num. Cela montre sans ambiguïté que le signal numérique souffre d'hésitations et qu'il n'est pas valide. Le problème est moins évident avec l'écran C, dont l'affichage des données sous forme numérique (SDA-Num) laisse penser que tout va bien, alors qu'en fait on est à la limite tolérée, et que seule la vue sous forme analogique (SDA-Ana) permet de s'en rendre compte.

Remarque : les petits « pics » visibles sur les traces SDA-Ana et SCL-Ana des quatre écrans A à D de la figure 6.7.1.a sont dus à un câblage imparfait du circuit sous test. Au départ, l'auteur voulait les supprimer en améliorant le câblage, mais il a finalement décidé de les laisser, car ils constituent des « marqueurs » qui coïncident avec les transitions d'horloge. Finalement, ces petits pics se révèlent peu gênants et fort pratiques (pour un exercice, pas pour un produit fini) ! [...]

Dans les chapitres qui vont suivre, nous allons étudier plusieurs types de liaisons, certaines sont de type différentiel et d'autres non, certaines sont avec fil d'horloge associé et d'autres non. Voyons de quoi il s'agit.

Une liaison asymétrique (non différentielle) est constituée d'un seul fil électrique pour transmettre des données (dans un sens ou dans un autre) en plus de l'indispensable fil de masse qui sert de référence de potentiel (0 V). [...]

Une liaison symétrique (différentielle) ou une boucle de courant est constituée de deux fils électriques pour transmettre les données (fils Data+ et Data-). Un fil de masse peut y être associé, mais il n'est pas utilisé comme référence de potentiel. [...]

Une liaison sans fil d'horloge associé (dans le câble de liaison) signifie que les données (*Data*) sont transmises seules, sans fil supplémentaire qui pourrait véhiculer un signal de synchronisation : le récepteur qui reçoit les données (*Data*) doit se débrouiller pour reconnaître le début des messages et en extraire les bits d'information. Une telle liaison peut se faire en mode asymétrique (1 seul fil de données) ou en mode symétrique (2 fils de données). [...]

Une liaison avec fil d'horloge associé possède, en plus du fil de liaison des données (*Data*), un fil supplémentaire pour transporter l'horloge (*Clock*) qui permettra au récepteur de se caler sur les données. Ce mode de liaison permet de s'affranchir des contraintes de vitesse imposées des deux côtés. [...]

6.7.2 – Décodage de données série (généralités)

Remarques préalables :

- L'analyse de données série impose l'usage d'un oscilloscope numérique, qui permet de capturer (memoriser) une ou plusieurs trames complètes et de les visualiser après coup. Un oscilloscope analogique traditionnel (sans mémoire) ne le permet pas. [...]
- Un oscilloscope numérique possède une certaine quantité (profondeur) de mémoire. De cette quantité de mémoire dépendra la durée maximale de capture d'un ensemble de données numériques. [...]
- La liste des protocoles de données série pris en charge par les oscilloscopes numériques diffère selon leur modèle. Les exemples proposés ci-après correspondent à des modèles d'oscilloscope précis, et si votre appareil est différent vous devrez consulter son manuel utilisateur. [...]

Un analyseur logique est un système qui mémorise une suite de transitions ou de niveaux logiques (bits à 0 ou 1), puis permet ensuite de les afficher sur écran pour analyse fine. Le nombre d'états logiques pouvant être capturés dépend là encore de la quantité (profondeur) de mémoire mise à disposition.

Un analyseur de protocole est une forme évoluée d'analyseur logique. Il permet de reconnaître l'organisation des paquets d'informations binaires capturés (principalement par la détection des début et fin de paquet qui sont facilement reconnaissables) et de les décoder pour afficher en clair leur valeur sous forme numérique, généralement au format décimal ou hexadécimal. [...]

6.7.3 – Décodage de données série I²C

Les échanges de données sur un bus I²C s'effectuent sur « 2 fils » SDA (**S**erial **D**Ata, données) et SCL (**S**erial **C**Lock, horloge). Pour effectuer un décodage de façon manuelle ou automatique, nous avons besoin de visualiser des deux signaux DATA et CLOCK en même temps à l'écran. Heureusement, un oscilloscope à deux voies suffit. [...] Avant d'analyser une capture de données série I²C, rappelons la méthode de transfert d'un point de vue général :

- 1 - Le maître élabore une condition de début de transmission, également appelé bit de start, qui indique aux périphériques que des données vont être envoyées. Ces périphériques se mettent alors en mode d'écoute active.
- 2 - Une adresse codée sur 7 bits est envoyée juste après le bit de start. Chaque périphérique possède une adresse personnelle (différente de celle des autres périphériques) et à cet instant tous les périphériques écoutent puisqu'ils ne savent pas encore à qui s'adresse le message. Une fois les 7 bits d'adresse transmis, un seul périphérique tiendra compte des données qui suivront : c'est celui dont l'adresse personnelle correspond à

l'adresse envoyée par le maître.

- 3 - Un bit « lecture/écriture » est transmis par le maître pour indiquer si l'opération souhaitée est une opération de lecture (bit à 1) ou d'écriture (bit à 0). [...]
- 4 - Selon le type de périphérique I²C sollicité, on peut ensuite avoir transmission de données dans un seul sens (commande) ou dans les deux sens (question/réponse), sur le fil SDA. Les données sont regroupées par paquets de 8 bits, un paquet équivalant à un octet (*byte*). Le nombre d'octets transmis peut être plus ou moins élevé et dépend là encore du type de périphérique I²C. [...]
- 5 - Après transmission (unilatérale ou bidirectionnelle) de tous les octets requis, une condition de fin, également appelée bit de stop, est élaborée. Ce bit de stop indique la fin du message. [...]

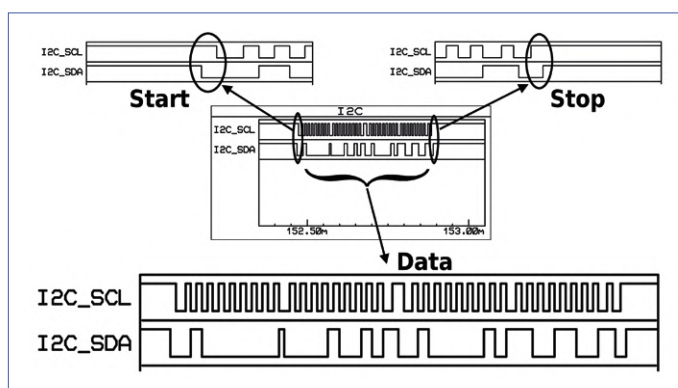


Figure 6.7.3.a – Au repos, les lignes de données SDA et d'horloge SCL sont toutes deux à l'état logique haut. Le bit de start correspond au changement d'état haut-bas de la ligne de données SDA immédiatement suivi du changement d'état haut-bas de la ligne d'horloge SCL. Le bit de stop correspond au changement d'état bas-haut de la ligne d'horloge SCL immédiatement suivi du changement d'état bas-haut de la ligne de données SDA. Les données utiles (*Data*) sont transmises entre les bits de start et de stop.

L'exemple de la figure 6.7.3.a montre des signaux d'horloge (SCL) régulièrement espacés, les messages I²C ayant été élaborés grâce à un module hard (MSSP) inclus dans un microcontrôleur PIC. Cette grande régularité se retrouve rarement quand les messages I²C sont élaborés de façon purement logicielle avec des *délais*, mais l'absence de non-régularité n'est généralement pas problématique si elle affecte en même temps les deux lignes SDA (*Data*) et SCL (*Clock*).

6.7.3.1 – Décodage manuel de données série I²C

Nous allons utiliser la trame de données visible en figure 6.7.3.a pour nous faire la main, voyons maintenant la figure 6.7.3.1.a qui la décortique en détail.

Pour commencer, nous allons essayer de retrouver les différents paquets de bits (octets) contenus dans la trame visible en figure 6.7.3.1.a, qui correspond à la lecture de l'état des lignes d'entrée/sortie d'un expandeur de port MCP23008. De façon purement visuelle et intuitive, nous constatons qu'il y a 4 octets dans cette trame (4 « gros paquets » de bits/coups d'horloge sur la ligne SCL). Maintenant, nous allons tracer un trait vertical en pointillés en regard de chaque front montant d'horloge sur la ligne SCL. Si en regard de ce trait vertical on découvre un état bas sur la ligne SDA, cela signifie que la valeur de la donnée correspondante est à 0. Si au contraire en face de ce trait vertical on a un état haut

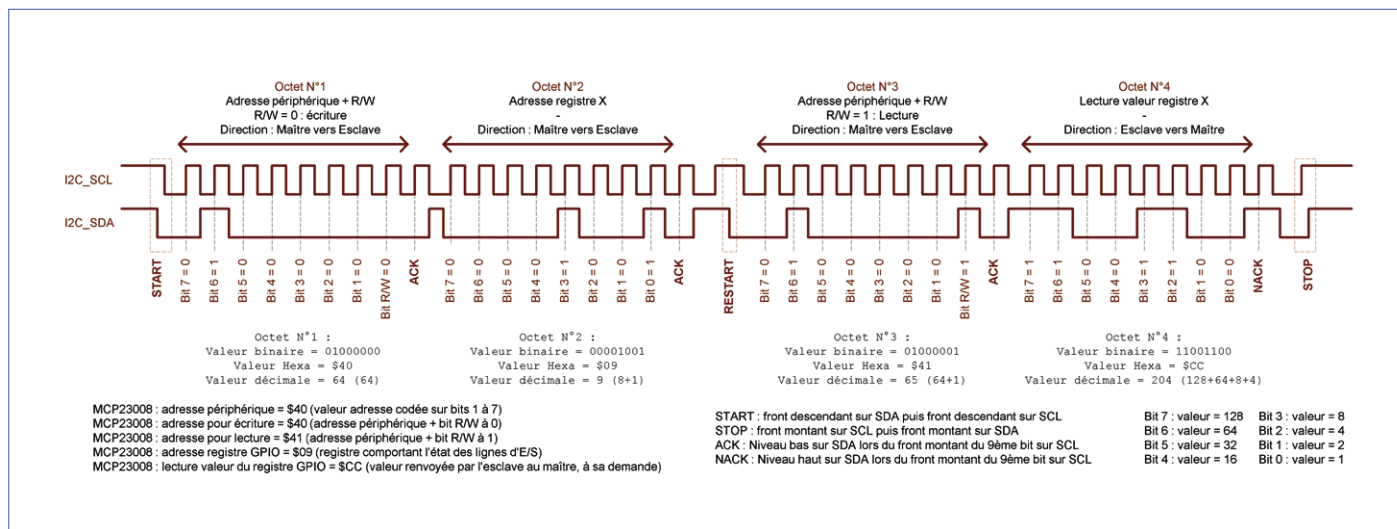


Figure 6.7.3.1.a – Même sans analyseur de protocole, on peut facilement lire la valeur des octets transmis dans une trame de données. Cela demande simplement un peu de temps...

sur la ligne SDA, cela signifie que la valeur de la donnée correspondante est à 1. En analysant de la sorte les 8 bits de chaque paquet, on peut en déduire la valeur de chaque octet (qu'on pourra exprimer en binaire, hexadécimal ou décimal). Le 9e coup d'horloge correspond à l'acquittement (ACK) ou au non-acquittement (NACK) de l'octet complet. C'est pourquoi on trouve 9 coups d'horloge pour chaque octet transmis. Comme vous pouvez le constater, la lecture des octets n'est pas difficile et demande simplement un peu de patience.

6.7.3.2 – Décodage automatique de données série I²C avec Picoscope 2204

L'exercice précédent a montré qu'il n'était pas difficile de décoder des octets bit par bit pour en déduire manuellement leur valeur. Maintenant, nous allons profiter d'une option de décodage de données opérée par l'analyseur de protocole inclus dans le logiciel Picoscope de PicoTech. La figure 6.7.3.2.a montre un exemple de décodage après capture. Les données ont été capturées dans un premier temps, puis dans un second temps le mécanisme de décodage série a été mis en œuvre.

Une fois l'acquisition faite des données à décoder, voici la procédure à suivre :

- Menu Outils > Décodage en série. La fenêtre « Décodage en série » apparaît ;
- Dans l'écran « Décodage en série », presser le bouton « Créer ». Dans la liste des protocoles pris en charge, sélectionner « I²C ». La fenêtre « I²C » apparaît ;
- Dans la zone « Canaux » de la fenêtre « I²C », spécifier l'entrée de l'oscilloscope qui reçoit le signal de données (liste déroulante Data) et spécifier l'entrée de l'oscilloscope qui reçoit le signal d'horloge (liste déroulante Clock). Dans notre exemple, il s'agit des entrées logiques D1 et D0, respectivement.
- Presser le bouton OK des deux fenêtres ouvertes, pour les fermer. Le décodage série démarre et l'on voit le résultat s'afficher à l'écran.

Les octets décodés sont visualisés sous trois formes :

- graphique (le décodage peut se faire depuis les entrées analogiques ou logiques)

- valeurs numériques dans des « paquets » avec libellé « Address » ou « Data »
- table horodatée avec indications détaillées pour chaque octet [...]

La procédure décrite ici est succincte, car l'objet n'était pas de faire un tutoriel. Le manuel utilisateur de PicoTech est suffisant, et il évoluera certainement au fil des versions.

6.7.3.3 – Décodage automatique de données série I²C avec Rigol DS1054Z

[...] Le décodage des données série I²C est assez intuitif, une fois qu'on a compris que les options de décodage sont accessibles par le bouton MATH situé en face avant de l'appareil. Lors de la première utilisation de l'option de décodage, l'auteur est resté perplexe devant les données décodées, qui ne semblaient pas conformes à celles attendues (comparées en parallèle avec le décodage série du Picoscope). Tout fut rentré dans l'ordre une fois que l'auteur eut configuré le mode d'affichage sur HEXA et non plus sur ASCII

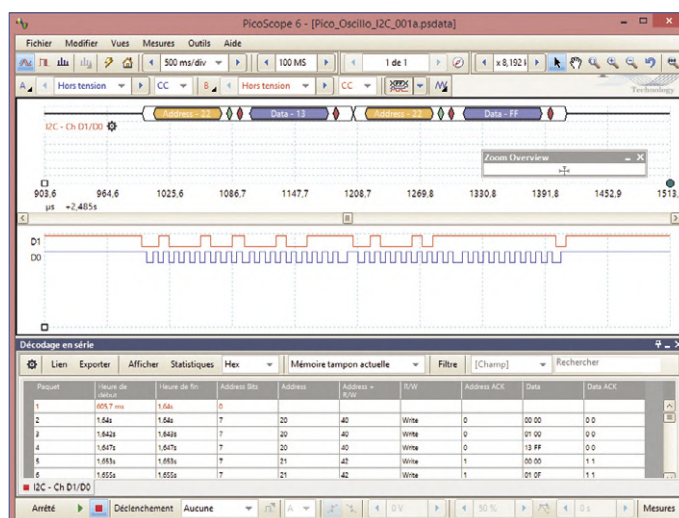


Figure 6.7.3.2.a – Le décodage série d'un flux I²C est très aisé avec l'option de décodage incluse dans le logiciel Picoscope (ici en version V6).

(format proposé par défaut). La figure 6.7.3.3.a montre le résultat d'un décodage opéré sur un flux I²C cadencé à 100 kHz. Les données ont été acquises sur une durée de plusieurs secondes, puis décodées après coup.

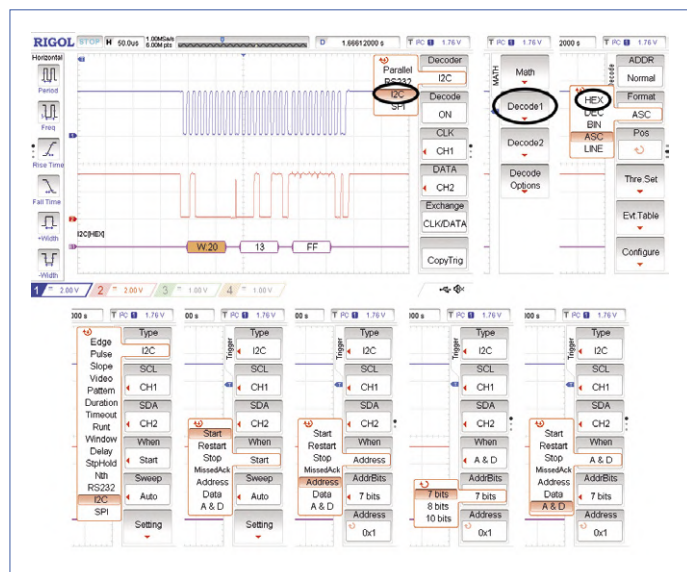


Figure 6.7.3.3.a – L’affichage des données décodées est très clair, et l’on peut choisir le format d’affichage (hexadécimal, décimal, binaire, ASCII). À de rares exceptions, l’auteur préfère le format hexadécimal qui permet une comparaison/vérification plus rapide par rapport au tracé. Les options de déclenchement (*Trigger*) permettent en outre de définir les conditions de capture de la trame I²C (filtrage sur adresse par exemple).

6.7.4 – Décodage de données série de données SPI

Les échanges de données sur un bus SPI peuvent s’effectuer sur « 3 fils » ou « 4 fils » :

- Liaison SPI « 3 fils » => SDIO ou SDA, CLOCK et CS ou SS ou CE
- Liaison SPI « 4 fils » => MOSI, MISO, CLOCK et CS ou SS ou CE.

Le fil d’horloge est toujours sous le contrôle du maître. Pour rappel :

- SDIO = **S**erial **D**ata **I**n **O**ut, (maître vers esclave et esclave vers maître)
- SDA = **S**erial **D**ata (maître vers esclave et esclave vers maître)
- MOSI = **M**aster **O**ut **S**lave **I**n (maître vers esclave seulement)
- MISO = **M**aster **I**n **S**lave **O**ut (esclave vers maître seulement)

Dans les exercices qui suivent, nous utiliserons des données transmises en mode « 4 fils », le mode de travail en « 3 fils » étant similaire au bus I²C avec son fil de liaison DATA bidirectionnel. Le mode « 4 fils » offre l’avantage d’une liaison *full duplex* (bidirectionnelle), le maître et l’esclave peuvent émettre des données et en recevoir en même temps.

Remarque : l’idéal ici est de posséder un oscilloscope à quatre voies, qui permet de montrer simultanément les données MISO, MOSI, CLOCK et CS. Si votre oscilloscope ne possède que deux voies, alors vous ne pourrez observer que les couples MOSI+CLOCK ou MISO+CLOCK. Ne comptez pas trop vous en sortir en visualisant le couple MISO+MOSI, en pratique c’est difficile, voire impossible [...]

Astuce : quand plusieurs périphériques sont câblés sur un même bus SPI, tous reçoivent en même temps les données véhiculées

par le bus, alors qu’un seul d’entre eux est réellement concerné (ou éventuellement plusieurs, s’ils sont chaînés). Si vous voulez capturer des données spécifiques à un seul périphérique, vous pouvez utiliser son fil de sélection de boîtier (CS, SS ou CE) pour déclencher la trace de l’oscilloscope, ou tout du moins utiliser ses changements d’état affichés à l’écran pour repérer « qui est où » (à quel périphérique les données sont destinées). [...]

Quelques faits nouveaux doivent être portés à votre connaissance :

- en SPI, il n’y a pas de bit d’acquittement (ACK) ni de non-acquittement (NACK). On aura donc 8 impulsions d’horloge par paquet (par octet) et non 9 comme en I²C. L’absence d’acquittement implique que le maître peut envoyer des données sans se rendre compte qu’il parle dans le vide (qu’aucun esclave ne lui répond).
- les impulsions du signal d’horloge peuvent être matérialisées par des états hauts avec un état de repos (Idle) bas (CPOL, Clock **P**OLarity = 0), ou être matérialisées par des états bas avec un état de repos haut (CPOL = 1).
- l’instant où doit s’opérer l’analyse de la ligne SDA peut se faire au début de l’impulsion d’horloge (CPHA, Clock **P**Hase = 0) ou à la fin de l’impulsion d’horloge (CPHA = 1). Pour résumer, il existe 4 Modes possibles pour une liaison SPI :

- Mode 0 : CPOL = 0 et CPHA = 0
- Mode 1 : CPOL = 0 et CPHA = 1
- Mode 2 : CPOL = 1 et CPHA = 0
- Mode 3 : CPOL = 1 et CPHA = 1

[...]

La figure 6.7.4.a montre les quatre modes utilisables pour la transmission d’un octet de valeur \$47 (valeur binaire 01000111).

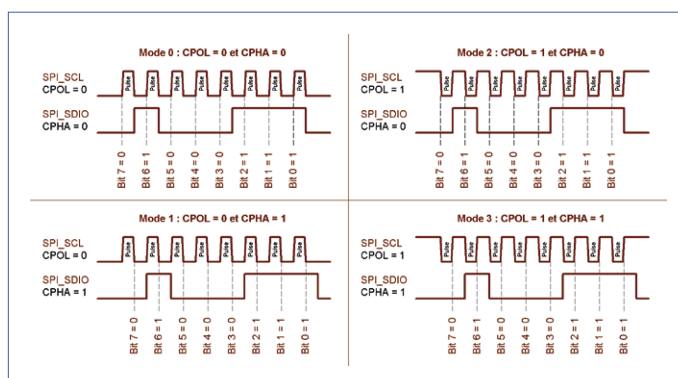


Figure 6.7.4.a – Octet de valeur \$47 (valeur binaire 01000111) transmis selon les 4 Modes possibles d’une liaison SPI. Le mode adopté conditionne la polarité (CPOL) et la phase (CPHA) du signal d’horloge SCL. Quand CPHA = 0, l’état de la ligne de donnée est analysé au début de l’impulsion d’horloge (*leading*). Quand CPHA = 1, l’état de la ligne de donnée est analysé à la fin de l’impulsion d’horloge (*trailing*).

Dans l’exemple donné avec transmission de l’octet de valeur \$47 (figure 6.7.4.a) c’est le bit de poids fort (MSB, Bit 7) qui est transmis en premier. Si le bit de poids faible (LSB, Bit 0) devait être transmis en premier, cela ajouterait 4 nouveaux cas de figure. Qui a dit que la configuration d’un bus SPI n’était pas amusante ?

200098-01

Rémy Mallard explique

les oscilloscopes anciens et modernes pour les débutants

376 pages (en couleur) | 17 x 23,5 cm | ISBN 978- 2-86661-208-5

Table des matières

1 – Introduction

- 1.1 – Qu'est-ce qu'un oscilloscope ?
- 1.2 – Fonctionnement de base d'un oscilloscope analogique
- 1.3 – Caractéristiques de base d'un oscilloscope
 - Sensibilité et échelle verticale
 - Base de temps, balayage et échelle horizontale
 - Bande passante
 - Caractéristiques spécifiques aux oscilloscopes numériques

2 – Sondes d'oscilloscope

- 2.1 – Qu'est-ce qu'une sonde ?
- 2.2 – Sécurité de l'utilisateur
- 2.3 – Sonde de tension passive
 - impédance et capacité
 - limite en fréquence | en tension
 - Influence du fil de masse de la sonde
 - quand utiliser une sonde passive ?
- 2.4 – Sonde de tension active
- 2.5 – Sonde de tension différentielle
 - limite en tension | en fréquence
 - isolation galvanique complète
- 2.6 – Autres types de sondes
 - haute tension | courant
 - champ proche | contact
- 2.7 – Types de connecteur
- 2.8 – Choix d'une sonde

3 – Choix d'un oscilloscope

- 3.1 – Pas trop vite, papillon...
- 3.2 – Analogique ou numérique ?
- 3.3 – Oscilloscopes de laboratoire, portatifs...
 - labo | portatifs ou « de terrain » (avec écran)
 - sur PC | autres types
 - DSO, MSO, DPO, MDO, DSO (bis)...
- 3.4 – Évaluation des besoins de l'utilisateur
 - lieu d'utilisation | nombre de signaux
 - tension et fréquence max. | nature des signaux | domaine fréquentiel
- 3.5 – Évaluation des performances d'un oscilloscope
 - performances générales | résolution et type d'écran
 - ergonomie | autres « petits détails »
- 3.6 – Mise à jour du matériel ou du logiciel (oscilloscope numérique)
 - mises à jour officielles et non officielles
 - maintenance et vérification métrologique (recalibrage)
- 3.7 – Oscilloscope d'occasion : une bonne affaire ?

4 – Configuration de l'oscilloscope

- 4.1 – Installation et position d'utilisation
- 4.2 – Prise en main et visualisation d'une trace

4.3 – Calibrage d'une sonde | LF | HF

4.4 – Modes de couplage

DC (continu) | AC (alternatif) | autres

4.5 – Modes de déclenchement

NORM, NORMAL | automatique (AT, AUTO, AUTO SETUP)
mono (MONO-COUP, ONE-SHOT, SINGLE)
source et pente de déclenchement | Autres types
HOLD-OFF (inhibition) | Mode Roll

4.6 – Réglages essentiels

Sensibilité | base de temps

4.7 – Visualisation simultanée de plusieurs signaux

traces séparées | traces combinées | calculées
évaluées

4.8 – Utilisation des curseurs (marqueurs) de mesure

4.9 – Fonctions « spéciales »

impédance | tension de décalage | voies isolées

5 – Mesures élémentaires

5.1 – Mesure d'une tension

continue | alternative | décalage (offset) | faible

5.2 – Mesures temporelles

intervalle | période | fréquence | rapport
cyclique
déphasage | montée | descente

6 – Mesures et fonctions évoluées

6.1 – Avertissements

6.2 – Mesure de la tension du secteur

sonde différentielle | transfo d'isolement | optocoupleur

6.3 – Mesures dans une alimentation secteur

linéaire | découpage | sans transformateur

6.4 – Mesure de bruit

6.5 – Mesure en XY et figures de Lissajous

déphasage | caractéristiques d'un dipôle

6.6 – Valeurs calculées et traces calculées

Valeurs calculées | Courbes calculées

6.7 – Analyse de protocole et décodage de données série

(I2C, SPI, UART, etc.)

validité électrique des signaux

données série (généralités) | I2C | SPI |

UART/RS232 | bus différentiel

6.8 – Analyse spectrale

6.9 – Mesures et fonctions spécifiques

puissance | impédance d'un dipôle
bande passante (réponse amplitude-fréquence)
jitter (gigue) | diagramme de l'œil
totaliseur/compteur d'événements
générateur de fonction intégré
Interface de communication
écran tactile | commande vocale

enregistrement des données
vérification d'une télécommande à infrarouge
signaux inconnus...

7 – Exercices (et solutions)

8 – Les générateurs de signaux

8.1 – Caractéristiques principales

technologie employée | type de signal
fréquence | amplitude du signal
types de sortie et protections
caractéristiques propres aux générateurs
numériques
modulations (AM, FM, sweep)
choix d'un générateur de signaux

8.2 – Utilité (et utilisation) d'un générateur de signal

mesure de bande passante | distorsion
stimulus avec signaux non permanents
comportement d'un étage logique ou analogique
vérification des protections d'un circuit
simulation de capteurs analogiques

Annexe 1 – Sécurité électrique

Annexe 2 – Numérisation d'un signal

Pourquoi numériser un signal analogique ?
Conversion analogique numérique (numérisation)
Fréquence d'échantillonnage (en hertz)
Quantification (en bits) et résolution
Conversion numérique analogique

Annexe 3 – Glossaire

Annexe 4 – Transformer...

un téléviseur en oscilloscope
un oscilloscope en téléviseur...
un oscilloscope en phasemètre audio
un oscilloscope en analyseur de spectre

Annexe 5 – Photographie oscilloscope analogique

Annexe 6 – Fabriquer soi-même une sonde ?

passive | active | différentielle

Annexe 7 – Petits trucs cachés

bande passante | Résolution verticale
résolution supérieure à 8 bits avec CAN 8 bits ?
taux de rafraîchissement (update rate)
tension d'entrée maximale
autres...
limitation de bande passante automatique
limitation de bande passante par logiciel

Annexe 8 – Liens utiles

Index

Liens et littérature

- [1] Table des matières complète (PDF) : www.elektor.fr/amfile/file/download/file/2103/product/9442/
- [2] Le livre dans la boutique en ligne d'Elektor : www.elektor.fr/les-oscilloscopes-anciens-modernes-pour-les-debutants
- [3] Rémy Mallard explique l'électronique pour les débutants qui sèchent les cours mais ne craignent pas de se brûler les doigts : www.elektor.fr/l-electronique-pour-les-debutants
- [4] Rémy Mallard explique les microcontrôleurs PIC pour les débutants qui veulent programmer sans patauger : www.elektor.fr/les-microcontroleurs-pic-pour-les-debutants

fonction FFT des oscilloscopes

représentation du signal dans le domaine spectral avec les oscilloscopes à mémoire numérique

Alfred Rosenkränzer (Allemagne)

Après le multimètre et l'oscilloscope, l'analyseur de spectre (AS) occupe certainement la troisième place dans la liste des instruments de mesure désirés par l'électronicien. Quoique justifié, ce vœu se voit rarement réalisé ; un AS reste aujourd'hui encore assez cher, même si les prix évoluent sous l'impulsion de sociétés telles que Siglent, Rigol et Signalhound. Nous décrivons ici les fonctions spectrales dont sont équipés les oscilloscopes à mémoire numérique (DSO) modernes, de sorte que vous pourrez décider par vous-même si un tel succédané du « véritable » AS pourrait vous donner satisfaction.

Les DSO (*Digital Storage Oscilloscope*) même bon marché disposent désormais de la possibilité de représenter un signal dans le domaine fréquentiel en plus du domaine temporel. Ce n'est pas étonnant, car un DSO dispose, grâce à sa mémoire et ses capacités de calcul, des outils essentiels pour décomposer un signal par FFT (transformation de Fourier rapide). Certes cette fonction affichera le spectre en fréquences comme le ferait un AS, mais elle ne pourra pas vraiment tenir tête à un véritable AS. Il est nécessaire de bien comprendre les fonctions d'un analyseur de spectre et celles d'un oscilloscope à mémoire numérique afin de pouvoir déterminer si la fonction FFT de ce dernier peut suffire à vos besoins.

Analyseur de spectre

Il existe plusieurs principes pour évaluer le spectre en fréquences d'un signal. La méthode la plus simple, mais qui n'est

plus utilisée aujourd'hui, consiste à balayer le signal à l'aide d'un filtre passe-bande dans le domaine de fréquences qui vous intéresse, et à évaluer l'amplitude du signal en sortie du filtre. Une autre méthode sera privilégiée, car il est difficile de réaliser un tel filtre ajustable avec une bonne dynamique et une largeur de bande suffisamment étroite en regard de la bande de fréquences scrutée, généralement beaucoup plus large. Comme le montre la **figure 1**, après atténuation ou amplification (non représentée ici), le signal d'entrée est mélangé au signal d'un oscillateur commandé. Cet oscillateur est piloté par un microcontrôleur ou équivalent, de sorte qu'on obtient en sortie du mélangeur une fréquence intermédiaire constante pour la bande de fréquences souhaitée par l'utilisateur. Le principe s'apparente au circuit de base d'un récepteur radio superhétérodyne [1]. Le signal FI est alors débarrassé de toutes composantes spectrales indésirables grâce à un filtre de fréquence intermédiaire, fixe et de grande qualité. Après amplification et redressement, l'amplitude du signal est mesurée, ce signal est ensuite utilisé conjointement avec le signal de commande de l'oscillateur pour attaquer l'affichage à deux dimensions. On peut aussi par ce principe balayer une bande de fréquences extrêmement large, tout en conservant une excellente dynamique. Comme on va le voir ci-après, un véritable AS fonctionne donc de manière très différente d'un DSO avec FFT.

DSO avec fonction FFT

N'importe quel oscilloscope numérique amplifie son signal d'entrée puis effectue une conversion A/N ; les données résultantes sont mises en mémoire pour un traitement numérique ultérieur. Certes le microcontrôleur, SoC ou FPGA d'un tel appareil peut représenter ces données numériques sous forme d'une courbe d'amplitude en fonction du temps, mais il peut faire bien plus

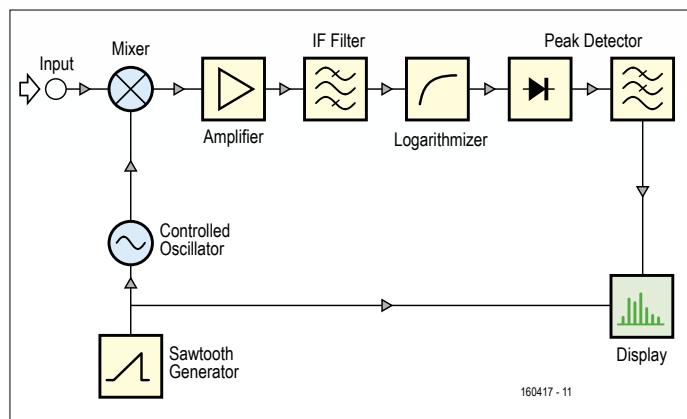
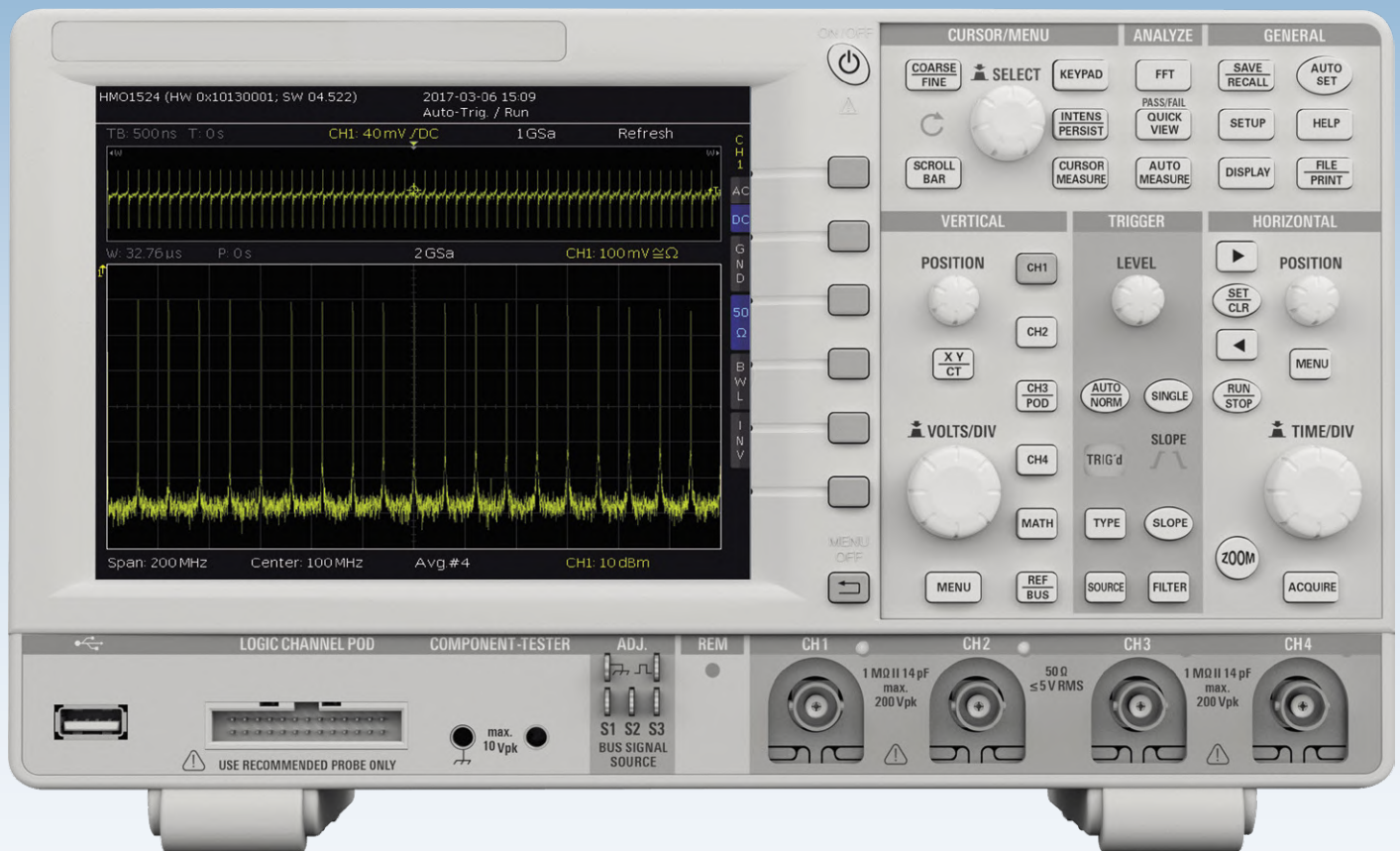


Figure 1. Synoptique simplifié d'un analyseur de spectre traditionnel.



avec ces données. Les DSO même bon marché peuvent généralement calculer et afficher les valeurs crêtes, valeurs moyennes ou efficaces, ainsi que les temps de montée, de descente ou autres durées de transitions, d'impulsions, etc. La fonction intégrée de fréquencemètre ne requiert par ailleurs que peu de ressources de calcul. L'affichage du signal dans le domaine fréquentiel est au moins tout aussi intéressant.

Le principe de base de l'affichage spectral par un DSO est illustré à la **figure 2**. Les données sont mémorisées après ajustage, amplification, filtrage passe-bas et conversion A/N. Les données FFT calculées sont affichées en amplitude par rapport à la fréquence. À cet effet, un nombre prédéfini d'échantillons du signal est d'abord écrit en mémoire à la fréquence d'échantillonnage du DSO, ou bien seule une certaine partie des données mémorisées est sélectionnée pour être analysée. Le nombre d'échantillons à analyser est généralement une puissance de deux pour un calcul de FFT le plus rapide possible. Les FFT peuvent travailler aussi avec un nombre différent d'une puissance de deux, mais elles requièrent alors une plus grande puissance de calcul, ou plus de temps.

À partir des 2^n échantillons, la FFT produit $2^{(n-1)}$ sections spectrales, ou canaux de fréquences. Par exemple $2^{16} = 65.536$ échantillons produiront 32.768 canaux de fréquences. La fréquence maximale obtenue correspond à la moitié de la fréquence d'échantillonnage. Avec un oscilloscope de relativement bonne qualité affichant une fréquence d'échantillonnage de 2 Géc/s, le spectre résultant atteint 1 GHz (voir aussi le théorème de Nyquist). La largeur de spectre d'un canal se calcule avec $1 \text{ GHz} / 32.768 = 30,5 \text{ kHz}$. Le canal le plus inférieur inclut la composante continue. Tout cela est bien beau, mais venons-en à présent aux limitations.

Tout d'abord, la largeur de spectre affichable est limitée non seulement par la fréquence d'échantillonnage, mais aussi par

la bande passante inférieure de la partie analogique du DSO. Bien plus décisif : un tel spectre est parfait seulement si les fréquences respectives du signal et d'échantillonnage sont cohérentes (ont un rapport stable). La mémoire doit contenir un nombre entier de périodes du signal. De plus, la fin de la forme d'onde doit correspondre parfaitement à son début.

► Spectres par FFT d'un DSO

Généralement ce n'est pas le cas. On utilise alors des « fenêtres » pour résoudre ce problème. À cet effet, on insère artificiellement le signal dans un genre d'enveloppe atténuant le début et la fin du signal selon un modèle défini, de sorte que l'on n'observe plus de (grands) sauts d'amplitude grâce à ces atténuations. Il existe différents types de fenêtres ou d'enveloppes ayant chacune une incidence différente sur le spectre résultant.

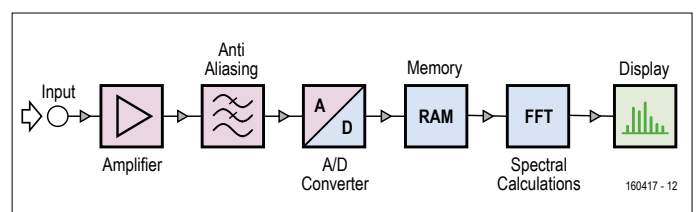


Figure 2. Schéma de principe de la fonction FFT d'un oscilloscope numérique.

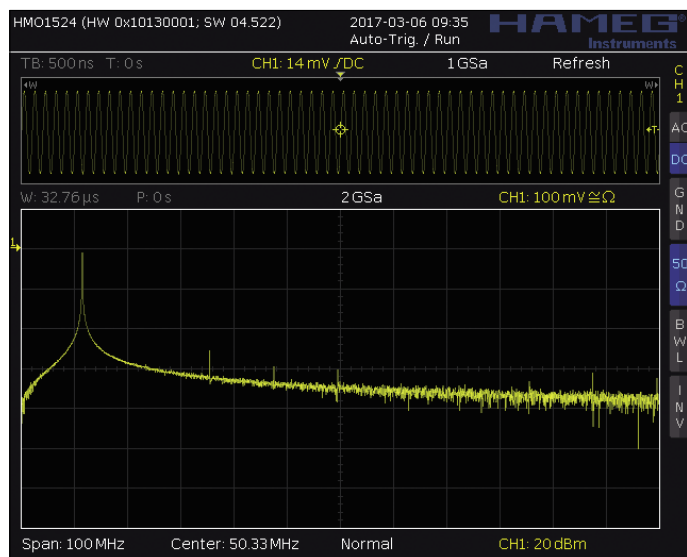


Figure 3. Spectre avec fenêtre rectangulaire. On notera les « jupes » larges (en anglais « skirt »). Ce spectre est inexploitable en tant que critère d'évaluation.

Vous trouverez sur l'internet des informations détaillées sur le « fenêtrage ».

La FFT en pratique

Avertissement : toutes les captures d'écran de DSO suivantes ont été faites avec un oscilloscope Hameg de type HMO1524. L'un ou l'autre détail pourra être différent avec un autre appareil. La partie supérieure de la **figure 3** montre la courbe d'amplitude en fonction du temps (affichage classique d'un oscilloscope) d'un signal sinusoïdal ; la partie inférieure, son spectre. Ce type d'affichage simultané d'un signal en fonction du temps et de la fréquence est un véritable avantage d'un DSO par rapport à un AS. Alors que la figure 3 montre un spectre plutôt inexploitable en raison de l'utilisation d'une fenêtre rectangulaire, la **figure 4** illustre l'effet de la fenêtre de type Hanning. La **figure 5** montre le spectre d'un signal carré avec une fenêtre de Hanning.

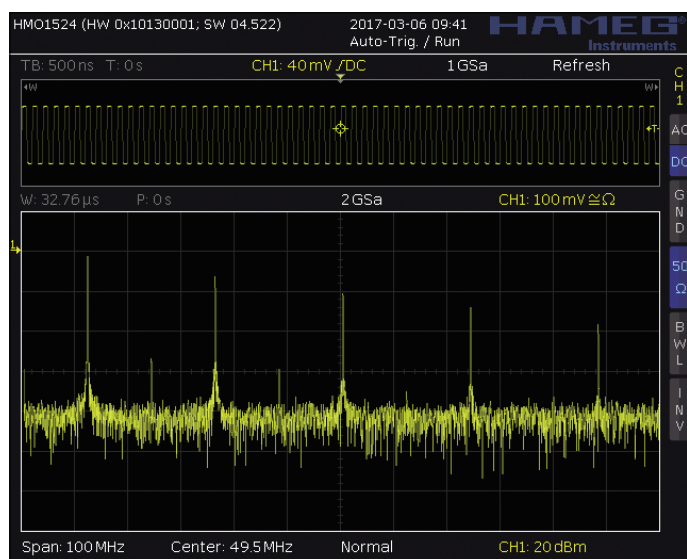


Figure 5. Spectre d'un signal carré avec son contenu harmonique bien connu.

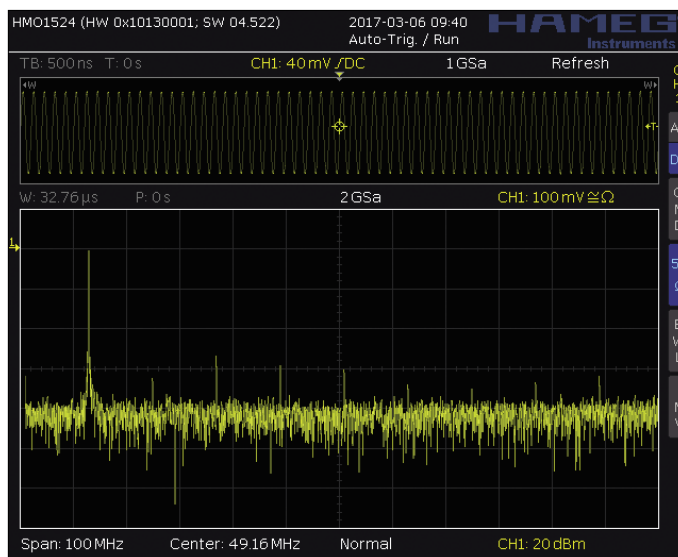


Figure 4. Ce même signal sinusoïdal avec une fenêtre de Hanning.

Les deux figures suivantes permettent d'effectuer une comparaison pour un signal composé de vingt fréquences de 10 à 200 MHz au pas de 10 MHz. La **figure 6** est la réponse d'un véritable AS d'Agilent. La largeur des composantes spectrales du signal à multiples fréquences est définie par « resolution bandwidth », qui à son tour agit sur le temps de balayage (pour mettre en évidence une zone de fréquences intéressante).

La **figure 7** montre ce même signal avec le DSO de Hameg. Les fréquences élevées sont certes atténuées (à tort), mais pour un oscilloscope à 150 MHz le spectre s'étendant jusqu'à 200 MHz est plutôt bien représenté. La largeur des canaux de fréquences est déterminée par le rapport fréquence d'échantillonnage / nombre d'échantillons. Pour les deux appareils, il est possible de réduire le niveau de bruit en calculant la moyenne de mesures effectuées plusieurs fois.

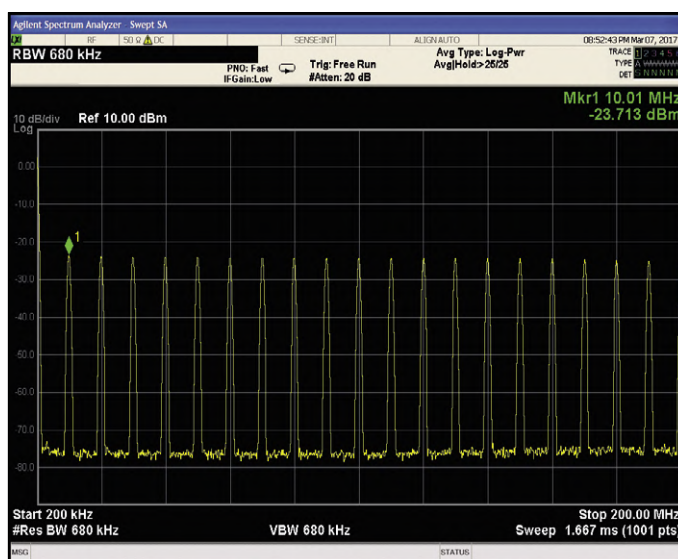


Figure 6. Spectre d'un signal à multiples fréquences, mesuré avec l'analyseur de spectre Agilent N9030 SA.

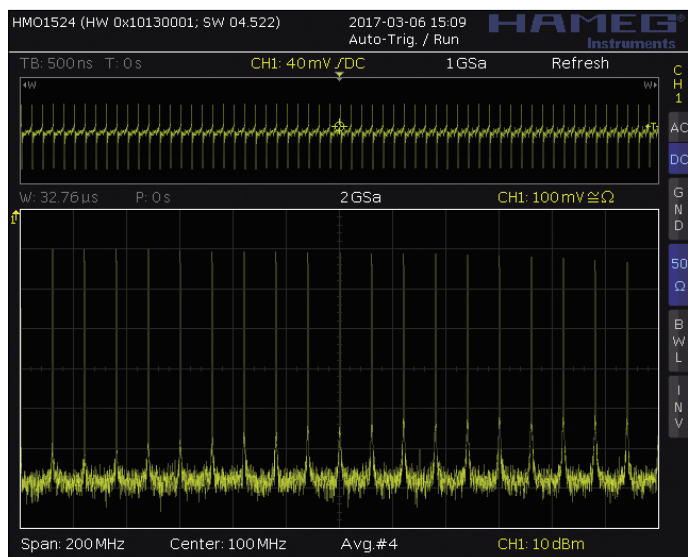


Figure 7. Spectre de ce même signal, mesuré avec le DSO HMO1524.

Selon le théorème d'échantillonnage, le signal d'entrée ne devrait comporter que des fréquences inférieures à la moitié de la fréquence d'échantillonnage. Cela nous donne 1 GHz pour le DSO à 2 GHz de Hameg. Avec un oscilloscope qui présente une bande passante nominale de 150 MHz, cela ne devrait pas poser de problèmes, car cette bande passante agit quasiment comme un filtre anti-repliement en atténuant plus ou moins fortement les composantes du signal au-delà de 150 MHz. Mais la prudence est de rigueur si l'on s'aventure à réduire la fréquence d'échantillonnage de sorte que des fréquences au-delà de la fréquence de Nyquist parviennent au convertisseur A/N. Ces fréquences apparaissent alors par miroir en deçà de la fréquence de Nyquist, et simulent ainsi des composantes de fréquences qui ne sont pas présentes en réalité dans le signal d'entrée (voir **fig. 8**). Une solution est un filtre approprié d'anti-repliement à l'entrée de l'oscilloscope. Néanmoins, un tel filtre passe-bas réglable est en général très peu présent sur les DSO. Mais la connexion en amont d'un filtre supplémentaire empêche une mesure en haute impédance à l'aide d'une sonde, ce qui constitue bien un avantage de plus du DSO sur l'AS.

En pratique : en mode FFT, le bouton « Time/Div » permet le réglage du « span » (gamme de fréquences). La valeur minimale est donnée approximativement par le produit du nombre de canaux affichables par la largeur des canaux. Le bouton « Position » décale le spectre (définit le milieu ou le « Center » de la gamme de fréquences). « dB/Div » se règle à l'aide de « Volts/Div » et le niveau de référence avec le bouton « Position ». Avant de procéder à des mesures sérieuses, il n'est certainement pas inutile de procéder à diverses expériences et de « jouer » avec un signal connu et différents réglages possibles. Ce n'est qu'ainsi que l'on pourra se forger une impression pratique des différentes options et de leurs effets.

Finalement, la **figure 9** représente un extrait du spectre du signal fourni par Unitymedia/KabelBW (opérateur de téléphonie/télévision/internet). On peut voir à gauche un canal de télévision numérique, et à droite un signal vidéo analogique avec sa porteuse couleur et les deux porteuses audio.

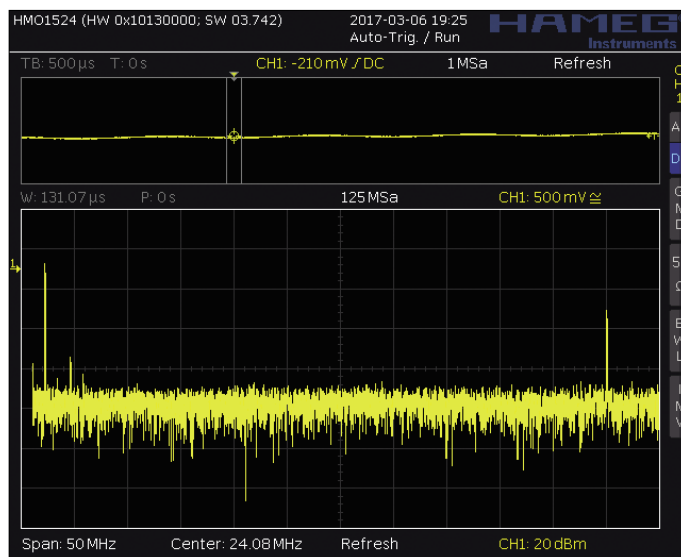


Figure 8. Spectre d'un signal composite à composantes réelles de 1 MHz et 80 MHz avec une fréquence d'échantillonnage de 125 Méc/s. Le signal de 80 MHz apparaît en miroir de la fréquence de Nyquist.

Conclusion

La fonction FFT d'un DSO ne remplace pas complètement un « véritable » AS, car non seulement la bande passante, mais aussi la dynamique restreinte due à une conversion numérique typiquement sur 8 bits limitent la résolution de l'affichage. En revanche, par une utilisation adéquate, il est tout à fait possible d'extraire des informations exploitables du spectre d'un signal. ◀

(160427 – version française : Xavier Pfaff)

Lien

[1] https://fr.wikipedia.org/wiki/Récepteur_superhétérodyne

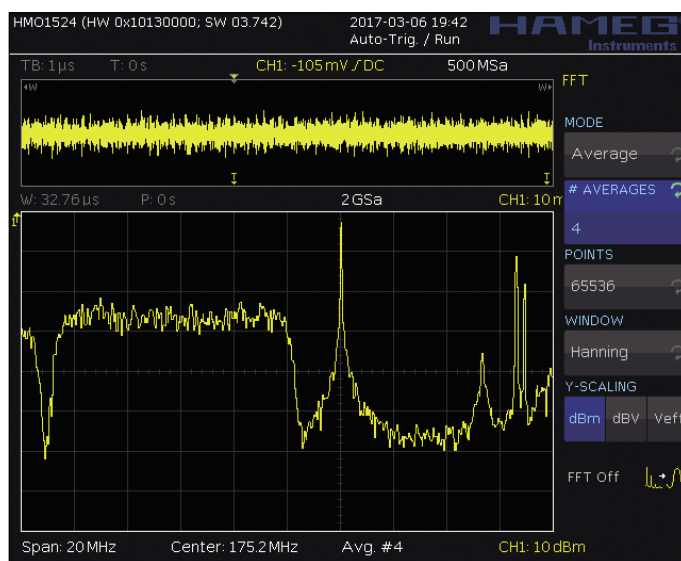


Figure 9. Extrait du spectre du signal de ma connexion au câble.

dis-moi comment tu ranges ton labo...

Visite guidée dans le saint des saints

Eric Bogers (Elektor Pays-Bas)

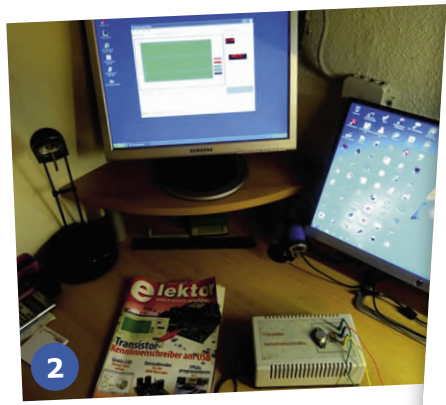
... et je te dirai qui tu es ! En juillet 2019, Elektor Labs organisait sur son site international un concours de photos de labos, afin de mieux connaître le biotope de ses lecteurs. Voici l'atelier d'électronique de Rainer Schuster, auteur de quelques articles publiés dans Elektor et sur le site Elektor Labs.

L'impression que laissent les images du labo de Rainer est celle d'une organisation rigoureuse, comme on n'en voit pas souvent dans l'électronique... Le maître des lieux écrit : « Mon labo est presque aussi vieux que l'édition allemande d'Elektor, du moins pour ce qui concerne l'équipement. En effet, tout a commencé en 1972 avec un multimètre et une alim. Au fil des ans sont arrivés les autres appareils (**fig. 1**). La plupart des appareils de test étaient des kits des sociétés RIM Electronic et Heathkit, qui, jusque dans les années 80, fournissaient des kits d'excellente qualité et à juste titre célèbres.

Sur la paillasse, on aperçoit le projet sur lequel Rainer travaillait lorsque la photo a été prise : un amplificateur de guitare pour son fils, basé sur l'application *Guitarix* qui fonctionne sur Raspberry Pi 4.

Le laboratoire comporte aussi des appareils de fabrication artisanale, comme le traceur de courbes de transistors du numéro de février 2009 d'Elektor (**fig. 2**) [1].

La troisième photo montre deux de ses propres projets : *HV Power Supply / Tube Curve Tracer* [2] (en haut) et *Electronic Load for DC and AC* [3] (en bas), tous deux publiés sur le site Elektor Labs.



Vous pouvez visiter le laboratoire de Rainer grâce à une vidéo [4] réalisée à l'occasion des *Olympiades vidéo d'Elektor* en 2017, où

il a obtenu une honorable troisième place grâce à sa description de quelques-uns des célèbres gags publiés jadis par Elektor dans ses numéros doubles d'été. ◀

200086-02

Liens

- [1] Traceur de courbes caractéristiques U/I pour transistors bipolaires et à effet de champ, sur port USB : www.elektormagazine.fr/magazine/elektor-200902/11207/
- [2] Alimentation électrique HV / Traceur de courbes tubulaires : www.elektormagazine.com/labs/hv-power-supply-with-tube-and-semiconductor-curve-tracer
- [3] Electronic load for DC and AC : www.elektormagazine.com/labs/electronic-load-for-dc-and-ac
- [4] Vidéo: <https://youtu.be/2f75IK12Wjo>



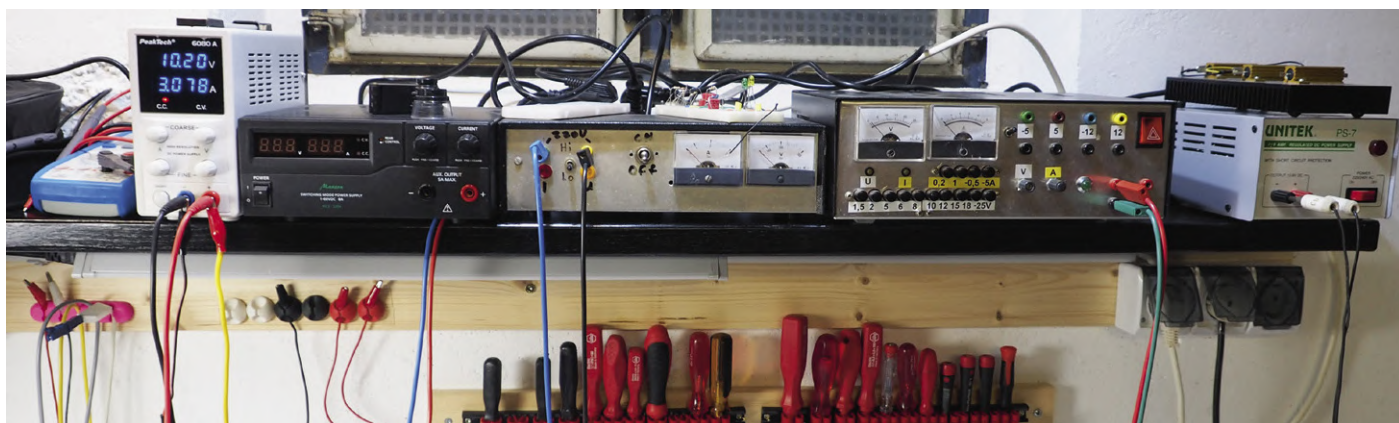
BANC D'ESSAI

alim de labo PeakTech 6080 A

Une alimentation nourissante à tout petit budget

Thomas Scherer (Allemagne)

Un électronicien n'a jamais « trop » d'alimentations. C'est vrai pour moi (voyez la photo de mon labo) et pour vous sans doute aussi. Plus surprenante pour l'époque actuelle est la constatation que la petite alim de labo PeakTech 6080 A est analogique. Pour un tout petit budget, elle offre robustesse et précision.



En matière d'alimentation, mon petit labo d'électronique ne manque de rien. Avec sa capacité de charge de 9 A, le bloc sous 12 V (à droite sur la photo) alimente mes petites perceuses/meuleuses. À côté trône le bloc d'alimentation de précision Elektor [1] de 1980 ! Après 40 ans, sa précision est toujours de 0,1 %. Il est complété par une source de tension secteur à haute impédance avec une isolation de sécurité et une capacité de charge d'environ 60 W. Ce n'est peut-être pas très joli, mais c'est pratique et ... ça a déjà sauvé des vies.



Figure 1. Le coffret de l'alim de labo est bien emballé dans une mousse épaisse.

Le bloc d'alim de labo numérique de 60 V et 8 A est puissant, sa seule faiblesse serait que je ne l'ai pas construit moi-même. Vous connaissez à présent le contexte dans lequel j'ai testé l'alim PeakTech 6080 A, qui ne fournit peut-être que 15 V et 3 A, mais a d'autres caractéristiques intéressantes.

Tombé du camion

Le colis de l'alim PT 6080 A n'est pas arrivé chez moi par la poste, il a été annoncé par téléphone. C'est un voisin qui m'a appelé. Ses parents avaient vu des paquets tomber de la camionnette du livreur... Si je vous raconte ça, c'est parce que si l'alimentation n'avait pas été très bien emballée dans sa boîte en carton avec de la mousse épaisse (**fig. 1**), elle n'aurait pas survécu aux aléas du voyage et je n'aurais pas pu boucler ce banc d'essai à temps.

Avec ses 80 x 160 x 225 mm (L x H x P), le coffret est vertical, et pas horizontal comme d'habitude (**fig. 2**), ce qui en réduit l'encombrement sur le plan de travail. Le radiateur maous, à l'arrière, est frappant. C'est à cela qu'on reconnaît une alimentation à régulation analogique, qui doit dissiper pas mal de puissance surtout quand la tension de sortie est dans le bas de la plage de régulation. Son poids est un autre signe : environ 2 kg, ce n'est pas léger pour 45 W. Sans doute la ferraille et le cuivre d'un vrai transfo. Et pas un petit transfo HF, comme sur les alimentations à découpage.

Sa tension de sortie sur deux prises bananes de sécurité est



Figure 2. L'alim de labo PT 6080 A vue de devant et de derrière (Photo : PeakTech)

réglable de 0 à 15 V. La limitation de courant est réglable de 0 à 3 A. Quatre boutons rotatifs sont prévus à cet effet — deux de chaque côté sont responsables du réglage grossier et fin de la tension (à droite) et du courant (à gauche). Il y a également deux afficheurs à quatre chiffres pour la tension et le courant, et deux LED marquées «C.C.» et «C.V.». Sur la figure 2, la LED C.V. est allumée. L'alim est donc en mode de *tension constante*. Quand c'est le courant qui est limité, la LED CC (*Constant Current*) s'allume.

Précision surprenante

Vérifier la précision de cet affichage. Sur mon alim numérique, tension et courant sont régulés numériquement à l'aide de deux codeurs. Il est généralement possible de passer d'un réglage grossier à un réglage fin dans un rapport de 1:10 en appuyant sur le bouton. Malheureusement, la valeur affichée par les alimentations numériques est la valeur de consigne, pas ce qui sort des prises. On se retrouve donc avec généralement de petits écarts de l'ordre de 100 mV ou de 10 mA. C'est parce que les fabricants d'alimentations numériques, surtout les modèles bon marché, font l'économie d'une puce de voltmètre numérique et ne donnent donc de la tension de sortie réelle qu'une valeur indicative.

Ce n'est pas le cas sur l'alimentation PT 6080 A. Au lieu de codeurs rotatifs, on emploie ici des potentiomètres. Le rapport du réglage grossier au réglage fin est d'environ 1:7. Ce réglage analogique m'a demandé un temps d'adaptation. Contrairement aux alimentations numériques, vous ne voyez pas immédiatement ce que vous réglez. Comme la tension appliquée et le

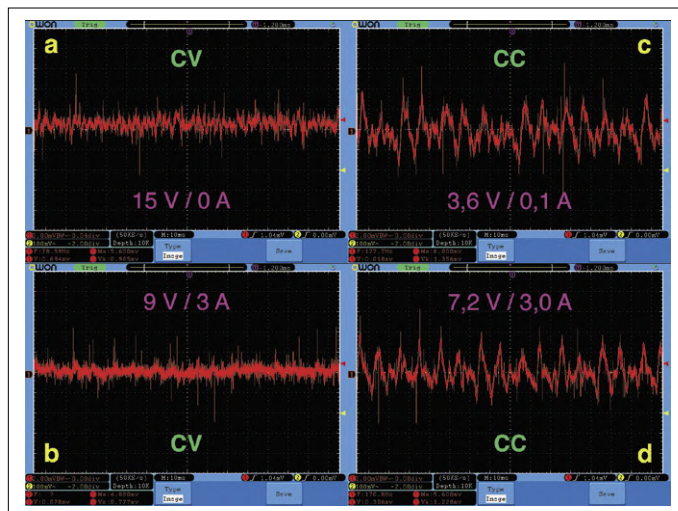


Figure 5. Diverses traces de la tension de sortie à l'oscillo.

courant en circulation sont mesurés numériquement ici et que les puces de convertisseur A/N typiques effectuent environ trois à quatre mesures par seconde, il y a forcément un (petit) décalage entre le mouvement du potentiomètre et l'affichage des valeurs correctes. Ce n'est rien de grave, mais c'est assez différent des alimentations numériques de labo pour ne pas passer tout à fait inaperçu.

La **fig. 3** confirme que ce n'est rien de grave. Ce qu'affichent l'alim de labo PT 6080 A et mon multimètre est identique à 10 mV = 0,07 % près pour la tension, et 0,3 mA = 0,08 % pour le courant (avec un courant de 3 A, l'écart est similaire). C'est étonnant, d'autant plus qu'avant d'effectuer cette mesure comparative, j'avais étalonné mon multimètre avec une source de tension de référence précise à 0,1 %. L'alim PT 6080 A est donc beaucoup plus précise que la spécification du fabricant de 0,5 % ±5 chiffres et vous pouvez vous fier à ce qu'elle affiche. Je ne m'attendais pas à ça !

La **fig. 4** montre la tension maximale en mode CV (a) et le courant maximal en mode CC (b). Le PT 6080 A délivre donc un peu plus que les 15 V et 3 A spécifiés.

Niveau de bruit

À quoi ressemble la tension de sortie en charge ? Si on achète une alim de labo analogique de puissance modérée, ce n'est pas pour alimenter principalement des circuits numériques. Son champ d'utilisation, ce sera plutôt de faire fonctionner des circuits analogiques sensibles aux interférences, amplificateurs BF ou circuits RF avec le niveau de bruit le plus faible possible sur la tension d'alimentation. À l'exception des



Figure 3. Comparaison de l'affichage du multimètre et de l'affichage de l'alimentation de laboratoire (à gauche : tension, à droite : courant).

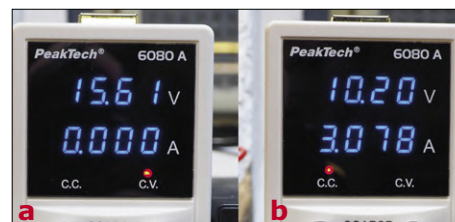


Figure 4. Tension maximale (a, CV) et courant maximal (b, CC).

appareils de labo très coûteux, les sorties des alimentations de labo numériques sont toujours (plus ou moins) encombrées par des parasites résultant des fréquences de commutation numérique et logiques, plus les harmoniques. Une alimentation analogique à régulation linéaire devrait faire mieux. Est-ce aussi le cas du PT 6080 A ?

Le fabricant indique que l'ondulation résiduelle de la tension de sortie est de $0,5 \text{ mV}_{\text{eff}}$. La **fig 5** montre quatre captures d'écran de la tension de sortie de l'oscillo dans différentes conditions. La sensibilité (2 mV/div.) et la base de temps (10 ms/div.) sont les mêmes partout. Le bruit et le ronflement en mode CV sont bien dans la plage spécifiée, à la fois à vide (a) et en charge (b). La tension de sortie ne se comporte pas aussi bien lorsque l'alim est en mode CC. Ici, l'ondulation est d'environ $2 \text{ mV}_{\text{rms}}$, indépendamment de la charge. La composante basse fréquence clairement visible avec 128 Hz (c) ou 171 Hz (d) n'est pas un ronflement du secteur, sinon une fréquence de 50 ou 100 Hz serait affichée. Cependant, ce niveau de bruit est encore bon, et comme les circuits analogiques sont fournis en mode CV, le niveau de bruit légèrement augmenté en mode CC est négligeable.

Vie intérieure

Un banc d'essai n'est pas une autopsie, mais quand j'ai sous la main un nouvel appareil comme celui-ci, je ne résiste pas longtemps à la tentation de dégainer mon tournevis. C'est vite fait. Six vis, le couvercle en tôle et voilà le travail (**fig 6**). Le transfo torique est massif. Son primaire est commutable pour les réseaux 230 et 115 V . Un enroulement de 30 V avec prise centrale alimente le tableau de commande/affichage à l'avant (c). À l'arrière se trouvent le circuit imprimé avec redresseur à quatre diodes de 5 A et le condo de lissage, bien dimensionné avec $4.700 \mu\text{F}/50 \text{ V}$, ainsi que le transistor de puissance vissé sur le dos du coffret. Le composant noir à droite des diodes est un relais, dont on entend le déclic lorsqu'on balaie la plage autour de $7,5 \text{ V}$ avec le potentiomètre. En dessous de ce seuil, le relais n'applique au redresseur qu'un enroulement secondaire de $12,5 \text{ V}$ pour ne passer à une tension alternative de $21,5 \text{ V}$ qu'au-dessus du seuil de $7,5 \text{ V}$ de sortie. Cette précaution astucieuse réduit la dissipation de puissance en cas de forte consommation de courant quand la tension de sortie de l'alimentation basse ($< 7,5 \text{ V}$).

Bilan

Le PT 6080 A de précision coûte à peine 60 € dans l'e-choppe d'Elektor. Son rapport qualité/prix est donc remarquable. Cet appareil se distingue en effet par la précision des valeurs affichées et m'a donc paru convaincant, bien plus d'ailleurs que ce à quoi je m'attendais. Un rapport de calibrage imprimé est fourni avec l'appareil, et mes propres mesures en ont confirmé l'exactitude. L'électronique est bien dimensionnée et je n'ai constaté aucun ronflement. Les tensions sont parfaitement régulées et ne fléchissent que légèrement sous pleine



Figure 6. Intérieur de l'alim de labo PT 6080 A avec transfo de puissance, carte de puissance et section de commande avec CMS.

charge. Les niveaux de bruit en mode CV sont bons. Grâce à l'astuce de la commutation par relais, l'alimentation ne chauffe pas beaucoup même en cas de forte charge continue. L'alim PT 6080 A est évidemment à l'épreuve des courts-circuits. En somme, cette petite alimentation analogique de labo mérite donc indubitablement la mention *recommandée par Elektor*. ◀

200009-02



@ WWW.ELEKTOR.FR

→ Alimentation de laboratoire PeakTech 6080 A DC
www.elektor.fr/19111

Liens

- [1] Alimentation de précision Elektor, article publié en nov. 1980 à la page 22 : www.elektormagazine.fr/magazine/elektor-198011/51386/
- [2] Caractéristiques de l'alimentation PT 6080 A (en anglais) : www.peaktech.de/productdetail/kategorie/acdc-stabilisierte-labornetzgeraete/produkt/peaktech-6080-a.html



bienvenue dans votre e-choppe

la rédaction recommande



Écran tactile portable Joy-View 13

Essayez donc de trouver un petit moniteur de qualité facile à transporter. L'offre est limitée. Les choses s'arrangent avec le Joy-View 13. Grâce à ses ports HDMI et USB-C, il peut être connecté aux appareils modernes. J'ai testé cet écran de 33 cm de diagonale avec différents ordinateurs. Avec un ordinateur portable, un deuxième moniteur est parfois très utile. Le problème n'est pas tant le prix, ni la variété des modèles, mais la **portabilité**. Vous voulez un écran compact, tactile et, de préférence, pas alimenté par le secteur. Ce sont exactement les propriétés

du Joy-View. Fin, robuste, alimenté au choix par le secteur, par un ordinateur portable ou une batterie. Par l'USB-C, il peut être connecté aux ordinateurs portables et aux téléphones tactiles les plus récents.

Harry Baggen (labo d'Elektor)



www.elektor.fr/19162 www.elektor.fr/19163

Vos favoris :

1. Archives 1978-2019 d'Elektor sur clé USB
www.elektor.fr/19221



2. Explore the Raspberry Pi in 45 Electronics Projects (3rd Edition)
www.elektor.fr/19190
3. Raspberry Pi 4 B (2 GB)
www.elektor.fr/18965
4. Les oscilloscopes anciens & modernes pour les débutants
www.elektor.fr/19124
5. Microcontroller Basics with PIC
www.elektor.fr/19188
6. Raspberry Pi 4 B (4 GB)
www.elektor.fr/18964

Explore the Raspberry Pi in 45 Electronics Projects (3rd Edition)

Pack d'alimentation CC JOY-IT JT-RD6006

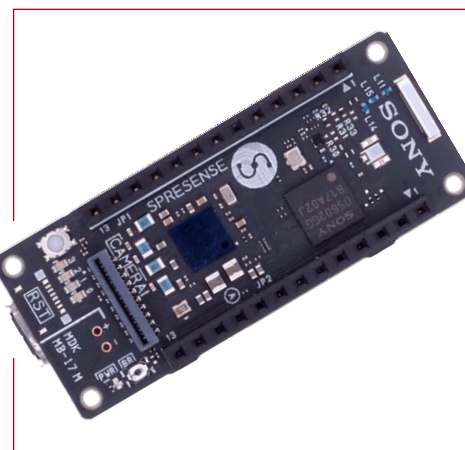
Carte Sony Spresense



La 3^e édition de ce livre en anglais, entièrement révisée, décrit 45 projets passionnants : gyrophare, commande de moteur électrique, traitement et génération de signaux analogiques, contrôle de la température et luxmètre. Des projets plus complexes sont également proposés : régulateur de vitesse de moteur, serveur web avec CGI (Common Gateway Interface), et des applications client-serveur.



Ce pack est une exclusivité d'Elektor. Il réunit l'alimentation programmable de labo JT-RD6006, un bloc d'alimentation industriel (60 V, 6 A), un grand et robuste coffret en aluminium et un module WiFi enfichable. Celui-ci permet à d'autres appareils (mobiles) de commander le JT-RD6006 sans fil. L'ensemble sera fourni dans un étui pratique.



Spresense est une carte de compacte basée sur le μ C multicœur CXD5602 de Sony, à faible consommation. Elle permet de créer des applications en très peu de temps et est prise en charge par l'EDI Arduino ainsi que par le SDK plus avancé basé sur NuttX. La carte principale utilise un processeur développé par Sony pour les applications IoT et de détection. La carte principale peut être utilisée seule ou avec extension.



Prix (membres) : 31,46 €

www.elektor.fr/19190



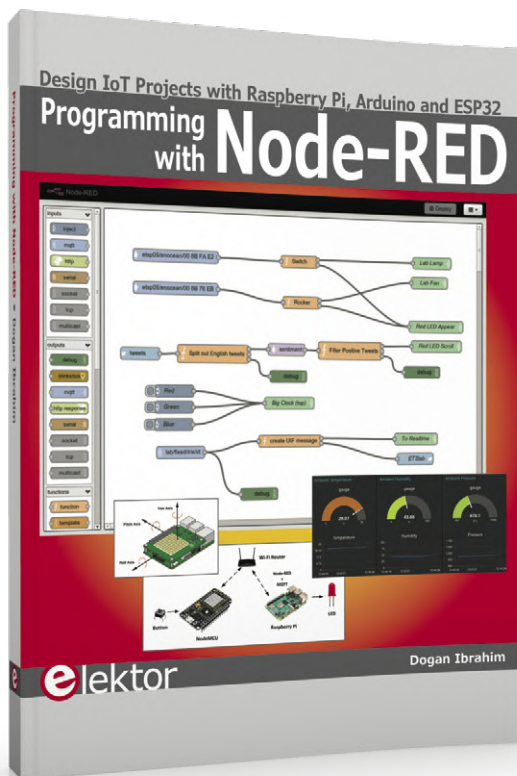
Prix (membres) : 175,46 €

www.elektor.fr/19211



Prix (membres) : 71,96 €

www.elektor.fr/19233



Programmer avec Node-RED

(en anglais)

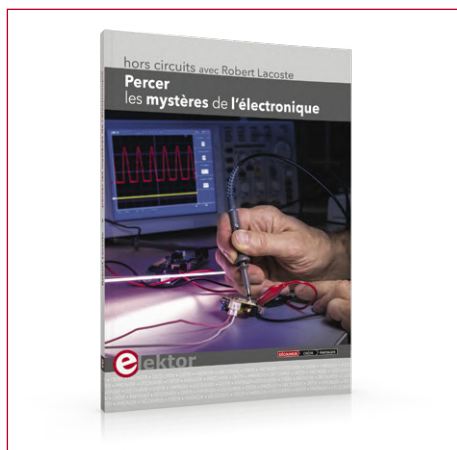
Node-RED est un éditeur visuel libre conçu par IBM pour l'Internet des objets. Il est livré avec un grand nombre de nœuds pour gérer une multitude de tâches. Les nœuds nécessaires sont sélectionnés et réunis pour effectuer une tâche particulière. Node-RED est basé sur la programmation de type flux où les nœuds sont configurés et joints pour former un programme d'application. Il existe des nœuds pour des tâches complexes, notamment l'accès au réseau internet, Twitter, le courrier électronique, HTTP, Bluetooth, MQTT, la commande des ports GPIO, etc. Avec Node-RED, le programmeur n'a pas besoin d'apprendre à écrire des programmes complexes.

Ce livre montre comment utiliser Node-RED dans des projets. La principale plate-forme matérielle des projets de ce livre est Raspberry Pi 4, mais il parle aussi de l'utilisation de Node-RED avec Arduino Uno, ESP32 DevKitC, et des cartes de développement de µC ESP8266 NodeMCU.



Prix (membres) : 33,75 €
www.elektor.fr/19190

Percer les mystères de l'électronique



Le monde de l'électronique est à la fois vaste et tout petit ! Tout touche à tout, le plus petit détail peut avoir les plus grandes conséquences. L'objectif de l'auteur, Robert Lacoste, n'est pas de vous tenir par la main. Il vous donne des pistes pour comprendre et vous permettre ensuite de progresser seul. Repoussez vos propres limites et apprenez à détecter celles du matériel et du logiciel que vous utilisez.



Prix (membres) : 33,75 €

www.elektor.fr/19080

Les oscilloscopes anciens & modernes pour les débutants



En électronique, si l'on veut progresser dans le plaisir et dans la compréhension, il faut un oscillo. Comment le choisir ? À peine cette question-là aura-t-elle trouvé sa réponse, il en viendra d'autres qui se résument ainsi : comment se servir de l'oscilloscope de telle sorte que ce qu'il affiche corresponde à la réalité des signaux ? Dans ce livre, Rémy Mallard, l'auteur de *l'électronique pour les débutants*, répond à ces questions-là et bien d'autres.



Prix (membres) : 33,75 €

www.elektor.fr/19124

Tactigon ONE – capture de gestes et de mouvements avec IA



Carte portable ultime, programmable avec l'IDE Arduino, avec GPIO, UART, système de recharge de batterie intégré, µC rapide et Bluetooth LE embarqué ! Mesurez les mouvements linéaires et angulaires avec un gyroscope à 3 axes et un accéléromètre à 3 axes. Un capteur magnétique supplémentaire à 3 axes offre une plus grande précision. L'unité de µC rapide et le SDK Arduino permettent l'exécution de croquis complexes.



Prix (membres) : 71,96 €

www.elektor.fr/19106



grand défi : internet des objets ou internet des déchets ?

Des produits sûrs pour l'IdO

Tessel Renzenbrink (Pays-Bas)

Ceintures de sécurité, coussins d'air, limitations de vitesse et lutte contre l'alcool au volant sont autant d'innovations et de réglementations qui visent à réduire le nombre et la gravité des accidents de voiture. Il en va de même pour d'autres produits de consommation que nous espérons acheter en toute confiance et en toute sécurité. L'internet des objets (IdO) ouvre une nouvelle boîte de Pandore. L'apparition de nombreux produits équipés d'électronique programmable et connectés à des réseaux nous expose à de nouveaux risques. Les exemples souvent inattendus ne manquent pas : les jouets en réseau, utilisables pour espionner les enfants, ou les véhicules dont les systèmes informatiques peuvent être piratés puis commandés à distance.

À la demande de la Commission européenne, le professeur Ross Anderson et ses collègues ont étudié, les mesures nécessaires pour combiner la sécurité traditionnelle des produits et l'avènement de l'IdO. L'UE prépare une nouvelle législation visant à garantir la sécurité des produits. D'où la requête faite à Anderson d'identifier ce qui est nécessaire pour cela. Celui-ci, professeur d'ingénierie de la sécurité à l'université de Cambridge, a réalisé l'étude avec ses collègues Eirann Leverett, chercheur sur les risques, et Richard Clayton, chercheur en sécurité. Leurs conclusions ont été publiées en 2017 [1].

Trilemme

D'après Anderson et ses collègues, il reste du chemin à faire ! Le nœud du problème est dans le hiatus entre sécurité de produits physiques et sécurité des techniques numériques. Les produits physiques sont testés et contrôlés avant d'être mis sur le marché, pour être validés par un certificat de sécurité ou un label d'homologation. En cas de modification (substantielle) du produit, une nouvelle certification est impérative. La sécurité des logiciels est tributaire du moindre changement ; leur surveillance est constante, ils sont surveillés et révisés sans cesse par des correctifs et des mises à jour. Anderson, qui a approfondi ce sujet lors du dernier congrès *Chaos Computer Club** en décembre 2019 [2], parle du *trilemme* (ou triple dilemme) des produits de l'IdO. Si vous vous en tenez à la certification avant commercialisation, vous ne pouvez pas modifier votre logiciel, ce qui signifie que votre produit n'est pas sûr. Si vous modifiez votre logiciel, vous perdez votre certification. Et si vous combinez certification et mise à jour, vous devez repasser par la case certification après chaque mise à jour de logiciel : vos coûts s'envolent !

Durée de vie et complexité

Produits traditionnels et services numériques ont des caractéristiques propres, désormais imbriquées. Selon Anderson, il faut éviter de se retrouver avec le pire des deux mondes. Prenez les téléphones et les voitures. La durée de la prise en charge

du micrologiciel et du système d'exploitation des téléphones portables est généralement de l'ordre de trois ans. Après la fin de la période d'assistance contractuelle, un téléphone devient potentiellement dangereux parce que les points de vulnérabilité du logiciel ne sont plus corrigés. Le matériel est encore en parfait état, mais le consommateur, même s'il en est satisfait, est poussé à remplacer son téléphone. Il faut éviter qu'une telle pratique fasse son apparition dans l'industrie automobile, déclare M. Anderson.

Une maintenance à vie apporte cependant d'autres défis particuliers. Supposons qu'un constructeur automobile s'engage à fournir le suivi pour un certain modèle pendant 20 ans au moins. L'ingénieur en logiciel qui écrit un programme pour un véhicule à commercialiser en 2022 devra tenir compte du fait que son code doit continuer à fonctionner jusqu'en 2042. La complexité constitue un autre défi. L'idée inhérente à l'IdO est de permettre à des appareils de communiquer entre eux et d'agir de manière autonome. Ce seront par exemple des véhicules autonomes qui échangent des informations avec l'infrastructure routière ou des réfrigérateurs qui font leurs propres courses. Les tests de sécurité de tels dispositifs pris individuellement ne suffisent plus. Il faut examiner aussi les risques découlant de leur interaction avec d'autres dispositifs.

Recommandations

La question de la sécurité des produits à l'ère de l'IdO est complexe. Aucune solution simple n'en viendra à bout. Dans leur rapport à la Commission européenne, les trois chercheurs formulent plusieurs recommandations dont la principale est la création d'une Agence européenne pour l'ingénierie de la sécurité et de la sûreté (ESSEA). Celle-ci doit disposer de l'expertise technique nécessaire pour élaborer des normes et informer des décideurs politiques. Elle doit abriter une base de données centrale sur les points de vulnérabilité des logiciels, les composants défaillants et les erreurs d'intégration du système. Pour l'instant, ce type d'informations, détenu par divers organismes et entreprises, est dispersé. La centralisa-



Photo : Gabor Kiss. Source : CC BY 2.0 www.flickr.com/photos/-nevi-/5404057758/

L'UE comme ligne de défense contre l'internet des déchets

Lors de sa présentation au CCC, M. Anderson a souligné la difficulté de faire appliquer les recommandations. De nombreuses parties sont impliquées, avec chacune ses intérêts propres. Ainsi *Facebook* et *Google* ont-ils fait pression contre la proposition de soumettre les services numériques aux réglementations existantes en matière de responsabilité. Toutefois, des avancées ont été obtenues depuis la publication du rapport. Par exemple, l'adoption du règlement n°2019/771 qui stipule que les consommateurs de produits comportant des éléments numériques bénéficient d'un droit de mise à jour pendant au moins deux ans, ou plus si l'on peut raisonnablement s'y attendre. Ce dernier point implique que des produits comme les machines à laver, dont la durée de vie

normale est plus longue, devraient également recevoir des mises à jour pendant une période plus longue.

L'énormité de la tâche décrite par Anderson, Leverett et Clayton pour garantir la sécurité des produits de l'IdO contraste fortement avec la lenteur du processus décisionnel dans l'Union européenne. Cependant, dans un article scientifique connexe, ils insistent sur le fait que leurs attentes s'adressent à l'UE :

« Puisque Washington s'en moque et qu'aucun autre acteur ne pèse assez lourd, l'UE est déjà le principal régulateur de la vie privée au monde ; son objectif devrait donc être de devenir également le principal régulateur de la sécurité — à défaut de quoi elle compromettrait la mission de sécurité qu'elle a déjà ». [3] ◀

200066-03

tion de ces informations permettrait aux intéressés de partager leur expérience et, ce faisant, d'éviter que les mêmes erreurs se reproduisent.

Une autre recommandation aux fabricants de produits dotés de capacités de réseau est de garantir la possibilité de mises à jour logicielles de ces produits.

En outre, l'UE devrait amender certaines directives ainsi que des règlements existants pour tenir compte des nouveaux développements. La directive sur la responsabilité du fabricant d'un produit, par exemple, devrait être étendue pour couvrir les services numériques. En vertu des dispositions de la directive actuelle, un fabricant d'appareils de navigation peut être tenu responsable des erreurs, mais pas les fabricants de *Google Maps*.

Liens et littérature

- [1] Eireann Leverett, Richard Clayton, Ross Anderson, Editor G.Baldini, *Standardisation and Certification of Safety, Security and Privacy in the 'Internet of Things'*, European Commission, Brussels, Belgium, 2017 : <https://op.europa.eu/en/publication-detail/-/publication/80bb1618-16bb-11e8-9253-01aa75ed71a1/language-en>
- [2] La durabilité de la sûreté, de la sécurité et de la vie privée [vidéo en anglais]. Conférence de Ros Anderson lors du Chaos Communication Congress à Leipzig, le 28 décembre 2019 : https://media.ccc.de/v/36c3-10924-the_sustainability_of_safety_security_and_privacy#t=1397
- [3] Eireann Leverett, Richard Clayton, Ross Anderson, *Standardisation and Certification of the 'Internet of Things'*, 2017 : <https://www.cl.cam.ac.uk/~rja14/Papers/weis2017.pdf>

* Le Chaos Computer Club, association européenne de hackers, fournit et diffuse des informations sur des questions techniques et sociétales (surveillance, vie privée, liberté d'information, activisme en micro-informatique, sécurité des données, etc.)
En dépit des apparences, le contenu des références [1] et [3] n'est pas le même.

fréquencemètre 1,2 GHz & générateur de signaux carrés d'Elektor (1992/93)

Il a franchi le mur du gigahertz

Jan Buiting (Elektor - Retronics)

Le projet intitulé *High Frequency Counter* (compteur HF) date de septembre 1992 et début 1993, réparti sur trois articles du magazine Elektor. Son caractère pro lui vient de l'attention portée à l'élaboration du boîtier et au style de sa face avant, à la place donnée au logiciel, à la facilité de tester l'appareil. Il a été disponible un temps sous forme de kit, accompagné d'un vrai livre comme manuel.



Pour ma rubrique *Rétronique*, je ne choisis pas les articles au hasard dans les vastes archives d'Elektor – elles remontent à 1974 pour le magazine en anglais. Comme d'autres instruments déjà décrits dans cette rubrique, cet appareil-ci a survécu à la

frénésie de destruction du matériel ancien, en particulier les prototypes du labo, qui avait frappé Elektor durant l'été 2006 : on avait alors fait venir une grande benne sous les fenêtres de notre labo pour y jeter « toutes ces vieilleries ». J'avais pu

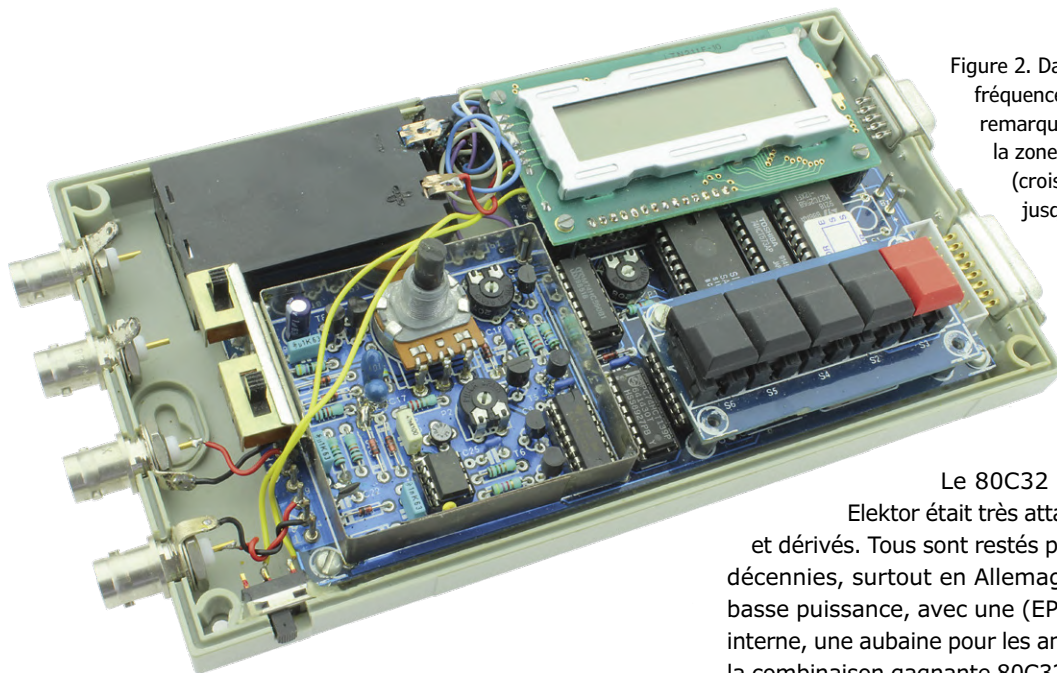


Figure 2. Dans le boîtier ouvert du fréquencesmètre multifonction de 1,2 GHz, remarquez le blindage métallique autour de la zone du prédiviseur où l'on peut trouver (croisons les doigts) des fréquences jusqu'à 1,2 GHz.

C'est ça une « vieillerie » ?

En regardant le schéma (fig. 1) – désolé, on n'a plus les originaux – pourrait-on imaginer en faire une version moderne du fréquencesmètre avec des composants de 2020 ? Essayons, juste pour voir :

- 80C32, EPROM, IC5, IC3 (section μC) → Arduino, ESP32
- K12, IC6, T2 et entourage (pseudo RS-232) → port USB
- 7805 (alim) → quelques régulateurs LDO récents
- alim 9 V → alim 6 V
- LCD → OLED
- BAT82 et HP28xx → diodes à ultra basse capacité

Spécifications principales

Modes de mesure :

- Fréquencesmètre : 1 mHz à 1,2 GHz
- Période : (1/fréquence) : 1 ps à 4 000 s
- Compteur de révolutions : 0,001 à 4×10^6 tours/min
- Compteur d'événements (impulsions) : 1 à environ 4×10^9
- Zéro du compteur : 1 à 4×10^9

Mode générateur :

- Temporisateur : 1 μs à 4 000 s
- Générateur d'impulsions : 8 500 s (0,117 mHz) à 4 μs (250 kHz)
- Rapport cyclique : 1:1 à $1:4 \times 10^9$
- Nombre de périodes : 1 à 4×10^9

Paramètres définis par l'utilisateur :

- Mesure unique/continue
- Durée de porte flottante
- Affichage des résultats intermédiaires des mesures
- Analyse de période (mesure précise en basse fréquence à de courtes périodes de porte)
- Niveau de repos
- Polarité des impulsions
- Réglage marche/arrêt du résonateur

Le 80C32 était une merveille à l'époque.

Elektor était très attaché aux séries 8051/8052 d'Intel et dérivés. Tous sont restés populaires pendant au moins deux décennies, surtout en Allemagne. Le C32 était un μC CMOS à basse puissance, avec une (EP)ROM externe et un peu de RAM interne, une aubaine pour les amateurs. Elektor a su tirer parti de la combinaison gagnante 80C32 + 27C128/256/512, en fournissant les mémoires mortes programmées (EPS).

Pour la puce du diviseur de GHz en amont, l'article proposait des alternatives, notamment les SP4731 et SP4732 de Plessey, le SDA4212 de Siemens, le U664B de Telefunken et le SAB6456 de Philips. À l'exception du SP4732, tous avaient tendance à osciller, et, dans ce cas, l'affichage divaguait, surtout en présence de signaux d'entrée trop faibles. On était aux limites des puces de l'époque. En y repensant, j'avais oublié ces imperfections de mon compteur de 600 MHz fait maison avec un diviseur 11C90, je les considérais comme un signe de grande sensibilité, à la manière d'un récepteur reflex au bord de l'accrochage. En amont du prédiviseur, ce modèle était équipé d'un préampli à large bande de 15 dB, lequel manque, à mon avis, au fréquencesmètre 1,2 GHz d'Elektor.

L'affaire du boîtier

Le labo s'est donné bien du mal pour obtenir le look professionnel du boîtier avec sa face avant tricolore. La photo du boîtier ouvert (fig. 2) montre que les cartes imprimées s'adaptent parfaitement au boîtier Vero (en deux nuances de gris). Une page entière de l'article est consacrée au gabarit de perçage.

On aperçoit :

- la carte principale (avec découpe pour le coupleur de pile PP3/6LR61)
- le clavier de commande (à poussoirs *Digitast*)
- le LCD (Philips LTN211-F10, remplacée plus tard par Sharp LM16A21 et Hitachi LM016L)

Allez donc trouver dans la presse électronique actuelle, dominée par les systèmes embarqués, autant de détails fouillés sur l'assemblage manuel d'un tel projet à partir de composants ! Il fallait blinder entièrement la zone du prédiviseur sur la carte. Sur la figure 2, on voit cette puce insérée dans un support DIL à 8 broches, sans doute pour l'interchangeabilité des composants. Une puce UHF, je me serais plutôt attendu à la voir soudée sur la carte afin de réduire les capacités parasites. La paire torsadée entre la prise BNC et l'entrée A de la carte n'est pas un exemple à suivre non plus. À l'époque, l'équipement du labo, un généré RF HP 8640B, ne permettait probablement pas de dépasser 512 MHz.

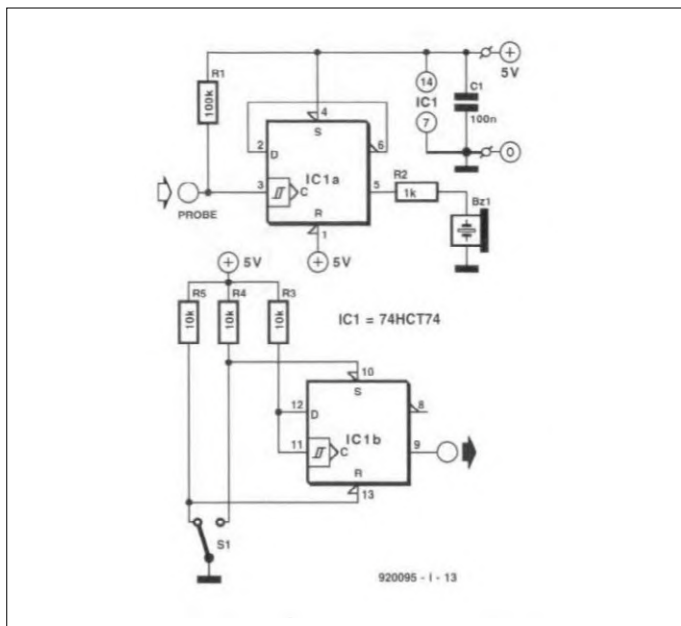


Figure 3. D'accord, on va loin de nos jours en triturant bits et octets avec du logiciel, mais avouez que le matériel ancien est épatant aussi. Ici, une seule puce 74HCT74 sert de générateur de signaux logiques et de sonde logique à sortie audible. Pas mal !

Tester et aligner

La procédure de test de ce projet en construction témoigne des débuts de la technique de logiciel embarqué : le micrologiciel de l'instrument comporte des routines intégrées pour interroger, tester ses propres composants et afficher les résultats. La routine d'autotest globale peut être divisée en trois groupes : LCD, carte mère, E/S en série. À l'époque, ce genre de procédure était novateur.

Si le processeur, l'EPR0M, le LCD, le quartz et le verrou d'adresse fonctionnent bien, on lance le menu de la routine de test en appuyant sur la touche MENU ou ENTER à la mise sous tension, idem pour quitter le menu (BREAK). Les trois tests sont assez complets et doivent sûrement avoir contribué à localiser bien des défaillances dans un circuit si complexe en cours de construction.

Aussi incroyable que cela paraisse aujourd'hui, ce circuit a été réalisé par bien des lecteurs qui ne disposaient pas même d'un oscilloscope. Il devait être facile à tester, ce fréquencemètre. Cependant, comme télécommande et lecture par RS-232 sont optionnelles, il n'était peut-être pas trop grave de devoir se passer d'un PC muni de cette interface.

La simplicité de l'ancien testeur auxiliaire de l'article mérite une mention (fig. 3). Un seul 74HCT74 bistable fait le travail, à la fois comme source d'impulsions et comme sonde numérique audible.

Qu'en est-il de la stabilité de fréquence et de la résolution de l'instrument ? Tout dépend du quartz à 12 MHz. Sur le proto, on a utilisé un quartz bon marché de PC. L'article dit que pour régler l'oscillateur central, le mieux est d'appliquer au fréquencemètre un signal de fréquence connue et de haute précision pour ajuster C2 jusqu'à ce que cette fréquence s'affiche. Avec 28 ans de recul, je trouve qu'on aurait dû cadencer le processeur à 10 MHz, ce qui aurait permis l'usage d'un étalon de fréquence pro, d'un quartz en four thermostatique ou même



Figure 4. L'ensemble du projet vendu par Elektor en Allemagne comprenait un livre et un kit complet. Cette offre groupée n'est apparue qu'en 1995, bien après la publication du fréquencemètre dans le magazine Elektor.

de l'étalon de fréquence de 10 MHz d'Elektor. Aujourd'hui, on prendrait un oscillateur asservi par GPS.

Le grand jeu

Peu après la publication de l'article, le chiffre de ventes de cartes a témoigné du vif intérêt porté par nos lecteurs à ce projet. Quelqu'un prit l'initiative de faire préparer par un revendeur (*Geist Elektronik ?*) des kits qui seraient vendus sous la marque Elektor. Avec un petit livre qui reprenait les articles ! La vente par Elektor de kits basés sur un article de magazine était rare. En ce temps, nous ne proposons que des circuits imprimés nus, laissant à nos annonceurs le soin (et le risque) d'en faire des kits.

Cet ensemble (fig. 4) est introuvable aujourd'hui. Il ne s'en serait vendu qu'une centaine, ce qui est peu comparé aux



Figure 5. Contenu du kit (1995). Pour préserver leur virginité authentique, je n'ai pas ouvert les sachets de composants électroniques.

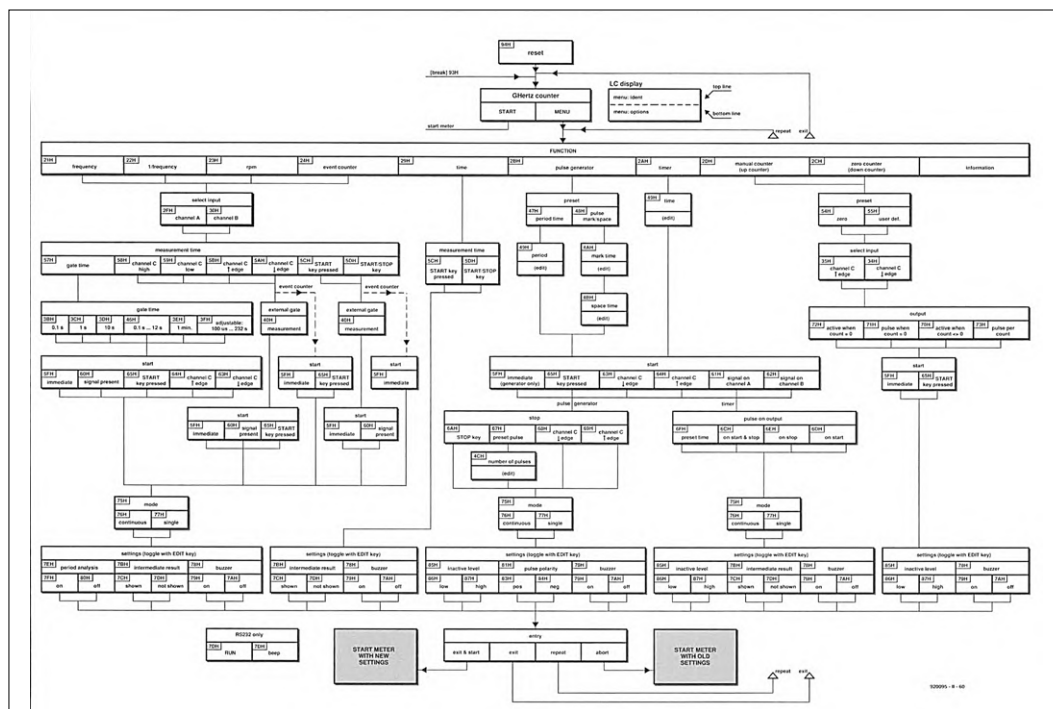


Figure 6. L'original de cet organigramme des menus et des options d'utilisation dessiné par la rédaction en 1993 mesurait 40 x 26 cm. Giga !

2 000 circuits imprimés, EPROM et faces avant écoulés par ailleurs. La boîte blanche contient le kit, avec le boîtier usiné, l'EPROM, l'adhésif pour la face avant et toutes les pièces mécaniques, comme au temps du *Heathkit*. Le contenu sauvé de la benne en 2006 est intact. J'ai été ravi d'y trouver aussi du fil nu rigide et les morceaux de tôle pour confectionner le blindage... Les résistances, les supports, les circuits intégrés, les poussoirs, l'afficheur, les prises BNC, l'EPROM programmée, tout est là, soigneusement emballé dans des pochettes de papier blanc (**fig. 5**).

Menu s'il vous plaît !

Un rédacteur a passé deux semaines à documenter la structure du logiciel jusque dans les moindres détails, pour aboutir à un organigramme imposant qui donne une image parfaitement claire de toutes les options, du texte affiché et de tous les codes d'erreur imaginables. La modeste reproduction (**fig. 6**) ne fait pas justice à ce magnifique document de 400 x 260 mm, mis au propre par un graphiste dans le style racé des illustrations d'Elektor, puis imprimé sur papier glacé (!).

Récalcitrant tantale

J'ai essayé de faire fonctionner l'instrument en insérant une

pile neuve de 9 V. Quand j'ai actionné le petit interrupteur à glissière, il ne s'est rien passé. J'ai sondé le câblage d'alimentation au voltmètre et mesuré +3 V derrière l'interrupteur ; apparemment il circulait beaucoup de courant. J'ai suspecté les condensateurs électrolytiques sur les rails d'alimentation, notamment ceux au tantale (C14 et C17). Heureusement que j'avais utilisé une pile, et seulement pendant quelques secondes, et pas une alimentation de table qui aurait grillé d'autres composants. Quelque chose me dit que cet instrument pourrait de nouveau fonctionner et j'ai l'intention de faire un

de ces jours les réparations nécessaires.

Les trois articles sur le

fréquencemètre 1,2 GHz

sont dans les archives en ligne d'Elektor sous forme de fichiers PDF gratuits pour les membres abonnés Green et Gold [1] [2][3]. Ils sont aussi sur le DVD d'Elektor des années 1990. Est-il besoin de préciser que, aussi giga soit-il, nous ne recommandons pas de vous lancer dans la construction de ce projet aujourd'hui, car plusieurs composants ne sont plus disponibles.

(190382-C-02 VF Robert Grignard)

EST^D 2004

Rétronique est une rubrique bimestrielle sur les pages glorieuses et jaunies de l'électronique, avec occasionnellement des montages de légende décrits dans Elektor. Merci de télégraphier à redaction@elektor.fr vos suggestions de sujets à traiter.

Liens

- [1] fréquencemètre 1,2 GHz et générateur de signaux carrés (1), Elektor, sept. 1992 : www.elektormagazine.fr/magazine/elektor-199209/35289/
- [2] fréquencemètre 1,2 GHz et générateur de signaux carrés (2), Elektor, janv. 1993 : www.elektormagazine.fr/magazine/elektor-199301/35344/
- [3] technique alternative de mesure de fréquence, Elektor, février 1993 : www.elektormagazine.fr/magazine/elektor-199302/35352



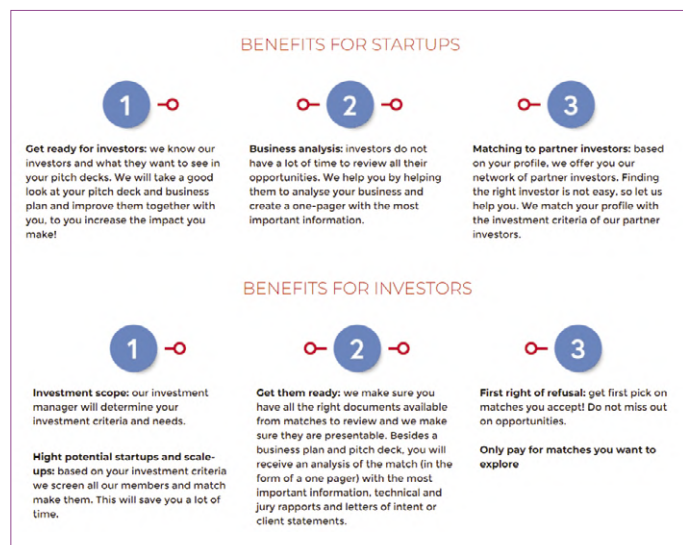
Le programme d'investissement Elektor Oser d'abord, doser ensuite – Enregistrez votre start-up !

Le programme d'investissement Elektor permet aux jeunes entreprises et aux investisseurs axés sur l'électronique d'établir entre eux de solides contacts et d'exceller ensemble. Ce ne sont pas les bonnes occasions qui manquent, encore faut-il les saisir !

Qu'est-ce que le programme d'investissement Elektor ?

Dans le cadre d'une start-up, il arrive un point où les solutions de bouts de ficelle ne marchent plus. Vous avez peut-être déjà quelques clients payants, mais il vous faut des fonds pour passer au niveau supérieur. Qu'il s'agisse d'embaucher une équipe, de trouver des marchés ou de vous développer, pour cette étape il faut des investisseurs. Vous êtes le fondateur de votre société, et vous vous demandez avec raison : *Mes propositions sont-elles crédibles ? Mon plan d'affaires et mon dossier sont-ils au point ? Les investisseurs me prendront-ils au sérieux ?*

L'équipe du programme d'investissement d'Elektor vous guide. Nous pouvons vous aider à élaborer votre plan d'affaires, à préparer votre présentation et vous mettre en relation avec des investisseurs de notre réseau.



Pourquoi Elektor a-t-il lancé ce programme ?

Si vous vous contentez d'envoyer votre dossier à des investisseurs en espérant les convaincre, vous serez probablement déçu pour diverses raisons. Et si c'était le mauvais investisseur ? Peut-être manque-t-il dans votre présentation des d'informations cruciales. Une chose est sûre : il faut mettre toutes les chances de votre côté dès vos premières démarches.

Le programme d'investissement d'Elektor est né de notre expérience avec les concours de start-ups que nous organisons, pour lesquels nous avons côtoyé de nombreuses start-ups prometteuses. L'utilité d'un programme d'accompagnement soutenu des jeunes entreprises dans la recherche de financements et de partenaires s'est logiquement fait sentir. Nous aidons aussi les investisseurs qui reçoivent des dizaines

de dossiers impossibles à lire tous, d'autant que la plupart font piteux effet. Notre expérience permet aux investisseurs de gagner du temps : nous établissons pour eux le contact dans notre réseau mondial avec des start-ups technologiques prometteuses et qui correspondent à leur champ d'investissement.

Prestations pour les start-ups

Vous êtes dans une start-up ou une entreprise établie à la recherche de financement ? En participant au programme d'investissement, vous pouvez augmenter vos chances d'obtenir un financement. Nous proposons :

- Une aide personnelle pour peaufiner votre présentation et votre plan d'affaires.
- Analyse d'entreprise pour établir le profil de votre entreprise.
- Mise en relation avec des investisseurs adéquats (en fonction de votre profil) de notre réseau.

Avantages pour les investisseurs

Notre équipe ne se contente pas d'organiser des concours de création d'entreprise. Nos professionnels de la création d'entreprise recherchent aussi dans le monde entier des entreprises prometteuses, sélectionnées par nos experts, ce qui ouvre aux investisseurs des pistes prometteuses. Nous fournissons :

- Accès à des start-ups techniques à fort potentiel et à des jeunes sociétés déjà établies adéquates pour votre investissement.
- Une documentation appropriée, des relations d'affaires professionnelles et une analyse commerciale de vos relations.

Vous souhaitez rejoindre le programme d'investissement d'Elektor ou en savoir plus ? Visitez le site du programme ou envoyez un courriel à Arthur Lampe, responsable du programme d'investissement d'Elektor. N'hésitez pas. Bonne chance !

200112-02

Liens

- [1] Elektor Investment Program:
www.elektormagazine.com/widgets/investment-program
- [2] Elektor Start-up Competitions:
www.elektormagazine.com/select/fast-forward-award

Hexadoku casse-tête pour elektorniciens

Votre magazine se termine toujours et encore par une grille de sudoku particulière. Éteignez le fer à souder, oubliez les écrans de PC et d'oscilloscope, rangez les pinces et les fils. Faites travailler votre matière grise d'une autre façon : attrapez un crayon pour remplir la grille d'hexadoku.

N'oubliez pas d'envoyer votre réponse pour tenter de gagner un bon d'achat.

Une grille hexadoku est composée de chiffres du système hexadécimal, de 0 à F. Remplissez le diagramme de 16 x 16 cases de telle façon que **tous** les chiffres hexadécimaux de 0 à F (0 à 9 et A à F) n'apparaissent **qu'une seule et unique fois** dans chaque rangée, colonne et carré de 4 x 4 cases (délimités par un filet gras).

Certains chiffres, déjà placés dans la grille, en définissent la situation de départ.

Pour participer, inutile de nous envoyer toute la grille, il suffit de nous donner **la série de chiffres** sur fond grisé.



Participez et gagnez !

Nous tirons au sort **cinq** des réponses internationales correctes reçues dans les délais ; leurs auteurs recevront chacun un bon d'achat Elektor d'une valeur de **50 €**. À vos crayons !

Où envoyer votre réponse ?

Envoyez votre réponse (les chiffres sur fond grisé) avec vos coordonnées par courriel, avant le **1^{er} juin 2020** à l'adresse **hexadoku@elektor.fr**

Les gagnants

La solution de la grille du numéro de mars/avril 2020 est **749DF**.

Les cinq bons Elektor d'une valeur de **50 €** vont à : Peter Tabatt (Allemagne) – Ria Van Broeck (Belgique) – Jose A. Gordo (Espagne) – Etienne Guyot (France) – Tomas Bakke (Norvège).

Bravo à tous les participants et félicitations aux gagnants !

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

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

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.

Rejoignez la communauté Elektor

Devenez membre GOLD maintenant !



GOLD

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



Également disponible

abonnement « zéro papier » GREEN !



GREEN

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



www.elektormagazine.fr/membres



COMPETE TO
LAUNCH
YOUR STARTUP AT
ELECTRONICA
2020

Votre start-up est-elle prête pour l'avenir ?
Alors le moment est venu pour votre
entreprise de présenter ses solutions sur le
marché mondial !

**PARTICIPEZ MAINTENANT !
C'EST RAPIDE ET FACILE.**

Sur le site elektormagazine.fr/fastforward vous pourrez
lire les conditions de participation et soumettre votre
candidature. Bonne chance !

LES AVANTAGES DES FINALISTES :

- **Premier prix = 75 000 €** de plan marketing d'Elektor, plus
un stand sur electronica 2022
- **deuxième prix = 50 000 €** de plan marketing d'Elektor
- **troisième prix = 25 000 €** de plan marketing d'Elektor
- **un stand (5 m²) exclusif et très avantageux**
(seulement 1 905 € !) sur electronica 2020
- **visibilité accrue** avant, pendant et après le salon
electronica 2020 (du 10 au 13 nov. 2020)



Sign up
here