





LoRa™

noeud LoRa d'Elektor

Uplink frequencies

carrier freq. MHz

Downlink frequencies

carrier freq. MHz

Legend:

- 125 kHz @ SF 1-7, for downlink slot 1
- 250 kHz @ SF7, for downlink slot 1
- 125 kHz @ SF9 for downlink slot 2

Commande à distance souple, 868 MHz, à longue portée, avec retour d'état et STM32 embarqué



sonnette ESP32 par Telegram



« Le facteur sonne rarement une deuxième fois »

carte Meadow F7 pour développeurs .NET



« Le bonheur est dans le pré ! »

photosonde pour oscillo dessine-moi un bouton pour l'IIdO Sigfox : un renard sur l'IIdO (3) multitâche en pratique avec l'ESP32 banc d'essai : HAT Enviro+ pour Raspberry Pi banc d'essai : microscope Andonstar AD407 banc d'essai : oscillo de poing 3 en 1 JOY-iT DMS02D72 hors-circuit : filtres analogiques à capacités commutées interpréteur BASIC pour ESP32 et ESP8266 j'assemble un PC pour mon labo



Rémy Mallard explique les oscilloscopes anciens & modernes pour les débutants



Points forts

- Livre orienté vers la pratique, avec « juste ce qu'il faut » de théorie
- Traite aussi le sujet des oscilloscopes numériques modernes
- Ouvrage utile avant l'achat pour guider le choix d'un oscilloscope
- Manuel d'utilisation à consulter régulièrement après l'achat
- Comme dans ses précédents livres, l'auteur partage sa bonne humeur communicative et sa grande expérience

elektor

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, répond à ces questions.

Auteur : Rémy Mallard
Pages : 375 (en couleur)
Format : 17 x 23,5 cm (broché)
ISBN : 978-2-86661-208-5
Prix : 37,50 €

Pour commander ce livre :
www.elektor.fr/19124



Elektor est édité par :

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

service@elektor.fr

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

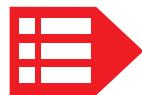
DROITS D'AUTEUR :

© 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.



Editorial

Notre labo a pas mal travaillé ces derniers mois sur des projets LoRa pour la transmission sans fil de données. Les distances couvertes peuvent être qualifiées de longues, et la consommation d'énergie de faible. Dans ce numéro, vous trouverez deux articles qui rendent compte dans les grandes largeurs de nos efforts sur ce sujet passionnant et exigeant. Le **nœud Elektor LoRa** est un module configurable et flexible qui, une fois couplé avec n'importe lequel de vos propres projets, le rendra compatible avec LoRa. À titre d'exemple, nous proposons une télécommande pour commuter des charges sous 230 V à plusieurs centaines de mètres de distance. Comme il convient à LoRa, un tel nœud, une fois muni de cellules Li-Ion, est autonome sur le terrain.

Si vous êtes intéressé par la mise en ligne de données issues de capteurs afin de les rendre disponibles sur l'internet, vouserez avec intérêt l'article sur LoRaWAN dans ce numéro. Il montre comment connecter facilement une carte au réseau ouvert *The Things Network*. Vous utiliserez au choix une passerelle existante dans cet environnement, ou installerez à moindres frais votre propre passerelle pour recevoir les données du capteur via LoRa.

Pendant ce temps, l'univers des microcontrôleurs se développe dans toutes les directions dont celle de l'intelligence artificielle n'est pas la moins prolifique. Nous espérons trouver dans ce domaine des innovations remarquables lors du grand salon *Embedded World*, dont nous reparlerons dans Elektor. Dans notre boutique, nous proposons déjà le kit MAix BiT. Programmable via l'EDI Arduino, il permet de réaliser vos propres expériences de reconnaissance de la parole et de l'image.

Jens Nickel

Rédacteur en chef

www.elektormagazine.fr

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

Une invitation à l'expLoRation

Projets du labo

6 nœud LORA d'Elektor

Commande à distance souple, 868 MHz, à longue portée, avec retour d'état et STM32 embarqué

29 sonnette ESP32 par Telegram

« Le facteur sonne rarement une deuxième fois »

36 LoRaWAN : décollage facile

Avec Blue Pill, passerelle LoRa et The Things Network

Projets de lecteurs

60 Sigfox : un renard

sur l'internet des objets (3)

Vos premiers pas sur l'internet des objets

80 photosonde pour oscilloscopes

Mesure de fluctuation de luminosité des systèmes d'éclairage

Savoir-faire

16 mises au point & mises à jour

Corrections - Questions - Réponses

18 électronique analogique

Étude de cas n° 1 - Section 2 : Préamplificateur optimisé pour les microphones MEMS

34 retour des petits circuits

... et des bonnes petites idées de projets d'Elektor

84 comment calculer le courant de court-circuit présumé ou PSCC

et choisir le bon disjoncteur

88 filtres analogiques à capacités commutées

Ceci n'est pas la rubrique Rétronique même si elle s'appuie sur un principe décrit en 1873 !

98 démarrer en électronique (2)

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

Logiciel

21 dessine-moi un bouton pour l'IdO

boutonnière n° 1 : Architecture de l'IdO

26 interpréteur BASIC pour ESP32 et ESP8266

Programmation avec Annex WiFi RDS

6



sonnette ESP32 par Telegram



« Le facteur sonne rarement une deuxième fois »

46 « Un pilier de l'internet ouvert »

Entretien avec Wienke Giezeman, initiateur du réseau *The Things Network*

49 carte Meadow F7

Une carte pour développeurs .NET

54 multitâche en pratique avec l'ESP32 (2)

Priorités des tâches



Start-ups

68 l'accélérateur de démarrage le plus performant d'Europe ?

HighTechXL, Eindhoven, Pays-Bas

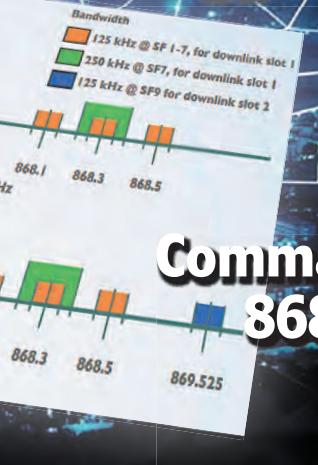
94 bureau d'études – Zone D

Trucs et astuces, bonnes pratiques et autres informations utiles

Entretien avec
Wienke Giezeman,

initiateur du réseau
The Things Network

noeud LoRa d'Elektor



Commande à distance souple,
868 MHz, à longue portée,
avec retour d'état et
STM32 embarqué



carte Meadow F7

Une carte pour développeurs .NET



46

Bancs d'essais

- 71 **banc d'essai :**
HAT Enviro+ pour Raspberry Pi
Mesure et collecte de la qualité de l'air avec RPi et le HAT Enviro+
- 76 **banc d'essai :**
microscope Andonstar AD407
Est-il le meilleur de sa catégorie ?
- 100 **banc d'essai :** oscillo de poing 3 en 1
JOY-iT DMSO2D72

Ils ont fait ça comme ça

- 67 **Raspberry Pi**
Aide-mémoire des commandes Bash
- 74 **expérience vécue**
Commandes de composants en Ukraine et en Russie
- 78 **Une soue ? Non, un labo d'électronique !**
Visite guidée dans le saint des saints
- 82 **projet TABULA – des nouveautés tangibles**
De l'importance du retour d'information
- 87 **afficheur à LED Monsanto MAN1**
drôle(s) de composant(s)
- 102 **j'assemble un PC pour mon labo**
Conseils pour le choix des composants

Classiques (grands & petits)

- 106 **rétronique :**
l'ordinateur de jeu d'échecs intelekt d'Elektor (1981)
Tiny Chess 86 porté sur l'Intel 8088
- 114 **hexadoku**
casse-tête pour elektorniciens



bientôt sur ces pages

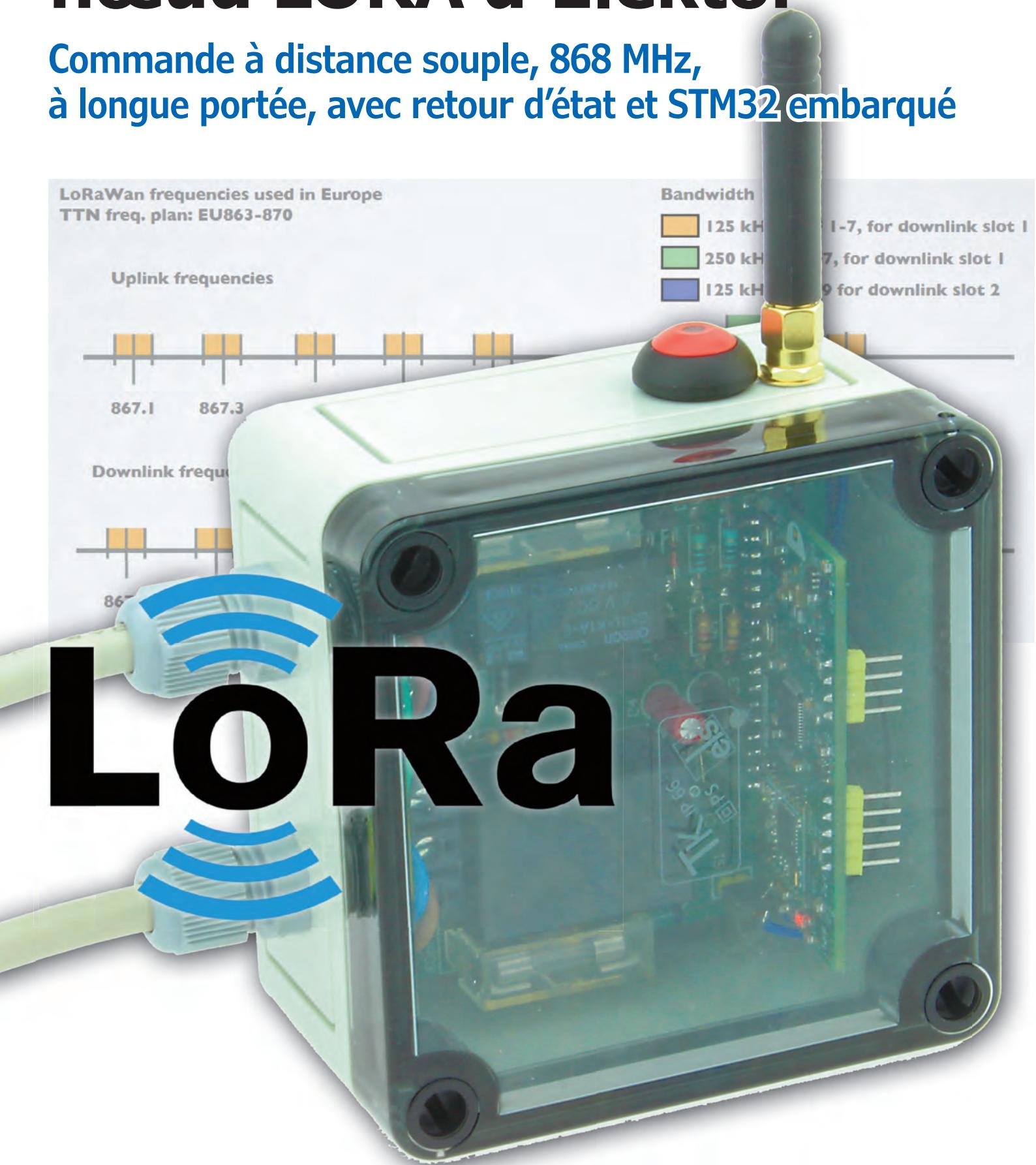
La rédaction et le labo travaillent sur le prochain numéro dont le menu sera copieux. En voici un avant-goût :

- Station météo 2020
- Expérimentations sur l'IA
- Thermomètre Nixie à bargraph 2.0
- Afficher les données Sigfox
- Démontage de scooter électrique
- Robot maintenu en équilibre avec Arduino
- Base pour ESP32 : Application
- Triac et ATmega
- FreeRTOS et ESP32 : temporisateur
- ARM A7 et M4 sur une même puce
- ... et bien davantage

Sous réserve de modification.
Le numéro de mai/juin 2020 paraîtra le 7 mai 2020.

nœud LORA d'Elektor

Commande à distance souple, 868 MHz,
à longue portée, avec retour d'état et STM32 embarqué



Luc Lemmens & Mathias Claussen (Elektor Labs)

Convaincu de la puissance combinée du logiciel et du matériel ouverts, Elektor montre ici que LoRa n'est pas réservé aux professionnels. Cet article raconte l'histoire de la ténacité d'un concepteur face à des défis ardu. Il présente un nœud LoRa, facile à reproduire, pour une commande à distance avec retour d'état. La portée est de 10 à 20 fois celle du WiFi grand public ou des communications à faible puissance (LPR) dans la bande Industrie-Science-Médecine (ISM) 433 MHz.



Nous partagerons ici le produit final et un bilan de qualité, mais aussi tous les problèmes résolus ainsi que les outils utilisés pour mener à son terme un projet ambitieux. Le récit aurait pu commencer comme ceci : « ce nœud LoRa est né de l'idée d'un circuit imprimé compact alimenté par des piles ou des batteries rechargeables et embarquant un micro-contrôleur (μ C) courant tel que le STM32 à boîtier LQFP48 ainsi qu'un module LoRa de type RFM95, le tout casé dans un coffret disponible chez l'un des plus gros distributeurs et quant au compilateur STM32, nous...» **STOOOP !** Reprenons tout depuis le début, mais en mode *staccato*.

C'est le coffret qui détermine tout : volume disponible et source d'énergie, c'est donc par lui qu'il faut commencer. Idée initiale : alimentation par piles non rechargeables. Abandonnée, trop de piles partout. Recherche d'une solution à batteries rechargeables, faciles à remplacer. Objectif : nœud de petite taille, étanche, facile à maintenir, fonctionnel au labo, mais tout aussi bien en plein air, et capable de transmettre des données d'environnement réel. Autres sujets de débats : la durée de vie des batteries et la liste des capteurs supportés, embarqués ou externes. On imagine de préférence une plateforme souple et utile pour explorer LoRa et ultérieurement LoRaWAN, pour un usage d'abord académique, dans l'ambiance feutrée d'un

labo, mais capable aussi de commander à distance une fonction marche/arrêt dans n'importe quel environnement difficile. Autrement dit : la quadrature du cercle entre souplesse, consommation et encombrement.

Cet article décrit les trois éléments suivants :

- **Nœud LoRa**, carte n° 180516-1, la plateforme expérimentale LoRaWAN, équipée pour le LoRa local seulement. Il en faut deux pour le projet.

- **Bouton LoRa**, carte n° 180666-2.

- **Interrupteur de puissance LoRa** pour courant alternatif, carte n° 180666-1.

Ensemble, ces trois éléments constituent non seulement une commande à distance avec une portée de 10 à 20 fois celle d'une liaison grand public à 2,4 GHz, mais aussi, en projet, une plateforme de développement avec LoRaWAN.

C'est parti pour l'élément principal, le nœud LoRa !

Caractéristiques (abrégées)

Nœud LoRa

- Plateforme expérimentale LoRaWAN avec équipement minimum
- Alimentation par batteries Li-Ion
- Batteries remplaçables par l'utilisateur
- Microcontrôleur STM32F072C8T6TR ARM Cortex-M0
- SPI F-RAM ou Flash (facultatif)
- Co-processeur de cryptage (facultatif)
- Module GPS (facultatif)
- Interface USB (facultatif)

Interrupteur LoRa pour courant alternatif (esclave)

- Retour de l'état du relais
- Pouvoir de coupure 5 A (1000 W à 230 V~; 500 W à 115 V~)

Bouton LoRa (maître)

- Faible consommation
- Afficheur OLED (facultatif)
- Antenne en hélice
- Portée vers l'esclave LoRa : 100 à 500 m

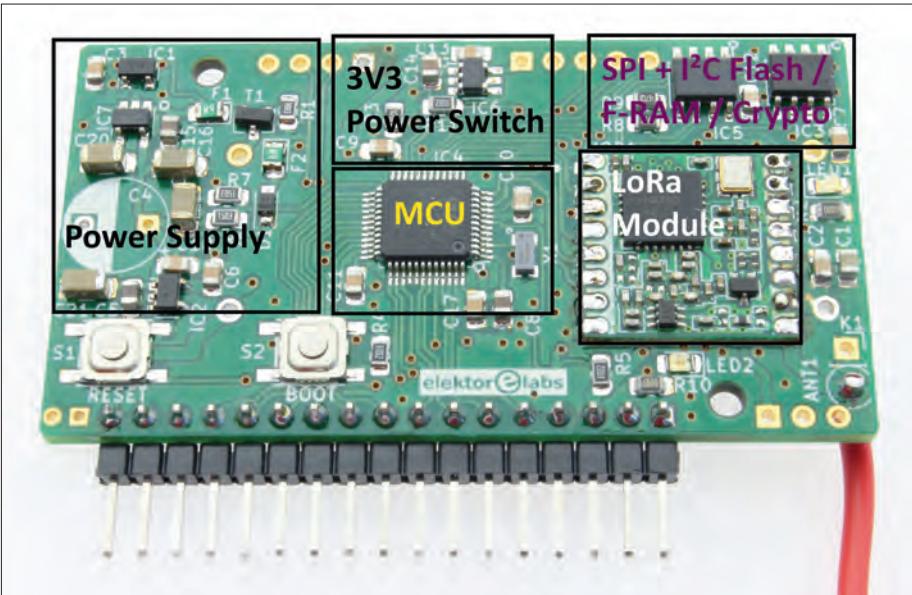


Figure 1 : La carte de gestion du nœud LoRa avec les espaces alloués aux blocs fonctionnels. En fait, ce que vous voyez, c'est une plateforme expérimentale du nœud LoRaWAN avec un équipement minimum.

Au tout début

Le circuit imprimé a été conçu avec KiCad, un outil dont la popularité augmente, aussi chez les pros. Si vous voulez vous y mettre, Elektor vous propose un excellent livre [1].

Après un faux départ avec un autre processeur et des tâtonnements, nous avons conçu une carte pour le STM32F072 en boîtier LQFP48, profitant ainsi de sa compatibilité avec l'EDI Arduino. Beaucoup de bibliothèques Arduino bien connues sont utilisables. Faute de modèle de quartz convenant à notre carte, nous nous sommes rabattus sur les options d'horloge interne et la valeur par défaut de 8 MHz, afin de réduire la consommation, mais parés à passer à 48 MHz si nécessaire.

Approche modulaire

La version précoce de la carte du nœud LoRa d'Elektor (fig. 1) n'est pas seulement la base du présent projet, elle servira aussi pour des projets dérivés à publier ultérieurement. C'est pourquoi nous allons commencer par faire le tour des composants et autres options susceptibles de se retrouver sur la carte. Les lecteurs chevronnés suivront en même temps sur le schéma (fig. 2).

Afin de respecter certaines exigences, nous nous en sommes tenus à la modularité. Si vous implantiez tous les compo-

sants, vous obtiendriez un système LoRaWAN avec GPS, coprocesseur de cryptage et mémoire SPI F-RAM ou Flash. Impressionnant, mais pas indispensable. Les composants laissés de côté réduisent le coût de la version de base ; inutile de payer pour des fonctions que vous n'utiliserez pas. Le brochage des SPI F-RAM et des SPI Flash est standard, vous pouvez donc utiliser votre module favori.

Idem pour le coprocesseur de cryptage ATECC608a. Vous pouvez aussi vous contenter d'une puce I²C-EEPROM ou F-RAM : vous disposerez d'un système souple et configurable selon son emploi. Le microcontrôleur STM32F072 à boîtier LQFP48 présent sur la carte est une puce Cortex-M0 avec 64 ko de mémoire flash et 16 ko de RAM, avec possibilité d'utiliser l'USB pour les mises à jour du logiciel et vos propres applications.

Le brochage de tous les membres de la famille STM32 est compatible ; vous pouvez donc changer de modèle sans modifier la carte. Ce peut être le STM32L072 à faible consommation, ou bien le STM32L151 pour sa puissance de calcul et sa mémoire Flash. Même le STM32F103, qu'on trouve habituellement sur la carte Bluepill, peut y prendre place.

L'encadré *Caractéristiques (abrégées)* est une (tentative de) démonstration de la souplesse d'utilisation de la carte.

Conception du circuit imprimé et défis

Après les premiers prototypes, des ajustements s'imposaient. Pour le module GPS, il a fallu améliorer la réception. Ajouter une petite antenne ? Il faut alors tenir compte du plan de masse. Le module GPS exige une taille minimum du circuit imprimé pour que l'antenne fonctionne. Pas solution satisfaisante en vue pour améliorer la qualité de la réception.

Après la version 1 le module Quectel L96 GPS a été remplacé par le module vendu par Elektor [2] moins cher et plus facile à connecter. Modification consécutive des connecteurs utilisés, de sorte que tout est maintenant au pas de 2,54 mm, compatible avec une carte d'expérimentation.

Nous avons ensuite le module LoRa RFM95W qui ne se passe pas d'antenne. Pour l'encombrement, nous voulions une antenne maison hélicoïdale pour 868 MHz. Pour améliorer la réception, une antenne extérieure est une option aussi. La leçon à retenir : la conception est un compromis entre encombrement, fonctionnalité et performances radio. Les batteries Li-Ion au format 10440 (AAA), comme on en trouve dans les e-cigarettes, sont faciles à remplacer par l'utilisateur et à recharger hors l'appareil par un chargeur approprié. Pour les protéger d'une décharge profonde, nous avons inclus une puce de surveillance de tension qui coupe le régulateur à faible chute de tension (LDO) de l'alimentation 3,3 V. Un connecteur femelle aux broches de 2 mm permet de connecter des paquets de batteries LiPo utilisées sur les drones. Plats, de petite taille, équipés d'un connecteur Molex 510005 ou JST, ils ont un ratio capacité/volume intéressant. Avantages : souplesse d'utilisation, batteries standard, faciles à trouver. La protection contre l'inversion de courant a été réalisée avec un *interrupteur de charge*. Nous avons fini par utiliser la diode idéale MAX40200, qui coupe l'alimentation dès qu'un certain seuil de courant inverse est atteint. Une seconde, sur l'alimentation du module GPS (K4), permet au µC de couper l'alimentation de ce module pour ménager les batteries. Si vous n'utilisez pas ce module, vous pouvez utiliser cette ligne d'alimentation 3,3 V pour commander la mise sous tension d'un circuit extérieur. Il est en principe possible d'alimenter le système par des batteries NiMH, mais

au prix d'un circuit supplémentaire, ici un convertisseur continu-continu élévateur. Comme ce module consommera

en permanence, cela aura un effet sur le temps de fonctionnement, et il nous faudra aussi revoir la protection de

sous-tension.

Nos défis majeurs semblent maintenant se concentrer dans le logiciel.

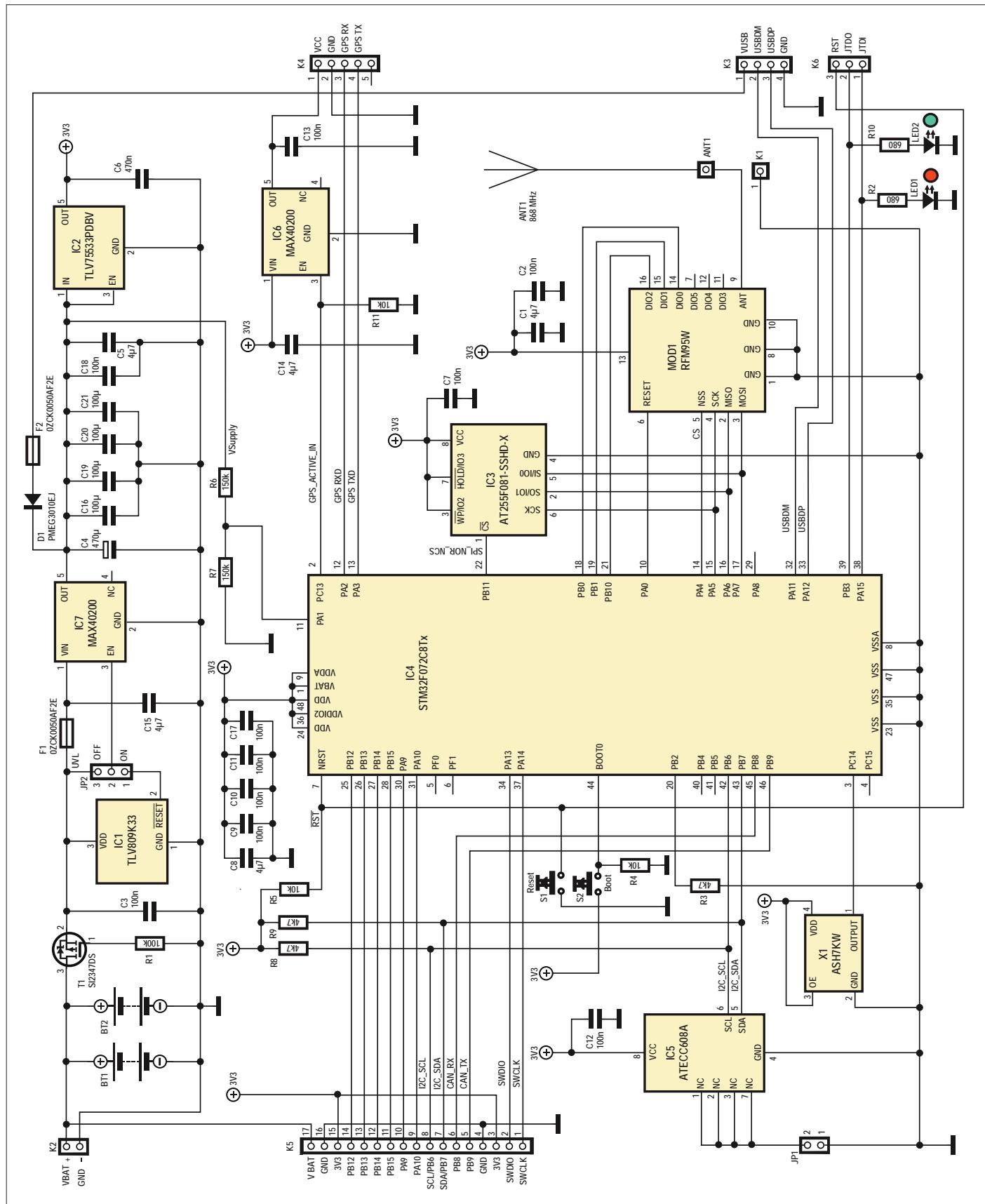


Figure 2 : Schéma de la carte de gestion du nœud LoRa. Ceci est fondamentalement la plateforme expérimentale du nœud LoRaWan équipée d'une série de puces spécifiques.

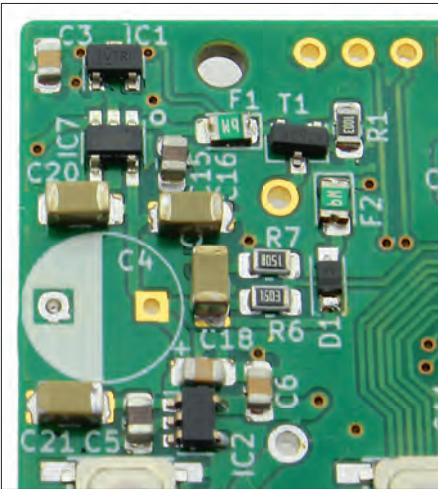


Figure 3 : L'espace de la carte réservé au condensateur électrolytique C4, l'un des composants encombrants.

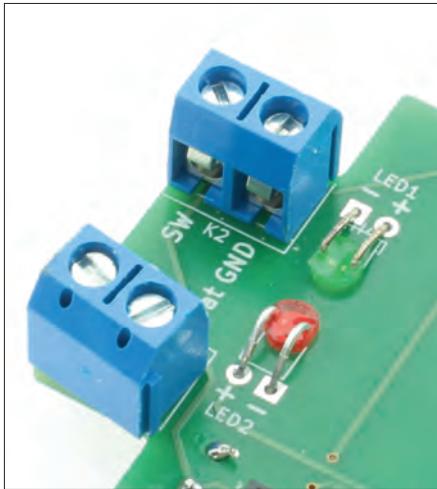


Figure 5 : Les LED sont montées de manière inhabituelle sur la carte bouton LoRa.

Aucunement intimidés, nous tentons la communication de point à point en mode RAW proposée par la bibliothèque LMIC. Problèmes au niveau des compo-

sants, avec un condensateur de 470 μ F monté en amont du circuit de protection de sous-tension IC1. Si la tension tombe en dessous de 2,97 V, ce circuit

coupe l'interrupteur HS IC7, ce qui interrompt l'alimentation pendant 20 ms. Si le condensateur tampon est insuffisant, les courants de démarrage élevés perturbent le contrôleur de sous-tension (UVLO) et sa tension d'entrée tombe en dessous de 2,97 V, ce qui provoque une oscillation permanente de conduction-blocage du régulateur LDO. Problème résolu par le gros électrolytique C4 (**fig. 3**). Ne l'ajoutez que si les deux condensateurs solides à céramique multicouche (MLCC) de 100 μ F vous paraissent hors de prix. Les feuilles de caractéristiques montrent aussi qu'il n'y a pas de FET ou de diode qui empêcherait les batteries d'être chargées par la tension fournie par l'USB. La MAX40200 coupe l'alimentation s'il circule un courant inverse. C'est également le cas si la tension USB est supérieure à la tension fournie par les batteries du système.

Les deux LED indiquent l'état de la carte fourni par le μ C sur ses ports PB3 et PA15, soit 'JTDO' et 'JTDI' sur le connecteur K6.

Logiciel : les fondations

Le logiciel du système a beau déterminer les fonctions du matériel dans des proportions toujours croissantes, certains n'y voient rien d'autre qu'un peu de magie noire à flasher dans une puce. Comment se fait-il que le temps passé sur le logiciel l'emporte à ce point sur celui passé sur le matériel ?

Vous pouvez programmer la carte avec votre EDI Arduino. Pour cela, le projet *stm32duino* [3] est d'une grande utilité, car il apporte le support d'une gamme étendue de μ C et de cartes STM32.

Y ajouter dans les règles la plateforme expérimentale du nœud LoRa d'Elektor est la première tâche administrative : éditer quelques fichiers du noyau Arduino, ajouter la définition de la carte. Après quelques erreurs et comportements imprévus du logiciel, le canevas fonctionne, nous pouvons écrire les premières lignes de code.

D'habitude on ne se préoccupe guère de la façon dont le μ C gère son horloge. Sur les puces AVR, on avait un quartz, et voilà. Les μ C récents ont davantage d'options pour l'horloge principale (PLL / FFL ou oscillateurs RC à haute fréquence internes). Pour commencer, la stabilité des sources internes suffira. La réduction de la fréquence d'horloge en cours de fonctionnement permet de réduire la consommation, tandis qu'une fréquence élevée augmente la puissance

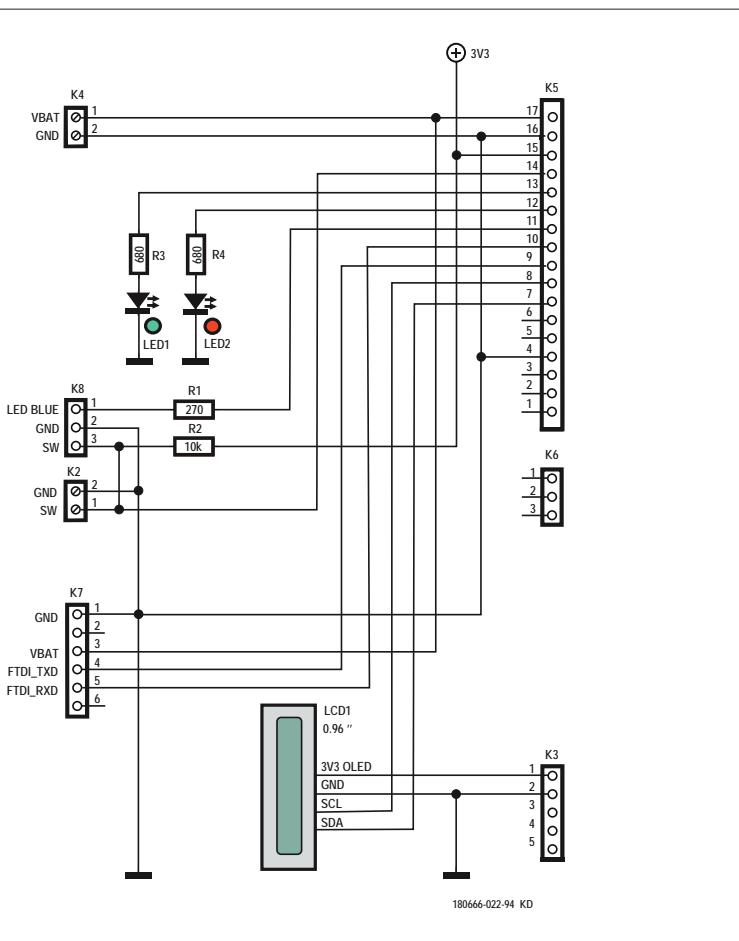


Figure 4 : Schéma du bouton LoRa. C'est la partie «émission» ou «maître» du nœud LoRa d'Elektor. L'afficheur OLED est facultatif et son usage devrait être subordonné à une réflexion approfondie sur le bilan énergétique !



Figure 6 : Utilisez des pinces bien coupantes pour tronquer le plastique entre les broches.

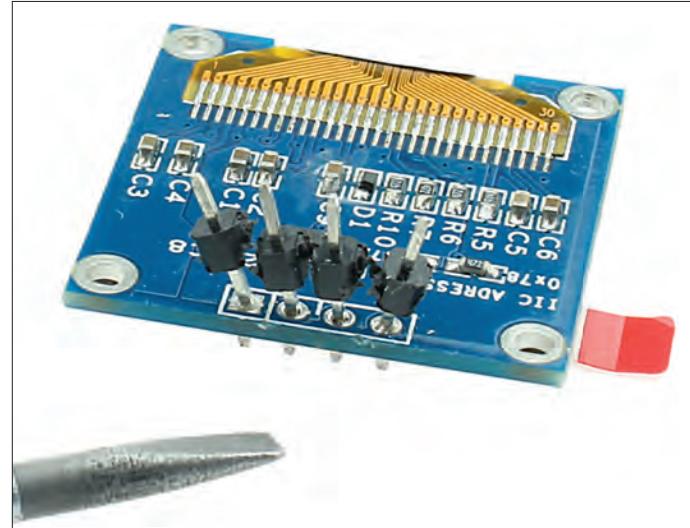


Figure 7 : Avec la pointe d'un tournevis, dégagéz et enlevez les morceaux de plastique.

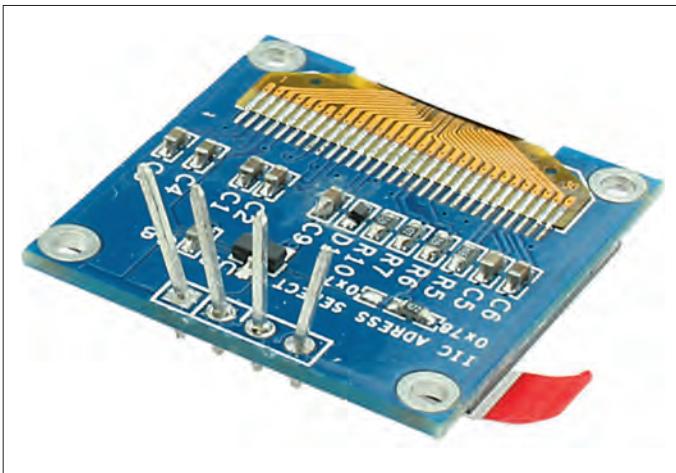


Figure 8: Vous avez maintenant une rangée de broches dénudées.

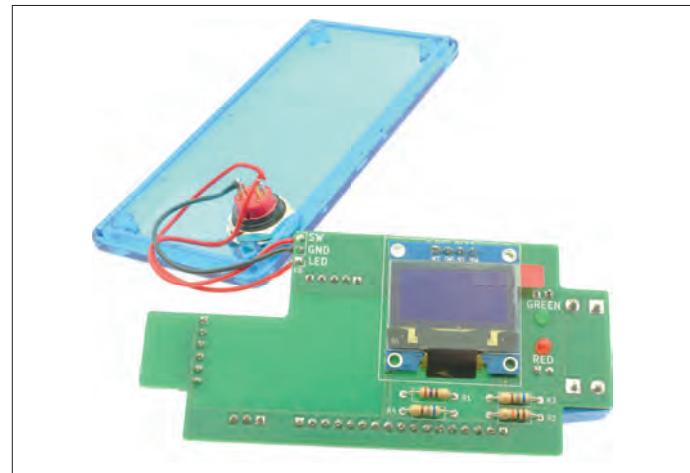


Figure 9 : La carte bouton LoRa au complet.

de calcul. La valeur par défaut de 8 MHz pour le cœur et les périphériques limite la consommation, mais reste suffisante pour la pile LoRa.

Pour l'horloge en temps réel, nous avons ajouté un oscillateur à 32768 kHz (X1) facultatif. Pour un µA de courant en plus, on a des UART cadencés par quartz ainsi qu'une horloge en temps réel capable de réveiller le µC après un délai donné.

Le bouton LoRa

Ceci est un simple organe de saisie, qui transmet son état par radio ; c'est le second composant du projet, le bouton LoRa (**fig. 4**) constitué d'un circuit imprimé ne comportant que quelques

résistances, une paire de LED, des borniers à vis et des barrettes mâles et femelles. Ces composants sont essentiels pour la version de base du noeud LoRa d'Elektor. L'afficheur OLED de la boutique Elektor [**3**] (composant : LCD1) est facultatif. Attention à l'assemblage, notamment pour les LED, retournées et insérées dans des trous du circuit, de sorte qu'elles émergent du côté inférieur de la carte (**fig. 5**).

Pour l'afficheur OLED, il faut s'y prendre avec précaution, car il va falloir soit dessouder complètement sa barrette de broches, soit la bricoler enlevant le plastique autour des broches. Pour cela, commencez par couper le plastique entre

les broches (**fig. 6**), puis dégagéz les morceaux un à un avec la pointe d'un tournevis (**fig. 7**). Si tout va bien, les broches sont alors dénudées (**fig. 8**).

On peut alors embrocher l'afficheur à plat dans la position prévue, suivie des barrettes à broches, des barrettes coudées et des borniers à vis. Cela termine l'assemblage, sauf le vrai bouton, bien sûr (**fig. 9**).

Interlude: une antenne 868 MHz maison

Pour émettre et recevoir à 868 MHz, le bouton LoRa doit être équipé d'une antenne. Une antenne quart d'onde 'Marconi' en fil de cuivre émaillé de 1 mm

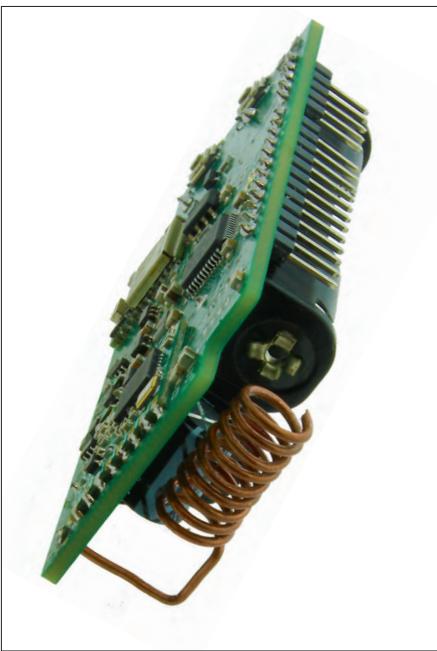


Figure 10 : Le point chaud de l'antenne quart d'onde 868 MHz maison est soudé sur la pastille ANT de la carte (ici, un des premiers prototypes). Remarquez la forme hélicoïdale de l'antenne.

est une bonne solution, peu coûteuse. Avec un bon plan de masse, le fil doit avoir une longueur théorique dans le vide de 8,635 cm, soit 8,2 cm en pratique. Cela est dû au coefficient de célérité de l'air, qui est de 0,95 [4]. Si l'espace disponible le permet, le fil peut être droit, équivalent à une antenne fouet, ou bobiné avec 1 mm entre les spires.

Dans les deux cas, le fil est soudé sur la pastille ANT. (fig. 10).

L'interrupteur esclave LoRa pour courant alternatif

Comme exemple d'organe de sortie asservi par le nœud LoRa d'Elektor, aussi à l'écoute du bouton LoRa, nous avons un interrupteur de puissance pour appareils alimentés par le secteur, jusqu'à 1 kW. L'état de l'interrupteur (ouvert/fermé) est lu par le nœud LoRa à travers un coupleur optique. La carte embarque aussi un convertisseur 100-230 V~ - 5V=, pour l'alimentation directe par le secteur du nœud LoRa et de l'interrupteur esclave.

La véritable intelligence de l'interrupteur LoRa esclave réside dans le matériel et le logiciel du nœud LoRa, lequel se présente sous la forme d'une carte emboîtable. L'interrupteur est toutefois doté des fonctions suivantes :

- rail d'alimentation 5 V=
- interrupteur de puissance 115-230 V~, 1 kW max.
- retour de l'état de l'interrupteur
- bouton-poussoir local marche/arrêt

Voyons les composants principaux de l'interrupteur LoRa esclave et leur assemblage sur le schéma (fig. 11).

Alimentation. Solution compacte : Un convertisseur alternatif/continu 5 V/2 W de chez *Mean Well*, sûr et relativement

bon marché. La documentation ne prescrit aucun fusible côté alternatif, mais nous avons ajouté F1 par précaution. La résistance NTC (à coefficient de température négatif) R1 limite la pointe de courant au démarrage, qui peut être élevée pour ce genre d'appareil. C1, C2 et le filtre de mode commun L1 en sortie du convertisseur peuvent paraître exagérés, parce que les affirmations de *Mean Well* au sujet de l'élimination des perturbations électromagnétiques ne nous ont pas convaincus.

Commutation de charge. Les relais bistables de la famille G5RL-K1 de *Omron* ont deux bobines, l'une pour l'ouverture, l'autre pour la fermeture. Une seule impulsion sur l'une d'entre elles suffit pour faire changer le relais d'état. Les contacts supportent 16 A, mais pour éviter de faire fondre les pistes du circuit imprimé le fusible F2 ne laissera passer que 5 A .

Retour de l'état du relais. En théorie, le µC sur la carte du nœud LoRa devrait garder en mémoire la dernière commande envoyée au relais et savoir s'il est ouvert ou fermé. Pour être sûr, un retour de son état n'est pas un luxe, ne serait-ce que parce que l'état d'un relais bistable est indéterminé après l'initialisation. Le circuit autour de l'optocoupleur IC1 n'indique que l'état de RE1 et la présence du secteur sur le bornier K2. Il ne dit rien sur la charge ni sur la puissance consommée, dont il faut s'assurer par ailleurs.

Commande manuelle. Le but est de commuter une charge à (grande) distance en appuyant sur un bouton, mais si vous êtes à proximité du récepteur, il est souhaitable de pouvoir la commuter manuellement sans avoir besoin du bouton LoRa, ni du nœud LoRa. De plus, en cas de perte de la connexion LoRa, quelle qu'en soit la raison, cette fonction viendra à point.

Assemblage et installation. Le soudage de la carte pose peu de problèmes. Sauf le filtre de mode commun et (peut-être) le connecteur K3, c'est une promenade de santé avec rien que des composants traversants. Les pastilles pour K3 sont de bonne taille et même L1 ne posera pas de problème.

K3 n'utilise que sept des 16 broches sur la carte. Il a deux options, décrites plus loin, qui toutes les deux supposent

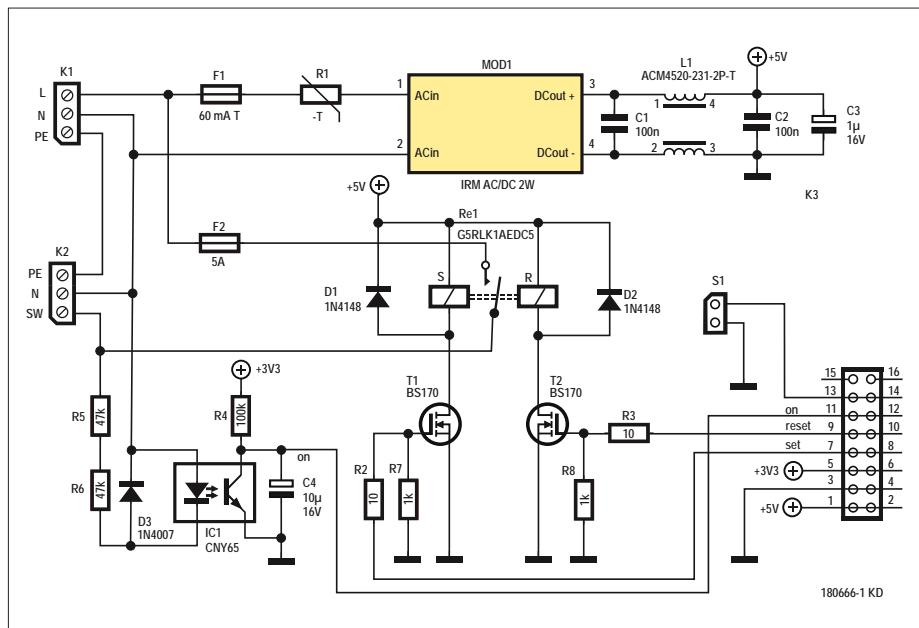


Figure 11 : Schéma de l'interrupteur LoRa. Ce n'est pas un élément passif : il renvoie l'état du relais au nœud LoRa.



LISTE DES COMPOSANTS

Pour la plateforme expérimentale LoRaWAN

Note: 2 exemplaires requis,
1 pour l'interrupteur, 1 pour le bouton

Résistances

R1 = 100 kΩ, couche épaisse, 5%, 0,1 W, 150 V
 R2, R10 = 680 Ω, couche épaisse, 5%, 0,1 W, 150 V
 R3, R8, R9 = 4,7 kΩ, couche épaisse, 5%, 0,1 W, 150 V
 R4, R5, R11 = 10 kΩ, couche épaisse, 5%, 0,1 W, 150 V
 R6, R7 = 150 kΩ, couche épaisse, 5%, 0,1 W, 150 V

Condensateurs

C1, C5, C8, C14, C15 = 4,7 µF, 16 V, X7R, SMD 0805
 C2, C3, C7, C9, C10, C11, C12, C17, C18 = 100 nF, 50 V, X7R, SMD 0805
 C4 = voir texte
 C6 = 470 nF, 25 V, X7R, SMD 0805
 C16 = omis
 C20 = omis
 C19, C21 = 100 µF, 10 V, SMD 1206 [3216 Metric], ± 20%, X5R

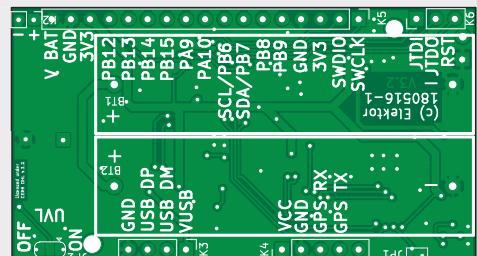
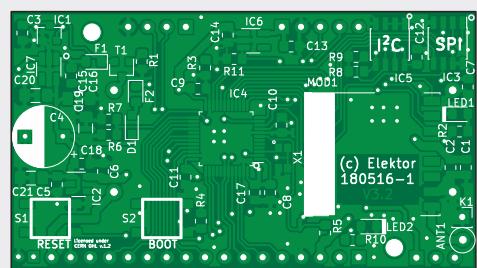
Semi-conducteurs

D1 = PMEG3010EJ, 115, diode, 30 V, 1 A

LED1 = rouge, faible consommation, SMD 0805
 LED2 = verte, faible consommation SMD 0805
 T1 = SI2347DS, MOSFET, p-channel, 5 A, 30 V, 0,033 Ω
 IC1 = TLV809K33DBVR, surveillance de tension
 IC2 = TLV75533PDBVR, 3V3 LDO, 500 mA, low-I_Q, SOT-23-5
 IC3 = (facultatif) espace pour AT25SF081-SSHD-T 8 Mbit flash
 IC4 = microcontrôleur STM32F072C8T6TR ARM Cortex-M0
 IC5 = (facultatif) espace pour ATECC608A-SSHDA-B, CryptoAuthentication
 IC6, IC7 = MAX40200AUK+T contrôleur 'diode idéale', 1 voie, 1 A
 MOD1 = RFM95W-868S2 LoRa transceiver, e-choppe Elektor SKU 18715

Divers

X1 = 32,768 kHz module oscillateur, SMD, 3,2 mm x 1,5 mm (Abracon ASH7KW-32.768KHZ-L-T)
 S1, S2 = bouton-poussoir à retour tactile, 12 V, 50 mA (Multicomp TM-553I-Q-T/R)
 Bt1, Bt2 = (facultatif) espace pour support de batteries AAA à broches pour circuit imprimé (Multicomp MP000341)
 F1, F2 = 500 mA PPTC fusible réarmable (Bel Fuse 0ZCK0050FF2E)
 K1 = broche pour circuit imprimé



K2 = connecteur à 2 broches

K3 = connecteur à 4 broches SIL

K4 = connecteur à 5 broches

K5 = connecteur à 17 broches

K6 = connecteur à 3 broches

K5, K6, K3, K4 = connecteur à 40 broches SIL

ANT1 = antenne en fil émaillé 1 mm,

longueur 8,2 cm

circuit imprimé 180516 V3.1

de l'e-choppe Elektor



LISTE DES COMPOSANTS

Liste des composants

Pour la carte interrupteur LoRa

Résistances

R1 = NTC, 120 Ω, Epcos type B57236S0121M000
 R2, R3 = 10 Ω, carbone, 5%, 0,25 W, 250 V
 R4 = 100 kΩ, carbone, 5%, 0,25 W, 250 V
 R5, R6 = 47 kΩ, carbone, 5%, 0,25 W, 250 V
 R7, R8 = 1 kΩ, carbone, 5%, 0,25 W, 250 V

Inductance

L1 = ACM4520V-231-2P-T filtre de mode commun (Farnell # 2455201)

Condensateurs

C1, C2 = 100 nF, 50 V, X7R, pas 5,08 mm
 C3 = 1 µF, 16 V, radial, 5 mm
 C4 = 10 µF, 16 V, radial, 5 mm

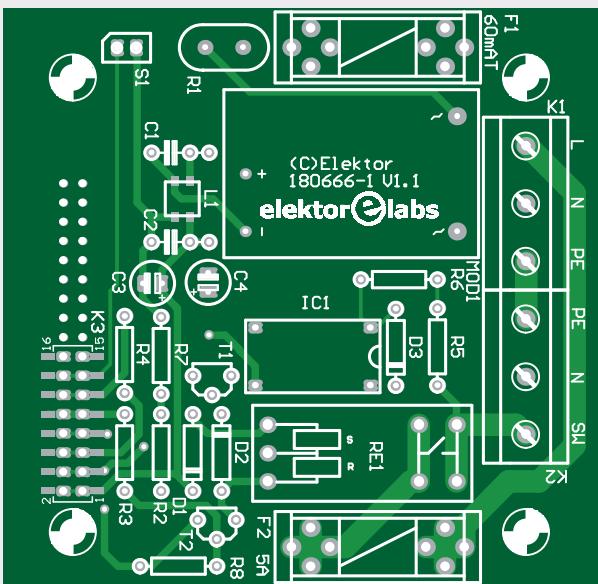
Semi-conducteurs

D1, D2 = 1N4148
 D3 = 1N4007
 T1, T2 = BS170

IC1 = CNY65 optocoupleur

Divers

RE1 = G5RLK1AEDC5 relais de puissance bistable, à deux bobines, 5 V, 16 A, SPST (Omron)
 K1, K2 = bornier à vis à 3 points, pas 7,62 mm, 500 V
 K3 = connecteur femelle à 2 rangées, 16 contacts, pas 2,54 mm, série WR-PHD, type CMS (Würth # 610316243021)
 S1 = bouton-poussoir, classe IP68 (Alcoswitch PB6B2FM3M1CAL07)
 MOD1 = alimentation secteur encartable 5 V, 200 mA (Mean Well IRM-02-5)
 F1, F2 = porte-fusible encartable 5x20 mm
 F1 = fusible 60 mA(T, 20 mm
 F2 = fusible 5AT, 20 mm



circuit imprimé 180666-1 V1.1

e-choppe Elektor

coffret HAMMOND 1591-ATBU (Ice Blue)



LISTE DES COMPOSANTS

Pour la carte bouton LoRa

Résistances

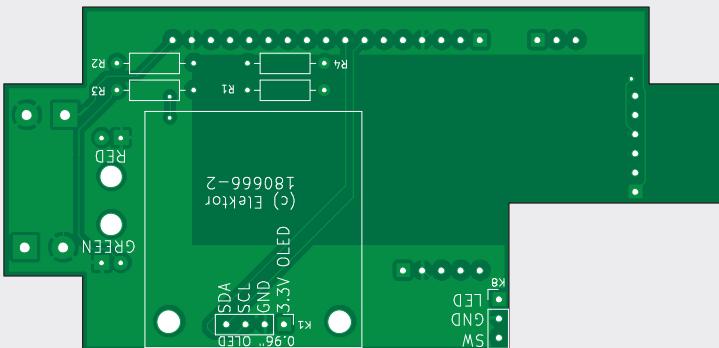
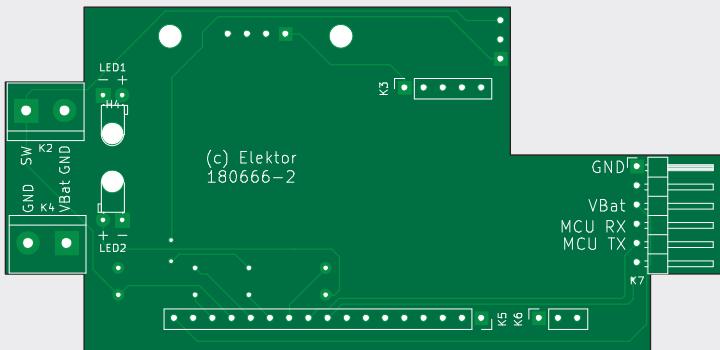
R1 = 270 Ω , couche carbone, 5%, 0,25 W,
250 V
R2 = 10 k Ω , couche carbone, 5%, 0,25 W,
250 V
R3, R4 = 680 Ω , couche carbone, 5%,
0,25 W, 250 V

Semi-conducteurs

LED2 = rouge, 3 mm
LED1 = verte, 3 mm

Divers

- K2, K4 = bornier à vis à 2 bornes,
pas 5,08 mm, 630 V
- K3 = barrette sécable à 1 rangée
de 5 broches verticales
- K5 = barrette sécable à 1 rangée
de 17 broches verticales (voir texte)
- K6 = barrette sécable à 1 rangée
de 3 broches verticales
- interrupteur à LED bleue intégrée
- afficheur OLED bleu 0,96" I²C, 4 broches
(facultatif)
- circuit imprimé 180666-2 V1.2
e-choppe Elektor



coffret Spelsberg
type TK PS 94 x 94 x 57 mm, IP66

que la carte du nœud LoRa possède en position K5 un connecteur coudé au pas de 2,54 mm. Les sept broches des positions 11 à 17 (voir la carte 180516-1) suffisent, mais K3 peut aussi être une barrette de 17 broches (tronquée) occupant la totalité de K5.

Le connecteur (*Würth Elektronik*) identifié K3 dans la liste des composants est moins facile à trouver qu'une barrette. Nous l'avons néanmoins utilisé à cause de son profil bas grâce auquel deux cartes, montées à angle droit, tiennent

dans le coffret choisi. Tant pis pour la deuxième rangée de contacts inutilisée, faute de connecteur à profil assez bas avec une seule rangée.

L'espace sur la carte pour K3 permet d'utiliser aussi des barrettes à broches traversantes avec un coffret plus profond.

Vous pouvez aussi vous passer de K3 et souder directement les broches coudées du nœud LoRa sur le circuit imprimé, mais il sera impossible dès lors de séparer les deux cartes.

Le bouton-poussoir S1 est monté sur le coffret et relié au circuit imprimé par deux fils.

Installation du logiciel

Le paquet de logiciel complet du nœud LoRa d'Elektor est disponible à l'endroit prévu [6]. Le fichier d'archive contient aussi une procédure d'installation omise ici faute de place.

Pour développer du logiciel pour la carte nœud LoRa vous pouvez choisir entre la méthode rude de la programmation en C et l'utilisation du STMCubeIDE, ou

Liens

- [1] Livre KiCAD Like a Pro : www.elektor.com/kicad-like-a-pro
 - [2] Module GPS de l'e-choppe Elektor : www.elektor.com/open-smart-gps-serial-gps-module-for-arduino-apm2-5-flight-control
 - [3] Projet stm32duino : https://github.com/stm32duino/Arduino_Core_STM32
 - [4] Coefficient de célérité : https://en.wikipedia.org/wiki/Velocity_factor
 - [5] Programmateur STM32Cube : www.st.com/en/development-tools/stm32cubeprog.html
 - [6] Logiciel du projet : www.electormagazine.com/180666-01
 - [7] Ajouter des cartes STM32 : <https://github.com/stm32duino/wiki/wiki/Getting-Started>

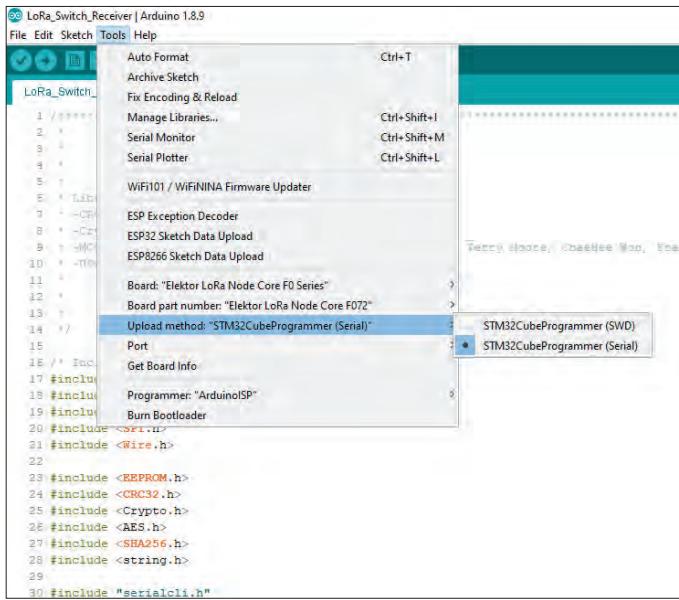


Figure 12 : Observez et copiez ces réglages de configuration de la carte.

```
boards.txt - Notepad
File Edit Format View Help
Midatronics.menu.rtlb.full.build.flags.ldspecs=


#####
# Elektor LoRa Node Core F0 Series

ElektorF0.name=Elektor LoRa Node Core F0
ElektorF0.build.core=arduino
ElektorF0.build.board=ElektorF0
ElektorF0.build.mcu=cortex-m0
ElektorF0.build.series=STM32F0xx
ElektorF0.build.cmsis_lib_gcc=arm_cortexM0_math
ElektorF0.build.extra_flags=-D{build.product_line} {build.xSerial}

# ELEKTOR_LORA_NODE_CORE_F072xC
ElektorF0.menu.pnum.ELEKTOR_LORA_NODE_CORE_F072xC=Elektor LoRa Node Core F072
ElektorF0.menu.pnum.ELEKTOR_LORA_NODE_CORE_F072xC.upload.maximum_data_size=16384
ElektorF0.menu.pnum.ELEKTOR_LORA_NODE_CORE_F072xC.upload.maximum_size=131072
ElektorF0.menu.pnum.ELEKTOR_LORA_NODE_CORE_F072xC.build.board=ELEKTOR_LORA_NODE_CORE_F072xC
ElektorF0.menu.pnum.ELEKTOR_LORA_NODE_CORE_F072xC.build.product_line=STM32F072xB
ElektorF0.menu.pnum.ELEKTOR_LORA_NODE_CORE_F072xC.build.variant=ELEKTOR_LORA_NODE_CORE_F072xC

# Upload menu
ElektorF0.menu.upload_method.swdMethod=STM32CubeProgrammer (SMD)
ElektorF0.menu.upload_method.swdMethod.upload.protocol=0
ElektorF0.menu.upload_method.swdMethod.upload.options=g
ElektorF0.menu.upload_method.swdMethod.upload.tool=stm32CubeProg

ElektorF0.menu.upload_method.serialMethod=STM32CubeProgrammer (Serial)
ElektorF0.menu.upload_method.serialMethod.upload.protocol=1
ElektorF0.menu.upload_method.serialMethod.upload.options={serial.port_file} -s
ElektorF0.menu.upload_method.serialMethod.upload.tool=stm32CubeProg
```

Figure 13 : Elektor a obtenu l'approbation du noeud LoRa (sur la *LoRaWAN Node Experimental Platform*) et son admission dans la famille STM32Arduino. Le logiciel du projet est bon pour le service, nos lecteurs n'auront pas à se préoccuper pas de cette procédure administrative.

bien vous faciliter la tâche avec l'EDI Arduino. Pour cette dernière, il vous faut télécharger l'EDI Arduino ainsi que le noyau STM32 Arduino. Dans l'EDI Arduino, ajoutez l'URL suivant dans le gestionnaire de cartes supplémentaires (Fichier → Préférences → Paramètres) : https://github.com/stm32duino/BoardManagerFiles/raw/master/STM32/package_stm_index.json et installez les cartes STM2 en suivant les instructions [7]. Vous avez maintenant le noyau de STM32, mais pour le noeud LoRa d'Elektor, il faut encore ajouter quelques petites choses à votre système de fichiers local. Sous Windows, allez à :

```
%localappdata%\Arduino15\packages\STM32\hardware\stm32\1.7.0
```

et ajoutez à la fin de «Boards.txt» le contenu du fichier *Add_To_Board.txt* que nous avons préparé pour vous. Copiez également le répertoire :

ELEKTOR_LORA_NODE_CORE_F072xC

dans le répertoire «Versions». À ce stade, vous êtes opérationnel et prêt à compiler le logiciel pour la carte. À la fin, il vous faudra définir la configuration de la carte sur la **figure 12**. Maintenant, vous êtes prêt à compiler du logiciel pour la carte. Pour effectuer

un téléchargement vers la carte, il faut installer le STM32CubeProgrammer [5]. Si vous voulez maintenant télécharger du code, mettez la carte en mode *bootloader* en appuyant sur le bouton *Boot* puis sur le bouton *Reset*. Vous voilà prêt pour un nouveau téléchargement de progiciel vers la carte.

À votre service

Au début de cet article, nous annoncions que cette histoire serait racontée dans un ordre à peu près chronologique. Depuis la mi-décembre 2019, les définitions de la carte de la plateforme expérimentale LoRaWAN d'Elektor ont été intégrées

au dépôt Git officiel STM32Arduino, en respectant les conditions assez strictes fixées par le comité *STM32duino* (**fig. 13**, merci, les gars !).

En conséquence, vous n'avez plus besoin de faire vous-même aucune des modifs décrites ci-dessus, vous n'avez plus qu'à choisir la carte noeud LoRa d'Elektor dans la liste des cartes approuvées.

C'est parti !

Gageons qu'on entendra encore parler ici du noeud LoRa d'Elektor. ▶

(180666-02 Version française : Helmut Muller)

WWW.ELEKTOR.FR

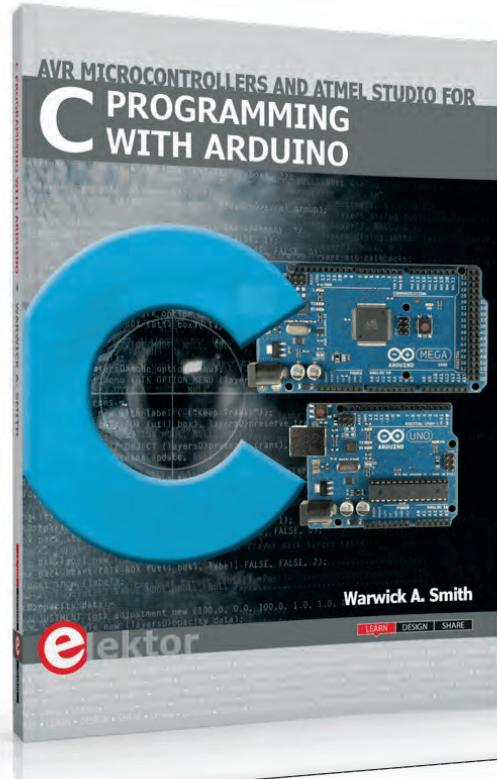
- Nœud LoRa d'Elektor, 180516 V3.1
www.elektor.fr/180516-1
- Carte Bouton LoRa, 180666-2 v 1.2
www.elektor.fr/180666-2
- Carte Interrupteur de courant alternatif LoRa, 180666 1V 1.1.
www.elektor.fr/180666-1
- RFM95W-868S2 Émetteur-récepteur LoRa
www.elektor.fr/18715
- Livre KiCAD Like a Pro
www.elektor.fr/kicad-like-a-pro
- Module GPS (facultatif)
www.elektor.fr/open-smart-gps-serial-gps-module-for-arduino-apm2-5-flight-control



mises au point & mises à jour corrections – questions – réponses

Clemens Valens (Elektor Labs)

Mises à jour et compléments d'informations 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.



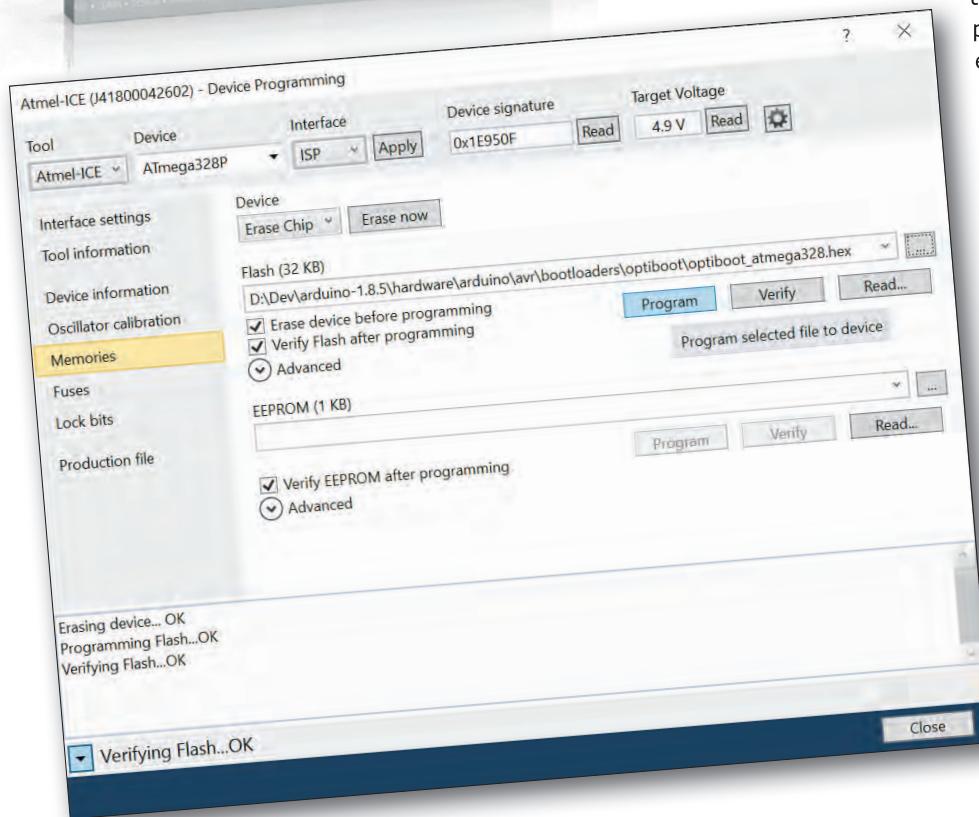
Arduino Uno : quand l'amorçage capote

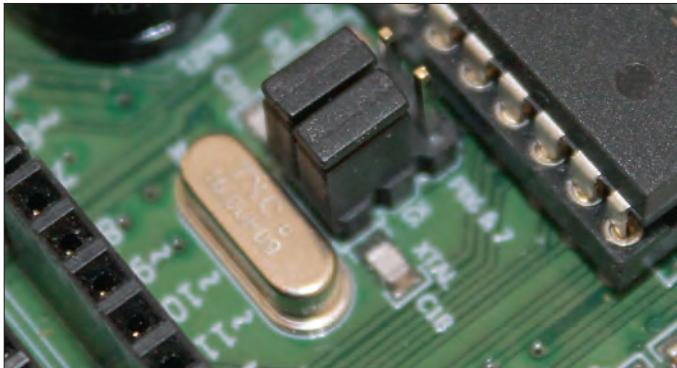
Q : J'ai acheté le livre *C Programming with Arduino* d'Elektor et m'y suis mis avec enthousiasme. J'ai aussi fait les frais d'un programmeur Atmel ICE de Microchip qui fonctionne bien avec Atmel Studio 7.0 (AS7), l'environnement (IDE) utilisé dans le livre. J'ai constaté que l'Arduino Uno ne peut plus être programmé par l'IDE Arduino. J'ai essayé un autre Uno qui lui fonctionne toujours avec l'IDE Arduino. Reconnecter cette carte, redémarrer l'ordinateur ou le logiciel ni aucune autre manipulation similaire n'a donné de résultat. J'ai cherché la solution en vain sur plusieurs forums.

R : C'est le chargeur d'amorce (dit *bootloader*) Arduino qui transforme un microcontrôleur (μ C) quelconque en μ C Arduino. Ce petit logiciel réside à demeure dans la mémoire de l'unité centrale et il est capable de charger dans cette même mémoire un programme d'application, reçu la plupart du temps via une connexion série. Et cela sans s'écraser lui-même ! L'IDE Arduino connaît évidemment ce *bootloader* et utilise le port série (via USB) pour programmer la carte Arduino.

En revanche, Atmel Studio ne connaît ni Arduino ni son chargeur d'amorce. C'est pourquoi il faut, pour inscrire un programme dans la mémoire du μ C, un adaptateur de programmation comme l'ICE Atmel. Celui-ci se connecte à l'interface de programmation interne du μ C (ISP) pour accéder directement à sa mémoire. En l'absence de précautions adéquates, cette opération érase le contenu de la mémoire, y compris le chargeur d'amorce. Voilà pourquoi l'Arduino Uno a cessé de fonctionner avec l'IDE Arduino : son *bootloader* a disparu !

Pour reprogrammer le chargeur d'amorce d'Arduino, on peut utiliser AS7 avec Atmel ICE. Le fichier nécessaire est [\[Arduino\]\hardware\arduino\avr\bootloaders\optiboot\optiboot_atmega328.hex](#) où [\[Arduino\]](#) est le dossier contenant le fichier [arduino.exe](#). Une fois dans AS7, ouvrez 'Device Programming' (Maj-Ctrl-P). Puis, dans l'onglet *Memories* de la section *Flash*, cliquez sur le bouton *Browse for file* et naviguez jusqu'au fichier HEX du *bootloader*. Cliquez sur *Program* pour faire écrire le logiciel d'amorçage dans le μ C. Il y a plusieurs façons de configurer AS7 pour Arduino afin d'éviter de supprimer le logiciel d'amorçage chaque fois que vous cliquez sur *Program*. Ces méthodes vont de l'installation





La carte Playground AVR avec un quartz de 16 MHz.

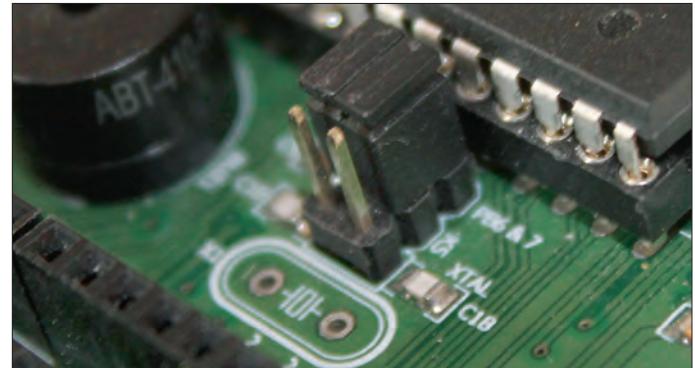
d'extensions pour AS7 à l'adaptation des paramètres du *linker*. Vous trouverez sur l'internet la méthode qui vous conviendra le mieux.

Q : Est-il possible d'utiliser l'ATmega328 d'une carte Arduino Uno avec la carte Playground AVR et *vice versa* ?

R : L'Arduino Uno et la carte Playground AVR utilisent le même µC ATmega328, mais vous ne pouvez pas simplement les interchanger. Cela est dû à leur oscillateur d'horloge. Celui de l'Arduino Uno a une horloge de 16 MHz asservie par son quartz de 16 MHz. Sur la carte Playground AVR, c'est l'oscillateur RC interne de 8 MHz du µC qui bat la mesure.

Vous pouvez remplacer le µC d'une carte Uno par un µC de Playground AVR. L'Uno fonctionnera alors comme un Playground AVR fonctionnant à 8 MHz. Cependant, l'inverse, c'est-à-dire remplacer le µC d'une Playground par celui d'une Uno, n'est possible que lorsque l'AVR Playground est équipé d'un quartz de 16 MHz (X1) et que les cavaliers de K5 sont sur les broches 1-3 et 2-4. Une fois ces conditions remplies, le Playground AVR fonctionnera comme un Uno.

La programmation du chargeur d'amorce dans le µC d'une carte compatible Arduino tient compte de la fréquence d'horloge du µC afin de configurer la vitesse du port série utilisé pour le



AVR Playground running from its internal 8-MHz RC-oscillator.

téléchargement des croquis : donc, si la fréquence d'horloge change, la vitesse du port série change aussi. L'IDE Arduino ne permet malheureusement pas à l'utilisateur de spécifier lui-même le débit du port série pour le téléchargement des croquis. Lorsque vous sélectionnez une carte Arduino Uno dans l'IDE Arduino, l'IDE suppose qu'elle fonctionne à 16 MHz et nécessite un débit de téléchargement de 115 200 bauds. Avec un Uno qui n'est pas cadencé à 16 MHz, le téléchargement d'un croquis échouera donc. De même, un Playground AVR doit, pour l'IDE, fonctionner à 8 MHz avec un débit de téléchargement de 57 600 bauds.

La simplicité de l'IDE Arduino est appréciée par la plupart de ses utilisateurs, mais elle implique des restrictions. L'utilisateur aventureux tentera de modifier le fichier `boards.txt` où sont réunies toutes les informations sur les cartes. Sachez qu'il peut y avoir plus d'un de ces fichiers dans votre installation de l'EDI Arduino. Les clés à modifier sont `[Board].upload.speed` et `[Board].build.f_cpu` (où `[Board]` est le nom de la carte). L'IDE doit être relancé pour prendre en compte les modifications apportées à ce fichier.

www.elektormagazine.com/labs/avr-playground-129009-2

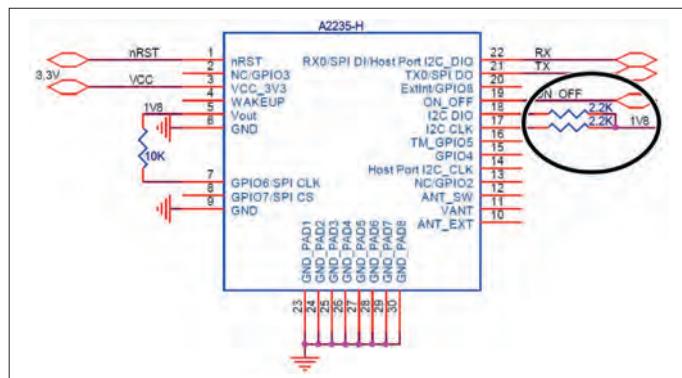
nouveau GPS pour une nouvelle horloge Nixie précise

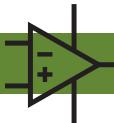
Le module GPS Maestro A2035-H utilisé dans ce projet n'est plus produit. Cependant, Lantronix, qui a acquis Maestro (été 2019), propose le A2235-H, plus récent, qui, bien que plus petit, s'adapte à notre circuit imprimé. Il suffit d'ajouter deux résistances de polarisation de 2,2 kΩ entre ses broches I²C et sa broche 1,8 V. Pour éviter tout court-circuit entre le blindage

du module et un via en dessous de celui-ci, intercalez une isolation (Kapton).

www.elektormagazine.com/labs/150189-6-digit-nixie-clock

190379-B-02





électronique analogique

Étude de cas n° 1 - section 2 : Préamplificateur optimisé pour les microphones MEMS

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. En effet, l'analogique n'est pas de la magie noire. Dans la première section de cette étude de cas, nous avons parlé du microphone MEMS utilisé pour la première fois dans un projet d'Elektor appelé *Bat Detector^{PLUS}*.

Approfondissons avec l'ampli de micro plutôt raffiné de ce projet.

Cet article d'une série sur la *conception électronique analogique* fournit une évaluation documentée et commentée du préamplificateur de microphone original utilisé dans le détecteur de chauves-souris Bat Detector^{PLUS} [1]. L'auteur discute des solutions pour l'optimiser pour le microphone MEMS examiné dans la section 1 de cette étude de cas [2]. Pour bien suivre, il convient de se référer au schéma de circuit du préamplificateur de microphone mentionné ; c'est la partie surlignée de la

fig. 1. Nous reproduisons délibérément le schéma complet du Bat Detector^{PLUS} pour bien montrer comment le préamplificateur est :

- relié au microphone (K1) ;
- chargé (par IC4A) ;
- alimenté (symétriquement sous $\pm 4,5$ V par IC1).

Finalement, l'amplificateur opérationnel de faible puissance TL062 fait encore bonne figure avec ses 200 μ A de courant d'alimentation par amplificateur au repos. Ce n'est pas mal comparé à de nombreuses versions de faible puissance actuelles, disponibles presque exclusivement comme CMS (toujours plus minuscules !), mais il y en a maintenant de bien meilleurs. Son produit gain-bande passante (GBW) de 1 MHz est la pierre d'achoppement de ce circuit, puisqu'il impose un facteur d'amplification de 1845. Si dans le gain total nous incluons les 400 Ω du micro MEMS, la bande passante avec un seul ampli op serait de 540 Hz. Pour élargir cette bande passante, il faut plusieurs amplificateurs successifs, trois exactement, chacun avec moins d'amplification et donc plus de bande passante. Cela implique une augmentation de la consommation d'énergie dont on peut se demander si elle est justifiée si on la compare à celle d'un ampli op beaucoup plus rapide. Il existe des amplis ultrarapides comme l'OPA2889 à deux étages avec un produit gain-bande passante de 75 MHz et un facteur 20 d'amplification pour une consommation de 0,92 mA pour les deux amplificateurs.

Mais revenons au nombre d'amplis op nécessaires. C'est difficile à calculer, car la bande passante par amplificateur peut ne pas correspondre à celle d'un filtre passe-bas de premier

ordre. Plus il y aura d'amplificateurs en série, plus la bande passante sera réduite comparée à celle d'un ampli unique. Par commodité, considérons que la réponse est celle d'un réseau RC de premier ordre. Cela vaut aussi pour des étages de filtrage similaires déterminant la limite inférieure ; plus d'étages signifie une fréquence de coupure plus haute pour la limite inférieure de la bande. Pour un filtre RC passe-haut du premier ordre, la fonction de transfert est :

$$j\omega RC / (1 + j\omega RC) \quad (1)$$

En raison de la rotation des phases, la distribution réelle de la tension est :

$$\omega RC / \sqrt{1 + (\omega RC)^2} \quad (2)$$

Pour le point de coupure, cela équivaut à $1/\sqrt{2}$. C'est le fameux point -3 dB point, que les tatillons noteront :

$$20 \log 1/\sqrt{2} = -3,0103$$

Nous voulons la largeur de bande réelle lorsque plusieurs étages seront en série. Au lieu de $1/\sqrt{2}$, on prend le facteur x . En réorganisant, on obtient :

$$\omega RC = x / \sqrt{1 - x^2} \quad (3)$$

Par exemple, pour un point de -3 dB, entrez la valeur $1/\sqrt{2}$ pour x , ce qui donne 1 exactement et la formule standard bien connue pour la fréquence de coupure à -3 dB :

$$f = 1 / (2nRC) \quad (4)$$

Nous voulons le nouveau point de coupure s'il est connu d'un étage. Prenons trois étages comme dans le circuit. Nous devrions trouver la fréquence à laquelle l'atténuation est de

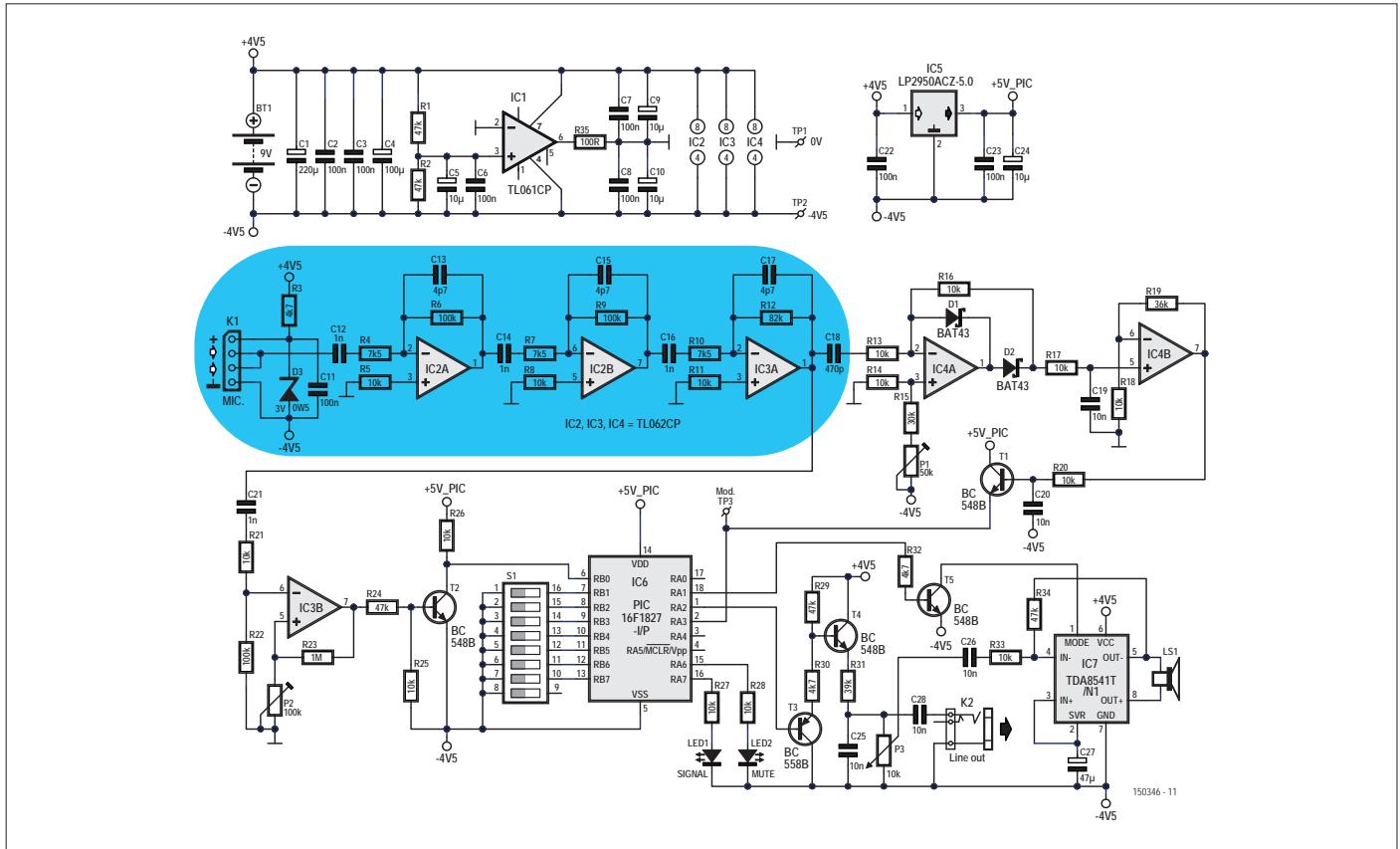


Figure 1 : Circuit complet du détecteur Bat Detector^{PLUS} avec le préamplificateur de microphone à 3 étages en surbrillance.

1 dB. Par commodité, nous ne prenons pas $-3,0103/3$ mais seulement -1 . De $20\log x = -1$ il s'ensuit que :

$$x = 10^{(-1/20)} \quad (5)$$

L'atténuation à trouver pour le point à -1 dB s'établit alors à 0,891251.

En l'entrant dans l'équation 3, ωRC est égal à 1,96523. Par conséquent, pour trois étages égaux, le nouveau point de coupure sera presque deux fois plus élevé. Dans le cas d'un filtre passe-bas, le point de coupure est alors presque deux fois plus bas. Pour n étages, on aura :

$$x = 10^{((-3/n)/20)} \quad (6)$$

Dans le cas du TL062, un gain de 13,3 fois (c'est-à-dire $100/7,5$) pour l'étage médian, signifie un point de coupure à 75 kHz. Le condensateur supplémentaire de 4,7 pF dans la boucle de rétroaction abaisse encore la fréquence de coupure à environ 60 kHz par étage et modifie la réponse en fréquence à un ordre supérieur.

D'abord simuler...

... avant de dessiner un PCB ou de construire un proto sur plaque d'essais. Ne serait-ce que pour s'assurer qu'il y a assez

étages et qu'il n'y a rien d'autre à corriger. Souvent, la simulation est plus rapide que les calculs. La simulation du préampli a montré que sa bande passante finale allait de 23 kHz à 58 kHz. Pas assez pour les chauves-souris ! Sur notre proto, la limite supérieure était même légèrement inférieure à environ 44 kHz, à cause probablement des tolérances des amplis op, et de la tension d'alimentation plus basse que celle que prévoit la fiche technique pour le GBW : soit une valeur **typique** de 1 MHz pour un TL062 ; max et min non spécifiés. La réponse de chaque étage n'est probablement pas non plus purement du premier ordre.

Combien d'étages alors...

... si on utilise ce bon vieux TL062 ? Il est recommandé de ne pas laisser la limite de l'ampli op déterminer la largeur de bande souhaitée, car le produit gain-bande passante GBW de l'ampli op peut présenter des dispersions considérables. Les effets tout aussi négatifs sur la distorsion et l'intermodulation ne posent pas vraiment de problème ici. Il est préférable que cette limite soit sensiblement plus élevée que nécessaire avant le filtrage, disons un facteur de 2, de préférence 3, et que l'on laisse un condensateur dans le circuit de rétroaction fixer la largeur de bande souhaitée.

Où cela nous mènera-t-il ? La limite supérieure de la bande passante effective de l'ampli op – et par conséquent le gain

maximal inférieur réglable par étage – a naturellement un impact sur le nombre d'étages requis. Si nous restons prudents en retenant une largeur de bande 3 fois plus étendue, l'amplification par étage ne devrait pas dépasser :

$$1 \text{ MHz} / 360 \text{ kHz} = 2,78 \text{ fois.}$$

Intuitivement, cela signifie que beaucoup d'étages seront nécessaires ! Appelons A l'amplification par étage (arrondie à 1850) et n le nombre d'étages. Nous aurons :

$$A^n = 1850 \quad (7)$$

Pour trouver n nous écrivons :

$$n = (\log 1850) / (\log A) \quad (8)$$

En substituant 2,78 à A , alors n est égal à 7,36. Dans ce cas, au moins huit étages sont nécessaires, ou deux amplis op quadruples, c'est-à-dire des TL064 avec une consommation de courant totale de 1,6 mA. Ce n'est pas très pratique et l'utilisation d'un ampli op (double) plus rapide semble être un bien meilleur choix, surtout eu égard au nombre de composants et à leur encombrement sur le circuit imprimé. L'avantage du plus grand nombre d'amplis op est un éloignement physique plus grand entre le signal de sortie relativement élevé et l'entrée de l'amplificateur, ce qui réduit le risque d'oscillations causées par un couplage parasite. Avec l'ampli op CMS rapide, les autres composants seront de préférence montés en surface également. Le couplage parasite avec le microphone n'a aucun effet, car ce composant est blindé. Au pire, l'effet peut se produire sur les connexions telles que les connecteurs et les câbles vers et depuis les circuits imprimés.

Recherché : bande passante !

Pour vérifier, la largeur de bande théorique peut être recalculée. Avec huit étages, il faut moins de gain par étage. Dans le cas de n étages, l'amplification par étage est :

$$A = \sqrt[n]{1850} \quad (9)$$

Avec huit étages, l'amplification nécessaire par étage n'est que de 2,561 fois. L'équation 6 donne 0,95775 pour x . En utilisant l'équation 3, vous pouvez calculer que le nouveau point de coupure se décale d'un facteur de 3,33. Ainsi, la nouvelle fréquence de coupure attendue, sans condensateur dans le circuit de retour, devient :

$$1000 / (2,561 \times 3,33) = 117,3 \text{ kHz.}$$

Ce n'est pas la largeur de bande souhaitée, pourtant nous sommes passés de trois à huit étages. Cela ne suffit pas pour les raisons mentionnées. Dans une simulation avec 7,5 kΩ à l'entrée et 19,2075 kΩ pour la rétroaction (pour une amplification théorique de 2,561), huit étages en série donnent une

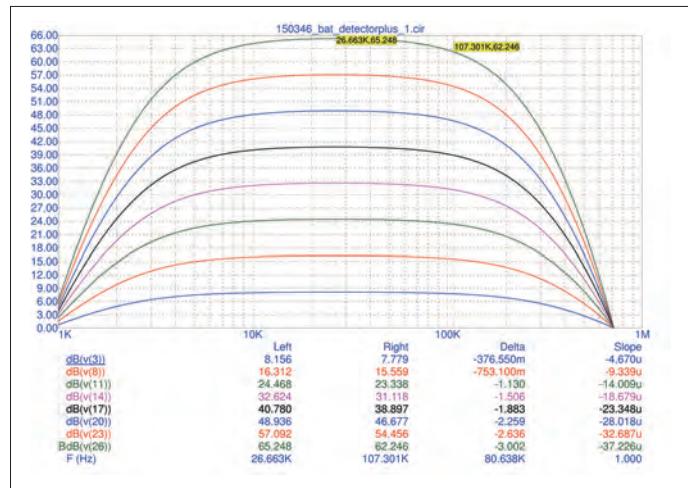


Figure 2 : Le graphique montre l'effet de huit amplificateurs identiques en série (source = 0 Ω).

bande de 107,3 kHz, comme le montre la **fig. 2**. Ce n'est pas la meilleure solution. De plus, le condensateur de couplage de 1 nF doit être ajusté ; si nous tenons vraiment à 10 kHz, cela donnerait une valeur tordue de 7 nF. Avec une valeur standard de 10 nF, la fréquence de coupure basse sera de 6,6 kHz. Une autre approche pourrait être : un ou plusieurs amplis op rapides et des filtres actifs comme amplificateur, avec p. ex. une caractéristique de Butterworth ou de Tchebychev.

Rapport signal/bruit

Pour le rapport signal/bruit, cette approche ne sera pas fameuse, car ce sont les premiers étages qui font la loi. À l'entrée, un ampli non-inverseur aurait peut-être été préférable. Il n'y aurait pratiquement pas d'atténuation en raison de la résistance d'entrée, de sorte que les résistances dans la contre-réaction du premier étage pourraient être considérablement plus faibles, ce qui est bon pour le rapport signal/bruit et la dynamique.

La résistance de sortie maximale du microphone MEMS utilisé est de 400 Ω. Avec un micro à électret, la résistance est encore plus forte. La réponse en fréquence n'a pas été adaptée au micro. La plupart des fabricants ne spécifient la réponse en fréquence qu'en dessous de 20 kHz, souvent ça ne va que jusqu'à 10 kHz, or c'est justement cette information-là qu'il nous fallait ! L'idéal serait de la mesurer soi-même, mais une précision absolue n'est pas nécessaire – on peut se contenter d'une comparaison avec la réponse connue d'une source d'ultrasons à large bande. Ne compliquons pas les choses...

À suivre

La section 3 de cette étude de cas sera publiée dans le prochain d'Elektor. Il abordera d'autres défis et solutions intégralement analogiques : la dynamique du préampli, le bruit d'entrée, les courants de polarisation de l'ampli et la modélisation des microphones à 2 et 3 broches.

Restez à l'écoute. ▶

(191230-04 VF)

Liens

- [1] détecteur de chauve-souris Bat Detector^{PLUS} : www.elektormagazine.fr/magazine/elektor-201611/39913
- [2] électronique analogique (1) : www.elektormagazine.fr/magazine/elektor-145/57082

dessine-moi un bouton pour l'IdO

boutonnière n° 1 : Architecture de l'IdO

Veikko Krypczyk (Allemagne)

Sur l'internet des objets (IdO), c'est le printemps toute l'année, et tout pousse vite. Témoin l'éclosion de ce bouton IdO, programmable à volonté pour les applications à votre guise. Il lui suffit d'une connexion au réseau, le reste se passe dans le(s) nuage(s). La première partie de cet article traite de l'architecture de l'IdO basée sur le *cloud computing* sans serveur et de l'interaction entre matériel et logiciel.



Les idées de nouvelles applications pour l'internet des objets (IdO) ne cessent de jaillir. Beaucoup restent expérimentales et ne prendront probablement pas racine dans le monde réel de l'IdO de si tôt. La vision du réfrigérateur à remplissage automatique est difficilement compatible avec le plaisir que nous éprouvons à choisir nos aliments en fonction du goût du jour et du marché. Le *frigIdO* attendra. À la frontière entre application sérieuse et activité ludique, on trouve les boutons IdO qui pourraient par exemple :

- allumer ou éteindre des appareils,
- commander des fonctions,
- passer des coups de fil,
- compter quelque chose.

Les applications possibles pour ce *button-internet* sont multiples, aussi

bien dans le secteur privé qu'ailleurs. L'idée est d'utiliser du matériel sous la forme d'un bouton pour envoyer un signal sur la toile. Peu importe, pour commencer, où le signal est envoyé, ni quelles seront les actions associées ni les systèmes connectés, ni encore le nombre de boutons. Abordons ce sujet comme une occasion de découvrir et comprendre l'architecture moderne de l'IdO. Nous commencerons par l'interaction des composants et explorerons les fonctions et les possibilités des services de traitement de données dématérialisés dans le nuage. Sur l'IdO, les expériences il est facile de se livrer à des expériences, ce qui s'applique également à l'utilisation des services de *cloud computing*. Avec un matériel (minimal),

quelques logiciels et quelques configurations, nous ferons nos premiers pas dans la deuxième partie de l'article et créerons ainsi les conditions pour tester nos propres idées autour de cette variété particulière d'IdO.

Architecture de l'IdO

L'architecture d'une solution IdO moderne peut être imaginée comme le montre la **figure 1**, indépendamment de l'application spécifique et du service choisi. Les appareils de la partie gauche peuvent être des capteurs ou des actionneurs, ou une combinaison des deux. Les capteurs mesurent par exemple les conditions environnementales telles que la température, surveillent l'état de systèmes ou envoient un signal lorsque

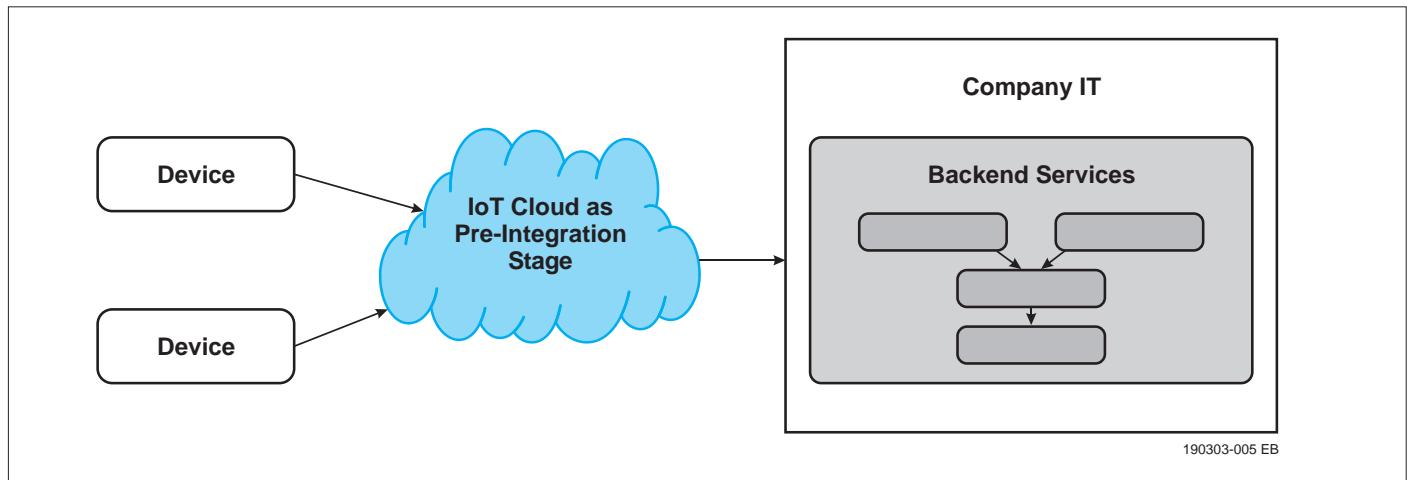


Figure 1 : Architecture de base d'une application IdO avec services d'infrastructure (back-end) dans le nuage.

l'état du système surveillé change, par exemple un chauffage central. Les actionneurs ouvrent ou ferment la vanne d'un radiateur par exemple. Dans certaines applications de l'IdO, on utilise conjointement capteurs et actionneurs pour former une boucle de contrôle action-réaction – gérée par un système d'application à distance. Dans d'autres scénarios de l'IdO, il s'agit seulement d'enregistrer ou de surveiller une condition. Avec notre simple bouton IdO, seul l'actionnement d'un bouton est surveillé.

L'information de changement d'état du bouton est transmise sur un réseau (généralement sur l'internet). Le destinataire des signaux est un service installé sur un serveur. Les services spécifiques adaptés aux exigences des applications de l'IdO sont regroupés dans ce que l'on appelle le nuage de l'IdO ou *IoT Cloud*. Le rôle de ce nuage IdO est central ; il assume le rôle de serveur et est donc le partenaire de communication des appareils IdO. Comme ici seules des fonctions spécifiques sont utilisées et non un serveur complet, on parle de *calcul sans serveur*.

Ces services fonctionnent sur le serveur d'un fournisseur, et vous n'avez donc pas à vous soucier ni de leur configuration, ni de leur administration, ni des accessoires. En cas de sollicitation accrue, p. ex. lors de l'augmentation du nombre des appels au service, des ressources supplémentaires sont fournies automatiquement pour satisfaire la demande (on parle de mise à l'échelle horizontale). En tant qu'utilisateur, vous n'avez pas à vous en

soucier. Les fonctions centrales du nuage IdO sont :

- **authentification des périphériques** : permet une variété de connexion et d'authentification des périphériques, à base p. ex. de jetons ou certificats.
- **communication sécurisée** : elle fournit un canal de communication sécurisé par lequel les données peuvent être échangées avec les dispositifs IdO.
- **transmission des données** : Certaines données peuvent être évaluées et traitées directement dans le nuage de l'IdO, d'autres doivent être transmises aux services pour un traitement ultérieur. Dans l'illustration, ces services sont conçus comme des services d'infrastructure (*backend*) d'une entreprise IT existante. Dans ce cas, il s'agirait d'une solution industrielle de l'IdO. Les applications plus simples se contentent de transmettre les données à un logiciel, pour évaluation et visualisation.

Boutons et boutonnières

Dans le cas du bouton IdO, il faut décider de la réaction à déclencher quand le bouton est pressé. Ça peut consister à enclencher ou déclencher un autre dispositif IdO, p. ex. un actionneur. En plus de ces tâches, on attend du nuage IdO qu'il soit capable de communiquer une grande variété d'appareils, et suffisamment flexible pour fournir des interfaces de programmation pour les différents systèmes. Idéalement,

ces interfaces sont mises à disposition dans différents langages cibles (par ex. C, Python, Java) via des SDK (bibliothèques) faciles à intégrer. Une communication générique via l'interface REST (voir **l'encadré**) devrait également être possible. Le nuage IdO doit également prendre en charge différents protocoles, p. ex. HTTPS, AMQP ou MQTT pour la communication avec les appareils.

Services dématérialisés de l'IdO

Les grands fournisseurs de traitement dans le nuage (*cloud computing*) proposent également des fonctions adaptées aux besoins spécifiques de l'IdO. Les acteurs principaux du marché des solutions IdO dans le nuage sont *Windows Azure*, *Amazon Web Services*, *Qloud* et *Oracle Cloud*. Le choix de la bonne plateforme n'est pas facile et dépendra des objectifs [1]. Les critères suivants doivent être pris en compte pour faire ce choix :

- **plate-forme** : quelles sont exactement les fonctions proposées, notamment la gestion des périphériques, l'authentification et l'autorisation, et le transfert de données ?
- **caractéristiques** : dont notamment les aspects de la protection des données, du cryptage et de l'interface utilisateur (tableau de bord) ;
- **connexion** : langages de programmation, SDK et interfaces pris en charge ;
- **autres critères** : tels que la communauté et la documentation, le soutien, les coûts et les modèles de tarification.

La plupart des services dématérialisés de l'IdO peuvent être testés gratuitement jusqu'à un certain niveau. Sur les sites des fournisseurs, il y a généralement des exemples documentés sous forme de code source et pour la configuration (*setup*).

Appareil IdO et connexion

La connexion entre appareils IdO et services dématérialisés se fait par l'internet public. Peu importe dans un premier temps comment ce lien est établi. En plus d'une connexion directe câblée (LAN) d'appareils fixes, la connexion peut également être sans fil via un réseau local (WLAN) ou par radio (LTE) ; tout dépend de l'application, de l'infrastructure existante, du matériel choisi et de la consommation d'énergie. Avec notre bouton IdO, nous supposons qu'il s'agit d'une intégration la plus simple possible dans un réseau local câblé local. Comme le bouton IdO doit être flexible dans son utilisation et que le boîtier doit être compact (p. ex. sous la forme d'un interrupteur pour un luminaire), les spécifications pour le matériel sont claires. Seules entrent en

ligne de compte des cartes peu encombrantes, dotées d'une connexion de réseau sans fil commode, et à faible consommation d'énergie, en particulier en mode veille. Nous reviendrons sur ces aspects dans la deuxième partie de l'article, quand nous expérimenterons avec nos premiers prototypes.

Si vous n'avez pas à vous soucier de restrictions de taille, d'alimentation électrique et de connexion sans fil au réseau, le choix de la plate-forme pour l'appareil IdO est beaucoup plus étendu et donc plus facile. Si p. ex. le bouton IdO est sédentaire, il peut être alimenté par la tension du réseau, de sorte que la consommation en veille est moins critique et une connexion LAN par câble est envisageable.

Fournisseur de services dans le nuage de l'internet des objets

Il faut maintenant sélectionner un fournisseur de services IoD dématérialisés. Le service *Azure IoT Hub* [2] est conçu par Microsoft pour une communication bidirectionnelle entre les dispositifs IdO et le portail *Azure*. Il s'agit d'une infrastructure hébergée dans le

nuage auquel une variété de dispositifs peuvent être liés. Les appareils sont intégrés via des SDK, disponibles pour différentes plateformes et différents langages de programmation. La passerelle IoD offre également des fonctions pour la gestion des appareils connectés. Chaque appareil a sa propre identité avec des données d'accès ou des certificats, chaque appareil peut être activé ou désactivé indépendamment des autres sur le tableau de bord, directement dans le navigateur.

Cette fonction est également offerte par le biais d'API pour permettre aux développeurs de gérer les dispositifs IdO dans leurs propres applications. La documentation en ligne fournit des instructions détaillées, étape par étape : Dans [3], par exemple, on verra comment connecter un Raspberry Pi à la passerelle IdO et recevoir des données du périphérique. L'utilisation de la passerelle IdO Azure est gratuite pour le prototypage ainsi que pour un usage commercial ou privé tant que l'on accepte une limitation du nombre d'appels. Le tarif *Standard* n'entraîne



Oh, aurions-nous oublié un événement ?

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

Envoyez-nous tous les détails à redaction@elektor.fr.

mars 2020

◊ **JEC World** (Salon intern. des composites)
du 03 au 05 mars à Paris Villepinte
www.jec-world.events/fr/

◊ **Salon du livre**
du 05 au 08 mars à Bruxelles
<https://flb.be>

◊ **EXPO ELETTRONICA**
le 07 mars 2020 à Faenza (Italie)
www.expoelettronica.it/

◊ **DATE** Salon de la conception et du test des circuits électroniques
du 9 au 13 mars à Grenoble
www.date-conference.com/

◊ **Salon du livre**
du 10 au 12 mars à Londres
<https://www.londonbookfair.co.uk/>

◊ **EMV** – Salon et congrès sur la compatibilité électromagnétique
du 17 au 19 mars à Cologne (Allemagne)
www.mesano.de/de/EMV/home.htm

◊ eLearning expo | I-EXPO

Salon des professionnels de l'information numérique
du 17 au 19 mars à Paris - Porte de Versailles
www.i-expo.net/
www.e-learning-expo.com

◊ IT MEETINGS

Salon B2B des réseaux, des télécoms, de la mobilité, du Cloud Computing, des datacenters et de la sécurité
du 17 au 19 mars à Cannes
www.it-meetings.fr/

◊ Rencontres de l'électronique imprimée

du 17 au 18 mars à Paris
www.rencontreselectroniqueimprimee.com

◊ Salon des Solutions MtoM et des objets communicants

Salon des systèmes embarqués & logiciels temps-réel
du 18 au 19 mars 2020 à Paris – Paris - Porte de Versailles
<http://www.embedded-mtom.com/>

◊ Salon du livre de Paris

du 20 au 23 mars 2020 à Paris
[https://www.livreparis.com/](http://www.livreparis.com/)

◊ Semaine de la presse et des médias dans l'école

du 23 au 28 mars
[https://www.clemi.fr/fr/semaine-presse-medias.html](http://www.clemi.fr/fr/semaine-presse-medias.html)

... suite à la page 65

aucun frais tant que vous n'échangez pas plus de 8 000 messages par jour avec l'appareil et que la taille d'un message ne dépasse pas 0,5 Ko.

L'offre de Google avec le service *Cloud IoT Core* va dans le même sens [4]. Il s'agit d'un service entièrement géré qui vous permet de gérer des appareils et d'échanger des données sur l'internet. *Cloud IoT Core* peut interagir avec d'autres services de Google pour le traitement dans le nuage, tels que *Google Big Data Analytics* et les services ML comme *Dataflow*, *BigQuery*, *BigTable*, *ML*, *Data Studio* ou les outils de BI des partenaires. Cela permet une évaluation, un traitement et une visualisation efficaces des données de l'IdO en temps réel. Le service supporte les protocoles courants MQTT et HTTP, ce qui assure un

haut niveau de compatibilité. L'une des fonctions essentielles de *Cloud IoT* est le gestionnaire de périphériques intégré. Les appareils gérés peuvent être paramétrés par l'intermédiaire d'un terminal, ou bien commandés par du code. Le Gestionnaire de périphériques détermine l'identité d'un périphérique et assure une authentification unique. Pour la transmission sécurisée des données, nous nous appuyons sur un cryptage dit de bout en bout. Sur le plan de la technique, le *Cloud IoT Core* est aussi un service sans serveur qui évolue horizontalement, sans latence notable, au fur et à mesure que les besoins augmentent en quantité.

Techniquement, une interface REST est utilisée, ce qui permet d'obtenir un haut degré d'indépendance par rapport aux systèmes des appareils. Le coût du

service est basé sur la consommation et est échelonné. Jusqu'à un volume de données mensuel de 250 Mo, le *Cloud IoT Core* de Google est gratuit. Ce service convient donc bien pour des expériences, le prototypage, les projets privés ou de petits projets commerciaux. Des offres similaires – mais presque exclusivement destinées aux clients industriels – sont proposées par *Oracle Cloud Services for IoT* [5], *Amazon AWS IoT Services* [6] et la plate-forme *Qloud IoT* [7].

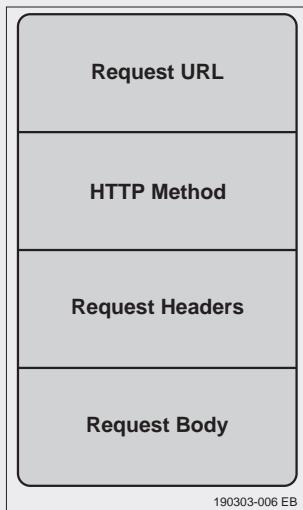
N'oublions pas les plateformes de *cloud computing* à code source ouvert proposées pour les besoins de l'IdO par *ThingsBoard* [8]. L'utilisation est gratuite si c'est vous qui installez le logiciel sur votre ordinateur. D'une certaine manière, vous créez votre propre infrastructure IdO, qui s'appelle *On Premise* : c'est

Et c'est quoi le REST ?

Quand client et serveur sont connectés via le réseau internet, l'API qui s'en charge est généralement basée sur REST [9][10]. L'acronyme vient de *REpresentational State Transfer*. Les API RESTful sont basées sur des procédures standardisées telles que HTTP/S, URI, JSON ou XML. Les principes suivants s'appliquent :

- **modèle client-serveur** : communication est basée sur un modèle client-serveur. L'objectif est une utilisation flexible et générique des services au-delà des frontières de la plate-forme.
- **apartheid** : communication toujours apartheid. Chaque requête du client vers le serveur doit être complète. Le serveur ne peut accéder aux données des requêtes précédentes.
- **mise en cache** : les clients peuvent stocker les réponses du serveur (possibilité de mise en cache). Cela permet de surmonter les pannes de courant ou d'activer temporairement un fonctionnement hors ligne. Lorsqu'une nouvelle requête intervient, les données mises en cache peuvent servir, sans attendre de nouvelle réponse du serveur.
- **uniformité** : Les services utilisent une interface uniforme, découpée du service fourni.

Une API REST est généralement mise en œuvre via http ou https. Les services sont utilisés via l'URL et les méthodes http telles que GET, POST, PUT, où le client envoie une *requête* au serveur et reçoit une *réponse*. Une requête comporte quatre volets : *endpoint*, *method*, *header* et *data*. Ici *endpoint* se compose de l'*endpoint* racine, du chemin et des paramètres



Structure d'une requête basée sur REST.

éventuels. Par exemple, <https://api.github.com> est l'*endpoint* racine de GitHub et /users/veikkoef/repos est le *chemin* vers les dépôts de l'auteur. Ce qui donne <https://api.github.com/users/veikkoef/repos>. Les paramètres peuvent suivre, p. ex. ?query1=valeur1&query2=valeur2. Pour les méthodes, on a le choix entre *GET*, *POST*, *PUT/PATCH* et *DELETE*, c'est-à-dire *AFFECTER*, *EFFECTUER*, *RECHANGER* ou *SUPPRIMER*.

GET porte sur des opérations de lecture de données. Comme les *requêtes GET* n'ont qu'un accès en lecture au serveur, elles peuvent être envoyées aussi souvent que nécessaire. *GET* est la méthode par défaut. *POST* est utilisé pour créer des opérations, c'est-à-dire pour l'enregistrement de données. *POST* n'est pas exempt d'effets secondaires. Un *appel POST* peut être utilisé pour modifier des champs dans une base de données ou pour lancer des processus sur le serveur. *PUT/PATCH* est utilisé pour les opérations de mise à jour et *DELETE* pour la suppression de données.

Le troisième élément d'une requête est l'*en-tête*. L'en-tête est utilisée pour fournir des informations au client et au serveur. Elle peut être utilisée à de nombreuses fins, par exemple pour authentifier et fournir des informations sur le contenu ultérieur.

Enfin, et comme dernier élément d'une requête, suit le corps des données (*Body*) que vous voulez transmettre au serveur. Ce corps n'est utilisé que pour les opérations *POST*, *PUT*, *PATCH* ou *DELETE*.

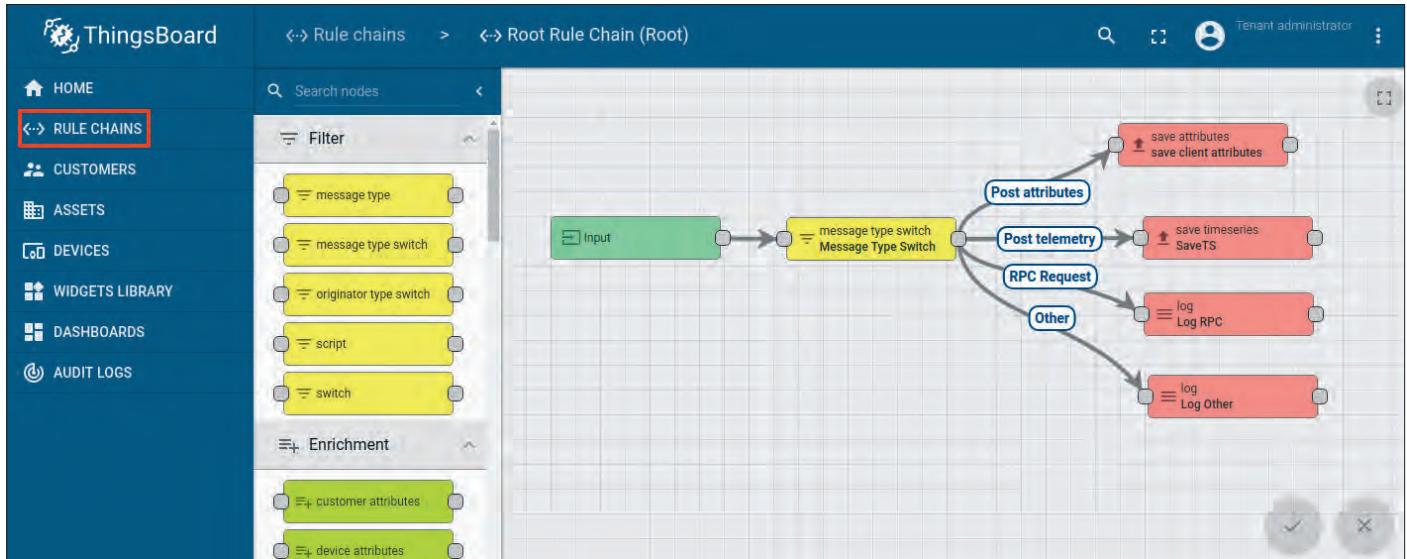


Figure 2 : Le traitement des événements peut être défini graphiquement (source : ThingsBoard).

vous qui êtes responsable de l'installation, de la maintenance et de l'exploitation. On appréciera la variété des systèmes supportés ; tout fonctionne sur différentes distributions Linux (*Ubuntu, CentOS, Red Hat*), *Windows* ou sur un Raspberry Pi 3. Ce dernier pourrait être intéressant pour les projets communautaires (Maker) et pour le prototypage.

Le logiciel de *ThingsBoard* est également proposé en tant que solution de nuage géré, avec des coûts échelonnés en fonction du volume d'utilisation, à partir de 10 \$/mois. Les fonctions de ce serveur ou de ce service de traitement dans le nuage comprennent les tâches typiques d'une infrastructure (*back-end*) de l'IdO, c'est-à-dire la gestion des appareils, la collecte et la visualisation des données, le

traitement des événements et les appels de procédures à distance pour commander les appareils. Les fonctions sont également proposées via une API REST et peuvent donc être utilisées indépendamment du système et de l'appareil. Grâce au moteur de règles, il est possible de définir de façon pratique, au niveau graphique, la façon dont seront traités les événements qui se produisent (**fig. 2**).

Conclusion et perspectives

Vous devriez avoir à présent une vue d'ensemble de l'architecture d'une

solution IdO, dont la composante centrale est un service dématérialisé (dans le nuage) pour la gestion des appareils IdO et pour la réception et l'envoi des données. Bientôt nous mettrons ces connaissances en pratique. Il s'agira de concevoir un prototype de dispositif IdO sous la forme d'un bouton IdO. De telles solutions sont disponibles toutes faites, mais cela vaut la peine d'expérimenter ; c'est le meilleur moyen de comprendre les connexions, ce qui vous facilitera la conception de vos propres solutions. ─

(190303-02 VF)



Liens

- [1] Comparaison de plates-formes : www.informatik-aktuell.de/betrieb/virtualisierung/iot-in-der-cloud-erkenntnisse-und-erfahrungen-eines-plattformvergleichs.html
- [2] Microsoft Azure : <https://azure.microsoft.com/de-de/services/iot-hub/>
- [3] Raspberry Pi sur Azure : <https://docs.microsoft.com/de-de/azure/iot-hub/iot-hub-raspberry-pi-kit-c-get-started>
- [4] Google Cloud : <https://cloud.google.com/iot-core/>
- [5] Oracle IoT Cloud : www.oracle.com/internet-of-things/
- [6] Amazon IoT Cloud : <https://aws.amazon.com/de/iot/>
- [7] Q-loud-Cloud : www.q-loud.de/
- [8] Thingsboard : <https://thingsboard.io/>
- [9] REST : www.codecademy.com/articles/what-is-rest
- [10] REST API : <https://restfulapi.net/>

interpréteur BASIC pour ESP32 et ESP8266

Programmation avec Annex WiFi RDS



Peter Neufeld (Allemagne)

Annex WiFi RDS (= *rapid development suite*) est un environnement conçu pour utiliser le langage de programmation BASIC sur des modules ESP8266. Il existe une version ESP32. Ceux qui ont programmé les µC à 8 bits d'Atmel/Microchip avec AVR-BASIC ont maintenant accès à la classe de performance 32 bits.

Puisque l'ESP32 est un ESP8266 en plus puissant, la version du BASIC pour l'ESP8266 devrait se changer facilement en une version fonctionnellement identique pour l'ESP32. Au moment de rédiger ces lignes, elle est encore en phase bêta, mais elle sera définitive en mars.

Si vous aimez développer de petits projets avec connexion réseau, vous n'aurez bientôt plus à vous battre avec l'IDE Arduino ni le C/C++. BASIC est plus simple, et gratuit pour un usage privé non commercial. Ça mérite d'être essayé ?

Annex BASIC

Annex WiFi RDS [1] est basée sur le concept original d'ESPbasic [2], auquel a coopéré Francesco Ceccarella a.k.a. *chicciocb*. Toutefois, Annex n'est pas seulement une variante BASIC de plus, mais une suite entièrement révisée qui offre des fonctions améliorées, une plus grande fiabilité et une utilisation optimisée des capacités des µC Espressif.

Et la documentation est à la hauteur.

La gamme de fonctions de l'IDE, y compris l'interprète BASIC,

Caractéristiques

Fonctions d'Annex WiFi RDS :

- IDE intégré via un serveur web, à utiliser avec un navigateur web (optimisé pour Chrome et Firefox)
- Interprète BASIC avec variables à virgule flottante (double précision) et variables de type chaîne, tableaux multidimensionnels (float et chaîne) et sous-programmes
- Présentation syntaxique avec aide contextuelle
- Serveur web et de fichiers programmable
- Prend en charge les mises à jour de microprogrammes OTA (via WLAN)
- Prend en charge les événements asynchrones (interruptions, minuteries, accès au web, UDP, etc.)
- Points d'arrêt, exécution immédiate des commandes, affichage des variables, étape unique

- Accès à toutes les broches d'E/S, 1-Wire, SPI, I²C, PWM, Servo, NeoPixel, USART
- Traitement des erreurs, chien de garde
- TCP (HTTP GET et POST)
- Communication UDP
- Envoi de courriers électroniques via le serveur SMTP SSL
- Communication AJAX, ESP-NOW, MQTT et FTP (client)
- Algorithmes de fusion IMU/AHRS 6 et 9 DOF (*Madgwick* et *Mahony*)
- Contrôleur PID (4 canaux)
- Suite d'utilitaires sous Windows (Annexe Toolkit) avec :
- Flasheur, gestionnaire de fichiers, sauvegarde/restauration, convertisseur HTML, moniteur de port série, serveur de mise à jour OTA, console UDP et outil d'analyse IP.

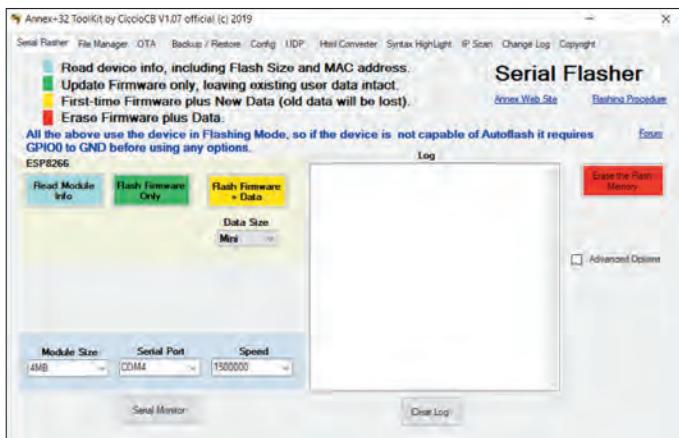


Figure 1 : L'outil Annex Serial Flasher pour le transfert du firmware vers un module ESP.

est étonnante (voir l'encadré **Fonctions**). Le logiciel complet se trouve dans le µC et fonctionne dans le navigateur web, qui affiche le site intégré. Soit le module ESP fournit un point d'accès WLAN et un serveur web sous l'adresse IP 192.168.4.1, soit vous connectez le module à votre WAN via DHCP (statique) et notez l'adresse IP reçue du routeur ou configurée à cet endroit. Grâce à ce dernier, l'ESP peut également synchroniser l'heure et la date avec un serveur de temps sur l'internet sans horloge matérielle supplémentaire.

IDE

Contrairement aux approches basées sur les compilateurs pour les µC ESP tels que l'EDI Arduino, le micrologiciel de l'annexe contient un ensemble fixe, non dépendant de la configuration, de fonctions intégrées et de pilotes pour les périphériques communs (encadré **Support matériel**). C'est à la fois une bénédiction et une malédiction : d'une part, dans le module il y a déjà tout pour adresser le matériel courant et exécuter immédiatement tout code de script utilisant ce matériel avec un support linguistique complet – d'autre part, ces fonctions consomment également de la mémoire dont vous ne disposez donc pas pour vos propres programmes.

Il y a ce qu'il faut pour communiquer avec des appareils

Installation

L'installation et la mise en service dans votre propre WLAN sont décrites en détail [4]. En voici un bref résumé :

- Le microprogramme et tout autre fichier sont transférés à un module ESP via l'interface série ou USB et sa mémoire peut être effacée au préalable.
- Connectez-vous en mode point d'accès (état de livraison) du module ESP ou intégrez mieux le module dans votre propre WLAN en tant que client WLAN.
- Définir les configurations de base telles que les paramètres du réseau local, les mots de passe, le fuseau horaire/STD, le programme Autostart BASIC, le serveur OTA, etc.
- Tout est maintenant prêt pour le chargement, l'édition, le débogage et l'exécution de programmes BASIC par navigateur.

externes au niveau du protocole à partir de vos propres sous-programmes BASIC. Il est également possible d'adresser avec votre propre code les périphériques non répertoriés ici via SPI ou I²C.

Annex WiFi RDS n'est disponible qu'en anglais. Le développeur est actif et répond rapidement aux commentaires sur le forum du projet [1]. Sur le site du projet [3], l'IDE pour les systèmes Windows disponible sous forme d'archive ZIP peut être décompressé n'importe où. L'application incluse [ANNEX-toolkit.exe](#) permet l'installation de l'interprète BASIC sur les modules ESP. Des conseils pour l'installation sont donnés dans l'encadré **Installation**.

Une fois installés sur le µC ESP le microprogramme comprenant l'interprète et l'IDE ainsi que des exemples optionnels, et que le µC est accessible via WLAN, tout peut être fait entièrement sur le PC dans la fenêtre du navigateur qui a chargé le site de l'IDE dans le µC. Le logiciel est optimisé pour Firefox et Chrome. Il est également possible de communiquer sur l'interface série en utilisant l'Annex Toolkit pendant le développement, ce qui est utile pour le dépannage. Parallèlement à la sortie des données via l'interface série, il existe également une sortie vers une zone de l'IDE appelée Wlog, qui permet le

Support matériel

Les dispositifs, actionneurs et capteurs suivants sont directement pris en charge par des commandes et des fonctions spécifiques :

- DHT11, DHT21 et DHT22 : capteurs de température/humidité
- DS18B20 : capteur de température.
- BME280 : capteur de température, d'humidité et de pression atmosphérique
- APDS9960 : capteur de distance, de lumière (intensité et couleur) et de gestes
- BNO055 : capteur d'orientation absolue
- HC-SR04 : Capteur à ultrasons (mesure de la distance)
- RTC DS1307 et DS3231 : Module d'horloge
- PCA9685 : Module PWM/SERVO
- LCD avec HD44780 via I²C ; 1/2/4 lignes de 16/20 caractères chacune

- LCD avec ST7920 avec 128 × 64 pixels monochrome
- Afficheur graphique OLED avec SSD1306 ou SH1106 ; 128 × 64 pixels monochrome
- LCD graphique avec ILI9341 ; 320 × 240 pixels en couleur 16 bits
- Afficheur à 7 segments avec TM1637 ; 4 chiffres
- Afficheur à 7 segments avec TM1638 ; 8 chiffres plus 8 LED et 8 boutons
- Afficheur à 7 segments avec MAX7219 ; 8 chiffres
- Afficheur à matrice de points avec MAX7219 ; 8 × 8 points
- LEDs couleur Neopixel WS2812
- Affichage à matrice de points avec le néopixel WS2812 ; 8 × 8 points
- Interface infrarouge ; bidirectionnelle (protocoles RC communs)

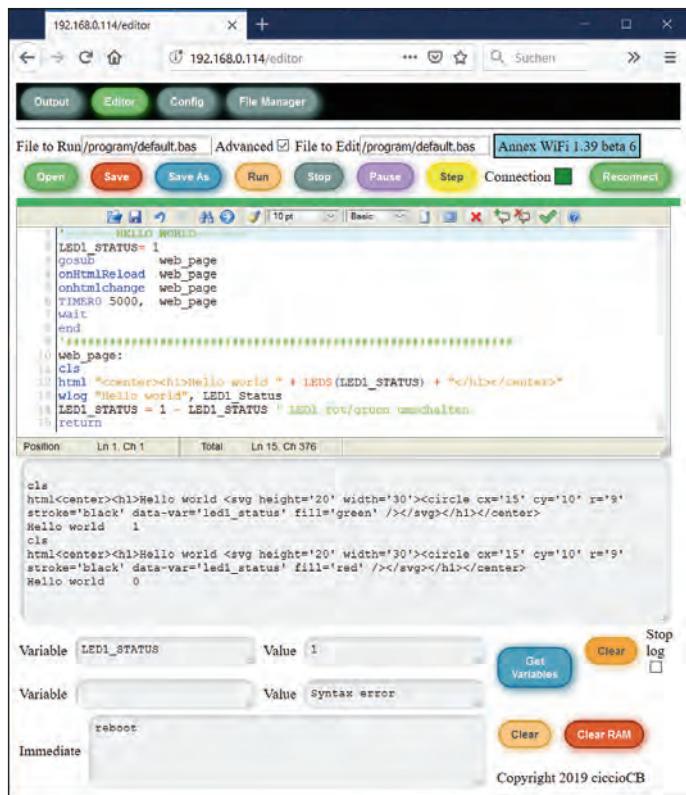


Figure 2 : L'éditeur Annex dans la fenêtre du navigateur avec l'exemple «Hello World».

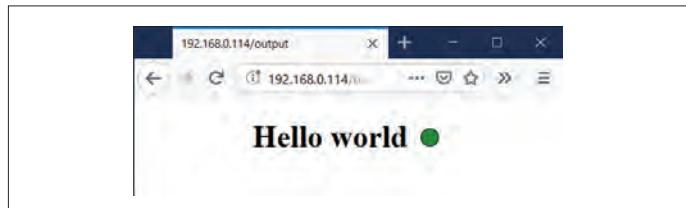


Figure 3 : Exemple «Hello World».

développement, par exemple avec un ordinateur portable sans connexion par câble au module ESP. Il est même possible de travailler avec plusieurs modules en parallèle.

Autres aspects

Pour que l'interpréteur BASIC utilise le moins possible de RAM, le script BASIC est copié de la mémoire SPIFF vers une zone désignée de la mémoire flash pour être exécuté, de sorte que

seuls la liste des lignes de programme, les identificateurs de branche et la liste des sous-programmes définis par l'utilisateur doivent être chargés dans la mémoire limitée du module. Cette méthode est légèrement plus lente que l'exécution directe en RAM, mais elle économise la RAM au profit des variables, ce qui permet d'avoir des programmes relativement longs. Le problème ne se pose pas avec l'ESP32, qui a quatre fois plus de RAM que l'ESP8266.

Une autre considération de performance est qu'un µC ESP doit toujours être capable d'exécuter de multiples activités en arrière-plan (serveur web, serveur de fichiers, etc.). Il a donc besoin de suffisamment de mémoire vive libre pour ces tâches. Les actions menées en parallèle affectent inévitablement la vitesse d'exécution de son propre code.

Selon le développeur, Annex-BASIC est largement compatible avec les variantes PC bien connues GW-BASIC et Visual Basic dont il partage de nombreux concepts, idées et syntaxes. Il existe une compatibilité avec le projet Micromite [5], une variante BASIC pour PIC32. L'interprète est basé sur le projet MiniBasic [6] et l'éditeur de texte sur le projet EditArea [7]. La vitesse d'exécution de l'interprète ne peut pas rivaliser avec un compilateur. Pour les tâches habituelles, le BASIC interprétré est néanmoins bien adapté aux µC ESP, suffisamment rapide, fiable et très facile à utiliser grâce à sa nature de navigateur. Toujours selon le développeur, le BASIC sur un ESP8266 dans le «banc d'essai comparatif BASIC des années 1980» est environ deux à quatre fois plus lent que sous Micromite à l'horloge de 48 MHz. Par rapport à ce qu'offre cet IDE, ce n'est pas une mauvaise performance.

Dans l'article **Sablier avec ESP8266 et Annex WiFi RDS** de la prochaine édition d'Elektor, vous trouverez un exemple pratique instructif basé sur cet IDE attractif. La version pour ESP32 a été équipée de quelques extensions utiles pour ce SoC. ─ (190400-02)

@ **WWW.ELEKTOR.FR**

→ ESP32 solo
www.elektor.fr/esp32-solo-1-wifi-bt-ble

→ ESP32 DevKitC
www.elektor.fr/esp32-devkitc

Liens

- [1] Annex WiFi RDS : <https://sites.google.com/site/annexwifi/home>
- [2] ESPbasic : www.esp8266basic.com
- [3] Téléchargements: <https://sites.google.com/site/annexwifi/downloads>
- [4] Premiers pas : <https://sites.google.com/site/annexwifi/home/first-steps>
- [5] Micromite : <http://geoffg.net/micromite.html>
- [6] MiniBasic : <https://malcolmmclean.github.io/minibasic/web/MiniBasicHome.html>
- [7] EditArea : www.cadolivet.com/editarea/

sonnette ESP32 par Telegram

« Le facteur sonne rarement une deuxième fois »



Luc Lemmens (Elektor Labs)

Cela nous arrive à tous : quelqu'un sonne à votre porte, mais vous n'entendez pas la sonnette ! Ou alors vous n'êtes pas à la maison. Si la personne à la porte connaît votre numéro de téléphone, elle pourra peut-être vous joindre. Et sinon, elle fait chou blanc et repart. C'est devenu particulièrement ennuyeux par exemple si vous êtes resté chez vous à attendre la livraison d'un colis.

Voici donc un projet de sonnette pas comme les autres, pour mettre fin à de tels désagréments.

Livreurs, facteurs et autres visiteurs ont rarement la patience d'attendre ou le temps de repasser. Plus tard, vous trouvez dans

votre boîte aux lettres la trace de leur passage, une note indiquant éventuellement le point de collecte de votre colis ou

l'annonce d'une nouvelle tentative de livraison le lendemain. C'est frustrant ! L'heure est venue de moderniser nos sonnettes.

INFOS SUR LE PROJET

	ESP32	Arduino	IoT
	débutant	connaisseur	expert
	±3 h		
	Arduino IDE, outils de labo courants		
	±85 €		

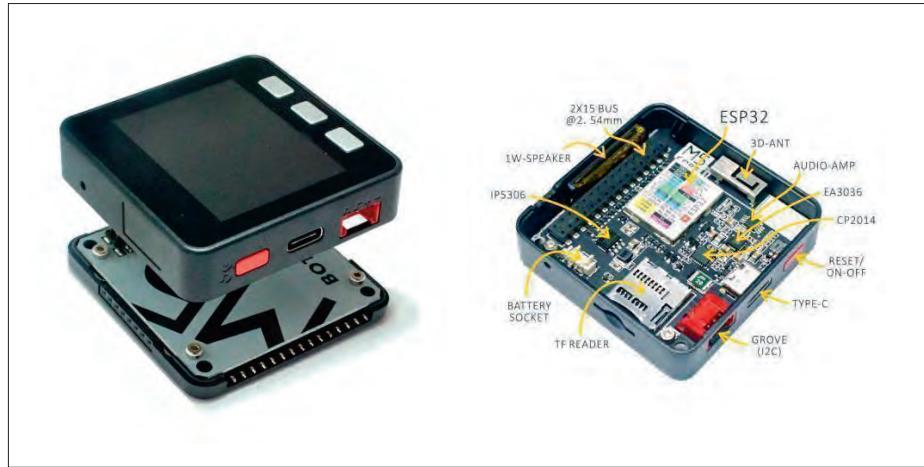


Figure 1 : Le module M5Stack ESP32 Basic Core disponible dans l'e-choppe d'Elektor.

que cette personne sonnera. À la limite, vous recevrez le message alors que le visiteur aura peut-être encore le doigt sur la sonnette. Même pas besoin de vous précipiter vers la porte d'entrée ! Si vous savez qui est ce visiteur, vous pouvez aussitôt renvoyer un message *Telegram* qui actionnera un ouvre-porte électrique.

Telegram rencontre ESP32

Nous verrons qu'il est facile de mettre en place un traitement intelligent automatisé avec *Telegram*. Communication et commande à distance sont assurées par un ESP32, plus précisément pour envoyer un message lorsqu'on appuie sur un certain bouton et pour répondre (éventuellement) par l'activation d'un relais. Les possibilités ne se limitent pas à cette application simple. Le principe pourra facilement être adapté et étendu pour répondre à d'autres besoins de commande à distance.

L'ESP32 (et son prédecesseur l'ESP8266) revient dans de nombreux projets et livres d'Elektor. Ici, nous utiliserons l'ESP32 comme module intelligent qui interagit avec le monde extérieur par Wi-Fi interposée. L'environnement de développement *Arduino* et la bibliothèque de (ro)bots *UniversalTelegram* simplifient *en principe* le développement d'applications comme cette sonnette. (pour la petite histoire énigmatique, voir le cadre *Mauvaise rencontre*).

Matériel : noyau M5Stack ESP32

Moyennant des modifications mineures du logiciel, ce projet pourra être utilisé avec n'importe quel module ESP32 ou 8266. Nous avons opté pour le module

M5Stack ESP32 Basic Core de la **fig. 1**, disponible dans l'e-choppe d'Elektor (référence en fin d'article). Ce module compact réunit tout ce qu'il faut :

- affichage graphique
- boutons poussoirs
- haut-parleur
- batterie rechargeable intégrée

Les E/S du *M5Stack ESP32 Basic Core* sont accessibles de l'extérieur sur des connecteurs. Ouvrez l'arrière de la boîte et retirez une carte enfichable, vous trouverez un connecteur d'extension à 30 voies pour votre matériel.

Pour les petites extensions comme celle-ci, le matériel supplémentaire s'installe sur une carte de même taille que le module M5Stack lui-même. Il existe pour le boîtier de nombreux modèles pour l'impression 3D [2]. Ce qui vous fera un appareil bien fini. Avec un minimum d'efforts, vous passez du stade de prototype expérimental à celui d'un boîtier qui se laisse voir aussi en dehors de l'atelier de bricolage. En bref, ce module M5Stack offre une base excellente pour de nombreux projets ESP32, depuis les plateformes de développement de projets aux dispositifs «fermés» ou «clé en main». Pour commencer, vous pouvez utiliser le M5Stack et notre carte d'extension pour construire une installation complète de sonnette. En termes de volume, le haut-parleur incorporé peut rivaliser avec une *vraie* sonnette de porte. Un bouton de sonnette séparé peut être câblé en parallèle avec (ou à la place du) transistor de sortie de l'optocoupleur. Le signal de sonnerie doit être ajouté dans le *sketch* Arduino, et

Un télégramme pour vous !

Bientôt, dès que votre *nouvelle* sonnette retentira, ce circuit enverra un message à votre téléphone tactile où tourne une application appelée *Telegram Messenger*. Vous connaissez peut-être cette application de messagerie mobile, appréciée pour sa légèreté, sa rapidité, sa sécurité et sa vitesse. On en trouve des déclinaisons pour les trois plateformes de communication courantes : Android, Apple et Windows ainsi que des versions de bureau pour PC/Mac/Linux, MacOS... Comme *WhatsApp*, *Telegram* est lié à votre numéro de téléphone portable, mais vous pouvez également enregistrer votre nom d'utilisateur, ce qui vous permet de garder votre compte même si vous changez de numéro. Sur les autres différences plus ou moins subtiles entre ces deux messageries, vous trouverez sur l'internet des comparaisons, des campagnes de dénigrement et de glorification, des cohortes d'admirateurs, de détracteurs, et beaucoup d'interminables débats. Ce qui frappe du côté de *Telegram*, c'est la sobriété. Voyez leur site [1].

Revenons à notre nouvelle sonnette. Si votre application *Telegram* est active et connectée au serveur du même nom (par Wi-Fi ou réseau mobile), elle vous avertira de la présence d'un visiteur à votre porte, où que vous soyez, dès

le niveau logique sur GPIO17 doit être inversé si le bouton est connecté ainsi. Nous n'irons pas plus loin, et nous nous limiterons ici à une extension de sonnette à raccorder à une installation existante.

Schéma

Nous supposons sur le schéma de la **fig 2** que le module M5Stack est alimenté par son interface USB. Le rôle du matériel supplémentaire est donc la commande d'un ouvre-porte par T1 et RE1, ainsi que, par l'intermédiaire de l'optocoupleur IC1, la connexion au bouton de sonnette existant. Notre extension de sonnette peut donc être raccordée directement à une sonnette existante, dont les deux fils du bouton sont donc connectés à K1. Une sonnette électrique est habituellement alimentée par une tension alternative, qui n'a jamais fait l'objet d'une standardisation. On trouve de tout entre 6 V et 24 V, et on tombe parfois sur des variantes exotiques. Cette tension alternative est présente sur les fils du bouton de la sonnette tant qu'il n'est pas enfoncé, et sur K1 également. L'optocoupleur traduit la présence de cette tension en un niveau logique sur la ligne GPIO17 de l'ESP32 : bas au repos (on ne sonne pas), haut quand on sonne. Ensuite, il y a la connexion BT1, prévue en option pour une batterie au lithium de 3,7 V montée en série dans le module M5Stack, rouge au positif, noir au moins de BT1.

La carte d'extension ne comporte que des composants traversants, faciles à assembler. Attention à K2, monté d'une manière inhabituelle (**fig 3**). Normalement, c'est le côté court des broches que l'on insère dans le PCB côté composants pour les souder de l'autre côté. Ici ce sera l'inverse : les broches longues sont insérées côté cuivre (sous le PCB) pour les souder du côté des composants. Le côté court des broches est inséré dans le support du M5Stack et la partie en plastique de K2 sert également d'entretoise entre les deux circuits imprimés.

Installation de Telegram

L'étape suivante consiste à créer un compte *Telegram* sur votre téléphone, lequel commencera par télécharger l'application de l'*App Store* ou de *Google Play*. L'installation est simple, le compte lié au numéro de téléphone. Il est recommandé d'installer également l'application *Telegram* sur votre PC ou Mac, elle

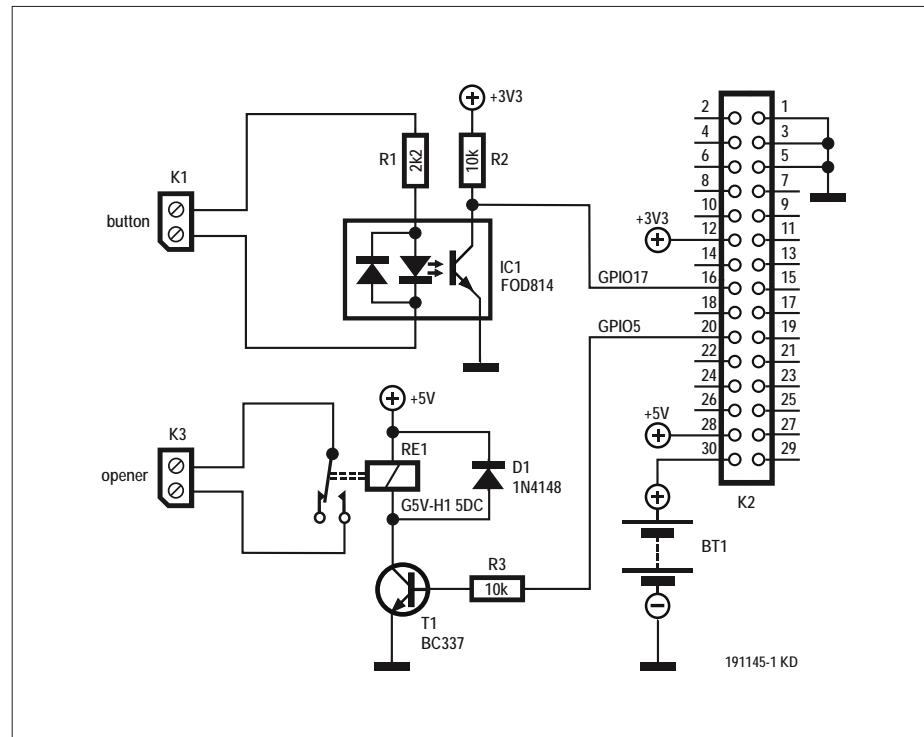


Figure 2 : Schéma de la sonnette de porte combinant les atouts d'Arduino, d'ESP32 et de l'application et du service de messagerie en ligne gratuit *Telegram*.

fonctionne différemment une fois que la connexion entre *Telegram* et le *sketch* ESP32 sera faite.

Connectez-vous à l'application *Telegram* sur votre ordinateur en utilisant votre numéro de téléphone pour vous identifier. Aussitôt l'application sur le téléphone

affichera un code de vérification à saisir dans l’application sur l’ordinateur pour y autoriser *Telegram*. Le programme sur l’ESP32 est un robot répondeur. Il peut répondre à des commandes reçues par un message *Telegram* ou envoyer des messages toujours par *Telegram*, déclen-

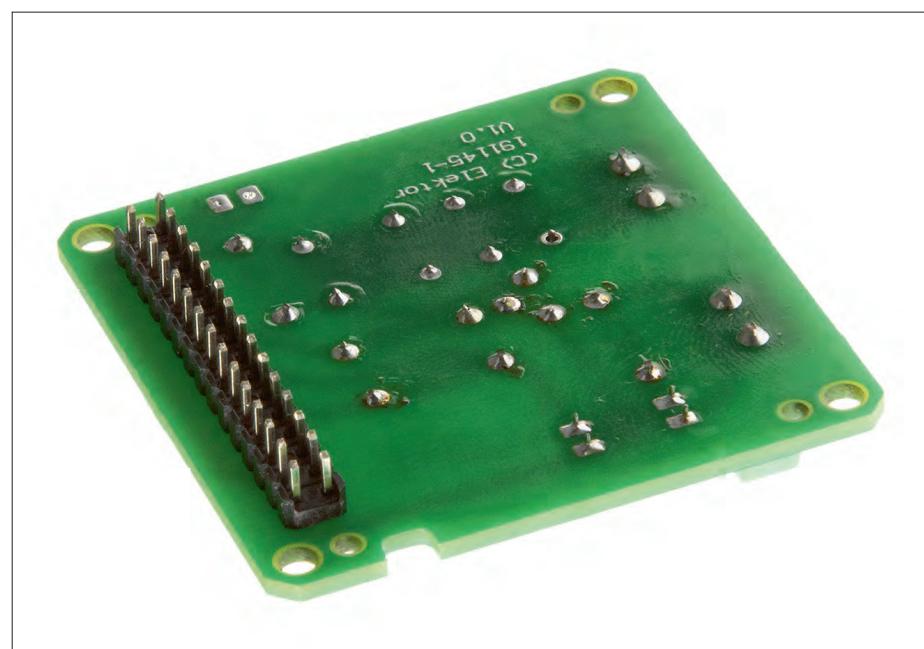


Figure 3 : Méthode recommandée pour le montage de K2 : la partie **longue** des broches est insérée par le côté cuivre (sous le PCB), pour être soudée du côté des composants.

Mauvaise rencontre (furtive)

Tout va bien, votre code source soigneusement étudié est prêt, il se compile sans erreur, il est téléchargé sur l'ESP32 sans problème, et le circuit fonctionne comme prévu. C'est bien parti, mais... vous constatez que le µC est initialisé chaque fois qu'un message arrive par Telegram ! Vous vérifiez et revérifiez. L'erreur persiste. Vous recompilez et testez le programme sur un autre ordinateur et là, à votre grande surprise, l'ESP32 fait ce qu'on attend de lui ! Les deux ordinateurs ont la même version de l'EDI Arduino et des bibliothèques. Aucune anomalie. Alors vous supprimez complètement 'Arduino' sur le premier ordinateur. Pas une simple désinstallation, qui laisse des traces dans des dossiers cachés. Vous procédez à une nouvelle installation d'Arduino et tous les problèmes ont disparu.

Quelle est donc la nature exacte de ce problème ?

Aucune idée. Ce qu'il faut en retenir c'est que l'IDE Arduino _peut_ se comporter de manière erratique. Au labo d'Elektor, je crois que c'est la première fois que nous avons rencontré un tel problème.

Espérons que ce fut aussi la dernière !

chés par des événements dans ou autour de l'ESP ; par exemple pour signaler une pression sur une touche ou envoyer une valeur de tension après conversion analogique-numérique ADC. Dans notre projet, les commandes commandent un ouvre-porte. Il faut d'abord créer ce (ro)bot dans *Telegram* en cherchant dans la liste de contacts un certain *BotFather*.

Celui-ci est lui-même un robot pour créer votre bot de sonnette en utilisant une discussion (*chat*). Après avoir saisi la commande /newbot le dialogue commence en demandant un nom (qui apparaîtra plus tard dans la liste des

contacts *Telegram*) et un nom d'utilisateur pour le (ro)bot. Ce dernier doit toujours se terminer par les caractères «bot». Notez la barre oblique (/) au début de chaque commande du bot. Lorsque ce processus est réussi, le *BotFather* vous en informe et affiche un code (*token*) nécessaire pour autoriser le nouveau (ro)bot à envoyer des requêtes à l'API.

Gardez ce jeton en lieu sûr, il pourrait permettre à n'importe qui de prendre le contrôle de votre (ro)bot de sonnette. C'est pourquoi nous l'avons flouté dans le dialogue de configuration de notre propre robot (**fig 4**).

Dans ce dialogue avec *BotFather*, cliquez sur le lien encadré ('t.me/...') pour ouvrir une fenêtre de discussion pour le nouveau robot, également ajouté à la liste de contacts. Ce (ro)bot est maintenant prêt à l'emploi, il ne reste qu'à le relier au *sketch* Arduino pour le prolongateur de sonnette.

Sketch Arduino

Dans le téléchargement du logiciel de ce projet, vous trouverez le *sketch DOORBELL.INO*, c'est-à-dire le code source du micrologiciel ESP32 [3]. Ce programme peut être retouché et, dans sa forme actuelle, il est destiné à montrer surtout comment Arduino/ESP32 et *Telegram* travaillent ensemble pour simplifier les systèmes de commande.

L'utilisation d'un ouvre-porte électrique est une brèche potentielle, car un intrus pourrait pirater le (ro)bot et ouvrir votre porte avec son téléphone (ou son ordinateur). Elektor décline donc toute responsabilité en cas de dommage résultant de l'application de ce projet !

Dans le *sketch*, notre module M5Stack nécessite une connexion au réseau local Wi-Fi, et l'accès au bot *Telegram* que vous venez de créer. Les identifiants de votre réseau Wi-Fi sont dans les lignes 17 et 18, tandis que le jeton attribué par *BotFather* est à la ligne 21. Vous comprenez pourquoi on vous conseillait de créer le bot sur votre ordinateur, afin de copier-coller facilement le jeton dans le *sketch*. Une fois l'installation de l'EDI Arduino terminée avec succès, l'esquisse se compilera sans aucun problème. Dans le menu Outils — lorsque le M5Stack est connecté à un port USB de l'ordinateur — la carte correcte (M5Stack-Core-ESP32) et le port COM virtuel doivent être réglés afin de télécharger le *sketch* sur l'ESP32.

Cela fait, le *sketch* se lance, se connecte au réseau, et le nom du réseau Wi-Fi et l'adresse IP du prolongateur de sonnette apparaissent sur le LCD du M5Stack. Le (ro)bot doit être activé après chaque redémarrage en envoyant la commande /start via la fenêtre de discussion dans *Telegram*. Le (ro)bot fournit alors un retour d'information avec vue d'ensemble des commandes reconnues.

Lorsque la sonnette retentit, un message est envoyé par l'application *Telegram* à votre ordinateur ou votre téléphone.

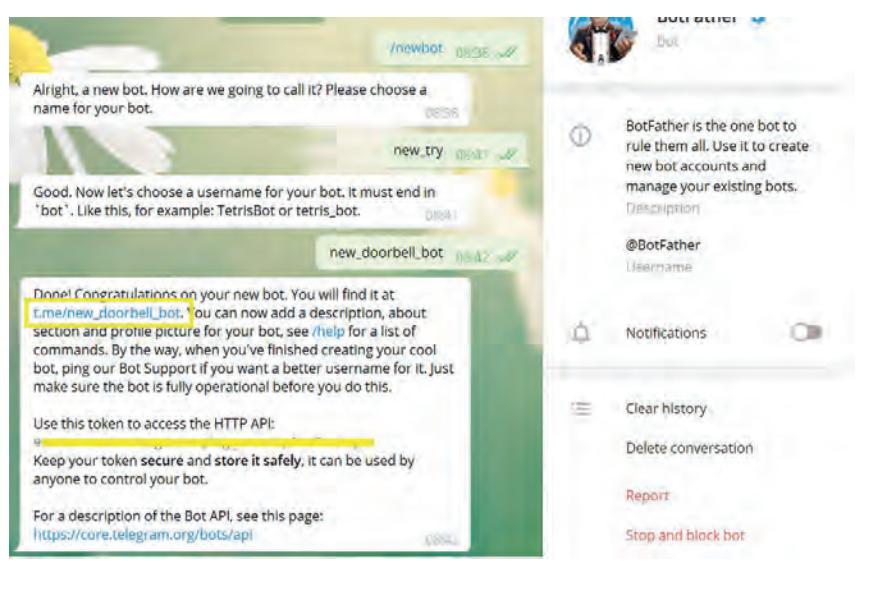


Figure 4 : Cette copie d'écran montre comment nous nous sommes débrouillés au labo pour l'utilisation de *BotFather*. Les jetons ont été cachés pour brouiller l'écoute.



LISTE DES COMPOSANTS

Résistances

R1 = 2,2 k Ω 5%, 0,25 W, 250 V
R2,R3 = 10 k Ω 5%, 0,25 W, 250 V

Semi-conducteurs

D1 = 1N4148

T1 = BC337

IC1 = FOD814A, optocoupleur AC avec sortie à transistor, 1 canal, DIP, 4 broches, 50 mA, 5 kV

Divers

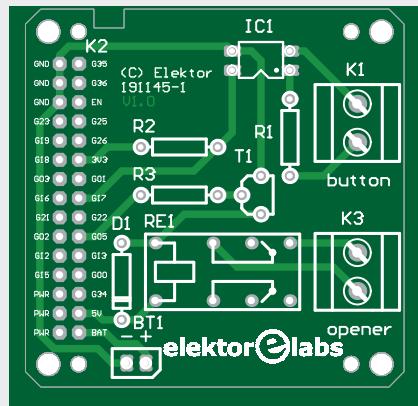
RE1 = G5V-2-H1 5DC, bobine 5 V_{cc}, DPDT,
1A (Omron)

K1,K3 = bornier à vis pour circuit imprimé à
2 voies (pas de 5,08 mm), 630 V

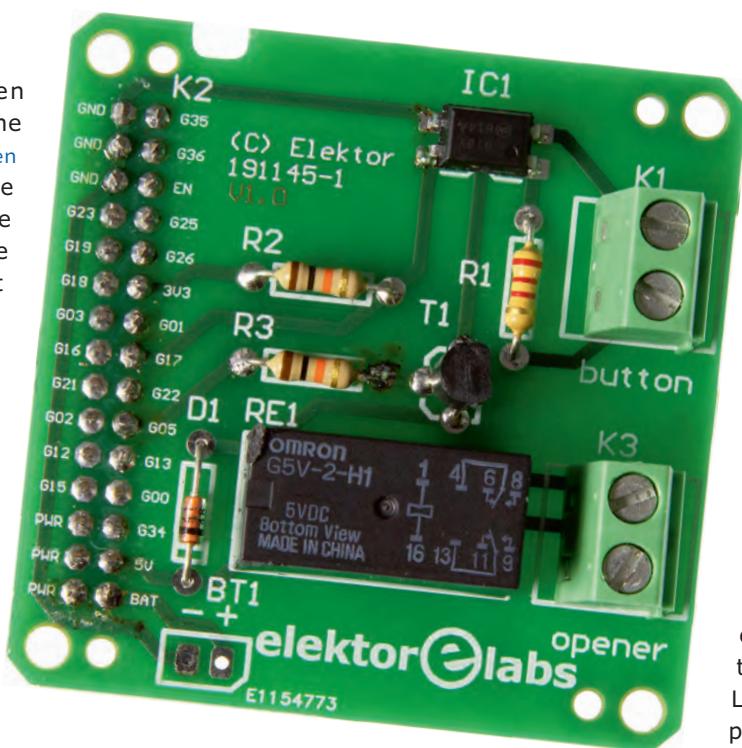
K2 = connecteur à 30 broches (2x15),
snap-off, vertical

module M5Stack ESP32 Basic Core, dans l'e-choppe d'Elektor (ou module ESP32 équivalent)

BT1 = batterie en option avec support
circuit imprimé 191145-1 V1.0 dans l'e-choppe
d'Elektor



Vous pouvez répondre en ouvrant la porte vous-même ou en utilisant le bouton `/open` pour actionner un ouvre-porte électrique (s'il y en a un). Le bouton `/Ignore` permet de renoncer à cette option, et le bouton `/status` (occupé à vérifier la connexion au prolongateur de sonnette) apparaît à nouveau en bas de l'écran de discussion. Ces commandes peuvent également être tapées et envoyées via la fenêtre de discussion, mais attention, elles sont sensibles à la casse ! C'est la bibliothèque de (ro)bots



[UniversalTelegram](#) [4], qui fait la plus grande partie du boulot dans ce projet. Grâce à sa documentation et aux exemples fournis, ce n'est pas bien sorcier. Notre *sketch* de sonnette appuie sur l'exemple du « clavier personnalisé », qui montre comment envoyer des boutons de commande à [UniversalTelegram](#), une option pratique, qui évite d'avoir à taper les commandes dans une fenêtre de discussion. La routine indispensable est `handleNewMessages` qui gère les messages entrants. Il y a quatre instructions `IF` qui permettent au (ro)bot de prendre :

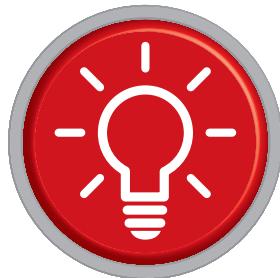
- start
 - status
 - Ignore
 - Open

Pour pouvoir envoyer des messages, le *sketch* a besoin de l'ID de discussion enregistré dans la variable `ThisChat` après avoir reçu la première commande `/start`. Le bouton de commande (c'est-à-dire la sonnette) est capturé au cours d'une interruption ; notez l'attribut `IRAM_ATTR` requis pour la déclaration d'une routine de service d'interruption ESP, tout cela pour un traitement correct bien sûr. ►

(191145-03 VF)

Liens

- [1] Telegram Messenger : <https://telegram.org/>
 - [2] modèles 3D pour le boîtier M5Stack :
<https://github.com/m4k3r-net/M5Stack-3DPrintFiles>
 - [3] sketch Arduino : www.elektrormagazine.com/191145-01
 - [4] bibliothèque de (ro)bots UniversalTelegram :
<https://Github.com/witnessmenow/Universal-Arduino-Telegram-Bot>



retour des petits circuits

... et des bonnes petites pépites d'Elektor

Dans le numéro d'Elektor de nov.-déc. 2019, nous présentions un circuit de démarrage en douceur avec de nombreuses fonctions, comme il convient pour des amplificateurs audio lourds. Parmi les petits circuits de la présentation édition, vous trouverez ci-dessous un autre circuit de ce genre, beaucoup plus simple, mais très efficace aussi...

Idée : Ton Giesberts (Elektor Labs)

Limitation du courant d'appel

Ce circuit (**fig. 1**) est très simple. La description de son fonctionnement sera brève. Supposons que vous allez mettre en marche un gros amplificateur, équipé de condensateurs électrolytiques de très forte capacité. Au moment où vous le mettez sous tension, pendant une fraction de seconde, il circule un courant de très forte intensité jusqu'à ce que ces condensateurs-réservoirs soient chargés. Et le claquement que vous entendez au loin, c'est le disjoncteur de la boîte à fusibles qui saute... Avec ce circuit de limitation de courant inséré en amont de l'ampli (ou d'une autre grosse charge), ce danger est éliminé. La résistance de $33\ \Omega$ limite le courant d'appel à un maximum d'environ 7 A. En circulant, ce courant provoque une chute de tension (considérable) aux bornes de la résistance, de sorte que le relais 230 V ne peut pas s'enclencher.

Après une fraction de seconde de seulement 5 à 10 périodes de la tension du réseau, la charge des condensateurs électrolytiques est complète et l'intensité se réduit au courant de fonctionnement normal de la charge. La chute de tension aux bornes de la résistance diminue en conséquence, la tension aux bornes de la bobine du relais augmente et le relais s'enclenche. En se fermant, son contact court-circuite le fusible et la résistance connectés en série ; c'est l'état de fonctionnement normal et stable.

Cette résistance est un colosse de 50 W. Durant la mise sous tension, la puissance de crête est brièvement bien supérieure à 50 W, mais ça se passe si vite que la résistance y fait face sans

problème. Idem pour le fusible : la pointe de courant d'appel est si courte et le fusible est si lent qu'il ne se déclenche pas. Si la charge est défectiveuse, p. ex. en cas de court-circuit, la forte intensité du courant d'irruption persiste un instant, jusqu'à ce que le fusible fonde, ce qui nous épargne une situation dangereuse : caramélisation de la résistance et début d'incendie. Pin-pon pin-pon...

Veillez à adapter les valeurs du fusible et de la résistance aux caractéristiques de votre charge.

Idée : Ton Giesberts (Elektor Labs)

Commande automatique de gain

Ce circuit ajuste automatiquement le gain des (petits) signaux d'entrée afin que son signal de sortie reste constant sur une large plage. La tension de commande nécessaire est obtenue par un montage en cascade.

Cette commande automatique de gain (CAG) sera utile pour détecter les signaux faibles. Imaginez un détecteur de chauve-souris : les cris (ultrasoniques) des chauves-souris sont captés par le microphone, amplifiés et déplacés dans la gamme audible pour les humains. L'objectif premier est de capturer ces signaux : la qualité de l'amplification est secondaire.

Pour l'amplification proprement dite, on utilise les deux moitiés d'un ampli op double TL072 (IC1A et IC1B). IC1A est un amplificateur non inverseur dont le gain est

$$A1 = R4 / (R2 + R3 || T1) + 1$$

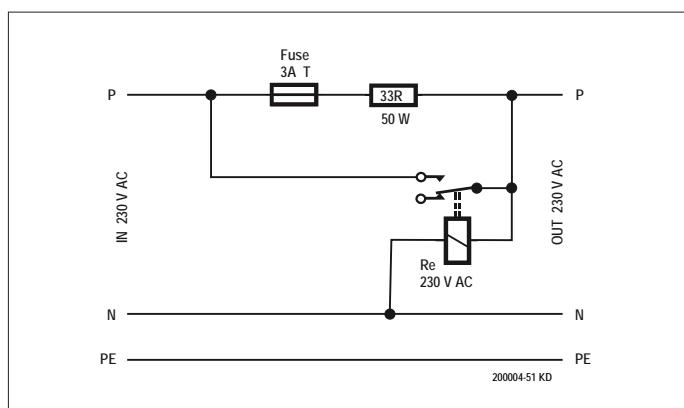


Figure 1. Difficile de faire plus simple que cette limitation du courant d'irruption.

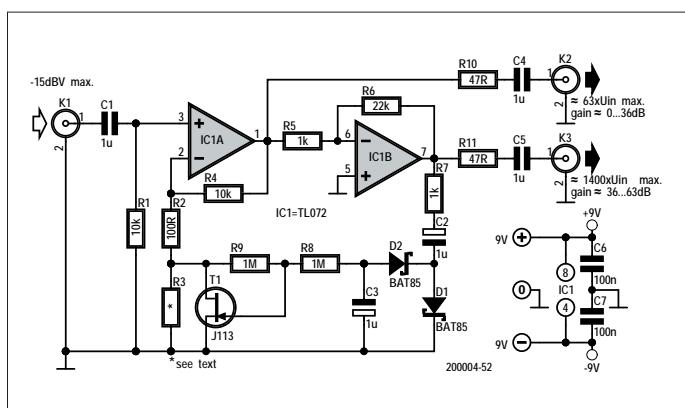


Figure 2. Une commande automatique de gain n'est pas un circuit forcément compliqué.

IC1B, par contre, est inverseur. Son gain est :

$$A2 = -R6 / R5$$

L'astuce de ce circuit, ce sont D1 et D2. En cascade avec C2 et C3, elles forment un doubleur de tension. Une tension de commande est dérivée de la tension de sortie de IC1B, utilisée pour piloter plus ou moins le JFET T1.

Un JFET peut être utilisé comme résistance variable. Sa résistance dépend de la tension appliquée à sa grille. Dans ce circuit, nous avons connecté une R3 en parallèle au JFET T1 pour assurer un facteur de gain minimum pour IC1A. Vous pouvez expérimenter : soit omettre complètement R3, soit jouer avec sa valeur.

Dans le circuit (**fig. 2**), le JFET est un J113. Les deux diodes sont Schottky (seuil bas et commutation rapide), mais ça fonctionne aussi avec d'autres FET et des diodes ordinaires 1N4148 au lieu des Schottky. Le circuit n'est donc pas très critique. Pour l'expérimentation, vous pouvez aussi le monter sur une plaquette d'expérimentation à trous. Une fois satisfait, soudez-le sur un circuit à pastilles.

Voici encore quelques détails sur ce circuit. L'impédance d'entrée du circuit AGC est réglée avec R1 sur $10\text{ k}\Omega$, la fréquence de coupure inférieure est de 16 Hz (C1). La valeur de R2 plus la résistance minimale du JFET T1 ($100\ \Omega$ selon la fiche technique pour $U_{GS} = 0\text{ V}$) détermine le gain maximal du circuit. Dans notre prototype, cette résistance minimale de T1 était légèrement inférieure (environ $60\ \Omega$), de sorte que l'amplification maximale du circuit s'applique à $U_{GS} = 0$:

$$A = (1 + R4/(R2 + RGS)) \times (R6/R5) \approx 1400$$

La vitesse de réaction de la CAG (attaque) est déterminée par R7. Elle est de l'ordre de quelques millisecondes. Ne réduisez pas la valeur de R7 par rapport à ce qui est spécifié dans le schéma, cela augmenterait la distorsion à la sortie K3. Le réseau R8+R9 plus C2 et C3 fixent le temps de récupération (décroissance) du circuit ; il est de quelques secondes. L'alimentation est assurée par deux piles de 9 V, la consommation est de 3,5 mA.

Nous avons évidemment testé le circuit au labo d'Elektor. La **fig. 3** montre que le niveau de sortie (ligne bleue) est à peu près constant sur une large plage du signal d'entrée (axe horizontal).

Idée : Elex

Récepteur radio à diode

Enfin, un autre legs du bon vieux temps : un récepteur qui, aussi incroyable que ça paraisse, ne comporte **ni pile ni un seul composant actif**, vous permet de recevoir (la nuit) des émissions radio ! Son principe est simple : avec la bobine, le condensateur variable forme un circuit résonnant. Le signal (modulé en amplitude) capté par l'antenne est détecté par la diode, est acheminé par le condensateur de sortie sur un écouteur à cristal de haute impédance qui permet de l'écouter sans amplification.

Ce récepteur radio rudimentaire ressemble aux tous premiers appareils de ce genre [**2&3**], à ceci près que le détecteur à

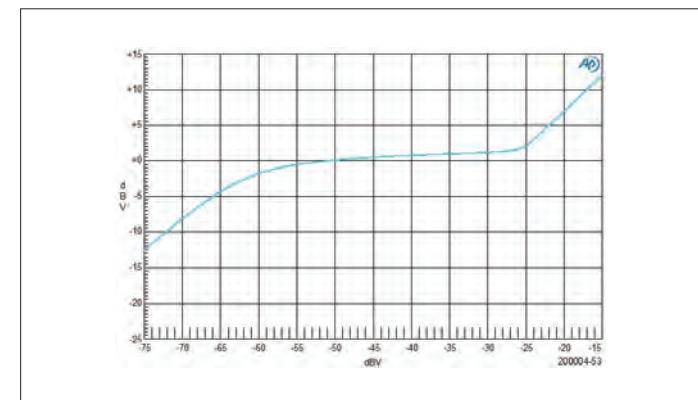


Figure 3. En pratique, la réponse de ce circuit AGC est étonnante.

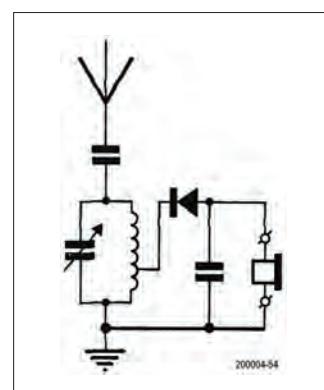


Figure 4. C'est ainsi qu'a commencé l'histoire de la radio... 200004-54

cristal de galène est remplacé par une diode ! Pour vos premières expériences, on peut enrouler environ 85 tours de fil de cuivre émaillé d'une épaisseur de 0,2 mm sur un bâton de ferrite d'environ 10 cm de longueur. La diode au germanium AA119 ou similaire est connectée à une prise intermédiaire de la bobine. L'écouteur à cristal doit être à haute impédance (environ $2000\ \Omega$) sinon vous n'entendrez rien. La capacité du condensateur d'accord est d'environ 200 pF, l'antenne est connectée à travers un petit condensateur d'environ 3,9 pF. Pour le condensateur de redressement du côté de l'anode de la diode, 10 nF est une bonne valeur.

Il vous faut une longue antenne filaire de quelques mètres au moins, et enfin, pour que ça fonctionne bien, une vraie terre, reliée à une conduite d'eau ou de chauffage central métallique ! La prochaine fois, nous épicerons la sauce avec des composants actifs. ▶ 200004-03



@ WWW.ELEKTOR.FR

→ Electronic Circuits For All (livre en anglais)
www.elektor.fr/electronic-circuits-for-all

Weblinks

- [1] Démarrage en douceur pour amplificateur : www.elektormagazine.fr/magazine/elektor-115/56859/
- [2] Récepteur à cristal : https://fr.wikipedia.org/wiki/Récepteur_à_cristal
- [3] Récepteur à cristal (en anglais) : https://en.wikipedia.org/wiki/Crystal_detector

LoRaWAN : décollage facile

Avec Blue Pill, passerelle LoRa et The Things Network



Mathias Claußen (Elektor Labs)

La technologie radio LoRa, proposée par *Semtech*, est un mode de transmission de données à longue portée et à faible consommation. LoRa convient aux capteurs en réseau dont il faut ménager la source d'énergie.

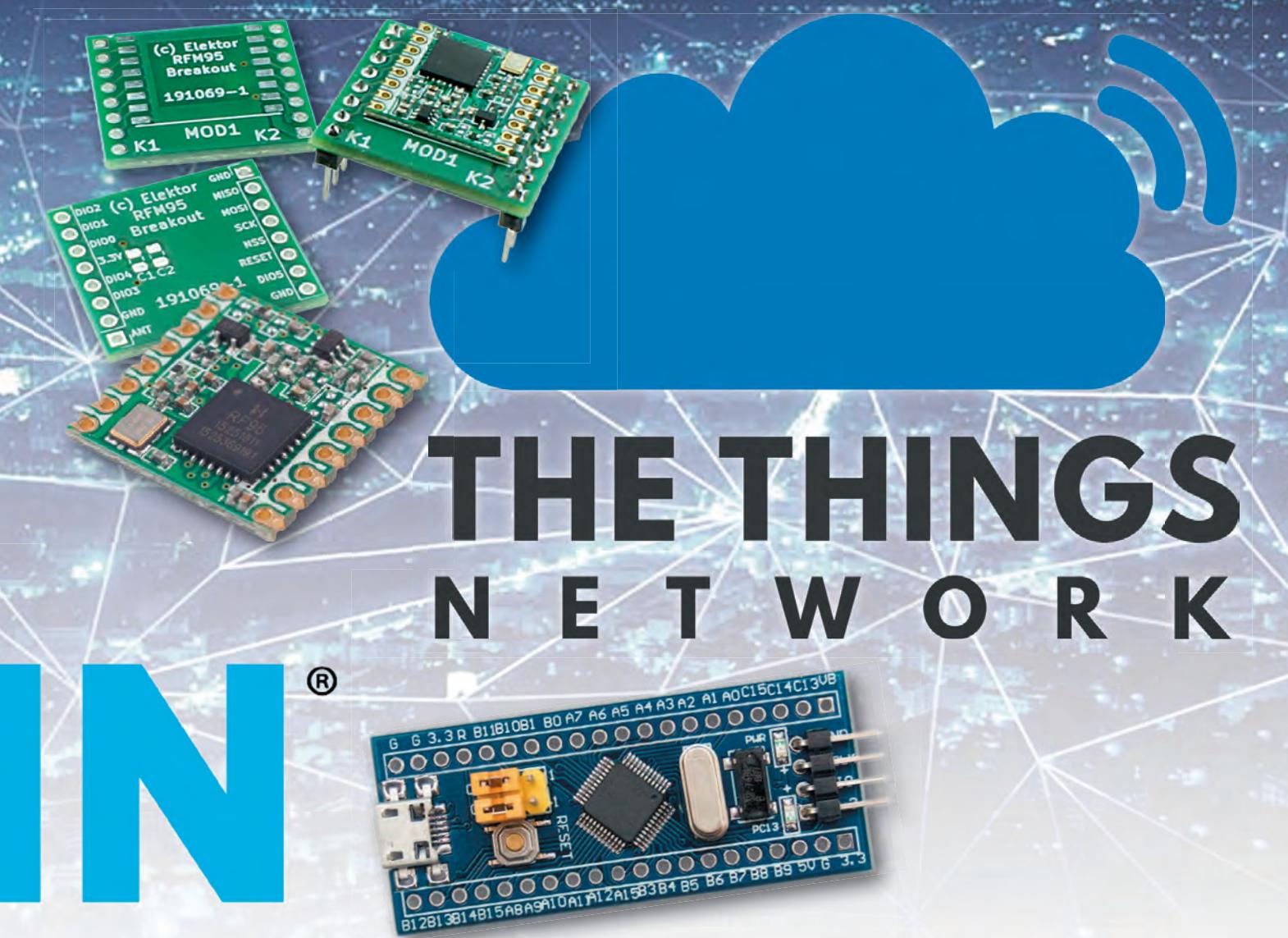
The Things Network est un réseau populaire et ouvert, qui peut recevoir des données émises par des capteurs et les rendre disponibles dans le monde entier.

Pour vos premières expériences, quelques euros et quelques cartes suffisent.

Comme son nom l'indique, un LoRaWAN utilise la technologie radio LoRa (=low range) pour faire circuler des données sur un réseau étendu (WAN = wide-area network). Il fournit l'infrastructure à distance pour les nœuds de capteurs en recueillant les données par des passe-

relles (stations de base) via LoRa et en les mettant à disposition sur l'internet. *The Things Network* est un LoRaWAN populaire en pleine expansion. Ce réseau communautaire gratuit jouit d'une bonne couverture. Pour utiliser activement ce réseau, il suffit d'un enregistrement et bien sûr le

matériel adéquat. Il n'y a ni frais mensuels ni facturation par message ; il suffit de se conformer à la politique d'utilisation équitable du réseau. Cette *Fair Use Policy* vise à permettre à tous les participants de transmettre leurs données et à éviter la surcharge des passerelles (*gateways*).



Petits moyens

La solution présentée ici n'est qu'imparfaitement conforme aux spécifications de LoRaWAN et reste donc réservée à l'expérimentation. Les coûts sont de l'ordre de 23 € pour le nœud du capteur, plus éventuellement une plaque d'expérimentation et quelques fils de connexion. La liste des composants pour le capteur LoRa est assez courte :

- Carte contrôleur STM32 *Blue Pill* [1]
- Module radio LoRa RFM95 [2]
- Convertisseur USB-série [3]

- Cavaliers
- Plaque d'expérimentation

Pour commencer, nous utilisons une carte *Blue Pill* arduinoïde avec un puissant contrôleur STM32-ARM Cortex, disponible pour quelques euros, débogueur compris. Elle n'offre pas moins 64 Ko de Flash, 16 Ko de RAM et un port USB. Ce dernier est à utiliser avec circonspection, car cette carte est disons fortement optimisée en termes de coûts (pour le dire avec des fleurs), de sorte que les problèmes avec le port USB sont malheureusement

INFOS SUR LE PROJET	
	LoRa LoRaWAN BoB
	débutant → connaisseur expert
	±30 min
	PC, plaquette d'essai et fils
	±30 €

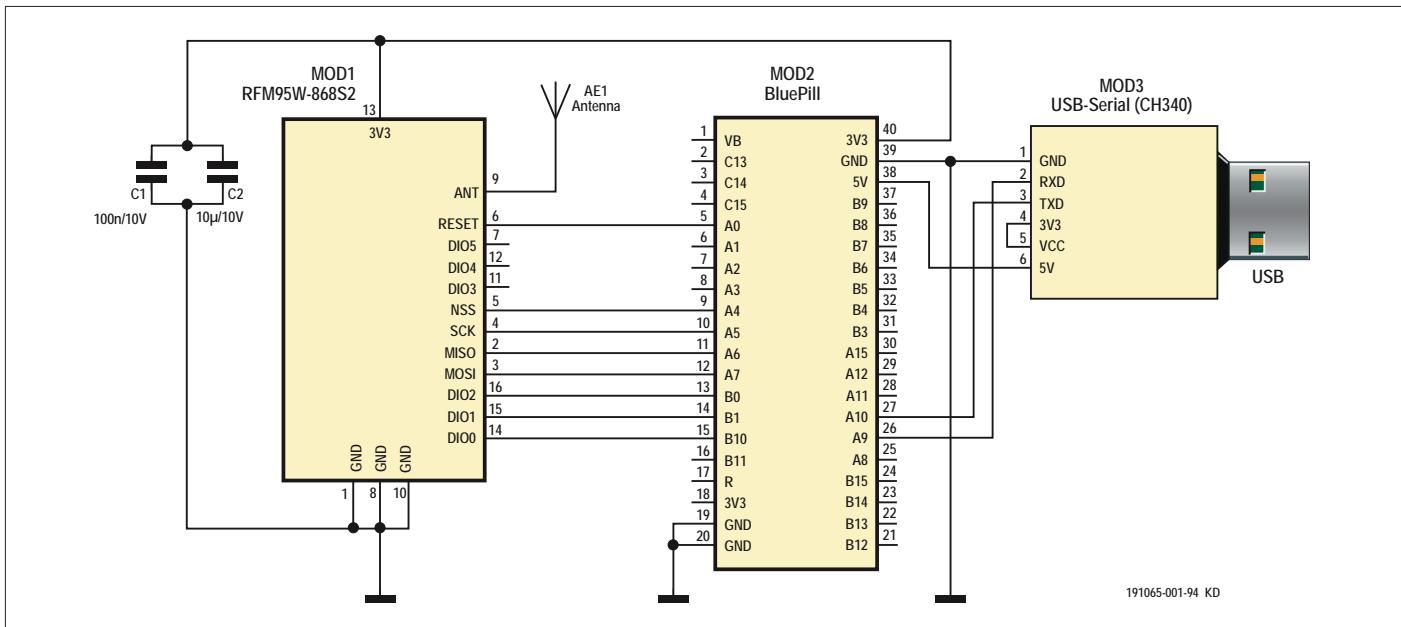


Figure 1 : Schéma du circuit du nœud LoRa à faible coût.

la règle plus que l'exception. Comme prise d'alimentation, pas de problème. Pour programmer la puce, on utilise le *bootloader* intégré. Le nouveau logiciel peut être programmé de manière fiable via l'interface série UART de la carte. Les quelques composants, le module LoRa, la carte *Blue Pill* et un convertisseur USB-série sont connectés comme indiqué dans le schéma (**fig. 1**). Comme le module LoRa avec son pas de 2,00 mm n'est pas compatible avec les plaques d'expérimentation, Elektor Labs propose une petite carte d'interconnexion (**fig. 2**). Grâce à elle, le module pourra être enfiché facilement sur une plaque

d'expérimentation. Il est pratique également de pouvoir monter les deux condensateurs C1 et C2 en même temps. Ce qui manque encore, c'est une antenne. Un morceau de fil de cuivre de 1 mm de diamètre fait l'affaire. Pour le calcul de la longueur requise d'une antenne $\lambda/4$ pour la gamme de 868 MHz dans laquelle fonctionne le module LoRa : $\lambda/4 = (c_0/868 \text{ MHz})/4 = (299792458 \text{ m/s})/(8680000*4/\text{s}) = 0,08635 \text{ m} = 8,635 \text{ cm}$

Cette formule est valable dans le vide, or la propagation du signal dans le cuivre est plus lente que dans le vide. On applique donc un coefficient de

raccourcissement de 0,95, de sorte que la longueur d'antenne $\lambda/4$ sera d'environ 82 mm.

On voit sur la **fig. 3** qu'il ne faut pas grand-chose pour construire notre capteur LoRa. Aucun capteur externe ne sera d'ailleurs utilisé lors des premiers tests ; car ce qui nous intéresse, c'est comment un nœud de capteurs (aussi appelé nœud) est configuré et mis en service.

Bien préparé...

Quelques logiciels sont nécessaires : l'IDE Arduino, la bibliothèque *MCCI LoRaWAN LMIC* et le paquet de support

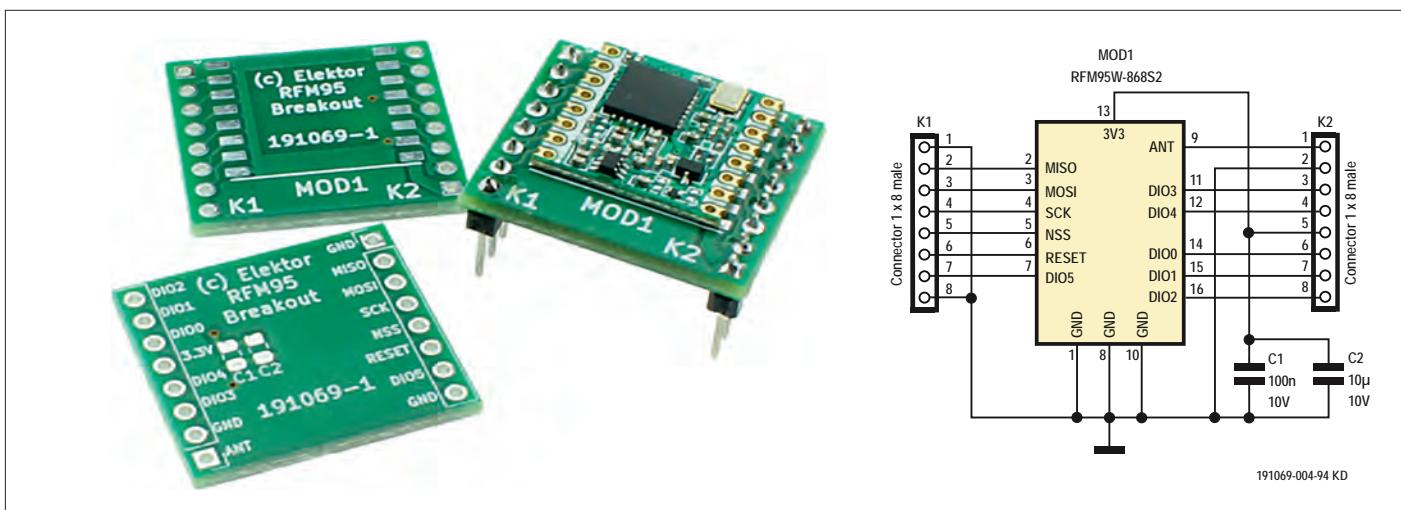


Figure 2 : Carte d'interconnexion pour le module LoRa.



LISTE DES COMPOSANTS

Condensateurs (0805) :

C1 = 100 n (10%, X7R, 10 V)
C2 = 10 µ (20%, X5R, 10 V)

Aussi :

K1,K2 = connecteur mâle 8x1, RM 2,54 mm
MOD1 = module émetteur-récepteur LoRa
RFM95W (cf. encadré)
circuit imprimé 191069-1 (cf. encadré)

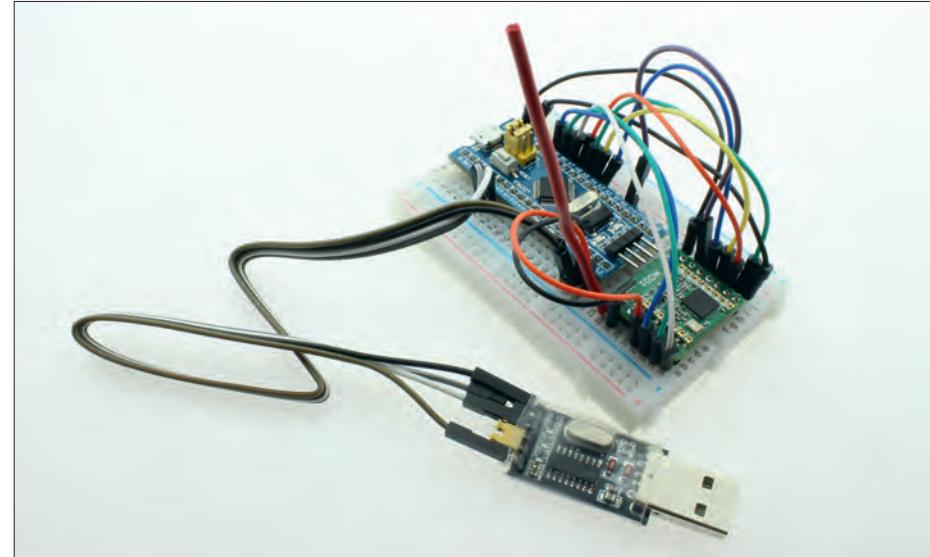
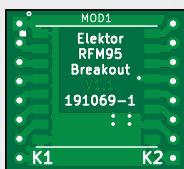


Figure 3 : Le matériel sur la plaque d'essais avec le module LoRa câblé.

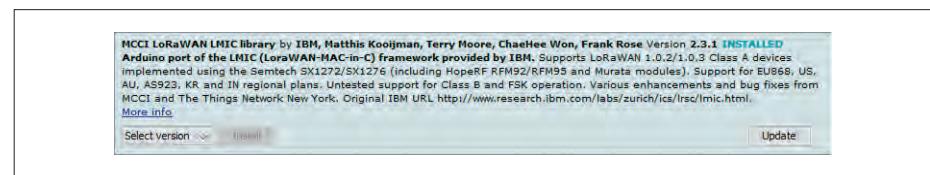


Figure 4 : Installation de la bibliothèque du LMIC.

Arduino-Core-STM32 Board.

La bibliothèque du LMIC peut être installée comme d'habitude à partir des bibliothèques (**fig. 4**). Dans le fichier de configuration du dossier de la bibliothèque du LMIC (sous Windows généralement parmi les documents de l'utilisateur) il faut indiquer dans quelle partie du monde vous vous trouvez et si c'est un module LoRa SX1276 (ce qui est le cas du RFM95). Pour l'Europe, dans le fichier \project_config\lmic_project_config.h on aura :

```
// project-specific definitions
#define CFG_eu868 1
##define CFG_us915 1
##define CFG_au921 1
##define CFG_as923 1
##define LMIC_COUNTRY_CODE LMIC_
    COUNTRY_CODE_JP /* for as923-
    JP */
##define CFG_in866 1
#define CFG_sx1276_radio 1
##define LMIC_USE_INTERRUPTS
```

Pour que l'EDI Arduino supporte la carte

Blue Pill, une URL [**4**] doit être ajoutée dans les *Préférences Fichiers*, sous *URL supplémentaires des administrateurs de carte*. Ensuite, nous pouvons rechercher «*STM32*» dans le gestionnaire de cartes et installer des coeurs *STM32*. À l'issue de quoi toutes les bibliothèques requises sont incluses. La dernière étape consiste à sélectionner la carte (**fig. 5**).

Avant d'entrer dans le premier code, la différence entre OTAA (*over the air activation = par voie hertzienne*) et APB (*activation by personalization = par personnalisation*) doit être mentionnée brièvement. Avec OTAA, le nœud LoRa rejoint activement le réseau. Il reçoit une adresse de périphérique du réseau et échange des clés avec le réseau.

Pour *The Things Network*, c'est la méthode préférée pour participer au trafic sur le réseau.

En mode APB, l'adresse de l'appareil et les touches de l'appareil sont inscrites de manière permanente dans le code. Ça facilite la mise en route du nœud, ça en affaiblit aussi la sécurité. Pour plus de détails, consultez la documentation [**5**] de *The Things Network*.

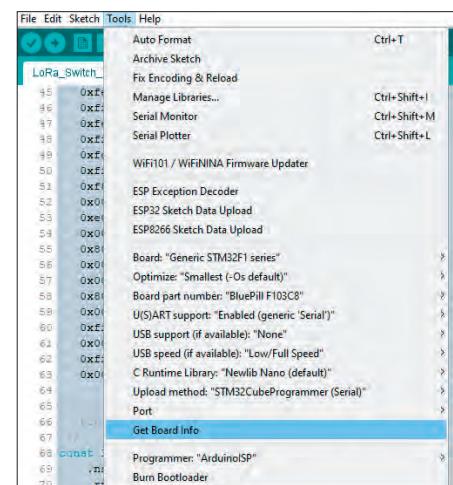


Figure 5 : L'information sur la carte.

Le programme d'exemple *ttn-abp* de la bibliothèque LMIC est utilisé comme point de départ. Le code envoie un *Hello, world!* toutes les 60 secondes et nécessite quelques ajustements concernant le matériel. À partir de la ligne 86 du croquis Arduino, l'attribution des broches est modifiée ainsi :

```

/*
 * Pin-Mapping for the RFM95 LoRa Module
 */
const lmic_pinmap lmic_pins = {
    .nss = PA4,
    .rxtx = LMIC_UNUSED_PIN,
    .rst = PA0,
    .dio = ,
};


```

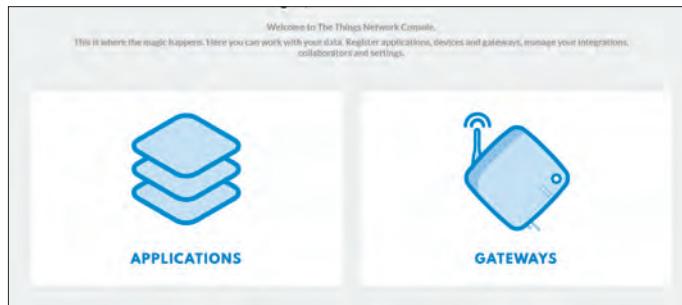


Figure 6 : Bienvenue dans le terminal de *The Things Network*.

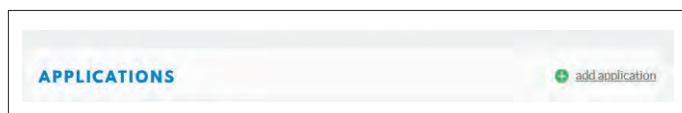


Figure 7 : *Add Application!*

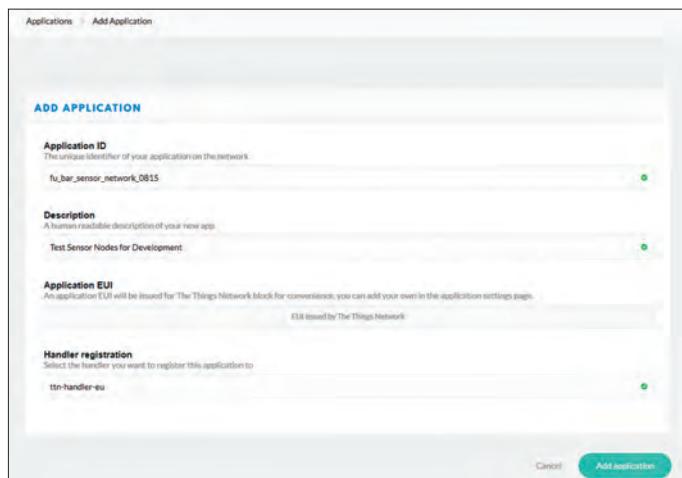


Figure 8 : Enregistrement d'une application.

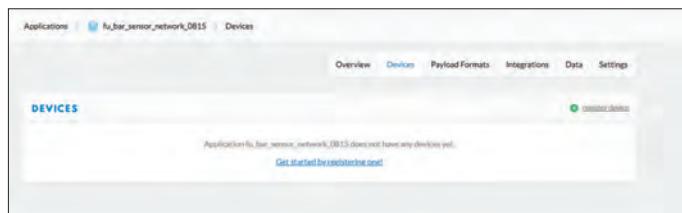


Figure 9 : Demande d'enregistrement de nœuds dans l'application.

Inscription dans *The Things Network*

En plus des broches, il faut configurer les données de réseau pour le nœud du capteur. Pour les obtenir, un nouveau nœud est d'abord créé sur *The Things Network*. Dans un premier temps, vous créez un compte d'utilisateur [6], dans lequel les nœuds sont créés et gérés. Pour qu'un utilisateur puisse garder la trace de ses nœuds, ceux-ci peuvent être regroupés sous *Applications* selon leur fonction. Après la connexion, les *applications* et les *passerelles* peuvent être gérées via le terminal (**fig. 6**).

S'il n'y a pas à proximité de passerelle utilisable, vous pouvez exploiter votre propre passerelle et la mettre à la disposition d'autres utilisateurs à proximité. La procédure sera expliquée plus tard.

Pour produire des données adéquates au nœud, une application doit d'abord être créée. Cliquez sur l'icône *APPLICATIONS* puis

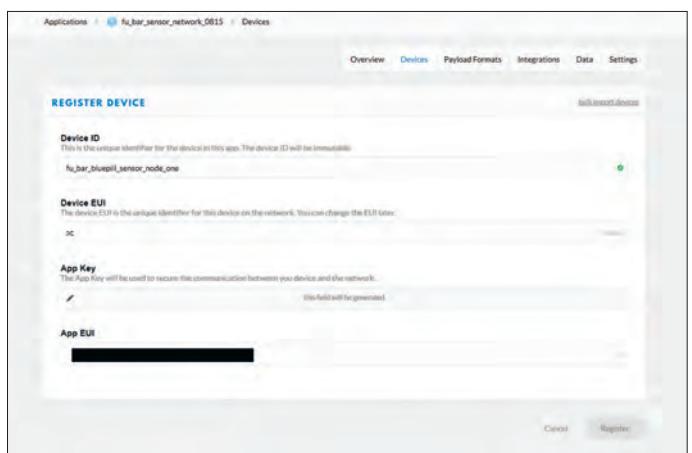


Figure 10 : Ici les données du nouveau nœud sont spécifiées.

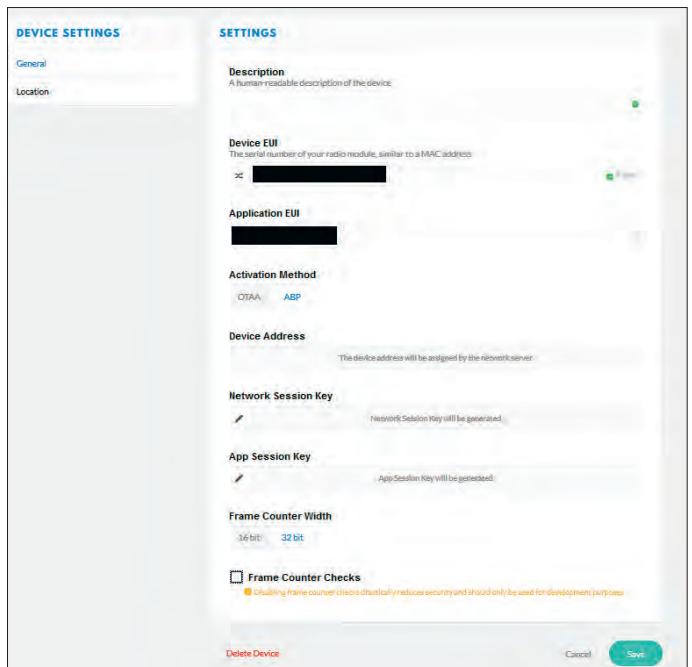


Figure 11 : Changer la méthode d'activation et désactiver (temporairement) les contrôles du compteur de trames.

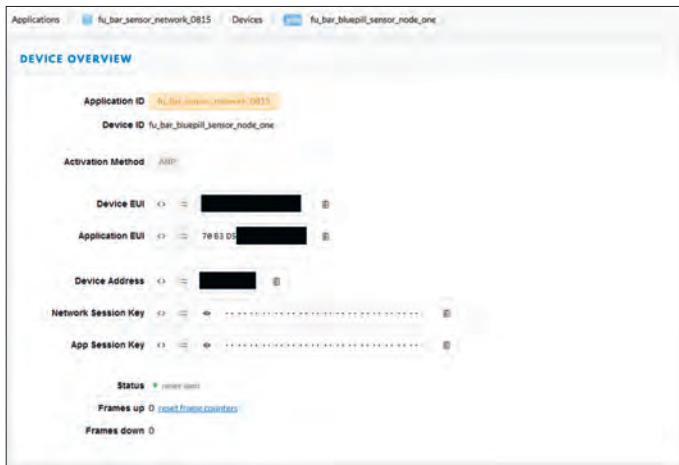


Figure 12 : Dans l’aperçu des appareils, il reste à régler correctement le format de données (LSB/MSB).

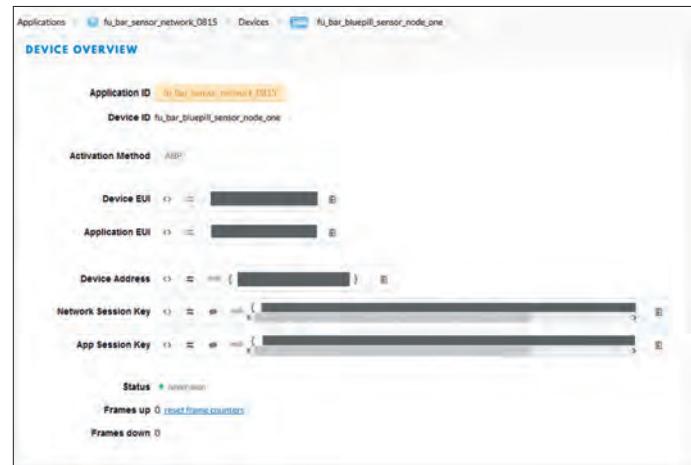


Figure 13 : L’appareil est configuré.

sur *Add application* (**fig. 7**). Remplir la fenêtre (**fig. 8**), puis le processus se termine par *Add application*. Dans l’application que l’on vient de créer, aucun nœud n’est encore enregistré. Cliquez sur *register devices* dans la fenêtre qui s’ouvre (**fig. 9**). Puis, dans la boîte de dialogue (**fig. 10**), saisissez le nom du nœud et cliquez sur le symbole de flèche sous *Device EUI* à gauche pour qu’apparaîsse le texte *this field will be generated*. En appuyant sur *Register*, vous envoyez ces spécifications à *The Things Network*, qui enregistre alors le nouveau nœud et fournit les données d’accès. Sous *Settings* (**fig. 11**), la méthode d’activation passe de *OTAA* à *ABP* et pour les premières expériences, la case *Frame Counter Checks* est décochée. Dans la vraie vie, ce réglage est risqué ! Une fois les premiers pas réussis, il faut donc, à la première occasion, remettre cette coche d’urgence ! Après l’enregistrement des réglages, de retour sur la *carte de l’appareil*, un clic sur l’onglet *Overview* permet de visualiser d’un coup d’œil toutes les données nécessaires pour le nœud (**fig. 12**). Le *Device EUI*, qui correspond à peu près à une adresse MAC [**5**], ne nous intéresse pas ici, pas plus que l’*Application EUI* attribuée par TTN, nécessaire pour la procédure OTAA. Seul le contenu des trois champs suivants est pertinent : *Device Address*, *Network Session Key* et *App Session Key*. Les formats de données sont une pierre d’achoppement : il faut indiquer correctement si les zones LSB ou MSB doivent être transférées. Le croquis nécessite, en tant que LSB, l’adresse du périphérique et la clé de session réseau et, en tant que tableau MSB, la clé de session de l’application. Les réglages peuvent être basculés en cliquant sur «< >» ; les réglages apparaissent à gauche des fenêtres (**fig. 13**).

Dans le sketch, la clé de session réseau est ensuite saisie dans NWKSKEY, la clé de session applicative dans APPSKEY et enfin l’adresse du périphérique dans DEVADDR.

```
// LoRaWAN NwkSKey, network session key
static const PROGMEM u1_t NWKSKEY[16] = { Network
Session Key };

// LoRaWAN AppSKey, application session key
static const u1_t PROGMEM APPSKEY[16] = { App Session
Key };
```

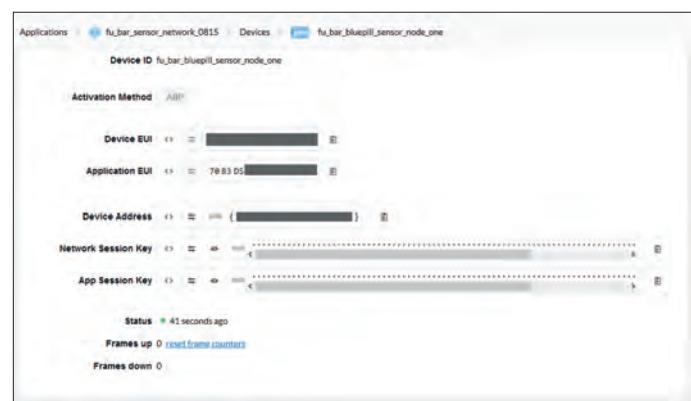


Figure 14 : Le noeud transmet !

```
// LoRaWAN end-device address (DevAddr)
// See http://thethingsnetwork.org/wiki/AddressSpace
// The library converts the address to network byte
order as needed.
static const u4_t DEVADDR = 0xDEVICE_ADDRESS;
// Change this address for every node!
```

Si une passerelle LoRa se trouve à proximité, le sketch peut être compilé et téléchargé tel quel. Dans le terminal TTN, vous voyez immédiatement que le nœud vient d’envoyer des données (**fig. 14**). Un clic sur *Data* affiche les données de l’utilisateur du nœud.

Mise en place de votre propre passerelle

Si aucune passerelle n’est à portée, vous pouvez en créer une. Il existe différentes possibilités, allant de la solution commerciale toute faite à la passerelle monocanal (non conforme à LoRaWAN) composée d’un RPi avec LoRaWAN-HAT. Une passerelle entièrement compatible est la passerelle LoRaWAN Indoor LPS8 de Dragino [**7**], disponible dans la boutique en ligne d’Elektor pour environ 120 € (voir encadré). Elle peut recevoir simultanément sur toutes les bandes de fréquences dans la gamme de 868 MHz. La variante Dragino LG02 avec deux modules LoRa est un peu limitée en réception, mais



Figure 15 : Un Raspberry Pi avec Dragino-LoRa-HAT comme passerelle.

aussi meilleur marché (80 €). Les passerelles sont basées sur la distribution Linux OpenWRT, spécialement conçue pour les routeurs, avec quelques ajustements pour fonctionner comme une passerelle LoRa.

Si vous possédez un Raspberry Pi, vous pouvez l'équiper d'un module LoRa et l'utiliser comme passerelle avec le logiciel [8] qui supporte le Dragino-LoRa-HAT [9]. Il est encore plus économique de connecter le module RFM95 directement au RPi avec quelques fils. Avec la carte d'interconnexion, c'est très facile. Pour le câblage et l'installation, vous suivez les indications [8] et la définition des broches pour le Dragino-LoRa-HAT.

Avec cette passerelle RPi, un nœud ne peut cependant être exploité qu'en mode ABP ; il n'est pas possible de transporter des messages du LoRaWAN vers le nœud. Le canal et le facteur d'étalement doivent également être réglés de façon permanente dans le noeud. Si vous voulez démarrer plus rapidement et avec moins d'embûches, vous devriez investir les quelques euros et utiliser la passerelle LG02 de Dragino, qui permet également le mode OTAA et l'envoi de données aux nœuds depuis le LoRaWAN. Pour plus de commodité (et moins de mauvaises expériences avec le RFM95 et les cavaliers), nous avons fait une razzia dans le stock d'Elektor et utilisé un Dragino-LoRa-HAT sur un Raspberry Pi 3B+ avec une image Raspbian à jour, fraîchement installée pour le test (**fig. 15**). Tout cela est rapidement mis en place, la carte SD est insérée, on branche moniteur, souris, clavier et câble Ethernet pour l'accès à l'internet, et c'est parti. Le système démarre directement sur le

bureau, mais il faut d'abord ouvrir le terminal et régler certains paramètres matériels.

Avec `sudo raspi-config` vous accédez au menu de configuration du RPi. Sous *Interface Options*, sélectionnez *P4 SPI* et activez cette interface avec un Yes. Après le redémarrage, l'interface SPI est active, ce qui permet de poursuivre la configuration.

Encore une fois, un terminal est ouvert pour télécharger certains paquets et le code source du *Packet-Forwarder*. Le code source est fourni par GitHub, donc la commande `git clone https://github.com/hallard/single_chan_pkt_fwd.git` téléchargera la dernière version. Des bibliothèques supplémentaires doivent être installées. La commande `sudo apt-get install wiringpi` installe la bibliothèque *WiringPi* dont nous avons besoin pour la compilation. Avec `cd single_chan_pkt_fwd` nous passons dans le répertoire du code source téléchargé, compilons le code source avec `make` d'abord dans un programme et l'installons avec `make install` ensuite.

La passerelle est maintenant configurée, mais la configuration des broches pour la LoRa-HAT dans le fichier *global_conf.json* reste à faire. La configuration suivante peut être utilisée pour le HAT de Dragino :

```
{
  «SX127x_conf»:
  {
    «freq»: 868100000,
    «spread_factor»: 7,
    «pin_nss»: 6,
    «pin_dio0»: 7,
    «pin_rst»: 0
  },
  «gateway_conf»:
  {
    «ref_latitude»: 0.0,
    «ref_longitude»: 0.0,
    «ref_altitude»: 10,
    «name»: «Enter your Gatewayname here»,
    «email»: «contact@whatever.com»,
    «desc»: «Dragino Single Channel Gateway on RPI»,
    «servers»:
    [
      {
        «address»: «router.eu.staging.thethings.network»,
        «port»: 1700,
        «enabled»: true
      },
      {
        «address»: «router.eu.thethings.network»,
        «port»: 1700,
        «enabled»: false
      }
    ]
  }
}
```

Avec ces réglages, la passerelle écoute sur 868,1 MHz avec un facteur d'étalement SF7. Pour pouvoir envoyer les données au TTN, il reste quelques réglages mineurs à effectuer. Il faut d'abord redémarrer la passerelle pour activer les modifications qui viennent d'être effectuées. Pour ce faire, saisissez `systemctl stop single_chan_pkt_fwd` dans le terminal, puis `systemctl start single_chan_pkt_fwd`. La passerelle est maintenant configurée comme un service ou peut être exécutée directement sous l'utilisateur root. Avec le `statut systemctl single_chan_pkt_fwd` vous interrogez le statut du service (**fig. 16**).

La passerelle est maintenant prête à fonctionner. Il suffit de la configurer dans TTN pour que les paquets puissent également être attribués et que la passerelle puisse être administrée. Cliquez sur l'icône *GATEWAYS* (**fig. 17**) puis sur *register gateway* (**fig. 18**) dans le terminal de *The Things Network*. Dans l'onglet *Passerelle de registre* (**fig. 19**), la coche doit être réglée sur «J'utilise le réacheminement de paquets Semtech».

Pour le *Gateway EUI*, l'adresse MAC du Raspberry Pi (déterminée avec `cat /sys/class/net/eth0/address`) est saisie et remplie avec **FF FF** au milieu. Par exemple, si **b8:27:eb:12:34:56** est l'adresse MAC, entrez **B8 27 EB FF FF 12 34 56** dans le champ *Gateway EUI* du terminal de *The Things Network*. Cliquez sur l'onglet *Gateway*, la passerelle est maintenant prête à l'emploi. Pour que le nœud LoRa et la passerelle fonctionnent ensemble, il faut définir une fréquence de transmission dans le code source et spécifier le facteur d'étalement. De plus, le nœud est limité à la fréquence définie dans la passerelle. L'extrait du code source montre les ajustements pour le nœud LoRa. Un exemple complet peut être téléchargé à partir de la page Elektor Labs [**10**].

```
...
// Disable link check validation
LMIC_setLinkCheckMode(0);

// TTN uses SF9 for its RX2 window.
LMIC.dn2Dr = DR_SF9;

// Set data rate and transmit power for uplink (note:
// txpow seems to be ignored by the library)
LMIC_setDrTxpow(DR_SF7,14);

//If we have a single channel one module gateway we
// need to add these lines
// Define the single channel and data rate (SF) to
// use
int channel = 0;
int dr = DR_SF7;

// Disable all channels, except for the one defined
// above.
// FOR TESTING ONLY!
for(int i=0; i<9; i++) { // For EU; for US use i<71
    if(i != channel) {
        LMIC_disableChannel(i);
    }
}

// Set data rate (SF) and transmit power for uplink
LMIC_setDrTxpow(dr, 14);

// Start job
do_send(&sendjob);

...
```

```
pi@raspberrypi:~/single_chan_pkt_fwd $ systemctl status single_chan_pkt_fwd
● single_chan_pkt_fwd.service - Lora Packet Forwarder
   Loaded: loaded (/lib/systemd/system/single_chan_pkt_fwd.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2019-11-19 15:43:11 CET; 2min 8s ago
       Main PID: 1190 (single_chan_pkt)
          Tasks: 1 (limit: 2200)
         Memory: 352.0K
        CGroup: /system.slice/single_chan_pkt_fwd.service
                 └─1190 /home/pi/single_chan_pkt_fwd/single_chan_pkt_fwd

Nov 19 15:43:11 raspberrypi systemd[1]: Started Lora Packét Forwarder.
pi@raspberrypi:~/single_chan_pkt_fwd $
```

Figure 16. Copie d'écran du terminal : Service OK !



Figure 17 : Dans le TTN, on fait les réglages de la passerelle.



Figure 18 : Enregistrement de la passerelle dans le TTN.

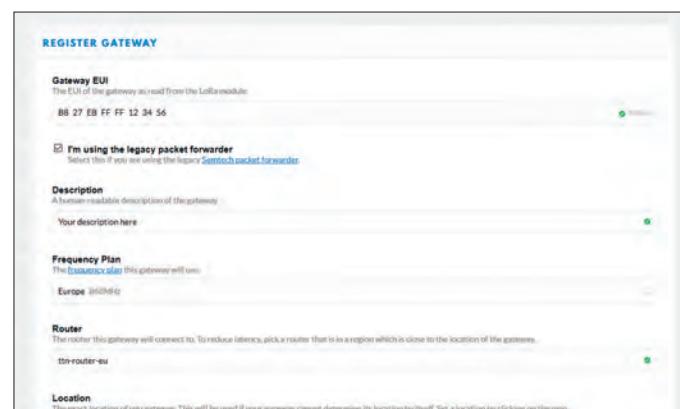


Figure 19 : Les données de la passerelle sont saisies ici.

```
// Set data rate (SF) and transmit power for uplink
LMIC_setDrTxpow(dr, 14);

// Start job
do_send(&sendjob);

...
```

Dans cet exemple, le nœud est réglé sur 868,1 MHz et SF 7. Nous utilisons également ces valeurs dans notre passerelle. La passerelle et le nœud ont maintenant leur logiciel pour que le test puisse démarrer. Après un court laps de temps, de nouveaux messages de notre nœud devraient arriver dans le terminal de TTN (**fig. 20**).

Conclusion et perspectives

Ainsi, pour faire ses premiers pas avec LoRaWAN, un nœud pour ±25 € et une passerelle basée sur un Raspberry Pi et un module RFFM95 suffisent. Cette solution reste très limitée et ne convient pas à une exploitation productive. Si après l'avoir essayé vous souhaitez faire plus avec LoRaWAN, vous devriez échanger la passerelle contre un appareil commercial entièrement compatible LoRaWAN, disponible pour moins de 120 € [7]. Avec une telle passerelle, il est possible d'utiliser tout le potentiel du LoRaWAN, y compris les messages de la passerelle vers le nœud.

Un point que nous n'avons pas encore examiné est la collecte et le traitement des données de *The Things Network*. Ce sera pour un autre article dans un prochain numéro. 

191065-02



Figure 20 : Youpie, les données du nœud arrivent !

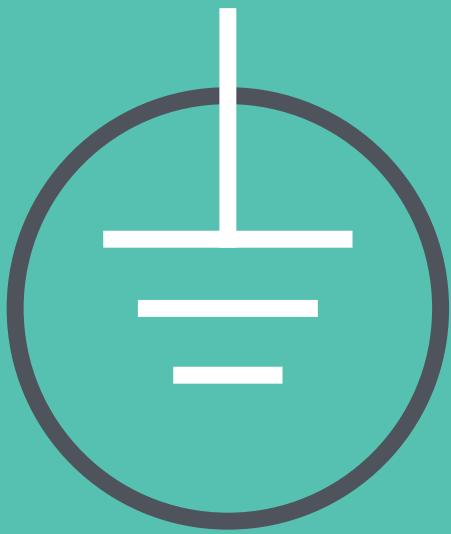
@ WWW.ELEKTOR.FR

- LoRa-Modul RFM95
www.elektor.fr/18715
- LoRa-RFM95-Breakout-Board
www.elektor.fr/191069-1
- USB/Seriell-Wandler CH340
www.elektor.fr/19151
- Dragino LPS8
www.elektor.fr/dragino-lps8-indoor-lorawan-gateway
- Dragino LG02
www.elektor.fr/dragino-lg02-dual-channels-lora-iot-gateway
- Dragino-HAT für RPi
www.elektor.fr/dragino-lora-gps-hat-for-raspberry-pi-868-mhz

Liens

- [1] Carte Blue Pill :
www.amazon.de/s?k=bluepill&__mk_de_DE=%C3%85M%C3%85%C5%BD%C3%95%C3%91&ref=nb_sb_noss_1
- [2] Module LoRa : www.elektor.de/rfm95-ultra-lora-transceiver-module-868-915-mhz
- [3] Convertisseur USB-série CH340 (3,3 V) : www.elektor.de/19151
- [4] Support de la carte : https://github.com/stm32duino/BoardManagerFiles/raw/master/STM32/package_stm_index.json
- [5] LoRaWAN : adressage et activation : www.thethingsnetwork.org/docs/lorawan/addressing.html
- [6] Créer un compte TTN : <https://account.thethingsnetwork.org/register>
- [7] Passerelle LPS8 : www.dragino.com/products/lora-lorawan-gateway/item/148-lps8.html
- [8] Logiciel HAT Gateway : https://github.com/hallard/single_chan_pkt_fwd
- [9] HAT Gateway : www.elektor.de/dragino-lora-gps-hat-for-raspberry-pi-868-mhz
- [10] La page de ce projet : www.elektormagazine.de/191065-01





NOS PRIX LA MEILLEURE ARME CONTRE LES COÛTS ÉLEVÉS

The best part of your project: www.reichelt.com/prix

Avec reichelt, optimisez votre budget.

L'efficacité de nos services logistique et informatique, développés par nos soins, ainsi que la concentration de nos achats sur des produits de qualité triés sur le volet nous permettent de vous faire bénéficier de prix très avantageux sur de petites quantités.

Alimentation de laboratoire avec sorties USB

0 - 60 V, 0 - 6 A

A des fins éducatives et de développement, cette alimentation de laboratoire offre deux sorties USB commutables avec leur propre écran LCD pour les valeurs de tension et de courant.

- puissance continue maximale : 150 W
- C.C. et C.V. (tension constante) Mode de fonctionnement une fois connecté
- Protection contre les courts-circuits, les surcharges et les surchauffes

NOUVEAU

N° de commande : PEAKTECH 6227

110,06
(91,72)



PeakTech®



Pinces à dénuder Secura 2K

pour conducteurs et torons de 0,2 à 6 mm²

- convient pour les conducteurs rigides et toronnés
- adaptation automatique à la section du câble
- Coupeur latéral intégré jusqu'à 2,5 mm²
- butée de longueur réglable



N° de commande :
JOK 20 100 30,20 (25,17)

Laser à ligne transversale à niveling automatique

avec une portée jusqu'à 30 m
En mode automatique, cette aide pratique crée toujours une croix de marquage exacte sur les surfaces droites.

- Auto-nivellement jusqu'à une orientation de base de 3° (±1°)
- jusqu'à 12 h de fonctionnement avec 2x piles AA (incluses)



N° de commande :
UT LM570R-I 50,87 (42,39)

NOUVEAU

UNI-T®



- Excellent rapport qualité prix
- Plus de 110 000 produits sélectionnés

- Livraison fiable - depuis l'Allemagne dans le monde entier

Assistance téléphonique: +33 97 518 03 04

www.reichelt.com

reichelt
elektronik - Tirer le meilleur parti de votre projet

Les réglementations légales en matière de résiliation sont applicables. Tous les prix sont indiqués en € TVA légale incluse, frais d'envoi pour l'ensemble du panier en sus. Seules nos CGV sont applicables (sur le site <https://rch.lt/CG-FR> ou sur demande). Sembables aux illustrations. Sous réserve de coquilles, d'erreurs et de modifications de prix.

reichelt elektronik GmbH & Co. KG, Elektronikring 1, 26452 Sande (Allemagne), tél. +33 97 518 03 04

PRIX DU JOUR! Prix à la date du: 3. 2. 2020



« Un pilier de l'internet ouvert »

Conversation avec Wienke Giezeman, initiateur du réseau The Things

Questions de **Mathias Claußen & Jens Nickel** (Elektor)

Pas d'IdO sans données, ni capteurs, ni transmission sans fil, ni faible consommation. Le protocole LoRaWAN permet une communication flexible entre nœuds de capteurs/actionneurs et passerelles servant de stations de base et d'interfaces avec l'internet. Les utilisateurs doivent construire leur propre infrastructure de réseau, ce que certains (écoles, particuliers,...) perçoivent comme lourd. Les réseaux existants sont souvent fermés ou inaccessibles. *The Things Network* (TTN) permet de surmonter ces obstacles. Partout dans le monde, les volontaires installent des passerelles ouvertes à d'autres utilisateurs désireux de transférer des données, tandis que les participants commerciaux peuvent également ouvrir leurs passerelles au réseau. TTN héberge également des serveurs de réseau LoRaWAN et offre du matériel abordable et beaucoup de support.

Elektor : Wienke, pourquoi avoir fondé TTN ?

Wienke : Johan Stokking et moi avons fondé *The Things Network* (TTN) en 2015 après avoir tous deux cherché à créer une nouvelle entreprise. Johan est entrepreneur en informatique depuis l'âge de 12 ans. En 2015, je viens de vendre mon entreprise précédente et nous nous sommes rencontrés pour prendre un café... c'est là que l'idée est née.

J'ai toujours été un fervent défenseur des causes ouvertes, code et internet surtout, aussi bien à titre personnel qu'en tant qu'entrepreneur. J'ai lancé la campagne *Startups for Net Neutrality* [2].

Elektor : Quels sont les avantages de LoRa ?

Wienke : Les avantages de la technique RF LoRa sont simples. La consommation est faible. Les piles tiennent 2 ou 3 ans, mais la couverture atteint plusieurs kilomètres, avec bien sûr les nuances d'usage selon les circonstances.

L'autre avantage est le protocole réseau bien géré, LoRaWAN. Grâce à lui, les appareils se connectent aux réseaux et les réseaux LoRaWAN se connectent aussi entre eux. Ainsi on passe d'une technologie comme simple outil du dernier kilomètre à une technologie Internet. Parce que tout le monde peut construire un réseau LoRaWAN, la croissance vient des réseaux que les entreprises et les individus construisent. LoRaWAN facilite leur interconnexion, créant ainsi un internet mondial des objets. Cela a été notre vision dès le début avec TTN. Ça a commencé à Amsterdam, mais rapidement ça a pris dans des centaines de villes. Nous gérons plus de 10.000 passerelles dans plus de 700 villes à travers le monde. Près de 100.000 développeurs sont actifs sur la plate-forme.



Les fondateurs de TTN, Wienke Giezeman (à gauche) et Johan Stokking (au milieu à gauche) avec leurs collègues. (photo : Rebekka Mell / The Things Network).

Elektor : Pouvez-vous imaginer des applications où l'on ne devrait absolument pas utiliser LoRa ? Et pourquoi pas ?

Wienke : Oui, beaucoup. Les techniques de l'internet des objets trouvent d'elles-mêmes leurs propres niche ou segment. Certaines campagnes de marketing essayent de faire croire qu'une technique donnée pourrait servir à tout. Je ne pense pas que ce soit le cas. Il faut une adéquation entre les exigences de l'entreprise et ce que la technologie offre. C'est donnant-donnant. Puisque la force du LoRaWAN réside dans sa faible puissance et sa longue portée, il faut accepter qu'il n'envoie que peu de données. On se réjouit de l'existence de passerelles à très faible coût ? Alors on accepte qu'il faille donc dans le nuage un serveur de réseau pour gérer le protocole. Vous pouvez construire le



Sur *The Things Conference*, les utilisateurs font des rapports sur les projets TTN, ici *Deutsche Bahn*. (photo : Rebekka Mell / The Things Network).

réseau vous-même sans mainmise du fournisseur, il faut donc accepter d'investir pour construire ce réseau.

Elektor : N'est-il pas un peu dangereux que tout dépende d'un seul fournisseur de technologie de puce ?

Wienke : Le protocole LoRa RF est la propriété de Semtech. Avec n'importe quelle pile d'IdO ou de logiciels, vous aurez des dépendances et des risques de mainmise du fournisseur. Ce cas ne fait pas exception : si vous construisez votre solution IdO, il vous appartient d'en tenir compte dans votre évaluation des risques. Si vous comparez une pile LoRaWAN typique à une pile Sigfox, NB-IoT ou LTE-M, vous découvrirez que la construction d'un produit dans LoRaWAN présente le risque le plus faible de mainmise du fournisseur. Avec notre logiciel, nous essayons de nous concentrer sur la réduction du risque de blocage des fournisseurs en utilisant des API ouvertes, des composants de base à code source ouvert et en utilisant autant que possible le standard ouvert LoRaWAN.

Elektor : TTN s'appuie sur une communauté de bénévoles qui installent des passerelles. Comment peuvent-ils s'inscrire ?

Wienke : *The Things Network* s'est construit à partir d'individus et d'entreprises qui mettent en place des applications LoRaWAN. La passerelle n'est qu'un moyen pour arriver à une fin. Le protocole est conçu pour que les réseaux partagent la capacité du réseau afin d'optimiser le terrain commun : le spectre ouvert, qui est rare. LoRaWAN permet une utilisation très efficace du spectre puisque les réseaux sont partagés et que le trafic LoRaWAN est délesté dès que possible vers des réseaux IP qui ne manquent pas.

Ainsi, le partage d'une passerelle ou d'un réseau n'a que peu à voir avec le fait d'être bénévole. Il s'agit d'optimiser la capacité et l'accès à votre propre réseau en rejoignant le réseau. Ce qui est très facile. Vous pouvez déjà obtenir une passerelle et une carte de développeur pour moins de 100 €. Connectez votre appareil et votre passerelle sur www.thethingsnetwork.org.

Elektor : Comment soutenez-vous les participants ?

Wienke : Nous fournissons un soutien par le biais de notre entreprise *The Things Industries*. Actuellement, nous aidons des centaines d'entreprises dans le monde entier à construire leurs réseaux LoRaWAN, aussi bien des organisations ferroviaires nationales, des sociétés maritimes mondiales que des grandes sociétés immobilières. Cela se fait par le biais d'une proposition payante où les entreprises obtiennent du soutien et des outils d'entreprise pour gérer leur réseau. Nous sommes assez fiers de notre aptitude à mettre en place un modèle commercial sur une plate-forme ouverte, sans compromettre ni le succès commercial ni notre foi dans les technologies ouvertes.

Elektor : Comment gagner de l'argent ? Y a-t-il un modèle commercial derrière TTN ?

Wienke : Nous gagnons de l'argent en vendant des services et des produits par le biais de notre société *The Things Industries*. Actuellement le TTN ne nous rapporte rien. Nous chercherons des moyens de financer plus équitablement les frais généraux au fil du temps, mais l'équilibre est parfait pour le moment et nous sommes en mesure de le financer par nos activités commerciales.

Elektor : Les utilisateurs peuvent-ils avoir leurs propres serveurs de réseau et également rejoindre le réseau ?

Wienke : Oui, pour les entreprises, nous avons notre nouveau serveur réseau LoRaWAN appelé *The Things Enterprise Stack*. Cela vous permet de mettre en place un serveur réseau LoRaWAN professionnel avec un contrôle complet de votre propre qualité de service et de votre niveau de sécurité. En attendant, vous pouvez activer le réseau de pair à pair avec les autres réseaux ouverts pour une efficacité mutuelle du réseau et du spectre. Nous avons également une version à code source ouvert qui vous permet de mettre en place le réseau vous-même dans son intégralité.

Elektor : Combien de passerelles faut-il pour une ville comme Amsterdam ?

Wienke : Difficile de répondre. Tout dépend de ce qui se trouve entre les deux. L'avantage est le faible prix des passerelles, 500 à 700 € pour une version extérieure et 60 à 100 € pour une version intérieure. Cela permet une approche allégée à laquelle vous ajoutez de la connectivité selon vos besoins.

Pour donner une réponse au moins approximative : disons que pour une ville comme Amsterdam, il faudrait 100 passerelles extérieures et que vous ajoutiez 300 passerelles intérieures pour combler les lacunes. Le coût de passerelle serait d'environ 50.000 € en tout, un chiffre modeste à cette échelle. Si l'on considère ce qui se passe actuellement dans le monde avec les passerelles et les réseaux connectés, ce coût se répartirait sur de nombreuses applications.

Elektor : Y a-t-il des passerelles à déconseiller avec TTN ?

Wienke : Toutes les passerelles qui utilisent les puces de passerelle Semtech et leur conception de référence sont compatibles. Pour l'instant, je ne connais pas de portail particulièrement

The Things Indoor Gateway - Spec



Une passerelle intérieure facile à utiliser.

ment mauvais. La plupart du temps, le coût est lié à la valeur qu'ils apportent. Une simple passerelle intérieure est moins chère, car elle crée un réseau plus petit.

Elektor : Prévoyez-vous également la prise en charge d'appareils de classe B ?

Wienke : Oui, il y a ça dans notre dernier logiciel.

Elektor : Offrez-vous aussi l'itinérance de messages vers d'autres serveurs de réseau et inversement ?

Wienke : Oui, pour cela nous avons lancé le concept du *Packet Broker* (www.packetbroker.org). La beauté de LoRaWAN réside dans la possibilité de construire un véritable internet des objets de passerelles et de réseaux interconnectés. Avec le *Packet Broker*, nous essayons de simplifier l'échange de trafic entre réseaux.

Elektor : Y a-t-il des partenariats avec des entreprises ?

Wienke : Regardez sur www.thethingsnetwork.org et cliquez sur *Market Place*, vous verrez tous nos partenaires. La force de l'écosystème LoRaWAN est dans leur grand nombre.

Elektor : Pouvez-vous nous parler du partenariat avec la société néerlandaise KPN ?

Wienke : KPN est un opérateur téléphonique historique qui installe des passerelles LoRaWAN à travers les Pays-Bas et facture environ 30 à 40 € par an et par appareil pour le faire connecter.

Elektor : Actuellement, en Europe, c'est la bande de 868 MHz qui est utilisée. Est-il prévu de l'étendre à la bande 433 MHz ?

Wienke : Pas pour le moment.

Elektor : Des puces Semtech arrivent pour la bande 2,4 GHz. Une intégration dans l'infrastructure actuelle serait-elle possible ?

Wienke : Aucun commentaire pour l'instant.

Elektor : Pouvez-vous nous parler de LoRa dans l'espace ?

Wienke : C'est super excitant ! Nous sommes fans de l'espace *Lacuna* en raison de leur supériorité technique dans l'envoi du message LoRaWAN directement de la terre vers un satellite. Gardez un œil sur cette société dès que les utilisateurs de *The Things Network* et *The Things Industries* pourront utiliser ce réseau mondial.

Elektor : Quelle est votre contribution à l'Alliance LoRaWAN ?

Wienke : Nous sommes un membre très actif qui contribue à la fois sur le plan technique et sur le plan marketing. Nous organisons les plus grandes conférences mondiales de LoRaWAN à Amsterdam, Hyderabad, Adelaïde, en Slovénie et au Royaume-Uni. Nous contribuons également au comité technique de l'Alliance LoRa en construisant par exemple la première solution de mise à jour de logiciel par voie hertzienne.

Elektor : Quels sont les plans pour une meilleure couverture du réseau ? Que faites-vous pour l'améliorer ?

Wienke : Nous lançons notre dernier serveur de réseau et le mettrons à contribution pour le site thethingsnetwork.org. Nous constatons une solide croissance annuelle de 100 %. Et même une croissance plus rapide des appareils connectés.

Elektor : Quel est l'avenir de LoRaWAN selon vous ?

Wienke : L'avenir de LoRaWAN dépend de la façon dont la partie WAN fonctionne. Comme je l'ai dit, il existe de nombreuses solutions d'IdO du dernier kilomètre qui peuvent amener vos données de A à B. Il n'y en a qu'une qui, grâce à une norme ouverte, permet aux entreprises de construire des appareils, des réseaux et des applications qui sont tous interopérables et d'étendre leur marché adressable.

Commercialement, le fait que LoRaWAN résout des problèmes dans toutes les industries, agriculture, immobilier, comptage, villes intelligentes, logistique, maritime, etc. plaide pour sa pérennité. La force de l'écosystème déterminera la vitesse à laquelle il se déploiera.

C'est pour cette raison que le thème de notre prochaine conférence LoRaWAN, www.thethingsconference.com, est l'interopérabilité. Nous montrons comment des partenaires et même des concurrents coopèrent pour faire avancer un réseau LoRaWAN interopérable à l'échelle mondiale. ►

(191193-03 VF)

Liens

- [1] [The Things Network](http://www.thethingsnetwork.org)
www.thethingsnetwork.org/
- [2] [Startups for Net Neutrality](http://www.startupsfornetneutrality.eu)
www.startupsfornetneutrality.eu

carte Meadow F7

Une carte pour développeurs .NET

Tam Hanna (Allemagne)



La variété des cartes de calculateurs de traitement (ou de *process*) disponibles pour le développement de nouvelles applications est étonnante, mais si vous voulez programmer en *Visual Basic* ou en C#, le choix se réduit. C'est comme ça qu'on tombe sur la nouvelle carte *Meadow F7* qui utilise un STM32F777 pour l'exécution du *runtime* .NET, tandis qu'un module ESP32 se charge du Wi-Fi.

(Source : Wilderness Labs)

Pendant son mandat de vice-président chez *Xamarin*, Bryan Costanich était occupé à porter l'environnement .NET sur Android et iOS. Il a ensuite acheté les droits de propriété intellectuelle à une société dirigée par Chris Walker et a créé *Wilderness Labs*. Leur dernier produit est l'ordinateur de traitement *Meadow F7*. Son ambition est de fournir aux développeurs .NET un accès de première classe à l'écosystème de l'IdO. Du point de vue de l'architecture, ça ressemble à ce qu'on obtenait jusqu'à présent en combinant Raspberry Pi ou Orange Pi et un cœur de processeur en temps réel. Un STM32F777 cadencé à 216 MHz se charge du *runtime* .NET, un module ESP32 de la connexion Wi-Fi.

Allons-y, le bonheur est dans le pré

Telle qu'elle est livrée, la carte *Meadow F7* (*meadow* signifie pré ou pâturage en anglais) se présente avec un système d'exploitation soit obsolète, soit pas installé du tout. L'installation se fait avec le chargeur d'amorce STM. Téléchargez [1] le système d'exploitation composé de deux fichiers. Il vous faut ensuite des DFU-utils [2].

La documentation de la carte indique que pour lancer le chargeur de démarrage il faut appuyer sur le bouton de r-à-z. Maintenez ce bouton enfoncé pendant que vous connectez la carte à votre poste de travail *Windows* à l'aide d'un câble micro USB. Nous utilisons [list](#) pour obtenir le numéro de série de l'ordinateur de traitement :

```
C:\dfu-util-0.9-win64>dfu-util --list  
dfu-util 0.9
```

```
...  
Cannot open DFU device 0483:df11
```

Oups - ça fonctionne sous d'anciennes versions de Windows, mais pas sous Windows 10 (voir l'explication [3]). Pour contourner le problème, suivre les instructions d'initialisation du pilote. Après cela, le numéro de série peut être déterminé ainsi :

```
C:\dfu-util-0.9-win64>dfu-util --list  
dfu-util 0.9  
...  
Found DFU: [0483:df11] ver=2200, devnum=4,  
cfg=1, intf=0, path="5-3", alt=3, name="@  
Device Feature/0xFFFF0000/01*004 e",  
serial="346B38733536"
```

Dans la liste de *Windows*, les cartes *Meadow* reconnues n'apparaîtront pas en un seul exemplaire, mais comme quatre périphériques. Pour nous, seul le numéro de série importe, nous en avons besoin dans l'étape suivante pour charger noyau et *runtime*. Pensez à modifier les lignes de commande présentées ici en accord avec votre propre situation. Assurez-vous

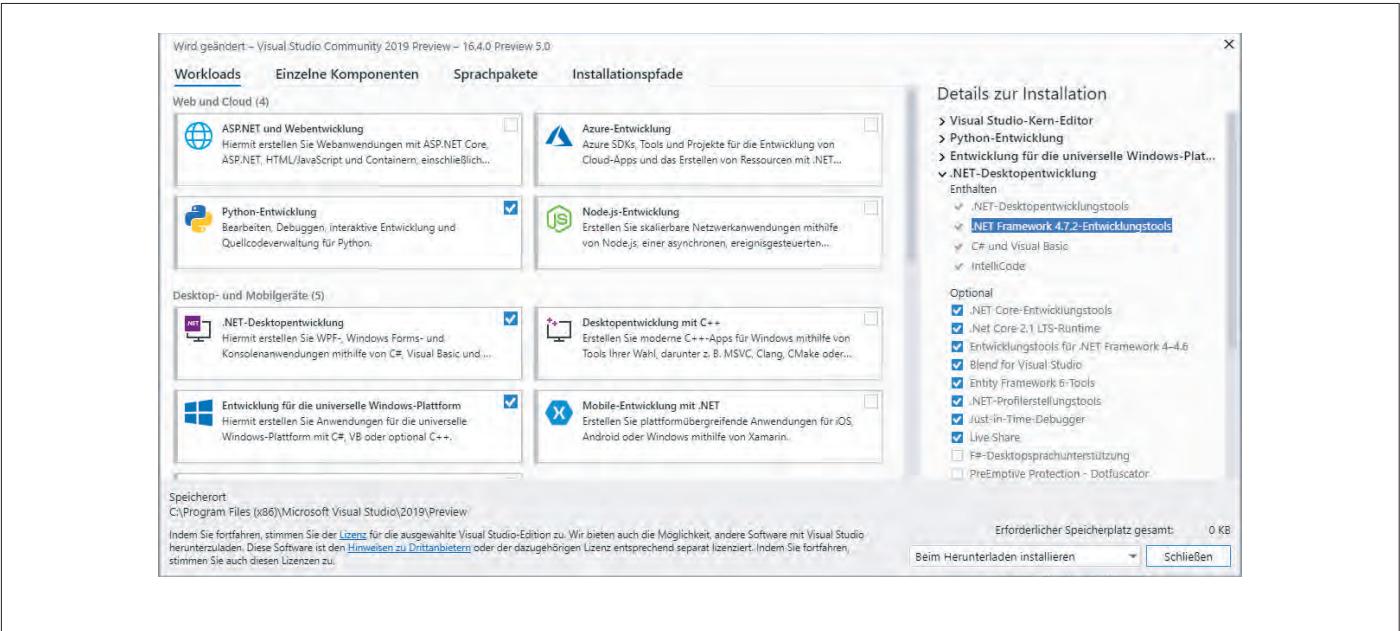


Figure 1. Meadow n'est guère amusant sans les outils de développement du SDK de .NET Framework 4.7.2.

de saisir les adresses hexadécimales sans faute de frappe et placez dans le bon dossier les deux fichiers `Meadow.OS_Kernel.bin` et `Meadow.OS_Runtime.bin` :

```
C:\dfu-util-0.9-win64> dfu-util -a 0 -S 346B38733536
-D Meadow.OS_Kernel.bin -s 0x08000000
dfu-util 0,9
C:\dfu-util-0.9-win64> dfu-util -a 0 -S 346B38733536
-D Meadow.OS_Runtime.bin -s 0x08040000
dfu-util 0,9
```

Pour garantir un bon démarrage, appuyez sur le bouton RST : la LED clignotera un instant. Comme environnement de développement, j'ai utilisé *Visual Studio* dans sa version communauté libre 2019.8. Pour l'installation, les composants mentionnés sur la **fig. 1** sont sélectionnés et téléchargés par l'assistant

d'installation, disponible dans le menu de démarrage sous *Visual Studio Installer*.

Ensuite, démarrez *Visual Studio* comme d'habitude. Si l'EDI ne démarre pas avec un projet déjà ouvert, le nouvel assistant de démarrage peut être désactivé en cliquant sur l'option *Continuer sans code*.

Wilderness Labs fournit le SDK actuel sous la forme d'un *plug-in Visual Studio*. Cliquez sur *Extensions Manage Extensions* pour charger l'assistant du *plug-in*. Nous passons ensuite à la section *Online* (**fig. 2**), et cherchons la chaîne *Meadow*.

Après un redémarrage obligatoire, il y aura un nouveau modèle appelé *Meadow Application*, qui vous servira de base pour vos propres expériences. Créez un nouveau programme appelé *ElektorSample* pour visualiser le code de l'exemple de LED clignotante fourni avec le chargeur de démarrage. Le code dans le fichier `MeadowApp.cs` devrait être explicite.

Les développeurs venus d'*Arduino* doivent tenir compte de l'absence de structure en boucle du *framework* lui-même. L'exemple ci-dessus initialise simplement les deux méthodes à partir du constructeur : une boucle sans fin est implémentée dans les `BlinkLeds` :

```
public MeadowApp()
{
    ConfigurePorts() ;
    BlinkLeds() ;
}
```

Tests : démarrage et ping

Si un ordinateur de traitement doit assurer des tâches difficiles en temps réel, le système d'exploitation doit pouvoir garantir des temps de réponse. Les systèmes basés sur des langages gérés tels Java ou C# sont généralement en peine à cet égard. Pas seulement à cause de faiblesses dans l'exécution, mais toutes les autres tâches sont bloquées tant que dure le

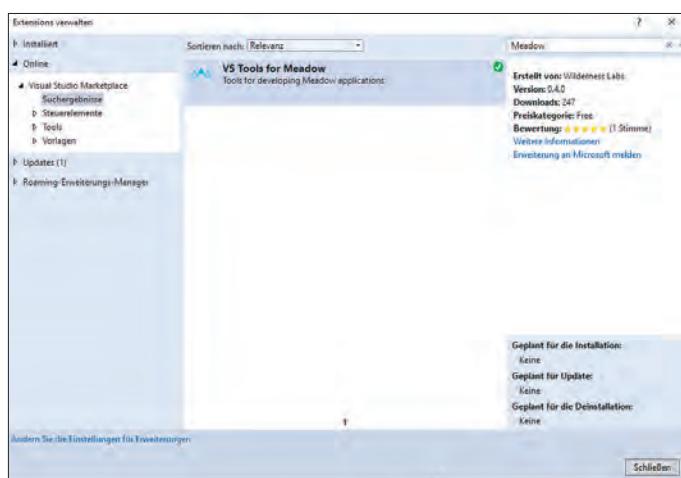


Figure 2. Sur demande, *Visual Studio* obtient automatiquement les mises à jour sur l'internet.

nettoyage de la mémoire (*Garbage Collector*). C'est pourquoi notre petit programme de test se passe presque entièrement d'allocation dynamique de mémoire. Cependant, si une tâche de commande alloue et libère beaucoup de mémoire, le risque de mise en marche du nettoyage de la mémoire augmente.

Pour tester le comportement de la carte *Meadow F7*, nous pouvons écrire une routine qui bascule simplement une broche de sortie pour produire une certaine forme d'onde de sortie.

Pour cela, les `ConfigurePorts` et les `BlinkLeds` sont utilisés comme suit :

```
public class MeadowApp : App<F7Micro, MeadowApp> {
    IDigitalOutputPort myOut;

    public MeadowApp()
    {
        public void ConfigurePorts() {
            Console.WriteLine("Creating Outputs...");
            myOut= Device.
CreateDigitalOutputPort(Device.Pins.D05);
        }

        public void BlinkLeds() {
            var state = false;

            while (true) {
                myOut.State = true;
                myOut.State = false;
                myOut.State = true;
                myOut.State = false;
            }
        }
    }
}
```

Pour interagir avec des éléments physiques, le logiciel *Meadow* utilise des classes d'abstraction. Notre port numérique est créé, p. ex. par une interface du type `IDigitalOutputPort` — si vous deviez proposer une extension qui expose également les broches GPIO, vous pourriez (avec un pilote approprié) déplacer le code entre le périphérique «ordinaire» et le nouveau périphérique.

Pour examiner la forme d'onde, on a utilisé un oscilloscope numérique à mémoire Tektronix TDS754D modifié pour obtenir une bande passante de 1 GHz. C'est du matériel importé des États-Unis, remis à neuf, fourni avec un LCD en option, à un prix raisonnable.

Puisque *Meadow F7* est vu par *Windows* comme un simple port COM, il faudra donner un coup de pouce à *Visual Studio* pour l'installation. Dans la première étape, cliquez sur *View Other Windows Meadow* (ou *Ctrl+Maj+M*) pour activer la fenêtre de sélection de l'appareil appelée *Meadow Device Explorer*. Sélectionnez votre *Meadow* pour lancer un débogage. Chaque fois que le programme est lancé, la fenêtre de la **fig. 3** s'affiche, que vous fermez en appuyant sur la touche Entrée.

Après l'installation, connectez l'oscillo à la broche D05 et observez la forme d'onde illustrée de la **fig. 4**. Il est normal que pendant l'exécution le *runtime* donne des erreurs de classes de donnée.



Figure 3. Pour fermer cette fenêtre, frappez la touche Entrée.

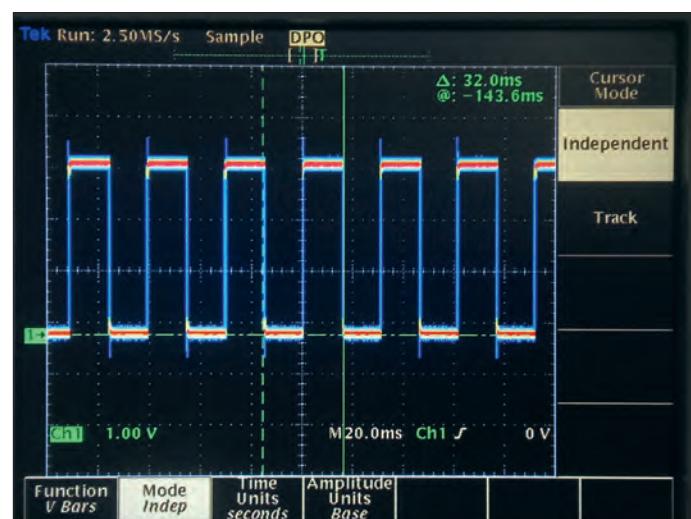


Figure 4. La commutation d'une broche de sortie prend environ 32 ms.

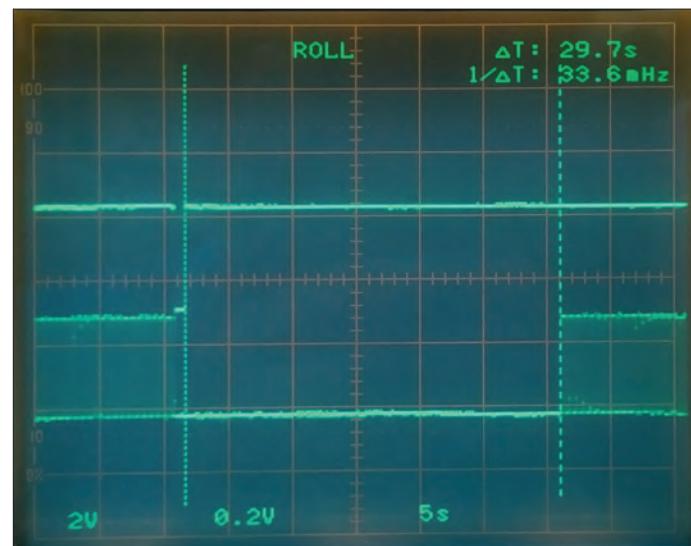


Figure 5. Après une remise à zéro (trace supérieure), il faut presque 3 s pour repartir.

Les différences de temps entre les bords de la forme d'onde sont très faibles. Cela indique que ce qui prend le plus de temps, c'est de changer l'état de la broche – un point d'optimisation possible.

La ligne de r-à-z n'est pas reliée seulement à un bouton-poussoir, mais aussi à une broche du long connecteur (la première connexion à côté du convertisseur de tension). Si votre oscilloscope a une option Roll-Mode et que vous avez de

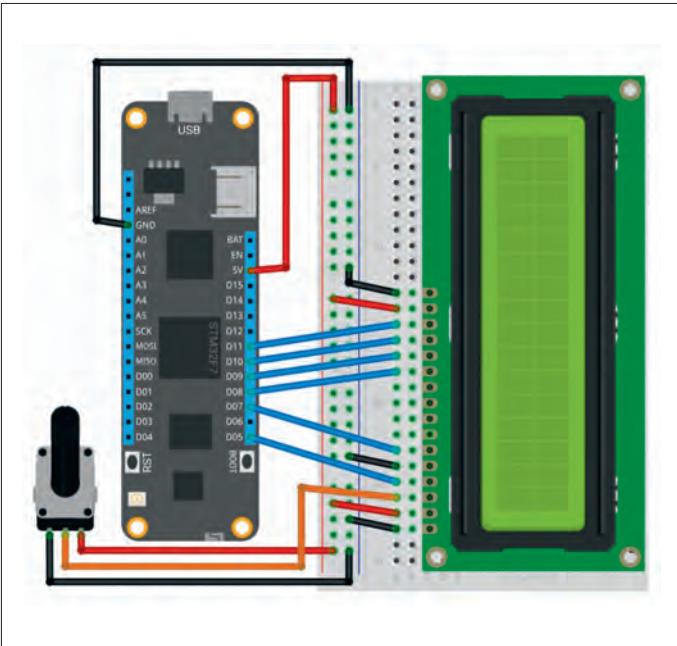


Figure 6. Branchez l'afficheur LCD sur une carte enfichable (Source : Wilderness Labs [5]).

la patience, vous pouvez utiliser l'initialisation pour voir que la carte met presque 30 s avant d'exécuter le programme (**fig. 5**) ! Pour les premiers tests, l'idéal est de connecter un simple afficheur (**fig. 6**) à Meadow F7 pour émettre quelques messages. Le paradigme de conception des classes d'abstraction décrites ci-dessus se poursuit également avec `CharacterDisplay`. Nous devons tout d'abord ajouter une autre instance à la classe `MeadowApp` qui sera chargée de communiquer avec l'afficheur :

```
public class MeadowApp : App<F7Micro, MeadowApp> {
    CharacterDisplay myDisplay;
```

La structure des pilotes matériels est toujours la même sous `Meadow`. Le constructeur accepte d'abord une référence à un objet de périphérique – celui qui transfère le périphérique demande au pilote d'utiliser le matériel de sortie de l'ordinateur de traitement. Comme le Kinect SDK de *Microsoft*, c'est là une façon d'augmenter la flexibilité (théoriquement, on pourrait aussi procéder par extension de GPIO).

À l'étape suivante apparaît un groupe de paramètres nommés pour décrire les broches de sortie utilisées. Les valeurs du `Enum Device.Pins` y sont transférées – dans ce groupe, un champ de bits spécifique correspond à chaque périphérique matériel du processeur STM32, ce qui facilite la commande :

```
public void ConfigurePorts()
{
    Console.WriteLine("Creating Outputs...");
    myDisplay = new CharacterDisplay(
        Device,
        pinRS: Device.Pins.D05,
        pinE: Device.Pins.D07,
        pinD4: Device.Pins.D08,
```

```
pinD5: Device.Pins.D09,
pinD6: Device.Pins.D10,
pinD7: Device.Pins.D11,
rows: 4, columns: 20
);
```

}

À ce stade, remarquons que *Visual Studio* ne peut pas résoudre la référence à la classe `CharacterDisplay`. Ceci est dû à la livraison modulaire de la structure (*framework*) – cliquez avec le bouton droit sur *Search (Rechercher)* dans l'explorateur de solutions et choisissez le gestionnaire de paquets NuGet (*package manager*). Ensuite, cherchez la chaîne `Meadow*Character` – l'astérisque se substitue à n'importe quel signe. Le paquet `Meadow.Foundation.Displays.LCD.CharacterDisplay` vaut la peine d'être installé, ce qui peut être fait comme une installation normale de *NuGet-Package*.

Il ne manque que la fonction d'édition réelle qui envoie le texte à l'afficheur. *Wilderness Labs* utilise l'infrastructure .NET ; la syntaxe de l'instruction `WriteLine` peut vous sembler familière. Le paramètre numérique supplémentaire définit la ligne sur laquelle la chaîne livrée doit être affichée. Si vous passez la valeur '1', elle apparaît sur la deuxième ligne en partant du haut :

```
public void BlinkLeds()
{
    var state = false;

    while (true)
    {
        myDisplay.WriteLine("Hello Elektor", 1);
        System.Threading.Thread.Sleep(1000);
    }
}
```

Téléchargez à nouveau le programme dans `Meadow` et voyez comment il apparaît à l'écran. Si vous utilisez le module inclus dans le kit de modification et que l'écran reste vide, modifiez le contraste de l'afficheur ; s'il est trop faible, vous ne voyez rien ! Remarquez aussi comme l'écriture à l'écran est lente – à l'heure de mettre sous presse cet article, il restait de la marge pour optimiser le code.

Gadget est mort... vive le gadget !

Quand *Microsoft* lançait en 2011 sa plate-forme .NET *Gadgeteer* (aujourd'hui disparue), son intention était de faciliter le prototypage aux développeurs mal à l'aise avec le matériel. Le système se composait d'une carte mère à microprocesseur prêté à l'emploi et d'une gamme de modules de capteurs et d'E/S qui s'y connectaient à l'aide de câbles plats.

L'esprit est le même chez *Wilderness Labs*. L'offre *Meadow F7 Micro Development Kit w/Hack Kit Pro* (**fig. 7**), disponible en précommande [6], comprend une carte Meadow F7, deux cartes de prototypage, un afficheur LCD alphanumérique 4x20 de grande qualité, un assortiment varié de composants actifs et passifs, d'actionneurs et de capteurs ainsi qu'une carte de développement en bois MDF (reconstitué).

En plus de cela, il y a une bibliothèque de pilotes très étendue.

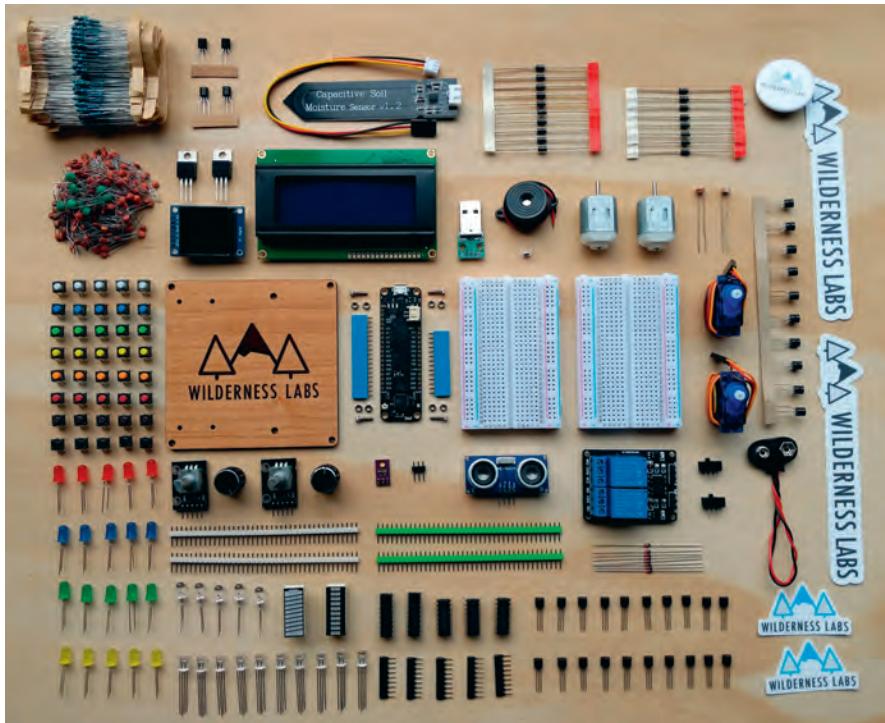


Figure 7. Le kit de développement est farci de petits cadeaux (Source : Wilderness Labs [6]).

Au moment où je boucle cet article, le pilote pour l'afficheur LCD en couleur (de qualité) n'est pas prêt, mais le kit lui-même devrait être disponible dès mars 2020. La gamme de capteurs apparaît dans la liste du matériel de *Wilderness Labs* [4].

Un mendiant n'a pas le choix, il prend ce qu'on lui donne. Ceux qui veulent utiliser .NET dans une application embarquée

n'ont le choix pour l'instant qu'entre la carte désuète *NetDuino* et la carte *Meadow F7*. Pour les applications compatibles avec les performances d'entrées/sorties de la carte, on trouve une vaste bibliothèque de pilotes qui facilitent l'assemblage et les tests rapides des prototypes - l'avenir du gadget s'annonce radieux ! ◀

(191190-03)



@ WWW.ELEKTOR.FR

→ Livre : Visual Basic pour des applications d'électronique (en anglais)
www.elektor.fr/visual-basic-for-electronics-engineering-applications-e-book

Liens

- [1] Meadow F7 : http://beta-developer.wildernesslabs.co/Meadow/Getting_Started/Deploying_Meadow/
- [2] DFU-utils : <http://dfu-util.sourceforge.net/releases/dfu-util-0.9-win64.zip>
- [3] Bogue de Windows :
www.hanselman.com/blog/HowToFixDfuutilSTMWinUSBZadigBootloadersAndOtherFirmwareFlashingIssuesOnWindows.aspx
- [4] Péphérique : <http://developer.wildernesslabs.co/Meadow/Meadow.Foundation/Peripherals/>
- [5] Character Display :
<http://beta-developer.wildernesslabs.co/docs/api/Meadow.Foundation/Meadow.Foundation.Displays.Lcd.CharacterDisplay.html>
- [6] Meadow Kit /w Hack :
<https://store.wildernesslabs.co/collections/frontpage/products/meadow-f7-micro-development-board-w-hack-kit-pro>

multitâche en pratique avec l'ESP32 (2)

Priorités des tâches

Warren Gay (Canada)

Avec un microcontrôleur comme plaque tournante d'un projet, les développeurs sont souvent confrontés à la nécessité d'exécuter plusieurs tâches à la fois. L'ESP32 et l'EDI Arduino facilitent la programmation de tâches, car le fameux FreeRTOS est déjà intégré dans les bibliothèques de base [1]. Ici nous nous intéressons aux priorités de tâches.

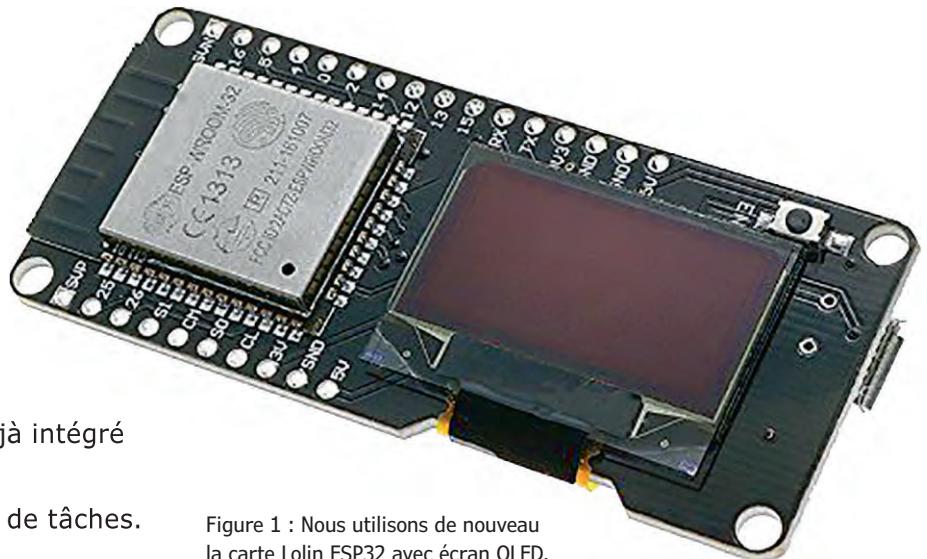


Figure 1 : Nous utilisons de nouveau la carte Lolin ESP32 avec écran OLED.

Dans l'implémentation pour l'ESP32 de l'ordonnanceur de FreeRTOS, les tâches sont exécutées selon leur priorité. La priorité est affectée lorsque la tâche est créée, mais peut être modifiée ensuite. Les tâches de rang élevé sont traitées en premier pour l'Unité Centrale (UC) configurée, tandis que les tâches de priorité nulle sont traitées en dernier. Le concept de priorité d'exécution peut vous être familier, mais l'ordonnanceur en temps réel de FreeRTOS fonctionne différemment de ce que vous connaissez de Linux ou Windows. Cet article étudie la différence au moyen d'une démonstration. L'implémentation pour l'ESP32 comporte 25 niveaux de priorité, de 0 à 24. Par défaut, les fonctions Arduino `setup()` et `loop()` s'exécutent au niveau de priorité 1 (ces fonctions font partie de la même tâche principale [1]).

Vive la différence

Comment l'ordonnancement des tâches peut-il être différent ? Sur un système Linux p. ex., la priorité affecte l'urgence relative du processus ou du *thread*. Même un processus de faible priorité obtient un peu de temps d'UC – ce qui allonge son temps d'exécution. Mais le processus **finit toujours par s'exécuter**. Voilà la différence. Dans un système en temps réel comme FreeRTOS, l'ordonnanceur **ne garantit pas** que les tâches de moindres priorités soient un jour exécutées. Si vous avez des tâches p. ex. de priorité 9 toujours *prêtes à s'exécuter*, aucune tâche de priorité 8 ou moins ne sera séquencée sur cette même UC. Les tâches de priorité 9 privent d'UC toutes les tâches de priorité inférieure.

Prête à s'exécuter

Il est important de comprendre ce que *ready* (prête) signifie pour FreeRTOS. Une tâche est *ready* lorsque qu'elle n'est pas bloquée en attente de quelque chose, que ce soit un événement,

une entrée poussée dans une *file d'attente* ou le déverrouillage d'un *mutex* (nous y reviendrons plus tard dans cette série). Une tâche prête à s'exécuter est insérée dans la *ready list* de l'ordonnanceur et sera exécutée une fois son tour venu. Comme c'est une liste classée par priorité, les tâches de plus hautes priorités sont traitées en premier. Les tâches d'égales priorités sont séquencées avec une approche *Round-Robin* (tourniquet). Trois tâches « *ready* » de priorité 9 (a, b et c) vont se succéder :

- tâche9a
- tâche9b
- tâche9c
- tâche9a
- tâche9b
- etc.

À moins qu'elles ne se bloquent, cela continue indéfiniment. Seule une tâche de plus haute priorité peut les préempter. Par exemple, la tâche ESP32 de haute priorité nommée *idc1* (pour l'UC 1), peut préempter vos tâches de priorité 9 pour effectuer certaines opérations. Dès qu'*idc1* quitte à nouveau l'état *ready*, les tâches de priorité 9 reprennent de là où elles s'étaient arrêtées.

Voici quelques exemples de sortie de l'état *ready* par une tâche FreeRTOS :

- mise en sommeil ou délai pour un moment (attente d'une temporisation) ;
- attente d'un *mutex* ou d'un sémaphore ;
- attente d'un message depuis une file d'attente vide ;
- attente d'insertion d'un message dans une file d'attente pleine ;
- attente d'un événement FreeRTOS ou d'un groupe d'événements ;

- attente de la fin d'une E/S ;
- suspension (soit par la tâche elle-même, soit par une autre tâche).

L'attente de réception d'un message depuis une file vide est une situation de blocage. Si la file est vide, la tâche n'a rien à faire, donc l'ordonnanceur retire la tâche de la *ready list* et en cherche d'autres à exécuter. Seules les tâches de la *ready list* seront traitées. Si aucune tâche n'est trouvée, c'est la tâche inactive du système qui est exécutée à la place.

Notez que l'appel à la fonction `taskYIELD()` n'est pas une des raisons listées. Lorsqu'une tâche cède la main, soit après avoir épousé sa tranche de temps soit volontairement par appel à `taskYIELD()`, le contrôle repasse à FreeRTOS pour qu'il puisse choisir une autre tâche à exécuter pour la tranche suivante. Céder la main n'est pas bloquant, car ces tâches restent prêtes à s'exécuter et retrouveront l'UC dès la prochaine occasion.

Modifications d'ESP-IDF pour FreeRTOS SMP

FreeRTOS a été conçu pour des microcontrôleurs mono-UC. Comme l'ESP32 est un dispositif à double-UC (sauf l'ESP32-S2), Espressif a adapté le composant ordonnanceur. À l'examen, les UC ESP32 suivantes sont présentes :

- CPU 0 alias PRO_CPU (Protocol CPU) ;
- CPU 1 alias APP_CPU (Application CPU).

Espressif indique que « les deux coeurs sont identiques en pratique et partagent la même mémoire ». Pour prendre en charge le traitement symétrique multiprocesseur (SMP), ils précisent que « l'ordonnanceur sautera des tâches lors de la mise en œuvre d'un ordonnancement Round-Robin entre plusieurs tâches à l'état Ready de même priorité ». Ceci provient de la limitation à utiliser une *ready list* conçue pour une UC unique sur une plateforme qui en a deux [2].

Le problème rencontré était que lorsque l'UC avait besoin d'un changement de contexte de tâche (pour exécuter la tâche suivante), l'UC n'a qu'une *ready list* de tâches à scruter. Alors si l'index courant de la liste pointe sur des tâches *ready* pour l'autre UC, ces entrées doivent être sautées jusqu'à ce qu'une entrée pour l'UC concernée soit trouvée. L'ordonnancement Round-Robin peut donc être loin de la perfection. Le résultat pour le développeur est que l'ordonnancement Round-Robin n'est pas complètement équitable dans les ESP32 à double-UC. Pour de nombreux projets, cela ne sera pas perceptible, mais si cela devient problématique, il y a des moyens de le contourner par codage. Pensez-y dans votre planification de tâches.

Démonstration

Un programme de démonstration Arduino existe pour la carte Lolin ESP32 OLED Display Module (**fig. 1**). En modifiant quelques macros du programme, vous pourrez y changer les priorités de quatre tâches différentes. Le programme est conçu pour afficher trois chenilles qui rampent d'avant en arrière sur toute la longueur de l'OLED. Chaque chenille n'ondule que si la tâche qui la pilote obtient du temps d'UC. Les tâches privées d'UC laisseront la chenille immobile ou lente.

Chaque chenille est pilotée par une tâche qui consomme du temps d'UC puis envoie un message à la quatrième tâche chargée de faire ramper cette chenille et de l'afficher.

Le code pour dessiner et gérer l'état de la chenille est défini

dans la classe `InchWorm` (pas présentée ici). Pour cet article, nous allons juste nous concentrer sur l'action sur chaque chenille de la méthode `InchWorm::draw()`. Chaque instance de la classe `InchWorm` gère son propre état et sa progression. L'écran et les instances de chenilles sont déclarés dans le programme comme suit :

```
static Display oled;
static InchWorm worm1(oled,1);
static InchWorm worm2(oled,2);
static InchWorm worm3(oled,3);
```

Chaque instance comporte une référence C++ (comme un pointeur en C) vers la classe de l'écran en premier argument et le numéro de chenille en second. La référence à l'écran permet de futures évolutions comme la prise en charge de plusieurs écrans. Le numéro de chenille détermine sa position sur l'OLED (1, 2 et 3 correspondent à la ligne du haut, du milieu et du bas). La tâche derrière chaque chenille est juste une boucle de consommation de temps d'UC et d'appel à l'envoi d'un message :

```
void worm_task(void *arg) {
    InchWorm *worm = (InchWorm*)arg;

    for (;;) {
        for ( int x=0; x<800000; ++x )
            __asm__ __volatile__(“nop”);
        xQueueSendToBack(qh,&worm,0);
        // vTaskDelay(10);
    }
}
```

Il est important de laisser la fonction `vTaskDelay()` commentée pour l'instant. Elle sera utilisée dans une prochaine expérimentation.

La même fonction de tâche sert pour les trois tâches, l'argument nommé `arg` précisant quelle instance de chenille on souhaite faire tortiller. L'adresse de la chenille provient d'un pointeur non typé et est stockée dans la variable locale `worm`. Elle n'est utilisée dans cette tâche que pour l'envoi d'un message à la tâche d'affichage (tâche principale) qui lui indique la chenille à faire tortiller.

Notez que lorsque `xQueueSendToBack()` est appelée dans cette démonstration, le paramètre temps d'attente a été fixé à zéro (3^e argument). Ceci indique à FreeRTOS de l'insérer dans la file si possible, mais échoue immédiatement si la file est pleine. C'est volontaire parce que si la file devient pleine, il ne faut pas que cela bloque l'exécution de notre tâche de chenille. La tâche ne doit pas relâcher l'UC pour cette démonstration afin qu'elle puisse vraiment monopoliser l'UC.

La boucle `for` extérieure fait que la tâche effectue ses opérations indéfiniment. La boucle `for` intérieure de consommation de temps d'UC exécute 800 000 fois une instruction *no operation* (*nop*). Le mot-clé `__volatile__` empêche que le compilateur n'optimise cette boucle en la supprimant. Quoiqu'en pense le compilateur, ce gaspillage est utile, nous y tenons. À la fin de la boucle de consommation de temps, nous envoyons l'adresse de la chenille à faire se tortiller à la file identifiée par le handle `qh`. Dès que le message est reçu par la tâche d'affichage, cela fait avancer notre chenille et affiche son déplacement.

La tâche principale `loop()` d'Arduino est utilisée comme tâche d'affichage pour faire se tortiller les chenilles :

```
void loop() {
    InchWorm *worm = nullptr;

    if ( xQueueReceive(qh,&worm,portMAX_DELAY) )
        worm->draw();
}
```

Cette boucle bloque l'exécution jusqu'à ce que l'une des tâches envoie l'adresse de la chenille à dessiner. Dès que ce pointeur de classe est reçu, on invoque la méthode `InchWorm::draw()` pour dessiner la chenille et la faire avancer.

La fonction `setup()` est illustrée dans le **listage 1**, qui montre comment les trois tâches chenilles et la file sont créées.

Listage 1 – La fonction `setup()`.

```
void setup() {
    TaskHandle_t h = xTaskGetCurrentTaskHandle();

    app_cpu = xPortGetCoreID(); // Quelle UC ?
    oled.init();
    vTaskPrioritySet(h,MAIN_TASK_PRIORITY);
    qh = xQueueCreate(4,sizeof(InchWorm*));

    // Dessiner au moins une chenille :
    worm1.draw();
    worm2.draw();
    worm3.draw();

    xTaskCreatePinnedToCore(
        worm_task, // Fonction
        «worm1», // Nom de la tâche
        3000, // Taille de la pile
        &worm1, // Argument
        WORM1_TASK_PRIORITY,
        nullptr, // Pas de retour d'identifiant
        app_cpu);

    xTaskCreatePinnedToCore(
        worm_task, // Fonction
        «worm2», // Nom de la tâche
        3000, // Taille de la pile
        &worm2, // Argument
        WORM2_TASK_PRIORITY,
        nullptr, // Pas de retour d'identifiant
        app_cpu);

    xTaskCreatePinnedToCore(
        worm_task, // Fonction
        «worm3», // Nom de la tâche
        3000, // Taille de la pile
        &worm3, // Argument
        WORM3_TASK_PRIORITY,
        nullptr, // Pas de retour d'identifiant
        app_cpu);
}
```

Changement de priorité

FreeRTOS autorise une tâche à changer sa propre priorité ou celle d'une autre tâche avec la fonction `vTaskPrioritySet()`. Par défaut la tâche qui invoque `setup()` et `loop()` s'exécute au niveau de priorité 1 (ces fonctions sont appelées par la même tâche principale). Pour cette démonstration, il faut que cette priorité soit plus haute que les trois autres tâches chenilles. La fonction `setup()` modifie la priorité de sa propre tâche :

```
static int app_cpu = 0; // Updated by setup()
...
void setup() {
    TaskHandle_t h = xTaskGetCurrentTaskHandle();

    app_cpu = xPortGetCoreID(); // Which CPU?
    ...
    vTaskPrioritySet(h,MAIN_TASK_PRIORITY);
```

Comme indiqué, la fonction `setup()` obtient son propre identifiant de tâche en appelant `xTaskGetCurrentTaskHandle()` et en le stockant dans `h`. En changeant la priorité de la tâche principale par l'appel à `vTaskPrioritySet()`, la priorité de la tâche utilisée par `loop()` est aussi affectée. C'est un exemple de la façon dont on peut régler les priorités des tâches.

Dans la première expérimentation, on affecte aux tâches chenilles les priorités 9, 8 et 7. Ceci implique que notre tâche (principale) d'affichage soit à une priorité 9 ou supérieure (nous prendrons 10). Si nous ne le faisons pas, la tâche principale `loop()` serait privée d'UC et incapable d'animer les chenilles.

Quelle UC ?

Dans l'extrait de `setup()` présenté, on voit une autre fonction API de l'ESP32 nommée `xPortGetCoreID()` qui cherche sur quelle UC l'application est exécutée. Le résultat est assigné à la variable statique `app_cpu` dans le programme de façon que le code sache pour quelle UC créer de nouvelles tâches. Pour les ESP32 à double cœur, la valeur de `app_cpu` sera 1 (exécution sur l'UC 1 dans une configuration à double cœur). Pour les plateformes à un seul cœur, `app_cpu` sera mis à zéro. Le coder ainsi permet normalement la portabilité entre plateformes simples ou doubles.

Toutefois, cette démonstration en particulier ne fonctionnera pas bien sur une plateforme à simple-UC à cause de la façon dont l'UC est monopolisée. Cela déclenchera le temporisateur du chien de garde (*watchdog*) et provoquera une RàZ. Mais la technique d'utilisation de `xPortGetCoreID()` illustre comment permettre la portabilité pour d'autres applications.

Configuration de la démo

Le code source de la démonstration est disponible [3]. Au début se trouvent les définitions des macros qui configurent chaque expérimentation :

```
// Worm task priorities
#define WORM1_TASK_PRIORITY 9
#define WORM2_TASK_PRIORITY 8
#define WORM3_TASK_PRIORITY 7

// loop() must have highest priority
#define MAIN_TASK_PRIORITY 10
```

Laissez-les d'abord telles quelles pour la première expérimentation.

Écran OLED personnalisé

Si vous n'utilisez pas la carte recommandée Lolin ESP32 avec son OLED intégré, les réglages de votre écran personnalisé peuvent être reconfigurés ici :

Display()

```
int width=128,  
int height=64,  
int addr=0x3C,  
int sda=5,  
int scl=4);
```

Si vos réglages sont corrects, l'OLED devrait devenir blanc immédiatement à l'initialisation du programme. Sinon, revoyez les connexions et réglages.

Démonstration 1

Avec le code téléchargé, il suffit de compiler, de graver et d'exécuter l'application. Votre OLED devrait devenir blanc immédiatement, avec trois Chenilles noires dessinées (**fig. 2**).

La configuration (encore) pour cette expérimentation est :

```
#define WORM1_TASK_PRIORITY 9  
#define WORM2_TASK_PRIORITY 8  
#define WORM3_TASK_PRIORITY 7  
#define MAIN_TASK_PRIORITY 10
```

Avec cette configuration, la Chenille du haut va traverser en se tortillant, pas les deux du bas. Pourquoi les Chenilles du milieu et du bas ne bougent-elles pas ?

```
--  
--  
--
```

Nous avons laissé la tâche principale à la priorité 10. Elle profite donc de la priorité la plus haute du jeu de tâches de notre application. La première Chenille, qui s'affiche sur la ligne du haut, pouvait se déplacer parce que c'était la seule tâche consommatrice d'UC capable de s'exécuter. Cette tâche de priorité 9 est capable de s'exécuter parce que la tâche d'affichage de priorité 10 fait des E/S vers l'OLED puis attend l'arrivée de messages dans la file des messages (devient bloquée). Lorsque la tâche d'affichage est bloquée, les autres tâches de priorité inférieures peuvent être séquencées.

Les tâches de priorité 8 et 7 (pour les Chenilles du milieu et du bas) sont privées d'UC et ne sont jamais exécutées parce que la tâche de priorité 9 monopolise complètement l'UC. C'est le propre de l'ordonnancement en temps réel dans FreeRTOS. À la différence de Linux ou Windows, les tâches de priorité inférieures n'ont aucune chance de s'exécuter.

Démonstration 2

Modifions la configuration pour donner aux trois Chenilles la même priorité, mais laissons la tâche principale d'affichage à la priorité 10. Mettons-les toutes trois à la même priorité 9 (ou 8 ou 7) :

```
#define WORM1_TASK_PRIORITY 9  
#define WORM2_TASK_PRIORITY 9  
#define WORM3_TASK_PRIORITY 9  
#define MAIN_TASK_PRIORITY 10
```

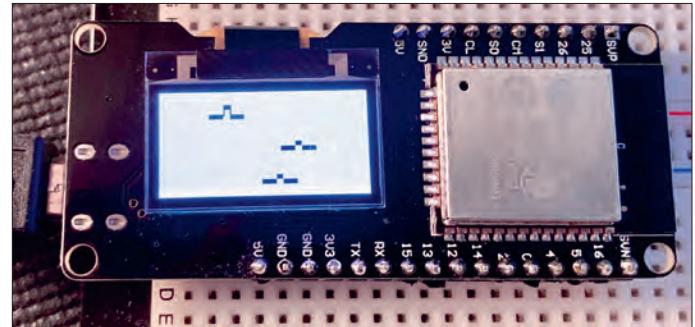


Figure 2 : Lorsque les tâches sont exécutées, les Chenilles de la démo se déplacent.

Lorsque vous recompilez et regravez l'ESP32, que constatez-vous ?

```
--  
--  
--
```

Elles vont marcher au même rythme (ou presque). Quand on laisse tourner la démonstration assez longtemps, une Chenille peut prendre un peu d'avance sur les autres.

Démonstration 3

Dans cette expérimentation, modifiez la configuration pour donner aux trois Chenilles la même priorité (comme dans la précédente) et donnez à la tâche principale d'affichage cette même priorité. Je vais utiliser la priorité 9 pour toutes ces tâches :

```
#define WORM1_TASK_PRIORITY 9  
#define WORM2_TASK_PRIORITY 9  
#define WORM3_TASK_PRIORITY 9  
#define MAIN_TASK_PRIORITY 9
```

Après recompilation, regravure et exécution du code, que constatez-vous ? Y a-t-il une différence ? Pourquoi avancent-elles à des vitesses différentes ?

```
--  
--  
--
```

Lorsque j'exécute cela, la Chenille du bas semble obtenir le plus d'UC (c.-à-d. qu'elle se tortille le plus vite). La Chenille du haut est la plus lente. C'est encore la faute à l'irrégularité du Round-Robin signalé par Espressif. Idéalement, la tâche d'affichage ne devrait prendre qu'un peu d'UC pendant le dessin de la Chenille. Par ailleurs, le temps d'UC restant devrait être réparti également entre les trois autres tâches qui pilotent les Chenilles. Pourtant l'ordonnancement est déséquilibré. Les deux UC répondent à des temporiseurs et autres interruptions. Ce sont les imperfections du code de l'ordonnanceur qui perturbent l'ordonnancement Round-Robin.

Démonstration 4

Jusqu'ici, dans chaque démonstration, chaque tâche Chenille a consommé tout le temps d'UC qu'elle a pu en obtenir. Que se passe-t-il si nous introduisons un petit retard (blocage) dans la boucle ?

Réinitialisez la configuration pour que la tâche principale d'affichage ait une priorité 10, et chacune des tâches chenilles respectivement une priorité 9, 8 et 7 :

```
#define WORM1_TASK_PRIORITY 9  
#define WORM2_TASK_PRIORITY 8  
#define WORM3_TASK_PRIORITY 7  
#define MAIN_TASK_PRIORITY 10
```

Puis ôtez le commentaire de la ligne où `vTaskDelay()` est appelée de sorte que la boucle de la tâche devienne :

```
void worm_task(void *arg) {  
    InchWorm *worm = (InchWorm*)arg;  
  
    for (;;) {  
        ...  
        for ( int x=0; x<800000; ++x )  
            __asm__ __volatile__(“nop”);  
        xQueueSendToBack(qh,&worm,0);  
        vTaskDelay(10); // Uncommented  
    }  
}
```

Maintenant, chaque tâche chenille va consommer de l'UC, essayer d'insérer une chenille dans la file puis se bloquer pendant 10 ms. Compilez, gravez et exécutez cet exemple. Que constatez-vous ?

La chenille du haut est la plus rapide et celle du bas la plus lente. Celle du haut avec la priorité 9 est la première à obtenir l'UC grâce à sa haute priorité (pendant que la tâche d'affichage est bloquée). Lorsque la tâche chenille est bloquée dans l'appel à `vTaskDelay(10)`, la tâche de priorité immédiatement inférieure (la chenille du milieu) obtient un peu d'UC à consommer et finit par appeler `vTaskDelay(10)`. Ceci permet à la tâche de priorité 7, encore plus basse, d'obtenir quelques cycles. Cet effet de cascade distribue le temps d'UC des niveaux les plus élevés aux plus bas.

Notez que les tâches de priorité 8 et 7 sont préemptées dès que la tâche de plus haute priorité 9 redevient *ready*. C'est pourquoi la chenille du haut est la plus rapide. Quelquefois celle du milieu préempte la tâche de priorité 7 : elle tend à être plus rapide que la chenille du bas.

Plus d'expérimentations

Que se passe-t-il si vous allongez le délai de `vTaskDelay()` à beaucoup plus que 10 ms ? Essayez d'intuiter la réponse puis exécutez-le. Pourquoi obtenez-vous ce résultat ? Que se passe-t-il si vous réduisez le délai à 1 ms ? À vous d'explorer...

Configuration de priorité

Nous n'avons pas encore parlé des interruptions dans l'ESP32, mais il est bon d'examiner le fichier d'entête `FreeRTOSConfig.h`, qui configure les priorités pour la plateforme et se trouve ici :

```
$IDF_PATH/components/freertos/include/freertos/  
FreeRTOSConfig.h
```

L'entête définit les valeurs de macro de priorité suivantes. Les valeurs compilées sont indiquées :

```
configMAX_PRIORITIES = 25  
configKERNEL_INTERRUPT_PRIORITY = 1  
configMAX_SYSCALL_INTERRUPT_PRIORITY = 3
```

La première macro définit le nombre maximum de priorités disponibles. Cela signifie que les niveaux de priorités valides vont de 0 à 24.

La seconde macro définit la priorité utilisée par le noyau lui-même pour les **interruptions**, tandis que la troisième macro fixe la priorité la plus haute utilisée par les interruptions du noyau. Tout appel d'une API FreeRTOS lancé depuis une *Interrupt Service Routine* (ISR) ne concernera que des fonctions API FreeRTOS dont le nom se termine par `FromISR()`. De plus, avec les valeurs indiquées, ces fonctions ne peuvent être appelées que par des tâches d'interruption de priorités 1 à 3. Si aucun appel `FromISR()` n'est fait, l'ISR peut librement opérer à des niveaux de priorité de 4 à 24.

Résumé

Que conclure de ces expérimentations ? Un concept de priorité, simple a priori, ne l'est pas. De sorte que si vos priorités de tâches ne sont pas bien planifiées, vous aurez des surprises – certaines tâches peuvent être privées d'UC. Nous n'avons pas encore parlé de l'impact des temporiseurs du chien de garde. Si p. ex. le temporisateur du chien de garde se déclenche dans l'UC 0, votre ESP32 redémarrer !

Pour les ESP32 à double cœur, il y a aussi le problème de l'ordonnancement *Round-Robin* des niveaux de priorité identiques qui se traduit par des inégalités de durée d'exécution, ce qui peut être problématique dans certaines applications, pas dans d'autres. Tout dépend de votre « système ».

Pour beaucoup d'applications, vous pouvez simplement créer des tâches qui fonctionnent en priorité 1. C'est la priorité configurée pour les tâches `setup()` et `loop()` d'Arduino. Des tâches de priorité supérieure peuvent être utilisées sans risque si elles se bloquent sur une file, un séaphore ou un autre événement. Lorsqu'une tâche se bloque ou est suspendue, l'UC est partagée entre les autres tâches de priorité égale ou inférieure. Une application avec des priorités de tâches correctement configurées fonctionnera comme une machine bien huilée. ─

(191195-02 VF Denis Lafourcade)



Liens

- [1] Multitâche en pratique avec l'ESP32, Elektor janvier 2020 p. 42 : www.elektormagazine.fr/190182-03
- [2] Traitement symétrique multiprocesseur : <https://thc420.xyz/esp-idf/file/docs/en/api-guides/freertos-smp.rst.html>
- [3] Code source du projet : https://github.com/ve3wwg/esp32_freertos/blob/master/priority-worms1/priority-worms1.ino

Robert Lacoste : grands et petits secrets 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 R. Lacoste, l'auteur de

Percer les mystères de l'électronique,

est de vous donner 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 utilisés. Repassez par les notions de base et distinguez les véritables progrès techniques des laïus commerciaux.

Ce livre vous aide à y voir plus clair. Sans formule mathématique qui ne soit pas à la portée d'un lycéen, il balaye tout le champ de l'électronique. L'auteur explique comment ça marche, pourquoi parfois ça ne marche pas, et différentes techniques pour que vos projets marchent toujours.

Auteur de plus d'une centaine d'articles dans les revues techniques spécialisées Elektor et Circuit Cellar, Robert Lacoste a plus de trente ans d'expérience dans le domaine des signaux mixtes : acquisition et traitement du signal, radiofréquences et antennes, hyperfréquences, électronique ultra-rapide..

Sommaire

- adaptation d'impédance : qu'est-ce-que c'est ?
- petite introduction aux microrubans
- jouons avec la rélectrométrie temporelle
- circuits imprimés : éviter les bourdes en HF
- le marquage CE pour les bœtiens
- le quartz
- magie de la PLL
- synthèse numérique directe
- comprendre l'amplificateur à transistor
- ampli de classe A, B, C, D, E, F, G, H : quesako ?
- le filtrage numérique sans stress (filtres FIR)
- le filtrage numérique sans stress (filtres CIC)



- le filtrage numérique sans stress (filtres IIR)
- l'ABC des CA/N (I)
- l'ABC des CA/N (II)
- bruit et sensibilité des récepteurs
- échange débit contre portée
- LoRaWAN
- corrélation numérique
- des condensateurs qui se rechargent tout seuls !
- composants parasites
- X7R, Y5V, NP0, quesaco ?
- comment consommer moins
- le BA-B.A des convertisseurs DC/DC
- élévation de tension



Prix : 33,75 € (membres)
www.elektor.fr/19080

nouveau livre
en français
248 pages en couleur



Sigfox : un renard sur l'internet des objets (3)

Vos premiers pas de goupil sur l'internet des objets

Frank Schleking & Bernd vom Berg (Allemagne)

Notre carte MKR FOX 1200 a été enregistrée sur le réseau Sigfox.

Rien ne s'oppose à une première tentative de communication.

Notre appareil Sigfox est censé envoyer sur le nuage le célèbre message d'accueil „Hello World !“.

Comment notre station se présente-t-elle dans le nuage Sigfox ? Connectez-vous à l'infrastructure dorsale (*backend*) de Sigfox [1] comme d'habitude, avec votre adresse e-mail et votre mot de passe. Vous serez sur la page d'accueil du portail Sigfox (**fig. 1**). Cliquez maintenant sur l'onglet *Device*. Vous verrez un aperçu des appareils Sigfox enregistrés ou activés par vous (**fig. 2**). Il ne devrait y avoir que la station MKR FOX 1200) :

- **Group** : Le nom du groupe a été créé automatiquement par Sigfox et ne peut pas être modifié par vous. Ce groupe contient maintenant tous vos appareils Sigfox. Cliquez sur le nom du groupe pour obtenir des informations (non modifiables).

- **Device type** = type d'appareil : dans le groupe, vous pouvez trier et classer les stations identiques (type, structure, fonctions) selon le type d'appareil, p. ex. les stations

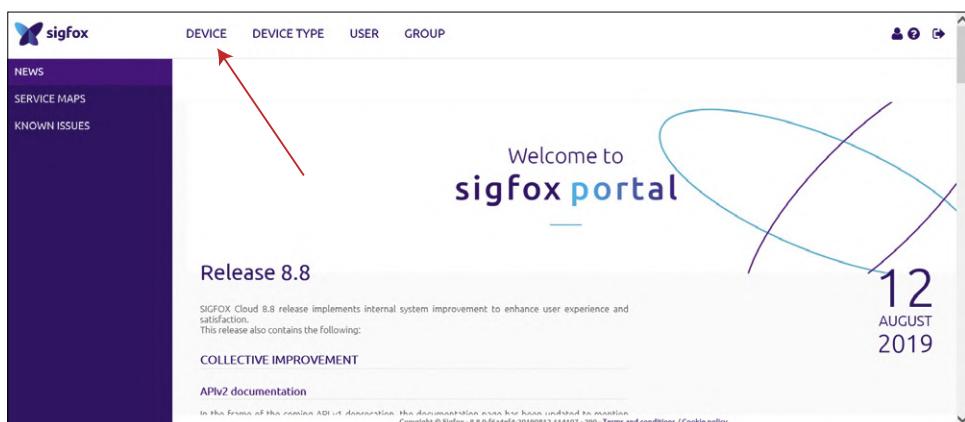


Figure 1 : Le portail Sigfox.

de mesure de température, les stations de lecture des valeurs analogiques des capteurs, les stations d'enregistrement de signaux binaires. Dans notre cas, nous n'avons bien sûr qu'un seul type de station, un seul *type de Device*, à savoir *Arduino_DevKit_1*. Plus tard, vous pourrez changer ce nom selon vos propres besoins.

- **Id** (= numéro d'identification) : le numéro d'identification unique de la station.
- **Last seen** = dernière vue : l'infrastructure saisit ici la date et l'heure de la dernière communication de la station, c'est-à-dire la date à laquelle le dernier télégramme a été reçu par la station. L'indication N/A signifie que l'infrastructure n'a encore reçu aucun télégramme de la station.
- **Nom** : Le nom de la station.
- **Token state** = statut du jeton : un jeton est un droit de transmission pour une station sur le réseau Sigfox. Quand on signe un contrat avec Sigfox (et payé les frais correspondants pour l'utilisation du réseau), on reçoit un certain nombre de jetons. Pour qu'une station du réseau Sigfox puisse devenir active, un jeton lui est attribué. Elle peut dès lors envoyer et recevoir des données sur le réseau. Quand vous achetez une carte MKR FOX 1200 et que vous l'enregistrez dans l'infrastructure Sigfox, vous recevez pour un an un jeton gratuit. Votre carte MKR FOX 1200 peut fonctionner comme station sur le réseau pendant un an. Sigfox attribuera son propre jeton unique à chaque nouvelle carte MKR FOX 1200. L'*état du jeton* indique si la station a déjà utilisé son jeton, c'est-à-dire si la station a déjà émis envoyé des données sur le réseau Sigfox. Comme ce n'est pas encore le cas pour nous, nous trouvons ici un point d'interrogation ; la station n'a pas encore revendiqué ses droits de diffusion. Dès que la station aura envoyé son premier télégramme et donc utilisé son droit de transmission, une coche apparaît ici. Ensuite seulement le jeton est finalement attribué à cette station.

Cliquez maintenant sur le *nom*. Un menu déroulant (**fig. 3**) vous propose l'option *Edit*. Dans la fenêtre d'édition suivante (**fig. 4**), certains paramètres de l'appareil sont définies ou modifiées. Changez ici le nom de la station Sigfox (si nécessaire), par exemple „MKR FOX - 1“. Les autres champs restent inchangés. Cliquez sur *Ok*.

La fenêtre *Device Information* (**fig. 5**) apparaît maintenant, qui résume toutes les informations sur le module Sigfox. Regardez, ne touchez ni changez rien. Si vous cliquez sur l'onglet *Device* en haut à gauche, vous revenez à l'aperçu des appareils, où vous verrez le changement de nom effectué. Fermez l'infrastructure de Sigfox.

Bonjour tout le monde !

Pour que l'activation soit effective et pour que la station puisse travailler sur le réseau Sigfox, il reste à envoyer un message valide qui marquera le début de votre abonnement Sigfox gratuit

The screenshot shows the 'Device - List' page of the Sigfox web interface. On the left, there's a sidebar with 'DEVICES' and 'DELETED DEVICES'. The main area has a search bar with fields for 'Id', 'Last seen from date', and 'Last seen to date'. Below the search is a table header with columns: Communication status, Device type, Group, Id, Last seen, Name, and Token state. A single row is shown: 'Arduino_DevKit_1' (PalmTec, 1D334D, N/A, 2019-08-28 12:12:46, MKR FOX - 1). At the bottom, there are buttons for 'New', 'New series', 'Edit series', 'Transfer series', 'Replace series', and 'Delete series'. The footer includes copyright information and links to terms and conditions.

Figure 2 : Les appareils Sigfox enregistrés.

This screenshot shows a dropdown menu for the 'Name' field of a device named 'MKR FOX - 1'. The menu options are 'Disengage sequence number', 'Edit' (which is highlighted in purple), and 'Transfer'.

Figure 3 : Le menu déroulant pour *le champ du nom*.

This screenshot shows the 'Device 1D334D - Edition' window. It includes sections for 'Device information' (Identifier: 1D334D, Name: MKR FOX - 1, PAC: 63/F8290422045A0, End product certificate: P_00C1_1366_01), location coordinates (Lat: 0.0, Lng: 0.0), and a map button. There are also buttons for 'Map' and 'Locate on map', and checkboxes for 'Subscription automatic renewal' and 'Locate on map'. At the bottom are 'Ok' and 'Cancel' buttons.

Figure 4 : La fenêtre de l'éditeur de l'appareil.

d'un an à raison de 140 messages émis et quatre messages reçus par jour.

Le premier message à envoyer à l'infrastructure Sigfox est le texte réjouissant „Hello World!“, soit exactement douze caractères ASCII (= octets) qui tient donc parfaitement dans une charge utile de Sigfox. Les fonctions Sigfox de radiodiffusion ont été décrites dans la première partie de cet article, de sorte que la routine de radiodiffusion du **listage 1** et ses fonctions sont faciles à comprendre :

The screenshot shows the 'Device 1D334D - Information' page. It includes sections for 'INFORMATION', 'LOCATION', 'MESSAGES', 'EVENTS', 'STATISTICS', and 'EVENT CONFIGURATION'. Key details include:

- Name:** MKR FOX - 2
- Protocol:** v1
- Activable state:**
- Sequence number:** 30 (2019-08-28 12:12:46)
- Trash sequence number:** N/A (N/A)
- Last seen:** 2019-08-28 12:12:46
- PAC:** 637F8290422D45A0
- Product certificate:** P_00C1_1366_01
- Latitude:** 0.000 (degrees)
- Longitude:** 0.000 (degrees)
- Device type:** Arduino_DevKit_1
- State:** UK
- Link Quality Indicator:**
- Communication status:**
- Contract:** palmett_66c7_b09e
- Activation date:** 2019-08-27 08:40:22
- Token validity:** 2020-08-27
- Unsubscription date:** N/A
- Subscription automatic renewal status:** Not allowed
- Subscription automatic renewal:**
- Creation date:** 2019-08-18 17:25:06
- Created by:** buy.sigfox.com
- Last edition date:** 2019-08-28 12:40:22
- Last edited by:** Bernd vom Berg

At the bottom, there are tabs for 'MODEM CERTIFICATE' and 'PRODUCT CERTIFICATE'.

Figure 5 : La fenêtre d'information de l'appareil.

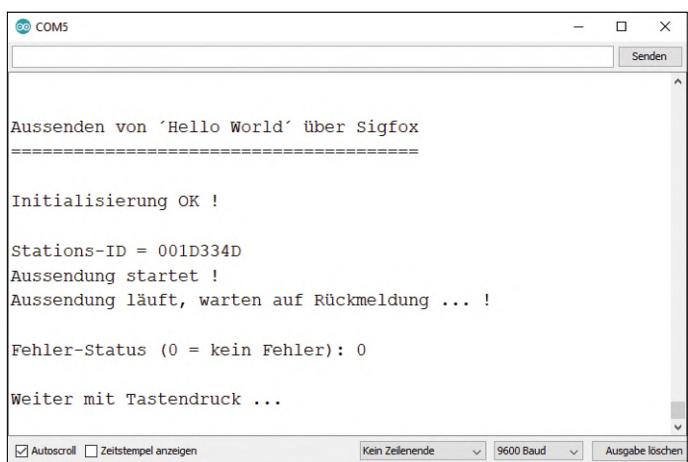


Figure 6 : La transmission de la chaîne de test.

Listage 1. Émission du message „Hello World !“.

```
// enable Sigfox modem and check for errors
if (!SigFox.begin()) // error occurred?
{
    Serial.println("Error in Sigfox module!");

    while (1); // after error, drop into infinite loop
}

// enable Sigfox modem debug mode
SigFox.debug();

// now we will send the string "Hello World!"

// prepare to transmit a packet
SigFox.beginPacket();

// assemble string (sequence of characters) into a Sigfox message
SigFox.print("Hello World!");

// final step in packet assembly and transmission; check for errors
// return code in variable ret
int ret = SigFox.endPacket();

Serial.print("\nError status (0 = no error): ");
Serial.println(ret);

// close Sigfox library and shut module down
SigFox.end();
```

- **SigFox.begin** : initialise la bibliothèque Sigfox et le modem Sigfox et signale une erreur le cas échéant.
- **SigFox.debug** : active le mode de débogage. Les fonctions d'économie d'énergie du microcontrôleur et du modem sont désactivées, la LED jaune de la carte MKR FOX 1200 clignote pour signaler l'envoi et la réception de données Sigfox.
- **SigFox.beginPacket** : lance la transmission d'un paquet (= charge utile) Sigfox.
- **SigFox.print** : place dans la charge utile la valeur ou la chaîne de caractères passée puis l'envoie. Les valeurs numériques sont envoyées sous forme de chaînes de caractères ASCII. Pour les données binaires (nombres „réels“, pas de représentation ASCII), on utilise la fonction **SigFox.write** (nous y reviendrons).
- **int ret = SigFox.endPacket** : termine la transmission du paquet Sigfox (charge utile Sigfox) et retourne 0 (= aucune erreur) ou 1 (= erreur).
- **SigFox.end** : ferme la bibliothèque Sigfox et le modem Sigfox

Où l'on voit l'étonnante simplification de la communication de données complexes grâce au recours à une bibliothèque puissante ! Chargez le sketch Sigfox sur la carte MKR FOX 1200, ouvrez le terminal et lancez le sketch. La fenêtre du moniteur affiche le menu principal du sketch. Sélectionnez maintenant le point de menu 2 *Transmettre „Hello World !*. La procédure d'envoi de la chaîne de test est exécutée, certaines informations de statut étant éditées

dans la fenêtre du terminal. Le processus de transmission est signalé par le clignotement de la LED jaune sur la carte MKR FOX 1200. La transmission complète devrait se dérouler sans erreur (**fig. 6**).

De retour dans l'arrière-boutique

Si vous avez une couverture radio Sigfox, l'infrastructure Sigfox reçoit vos données dans un délai très court. Vérifiez en démarrant l'infrastructure (*backend*), en vous connectant et en cliquant sur l'onglet *Device* sur l'écran de démarrage. Dans la liste des appareils Sigfox actifs, cliquez (exactement) sur le champ *Id* de la station pour faire apparaître la page d'information de l'appareil et là, à gauche, sur *Messages*.

La fenêtre suivante (**fig 7**) énumère tous les télégrammes reçus par l'infrastructure Sigfox. Le champ *Time* est explicite ; le champ *LQI* (Link Quality Indicator) indique l'intensité du champ avec lequel le télégramme Sigfox a été reçu par les stations de base. Si vous placez le curseur sur ce champ, vous verrez une description de la qualité de réception et l'opérateur local du réseau Sigfox dans lequel se trouve la station Sigfox sera affiché. Cet opérateur ne s'appelle pas nécessairement Sigfox, il peut aussi être un partenaire (p. ex. à l'étranger). Lorsque vous cliquez sur *Location*, une carte montre où approximativement se trouve la station Sigfox.

Venons-en maintenant aux informations importantes dans le champ *Data/Decoding* (Données/Décodage), où le contenu de la charge utile est affiché, une fois sous forme de

données brutes et une fois sous forme décodée. L'infrastructure ignore encore comment interpréter les données brutes, il faut le lui apprendre. Examinons le champ de donnée :

```
48 65 6c 6c 6f 20 57 6f 72 6c 64 21
H e l l o   W o r l d !
```

La chaîne est composée des codes ASCII des caractères du message de test „Hello World ! (20 est le code ASCII pour l'espace). Le mode de décodage peut être réglé à partir de la fenêtre *Device List* (fig. 2). Cliquez sur le nom *Arduino_DevKit_1* sous *Device type*, vous accédez alors au champ d'information de ce type de périphérique (**fig 8**). Après avoir cliqué sur *Edit*, vous accédez à la fenêtre d'édition correspondante (**fig. 9**). Dans la zone *Payload display* (= affichage des charges

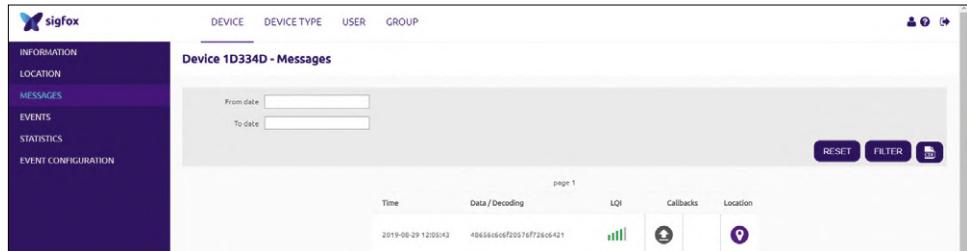


Figure 7 : La fenêtre de message de l'appareil.

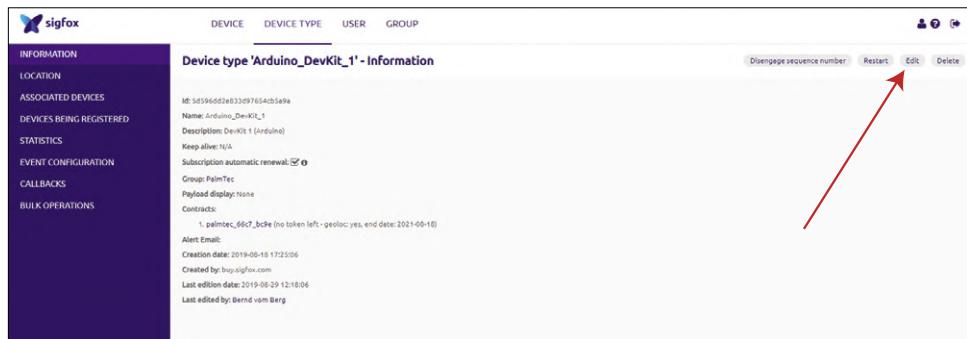


Figure 8 : Le champ d'information pour le type d'appareil.

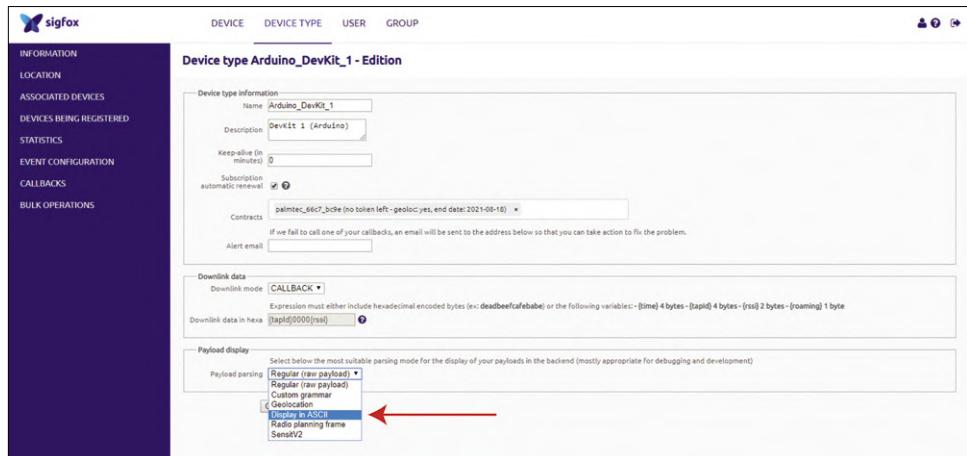


Figure 9 : La fenêtre d'édition pour le type d'appareil.

utiles) modifiez (seulement) *Payload parsing* (= analyse des charges utiles) en choisissant *Display in ASCII* (= afficher en ASCII) et confirmez cette modification. De retour dans la fenêtre *Messages*, vous verrez pour cette station une ligne supplémentaire avec le contenu de la charge utile affiché en caractères ASCII (**fig. 10**).

Deux étapes fondamentales sont nécessaires pour transférer des valeurs numériques dans la charge utile. Vous devez d'abord déterminer le type des données à transférer dans la charge utile de 12 octets, car à chaque type de données correspond un nombre spécifique d'octets ou. Les plus courants sont réunis dans le **tableau 1**. Dans la charge utile, vous pouvez organiser les types de données à transférer comme dans le **tableau 2**. Sur les douze octets disponibles dans la charge utile, dix octets seraient utilisés.

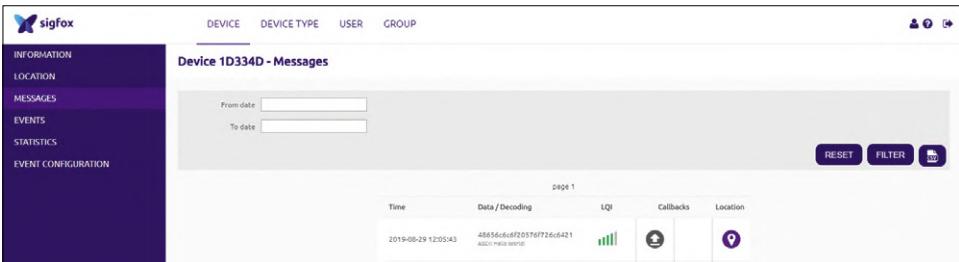


Figure 10 : Décodage du contenu de la charge utile en caractères ASCII.

Tableau 1. Nombre d'octets de différents types de données.

données	octets
char / unsigned char	1
int8_t / uint8_t	1
int16_t / uint16_t	2
int / unsigned int	4
int64_t / uint64_t	8
float	4

Listage 2. Structure du type de données Sigfox requis Sigi_Dat_1.

```
// structure and data type 1 for our example:
// transmission of fixed numeric values
typedef struct __attribute__((packed)) sigfox_message {
    unsigned char value_1; // first value in payload: 1 byte long
    unsigned int value_2; // second value in payload: 4 bytes long
    float value_3; // third value in payload: 4 bytes long
    unsigned char value_4; // fourth value in payload: 1 byte long
} Sigi_Dat_1;
```

Tableau 2. Organisation possible des types de données dans la charge utile.

Valeur dans la charge	données	octets
1	unsigned char	1
2	unsigned int	4
3	float	4
4	unsigned char	1

Listage 3. Émission de valeurs numériques fixes.

```
// definition of four fixed numeric values
unsigned char AIN_1 = 0x1f; // first value, 1 byte long
unsigned int pressure = 0x12345678; // second value, 4 bytes long
float temp = 1233.56; // third value, 4 bytes long
unsigned char AIN_2 = 0x55; // fourth value, 1 byte long

// enable Sigfox modem and check for errors
if (!SigFox.begin()) // error occurred?
{
    Serial.println("Error initializing Sigfox module. RESET to continue");
    while (1); // after error, drop into infinite loop
}
else
{
    Serial.println("Sigfox modem OK\n");
}

// enable debug LED and disable power-saving modes
SigFox.debug();

// deal with all pending interrupts
SigFox.status();
delay(1);

// now we write the current values that are to be transmitted into
// the SF_send structure variable
// this process assembles the payload contents
SF_send.value_1 = AIN_1; // unsigned char value: 1 byte
SF_send.value_2 = pressure; // unsigned int value: 4 bytes
SF_send.value_3 = temp; // float value: 4 bytes
SF_send.value_4 = AIN_2; // unsigned char value: 1 byte
```

Ensuite, un type de données spécial est défini à partir de ces spécifications, dans lequel les données souhaitées pour la charge utile peuvent être stockées ensemble très facilement. Dans notre exemple, cela ressemble au **Listage 2**. Pour bien comprendre cette construction en détail, une connaissance approfondie de C/C++ est nécessaire, mais une explication simplifiée est possible.

Avec l'instruction `struct sigfox_message` et les spécifications suivantes entre accolades, nous avons créé une structure appelée `sigfox_message`. Ce n'est rien d'autre qu'un ensemble de données connexes sous un nom (principal) commun. C'est comme pour un tableau ou une matrice, mais ceux-ci sont toujours constitués de données du même type. Une structure rassemble des données de différents types, ici à partir de deux caractères non signés `unsigned char`, un entier non signé `unsigned int` et une valeur flottante `float`. Les types de données des éléments de la structure doivent correspondre exactement à la définition de la charge utile et doivent également se présenter dans le bon ordre.

Suite à la page suivante

Il est facile d'accéder maintenant à un élément spécifique de la structure : **Nom de la structure.Nom de l'élément**. Ainsi, par exemple, avec **sigfox_message.valeur_3 =25.78;** la valeur de l'élément flottant interne **valeur_3** se voit attribuer la valeur **25.78.**

Le pas suivant et nous créons à partir de cette structure un nouveau type de données propre. Bien sûr, vous connaissez les types de données de base en C comme les caractères **non signés, int ou float**. Grâce à l'instruction supplémentaire **typedef** au début de la définition de la structure, nous créons un autre type de données qui correspondra exactement à la structure qui suit. Ce nouveau type de données nécessite naturellement aussi un nouveau nom propre, lequel est spécifié tout à la fin de la définition, ici **Sigi_Dat_1**. Le paramètre **__attribut__ ((packed))** permet de supprimer de la structure les octets de remplissage superflus afin que

Suite à la page précédente :

```
// next we use the Sigfox library to transmit the data we have
// assembled in the structure variable SF_send, which will become the
// payload contents

// prepare to transmit a packet
SigFox.beginPacket();

// pass structure variable to the Sigfox library
SigFox.write((char*)&SF_send, sizeof(SF_send));

// error check: if endPacket() returns 1, report an error
int ret = SigFox.endPacket();
if (ret > 0)
{
    Serial.println("Error: transmission failed. RESET to continue");
    while(1); // infinite loop
}
else
{
    Serial.println("Sigfox transmission OK");
}

// close Sigfox library and shut module down
SigFox.end();
```

suite de la page 23

◊ **Salon des seniors**
du 25 au 28 mars à Paris
Porte de Versailles
www.salondesseniors.com/

◊ **L'INDUSTRIE DE DEMAIN S'INVENTE SUR GLOBAL INDUSTRIE**
31 mars au 3 avril à Paris Nord - Villepinte
<https://www.global-industrie.com/fr>



avril 2020

◊ **Congrès sur la cybercriminalité et la protection des données on-line**
du 01 au 02 avril à Paris
<http://akjassociates.com/event/france>

◊ **Maker Faire Lille**
du 03 au 05 avril 2020 à Lille - Gare Saint Sauveur
<https://lille.makerfaire.com/>

Photo : Quentin Chevrier



◊ **12^{ème} Journée Française des Tests Logiciels**
le 7 avril au Beffroi de Montrouge
<http://www.cftl.fr/JFTL/accueil/>

◊ **RICV - Rencontres internationales de cerfs-volants**
du 6 au 14 avril à Berck-sur-Mer
www.cerf-volant-berck.com



Photo : RICV - mairie de Berck sur Mer

◊ **ProDurable** (économie durable)
Du 27 au 29 avril - Paris Palais des congrès
www.produrable.com

◊ **CONCOURS LEPINE** Salon international des inventions
du 30 avril au 11 mai à Paris
www.concours-lepine.com/

Tableau 3. Charge utile entrante.			
1f	78563412	ec319a44	55
(A)	(B)	(C)	(D)

seuls les octets de données utilisés soient transférés à la charge utile puis émis.

Toute cette structure est maintenant placée tout au début du sketch comme variable globale, de sorte que ce type de données est accessible depuis n'importe quel point du sketch.

Pour utiliser ce type de données, il faudra bien sûr définir des variables pour ce type de données. Cela se passe comme d'habitude :

Type_de_data Nom_de_variable;

La ligne Sigi_Dat_1 SF_send; crée une variable nommée SF_send qui a exactement la structure définie précédemment (il est préférable de la définir aussi comme variable globale). Les valeurs individuelles de ces variables peuvent alors être accessibles avec

SF_send.valeur_1 = 12; par exemple.

Émettre

Pour voir comment les valeurs numériques sont empaquetées dans la charge utile Sigfox, nous écrivons une fonction qui envoie des valeurs numériques fixes sur le réseau Sigfox (**listage 3**). La fonction est appelée par l'option 3 du menu dans le sketch de démonstration. Il n'y a plus qu'à ajouter que c'est la fonction SigFox.write qui est maintenant utilisée pour envoyer la charge utile, puisque nous envoyons maintenant des valeurs numériques au lieu de caractères. Si tout fonctionne, la charge utile apparaît dans l'infrastructure comme l'indique le **tableau 3**.

Pour finir, nous allons écrire une fonction pour la transmission de valeurs de mesure réelles (option 4 du menu dans l'exemple de sketch Sigfox), à savoir la température mesurée par le capteur BMP280 (float, 4 octets), la pression atmosphérique mesurée par le capteur BMP280 (float, 4 octets) et la valeur mesurée par la LDR (pour cela, le cavalier JPS doit être placé en position 1-2 sur la carte mère) à l'entrée analogique A2 (uint16_t, 2 octets). Ainsi, dix des douze octets de la charge utile sont occupés.

Pour cet ensemble de données, nous définissons un autre type de données Sigfox (**listage 4**) et une nouvelle variable appelée SF_send_mw :

```
// Variable pour le type de données 'Sigi_Dat_2' :
Sigi_Dat_2 SF_send_mw;
```

Dans une nouvelle fonction du sketch, les trois valeurs mesurées sont enregistrées, écrites dans la variable de structure et envoyées (**listage 5**). Les sorties de contrôle via le terminal

Listage 4. Nouveau type de données „Sigi_Dat_2“ pour l'émission des valeurs mesurées.

```
// structure and data type 2 for our example:
// transmission of sensor readings
typedef struct __attribute__((packed)) sigfox_message_2 {
    float      value_1;      // first value in payload: 4 bytes long
    float      value_2;      // second value in payload: 4 bytes long
    uint16_t   value_3;      // third value in payload: 2 bytes long

    // this structure therefore occupies a total of ten bytes
} Sigi_Dat_2;
```

Listage 5. Écriture des valeurs de mesure dans la variable de structure.

```
// now we write the current values that are to be transmitted into
// the SF_send_mw structure variable
// this process assembles the payload contents
SF_send_mw.value_1 = temp;           // float value:        4 bytes
SF_send_mw.value_2 = pressure;       // float value:        4 bytes
SF_send_mw.value_3 = LDR;            // uint16_t value:     2 bytes
```

série et sur l'afficheur ePaper affichent les valeurs de mesure courantes ainsi que le processus de communication Sigfox. Des explications sont données dans les commentaires détaillés du programme.

À suivre

Nous voici à la fin de notre courte présentation du réseau 0G-IoT Sigfox, faite à l'aide de la carte Arduino Maker MKR FOX 1200. Avec les programmes et fonctions que nous avons créés, vous disposez d'une base solide pour faire vous lancer sur le réseau Sigfox.

Ce qui manque encore, c'est une visualisation des données du côté de l'utilisateur, c'est-à-dire sur son ordinateur ou son téléphone à l'aide d'un tableau de bord configurable. Ce sera le thème du prochain et dernier épisode de cette série. ─

(190281 - 02 VF)

@ WWW.ELEKTOR.FR

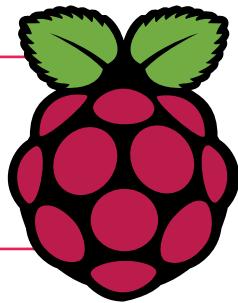


→ Arduino MKR FOX 1200 : www.elektor.fr/19096
 → Arduino Antenne 868 MHz : www.elektor.fr/19095

Lien

- [1] Identification dans l'infrastructure : <https://backend.sigfox.com/auth/login>

Raspberry Pi aide-mémoire des commandes Bash



Clemens Valens (Elektor Labs)

Il est presque impossible de travailler sur Raspberry Pi, ni d'ailleurs sur Linux en général, sans passer par des commandes saisies dans un terminal. Voici donc une liste des commandes du langage Bash fréquemment utilisées.

Le terminal ou la console est cette fenêtre noire dans laquelle vous ne pouvez rien faire d'autre que taper du texte. C'est une interface de ligne de commande ou CLI. Les commandes que l'usager saisit avec la console sont exécutées par l'interpréteur de commandes 'Bash'.

Les commandes sont nombreuses et la plupart d'entre elles acceptent toutes sortes de paramètres et d'arguments. Pour

obtenir de l'aide sur une commande donnée, il suffit dans le terminal d'y ajouter «--help» (attention, il y a deux tirets), par exemple comme ceci : `rm --help`
Dans ce qui suit, [path] (*chemin* en français) se réfère à un chemin relatif ou absolu. Un chemin absolu commence par '/', par exemple : `/home/pi`. ▶

191250-02

<code>pwd</code>	Afficher le nom du répertoire de travail en cours
<code>ls</code>	Lister le contenu du répertoire actuel
<code>ls [path]</code>	Lister le contenu du répertoire [path]
<code>ls -l</code>	Lister le contenu du répertoire courant avec des informations supplémentaires
<code>ls -a</code>	Lister tous les fichiers, y compris les fichiers cachés dont le nom commence par un point (<i>dotfiles</i>)
<code>cd [path]</code>	[path] devient le répertoire actuel
<code>cd ..</code>	Passer au répertoire parent (notez l'espace entre «cd» et «..»)
<code>cd /</code>	Passer au répertoire racine (notez l'espace entre 'cd' et '/')
<code>cd ~</code>	Passage au répertoire d'origine (déterminé par la variable d'environnement \$HOME)
<code>mkdir [name]</code>	Créer le répertoire [name] dans le répertoire de travail actuel
<code>rmdir [name]</code>	Supprimer le répertoire vide [name] du répertoire de travail actuel
<code>rm [name]</code>	Supprimer le fichier spécifié
<code>rm *</code>	Supprimer tous les fichiers du répertoire de travail actuel
<code>rm -r *</code>	Supprimer tous les fichiers et les sous-répertoires du répertoire de travail actuel
<code>cp [from] [to]</code>	Copier un fichier de la source [from] vers la destination [to]
<code>cp -r [from] [to]</code>	Copier tout, y compris les répertoires, de la source [from] à la destination [to]
<code>mv [from] [to]</code>	Déplacer un fichier de la source [from] à la destination [to]
<code>mv -r [from] [to]</code>	Déplacer tout, y compris les répertoires, de la source [from] à la destination [to]
<code>find</code>	Rechercher des fichiers correspondant à certains modèles
<code>sudo [command]</code> <code>sudo raspi-config</code> <code>sudo reboot</code> <code>sudo shutdown -h now</code> <code>sudo apt-get install [package]</code> <code>sudo apt-get update</code> <code>sudo apt-get upgrade</code> <code>sudo chown pi:root [name]</code> <code>sudo su</code> <code>sudo ku</code>	Exécuter [command] avec priviléges élevés. Permet de faire des choses auxquelles vous n'avez pas droit. Parmi les exemples les plus courants, on peut citer: Lancer l'outil de configuration Raspberry Pi Redémarrer votre système en toute sécurité Arrêter le système en toute sécurité maintenant Installer un paquet Mettre à jour la liste des paquets sans rien installer Mettre à jour les paquets installés vers les versions obtenues avec 'apt-get update' Changer le propriétaire de [name] en «pi» et mettre le groupe en «racine» Devenir super-utilisateur pour plus d'une commande ?
<code>cat [name]</code>	Afficher le contenu d'un fichier
<code>head [name]</code>	Afficher le début d'un fichier
<code>tail [name]</code>	Afficher la fin d'un fichier
<code>chmod [who][+, -, =][permissions] [name]</code>	Modifier les autorisations d'un fichier
<code>chmod u+x [name]</code>	Ajouter une autorisation d'exécution pour le propriétaire du fichier
<code>chmod 777 [name]</code>	Autoriser chaque utilisateur à lire, écrire et exécuter le fichier [name]
<code>tar -cvzf [name] [path]</code>	Créer le fichier comprimé [name] à partir du contenu de [path]
<code>tar -xvzf [name]</code>	Extraire le contenu d'un fichier comprimé
<code>wget [url]</code>	Télécharger un fichier de l'internet
<code>man [command]</code>	Afficher la page de manuel pour une commande
<code>man man</code>	Voir la page du manuel de la commande «man»
<code>grep 'string' [name]</code>	Rechercher dans un ou plusieurs fichiers les occurrences de «string»



I'accélérateur de démarrage le plus performant d'Europe ?

HighTechXL Eindhoven, Pays-Bas

Terry Boyd, HighTechXL

HighTechXL est une société de capital-risque de technologies de pointe soutenue entre autres par Philips et l'Université d'Eindhoven. Pour réunir des équipes et attirer des talents (ingénieurs, directeurs commerciaux, spécialistes du marketing) autour des technologies innovantes du CERN et d'autres institutions de recherche, *HighTechXL* organise deux *FasTrackathons* par an. En cinq ans, *HighTechXL* est devenue une des entreprises les plus en vue d'Europe.

Presque tous les critères montrent que le succès de *HighTechXL* est aussi un moteur de développement économique pour Eindhoven et sa région : au moins 2 000 nouveaux emplois créés aux Pays-Bas et dans d'autres pays ; plus de 60 entreprises actives dans les industries de la prochaine génération et de nouvelles technologies.

En cinq ans, *HighTechXL* [1] est en passe de devenir la plus grande entreprise européenne de création de valeur. Et le tournant vers la *technologie profonde (deep-tech)* pris en 2018 rend cette évolution encore plus probable. En 2018, Guus Frericks, fondateur de *HighTechXL*, a annoncé que l'accélérateur de technologie passerait à un effort de création de

capital-risque de technologie plus pertinent pour l'écosystème de la ville d'Eindhoven. Celle-ci a produit des entreprises de semi-conducteurs incontournables comme ASML, qui fabrique en coulisse les machines d'où sortent les semi-conducteurs des géants tels qu'Intel. Le bilan de l'effort initial, qui remonte à 2013, est inégalé.

Certes, des entités telles que le *Y Combinator* et *500 Startups* ont elles aussi engendré des rejetons très célèbres et des milliards de revenus, mais l'une et l'autre opèrent déjà depuis plus longtemps. Leur taux d'échec est beaucoup plus élevé : environ 90 %. En 6 ans, plus de 60 % des entreprises *HighTechXL* ont survécu et 17 opèrent à l'échelle mondiale. Elles sont dans le 'Star Portfolio' et vont jusqu'à décupler les rendements.

À propos de HighTechXL

HighTechXL est une entreprise de création de capital-risque de technologies de pointe soutenue par l'*Eindhoven Startup Alliance* et ses membres (ASML, Philips, NTS-Group, l'Université de technologie d'Eindhoven (TU/e), BOM, *High Tech Campus*, EY, HVG Law et ABN AMRO). *HighTechXL*, grâce à un partenariat avec le centre de recherche néerlandais Nikhef (Institut national de physique subatomique), a accès à certaines des technologies les plus avancées au monde, y compris des accords pour les mettre sur le marché. Pour constituer des équipes autour de ces technologies issues du CERN et d'autres institutions de recherche, *HighTechXL* organise deux *FasTrackathons* annuels, qui attirent les talents et mêlent ingénieurs et physiciens aux directeurs commerciaux et marketeurs. Les nouvelles technologies y sont présentées aux participants qui se joignent ensuite aux équipes dans leurs domaines d'expertise et d'intérêt. *HighTechXL* prend une participation dans chacune des équipes qui sortent du programme de 9 mois, puis reste en collaboration étroite avec elles dans le cadre d'un programme de mise à l'échelle.

Démarrages réussis

- *Accerion* [2] (Venlo, Pays-Bas) spécialiste du positionnement pour les robots mobiles et les véhicules guidés autonomes, vend ses produits sur plusieurs marchés mondiaux et a récemment reçu un important *A Round* [3] soutenu par *Phoenix Contact Innovation Ventures*, basé en Allemagne, conjointement à l'agence de développement économique de la province du Limbourg (LIOF).
- *Amber Mobility* [4] (Eindhoven) a créé un réseau de covoiturage de véhicules électriques, et envisage le recours à des véhicules autonomes. Depuis sa première levée de 500 000 € auprès d'un cercle d'amis et de familles, *Amber* a levé des millions de capitaux, aussi auprès de *Pala Group BV*, et a implanté ses plateformes dans tous les Pays-Bas.
- *Bambi Medical* [5] (Eindhoven) a conçu un dispositif sans fil de surveillance des signaux vitaux pour prématurés et levé au moins 4 M€ auprès d'investisseurs privés et 2,4 M€ auprès de *Horizon2020*.
- *byFlow* [6] (Eindhoven) fabrique des imprimantes alimentaires 3D pour la restauration et a bénéficié d'une couverture



Soixante-quinze personnes ont participé au deuxième *FasTrackathon* proposé en juin par *HighTechXL*, où des équipes se sont formées autour de technologies avancées du CERN, du TNO et de Philips. À gauche : Corne Rentrop, chef de projet TNO dans le domaine de l'électronique hybride imprimée et flexible, était sur place pour discuter des possibilités de commercialisation des équipes de technologie de l'électronique imprimée. (Photo : *HighTechXL*)

média internationale (BBC, CBS Morning, etc.).

- *LifeSense Group* [7] fabrique des capteurs et des articles de santé portables utilisés pour détecter l'énucléose chez les adultes et les enfants. La société a levé plusieurs millions de dollars de capital d'investissement grâce à ses activités en Asie et prévoit une extension sur le marché américain.
- *Manus VR* [8], (Geldrop, Pays-Bas) a mis au point des gants de réalité virtuelle pour la formation à la RV et a construit un environnement de test. Parmi ses clients, *Manus* compte la NASA et plusieurs sociétés de jeux. En 2019, *Manus* a réussi un tour de table de 2 M€ auprès d'*Eindhoven Venture Fund II*.
- *Sustainable* [9] (Amsterdam) combine des déchets de plastique d'emballage recyclés avec du quartz afin de produire des plans de travail de cuisine et d'autres utilisations de la première pierre composite jamais obtenue par recyclage. Une levée de plusieurs millions d'euros de capital, dont 2,2 M€ auprès de *Horizon2020* leur permet d'envisager une extension mondiale.
- *ULU* [10] applique les techniques IoT aux véhicules, avec *Cartracker* pour le suivi des flottes de livraison. Avec plus de 15 000 abonnés, la société possède des bureaux à Amsterdam, Ljubljana, Londres et Shenzhen.

Alliance avec le CERN

En juillet 2018, *HighTechXL* est passé d'un accélérateur de démarrage de technologie de pointe aguerri à un effort de création d'entreprises de technologie de pointe.

Guus Frericks : « Nous nous sommes demandé : « Que se passerait-il si nous ne travaillions qu'avec des propositions pour lesquelles nous, en tant qu'instance régionale, pourrions faire la différence ? Partons-nous de *start-ups* de technologie de pointe ou prenons-nous comme point de départ la technologie révolutionnaire elle-même ? »

C'est là que *HighTechXL* a conclu une alliance historique avec

le CERN pour mettre sur le marché les dernières innovations technologiques du plus grand laboratoire au monde de physique des particules.

Frericks poursuit : « Le CERN a répondu : « Eh bien, s'il y a en Europe une région susceptible de faire la différence dans certains domaines, c'est bien Eindhoven ». Ce qui a renforcé notre confiance dans notre aptitude à exploiter ces technologies du CERN.

« Nous sommes passés du soutien aux équipes qui innovent à la recherche, auprès d'institutions de recherche, de technologies qui innovent, pour former ensuite les bonnes équipes susceptibles de commercialiser ces techniques.

« Le nouveau *HighTechXL* est nourri par l'écosystème local », poursuit M. Frericks.

La confiance des acteurs principaux – entrepreneurs, chefs d'entreprise et talents technologiques – a permis de mobiliser plus facilement la puissance technique collective de la région pour la nouvelle approche *HighTechXL*. Voici comment.

Ingénieurs, physiciens et professionnels invités

HighTechXL s'approvisionne en techniques de pointe auprès d'instituts de recherche tels le CERN [11], TNO et Philips.

Puis ils organisent un *FasTrackathon* – un marathon inversé – où les technologies et les domaines d'application potentiels sont mis en vitrine. *HighTechXL* assure la promotion du *FasTrackathon*, en invitant ingénieurs, physiciens et autres professionnels (experts en développement commercial, directeurs financiers et entrepreneurs) à se joindre aux équipes. Les individus et les équipes assistent à une séance d'idées d'une demi-journée, construisent un canevas de modèle d'affaires et présentent leurs idées en fin de journée.

HighTechXL recrute d'autres équipiers intéressés par la commercialisation de la technologie et organise des séances pour renforcer les équipes. Après un processus de sélection, les équipes entament un programme de création d'entreprises de 9 mois.



Au *FasTrackathon*, les papillons de couleur recueillent les idées qui jaillissent pendant les séances de remue-méninges brainstorming sur l'application des technologies de pointe. (Photo : HighTechXL)



Il peut faire chaud aux Pays-Bas... L'équipe de *Incooling* fait ce qu'elle sait faire de mieux : se rafraîchir. De gauche à droite : Karen Huaracha Magaña, Helena Samodurova, Hossein Beikai et Hejran Mehrtak. (Photo : Incooling)

Première cohorte de deep-tech

- *Incooling* [12], a présenté au CES 2020 [13] sa nouvelle technique de refroidissement des CPU et des GPU dans les centres de données. En 2019, des membres de l'équipe se sont rendus à Taiwan à plusieurs reprises et partout en Europe pour participer à des événements technologiques et de lancement de jeunes sociétés. Avec Helena Samodurova, l'équipe a remporté des compétitions de présentation à Amsterdam, Berlin et Taipei. Elle a levé plus de 600 000 € auprès d'investisseurs privés.
- Les membres de l'équipe *Dynaxion* [14] se sont rendu deux fois en tant que finalistes du *Opioid Detection Challenge*, financé aux États-Unis par le Département de la sécurité intérieure, le Service d'inspection postale et le Service des douanes. Ils ont également reçu 100 000 \$ dans le cadre de cet effort. *Dynaxion* utilise l'accélérateur de petites particules du CERN pour analyser colis et marchandises au niveau atomique, et détecter la présence de substances illicites.

• *Aircision* [15] utilise le laser pour construire le lien le plus fiable dans la matrice de communication 5G. En pourparlers avec les leaders de l'industrie des télécoms pour établir des partenariats, *Aircision* a été sélectionné comme *Deep Tech Pioneer* lors du salon *Hello Tomorrow* à Paris en mars 2020.

Le prochain groupe d'entreprises de technologies de pointe est en place depuis avril 2019, avec plusieurs start-ups prometteuses construites autour de la technologie du TNO et du CERN.

« Le succès de la première cohorte prouve qu'Eindhoven est une ville de technologie de pointe, soutient M. Frericks. Pour pousser ces propositions, il faut mettre au travail le réseau et la région. Notre idée de création d'entreprises autour d'un technique de pointe est la bonne, au bon endroit et au bon moment. Il faudra aussi un peu de chance. » ▶

(191194-02 VF)



Les membres de l'équipe *HighTechXL* se présentent à la nouvelle cohorte d'entreprises de technologies de pointe avant d'entamer le programme de création d'entreprise de neuf mois.

Liens

- [1] www.hightechxl.com/
- [2] <https://acerion.tech/>
- [3] <https://acerion.tech/news/acerion-closes-series-a-investment-plans-major-expansion/>

- [4] <https://driveamber.com/en/>
- [5] www.bambi-medical.com/
- [6] www.3dbyflow.com/
- [7] www.lifesense-group.com/
- [8] <https://manus-vr.com/>
- [9] www.sustainable.com/
- [10] <https://driveulu.com/en/>
- [11] <https://home.cern/>
- [12] www.incooling.com/
- [13] <https://www.ces.tech/>
- [14] <https://dynaxion.nl/>
- [15] www.aircision.com/



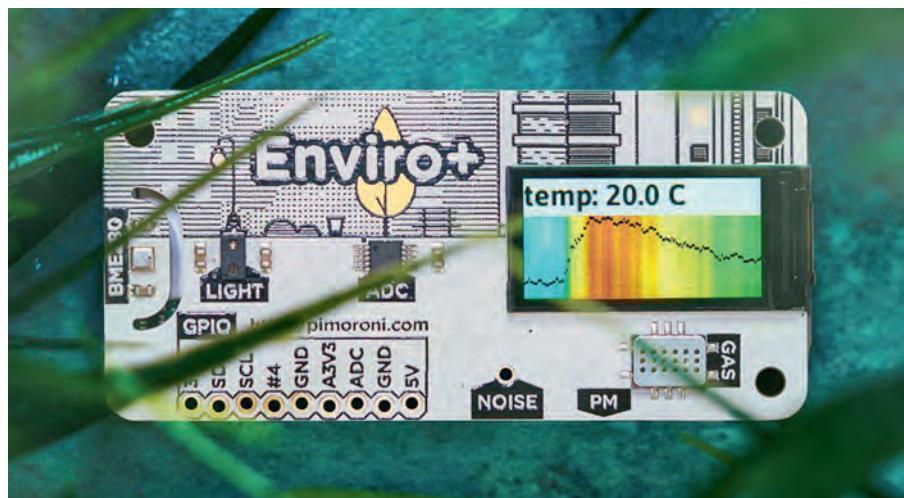
BANC D'ESSAI

HAT Enviro+ pour Raspberry Pi

Mesure et collecte de la qualité de l'air avec RPi et le HAT Enviro+

Thomas Scherer (Allemagne)

Pour une petite station de mesure de données environnementales à réaliser soi-même, la recette classique passe par une carte à microcontrôleur et des capteurs associés. Pour simplifier, prenez donc un RPi. Et pour aller plus vite, associez-y un HAT adapté comme l'Enviro+ de Pimoroni. C'est parti mon kiki.



La roue n'a été inventée qu'une fois, que je sache. Pourquoi recommencer sans cesse ? Inutile de faire votre propre étude de circuit à µC pour collecter des données environnementales essentielles. Il suffit d'une carte avec un µC approprié et une

carte d'extension toute faite plus des capteurs. Interconnectez tout ça et il ne restera qu'à tricoter le logiciel qui sonde les capteurs.

Si vous cherchez le confort, p. ex. un réseau WLAN, c'est l'ordinateur monocarte le plus répandu qui s'impose : le Raspberry Pi vous offre tout ce dont vous pourriez avoir besoin et bien davantage. Parmi ses nombreux avantages, il y a la pléthore de modules enfichables, prêts à l'emploi, sous la forme standardisée de HAT (*Hardware Attached on Top*), grâce auxquels on dispose d'une ribambelle de

capteurs différents. Il n'y a pas ou peu de bidouille sur le matériel, et pour le gérer, capteurs compris, on trouve le plus souvent des bibliothèques (Python) appropriées.

Enviro+

C'est exactement le cas d'Enviro+ [1], un petit HAT au format du Raspberry Pi Zero W [2], qui s'adapte à presque tous les modèles RPi grâce à sa compatibilité avec le connecteur à 40 broches. Il arrive emballé dans une pochette antistatique avec autocollant et du plastique à bulles – rien d'autre. Pas la moindre note, toute la doc est en ligne. Où il faut encore aller la chercher, ce qui n'est pas tout-à-fait une sinécure, comme on va le voir... Rien de tragique, mais tant qu'à donner dans le tout-numérique, Pimoroni aurait au moins pu mettre un code QR avec un lien.

Le code à barres donné n'est d'aucune utilité pour l'usager. L'URL imprimée aurait pu être raccourcie pour en faciliter la saisie manuelle. Et on aurait apprécié que cette URL conduise directement à une page qui... existe. Moi je suis tombé sur une pub pour un antique modèle de Peugeot «HTTP/1.0 404 Not Found». Tout ça est inutilement frustrant, ces errements auraient facilement pu être évités au client pressé de se mettre au travail... Espérons que d'ici que vous en serez là, ces petites négligences seront corrigées.

Caractéristiques

Voici de quoi est composée la station de mesure environnementale Enviro+ :

- BME280 : capteur de température, de pression atmosphérique et d'humidité [3].



Figure 1 : Voici ce que vous recevez : un circuit imprimé en sachet.

- LTR-559M : capteur de lumière et de proximité [4]
- MICS6814 : capteur de gaz analogique [5]
- ADS1015 : convertisseur A/N [6]
- SPH0645LM4H-B : microphone MEMS [7]
- afficheur en couleur OLED 2,5 cm 0,96 pouces 160×80 pixels
- connexion pour capteur de particules (en option) PMS5003 [8]

Assemblage

Le montage de la station de mesure est rapide : sortez la carte de sa pochette, embrochez-la délicatement sur un RPi et c'est presque fini.

Si c'est un RPi Zero W que vous avez, après avoir effectué le branchement du HAT (avec des entretoises appropriées), vous aurez devant vous un module joliment compact (**fig. 2**), facile à intégrer dans un petit boîtier. Le modèle fourni ne m'a pas convaincu, car il faudrait, pour obtenir des mesures faibles, une bonne ventilation du sandwich formé par les deux cartes. Et pour effectuer des mesures de luminosité, c'est un coffret transparent qu'il faudrait.

Comme le HAT Enviro+ tient aussi sur un Raspberry Pi normal, je l'ai mis sur un RPi 3B+ qui n'avait rien d'autre à faire. J'ai vite compris que le couvercle du boîtier en alu doit rester ouvert (**fig. 3**) pour que l'air et la lumière circulent jusqu'aux capteurs de l'Enviro+.

Ce que vous voyez-là, ce sont les radiateurs sur les puces. Le plus encombrant surtout, sur le SoC, pose des problèmes avec de nombreux HAT et l'Enviro+ ne fait pas exception. Par précaution, il est préférable de couvrir le dessous du petit circuit imprimé d'un morceau d'isolant (**fig. 4**) pour éviter tout court-circuit lorsque vous le branchez sur le RPi.

Le moment est alors venu de brancher Enviro+ sur l'embase à 40 broches du Raspberry Pi (**fig 5**). Les possibilités de ce HAT dépassent celles du GPIO. En effet, à l'avant, sur le bord de la



Figure 2 : Avec un Raspberry Pi Zero W, ça donne un sandwich compact.



Figure 3 : Avec un Raspberry Pi 3 ou 4, le couvercle doit être retiré si la carte est installée dans un boîtier opaque ou même métallique.

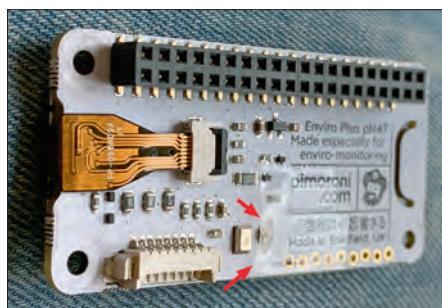


Figure 4 : Le dessous de l'Enviro+ protégé contre les courts-circuits par du ruban adhésif.



Figure 5 : Enviro+ avec son sparadrapp isolant sur un Raspberry Pi 3B+.

carte, il y a des pastilles identifiées, qui donnent accès à certains signaux intéressants.

Exemples et documentation

En dépit du lien rompu déjà mentionné, vous trouverez la documentation [9] sur le site de *Pimoroni*. Ce site est très bien fait et fournit des informations de base sur les capteurs inclus et la façon d'interpréter leurs valeurs.

J'ai particulièrement apprécié de trouver là des liens vers des instructions sur la façon d'installer facilement un Raspberry Pi de base, un savoir-faire recherché par de nombreux débutants. Ensuite, vous apprendrez à installer, avec quelques commandes de terminal, le logiciel supplémentaire pour Enviro+, y compris

Liens

- [1] Enviro+ : www.elektor.fr/enviro-environmental-monitoring-station-for-rpi
- [2] Raspberry Pi Zero W : www.elektor.fr/raspberry-pi-zero-wh-with-pre-mounted-header
- [3] BME280 : https://ae-bst.resource.bosch.com/media/_tech/media/datasheets/BST-BME280-DS002.pdf
- [4] LTR-559 : http://optoelectronics.liteon.com/upload/download/ds86-2013-0003/ltr-559als-01_ds_v1.pdf
- [5] MICS6814 : www.sgxsensor.com/content/uploads/2015/02/1143_Datasheet-MiCS-6814-rev-8.pdf
- [6] ADS1015 : www.ti.com/lit/ds/symlink/ads1015.pdf
- [7] SPH0645LM4H-B : <https://media.digikey.com/pdf/Data%20Sheets/Knowles%20Acoustics%20PDFs/SPH0645LM4H-B.pdf>
- [8] PMS5003 : http://www.aqmd.gov/docs/default-source/aq-spec/resources-page/plantower-pms5003-manual_v2-3.pdf
- [9] Pour commencer : <https://learn.pimoroni.com/tutorial/sandyj/getting-started-with-enviro-plus>
- [10] Station «Luftdaten» : <https://learn.pimoroni.com/tutorial/sandyj/enviro-plus-and-luftdaten-air-quality-station>
- [11] Carte de France des stations de mesure : <https://france.maps.luftdaten.info/#6/47.218/7.548>

```

pi@raspberrypi:~/enviroplus-python/examples
Datei Bearbeiten Reiter Hilfe
Pressure: 982.37 hPa
Relative humidity: 15.56 %
2019-12-08 20:40:58.165 INFO Temperature: 35.80 °C
Pressure: 982.38 hPa
Relative humidity: 15.58 %
2019-12-08 20:40:59.190 INFO Temperature: 35.80 °C
Pressure: 982.39 hPa
Relative humidity: 15.59 %
2019-12-08 20:41:00.215 INFO Temperature: 35.80 °C
Pressure: 982.40 hPa
Relative humidity: 15.57 %
2019-12-08 20:41:01.239 INFO Temperature: 35.79 °C
Pressure: 982.40 hPa
Relative humidity: 15.57 %
2019-12-08 20:41:02.264 INFO Temperature: 35.79 °C
Pressure: 982.40 hPa
Relative humidity: 15.56 %

```

Figure 6 : Température, pression et humidité mesurées par le capteur atmosphérique BME. Au voisinage du SoC du Raspberry Pi, ce capteur est bien au chaud...

```

pi@raspberrypi:~/enviroplus-python/examples
Datei Bearbeiten Reiter Hilfe
2019-12-08 20:42:26.028 INFO Compensated temperature: 20.13 °C
2019-12-08 20:42:27.938 INFO Compensated temperature: 20.27 °C
2019-12-08 20:42:28.948 INFO Compensated temperature: 20.41 °C
2019-12-08 20:42:29.958 INFO Compensated temperature: 20.41 °C
2019-12-08 20:42:30.968 INFO Compensated temperature: 20.39 °C
2019-12-08 20:42:31.978 INFO Compensated temperature: 20.39 °C
2019-12-08 20:42:32.988 INFO Compensated temperature: 20.38 °C
2019-12-08 20:42:33.998 INFO Compensated temperature: 20.52 °C
2019-12-08 20:42:35.008 INFO Compensated temperature: 20.67 °C
2019-12-08 20:42:36.018 INFO Compensated temperature: 20.69 °C
2019-12-08 20:42:37.028 INFO Compensated temperature: 20.56 °C
2019-12-08 20:42:38.038 INFO Compensated temperature: 20.55 °C
2019-12-08 20:42:39.048 INFO Compensated temperature: 20.41 °C
2019-12-08 20:42:40.058 INFO Compensated temperature: 20.23 °C
2019-12-08 20:42:41.068 INFO Compensated temperature: 20.08 °C
2019-12-08 20:42:42.078 INFO Compensated temperature: 20.34 °C
2019-12-08 20:42:43.088 INFO Compensated temperature: 20.47 °C
2019-12-08 20:42:44.098 INFO Compensated temperature: 20.46 °C
2019-12-08 20:42:45.108 INFO Compensated temperature: 20.60 °C
2019-12-08 20:42:46.118 INFO Compensated temperature: 20.87 °C
2019-12-08 20:42:47.128 INFO Compensated temperature: 20.87 °C
2019-12-08 20:42:48.138 INFO Compensated temperature: 20.86 °C
2019-12-08 20:42:49.148 INFO Compensated temperature: 20.86 °C

```

Figure 7 : Un autre exemple fournit des températures réalistes en pondérant la chaleur parasite rayonnée par le Raspberry Pi.

```

pi@raspberrypi:~/enviroplus-python/examples
Datei Bearbeiten Reiter Hilfe
Proximity: 0.00
2019-12-08 20:44:43.567 INFO Light: 57.41 Lux
Proximity: 0.00
2019-12-08 20:44:44.589 INFO Light: 57.41 Lux
Proximity: 0.00
2019-12-08 20:44:45.608 INFO Light: 59.18 Lux
Proximity: 0.00
2019-12-08 20:44:46.627 INFO Light: 58.30 Lux
Proximity: 0.00
2019-12-08 20:44:47.646 INFO Light: 57.41 Lux
Proximity: 0.00
2019-12-08 20:44:48.665 INFO Light: 58.30 Lux
Proximity: 0.00
2019-12-08 20:44:49.684 INFO Light: 60.07 Lux
Proximity: 0.00

```

Figure 8 : L'éclairement en lux avec avec le capteur de lumière. Cet exemple n'a rien donné en revanche avec le détecteur de proximité.

```

pi@raspberrypi:~/enviroplus-python/examples
Datei Bearbeiten Reiter Hilfe
Reducing: 384000.00 Ohms
NH3: 65621.61 Ohms
2019-12-08 20:47:19.483 INFO Oxidising: 1302.33 Ohms
Reducing: 387165.47 Ohms
NH3: 65980.20 Ohms
2019-12-08 20:47:20.555 INFO Oxidising: 1302.33 Ohms
Reducing: 387165.47 Ohms
NH3: 67200.00 Ohms
2019-12-08 20:47:21.627 INFO Oxidising: 1302.33 Ohms
Reducing: 387165.47 Ohms
NH3: 67943.66 Ohms
2019-12-08 20:47:22.699 INFO Oxidising: 1302.33 Ohms
Reducing: 387165.47 Ohms
NH3: 68696.36 Ohms
2019-12-08 20:47:23.770 INFO Oxidising: 1355.68 Ohms
Reducing: 390376.81 Ohms
NH3: 69458.25 Ohms
2019-12-08 20:47:24.842 INFO Oxidising: 1355.68 Ohms
Reducing: 390376.81 Ohms
NH3: 70229.51 Ohms
2019-12-08 20:47:25.914 INFO Oxidising: 1355.68 Ohms
Reducing: 390376.81 Ohms
NH3: 71272.93 Ohms

```

Figure 9 : Le capteur de gaz se compose de trois capteurs qui détectent différents composants de l'air. L'affichage en Ω est inhabituel et demande une conversion.

la bibliothèque Python et «Examples». C'est très vite fait et aussitôt après vous pouvez essayer les exemples. Les **illustrations 6 à 9** sont des copies d'écran de quatre exemples. Le petit afficheur (**fig. 10**) est particulièrement utile, si vous souhaitez planter l'électronique dans un endroit particulier, pour vérifier sur place son statut ou les valeurs collectées.

Verdict des géomètres de l'atmosphère

L'Enviro+ permet de créer facilement et à moindres frais une station de collecte de données environnementales. Grâce à la bibliothèque Python adaptée, l'interrogation des capteurs, etc. est simplifiée. Bien utiles sont aussi les exemples qui consistent en quelques lignes de code.

Mais ce n'est pas tout. Sur le site de *Pimoroni*, vous trouverez une notice de montage complète pour une station de mesure de l'air [10] connectée au nuage de données environnementales «luftdaten.info». Qui sera surpris de constater cette initiative allemande a été bien accueillie par les *géomètres de l'atmosphère* dans d'autres pays, y compris la France [11].

En un mot, bravo Enviro !

(191248-02 VF)

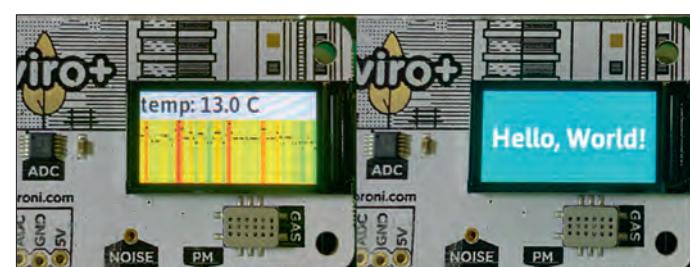


Figure 10 : Différentes informations peuvent être affichées. Pour cela aussi il existe des fonctions Python.



expérience vécue

Commandes de composants en Ukraine et en Russie

Ilse Joostens (Belgique)

Je repense parfois avec nostalgie au temps où je partais en sifflant sur deux roues acheter mes composants au magasin d'électronique du coin. Comme tout le monde, je suis tentée de dire que « c'était mieux avant ». Le bon temps, vraiment ? Non, tout n'était pas si rose avant. Combien de fois ne suis-je pas rentrée de courses bredouille, trempée par la pluie ou d'avoir pédalé contre le vent, frustrée de ne pas pouvoir poursuivre mon projet. Maintenant je commande tout en ligne, aux quatre coins du monde et ce n'est pas toujours un parcours de santé. *Caveat emptor* comme disaient les légionnaires *Fegaforus* et *Taxensus*.

Pour trouver du composant franchement exotique, vous êtes forcés de vous contacter des fournisseurs tout aussi exotiques. Cela vaut pour la verroterie électronique de qualité, encore assez facilement disponible en Russie et en Ukraine. On peut dire que l'achat de composants dans les pays de l'Est a son charme mais aussi ses pièges. Genre *Les tubes qui venaient du froid*, si vous voyez ce que je veux dire.

Techniques d'emballage russes : une introduction

Ma première horloge à tubes nixie, construite vers 2007 après Jésus-Christ, comporte des tubes IN-14. Ce n'était qu'un loisir à l'époque mais deux décennies plus tard, je suis toujours dans les tubes nixie et les composants antiques. Les ventes se portent plutôt bien comparées aux produits électroniques d'aujourd'hui.

Grâce à eBay, j'ai fait la connaissance de nombreux vendeurs de ces tubes. La plupart vivent en Ukraine et en Russie, mais aussi en Bulgarie, en Roumanie et en Moldavie, des pays qui

me font rêver, car, et je le confesse publiquement, j'ai une préférence immédiate pour Dracula et les films d'horreur. On apprend vite à distinguer des autres un colis postal en provenance d'un ancien pays communiste. Non seulement ils ont quelque chose en commun, mais chaque vendeur a son propre style. Voici une petite liste des caractéristiques communes et postcommunistes :

- apparence générale miteuse ;
- ficelle d'emballage (\pm réglementaire) ;
- carton enveloppé de papier d'emballage gris ou brun
- emballage cabossé (carton trop souple, trop mince) plus ou moins bien contenu par du ruban adhésif incolore ;
- grand nombre de timbres, tamponnés par une poste visiblement méticuleuse et musclée ;
- matériau de remplissage : feuilles de mousse ; papier journal ; sacs en plastique ;
- notes manuscrites étranges en caractères cyrilliques ;



Emballage soigné dans la mousse.



Recyclage de vieilles boîtes.

- boîtes recyclées de produits de consommation courante : articles de toilette pour femmes, jouets, tasses de café, pots de peinture.

Certains vendeurs vous enveloppent soigneusement vos tubes (qui étaient les leurs) dans des feuilles de mousse, d'autres vous les fourrent dans un sac en plastique. À tel point qu'en déballant certains paquets, j'ai l'impression de faire les poubelles. En dépit de ces techniques d'emballage qui nous paraissent insensées, neuf fois sur dix tout se passe bien. Ce qui n'exclut pas la casse ici ou là... il m'est arrivé de sortir la pelle et la balayette à la main pour récupérer ma marchandise au fond de la camionnette du livreur.

J'ai aussi reçu des tubes d'une saleté repoussante, comme s'ils avaient séjourné dans une cave humide pendant des années, et même une fois des tubes usés jusqu'à la grille. Dans de tels cas, je parviens généralement à récupérer mon argent, mais ces mésaventures restent agaçantes. Heureusement, pendant toutes ces années, je n'ai pas eu à déplorer de véritable escroquerie.

Quiconque achète des tubes en vrac sait qu'il faut compter avec dix pour cent de déchet.

Paiement

La plupart du temps ça passe par *PayPal*. Il y a quelques restrictions, en Ukraine surtout, sur les comptes *PayPal*. La plupart des vendeurs sont contraints d'utiliser des intermédiaires tels que *Western Bid* pour recevoir de l'argent via *PayPal*. Certains passent par des comptes étrangers. Ne vous inquiétez donc pas si on vous demande de passer par un compte *PayPal* d'un tiers inconnu, éventuellement en Thaïlande ou au Japon.

La douane à qui je déclare ma rage

Sur vos achats de marchandises en dehors de l'UE, vous devez généralement payer la TVA à l'importation plus parfois des droits d'importation. Pour les envois qui passent par DHL, UPS et Fedex, le traitement est en principe une formalité. Comme les tarifs de ces messageries dans les anciens pays de l'Est sont rédhibitoires, leurs ressortissants préfèrent expédier leurs colis par poste aérienne. Dans mon cas, ces colis sont livrés par la poste belge qui s'ingénie à compliquer les choses.



... ils arrivent donc cassés...

La procédure d'apurement complète est très longue (3 à 4 semaines), mais vous ne recevez pas toujours les documents nécessaires pour traiter ces frais dans votre comptabilité, de sorte que vous ne récupérez pas la TVA payée à l'importation. J'ai constaté que les erreurs sont nombreuses et généralement à mon désavantage. J'ai appris qu'il était inutile de réclamer un colis en souffrance, au risque de le voir renvoyé illiko à l'expéditeur sans attendre que le dossier soit traité. En cas de paiement indu, à quoi bon réclamer, puisqu'on me fera payer 85 € de frais de dossier ! Je sais de quoi je parle, pour avoir eu plus de 180 commandes en attente parce qu'un colis de composants était coincé... à la douane. Je ne téléphone plus dans un tel cas, car il m'est arrivé qu'un correspondant se venge en remettant mon colis sous la pile. Je n'ai curieusement jamais eu à payer un seul kopeck pour des colis envoyés de Chine par courrier ordinaire. J'ignore si Elektor est lu dans les bureaux de la douane belge, mais je me demande si elle a pour mission de faciliter le commerce avec la Chine laquelle pourtant n'est pas non plus un pays de l'UE.

Si un fonctionnaire belge ou un personnage influent me lit : faites quelque chose ! ► (191266-02 VF)



Étiquetage clair.



Intéressant pour les philatélistes.



BANC D'ESSAI

microscope Andonstar AD407

Est-il le meilleur de sa catégorie ?

Luc Lemmens (Elektor Labs)

On trouve depuis pas mal de temps des appareils USB bon marché présentés comme des « microscopes », mais que leurs caractéristiques ne qualifient pas pour fonctionner correctement dans un labo d'électronique, ne serait-ce qu'à cause du champ de travail trop exigu pour souder ou dessouder. Faute de microscope binoculaire pour souder les CMS, l'électronicien que je suis s'est donc longtemps contenté d'une grosse loupe éclairante, de lunettes spéciales et d'autres accessoires d'une efficacité inégale... jusqu'à ce que je fasse une rencontre.

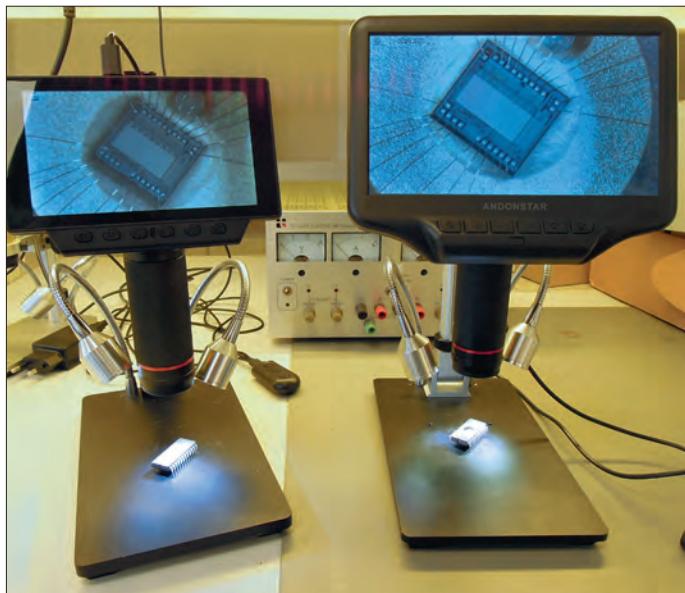


Figure 1. L'Andonstar AD407 (à droite) à côté de son (légèrement plus) petit frère.

Quand l'Andonstar A1 est arrivé sur mon plan de travail, j'ai senti que quelque chose avait changé : enfin un microscope abordable et raisonnablement stable, pour inspecter et modifier les soudures, et connectable à un PC par l'USB pour faire des copies d'écran. Peu après ce même fabricant proposait un modèle avec un petit afficheur LCD intégré, lequel permettait de se passer (au moins partiellement) d'un ordinateur : quelle amélioration ! Avec chaque nouveau membre de la famille Andonstar, les caractéristiques ont évolué, des fonctions se sont ajoutées et d'autres ont été améliorées. À telle enseigne que je pensais qu'après quatre modèles successifs proposés dans l'e-choppe d'Elektor, c'était fini. Or voici l'AD407, dont vous aimeriez sans doute comme moi savoir ce qui justifie son existence de cinquième de la dynastie. Ou serait-ce la cinquième roue du carrosse ?

Sa conception et son prix rapprochent ce nouveau microscope de son prédecesseur, l'ADSM302, dont les caractéristiques et spécifications nous serviront ici de référence pour évaluer l'AD407. Celles de leurs prédecesseurs ne font plus le poids.

Affichage et stockage d'images

Placez les deux microscopes l'un à côté de l'autre (**fig. 1**), vous remarquez immédiatement la différence de taille de l'afficheur. Avec plus de 17 cm, la diagonale du pavé LCD du nouveau modèle est plus grande de 5 cm que celle du précédent, ce qui est considérable. Bien que confortable, l'afficheur de 5 pouces de l'ancien modèle dans notre labo est la plupart du temps connecté via la prise HDMI à un moniteur externe de 25 cm qui offre une meilleure image.

La documentation déjà limitée d'Andonstar est assez floue sur les caractéristiques du capteur. Selon le fabricant, celui de l'AD407 fait 4 mégapixels, celui de l'ADSM302 ne dépassait pas 3 mégapixels.

Les deux microscopes offrent une résolution photo de 4032x3024, soit 12 mégapixels selon les spécifications. Ces valeurs correspondent à celles des photos (JPEG) produites par le microscope quand vous les ouvrez sur un PC. S'il y a plus de pixels, l'image du nouvel Andonstar devrait être meilleure que celle de l'ancien. Il ne va pas être facile de choisir entre les deux : l'ancien a un zoom numérique jusqu'à 4x, mais le nouveau ne va pas plus loin que 3x. La reproduction des détails est similaire, même si l'image de l'AD407 présente sensiblement plus de piqué.

La sortie vidéo par HDMI est bien meilleure sur le nouveau modèle tant pour la résolution que pour la fréquence d'images. Ceci n'est pas une amélioration cruciale pour l'usage que nous faisons de ce microscope au labo, mais ces spécifications meilleures seront déterminantes si vous avez à faire des présentations ou des démonstrations sur grand écran.

Avec l'AD407, le transfert d'images vers un ordinateur via USB n'est plus possible, ce qui ne me manquera pas. Avec d'une part la sortie HDMI et d'autre part le grand afficheur à cristaux liquides, je ne vois pas vraiment l'utilité de cette

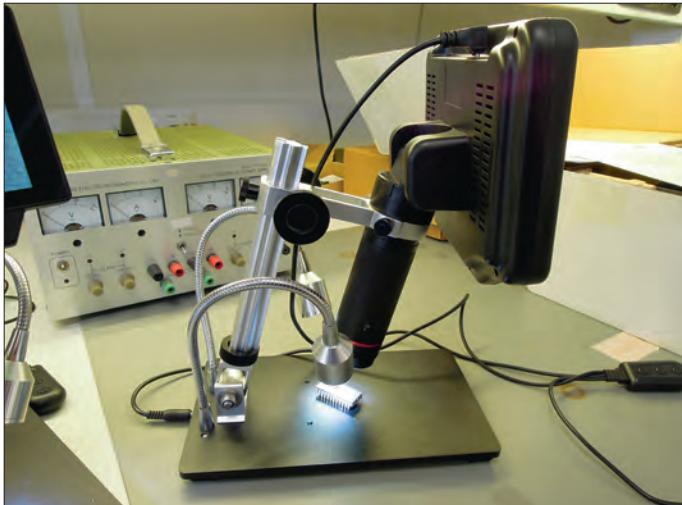


Figure 2. Il est également possible de voir les objets de biais.



Figure 3. Grâce à la prise USB coudée, le câble s'usera moins vite.

fonction qui d'ailleurs n'était pas dépourvue de problèmes sur les anciens modèles d'Andonstar. Si vous souhaitez enregistrer des images, photos et/ou vidéos, il est facile de le faire sur une carte micro-SD que vous lisez ensuite sur un ordinateur.

Colonne

Les deux modèles sont équipés d'une colonne solide. Pour l'éclairage de l'objet, ils sont équipés de deux mini projecteurs à LED montés sur flexibles, qui donnent certes beaucoup de lumière, mais sont presque toujours soit mal orientés, soit gênants. Quand vous souhaitez travailler sous le microscope, ce sont soit vos mains ou les outils que vous tenez qui butent contre les loupiotes, soit elles entrent dans le champ et vous ne voyez plus ce que vous faites.

Le réglage de la position des projecteurs n'est pas facile non plus, quand vous les relâchez ils ne gardent pas exactement la position souhaitée. Pourquoi s'accroche-t-on à ce système chez Andonstar alors qu'un anneau de LED autour de l'objectif serait bien préférable à tous égards ?

La mécanique des colonnes de l'ADSM302 et de l'AD407 n'est pas du tout la même. Selon les objets que vous souhaitez voir, ces particularités peuvent être décisives pour votre choix. Un premier grand avantage de la colonne de l'ADSM302 : le dégagement maximal entre l'objet observé et l'objectif est d'environ 12 cm. Avec la colonne du modèle plus récent, l'objectif n'est qu'à 8 cm du plateau. Il y a un remède à cela au prix d'un petit aménagement, mais dans le cadre d'une utilisation normale, le prédécesseur est plus commode à cet égard que le nouveau venu. Ce dernier possède en revanche une fonction supplémentaire fort utile : vous pouvez régler l'angle de la colonne (**fig. 2**) ce qui permet notamment d'examiner les points de soudure de biais. Cette option utile manque au prédécesseur.

Accessoires

L'AD407 est livré tout comme l'ADSM302 avec une alimentation et un câble muni d'une fiche jack pour l'éclairage ainsi qu'un câble avec une fiche micro-USB pour alimenter le microscope lui-même. La connexion USB du nouveau modèle est judicieusement

coudée, de sorte que le câble d'alimentation vieillira mieux que celui des anciens modèles (**fig. 3**).

Les deux microscopes sont fournis avec une télécommande infrarouge utile surtout si vous prenez des photos : dans ce cas, pour que l'image soit nette, il vaut mieux pour faire la photo appuyer sur le bouton d'un téléphone que d'avoir à toucher le microscope lui-même.

L'AD407 est livré avec une clé Allen pour l'assemblage de la colonne et, selon les instructions, un filtre UV qui manquait dans l'emballage que j'ai reçu pour ce banc d'essai. Je confesse que je n'en aurais d'ailleurs pas eu le moindre usage.

Reste à répondre à la Grande Question : si vous avez déjà un ancien modèle Andonstar, faut-il envisager l'achat du nouvel AD407 ? Si vous avez un A1, V160 ou un ADSM201, ma réponse est résolument **oui**. Si vous avez un ADSM302, il faut vous poser les bonnes questions.

Il est utile de pouvoir examiner un objet sous un angle en faisant basculer la colonne, mais avez-vous nécessairement besoin d'une autre colonne pour cela ? Vous pouvez aussi incliner l'objet.

Le grand écran à cristaux liquides du nouveau modèle en fait incontestablement un véritable microscope autonome. Si ici au labo d'Elektor nous avons souvent un grand moniteur branché sur l'ADSM302, il est probable qu'avec l'AD407 ce serait superflu, car son afficheur intégré est assez grand : presque 18 cm. Tout bien pesé, j'ai donc une nette préférence pour le nouvel Andonstar pour un nouvel achat. Et si j'avais déjà un ADSM302 et mon budget le permette, je le remplacerais certainement aussi par un AD407. ▶

(191154-03 VF)

@ WWW.ELEKTOR.FR

→ Microscope numérique Andonstar AD407

avec écran LCD de 7 pouces et sortie HDMI

www.elektor.fr/19079

Une soue ? Non, un labo d'électronique !

Visite guidée dans le saint des saints

Eric Bogers (Elektor, Pays-Bas)

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 du gagnant Philippe Demerliac, qui a aménagé son atelier dans une ancienne soue.



Figure : La photo gagnante.

Voici ce que nous écrit Philippe, que vous connaissez probablement déjà par son pseudo Cyrob, à propos de son labo et de sa passion :

« Mon père travaillait pour une grande entreprise d'électronique, il m'a légué du matériel d'occasion. C'est comme ça que j'ai pris goût aux instruments de mesure. Je veux savoir comment ils fonctionnent, je veux pouvoir les réparer, les améliorer et en concevoir moi-même de nouveaux.

« Dans les années 80, j'ai acheté une ferme au sud de Nantes, non loin de la côte, car ma femme élève des chevaux et a donc besoin d'espace.

« Longtemps je n'ai eu qu'un équipement réduit – un oscillo Tektronix 2225, une alim maison et un géné du même fabricant, etc... Comme tout le monde, je suppose.
« En 2004, j'ai décidé d'agrandir un peu mon petit labo à domicile ; j'ai opté pour la porcherie qui ne servait plus, avec



Figure 2 : Au début, le laboratoire paraissait immense...



Figure 3 : Établi et console en cours d'aménagement.

des murs de 80 cm d'épaisseur ! J'en ai isolé le sol et recouvert les murs de panneaux.

« Puis est venu l'établi avec beaucoup de prises, plus une console pour mes serveurs et mon système de vidéosurveillance.

« Mon laboratoire a trois circuits d'alimentation indépendants : l'un pour l'éclairage (pour ne pas être dans le noir quand un fusible saute), le deuxième que j'éteins quand je quitte les lieux (pour que l'équipement ne reste pas sous tension), et le dernier qui reste allumé en permanence (pour les alarmes, les serveurs et autres).

« C'est bien isolé et chauffé par quelques radiateurs électriques. Pour stabiliser l'hygrométrie, j'ai un déshumidificateur.

« Le matériel ne manque pas. Comme je suis aussi collectionneur il y en a beaucoup plus que ce dont j'ai besoin. Des choses relativement anciennes, mais j'ai aussi des appareils modernes comme ce fantastique générateur AFG31000.

« Vous ne serez pas surpris d'apprendre que ma plus grande préoccupation reste le manque d'espace. Faute de pouvoir

repousser ces murs épais, je dois me contrôler... et faire un peu de place pour les nouveaux arrivants.

« N'hésitez pas à me faire signe sur info@cyrob.org et visitez mon site [1] et ma chaîne YouTube [2]. ».

(191265-04)

Liens

[1] Site de Philippe Demerliac : www.cyrob.org

[2] Chaîne YouTube de Philippe Demerliac : www.youtube.com/c/PhilippeDemerliac_Cyrob



Figure 4 : Serveurs et système de télévision en circuit fermé.



Figure 5 : Un grand génér...

photosonde pour oscilloscopes

Mesure de fluctuation de luminosité des systèmes d'éclairage

Alfred Rosenkränzer (Allemagne)

La mesure de signaux électriques à l'oscillo ne vous suffit pas, vous voulez mesurer aussi des signaux optiques ? Bonne idée, mais il vous faut donc une sonde *optique* ou une photosonde qui convertisse les fluctuations de l'intensité lumineuse en tensions mesurables par un oscilloscope. Vous pouvez construire une telle sonde vous-même à faible coût.

Vous souvenez-vous de cet article sur les problèmes des lampes à LED intitulé *pollution électromagnétique causée par les lampes à LED* dans Elektor de mars/avril 2018, page 12 ? En faisant mes mesures de compatibilité électro-

magnétique pour cet article (voir le lien ci-dessous), je m'étais également intéressé aux variations de luminosité des lampes qui fonctionnent à 100 Hz ainsi qu'à la fréquence de commutation de leur alimentation à découpage.

Scintillement

Pour cette étude, j'avais construit un prototype de photosonde sur carte d'essais. Peu après, on a remplacé les lampes fluorescentes du bureau sans cloisons où je travaille par des lampes à LED à culot compatibles. Bien qu'assez silencieuses, ces nouvelles lampes scintillaient visiblement, à la fréquence du secteur ou un multiple.

Cela passait d'autant moins inaperçu que les écrans de nos téléphones de bureau semblaient soudain clignoter, en raison d'une interférence du clignotement des lampes avec le rafraîchissement des afficheurs. La plupart des appareils photo numériques souffrent également de telles interférences entre le scintillement d'une lampe et la fréquence de balayage du capteur (et de l'afficheur).

La généralisation de ce scintillement lié aux nouvelles lampes m'a incité à chercher à élucider ce phénomène. Et ça m'a motivé à mettre au propre mon prototype de photosonde, avec carte, boîtier et connexion par câble.

Le circuit

Le circuit de la sonde (**fig. 1**) est archi simple : la conversion de la lumière en signal électrique est faite par la photodiode D5, ici une BPW34, mais d'autres font aussi l'affaire, avec l'ampli op IC3a. La tension ainsi obtenue est multipliée

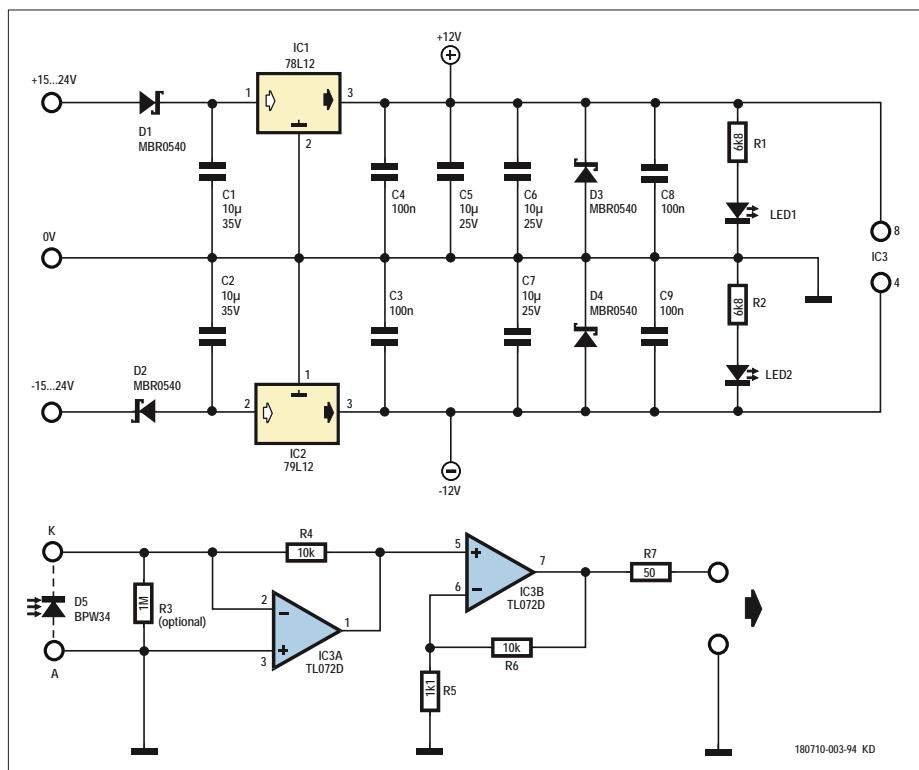


Figure 1. Schéma du circuit de la photosonde.

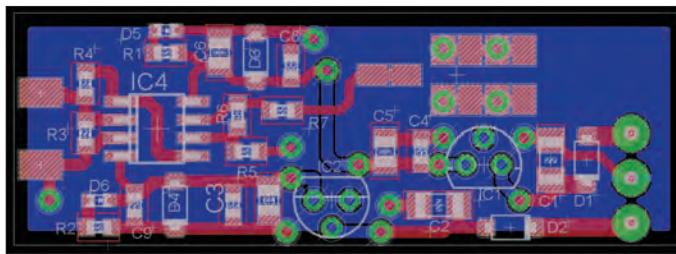


Figure 2. Disposition des composants de la sonde, alimentation comprise. Le fichier Eagle est disponible sur la page web de cet article.

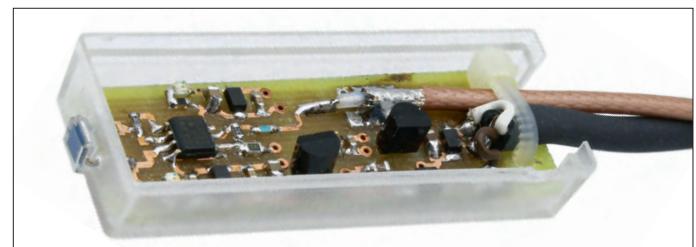


Figure 3. Ma photosonde avec circuit imprimé fraisé. À l'avant (à gauche sur la photo), vous pouvez voir la photodiode.

par dix par un autre ampli op (IC3b). La résistance R7 de $50\ \Omega$ assure l'adaptation d'impédance au câble coaxial connecté à la sortie.

IC3 est un double ampli op en boîtier SO8, mais d'autres types conviennent aussi. Le TL072, rapide et à faible bruit, utilisé ici présente l'avantage d'une forte impédance d'entrée (FET). La résistance R3 parallèle à la photodiode est optionnelle. Elle a été omise ici.

La photodiode et les amplis op sont alimentés par une tension symétrique stabilisée de $\pm 12\text{ V}$, fournie par les deux régulateurs de tension linéaires 78L12 et 79L12. Pour IC1 et IC2, j'ai opté pour la version traversante afin d'élargir le choix. Si les amplis op le supportent, vous pouvez également utiliser des versions avec une tension de sortie de 5, 8, 9 ou 15 V. La tension d'entrée des régulateurs devra être plus élevée d'au moins 3 V. Les diodes D1 et D2 servent de protection des entrées contre les inversions de polarité. Lors de la mise en marche et lors de la coupure de l'alimentation, D3 et D4 empêchent que la tension positive ne devienne négative et que la tension négative ne devienne positive. Les LED1 et LED2 indiquent la présence des tensions de service.

L'alimentation électrique peut être assurée par une alim de labo (symétrique) ou par un petit bloc d'alimentation composée à partir d'un transformateur de 1,5 VA avec par exemple au secondaire $2 \times 12\text{ V}$, un pont redresseur et deux condensateurs de lissage de $220\ \mu\text{F}$ chacun. Vérifier la rigidité diélectrique des condensateurs électrolytiques qui doivent tenir au moins 25 V. Évitez de surcharger la tension d'entrée des régulateurs – la plupart tolère jusqu'à 35 V.

Disposition

Le circuit imprimé à double face (**fig. 2**) a été dessiné pour un boîtier qui ressemble à celui d'une clé USB, lequel a déjà

fait ses preuves pour d'autres de mes sondes. Les vias sont placés de telle manière que la carte puisse également être produite par fraisage mécanique. Dans ce cas, les manchons doivent être insérés manuellement et soudés de part et d'autre de la carte.

Sonde terminée

Sur la **fig. 3**, vous pouvez voir mon exemplaire de la photosonde. Avec la photodiode BPW34 et un TL072, on a obtenu une réponse en fréquence de plusieurs centaines de kHz. Cela permet de voir facilement non seulement la fréquence du réseau (ou le double de sa fréquence), mais aussi de mesurer les fluctuations de luminosité à haute fréquence (**fig. 4**), qui résultent de l'ondulation de l'alimentation à découpage de certaines lampes LED (typiquement 30 à 60 kHz). On distingue aussi très bien le scintillement des lampes à LED commandées par un gradateur. Pour ces fréquences, il n'est pas nécessaire de charger le câble avec $50\ \Omega$ à l'entrée de l'oscillo. Une telle terminaison aurait d'ailleurs pour conséquence, en présence de fortes amplitudes, de malmenner la plupart des amplis op, car le maximum typique de 20 mA donne seulement 1 V à $50\ \Omega$. Le test de fonctionnement peut être effectué facilement avec une LED, alimentée directement par un générateur de fonctions. Une résistance série appropriée doit être utilisée pour les sorties à faible impédance. Si sur le générateur de fonctions utilisé il faut régler non seulement l'amplitude de la tension alternative, mais aussi son décalage (*offset*), alors ces deux réglages seront faits de

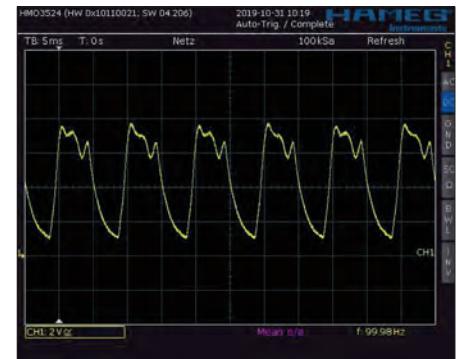


Figure 4. Oscilloscopie de l'ondulation de la lumière à 100 Hz, surmonté de composantes de fréquence plus élevée.

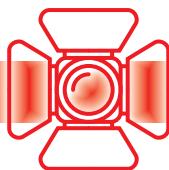
manière à ce que le courant maximum de la LED (typiquement 20 mA) ne soit pas dépassé et qu'un très faible courant (<1 mA) traverse la LED durant la demi-onde négative. Le courant peut facilement être visualisé sur l'oscillo comme une chute de tension au niveau de la résistance série. Le résultat est une lumière pulsée avec une luminosité minimale et maximale réglable.

Si votre générateur est un modèle plus simple, une onde carrée fera l'affaire. La tension sinusoïdale ou triangulaire produit d'intéressantes courbes écrêtées. Il va sans dire que la LED doit éclairer directement la photodiode. Tourner la photodiode pour inverser la polarité de son signal de sortie. Sur la page de cet article sur le site d'Elektor [1], vous trouverez le tracé de la carte au format *Eagle* ainsi qu'une vidéo qui montre bien l'interférence du scintillement des LED avec le capteur de la caméra. ■

(180710-04)

Liens

- [1] Vidéo et fichier CAO : www.elektormagazine.fr/180710-04
- [2] pollution électromagnétique causée par les lampes à LED : www.elektormagazine.fr/magazine/elektor-201803/41401/



projet TABULA – des nouveautés tangibles

De l'importance du retour d'information

Elektor Team (avec Christian Cherek, Chaire d'informatique 10 — Information des médias et interaction homme-machine, RWTH, Aix-la-Chapelle)

Le projet TABULA de l'Université RWTH d'Aix-la-Chapelle a progressé au cours des deux dernières années. Le matériel des *tangibles* semble au point maintenant pour assurer les tâches requises par différentes applications, notamment le retour d'information par un vibreur et des LED RGB. Il faut maintenant des programmes (d'apprentissage) pour tous les domaines imaginables.

Dans l'édition d'Elektor de sept.-oct. 2017, nous présentions cet intéressant projet de recherche de l'Université technique de Rhénanie-Westphalie d'Aix-la-Chapelle, en Allemagne, soutenu par Elektor. TABULA est basé sur des objets tactiles, appelés *tangibles*, (dé)placés sur une grande surface tactile capacitive [1]. Le système doit pouvoir déterminer à tout moment la position de chaque tangible sur la surface et transmettre cette information à l'ordinateur connecté. Cette utilisation de la reconnaissance tactile capacitive n'est pas un gadget de chercheurs comme le prouve le vif intérêt des utilisateurs potentiels : l'éventail des applications s'étend de l'enseignement de l'informatique à la production musicale et à la simulation d'apprentissage.

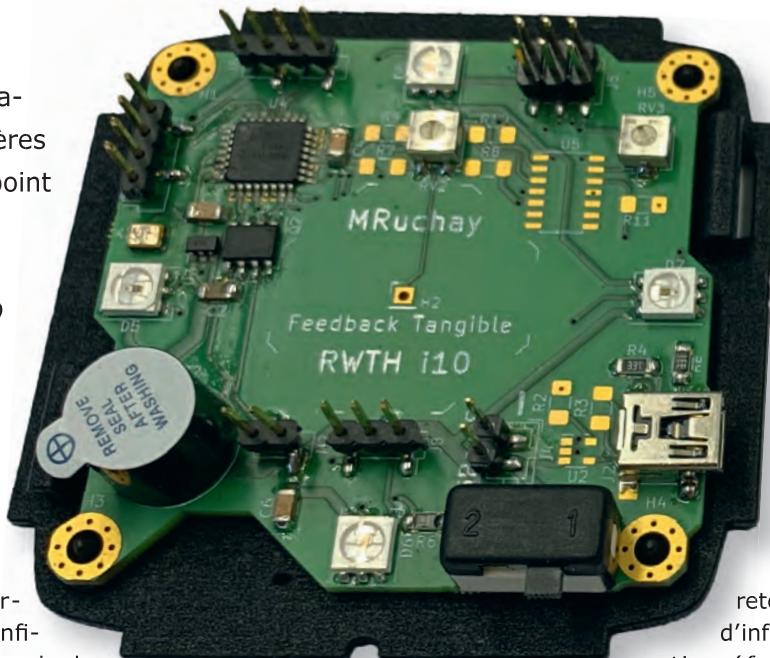
Récapitulation

Voici en bref les fonctions et les propriétés les plus importantes des tangibles. Les tangibles sont dotés d'un marqueur individuel qui permet à la surface tactile de les identifier. La fiabilité de la reconnaissance n'est pas une mince affaire, comme le montre l'article publié par Elektor [2].

Dans certaines configurations de la surface ou, si les tangibles sont trop rapprochés, les marqueurs sont insuffisants pour permettre de les distinguer clairement. Des mesures matérielles s'imposaient pour épauler le logiciel. Lorsqu'il est sur la table, un tangible envoie donc un signal de présence régulier au système via Bluetooth. D'autre part, la surface tactile envoie des éclairs de lumière aux tangibles détectés, lesquels reçoivent ce signal par un capteur de lumière. C'est la seule façon d'assurer une identification individuelle fiable.

Nouveautés

En deux ans, matériel et logiciels ont été perfectionnés, notamment par le retour d'information pour les utilisateurs. C'est ce que montre le circuit encore inachevé du «nouveau» tangible. Nous avons constaté qu'en cas d'erreur les personnes introverties et indécises se retirent plus rapidement et plus longtemps des jeux d'apprentissage que les personnes résolues et confiantes [3]. Différents canaux, permettent d'adapter individuellement le



retour d'information (feedback), grâce à quoi un participant qui se trompe n'a plus à craindre les réactions des autres. Il sera moins enclin à fuir le processus d'apprentissage.

Pour obtenir un retour haptique variable, nous avons ajouté des LEDs, un buzzer et un moteur vibrant sur la carte (fig. 1). Les quatre LED RVB NeoPixel D5 à D8 [4] sur les bords de la carte sont utilisées pour le retour d'information vers l'utilisateur. Elles peuvent être prolongées jusqu'en haut du boîtier par des canaux lumineux en plexi fluorescent qui en diffusent la lumière. Grâce aux quatre LED il est peu probable que l'information de retour soit jamais masquée entièrement par la main de l'utilisateur.

Le buzzer et le moteur (vibreur) fournissent également un retour d'information à l'utilisateur. Les trois variantes de rétroaction (et même les quatre LED) sont adressables séparément. Ainsi différents modes peuvent être essayés ou

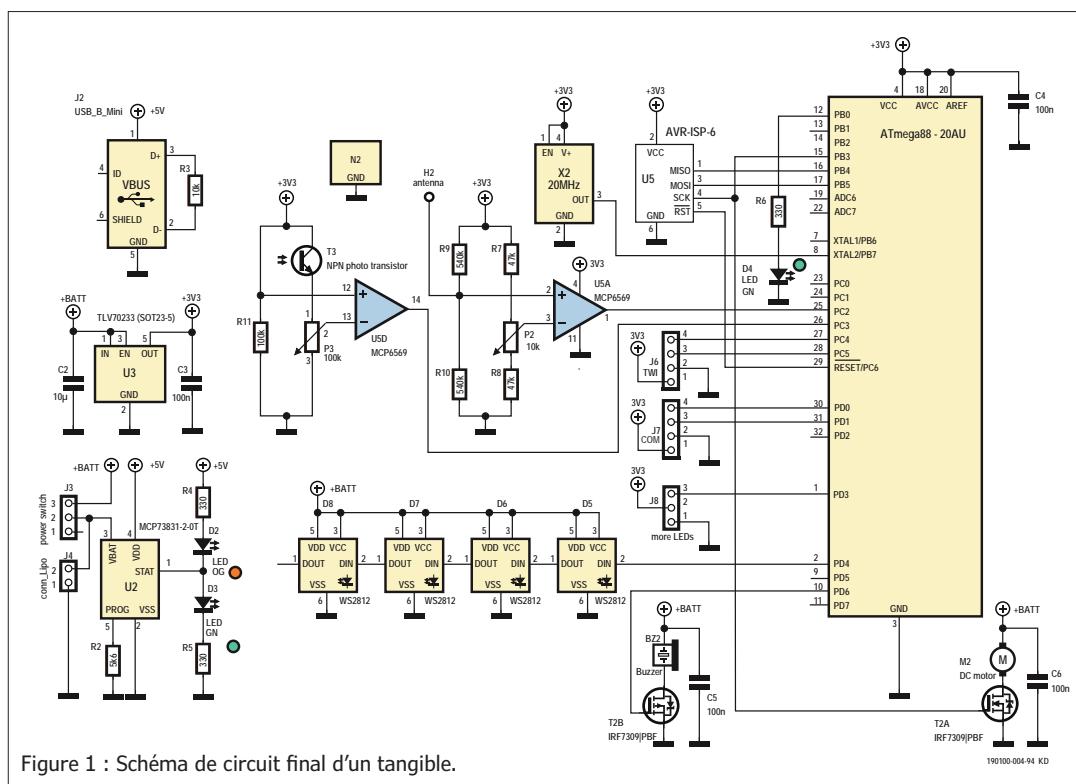


Figure 1 : Schéma de circuit final d'un tangible.

différents types d'information émis selon l'application. L'intensité du moteur et du buzzer est réglable. Des motifs mélodiques simples peuvent être joués. De sorte que les objets tangibles, entre les doigts de l'utilisateur ou posés sur la table, attirent mieux l'attention. Ceci stimule également la coopération autour de la table, car chacun capte plus rapidement ce qui se passe chez les autres [5]. Une LED (D4) tournée vers la table tactile a été ajoutée sous la carte. Commandée par le µC, elle s'allume lorsque le tangible est allumé. La nouvelle prise micro-USB

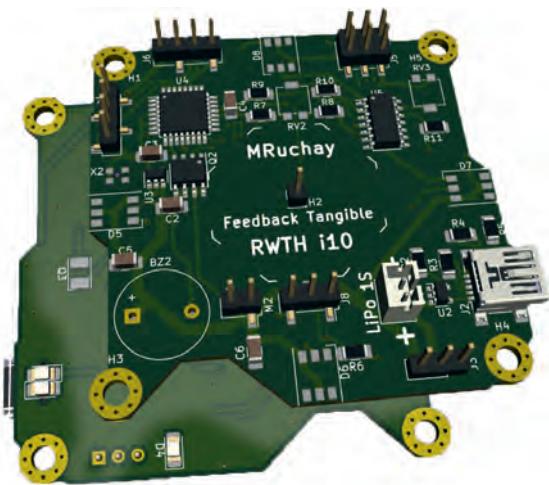
peut également être utilisée pour charger le tangible dans le nouveau boîtier. Il y a une électronique de charge pour une batterie LiPo et un interrupteur à glissière pour passer de la charge au fonctionnement. Deux autres LED (D2, orange, et D3, rouge) indiquent le mode (charge/normal). Le capteur de lumière encombrant du premier modèle a cédé la place à un CMS sous la carte.

Les nouvelles fonctions de retour d'information jouent un rôle important dans les premières applications TABULA (Demonstrator 1 et 2).

rôle d'opérateurs logiques NOT, AND, OR, NAND, NOR ou XOR assemblés en circuits numériques. Ils indiquent les niveaux logiques au moyen des LED. La couleur des LED RGB permet d'indiquer si le circuit est en mode édition (p. ex. blanc) ou simulation (p. ex. rouge/vert). Le matériel est terminé, mais il reste à faire.

L'étape suivante consisterait à financer des recherches de suivi dans le cadre desquelles les applications pourraient être effectivement utilisées et testées.

(190100-03)



Jens

- [1] RWTH Aix-la-Chapelle : <https://hci.rwth-aachen.de/TABULA>
 - [2] Article TABULA dans Elektor : www.elektormagazine.fr/160123
 - [3] Ehlenz et al. The lone wolf dies, the pack survives?:
Analyzing a Computer Science Learning Application on a Multitouch-Tabletop.
Koli Calling '18. ACM, New York, NY, USA, Article 4, 8 pages:
<https://doi.org/10.1145/3279720.3279724>
 - [4] LED NeoPixel : www.world-semi.com/DownLoadFile/108
 - [5] Cherek et al.. 2018. Tangible Awareness: How Tangibles on Tabletops Influence Awareness of Each Other's Actions. CHI '18. ACM, New York, NY, USA, Paper 298, 7 pages: <https://doi.org/10.1145/3173574.3173872>

COMMENT calculer le courant de court-circuit présumé ou PSCC et choisir le bon disjoncteur



Clemens Valens (Elektor Labs)

Le courant maximal susceptible de circuler dans un circuit court-circuité est dit *potentiel* par les uns, *prospectif* par les autres, ou encore *présumé*. Dans tous les cas, il est nécessaire de connaître son intensité pour bien choisir des dispositifs tels que les disjoncteurs et les fusibles afin d'éviter d'endommager ces circuits. Ah oui ? Et comment fait-on ?

Cet article porte sur l'électrotechnique plutôt que sur l'électronique. Les *circuits* mentionnés sont donc des circuits d'alimentation électrique avec des lampes, des interrupteurs, des prises de courant et des appareils, et bien sûr tout leur câblage.

Qu'est-ce que le PSCC ?

Nous retiendrons l'acronyme anglais PSCC pour désigner cette intensité maximale dans un circuit électrique court-circuité. On parle aussi de courant de *défaut disponible* ou de courant de *fermeture de court-circuit*. Comme tout courant, il est soumis à la loi d'Ohm. Par conséquent, la tension d'alimentation du circuit et son impédance déterminent ensemble la valeur du PSCC recherchée.

À quoi bon connaître le PSCC ?

Pour choisir les disjoncteurs et les fusibles qui protégeront une installation électrique, il faut connaître l'intensité du PSCC. Pour une protection fiable, ces dispositifs de protection doivent tenir le PSCC. Si leur pouvoir de coupure ou d'interruption est insuffisant, le PSCC risque de les détruire ou de provoquer un arc électrique (**fig. 1**). Dans les deux cas, le dispositif de coupure peut ne pas fonctionner correctement et des situations dangereuses peuvent survenir.

Et le courant de déclenchement ?

Le *pouvoir de coupure* d'un disjoncteur n'est pas son *courant de déclenchement* (= *trip current* en anglais). Ce dernier est l'intensité maximale qu'un disjoncteur peut conduire dans des conditions normales et en toute sécurité ; le courant de coupure est l'intensité supportée par le dispositif sans être endommagé. Ainsi le disjoncteur miniature (MCB) 1+N de type SN201 L C32-L d'ABB a un courant nominal de déclenchement (I_n) de 32 A et un courant nominal de court-circuit (I_{cn}) de 4,5 kA (sous 230/400 V_{AC}). Le courant de court-circuit dépend de la puissance de la source d'alimentation et n'est pas lié au courant tiré par la charge protégée par le disjoncteur.

Mesurer le PSCC

Vous pouvez mesurer le PSCC d'une installation électrique avec un testeur de boucle et de courant de court-circuit PSC. C'est un instrument facile à utiliser qui calcule l'intensité PSCC d'un circuit en ampères (A) et en kilo ampères (kA). Il suffit généralement d'appuyer sur le bouton *Test* pour obtenir la mesure, mais pour connecter correctement l'instrument, il faut bien comprendre ce que vous essayez de mesurer. En général, le test PSCC est fait au tableau de distribution entre phase (P) et neutre (N). Si le test est fait sur une prise

Tableau 1. Caractéristiques de mesure de la résistance de boucle du PeakTech 2715.

calibre	résolution	durée	précision à pleine échelle
20 Ω	0,01 Ω	25 A / 20 ms	
200 Ω	0,1 Ω	2,3 A / 40 ms	±2% de la pleine échelle ±5 u
2000 Ω	1 Ω	15 mA / 280 ms	

Tableau 2. Caractéristiques de mesure du courant de court-circuit présumé du PeakTech 2715.

calibre	résolution	durée	précision à pleine échelle
200 A	0,1 A	2,3 A / 40 ms	
2 kA	1 A	25 A / 20 ms	±2% de la pleine échelle ±5 u
20 kA	10 A	25 A / 20 ms	

de courant en utilisant le cordon de test moulé fourni avec le testeur, il sera effectué entre conducteurs de phase et de terre (PE ; E). Avant d'appuyer sur le bouton *Test*, assurez-vous que le testeur indique qu'il peut le faire en toute sécurité.

Que fait le testeur PSC ?

Un testeur PSC mesure d'abord la tension aux bornes du circuit ouvert (V_S , **fig. 2**). Ensuite, il applique une petite charge pendant un court instant pour faire circuler dans le circuit un courant d'une intensité connue (I_T , **fig. 3**). Pendant que ce courant d'essai circule, l'appareil mesure à nouveau la tension V aux bornes du circuit. En raison de l'impédance (Z_{LN}) du circuit, V sera un peu plus faible que V_S . Cette impédance s'écrit :

$$Z_{LN} = (V_S - V) / I_T [\Omega]$$

En supposant Z_{LN} constante, le testeur calcule le PSCC ainsi : V_S / Z_{LN} .

L'intensité du courant d'essai dépend de la plage de mesure sélectionnée et va de 2 A p. ex. à 25 A ou plus. La durée de la mesure varie aussi avec la gamme. Elle est de quelques dizaines de millisecondes. Les **tableaux 1** et **2** donnent des mesures effectuées avec un testeur PSC réel.

Qu'est-ce que le test de boucle ?

Un testeur PSC mesure également l'impédance de la boucle de défaut à la terre Z_S ou Z_E ou le courant de défaut prospectif PFC (aussi appelé PEFC). Cela correspond à l'impédance du circuit entre les conducteurs P et (P)E en cas de court-circuit entre les deux (**fig. 4**). Une faible impédance entraînera un courant de défaut élevé, qui déclenchera rapidement les protections. Il permet également d'assurer une petite différence de potentiel entre le conducteur (P)E sur la prise et la terre sur laquelle vous vous trouvez.

Mesurée sur une prise de courant, l'impédance de la boucle de défaut à la terre est appelée Z_S . Mesurée à l'entrée de l'alimentation, elle sera appelée Z_E . On aura donc :

$$Z_S = Z_E + R_L + R_E [\Omega]$$

où R_L et R_E sont l'impédance respective des conducteurs P et (P)E entre le point d'entrée de l'alimentation et la sortie.

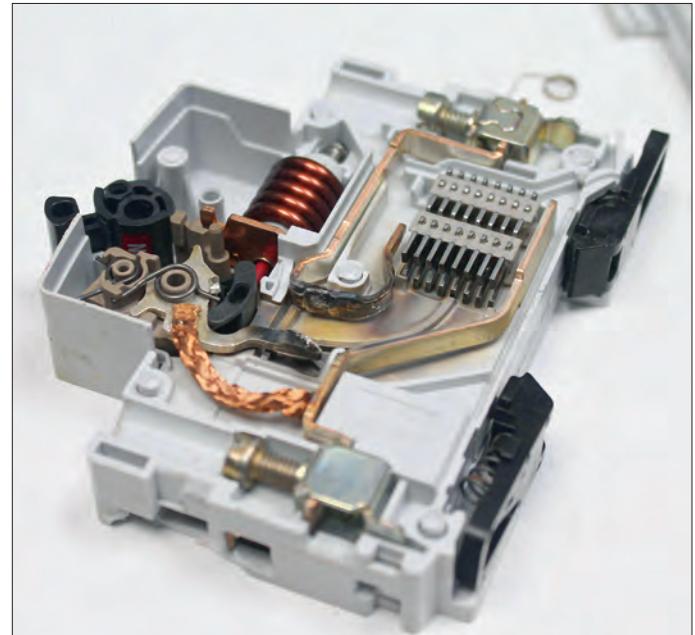


Figure 1. Voici ce qui peut arriver à un disjoncteur miniature (MCB) quand l'intensité du courant de court-circuit est trop forte pour lui.

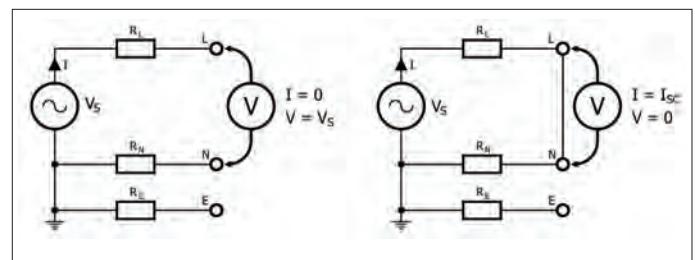


Figure 2. Mesure de la tension en boucle ouverte (à gauche) et du courant de court-circuit en boucle fermée (à droite).

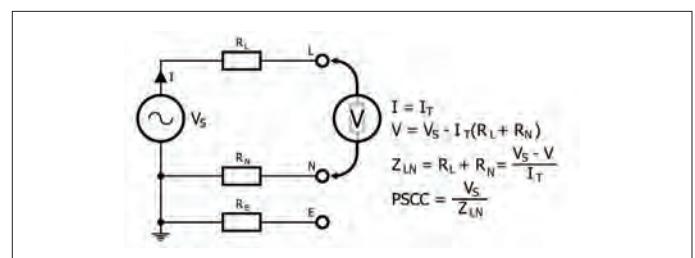


Figure 3. Pour calculer l'impédance du circuit, le testeur PSC y fait circuler un courant de test d'une intensité connue.

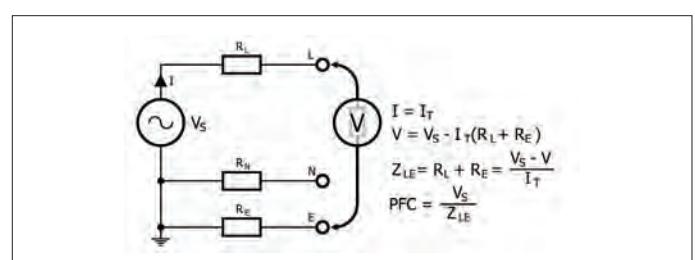


Figure 4. Calcul de l'impédance de défaut à la terre Z_S (ou Z_E) ou du courant de défaut à la terre P(E)FC.

Même avec un faible courant d'essai, un dispositif à courant résiduel (DCR) peut se déclencher lorsqu'il est sensible ou lorsqu'il y a une fuite dans le circuit testé. Pour éviter cela, vous pouvez contourner temporairement le RCD (n'oubliez pas de rétablir ses connexions après !)

Calculer la PFC

L'impédance de la boucle de défaut à la terre est mesurée de la même manière que le PSCC, mais là encore, selon la plage sélectionnée, le courant d'essai pourra se réduire à quelques dizaines de milliampères. D'autre part mesurez entre les conducteurs de phase et de terre plutôt qu'entre phase et neutre. Le P(E)FC est obtenu en calculant V_s/Z_{LE} .

Remarques

Selon le câblage, il arrive que les valeurs PSCC et P(E)FC diffèrent. Retenez alors l'intensité la plus forte pour le choix du disjoncteur approprié.

Vous courez un risque mortel en intervenant sur des circuits électriques sous tension. Vous voilà prévenu ! ►

(191160-02 VF)



Lien

[1] Vidéo : Banc d'essai du testeur de boucle et de PSC PeakTech 2715 : <https://youtu.be/9dBhz3acowc>

Types de réseaux de distribution électrique

Si vous vous intéressez au réseau électrique domestique, vous rencontrerez les abréviations TT, TN ou TN-S et d'autres que voici par ordre alphabétique :

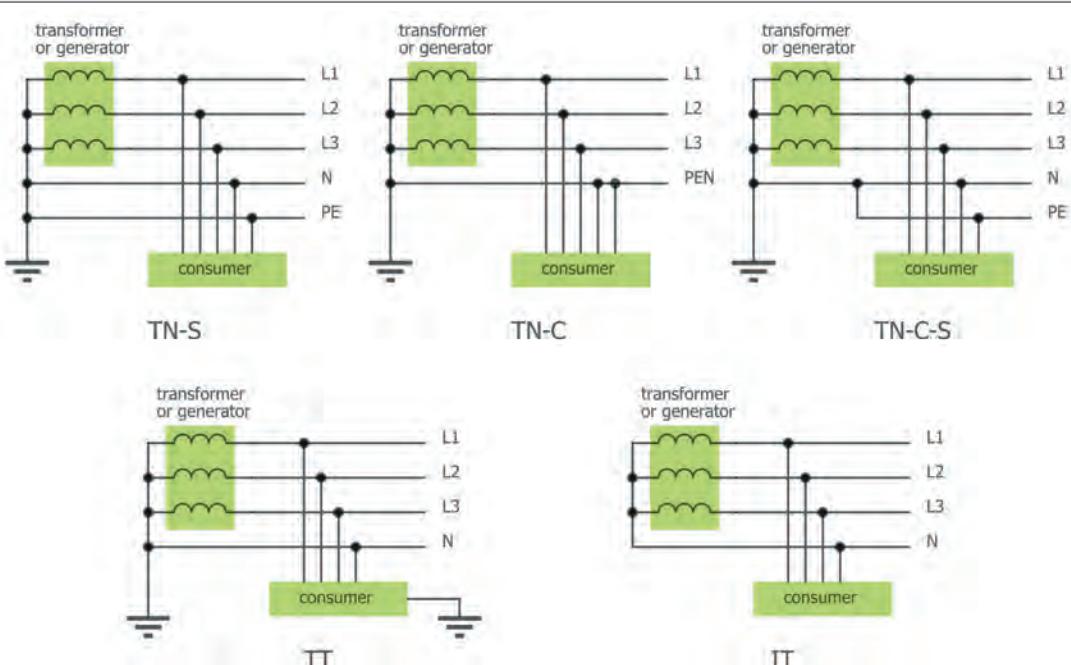
- C - Combiné
- I - Isolé
- L - Ligne
- N - Neutre
- S - Séparé
- T - Terre
-

Notez que ces dénominations sont internationales.

Les réseaux TN sont courants dans les réseaux domestiques et industriels en Europe. Un tel réseau dispose d'un seul raccordement à la terre de protection (PE) du côté de l'entrée ou du générateur auquel les consommateurs sont raccordés. Dans un réseau TN-S, la connexion se fait au moyen de fils N et PE séparés. C'est la topologie la plus sûre. Dans un réseau TN-C, les fils PE et N sont combinés en un seul fil PEN. Pour compliquer les choses, les réseaux TN-S et TN-C peuvent être combinés pour for-

mer un réseau TN-C-S. Ils sont également connus sous les noms de réseaux PME (mise à la terre multiple de protection), MEN (neutre multiple à la terre) ou MGN (neutre multi-terre).

Dans un réseau TT, le générateur et le consommateur ont tous deux leurs propres connexions de terre de protection (p.ex. des piquets de terre) ; il n'y a entre eux ni fil de terre ni mise à la terre. Un réseau informatique n'a aucune mise à la terre.



Monsanto MAN1 LED Display

drôle(s) de composant(s)

Neil Gruending (Canada)

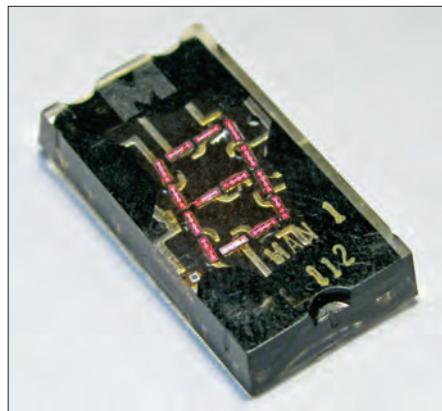


Figure 1 : Construction de l'afficheur MAN1 de Monsanto [1].

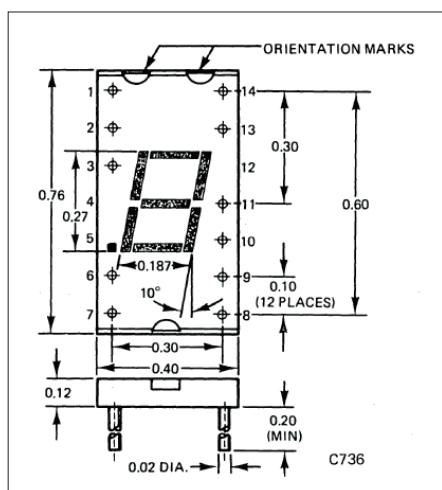


Figure 2 : Dimensions de l'afficheur MAN1 de Monsanto [2].



L'idée de représenter les caractères alphanumériques avec sept segments date des années 1900, pour la transmission de caractères par télégraphe puis leur impression sur une bande de papier. Peu après sont apparus des afficheurs à 7 segments où des lampes à incandescence illuminaient les différents segments. Ces afficheurs ne sont devenus populaires qu'avec l'apparition de versions miniaturisées à LED, au début des années 70. En voici un des premiers exemples, le Monsanto MAN1 de *Monsanto AlphaNumeric*.

Les afficheurs à 7 segments à base de LED ont emporté la mise grâce à leur faible consommation comparée à celle d'autres afficheurs, comme les tubes Nixie, les Minitrons et les afficheurs Panaplex. C'est aussi cette modeste consommation d'énergie qui rendit possible l'utilisation d'appareils à piles comme des horloges comme des montres-bracelets sans aiguilles telles le *Pulsar Time Computer*, ou des calculatrices comme la HP35. La durée de vie des batteries se comptait alors en heures. On en sourit aujourd'hui. Les concepteurs s'ingéniaient à la prolonger au maximum en éteignant les afficheurs dès que possible.

Le MAN1 de *Monsanto* utilisait deux matrices de phosphate d'arsénure de gallium (GaAsP) contenant quatre LED pour éclairer chaque segment, soit un total de 57 diodes, point décimal compris (**fig 1**). La hauteur d'un chiffre était 6,8 mm (**fig 2**), ce qui a facilité leur intégration dans d'innombrables applications portables comme les calculatrices. Souvent sous des lentilles de grossisse-

ment, pour faire paraître les chiffres plus grands. La luminosité des LED n'était pas ce qu'elle est maintenant. Aussi les disposait-on derrière un filtre rouge pour en augmenter le contraste. Ce halo rouge a marqué toute une époque. Le boîtier du MAN1 était en plastique transparent. Celui de la version MAN1A était un plastique rouge qui cachait le circuit interne (**fig 3**).

Poussées par la féroce concurrence, les technologies d'affichage à LED ont évolué assez rapidement. HP a bientôt présenté des matrices de points 5×7 à LED avec décodeurs décimaux à codage binaire (BCD) intégrés. En 1977, Litronix, une filiale de *Monsanto*, sortait un afficheur à LED intelligent, avec pilotes de rangées /colonnes et ROM de caractères intégrée. Ces premiers afficheurs MAN1 ont quelque chose d'attendrissant. Si vous souhaitez en savoir plus, vous trouverez des informations [**2**] chez *General Instruments*, qui a absorbé *Monsanto* en 1979 [**3**].

190383-04

Liens

- [1] Afficheur Monsanto MAN1 : www.decadecounter.com/vta/articleview.php?item=463
- [2] Dimensions du Monsanto MAN1 : <https://bit.ly/2BuRRgf>
- [3] Historique des afficheurs Monsanto : www.datamath.org/Display/Monsanto.htm

filtres analogiques à capacités commutées

Ceci n'est pas la rubrique Rétronique même si elle s'appuie sur un principe décrit en... 1873 !

Robert Lacoste (Chaville)

J'adore relire des anciens magazines d'électronique dont bien sûr ceux d'Elektor que je garde depuis le n° 1, ce qui ne nous rajeunit pas... Les articles de ces années-là sont une superbe source d'inspiration. À l'époque, il n'y avait pas de microcontrôleurs, les concepteurs devaient être ingénieux ! Les techniques utilisées restent pour la plupart d'actualité. Prenons cet article intéressant sur les filtres à capacités commutées du numéro de janvier 1981 d'Elektor.

Savez-vous de quoi il s'agit ? Les filtres sont des circuits présents dans de nombreux projets, tout électronicien doit savoir se débrouiller pour bricoler et mettre au point un tel circuit.

Eh bien, voici donc un bon sujet pour le dernier épisode de cette série d'articles « hors circuits ». Mais avant de finir, il faut commencer.

Un filtre élémentaire...

Comme toujours, j'évite ici les concepts théoriques compliqués, nous préférerons tous les exemples concrets. Commençons par le filtre le plus simple : le filtre RC d'ordre 1, composé d'une résistance série et d'un condensateur vers la masse. Tout changement rapide du signal d'entrée est ralenti par le temps de charge et de décharge du condensateur à travers la résistance. En freinant les variations rapides du signal, ce filtre en atténue donc les fréquences élevées : c'est un filtre passe-bas. Jetez un œil à la simulation de ce circuit avec un signal carré de 500 Hz appliqué sur son entrée (**fig. 1**). La sortie est un signal vaguement carré, mais avec des montées et descentes amorties, caractéristiques d'un tel filtre RC. J'ai effectué les simulations pour cet article avec le logiciel Proteus VSM (*Labcenter*), mais vous les reproduirez facilement avec n'importe quel logiciel compatible Spice, comme l'excellent et gratuit LT-Spice (*Analog Devices*). Analysons maintenant ce circuit, en supposant p. ex. qu'à un moment donné la tension d'entrée est supérieure à la tension du condensateur. Un courant I traversera la résistance et chargerà



le condensateur. Je ne vous ferai pas l'affront de vous expliquer que l'intensité de ce courant est donné par la loi d'Ohm, hein ?

$$I = U/R = (U_{\text{entrée}} - U_{\text{condensateur}}) / R = (U_{\text{entrée}} - U_{\text{sortie}}) / R \quad (1)$$

Comment se comporte le condensateur maintenant ? Il se charge, oui, mais de combien de volts ? Si l'on considère un petit intervalle de temps dt, la charge Q stockée dans le condensateur sera augmentée de :

$$dQ = I \times dt \quad (2)$$

condensateur commuté

Dans toutes les branches, dans toutes les applications possibles de l'électronique, il faut filtrer certains signaux. Dans certains cas, il faut sélectionner certains signaux dans un but précis : par exemple, la sortie d'un amplificateur peut dépendre de la possibilité de sortir un signal en diminution à la fin de toute un foulée de réactions simultanées. On peut utiliser des méthodes électroniques pour faire aussi efficacement que possible tout ce qui précède. Il existe donc tout un arsenal de filtres passe-haut, passe-bas, passe-bande ou réjecteurs. Leur conception porte des noms fameux comme Butterworth ou Chebychev.

Il est possible de faire un peu étrange, mais c'est possible, comme le montre la figure 1a, de faire varier la résistance d'un filtre lorsque se trouve une tension u_1 de l'autre une tension u_2 . Pour simplifier, nous supposons que u_1 est supérieure à u_2 . Un courant I va traverser alors la résistance R de gauche à droite. D'après la loi d'Ohm :

$$R = \frac{u_1 - u_2}{I}$$

D'arrière en avant

La figure 1a représente la "fausse résistance". L'interrupteur S bascule continuellement l'arrangement en avant (et vice-versa) entre u₁ et u₂. Soit la fréquence à laquelle cela se produit. Chaque fois que S est dans la position a, le condensateur C se charge jusqu'à la tension u_1 , chaque fois que S est dans la position b, le condensateur se décharge jusqu'à la tension u_2 (en supposant toujours que u_1 est supérieure à u_2). Cela permet de transférer une charge de u_1 à u_2 , et c'est exactement ce que fait une résistance ; toutefois, une résistance fonctionne avec un courant constant, tandis que le condensateur commuté fonctionne par pas. Néanmoins, dans la mesure où la fréquence f du commutateur est suffisamment élevée, vous ne verrez pas la différence.

Il est facile de calculer la "résistance" du condensateur commuté représenté sur la figure 1b. Au moment où l'interrupteur S passe de la position a à la position b, C est chargé à la tension u_1 . Cela signifie que la charge de C (d'après la définition d'un condensateur) est égale à $C(u_1)$ (coulomb). Dès que S passe de b à a, C se charge à la tension u_2 et le condensateur contient une charge égale à $C(u_2)$. En d'autres termes, commuter S d'arrière en avant de u_1 à u_2 provoque le transfert d'une quantité de charge égale à

$$C(u_1 - u_2) = C(u_1 - u_2)f$$

Comme la fréquence de commutation est f, S effectue f cycles de commutation complets par seconde. La charge transférée par seconde est donc égale à $C(u_1 - u_2)f$. Et la charge (transférée) par seconde est identique à un courant. lorsqu'on porte cette expression dans la

$$R = \frac{u_1 - u_2}{C(u_1 - u_2)f} \text{ ou encore}$$

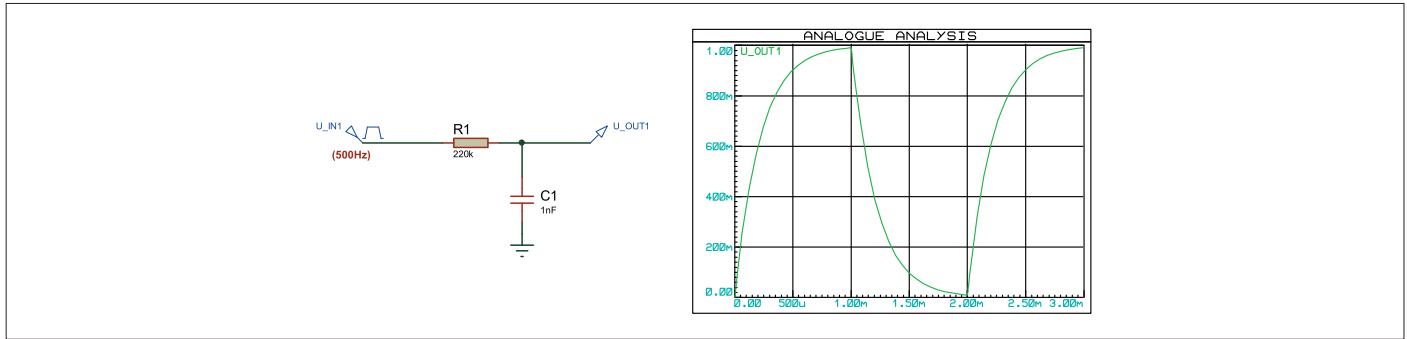


Figure 1. Un filtre passe-bas RC, simulé avec ici une onde carrée en entrée.

En combinant ces deux équations, vous trouverez que :

$$dQ = (U_{\text{entrée}} - U_{\text{sortie}}) \times dt / R \quad (3)$$

Par ailleurs la charge Q d'un condensateur est simplement égale à la tension à ses bornes multipliée par sa valeur en farads, donc ici :

$$Q = C \times U_{\text{sortie}} \quad (4)$$

d'où :

$$dQ = C \times dU_{\text{sortie}} \quad (5)$$

Il suffit de remettre tout ça avec l'équation 3 et de bien agiter pour obtenir :

$$U_{\text{entrée}} - U_{\text{sortie}} = RC \times dU_{\text{sortie}} / dt \quad (6)$$

Pas de panique avec les formules, celle-ci sera la dernière... de ce paragraphe. Vous retiendrez que c'est une équation différentielle. Vous pourriez même vous amuser à la résoudre pour obtenir une formule donnant la tension de sortie en fonction du temps, et ce sera de beaux tronçons d'exponentielles (fig. 1). Notez que la fréquence de coupure sera proportionnelle à $1/RC$.

Passer de R à C ?

Laissons là ce brave filtre RC et intéressons-nous aux filtres à capacités commutées. Tout d'abord, quels sont les inconvénients d'un filtre RC ? Imaginez que vous ayez besoin de faire varier facilement la fréquence de coupure du filtre, par exemple en la pilotant par un microcontrôleur. Comment faire avec un filtre RC ? Vous n'aurez pas d'autre solution que de faire varier électriquement la valeur de R ou de C, ce qui n'est pas si simple à faire avec précision.

À la fin des années 70, les fabricants de circuits intégrés ont cherché à concevoir des filtres performants et flexibles sous forme de circuits intégrés et ont buté sur cette difficulté. De plus, la technique d'implantation des filtres RC sur du silicium est complexe, alors qu'il est facile de réaliser des condensateurs de petite valeur sur une puce. D'où l'idée de remplacer la résistance et le condensateur... par deux petits condensateurs et un peu plus d'électronique. C'est ce qui a donné l'architecture dite à *capacités commutées*. Vous verrez pourquoi cette solution, en prime, permet de régler facilement la question de la fréquence de coupure du filtre. Comme le souligne Philip Golden (*Intersil*) dans son article historique [2], il est intéressant de savoir que cette idée a été décrite par James Clerck Maxwell en... 1873 !

Revenons à l'électronique. La **fig. 2** montre la cellule de base

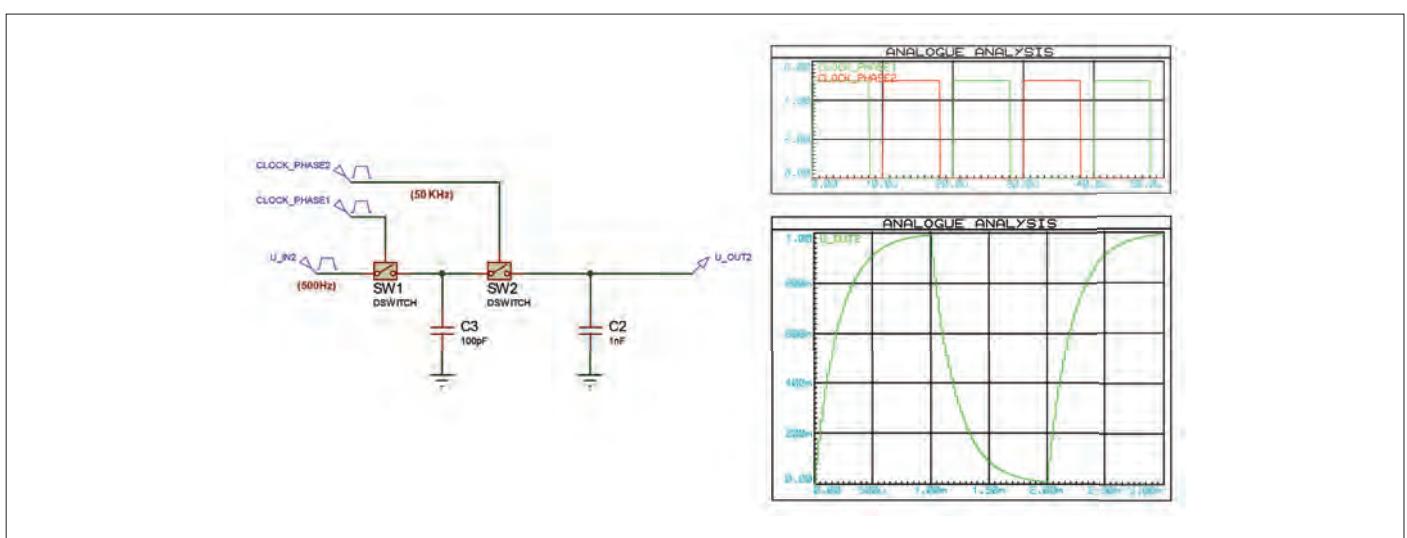


Figure 2. Un étage de filtre à capacité commutée. La résistance est remplacée par un condensateur et deux commutateurs analogiques. Le signal de sortie est très proche de la version analogique, mais avec de petites marches d'escalier causées par l'échantillonnage.

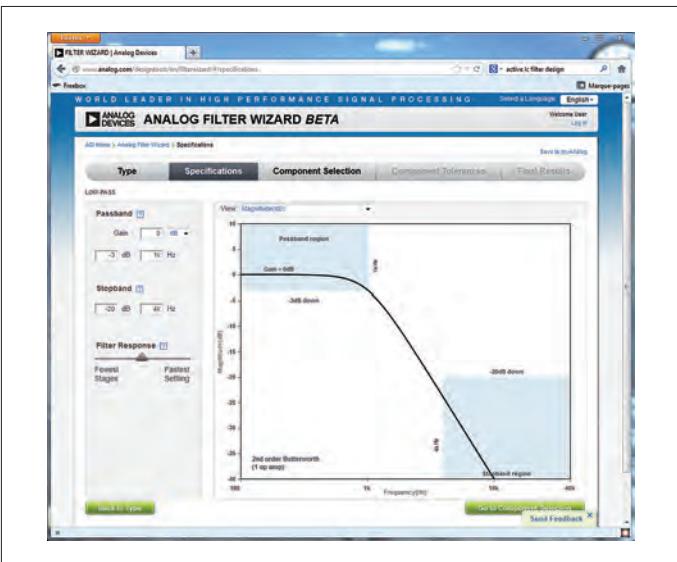


Figure 3. L'outil *Filter Wizard* d'*Analog Devices* permet de calculer facilement les filtres analogiques classiques.

d'un filtre à capacités commutées. Comparez donc ce schéma à celui du filtre RC (fig. 1), vous verrez que la résistance série a été remplacée par un condensateur relié à la masse et deux interrupteurs. Regardez la simulation associée, la forme d'onde filtrée ressemble beaucoup à celle du filtre RC, non ? Pour comprendre comment ça marche, revenons un peu à l'équation (3) :

$$dQ = (U_{\text{entrée}} - U_{\text{sortie}}) \times dt / R$$

Cette équation nous dit qu'avec le filtre RC, à chaque petit inter-

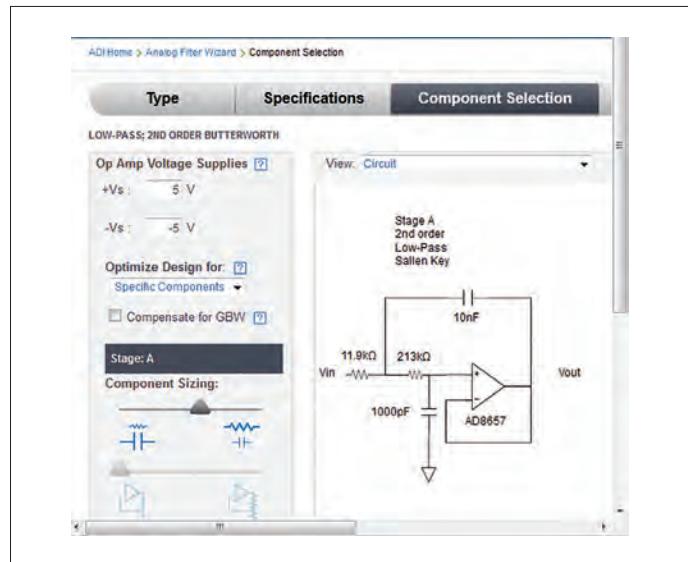


Figure 4. Le schéma produit par *Filter Wizard* est un filtre Sallen-Key.

valle de temps dt , la résistance transfère une petite charge dQ de l'entrée vers la sortie, c'est à dire vers le condensateur C . Pour y voir clair, une autre façon de faire serait de charger un second petit condensateur C' avec la tension d'entrée, puis de le décharger dans le condensateur C ! Si on calcule la valeur de C' pour transférer une charge égale à dQ et qu'on commute ce condensateur toutes les dt secondes, alors on devrait obtenir le comportement d'un filtre RC. Et c'est exactement comme ça que fonctionne un filtre à capacités commutées !

Passons quelques minutes encore sur nos petites formules : à

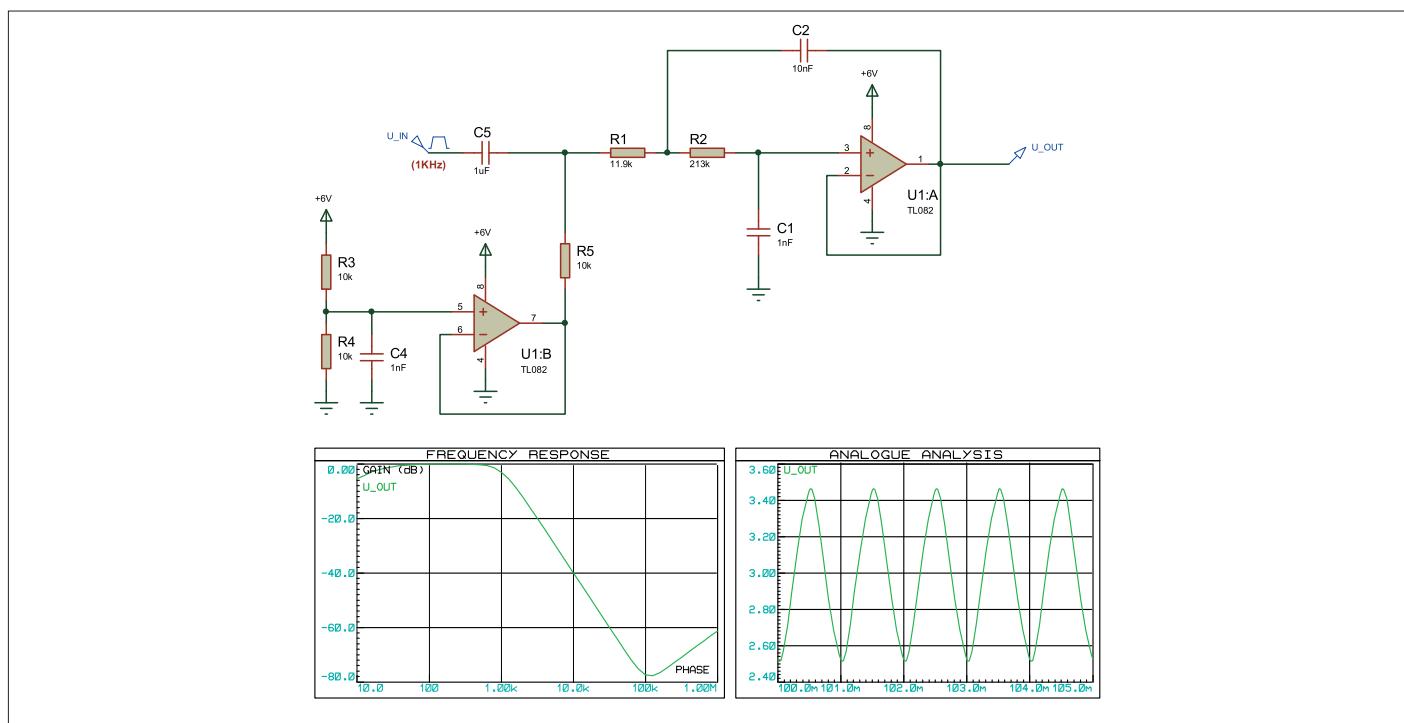


Figure 5. Ce schéma reproduit le filtre passe-bas 1 kHz analogique. La simulation la plus à droite montre le signal de sortie en présence d'une onde carrée de 1 kHz appliquée à l'entrée.

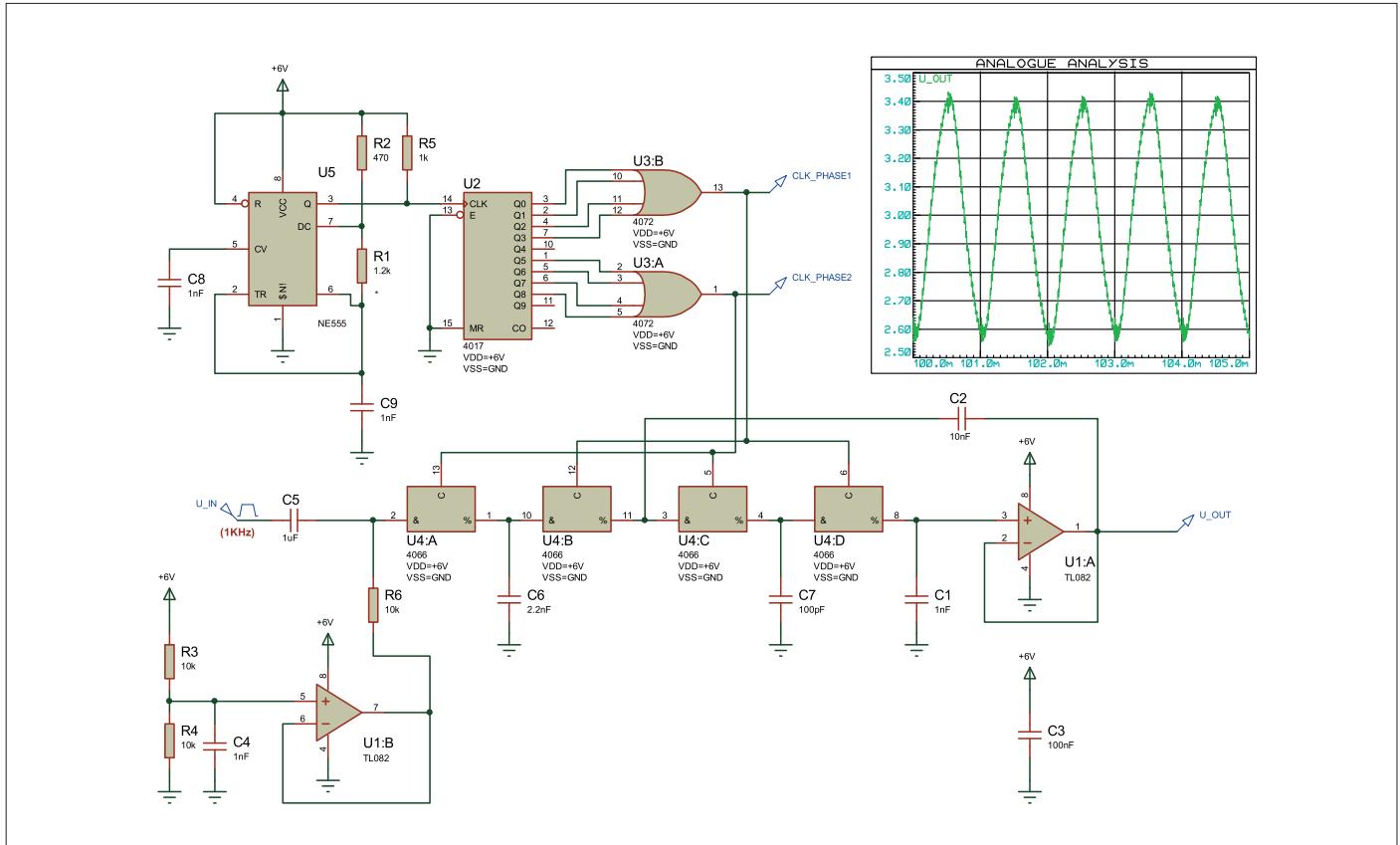


Figure 6. La version à capacités commutées est obtenue à partir de la figure 5 en remplaçant chacune des deux résistances par un condensateur et une paire de commutateurs. La partie supérieure est le générateur d'horloge. La simulation montre un signal très proche de celui de la version analogique.

chaque cycle de commutation, le condensateur C' sera chargé avec une charge $Q = C' \times U_{\text{entrée}}$. En supposant que le condensateur de sortie C est beaucoup plus grand que C' , alors lorsque C' sera commuté sur la sortie sa charge sera quasi instantanément réduite à $Q = C' \times U_{\text{sortie}}$. Par conséquent, à chaque cycle de commutation, la charge transférée est :

$$dQ = C' \times U_{\text{entrée}} - C' \times U_{\text{sortie}} = (U_{\text{entrée}} - U_{\text{sortie}}) \times C' \quad (7)$$

Si vous comparez cette formule à l'équation 3, vous verrez facilement que la cellule de filtre à capacités commutées est équivalente à un filtre RC, avec la correspondance suivante :

$$C' = dt/R, \text{ c'est à dire } C' = 1 / (F_{\text{commutation}} \times R) \quad (8)$$

Et voilà pourquoi un filtre à capacités commutées permet de faire facilement varier la fréquence de coupure : il n'est pas nécessaire de modifier la valeur des composants, mais simplement de changer la *fréquence de commutation des interrupteurs*. Ce qui est bien plus facile à faire dans un circuit intégré numérique !

Un filtre du 2^e ordre

Pour vous prouver que cela marche, je vous propose d'essayer de concevoir un filtre à capacités commutées un peu plus complexe, de le simuler... puis de le construire vraiment.

Prenons l'exemple d'un filtre passe-bas de 1 kHz du 2^e ordre. Commençons par concevoir ce filtre sous la forme d'un filtre RC

classique. Le plus simple pour cela est d'utiliser l'un des outils gratuits disponibles sur la toile, comme celui proposé par *Analog Devices* (**fig. 3**). Il suffit d'entrer les spécifications désirées et sélectionner la topologie de filtre souhaitée, ici un filtre dit *Sallen Key* d'ordre 2. Vous obtiendrez vite le schéma reproduit sur la **fig. 4**. Bien sûr, ce schéma propose d'utiliser un amplificateur opérationnel d'*Analog Devices*, ici un AD8657. Regardez bien le schéma : comme le filtre est du 2^e ordre, sa fréquence de coupure est définie par deux condensateurs (1 nF et 10 nF), et par deux résistances (11,9 kΩ et 213 kΩ).

J'ai simulé pour vous ce schéma, juste pour confirmer qu'il fonctionne comme prévu (**fig. 5**). Comme je n'avais aucun AD8657 sous la main, je l'ai remplacé par un antique TL082, le principe reste le même. Pour pouvoir d'utiliser une alimentation positive, la seconde moitié de ce TL082 est utilisée pour créer une masse virtuelle à la moitié de la tension d'alimentation et pour centrer le signal d'entrée grâce à C_5 et R_5 . Comme le montre la simulation, un signal carré de 1 kHz est transformé en une quasi-sinusoidale, ce qui est normal puisque le filtre atténue fortement les harmoniques. La simulation de la réponse en fréquence est rassurante puisqu'elle confirme qu'on a bien un filtre passe-bas de 1 kHz.

Passons aux condensateurs !

Comment obtenir un filtre à capacités commutées à partir de ce schéma ? On a vu dans l'exemple précédent qu'il suffit de remplacer chacune des deux résistances par un petit condensateur.

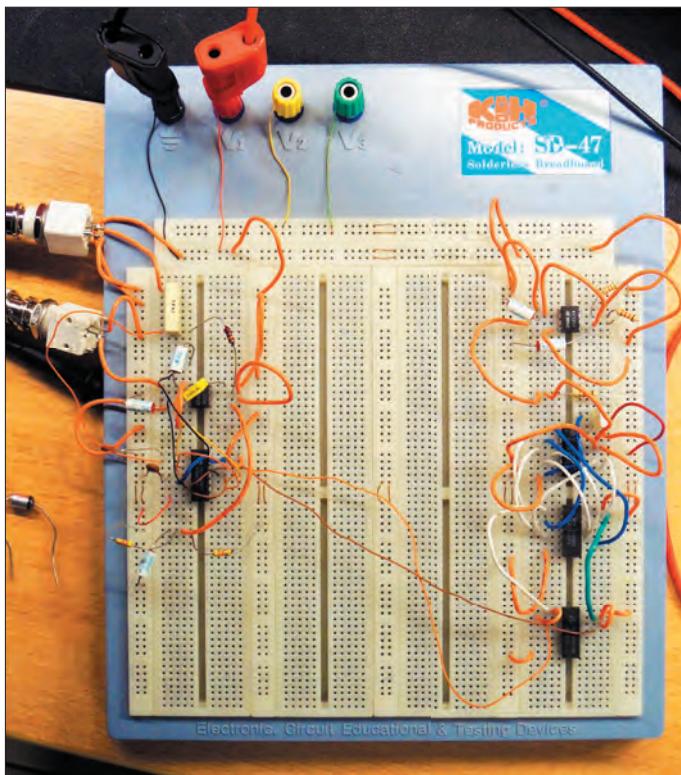


Figure 7. Ma maquette expérimentale. Pas très jolie, mais fonctionnelle. L'horloge est en haut, le filtre en bas.

au rythme de l'horloge. Avec deux opérateurs logiques NOR à quatre entrées (CD4072) on obtient les deux phases d'horloge : la sortie de la première est au niveau logique haut pendant 4/10 du temps, puis les deux lignes d'horloge sont basses pendant 1/10 du temps, puis c'est le tour de la deuxième ligne d'horloge, assurant ainsi l'absence de chevauchement.

On pourrait bien sûr obtenir un résultat similaire avec un peu de code et les sorties à modulation de largeur d'impulsion d'un microcontrôleur. J'ai préféré ressortir de vieilles puces CD40XX du fond de mes tiroirs... La **fig. 6** montre la simulation du circuit complet. Le signal de sortie, avec un carré de 1 kHz à l'entrée, ressemble bien à celui de la version analogique.

Est-ce que ça marche ?

Un simulateur est un superbe outil, mais cela ne nous dispense pas de vérifier que le circuit fonctionne réellement. Comme les fréquences utilisées restent assez basses, je me suis permis d'utiliser une simple carte de prototypage. Le résultat (**fig. 7**) est, euh, très éloigné d'un circuit analogique bien propre, mais je n'avais qu'une demi-heure devant moi... J'ai raccordé ensuite les deux signaux d'horloge à un oscilloscope pour vérifier l'oscillation et les deux phases de l'horloge alternées sans chevauchement (**fig. 8**). Ensuite, j'ai injecté un signal carré de 1 kHz à l'entrée. Regardez le signal de sortie (**fig. 9**) : il est très proche de la simulation, non ?

L'idée des capacités commutées permet de régler facilement la question de la fréquence de coupure du filtre.

Le principe a été décrit par James Clerck Maxwell en... 1873 !

sateur vers la masse et deux commutateurs analogiques. La **fig. 6** montre le résultat. Regardez d'abord la partie basse : c'est le schéma du filtre analogique dont les résistances de 11,9 kΩ et 213 kΩ sont remplacées chacune par un petit condensateur et une paire de commutateurs analogiques, ici des CD4066. Il faut ensuite choisir une fréquence de commutation suffisamment élevée par rapport aux fréquences des signaux, ici j'ai choisi 45 kHz (j'y reviendrai). Enfin il suffit de calculer les valeurs des deux condensateurs en utilisant l'équation 6, qui donne des valeurs de 2,2 nF et 100 pF.

La partie supérieure du schéma est le circuit d'horloge à 45 kHz. Celui-ci paraît bien complexe... C'est parce que tout circuit à capacités commutées requiert deux signaux d'horloge déphasés de 180°, chacun pilotant l'un des commutateurs de chaque paire. Ces deux signaux ne doivent jamais être actifs simultanément, ne serait-ce qu'une nanoseconde. Sinon les lignes d'entrée et de sortie sont court-circuitées, et le filtre ne fonctionne plus du tout. Concevoir deux horloges qui ne se chevauchent pas est moins trivial qu'on le pense. Comme on se contente ici d'une horloge *lente*, 45 kHz, j'ai utilisé une méthode éprouvée : l'antique NE555 délivre une horloge de $10 \times 45 \text{ kHz} = 450 \text{ kHz}$, envoyée à un compteur binaire CD4017. Les dix sorties d'un tel circuit sont activées l'une après l'autre

Enfin, comme j'ai la chance d'avoir dans mon labo un analyseur de spectre HP3585A doté d'un générateur de poursuite, j'en ai profité pour mesurer la réponse en fréquence de ce filtre, en versions capacités commutées et en RC classique (**fig. 10**). Très proche, non ?

Des inconvénients ?

Vous retiendrez qu'un filtre à capacités commutées n'est pas plus difficile à concevoir que ça. Évidemment, tout dépendra des contraintes de votre projet. Revenons une peu sur la sélection de la fréquence d'horloge. Pourquoi ai-je pris 45 kHz ? En fait, le filtre à capacités commutées est un circuit qui découpe (ou échantillonne) le signal analogique, comme le fait un convertisseur analogique/numérique (CAN) ou son inverse le convertisseur numérique/analogique (CNA). À ce sujet, et comme pour tout autre circuit de ce type, Monsieur Nyquist nous dit que l'horloge du découpage du signal analogique doit impérativement être au double de la fréquence la plus élevée du signal traité. Attention, ce n'est pas de la fréquence de coupure du filtre qu'on parle ici, mais de la fréquence la plus élevée susceptible d'être présente dans le signal d'entrée. À défaut de cette précaution, nous aurions de gros soucis avec le *repliement du spectre* qui perturberait gravement la réponse du filtre.

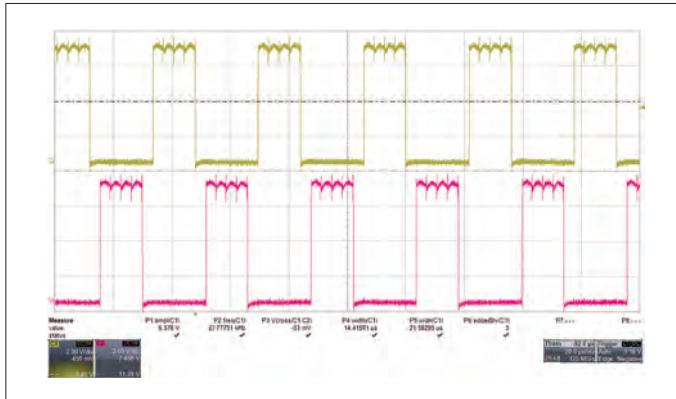


Figure 8. Comme prévu, les deux phases de l'horloge ne se chevauchent pas. Les petits sauts de tension lors des transitions sont des conséquences directes du câblage en l'air...

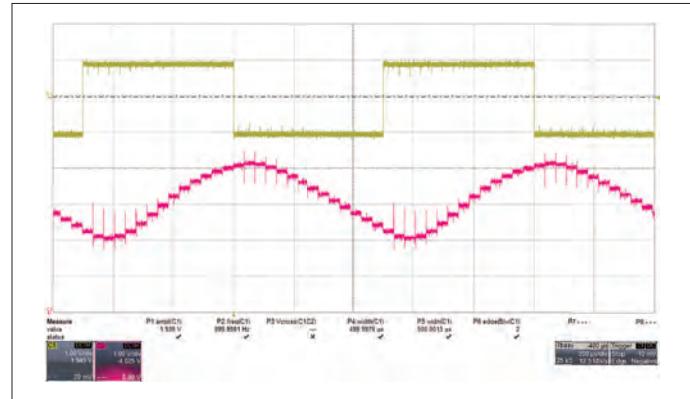


Figure 9. Voici le signal de sortie (rouge) lorsqu'un carré est appliqué à l'entrée (vert). On distingue bien les petites marches d'escalier, qui devront être supprimées par un filtre passe-bas analogique.

Tout comme avec un convertisseur analogique-numérique, la sortie d'un filtre à capacités commutées est un signal échantillonné. Sur l'oscilloscopogramme mesuré sur la maquette (fig. 3), on voit des petites marches d'escalier causées par la commutation des condensateurs. Pour éliminer ces petits sauts après un tel filtre, on ajoute en général un autre filtre analogique, appelé *filtre de reconstruction*. Exactement comme avec un convertisseur A/N ou N/A. Pour que ce filtre soit facile à fabriquer et ne perturbe pas trop le signal utile, vous choisissez en pratique une fréquence d'horloge *largement* supérieure au double de la fréquence maximale du signal : un facteur 10 n'a rien d'exceptionnel...

Pour conclure

Notre petite excursion dans le monde des filtres à capacités commutées touche à sa fin. Il est rare d'avoir à concevoir son propre filtre à partir de composants discrets comme je l'ai fait dans cet article. Les fabricants de circuits intégrés s'en sont donné à cœur joie en matière de filtres intégrés : voyez p. ex. chez *Maxim*, *Analog Devices* ou *Cypress*. Pour comprendre comment ils fonctionnent, pour les utiliser efficacement et à bon escient, rien de tel que mettre soi-même la main à la patte. Je vous encourage chaudement à expérimenter vous-même. Commencez peut-être par reproduire les exemples donnés dans cet article avec un simulateur comme LT Spice, puis sortez le fer à souder si vous en avez envie : c'est de loin la meilleure méthode pour aller au fond des choses et en revenir mieux

armé ! Jetez un œil aux articles cités en référence, ils vous permettront d'aller encore un peu plus loin. ↗

200005-01

*Cet article a été publié en anglais
dans la revue Circuit Cellar (n° 277, août 2013)*



Figure 10. Voici la réponse en fréquence du filtre à capacités commutées comparée à celle de la version analogique.

Liens et littérature

- [1] filtres à capacités commutées Elektor, janvier 1981 (téléchargement gratuit): www.elektormagazine.fr
- [2] The Story of James Clark Maxwell and Switched Capacitor Filters Philip Golden, Intersil capacitor-filters
- [3] Switched capacitors | Take the Mystery Out of the Switched-Capacitor Filter: The System Designer's Filter Compendium Richard Markell, Linear, Application note AN40: <http://cds.linear.com/docs/en/application-note/an40f.pdf>
- [4] Simulateur de signal mixte Proteus VSM : www.labcenter.co.uk
- [5] Oscilloscope numérique Teledyne Lecroy WaveRunner 610zi : <http://teledynelecroy.com/oscilloscope/oscilloscopemodel.aspx?modelid=4781>
- [6] Analyseur de spectre Hewlett Packard HP3585A (antiquité)

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

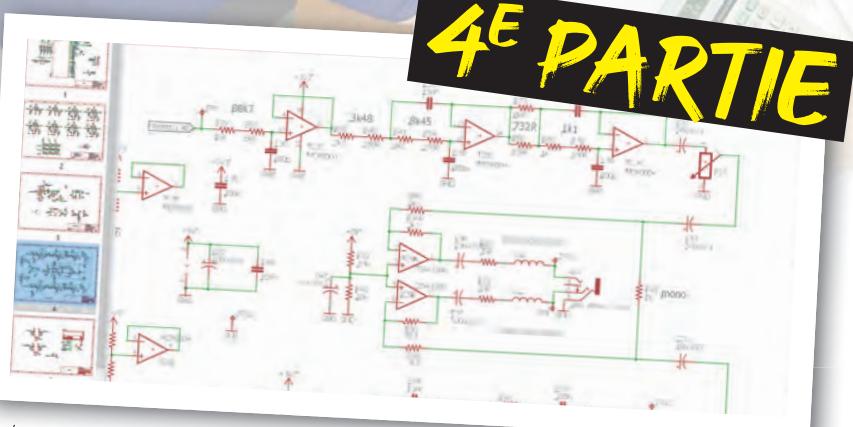
La mise au point de notre nouveau produit progresse lentement. Comme nous disposons d'une démonstration de faisabilité ou *preuve de concept* (PoC), le moment est venu de créer les documents de conception.

DOCUMENTEZ VOTRE TRAVAIL

Le PoC proposé dans l'épisode précédent se composait d'une ou plusieurs cartes d'assemblage et/ou de prototypage, de circuits imprimés soudés et câblés à des kits ou cartes d'évaluation et de développement. Ce qui importe, c'est que ça marche, même si c'est encore un sac de noeuds. Cependant, vous ne pouvez pas continuer comme ça. Il est primordial d'une part de consolider un peu tout ça et d'autre part de documenter votre travail.



Il n'est jamais trop tôt pour commencer à documenter votre travail. Y passer un peu de temps maintenant vous en fera gagner beaucoup plus tard.



Il n'est jamais trop tôt pour commencer à documenter votre travail.
Y passer un peu de temps maintenant vous en fera gagner beaucoup plus tard.

LA POC DOIT RESTER EN VIE

Dessiner le schéma d'une PoC complexe est un exercice périlleux, surtout si elle contient des composants qui n'existent pas dans les bibliothèques du logiciel de capture de schéma utilisé. Votre PoC était probablement basée sur schéma de prototype qui a peut-être évolué. Dans ce cas, les modifications doivent être soigneusement reportées sur le schéma sans erreur. Pour cette tâche humaine, les lois de Murphy s'appliquent. Le comptage des broches sur un circuit intégré à pas fin est une affaire délicate. Et puis une faute de frappe est si vitre faite... comme disait le laveur de carreaux. Si pour y voir clair, vous défaites des connexions, assurez-vous de tout remettre en place ensuite. Tant que vous n'en avez pas d'autre qui soit opérationnel, c'est à cette PoC-ci qui fonctionne que tient votre projet.

FAITES-VOUS CONFIANCE À VOS COMPOSANTS ?

Vous devez pouvoir vous fier aux composants de votre bibliothèque pour la capture de schémas. Vous n'y trouverez pas toujours les composants dont vous avez besoin ! 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 (oui, ça existe) ? Dans tous les cas, vérifiez soigneusement chaque composant avant de vous y fier. Idem pour votre bibliothèque de composants *standard* (qui elle aussi peut contenir des erreurs). Nous ne parlons pas (encore) des empreintes pour la conception de circuits imprimés (PCB) - nous y reviendrons plus tard.



Les composants que vous comptez utiliser sont-ils encore disponibles aujourd’hui ? Et demain ? Les avez-vous sous la main ? Demandez des échantillons et vérifiez les empreintes et les tailles

PAS DE SUBSTITUTS

Une erreur fréquente lors de la capture, quand un composant désiré ne figure pas dans votre bibliothèque, consiste à introduire dans le schéma un composant de substitution, similaire par sa fonction ou même son brochage. Cela accélérera peut-être la phase de dessin, mais c'est courir à l'erreur. La plus fréquente est l'oubli de corriger la valeur de ce composant de substitution. Pour les composants complexes comme les µC, le risque d'oubli est réduit, mais grand pour les connecteurs par

exemple. À l'exportation de la nomenclature (ou liste de composants), des problèmes plus subtils peuvent surgir : les substituts ont souvent des attributs qui ne correspondent pas au composant original. Ah, si ça correspond quand même ? Alors le temps gagné par l'utilisation de remplaçants pendant le dessin, vous le perdrez en fastidieuses vérifications. (Nous nous occuperons des empreintes plus tard).

COMPLÈTE ET CORRECTE

À ce stade de la conception, le schéma est le document principal. Il doit donc contenir autant d'informations que possible et être correct. Un bon outil de capture de schéma permet à l'utilisateur de spécifier toutes sortes d'attributs pour les composants et les connexions. Utilisez ces fonctions pour que l'exportation de la BoM produise un document

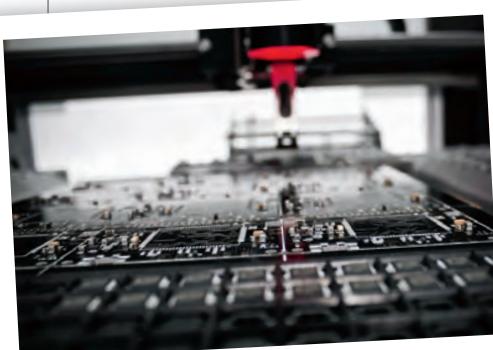
correct et complet. Il doit en être de même pour la *netlist* exportée. Commentez s'il y a lieu. Si p. ex. un transistor a besoin d'être refroidi, ou si une connexion voit circuler un courant de forte intensité, ajoutez l'information dans une note. Des remarques sur le placement des composants sont également utiles.

CONCEPTION POUR LA PRODUCTION

Outre les coûts des composants, un autre facteur important se cache dans la fabrication, à savoir les coûts de production. Ce que vous avez imaginé peut-il être fabriqué ? À supposer que ce soit le cas (à un coût raisonnable), comment le

produit sera-t-il testé ? Comment seront programmés les composants programmables ? Quid

des options matérielles ? Des points de test ou, si possible, un connecteur de test spécial peuvent être requis. Une procédure de test est également souhaitable. Ces tests doivent être à la fois simples et complets. Peut-être que le circuit doit être modifié pour cela ? Le logiciel pourrait-il s'en charger ? Dans ce cas, la PoC sera-t-elle compromise ? Marquez les valeurs et les procédures de test dans le schéma. N'oubliez pas non plus les options d'implantation et d'assemblage (*board stuffing*).



Puisque votre conception devra probablement passer par une machine de ce genre, assurez-vous qu'elles sont compatibles.

CERTIFICATION

Il se peut que votre produit ne soit commercialisable qu'avec certification. Il n'y a pas que les classiques CE et FCC. Pour espérer obtenir l'homologation, il faut s'y préparer dès les premières de la conception. Ainsi pour la PoC, on ne se

préoccupe généralement pas de mesures de CEM comme le blindage et le filtrage, mais il faut se demander si elles pourront apparaître ultérieurement sans compromettre la PoC...

VERSIONS

Personne n'obtient un schéma parfait dès le départ. Les choses évolueront, tandis que vos connaissances et votre expérience augmenteront. Plusieurs itérations seront nécessaires. Gardez une trace bien documentée de ces révisions pour éviter de refaire la même erreur. Surtout si un projet se décline en plusieurs versions. Utilisez des identificateurs uniques et clairs pour les versions et les itérations et soyez cohérent.

LA SUITE ?

Les résultats de l'étape de conception décrite ici sont d'abord un document agréable à consulter pour référence future, puis une nomenclature et une *net list* à la fois complètes et correctes. Avec ces documents, vous pouvez vous lancer dans la conception d'un circuit imprimé (PCB). ▶

190369-B-02

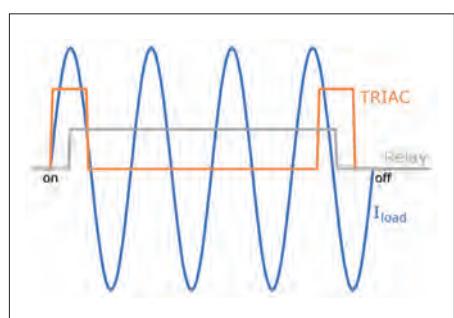
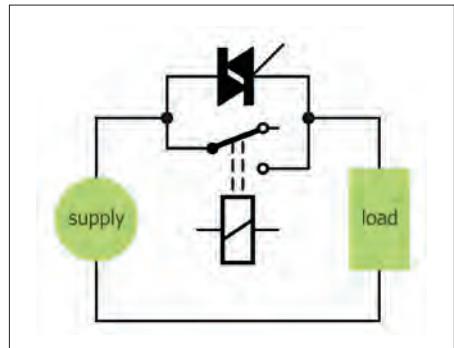
Bonnes pratiques :

Le relais hybride

Rapides, petits et bon marché, les triacs et les thyristors (ou SCR pour *silicon controlled rectifiers*) sont excellents pour commuter des charges en courant alternatif. Ils présentent hélas une résistance relativement élevée au démarrage, ce qui entraîne des pertes de puissance et impose un refroidissement dès que la charge consomme plus de quelques ampères. Les relais se prêtent bien eux aussi à la commutation de charges CA. Leur résistance à l'enclenchement est faible, ils sont faciles à commander et leur signal de commande reste isolé de la charge. Et si on utilisait un relais en parallèle avec un triac pour qu'ils commutent une charge ensemble?

Mise sous tension

On commence par enclencher le triac pour appliquer sa tension d'alimentation à la charge. Le triac est rapide, on peut donc l'allumer et l'éteindre lors du passage par zéro de la tension d'alimentation alternative et réduire ainsi les interférences électromagnétiques (EMI). Le triac peut être utilisé en mode de découpage de phase, lequel permet le démarrage en douceur d'une charge. Une fois la charge sous tension, c'est le moment d'activer aussi le relais. La résistance du contact de l'interrupteur mécanique fermé est beaucoup plus faible que celle du triac quand il conduit, de sorte que la plus grande partie du courant finit par circuler par l'interrupteur fermé. Le triac devenu inutile peut donc être désactivé.



Mise hors tension

La mise hors tension de la charge commence par la remise sous tension du triac, suivie par la mise hors tension du relais. Comme le triac est conducteur à ce moment, la tension aux bornes de l'interrupteur mécanique reste faible. Ouvrir l'interrupteur ne présente donc aucun danger de formation d'étincelles ni d'arcs électriques de rupture. Là encore, la production d'interférences reste faible, les contacts de l'interrupteur ne sont pas malmenés. Une fois le relais ouvert, le déclenchement du triac est interrompu et la charge mise hors tension dès que le triac cesse de conduire.

Avantages

Construit et utilisé ainsi, un relais hybride présente de nombreux avantages : faible dissipation de puissance du fait de la faible résistance du relais ; faible encombrement et coûts réduits par l'absence de (gros) radiateur. Les interfaces restent faibles car en présence du triac il n'y a pas de rebonds de contact. Le fonctionnement est rapide et sans étincelle, la commutation plus sûre et plus robuste.

Le principal inconvénient d'un relais hybride est le nombre de composants requis. Sa commande est plus complexe et nécessite généralement un microcontrôleur.

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

FACTORY OF THE FUTURE

powered by elektor

electronica Fast Forward 2020

plateforme pour start-ups propulsée par Elektor

The Elektor Team

Votre start-up a de grandes ambitions en électronique. Attirez sur votre jeune entreprise l'attention de clients et d'investisseurs potentiels ! La compétition **electronica fast forward (e-ffwd), la plateforme pour start-ups propulsée par Elektor**, vous offre l'occasion unique de vous présenter au monde entier. Vous pouvez gagner un plan marketing d'Elektor d'une valeur de 75.000 € et bénéficier sur le salon electronica 2020 d'une reconnaissance internationale ! Inscrivez-vous sur www.elektormagazine.com/fastforward au concours e-ffwd 2020

Lancer et gérer avec succès une start-up axée sur l'électronique n'est pas une tâche facile. Les dirigeants intelligents savent qu'il ne suffit pas d'un produit innovateur, ni d'une nouvelle technologie ou d'un super code. Ils savent qu'il faut sans relâche cultiver ses relations avec les principaux dirigeants et experts du secteur. Ils reconnaissent l'intérêt de communiquer les caractéristiques, les fonctions et les avantages de leurs produits à un prix abordable aux publics appropriés. Et ils comprennent l'importance de mettre en œuvre des campagnes de marketing ciblées qui élargissent leur terrain d'action et stimulent les ventes.

En 2020, Elektor et le salon *Messe München* s'associent une nouvelle fois pour donner aux innovateurs l'occasion de faire progresser leurs entreprises et de promouvoir leurs produits auprès d'un public international. La compétition **electronica fast forward, la plateforme pour start-ups propulsée par Elektor**, vous offre la chance de gagner un plan marketing Elektor d'envergure mondiale, et vous permet de bénéficier d'une exposition internationale sur le salon *electronica 2020* (<https://electronica.de>) – le plus grand salon mondial de l'électronique – à Munich, en Allemagne (du 10 au 13 novembre 2020). Les start-ups les plus prometteuses pourront promouvoir leurs solutions auprès de clients potentiels, d'experts du secteur et d'investisseurs éventuels. Êtes-vous prêt à concourir ?

- **Date limite d'inscription :** 1er juillet 2020 www.elektormagazine.com/fastforward
- **Finale :** electronica 2020 <https://electronica.de> à Munich, Allemagne (du 10 au 13 novembre 2020)
- **Prix et avantages :**
 - Premier prix = plan marketing Elektor d'une valeur de 75.000 €, plus un stand sur le prochain salon electronica 2022.
 - Deuxième prix = plan marketing Elektor d'une valeur de 50.000 €
 - Troisième prix = plan marketing Elektor d'une valeur de 25.000 €



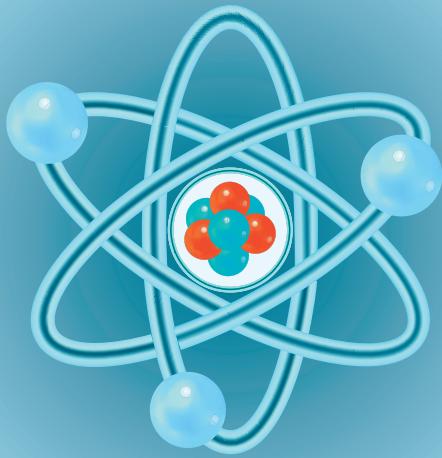
- Bénéficiez d'un espace de stand (5 m²) exclusif et à prix réduit (seulement 1.905 € !) sur le salon electronica 2020 (mobilier de base, éclairage, alimentation, nettoyage quotidien et entretien, etc. inclus)
- Votre start-up apparaît dans le catalogue du salon (imprimé et en ligne).
- Présentez vos produits innovateurs et entrez sur place dans le réseau des investisseurs, partenaires et clients potentiels.
- Défrichez de nouvelles pistes et bénéficiez d'une grande visibilité internationale avant, pendant et après le salon electronica 2020.
- Appréciez mieux les forces et les faiblesses de votre entreprise.
- Et bien davantage !

Votre start-up s'appuie-t-elle sur une solution d'IA supérieure,



d'une application d'IdO qui change la donne ou d'un projet de robotique de pointe ? Peut-être votre équipe a-t-elle mis au point récemment un nouveau système intégré ou un outil de mesure utile ? Avez-vous inventé une plateforme de prototypage rapide super abordable ou un gadget électronique pratique ? Inscrivez-vous à **electronica fast forward, la plateforme pour start-ups propulsée par Elektor** (www.elektormagazine.com/fastforward) et venez le faire savoir au monde entier !

Bonne chance.



démarrer en électronique (2)

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

Eric Bogers (Elektor Pays-Bas)

Dans certains passe-temps, il n'y a jamais rien à calculer. Pour la peinture à l'huile par exemple, il faut des tubes et des pinceaux mais personne, ni avant ni après le barbouillage, ne mesure jamais le degré de verdure de l'herbe. L'électronicien, pour réussir ses tableaux, ne peut se passer de calcul (précis), sinon les fusibles fondent et les composants se décomposent

Conductivité

Nous avons parlé de tension et de courant dans le premier article ; voyons maintenant ce qu'on peut en faire.

Nous savons tous que certains matériaux, notamment les métaux, conduisent mieux que d'autres le courant électrique ; certains matériaux résistent plus au courant qu'ils ne le conduisent. Certains ne le conduisent pas du tout, ce sont des isolateurs. Le verre par exemple, ou l'eau distillée ou purifiée, sont aussi de bons isolants.

Une grandeur a été définie en physique qui spécifie cette propriété matérielle : la *conductivité*.

Conductivité électrique

Il existe une belle relation linéaire entre la tension qui règne aux bornes d'un conducteur et le courant qui le traverse, exprimée sous la forme suivante :

$$\kappa = I / U$$

En d'autres termes, le quotient du courant et de la tension est la conductivité - plus le courant circule sous une tension donnée, plus la conductivité κ (lettre grecque kappa minuscule) est grande. L'unité de conductivité est le siemens (S).

Unité de résistance

Comme en pratique la valeur de tension est généralement numériquement supérieure à celle du courant, la conductivité sera presque toujours une valeur très inférieure à l'unité. Les électroniciens, qui n'aiment pas calculer avec de si petites valeurs, préfèrent faire leurs calculs avec l'inverse de la conductivité : la *résistance*. Ça tombe bien, car mieux un conducteur conduit le courant, plus sa résistance est faible. La résistance R est exprimée par l'unité ohm (symbole : Ω , la lettre grecque oméga majuscule). Ce qui donne :

$$R = U / I$$

C'est ça la fameuse loi d'Ohm, celle que nul ne peut ignorer s'il veut faire de l'électronique.

Intermezzo

Dans la pratique quotidienne de l'électronique, l'ohm est une unité assez petite. Les quantités d'ohms rencontrées sont généralement beaucoup plus grandes, par exemple 8200 Ω . Pour le farad, en revanche, l'unité de capacité utilisée pour les condensateurs et à laquelle nous reviendrons, c'est l'inverse. Pour simplifier la manipulation de valeurs les unes très grandes

Multiples		
nom	symbole	valeur
tera	T	10^{12}
giga	G	10^9
mega	M	10^6
kilo	k	10^3
hecto	h	10^2
deca	d	10^1
Fractions		
nom	symbole	valeur
deci	d	10^{-1}
centi	c	10^{-2}
milli	m	10^{-3}
micro	μ	10^{-6}
nano	n	10^{-9}
pico	p	10^{-12}
femto	f	10^{-15}

(p. ex. une résistance de huit mille deux cents ohms) les autres très petites (p. ex. un condensateur d'un dix-huit millionième de farad), nous utilisons les préfixes décimaux du tableau.

On parle donc d'une résistance de $8,2 \text{ k}\Omega$ (kiloohms) et d'un condensateur de $18 \mu\text{F}$ (microfarads). Certains, parce qu'ils ne trouvent pas sur leur clavier lettres Ω et μ , ont recours à des subterfuges, ce qui est souvent la cause de malentendus :

$$8200 \Omega = 8,2 \text{ k}\Omega = 8k2 = 8 \text{ k}\Omega \\ 0,000000056 \text{ F} = 5,6 \text{ nF} = 5n6 \text{ ou } 5nF6$$

Un circuit électrique

Si le courant électrique circule, c'est donc qu'il vient de quelque part et qu'il va quelque part. Pour simplifier, disons que le courant vient (de la borne *plus*) d'une batterie et circule vers la borne *moins* de cette même batterie, en un circuit fermé. Dans un circuit «ouvert», aucun courant ne peut circuler ! Derrière ce mouvement conventionnel du courant électrique du *plus* vers le *moins* se cache le mouvement réel des électrons

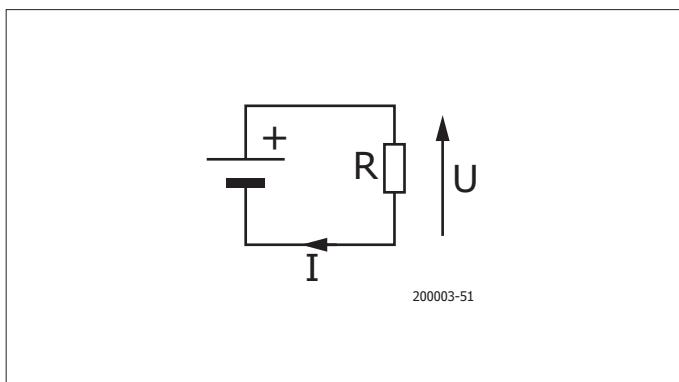


Figure 1 : Le circuit (fermé) dans sa plus simple expression.

qui se déplacent dans la direction opposée ; le sens conventionnel est un héritage du passé, lorsque les électrons étaient encore inconnus.

Examinons le circuit (**fig. 1**) le plus élémentaire que l'on puisse imaginer : il n'est constitué que d'une batterie (à gauche) et d'une résistance (à droite). Les fils du pôle *positif* de la batterie à une patte de la résistance et de l'autre patte de cette résistance au pôle *négatif* de la batterie sont supposés idéaux, c'est-à-dire dépourvus de résistance. Ce n'est pas vrai en réalité mais, dans les circuits simples, cette hypothèse réductrice ne pose pas de problème. Aux bornes de la résistance règne donc la tension de batterie U , comme indiqué par la flèche de tension ou de potentiel (du *moins* au *plus*). Dans ce circuit circule un courant I , également indiqué par une flèche de courant, un courant qui, selon la représentation conventionnelle, va du *plus* au *moins*).

Le moment est venu de faire un peu d'arithmétique. Le potentiel de la batterie de 9 V (6LR22), la résistance est de 2200Ω ou $2,2 \text{ k}\Omega$. Quelle est l'intensité du courant qui circule dans le circuit ?

$$R = U / I \rightarrow I = U / R = 9 \text{ V} / 2200 \Omega = 0,00409 \text{ A} = 4,09 \text{ mA}$$

Un courant d'environ 4 mA circule dans cette résistance, laquelle, en s'opposant au passage du courant, chauffe – c'est le principe du radiateur électrique. Dans l'épisode précédent, nous avions calculé la puissance convertie (ou dissipée) en chaleur par la résistance :

$$P = U \cdot I = U^2 / R = 812 \text{ V} / 2200 \Omega = 0,0368 \text{ W} = 36,8 \text{ mW}$$

Ce n'est pas beaucoup. Si dans ce modeste « chauffage électrique » nous utilisons une résistance de 220Ω , donc dix fois plus petite, la dissipation sera dix fois plus forte, de l'ordre de 368 mW. Comme on le voit dans les listes de composants des projets décrits dans ce magazine, la puissance maximale des résistances dans les circuits électroniques habituels est de 250 mW, soit $\frac{1}{4}$ de watt. Notre courant de 368 mW aurait donc vite fait de carboniser une telle résistance...

Un dernier exemple : nous allons connecter un projecteur de 1000 W à une prise murale à l'aide d'une rallonge. Quelle intensité de courant ce câble devrait-il supporter sans risque de mettre le feu à la baraque ?



Réponse :

$$P = U \cdot I \rightarrow I = P / U = 1000 \text{ W} / 230 \text{ V} = 4,35 \text{ A}$$

Attention ! Ceci **n'est pas** un conseil de sécurité mais un exemple de calcul.

La prochaine fois, nous poursuivrons avec des circuits plus complexes avec des résistances. ▶

200003-03

@ WWW.ELEKTOR.FR

🛒

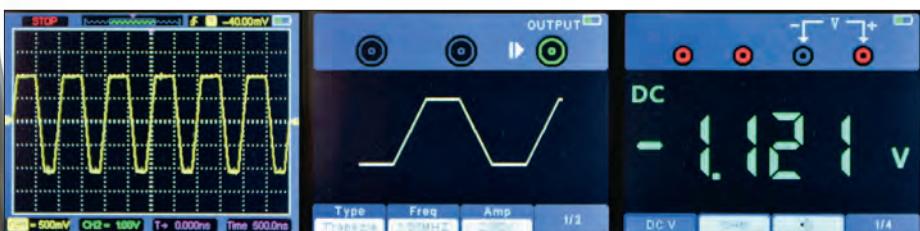
- L'électronique pour les débutants
www.elektor.fr/l-electronique-pour-les-debutants
- L'électronique en pratique
www.elektor.fr/l-electronique-en-pratique
- The Art of Electronics
www.elektor.com/the-art-of-electronics-3rd-edition



BANC D'ESSAI oscillo de poing 3 en 1 JOY-iT DMSO2D72

Harry Baggen (Pays-Bas)

Pour les mesures en extérieur, il faut un instrument portable. Celui-ci sera d'autant plus utile qu'il combinera plusieurs fonctions de mesure. Le JOY-iT DMSO2D72 est à la fois un oscilloscope, un multimètre et un générateur de formes d'ondes arbitraires réunis dans un même boîtier de poing, solide et pratique. Je l'ai testé pour vous.



Côte à côté : l'écran de l'oscilloscope, l'AWG et le multimètre.

Qu'est-ce que c'est ?

Avec autant de fonctions dans le même appareil, on se demande quelle est la principale. C'est sans aucun doute l'oscilloscope. Ici une version à deux voies, avec bande passante de 70 MHz et un taux d'échantillonnage de 250 Méch/s (ou la moitié si l'on utilise les deux voies ensemble). Ce sont de belles spécifications pour un instrument de l'ordre de 200 € (si vous êtes membre d'Elektor). En plus il y a un générateur de fonctions dont les caractéristiques se laissent voir (250 Méch/s) et un multimètre à sélection automatique de calibre avec ses entrées propres. Que vouloir de plus ? Le DMSO2D72 est logé dans un boîtier robuste d'environ 20x10x4 cm, livré avec un bloc d'alimentation et câbles de mesure et sondes. Le tout ressemble à un gros multimètre. Le boîtier est protégé par des pare-chocs en caoutchouc. À l'arrière, un support escamotable permet de le placer à la verticale. Là se trouve

également le compartiment de la batterie, composée de deux éléments Li-ion 18650 standard, donc faciles à remplacer si nécessaire. Sur le côté droit, sous un petit couvercle, une connexion USB-C utilisée pour l'alimentation et pour la communication avec le PC. En haut, trois connecteurs BNC (bien en retrait !) pour les sondes de l'oscillo et la sortie du générateur de fonctions arbitraires (AWG). À l'avant, en haut le grand écran de 7 cm (2,8 pouces) de diagonale, avec en dessous les boutons de commande et en bas les entrées du multimètre. L'instrument est livré dans un étui confortable, avec de nombreux accessoires.

Oscilloscope

Vous tâtonnerez peut-être un peu, mais vous utiliserez rapidement votre DMSO2D72 sans même lire le manuel d'abord. Trois des boutons bleus permettent de changer de fonction, le quatrième donne accès à un menu avec différentes options. Sous l'écran se trouvent quatre touches dont les fonctions apparaissent sur l'écran. La

lecture de l'écran en couleur est agréable. Il fournit beaucoup d'informations sur 320x240 pixels. Il aurait été un peu plus grand, je n'aurais pas protesté, mais l'appareil serait plus cher.

Quatre touches de curseur permettent de régler, entre autres, la base de temps, la sensibilité, le niveau de courant continu et le niveau de déclenchement. Ça demande une familiarisation, car les fonctions des touches changent selon que vous avez appuyé auparavant sur le bouton *Channel* ou le bouton *Time*. Malheureusement, l'écran n'indique pas dans quel mode (*Channel* ou *Time*) l'oscilloscope se trouve.

L'oscilloscope possède presque toutes les caractéristiques d'un oscillo ordinaire, comme le réglage automatique de l'image fixe, l'utilisation de repères de curseur et l'affichage de différentes valeurs de mesure. Les possibilités de déclenchement sont plutôt limitées, mais suffisantes pour la plupart des situations pratiques. L'oscillo dispose même d'une mémoire de données de 6 kéch (1 voie) ou 3 kéch (2 voies), dont le contenu

est affiché en haut de l'écran. Une fois l'échantillonneur gelé, vous pouvez parcourir sa mémoire avec les touches de curseur.

AWG et multimètre

Le générateur de fonctions arbitraires intégré (AWG) offre un certain nombre de fonctions standard et dispose de quatre mémoires pour les formes d'onde que l'utilisateur programme à sa guise, avec un logiciel pour PC fourni. La fréquence maximale n'est pas la même selon la forme d'onde. Le sinus monte à 25 MHz, le carré à 10 MHz mais le triangle plafonne à 1 MHz. La fréquence des fonctions arbitraires peut aller jusqu'à 5 MHz.

Avec les touches de curseur, vous pouvez balayer l'échelle des fréquences, mais si vous appuyez deux fois sur la touche de fonction de fréquence, un clavier apparaît sur l'écran où vous pouvez saisir une fréquence directement. Il est possible d'utiliser l'AWG et l'oscillo en même temps. Après avoir réglé l'AWG et activé sa sortie, vous passez à l'oscillo tandis que l'AWG reste actif. C'est idéal pour tester ; vous appliquez le signal AWG à l'entrée du circuit à tester pour le relever ensuite avec l'oscillo en divers points du circuit ou à sa sortie. La valeur affichée de la tension de sortie réglée est valable si la sortie est chargée par $50\ \Omega$, sinon c'est le double de cette valeur !

Le courant d'alimentation est fourni par deux cellules Li-ion 18650 standard. La présence du multimètre est un bel ajout. Ses caractéristiques correspondent, comme on peut s'y attendre, à celles d'un multimètre moyen, dont la précision n'est pas parfaite (4000 points, précision de base : 0,8%). Pour un outil de poing, ce n'est pas un handicap.

Le calibre automatique fonctionne bien et l'écran indique également quelles douilles utiliser pour quelle mesure. Les calibres mA et A ont des entrées séparées. Attention, sur le calibre 10 A il n'y a pas de fusible ! Le multimètre comporte aussi une fonction ohmmètre, un testeur de diodes et d'un testeur de continuité. Et il mesure même les capacités.

Le logiciel pour PC

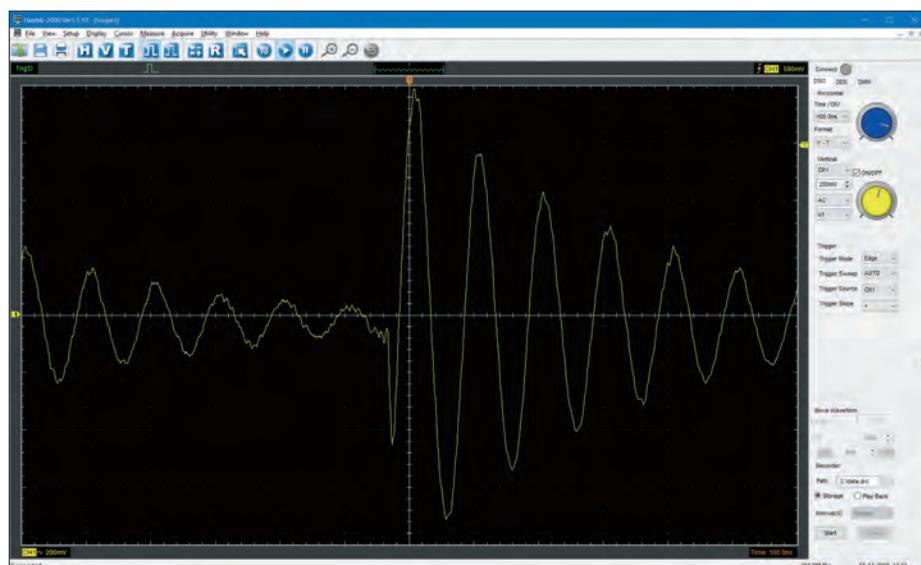
Sur le site de JOY-iT, vous trouverez un logiciel qui vous permettra de commander l'appareil depuis votre ordinateur. Chez moi, l'installation s'est déroulée sans problème, mais après le démarrage, un message d'erreur (*read allset failed*)



Le DMSO2D72 est livré avec de nombreux accessoires.



Le retrait profond des douilles BNC dans le haut du boîtier est pratique.



Les trois fonctions du DMSO2D72 peuvent être utilisées avec le logiciel pour PC.

apparaissent périodiquement. De sorte que de nombreux paramètres étaient incorrects, d'autres ne fonctionnaient pas du tout.

Il semble que ce défaut soit connu : le logiciel s'embrouille dans les paramètres de l'appareil connecté. Pour résoudre le problème, il a suffi d'installer une ancienne version (1.1.10). Comme après ça tout semblait de nouveau fonctionner, j'ai pu réinstaller la version actuelle (1.1.11). Le transfert de données en mode oscilloscope est malheureusement assez lent, mais le programme offre beaucoup de possibilités ; vous pouvez p. ex programmer des formes d'onde pour les envoyer au DMSO2D72.

Bilan

Pour environ 200 €, vous avez un instrument de mesure à trois fonctions, pratique et robuste, pour presque toutes les tâches ordinaires de l'électronique. Les calibres de l'oscillo et du générateur sont assez étendus et une fois que vous savez comment le faire fonctionner, l'appareil est d'un emploi facile. Ce n'est pas l'instrument de mesure le plus précis, mais à ce prix et pour l'usage pour lequel il est conçu, il faut savoir modérer ses attentes. La combinaison des trois fonctions de mesure en fait un appareil pratique, idéal pour quiconque a besoin d'une solution portable à prix très modéré. ▶

(191264-03 VF)

@ WWW.ELEKTOR.FR

→ JOY-iT 3-in-1 Handheld Oscilloscope, Signal Generator and Multimeter (DMSO2D72)
www.elektor.fr/19157

j'assemble un PC pour mon labo

Conseils pour le choix des composants

Thomas Scherer (Allemagne)

Certains petits labos vivent le traumatisme des « petits frères » : ils ne portent que des vêtements déjà usés par leurs aînés. Entre les vieux PC dépassés, et un PC quelconque du commerce, prêt à mouliner, que choisir ? C'est encore le nouveau PC sur mesure qui tirera le mieux son épingle du jeu. Voici ce qu'il faut avoir si vous vous apprêtez à assembler le vôtre.



Figure 1 : Les entrailles du PC de labo : carte-mère, RAM, SSD, CPU et pâte thermique.

L'avantage des vieilles bécane, c'est que si vous en flinguez une, ce n'est pas un drame. Un autre avantage, ce sont leurs interfaces antiques, d'avant l'USB ou même l'Ethernet, peut-être obsolètes, mais toujours bien utiles pour les projets anciens et pour communiquer avec certains appareils de mesure antédiluvien toujours en service. Cela dit, les inconvénients des vieux PC sont patent.

D'abord, leur manque de fiabilité. Certes, il y a des PC qui tiennent dix ans, mais ce n'est pas pour rien que l'amortissement typique, c'est trois ans. Un élément peut vous lâcher à tout moment, furtivement ou massivement, ce n'est pas vous qui choisissez, mais en plus le système d'exploitation d'un vieux PC est le plus souvent un facteur d'insécurité. Je parierai que si vous êtes encore en train de me lire, c'est que vous avez encore un PC qui tourne sous Windows XP... Plus question d'y installer un système d'exploitation récent. Il manque les pilotes, le matériel ne suit plus. Ce qui nous révèle leur 3^e gros inconvénient : les piétres performances.

Vous éviterez tout cela en vous construisant votre propre nouveau PC, spécialement pour votre labo d'électronique. Pour pas cher, vous aurez exactement ce qu'il vous faut.

Les composants du PC

Pour un électronicien, construire un PC lui-même ne devrait pas être une grosse aventure. Il faut savoir distinguer le rouge du bleu, le plus du moins, et ne jamais oublier la menace que constitue l'électricité statique. À part ça, il n'est pas bien sorcier de brancher les composants (**fig. 1**). Ça m'a pris à peine une soirée.

- carte mère
- CPU

- radiateur de CPU
- modules de RAM
- mémoire SSD (plus de disque dur !)
- alimentation
- boîtier

Ne pas oublier certaines petites pièces mécaniques comme les plaquettes de tôle avec les connecteurs des interfaces série et parallèle, celles-ci ne sont pas toujours fournies avec les cartes. Si vous avez des exigences graphiques particulières (3D, etc.), il faut prévoir une carte graphique en plus du processeur graphique intégré à l'unité centrale CPU.

Ce n'est pas le choix qui manque parmi les offres et vous auriez tort de vous passer des services d'un moteur de recherche de prix. Mais avant de plonger, munissez-vous de quelques biscuits, je veux dire informez-vous d'abord sur les composants individuels.

Carte mère : Intel ou AMD ? Que choisir ?

Commençons par la décision de principe : Intel ou AMD ? Le rapport qualité-prix des nouveaux processeurs d'AMD est meilleur. Cela met Intel sous pression et permet d'obtenir de meilleurs prix. Si vous n'utilisez que Windows 10, vous vous entendrez bien avec AMD. Moi qui installe sur plusieurs volumes du même PC non seulement Windows 10, mais aussi Linux et Mac OS, je n'ai pas le choix, c'est Intel.

S'il vous faut de l'espace pour des cartes enfichables, la prochaine décision portera sur le format de la carte en faveur de l'ATX standard. Le format µATX est possible aussi, mais c'est dans l'ATX qu'on trouve les dernières (?) cartes avec un connecteur PCI. C'est important s'il vous reste des cartes PCI

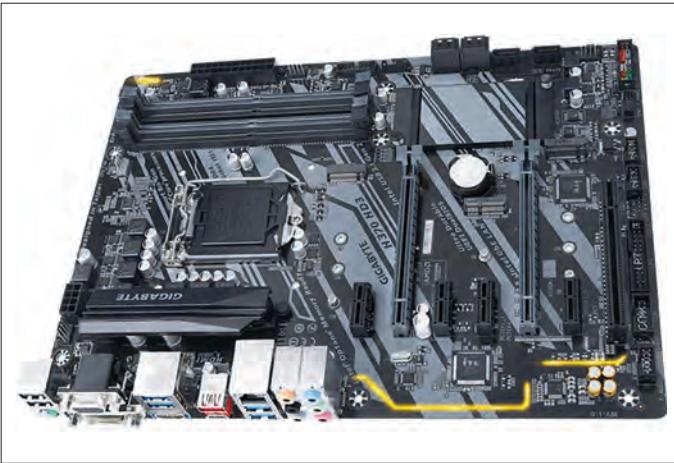


Figure 2 : Carte-mère avec le chipset H370 d'Intel pour le connecteur 1151-2 (source : Gigaocet [1]).



Figure 3 : Le i3-9100F d'Intel «en boîte» - un CPU à quatre cœurs, rapide, sans GPU

anciennes auxquelles vous tenez. Sinon, vous vous limitez aux cartes PC récentes.

Si vous avez opté pour un processeur Intel, vous aurez le choix entre des cartes avec plusieurs variantes des puces X3XX. Ce n'est pas le moment d'essayer de rabioter quelques euros ; évitez les puces H310, B360 et B365. Ces variantes pour PC de bureau sont limitées en nombre de voies PCIe utilisables, ce qui ralentit leur débit, en particulier lors de l'utilisation de SSD M2 NVME et de cartes graphiques. Il n'est pas nécessaire que ce soit une carte Z390 coûteuse. Les variantes H370 ou Z370 font l'affaire. Le «Z» initial indique que l'horloge CPU n'est pas fixée. Comme je ne souhaite nullement bidouiller l'horloge de ce PC de labo, une version bon marché du H370 fera l'affaire : le *Gigabyte H370 HD3* (**fig. 2**). Avec son connecteur PCI, on le trouve à partir de 100 €.

Unité centrale

Après avoir opté pour Intel ou AMD, le dilemme suivant est celui de la performance souhaitée, qui de nos jours dépend largement du nombre de cœurs. Intel propose des CPU modernes pour un peu moins de 50 € sous le nom de *Celeron* ou *Pentium*, qui font bien souvent l'affaire, il s'agit cependant de CPU à double cœur et aux épaules un peu frêles. En étant moins radin, vous pouvez vous offrir une unité centrale puissante et rapide dans la classe des 100 € en version à quatre coeurs.

Avec AMD, ce seraient les types Ryzen 3 ou 5 – évitez les CPU moins récentes et plus lentes ou même d'autres modèles que ceux équipés de socle AM4. Un choix raisonnable chez Intel commence avec le type i3-9100. Attention, les références qui finissent par «F» sont un peu moins chères, car le processeur graphique intégré est désactivé. Comme j'avais dans mes réserves une carte graphique AMD moderne récupérée lors de la mise à jour de mon PC principal, j'ai choisi le minimum rationnel : un i3-9100F emballé (*boxed*) pour 80 € (**fig. 3**), avec quatre cœurs et une horloge de base de 3,6 GHz (Turbo = 4,2 GHz). Il y a cinq ans, on déboursait facilement 400 € pour quelque chose de ce genre. Les CPU en vrac (*étiquetées tray*) sont à peine meilleur marché – pas assez pour m'inciter à renoncer à un modèle dans son emballage d'origine.

Quand paraîtra ce numéro d'*Elektor*, il se pourrait qu'il y ait déjà des successeurs. Auquel cas, je préférerais le type i3-10100.

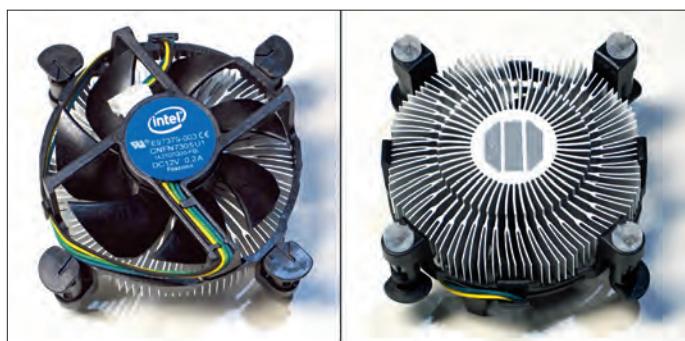


Figure 4 : Radiateur de CPU fourni dans la version «boxed»

Pour qui veut plus de cœurs, je trouve qu'avec le prix de ses CPU de type i5, i7 et i9, Intel n'y va vraiment pas avec le dos de la cuiller.

Radiateur de CPU

Une unité centrale i3 emballée est fournie avec un radiateur adéquat (**fig. 4**). Pas silencieux, mais supportable.

Plus votre unité centrale est gourmande en énergie, plus il faut la refroidir. On accepte qu'un PC de labo soit moins silencieux qu'un modèle de bureau, mais il faut au moins 20 € pour un radiateur avec caloducs et un ventilateur de 120 mm (**fig. 5**). Le choix est énorme. Nul besoin de prévoir un refroidissement à l'eau, puisque vous ne jouez pas au labo, n'est-ce pas ?

Mémoire vive

Combien de mémoire de travail ? Aujourd'hui, 16 Go est une capacité courante et presque toujours suffisante. Vous commandez donc un jeu de deux barres de 8 GB, pour permettre un accès entrelacé (*interleaved*). Si vous voulez 32 Go, un kit 2x16 consomme moins d'énergie qu'un kit 4x8.

Les processeurs utilisent depuis longtemps la mémoire DDR4. Ces modules à 260 broches sont nécessaires pour les CPU Ryzen à partir d'une fréquence d'horloge maximale de 2 933 MHz. Avec les processeurs *Coffee Lake* d'Intel (iX-8XXX et iX-9XXX), 2 400 MHz sont déjà suffisants. Il est toujours possible d'en faire



Figure 5 : Radiateur typique : Artic Freezer 34 avec quatre caloducs (source : Arctic [2]).

plus et de le débloquer via le BIOS (XMP = *eXtreme Memory Profile*), mais c'est aussi plus cher.

Outre l'horloge, la latence d'accès est une autre caractéristique décisive des modules de RAM, spécifiée par le code « CLXX-XX-XX-XX-XX ». Moins c'est mieux. Je suis resté sage en achetant pour seulement 55 € un kit *G.Skill* de la série *Value* siglé CL15-15-15-35 de 2x8 GB sous 1,2 V.

Mémoire de masse

Il y a deux critères ici : capacité et vitesse. À vous de voir combien il vous faut. Pour Win 10, et différents environnements de développement ainsi que les logiciels de CAO plus *Office*, 250 Go me suffisent largement.

La question de l'interface est plus décisive. Ma carte dispose de six interfaces SATA-3, leur débit de données n'est que de 600 Mo/s maximum. Or les modules de mémoire statique SSD avec interface M2 NVME sont meilleurs et pas tellement plus chers. Ma carte offre deux emplacements pour cela. Grâce aux quatre voies PCIe 3, il est possible d'obtenir un débit de données

brut jusqu'à 4 Go/s. Les vrais SSD offrent facilement 3 Go/s. Un critère moins évident est celui des opérations d'E/S possibles par seconde, car celles-ci sont décisives pour la réactivité des systèmes d'exploitation modernes avec leurs nombreux accès à la mémoire de masse. L'ACE-A80 de *Silicon Power* que j'ai choisi (**fig. 6**) offre 500/600 k/s (lecture/écriture), ce qui est cinq fois mieux que le SATA 3 habituel. Le débit de données de 3 400/3 000 Mo/s (lecture/écriture) ne doit pas non plus être négligé et se rapproche de la limite théorique. Puisqu'un tel SSD ne coûte que 45 €, toute envie de disque dur à l'ancienne m'est passée.

Un dernier détail : évitez les QLC-SSD ! Ces *Quad Level Cells* fonctionnent en fait avec 16 niveaux de tension par cellule et peuvent donc stocker 4 bits par cellule. C'est une solution intéressante, mais aussi boiteuse, et peu durable. Sans les acrobaties du contrôleur (mot-clé « *Wear Leveling* » et blocs de réserve), un secteur serait déjà instable après environ seulement 200 opérations d'écriture (contre 2000 avec le TLC). Le jeu ne vaut pas la chandelle !

Alimentation

Par souci de l'environnement, il convient ici d'accorder une attention particulière au rendement. Sous le label « 80 plus », les alimentations sont réparties dans les catégories bronze, argent et or. N'achetez rien qui ne soit pas au moins du bronze. La puissance fournie est également un critère crucial. Avec un processeur de 65 W, si vous utilisez le processeur graphique intégré, la consommation ne devrait guère dépasser 100 W même à pleine charge. Avec une carte graphique, il faut évidemment tenir compte de sa consommation. En principe, pour un PC de labo, pas besoin de modèle d'alim de haut de gamme de plusieurs centaines de watts comme en ont les joueurs. Un PC avec carte graphique de la classe moyenne inférieure devrait se cantonner à quelque 200 W. Quand il n'a pas grand-chose à faire, un PC de laboratoire ronronne entre 20 et 40 W. Il n'existe plus depuis longtemps d'alimentations ATX 250W valables. À puissance élevée, leur rendement n'avait d'ailleurs rien d'optimal. J'ai adopté un modèle de 350 W de LC-Power du type LC420-12 (**fig. 7**), un compromis satisfaisant, qui ne m'a coûté que 25 €. Ses connecteurs répondent à la norme ATX V2.31. Son rendement est de 88% et il est exemplaire en mode de veille. Le ventilateur intégré de 120 mm est assez silencieux (en général, plus le ventilateur est grand, plus il est silencieux).

Boîtier

L'habillage du PC est en grande partie une question de goût. J'ai retenu le modèle CS-1103 d'*Aerocool* : il ne coûte que 20 € mais offre deux prises USB 2.0 et une prise USB 3.0 à l'avant et en haut et sa tôlerie frontale est en alu noir (**fig 8**). Que demander de plus à ce prix ?

... mais aussi

La carte graphique mentionnée est une Radeon 560 de *Sapphire* avec 2 Go de RAM. C'est suffisant pour un moniteur 4K avec un rafraîchissement parfaitement stable à 60 Hz. À l'arrière du boîtier, j'ai rajouté un ventilateur de 90 mm à basse vitesse. Si vous souhaitez une ventilation vigoureuse, préférez un boîtier à plusieurs ventilateurs de 120 ou même 140 mm. J'ai aussi rajouté une plaque de tôle câblée pour les interfaces série et parallèle de la carte que je compte utiliser également. En



Figure 6 : NVME-SSD ACE-A80 Fast M2 de Silicon Power (source : Silicon Power [3]).

effet, l'émulation par l'USB d'une interface série pose souvent des problèmes en cas de transferts fréquents et rapides de quelques octets.

Montage

Si la plupart des composants sont sous emballage antistatique, ce n'est pas pour rien. Équiper votre poste de travail d'un bracelet de mise à la terre n'est pas un luxe. La charge électrostatique de votre corps varie selon que vous avez les mains plus ou moins propres et moites. Quand vous touchez un composant, faites-le sur ses plans de masse et évitez les revêtements de sol en plastique ou en fibres synthétiques.

Pour manipuler les cartes, tenez-les par les parties avec les interfaces, là où il y a beaucoup de métal avec un potentiel de terre. Avant de monter la carte principale dans le boîtier, il faut y installer la CPU que vous pouvez saisir d'une main tandis que de l'autre vous toucherez la carte aux interfaces. Avant d'ouvrir le levier avec la prise CPU, retirez le couvercle en plastique noir, puis insérez l'unité centrale correctement. Grâce aux détrompeurs, il est beaucoup plus difficile de la monter à l'envers qu'à l'endroit. Verrouillez ensuite la CPU, placez la tôlerie d'interface et vissez la carte. En général, il y a entre six et neuf vis, fournies avec le boîtier.

Maintenant le radiateur de la CPU. Il est généralement déjà recouvert d'une abondante couche de pâte thermique. Appuyez doucement le refroidisseur sur la CPU, en le déplaçant un peu pour bien étaler la pâte, puis retirez-le à nouveau. Avec un mouchoir en papier, essuyez l'excès de pâte sur l'unité centrale. Ainsi l'épaisseur de la pâte restée sur le radiateur devrait mieux s'adapter. En fait, pour simplifier, j'ai tout de suite retiré complètement la pâte d'origine et appliqué avec une spatule une couche fine d'une meilleure pâte contenant des particules métalliques.

C'est le tour du module SSD à présent et des deux réglettes de RAM. La notice de la carte indique comment les monter pour obtenir l'accès entrelacé. Il faut encore soigneusement et patiemment brancher tous les câbles du boîtier conformément aux instructions.

Viennent alors les fiches d'alimentation. Le connecteur à six broches est généralement celui de la carte graphique. Pour l'alimentation supplémentaire sous 12 V de la carte, il faut un connecteur à quatre ou huit broches. C'est très important ! Une éventuelle carte graphique sera logée dans le connecteur PCIe à 16 voies le plus proche de l'unité centrale.

Vérifiez tout puis allumez le PC. Sur le moniteur que vous aurez connecté, le BIOS doit apparaître. Windows 10 s'accommode des paramètres par défaut. D'autres systèmes d'exploitation nécessiteront un réglage fin. Il faut encore de la patience pour



Figure 7 : Alimentation LC420-12 de 350 W de LC-Power (source : LC-Power [4]).

installer tous les logiciels nécessaires. Ce n'est pas négligeable, car l'opération m'a pris plus de temps que l'assemblage de ce PC de laboratoire très bon marché. C'est subjectif, mais à l'usage il me semble presque aussi musclé que mon ordinateur principal beaucoup plus cher. ◀ (191191-02 VF)



Figure 8 : Dedans / dehors : le PC de labo fini.

Liens

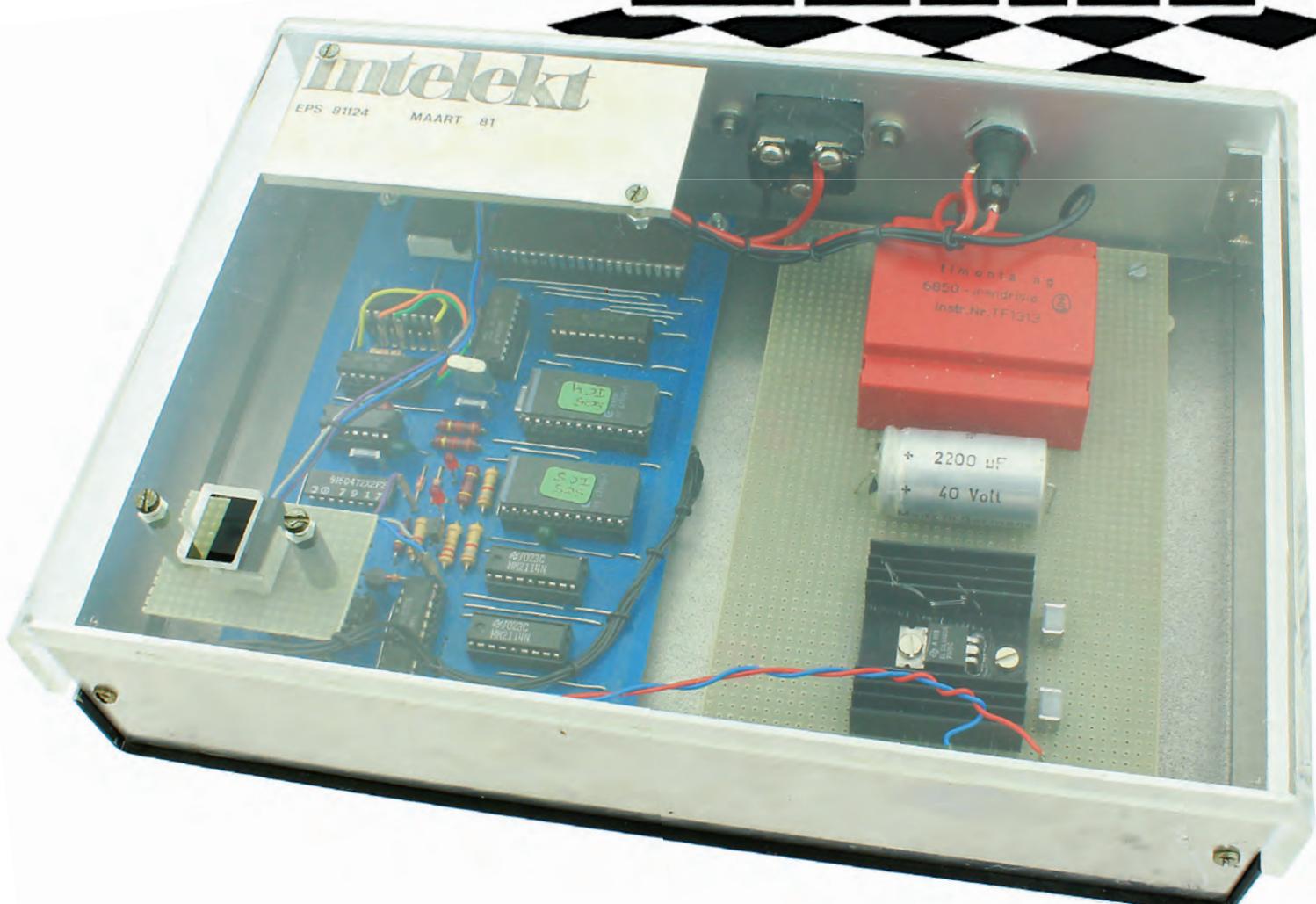
- [1] www.gigabyte.com/fr/Motherboard/H370-HD3-rev-10#kf
- [2] www.arctic.ac/eu_en/freezer-34.html
- [3] www.silicon-power.com/web/product-P34A80
- [4] www.lc-power.com/en/product/pc-power-supply-units/office-series/lc420-12-v231/

I'ordinateur de jeu d'échecs *intelekt* d'Elektor (1981)

Tiny Chess 86 porté sur l'Intel 8088

Jan Buiting (Elektor – Retronics)

« Savez-vous jouer aux échecs ? Êtes-vous toujours à la recherche du partenaire disponible 24 heures sur 24 ... qui ne s'impatiente jamais ... qui joue de façon honnête à un bon niveau ... qui ne dira rien si vous « traficotez » à la condition de le lui demander gentiment ? Si votre réponse est affirmative, il vous faut faire la connaissance de Intelekt ! »



L'invitation ci-dessus est tirée mot pour mot du magazine Elektor d'avril 1981. Sa réimpression ici est le fruit de l'heureuse péripétie du prototype d'*intelekt*, échappé intact de la razzia de juin 2006, quand Elektor a quitté ses bureaux historiques de Beek (au sud des Pays-Bas) pour emménager au château de Limbricht, toujours aux Pays-Bas. La dernière convulsion de ce déménagement a consisté, par une chaude après-midi de juin, à vider le laboratoire d'Elektor. Ce jour-là, une tonne de matériel de labo, de démo, de recueils de données, de magazines, d'originaux d'œuvres d'art, de circuits imprimés et de prototypes, tous jugés « dépassés » par la direction de l'époque, ont été balancés par deux fenêtres grand ouvertes pour finir dans une benne à ordures. Certains des prototypes du labo toujours présentables, planqués dans des recoins, ont heureusement été sauvés de justesse de la destruction. Depuis ils constituent la petite collection *Retronics*, sauvegardée dans le grenier du château une dizaine d'années jusqu'au déménagement suivant, à Aix-la-Chapelle en Allemagne.

J'ignore qui a sauvé ce prototype d'*intelekt* de la décharge, mais il doit sans doute sa survie à son magnifique coffret. Après 30 ans, il est toujours fringant, avec son couvercle en acrylique transparent conçu pour les nombreux salons où Elektor exposait ses créations à travers toute l'Europe. Dans les années 80, trois personnes travaillaient quasiment à temps plein sur la préparation de ce matériel d'exposition. Les lecteurs attentifs auront remarqué la minuscule initiale, conforme à la charte graphique d'*elektor* dont le nom s'écrivait aussi avec un **e** minuscule.

Présentable, il l'était vraiment

Le boîtier pour l'*intelekt* est fait non seulement pour être montré à un vaste public, mais conçu pour survivre à des hordes de visiteurs passionnés, impatients de toucher enfin de leurs propres mains ces appareils qu'ils voyaient en photo dans le magazine ! Lors de ces foires, nos lecteurs affluaient sur le stand d'Elektor. Contrairement aux salons aseptisés d'aujourd'hui, l'électronique était une discipline vivante. Les files d'attente n'étaient pas rares, les cris d'enthousiasme non plus, les circuits imprimés et les livres se vendaient par centaines en quelques jours, et les discussions autour des modèles exposés se poursuivaient sur le stand d'Elektor jusqu'à l'heure de fermeture et au-delà. Les lecteurs avaient hâte de rencontrer la rédaction d'Elektor, pour faire la connaissance des personnes qui produisaient cette publication unique en son genre venue de Hollande — et faire de bonnes affaires, bien sûr.

Revenons à l'*intelekt*. Un détail nous remet aussitôt dans l'ambiance des salons : le poussoir encastré est la commande d'interruption. Le rôle du guichet de 12 × 17 mm dans le couvercle en acrylique est de barrer la route aux doigts baladeurs d'un visiteur espionné tenté de déranger une partie d'échecs en cours de démonstration ! Ce *Digitast* est disposé à quelques centimètres sous le couvercle en plexi, sur un bout de circuit à pastilles (fig. 1).

Autre détail caractéristique : l'étiquette avec la mention « *intelekt* », faite de deux rectangles d'acrylique astucieusement interchangeables en fonction de la langue du pays dans lequel se tenait le salon où l'appareil était présenté (fig. 2) : la version néerlandaise (à droite), sobre, avec juste le titre et le mois de publication, et à gauche la plaque allemande, où l'on reconnaît un bout de la couverture du numéro d'avril 1981 d'Elektor.

ESTP 2004

www.elektor.tv



Merci de télégraphier à redaction@elektor.fr vos suggestions de sujets à traiter.

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

Seul le couvercle est en plexi transparent (3 mm), conformé à chaud sur une machine faite maison. Ce matériau à l'épreuve des chocs remplissait bien son rôle : susciter chez les visiteurs des salons la fringale d'électronique. Le fond du boîtier est métallique avec fusible, connecteur secteur IEC, interrupteur marche/arrêt et E/S série sur une prise DIN à 5 voies à l'arrière (fig. 3).

16 bits, ô merveille !

Dans son numéro d'avril 1981, Elektor présentait pour la première fois les µC à 16 bits, tous nouveaux et encore inconnus de la plupart des lecteurs, dont certains étaient familiers en revanche des microprocesseurs à 8 bits comme le 6502



Figure 1. Le boîtier de l'*intelekt* sans son couvercle. Le poussoir est de type Digitast qui offrait un subtil retour tactile. On le voit ici soudé sur un petit morceau de carte à pastilles, sur deux entretoises et connecté à la carte par des fils.

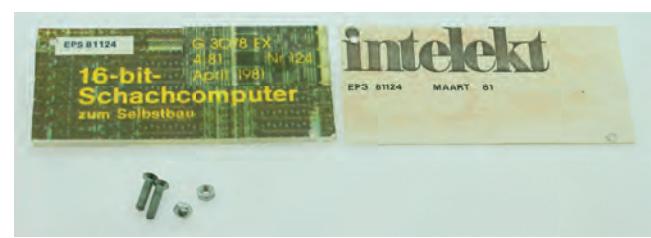


Figure 2. Les deux versions de la plaque d'identification sont interchangeables selon la langue, néerlandais (à droite) ou allemand (à gauche), du pays dans lequel *intelekt* était présenté.

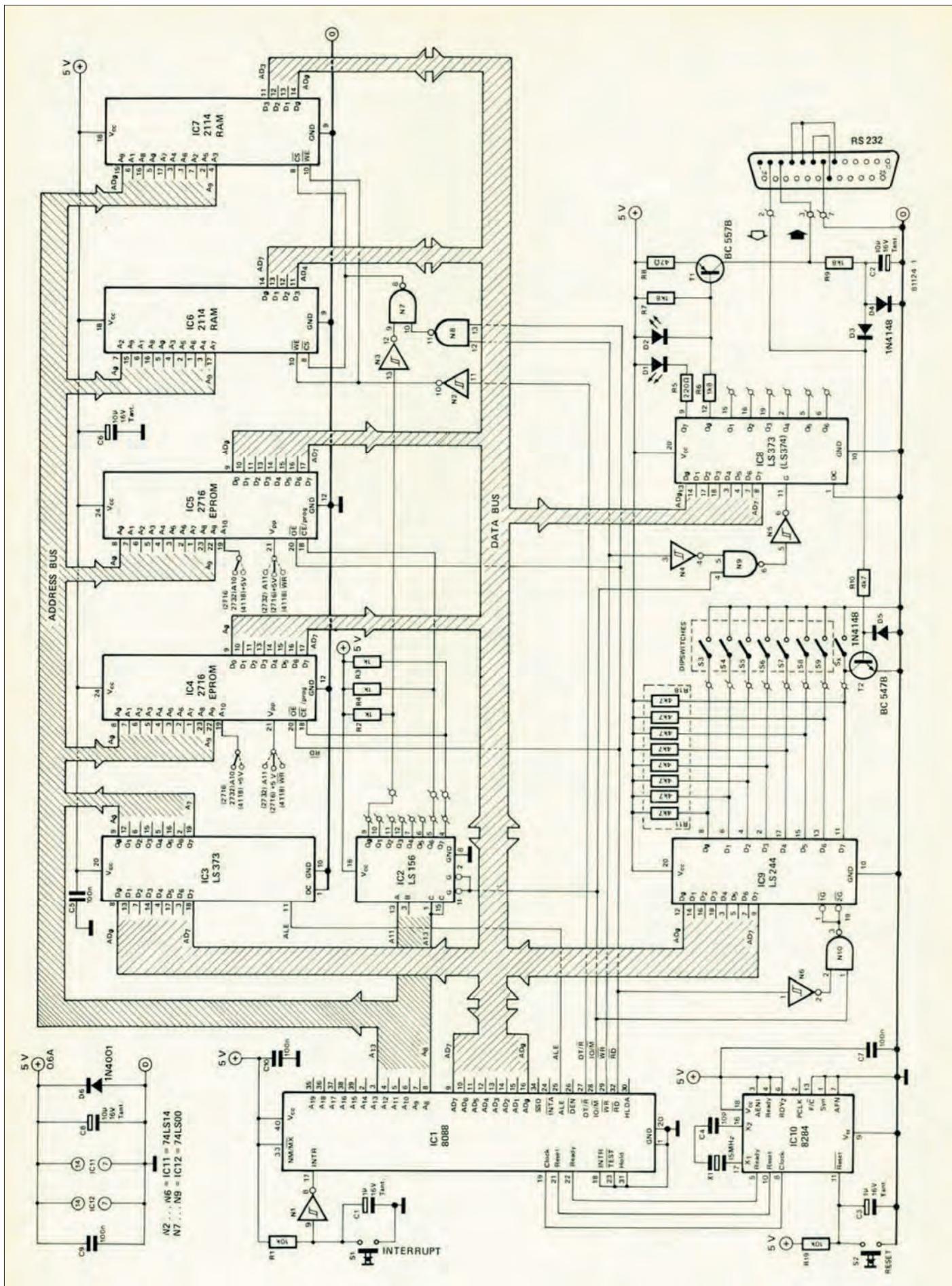


Figure 4. Numérisation du schéma de l'intelekt imprimé dans le magazine Elektor d'avril 1981.

et le Z80. Les puces à 16 bits comme le 8088 d'Intel, c'était pour les systèmes professionnels et industriels. Ce numéro en montrait avant tout l'architecture.

L'article sur *l'intelekt* (en mai 1981) explique longuement (à un public familier des 8 bits) comment transmettre le deuxième octet pour former un mot de données de 16 bits en sortie du 8088 (l'unité centrale du PC IBM originel). Il fallait d'ailleurs un gros effort d'imagination pour comprendre comment traiter 16 bits en parallèle avec si peu de broches disponibles sur la puce. La réponse est : *intelligemment*, en les multiplexant avec le bus d'adresses à l'aide du signal de synchro du processeur. La suite est aujourd'hui de la vieille histoire, puisque l'architecture 8088 n'a pas convaincu Elektor, où l'on s'est montré plus enclin à frayer avec les 8086 et 8085 (à 8 bits !).

L'article s'efforce de montrer qu'un automate joueur d'échecs ne peut se passer des 16 bits pour le traitement des données. Dans sa version anglaise, il fait d'ailleurs référence à des publications antérieures mais prospectives d'Elektor : *How I beat the monster* de David Levy et *Computers and Chess*. Il s'agissait visiblement pour Elektor de consacrer en grande pompe l'heureuse union d'un processeur Intel à 16 bits et du noble jeu d'échecs.

Les racines

L'intelekt est assez célèbre pour être mentionné sur le Wiki consacré à la programmation du jeu d'échecs [1]. On y lit qu'*intelekt* était une version adaptée au 8088 de *Tiny Chess* écrite pour le 8086 par Jan Kuipers. Le TC86 a été écrit en code assembleur (!). L'échiquier *intelekt* d'Elektuur (le nom néerlandais d'Elektor) aurait connu son inauguration publique lors de la compétition néerlandaise ouverte d'échecs par ordinateur (DOCCC) en 1981. L'exemplaire décrit ici était probablement de la partie !

Si j'ai retrouvé la piste de *Tiny Chess*, c'est grâce à une discrète mention dans l'article de 1981, à propos de la communication avec *intelekt*. Je cite :

Intelekt répondra en affichant le message suivant :

```
TINY CHESS V1.0 (sic)
LEVEL IS 1 CHANGE TO _
```

Sans ce détail, je n'aurais jamais su d'où sortait le logiciel de ce projet. Il occupait deux EPROM 2716.

Intelekt communique ses mouvements et lit les vôtres par une liaison sérielle à 3 fils vers un terminal passif muet.

L'insaisissable logiciel *intelekt*

Pendant toutes ces années, les micrologiciels utilisés dans les projets avec µP (puis µC) d'Elektor étaient fournis par le service logiciel d'Elektor (ESS) sous forme d'EPROM. Or, il n'y a pas un mot à ce sujet dans l'article de 1981, ni dans la liste des composants sur la disponibilité des deux EPROM 2716. L'auteur de l'article reste évasif :

« Dans cet article, nous ne ferons état que de la partie 'matériel' (schéma et circuit imprimé) ; nous ne donnerons aucune information sur le logiciel du programme d'échecs. Mais nous parlerons longuement de ses qualités — car finalement, c'est de cela qu'il s'agit. »

Aujourd'hui, à l'heure du code ouvert, ce serait un scandale. J'ai parcouru toutes les listes ESS/EPS de l'ensemble du volume

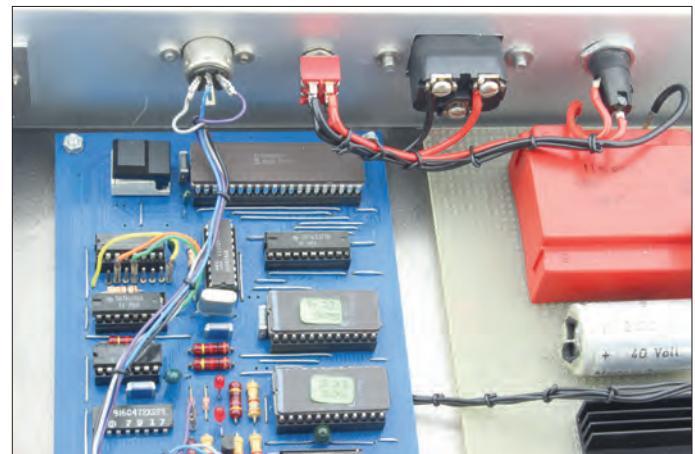


Figure 3. Les composants installés à l'arrière du boîtier de *l'intelekt* normalement inaccessibles au public. Notez également l'espace réservé sur le circuit imprimé pour le poussoir INT que l'on peut aussi câbler en externe.

1980-1989 d'Elektor et n'ai trouvé aucun logiciel lié au projet *intelekt*, seulement le circuit imprimé, disparu après 3 ans. Pas de trace du 505 écrit à la main sur les étiquettes des deux 2716 d'*intelekt*. Je subodore des restrictions de droits d'auteur ou contractuelles comme il en arrive de temps à autre. Curieusement, j'ai vu apparaître dans les pages de publicité des numéros suivants des annonces d'un kit *elektor intelekt complet avec le logiciel*.

Au fait, savez-vous que le contenu intégral du magazine Elektor (en anglais) de 1980 à 1989 est disponible sur DVD ? Une version française est en préparation.

Le matériel

L'auteur de l'article *Intelekt*, signé Jan Kuipers, un pionnier de la programmation du jeu d'échecs, avait déjà expliqué qu'il fallait une chronologie d'horloge complexe pour que le µP 8088 exerce sa magie avec le surplus de 8 bits de données. On reconnaît la manière d'Intel à la puce d'oscillateur d'horloge séparée, ici un 8284 qu'on voit sur le schéma (**fig. 4**) et sur

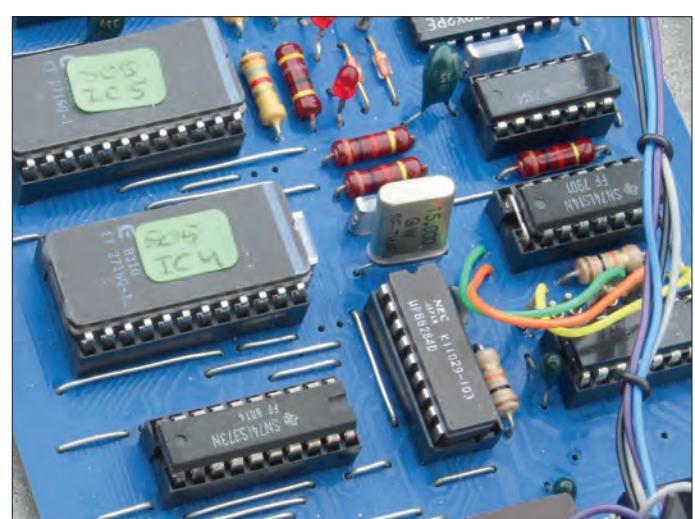


Figure 5. Le 8088 n'a pas de générateur d'horloge intégré, il lui faut un circuit spécial, le 8284.

Bouton poussoir Digitast

L'article de 1981 présente aussi un élégant bouton à monter sur circuit imprimé, appelé *Digitast*. Ces composants mécaniques étaient très prisés au labo d'Elektor dans les années 80. Ils n'avaient pas que des qualités.

Leur prix les mettait hors de portée des amateurs impécunieux qui préféraient investir dans des composants, avec un meilleur rapport performance/prix.

Le nom *Digitast* est probablement un mot-valise allemand composé sur *digital* et *Taster*, où *Taster* signifie *bouton*. L'*intelekt* en utilise deux, *Reset* et *Interrupt*. Ici seul *RES* se trouve sur le PCB, tandis que *INT* y est relié par des fils et fixé en retrait sous le couvercle. L'ennui



c'est que tous les *Digitast* ont la même empreinte, mais le brochage de certains diffère par une liaison interne que les autres n'ont pas. Donc, si vous utilisez un *Digitast* non monté sur la carte, il y a une liaison à rajouter pour assurer une bonne connexion. À défaut de quoi votre 8088 risque de ne pas fonctionner en dépit des heures passées à le déboguer, que ce soit en 8 ou en 16 bits ! Heureusement, cette bizarrerie électromécanique est mentionnée dans l'article, tout comme la nécessité de souder pas moins de 43 points de câblage sur le circuit imprimé à simple face.

la carte (**fig. 5**), près du quartz de 15 MHz. Le prix de cette puce (à 18 broches) était disproportionné par rapport au 8088 dans son boîtier DIP à 40 broches.

Le schéma arbore fièrement un connecteur sub-D à 25 broches pour la liaison série avec le terminal et un « RS232 » (sic) imprimé à côté, cette interface ne mérite pas ce nom, elle n'a que deux lignes, RxT et TxT, aucun signal d'acquittement, et une amplitude restreinte de 0 à 5 V au lieu de ±10 V. De tels ersatz de RS-232 avec un BC54x pulluleront dans Elektor, au désespoir de lecteurs équipés de terminaux professionnels et d'autres appareils conçus pour le vrai RS-232, qui ne communique pas avec ces projets de %#^¥<@. C'était bien avant l'arrivée du MAX232 et consorts avec leurs pompes de charge pour assurer l'alternance négative.

Affichage spartiate

Intelekt n'est pas un ordinateur d'échecs graphique. Après le message d'accueil, il affiche simplement la situation initiale de l'échiquier sur l'écran du terminal, dans ce cas l'Elekterminal et ses 20 lignes de 40 signes. La représentation des pièces avec des croix et des points est spartiate, et je pense que la plupart des joueurs se passeraient de l'écran, liraient les coups sous forme de texte et déplaceront les pièces sur un véritable échiquier. Pour saisir votre coup, vous tapez :

- la case contenant la pièce à déplacer
- un espace
- la case de destination
- CR (retour chariot)

La réponse d'*intelekt* à votre mouvement peut prendre de 25 s au niveau 1, le plus bas jusqu'à, euh, disons... peut-être 1 h au niveau 8, son niveau le plus élevé. Voici un exemple de dialogue personne-machine :

01W: e2 e5 — mouvement illégal, alors :

01W: e2 e4 (CR)

01B: c7 c5

02W: (en attente du mouvement suivant)

Le logiciel connaît quelques commandes de configuration de jeu comme le changement de joueur (CTRL-X), l'*autoplay* (CTRL-

A) et le réglage du niveau de jeu (CTRL-N).

La seconde moitié de l'article de 1981 est une longue section sur un exemple de jeu d'échecs joué contre *intelekt*, commenté et analysé, avec des listes de coups, des escarmouches palpitan tes et des commentaires. L'auteur suggère de modifier le logiciel dans le but d'améliorer la fin de partie et de permettre l'échange de pions victorieux contre d'autres pièces qu'une dame. Non sans avertir le lecteur que toute modification du logiciel implique l'accès à une mémoire de plus et ralentit la réponse. On n'était visiblement pas à une incohérence près, puisque la carte *intelekt* est prévue en fait pour des EPROM plus grandes, de type 2732.

Il faut se remettre dans l'ambiance de l'époque pour comprendre que cet article essayait d'éclairer le lecteur sur les points forts et les faiblesses du premier ordinateur autonome d'Elektor réservé à un usage spécifique, le jeu d'échecs. Il se félicite des temps de réponse raisonnables et du choix des 16 bits pour s'affranchir des défauts d'une procédure simpliste : *intelekt* était conçu comme un adversaire, pas comme un penseur solitaire du jeu d'échecs.

Un de ces jours je remettrai sous tension notre *intelekt* pour voir s'il fonctionne toujours, éventuellement contre un adversaire à 32 bits. Auparavant, il faudra que je remplace tous ces condensateurs au tantale, avec lesquels *intelekt* risque d'exploser avant même le début de la partie. ─

(190382-B-02 Version française : Robert Grignard)

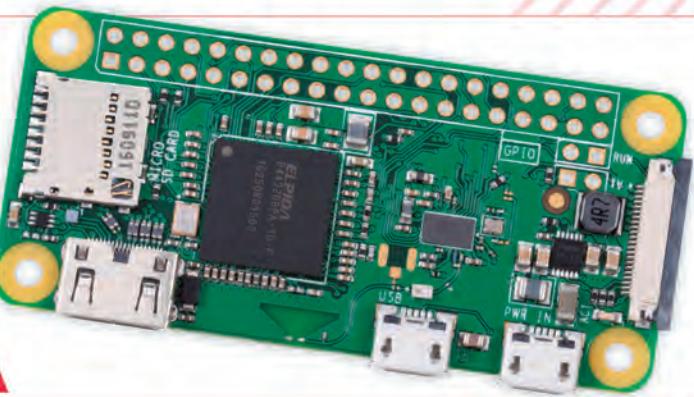
@ WWW.ELEKTOR.FR

→ DVD Elektor 1980 through 1989
www.elektor.fr/dvd-elektor-1980-through-1989

A photograph of a DVD cover for "Elektor 1980 through 1989". The cover is blue and features the Elektor logo at the top. Below it, there's a grid of small thumbnail images showing various projects and articles from the period. The text "1980 through 1989" and "10 YEARS OF ELEKTOR ON DVD" is prominently displayed.

ABONNEZ-VOUS ET RECEVEZ

RPI ZERO W GRATUIT



SEULEMENT
54,95 €
PAR AN
(6 NUMÉROS)

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



Souscrivez dès maintenant un abonnement d'un an au magazine MagPi, nous vous offrons :

- Six numéros du magazine MagPi
- Une carte Raspberry Pi Zero W
- Un boîtier avec trois couvercles différents
- Un connecteur pour module de caméra
- Un câble HDMI/mini-HDMI et un câble micro-USB/USB OTG

Vos avantages :

- Prix au numéro réduit
- Chaque numéro directement dans votre boîte aux lettres
- Tous les numéros disponibles sous forme numérique (PDF)
- Cadeau de bienvenue d'une valeur de 22,95 €
- Découverte de chaque nouveau numéro avant sa sortie en kiosque

ABONNEZ-VOUS : WWW.MAGPI.FR

bienvenue dans votre e-choppe

la rédaction recommande



Joy-IT DMS02D72 oscilloscope portable +multimètre + génér 3 en 1

Avec autant de fonctions réunies dans un seul appareil, on se demande quelle en est la principale. Il s'agit bien sûr de l'oscillo, en l'occurrence une version à deux voies avec une bande passante de 70 MHz et un taux d'échantillonnage de 250 Méch/s (la moitié lorsque l'on utilise 2 canaux simultanément). Ce sont de très belles spécifications pour un instrument de mesure de moins de 200 € (pour les membres d'Elektor). En plus de cela, c'est aussi un générateur de formes d'ondes arbitraires aux propriétés remarquables (taux d'échantillonnage de 250 Méch/s) et un multimètre à sélection de calibre automatique avec des entrées séparées. Que demander de plus ?

Harry Baggen (labo d'Elektor)

www.elektor.fr/190157



Percer les mystères de l'électronique

IoT Home Hacks with ESP8266

Vos favoris :

1. livre Microcontroller Basics with PIC

www.elektor.fr/19188



2. livre Learning Python with Raspberry Pi

www.elektor.fr/19106

3. Raspberry Pi 4 B (4 GB)

www.elektor.fr/18964

4. kit Elektor SDR Hands-on

www.elektor.fr/19041

5. livre Percer les mystères de l'électronique

www.elektor.fr/19080

6. livre Les oscilloscopes anciens & modernes pour les débutants

www.elektor.fr/19124



Elektor Ultimate Sensor Kit



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.

Les plateformes dites « compatibles Arduino » ne manquent pas. Une des plus réussies est l'ESP8266 sous la forme du Wemos D1 Mini Pro. Cet appareil remarquable offre un accès internet WiFi et un système de fichiers flash utilisant jusqu'à 16 Mo de mémoire flash externe. En plus, vous disposez de nombreuses broches d'entrée/sortie (dont une analogique), PWM, I²C, et one-wire. Exactement ce qu'il faut pour construire facilement de nombreux petits dispositifs IdO !

Prix (membres) : 33,75 €

www.elektor.fr/19080

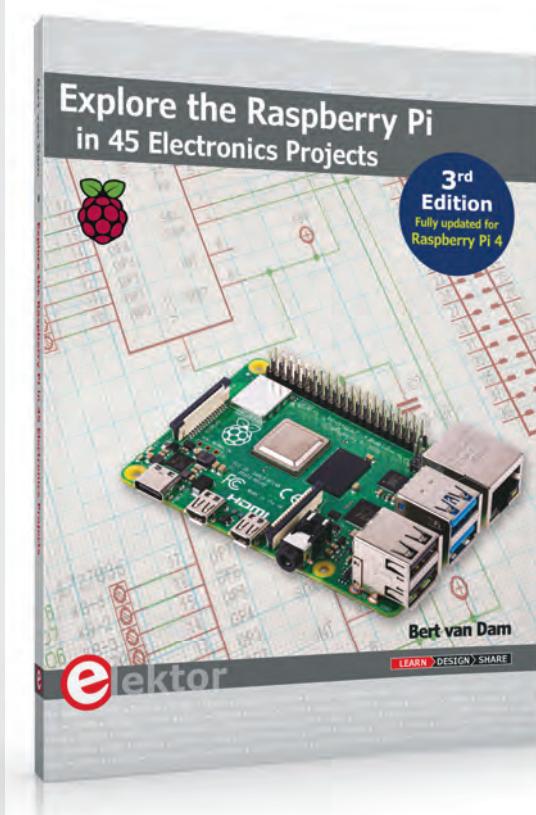
Prix (membres) : 31,46 €

www.elektor.fr/19158

Cet ensemble réunit un kit de capteurs de haute qualité et un livre de projets complet. Il a été conçu spécialement pour vous guider dans l'utilisation de capteurs sur plusieurs plateformes de microcontrôleurs à code source ouvert, notamment Arduino, Raspberry Pi et ESP32. L'ensemble contient en tout 40 capteurs différents mis en œuvre dans des projets variés. Tous les projets décrits sont accompagnés d'exemples de logiciels.

Prix (membres) : 71,96 €

www.elektor.fr/19104



nouveau

Explore the Raspberry Pi in 45 Electronics Projects (3rd Edition)

Dans la nouvelle édition (3^e) entièrement révisée de notre best-seller *Explore the Raspberry Pi in 45 Electronics Projects* le lecteur trouve la description de 45 projets. Des moyens et des petits : feu clignotant alternatif, commande de moteur, production et traitement de signaux analogiques, thermomètre numérique, photomètre. Mais aussi des grands : commande de vitesse de moteur, serveur web avec CGI (*Common Gateway Interface*).

Après une brève présentation du Raspberry Pi, vous installez le logiciel requis. La carte SD qui peut être achetée en même temps que ce livre contient tout ce qu'il faut pour commencer à utiliser le Raspberry Pi. Le livre se poursuit par une introduction concise au système d'exploitation Linux, après quoi vous commencez à programmer en Bash, Python 3 et Javascript.



Prix (membres) : 31,46 €
www.elektor.fr/19190

Microcontroller Basics with PIC

Les oscilloscopes anciens & modernes pour les débutants

Learning Python with Raspberry Pi



L'auteur présente tous les aspects essentiels de la programmation des microcontrôleurs, sans surcharger le lecteur d'informations inutiles. Après avoir lu ce livre, vous devriez être capable de comprendre et de programmer des microcontrôleurs 8 bits. L'initiation à la programmation s'appuie sur des microcontrôleurs de la série PIC.

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.

Ce livre traite de l'apprentissage du langage Python à l'aide de l'ordinateur Raspberry Pi 4 pour les électroniciens. Après la présentation de Raspberry Pi 4 il explique les sujets suivants du langage Python : variables, chaînes de caractères, tableaux, matrices, tuples, listes, dictionnaires, fonctions utilisateur, flux de contrôle, impression, saisie au clavier, graphiques, interface graphique, programmation orientée objet et bien d'autres.

Prix (membres) : 31,46 €

www.elektor.fr/19188

Prix (membres) : 33,75 €

www.elektor.fr/19124

Prix (membres) : 31,46 €

www.elektor.fr/19106

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 **5 avril 2020** à l'adresse **hexadoku@elec**

Les gagnants

La solution de la grille du numéro de janvier/février 2020 est **C7A13**.

Les cinq bons Elektor d'une valeur de **50 €** vont à : Francis Biette (France) – Annie Tigchelaar (Pays-Bas) – David Turnbull (Royaume-Uni) – Sabine Lamprecht (Allemagne) – Gabi Kirchhof (Allemagne).

Bravo à tous les participants et félicitations aux gagnants !

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

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.

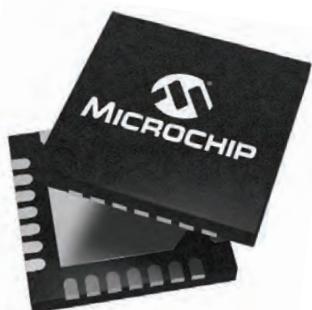


Défendez votre propriété intellectuelle, votre marque et votre chiffre d'affaires

Des solutions de sécurité faciles à ajouter et difficiles à pirater

Laissez Microchip vous aider à sécuriser non seulement vos systèmes, mais aussi votre marque et votre chiffre d'affaires. Forts de deux décennies d'expérience dans la sécurité, nos experts vous permettent d'intégrer la sécurité à vos systèmes sans aucune appréhension et en vous passant d'une expertise coûteuse en interne. Combinez cette expertise avec nos sites de production sécurisés et nos services d'approvisionnement et vous comprendrez pourquoi tant d'entreprises font confiance aux experts de Microchip pour les guider dans la conception de leurs systèmes.

Du chiffrage sécurisé jusqu'aux environnements d'exécution de confiance, trouvez les mises en œuvre de sécurité qui répondent à vos besoins spécifiques grâce à notre large éventail de solutions logicielles et matérielles.



Sécurisez votre système en allant sur www.microchip.com/Secure

Le nom et le logo de Microchip ainsi que les logos de Microchip sont des marques déposées de Microchip Technology Incorporated aux États-Unis et dans d'autres pays. Toutes les autres marques commerciales ci-dessus mentionnées sont la propriété exclusive de leurs propriétaires respectifs.
© 2019 Microchip Technology Inc. Tous droits réservés. DS00002767A. MEC2236Fre03/19





electronica
fast forward
the startup platform

powered by elektor

Gagnez jusqu'à
75 000 €
de plan marketing
d'Elektor et un stand au
salon **electronica**

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

elektormagazine.fr/fastforward

electronica Fast Forward is brought to you by



electronica

